

Seminararbeit: Das Spiel Brainball aus der Sicht der Informatik

Peer Steldinger

Seminar: Optimierung in der Praxis
(Professor Lang)

WS2000/2001

ACHTUNG! BrainBall ist eingetragenes Warenzeichen der BrainWay Gesellschaft für Marketing und Vertrieb mbH. In der folgenden Arbeit wird der Name „Brainball“ der Kürze halber ohne Copyright-Hinweis verwendet. Die in dieser Arbeit verwendeten Graphiken wurden freundlicher Weise von Andreas Unsicker, dem Entwickler des Brainballs zur Verfügung gestellt.

Inhaltsverzeichnis

1	Einleitung	2
1.1	Ziel dieser Arbeit	2
1.2	Was ist ein Brainball?	2
2	Modellierung	3
2.1	Grundlegende Definitionen	3
2.2	Rochaden	5
3	Die Aufgabenstellungen	6
3.1	Logikspiel	6
3.2	Strategiespiel	6
3.3	Wettkampfspiel	7
3.4	Große Strategie-Aufgabe	7
4	Kombinatorik	9
4.1	Wie viele Zustände gibt es?	9
4.2	Erreichbarkeit	11
5	Berechnen von Rochaden	13
5.1	Farbrochaden	13
5.2	Zahlenrochaden	14
5.3	Kürzungsregeln	14
6	Die große Strategie-Aufgabe	17
6.1	Breitensuche / Dijkstras Algorithmus	17
6.2	Tiefensuche	18
6.3	Iterativ begrenzte Tiefensuche (Iterative Deepening)	19
6.4	A^* -Algorithmus	20
6.5	Bidirektionale Suche	21
6.6	Inselbasierte Suche	21
7	Eigenschaften des Suchgraphen	22
8	Ausblick	24

1 Einleitung

1.1 Ziel dieser Arbeit

In dieser Seminararbeit beschäftige ich mich mit einem Logikspiel mit dem Namen Brainball. Dieses Spiel ist - grob - vergleichbar mit dem bekannten „Rubik's Cube“. Man kann den Brainball in sehr viele Zustände transformieren. Eine mögliche Aufgabenstellung ist zum Beispiel, in möglichst wenigen Schritten einen bestimmten Zustand zu erreichen. Seit ich mir im Herbst 1999 einen Brainball gekauft habe, habe ich mich auf verschiedene Weise informatisch und mathematisch mit dem Brainball beschäftigt. Ziel dieser Arbeit ist es, die interessantesten Ansätze und Ergebnisse zusammenzufassen und einen Ausblick auf dabei aufgetretene, noch ungeklärte Fragen zu geben. Dabei werden unter anderem graphentheoretische Aspekte und auch die „Kürzeste-Pfad-Suche“ in sehr großen Graphen betrachtet.

1.2 Was ist ein Brainball?

Der Brainball besteht aus einem Zahlenkranz mit 13 Spielsteinen und einer Kugel in der Mitte.

Die Spielsteine auf dem Kranz sind von 1 bis 13 durchnummeriert und besitzen auf jeder Seite eine andere Farbe (aber dieselbe Zahl).



Die Kugel besteht aus drei Segmenten. Die beiden äußeren (blauen) Segmente sind verschieden groß (3 vs. 4) und sind mechanisch miteinander verbunden. Sie lassen sich (nur gleichzeitig!) um die Mittelachse (rot) kippen. Im folgenden verwenden wir die Begriffe *kleines*, beziehungsweise *großes Kippsegment* und *statisches Segment*.

Bei dem Brainball gibt es drei grundlegende Spielzüge. Zum einen kann man den Zahlenkranz verschieben und zum anderen kann man mit Hilfe der blauen Segmente Teile des Zahlenkranzes kippen, wodurch sowohl die Position der betroffenen Spielsteine als auch deren Farbe auf der „Vorderseite“ geändert. Ausserdem kann man noch den gesamten Brainball umdrehen, da eigentlich keine der beiden Seiten des Brainballs als Vorderseite ausgezeichnet ist. Mit diesen grundlegenden Spielzügen gilt es nun, den Brainball umzuformen.



2 Modellierung

Für viele der folgenden Betrachtungen ist es nötig, den Brainball durch mathematisch formale Definitionen zu modellieren.

2.1 Grundlegende Definitionen

Sei $\mathbb{N}_{13} := \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13\}$ und $\Gamma := \{\text{weiß, gelb}\}$ (Im Folgenden sei $\neg\text{weiß} := \text{gelb}$ und $\neg\text{gelb} := \text{weiß}$). Als *Zustand* eines Brainballs bezeichnen wir ein Tupel (π, γ) mit einer Permutation π der Menge \mathbb{N}_{13} und einer Abbildung $\gamma : \mathbb{N}_{13} \rightarrow \Gamma$. π beschreibt, welche Zahl an welcher Stelle des Zahlenkranzes steht. Dabei bezeichnet $\pi(1)$ (=: Basiszahl) den Stein, der in der Mitte des kleinen Kippsegments liegt, $\pi(2)$ bis $\pi(13)$ bezeichnen entsprechend die im Uhrzeigersinn folgenden Steine. $\gamma(i)$ beschreibt jeweils die oben liegende Farbe des Steins mit der Nummer i . Wir notieren einen solchen Zustand in Zukunft als mit der Basiszahl beginnende Permutationsfolge von Zahlen aus \mathbb{N}_{13} , wobei eine Zahl p jeweils genau dann überstrichen wird, wenn $\gamma(p) = \text{gelb}$ gilt: Beispiel:



$$\simeq (8, 9, \overline{13}, \overline{12}, \overline{11}, \overline{10}, 1, 2, 3, \overline{6}, \overline{5}, \overline{4}, 7)$$

$\mathbb{B} := S_{13} \times \Gamma^{13}$ sei nun die *Zustandsmenge*, also die Menge aller denkbaren Zustände. Auf \mathbb{B} lassen sich nun die Wirkungen der drei grundlegenden Spielzüge als unäre Funktionen definieren:

Sei $z = (\pi, \gamma) \in \mathbb{B}$ ein beliebiger Zustand.

Zahlenkranz verschieben:

Wir definieren eine Funktion w_1 , die der Wirkung des Verschiebens des Zahlenkranzes um eine Position im Uhrzeigersinn entspricht:

$w_1 : \mathbb{B} \rightarrow \mathbb{B}$, $w_1(z) := (\pi', \gamma')$ mit

$$\pi'(i) := \begin{cases} \pi(i-1) & \text{für alle } i \in \{2, 3, 4, \dots, 12, 13\} \\ \pi(13) & \text{für } i = 1 \end{cases}$$

$$\gamma' := \gamma$$

Nun können wir durch Hintereinanderausführung auch die Wirkung mehrschrittiger Drehungen definieren:

$$\forall i \in \mathbb{Z}_{13} : \quad w_i : \mathbb{B} \rightarrow \mathbb{B}, w_i(z) := (w_1(z))^i$$

Da insbesondere $z = (w_1(z))^{13}$ gilt, erweitern wir die Definition auf alle ganzen Zahlen:

$$\forall j \in \mathbb{Z} : \quad w_j : \mathbb{B} \rightarrow \mathbb{B}, w_j(z) := (w_1(z))^j \text{ mit } i \in \mathbb{Z}_{13}, j \equiv i \pmod{13}$$

In Zukunft beschränken wir uns – in Anlehnung an die in der Brainball-Spielanleitung verwendete Notation – nur die ganzen Zahlen von -6 bis 6 als Schiebeindex. Die *inverse Funktion* w_i^{-1} zu einer Funktion w_i ist gegeben durch $w_i^{-1} := w_{-i}$, da $\forall i \in \mathbb{Z} : w_{-i}(w_i) = id_{\mathbb{B}}$ gilt.

Kippsegmente kippen:

Die Wirkung des Kippens der beiden blauen Kugelsegmente um 180 Grad wird beschrieben durch die Funktion $w_j : \mathbb{B} \rightarrow \mathbb{B}, w_j(z) := (\pi', \gamma')$ mit

$$\begin{aligned} \pi'(i) &:= \begin{cases} \pi(i) & \text{für alle } i \in \{1\} \cup \{3, 4, 5\} \cup \{12, 11, 10\} \\ \pi(15 - i) & \text{sonst} \end{cases} \\ \gamma'(\pi(i)) &:= \begin{cases} \gamma(\pi(i)) & \text{für alle } i \in \{3, 4, 5\} \cup \{12, 11, 10\} \\ \neg\gamma(\pi(i)) & \text{sonst} \end{cases} \end{aligned}$$

Man sieht leicht, dass $z = (w_j(z))^2$ gilt (Kippen um 180 Grad). Also ist w_j seine eigene *inverse Funktion*: $w_j^{-1} = w_j$

Brainball umdrehen:

Die Wirkung des Umdrehens des gesamten Brainballs (180 Grad) definieren wir entsprechend als $w_D : \mathbb{B} \rightarrow \mathbb{B}, w_D(z) := (\pi', \gamma')$ mit

$$\begin{aligned} \pi'(i) &:= \begin{cases} \pi(i) & \text{für } i = 1 \\ \pi(15 - i) & \text{sonst} \end{cases} \\ \gamma'(\pi(i)) &:= \neg\gamma(\pi(i)) \end{aligned}$$

Auch hier gilt $z = (w_D(z))^2$. So ist auch w_D seine eigene *inverse Funktion*: $w_D^{-1} = w_D$

Wenn man versucht, den Brainball in eine bestimmte Form zu bringen (zum Beispiel alle weißen Steine auf eine Seite, siehe die Logik-Aufgaben weiter hinten), so zeigt sich schnell, dass es sinnvoll ist, über die einzelnen durchgeführten Schritte Buch zu führen. Dazu können wir direkt die gerade eingeführten mathematischen Formalismen nutzen. Zur Abkürzung schreiben wir Zugfolgen im Folgenden als Wörter über dem Alphabet $\mathbf{R} := \{/, D, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6\}$. Die Zugfolgen, beziehungsweise Wörter nennen wir *Rochaden*. Die Wirkung $w(r) : \mathbb{B} \rightarrow \mathbb{B}$ eines Zeichens aus \mathbf{R} , also eines einstelligen Wortes $r \in \mathbf{R}$ sei

nun $w(r) := w_r$ (Wir erweitern nun auch den Begriff der inversen Wirkungsfunktionen auf den des inversen einstelligen Wortes). Die Wirkung mehrstelliger Wörter sei nun rekursiv durch Konkatenation definiert:











$$w(sr) := w_r(w(s)) \quad \text{für alle } r \in \mathbf{R}, s \in \mathbf{R}^+$$

Mit $w(\lambda) := id_{\mathbb{B}}$ ist die Wirkung nun auf ganz \mathbf{R}^* definiert. Entsprechend ist zu einer gegebenen Rochade r die *inverse Rochade* dadurch definiert, dass man die Rochade r in umgekehrter Reihenfolge notiert und dabei jedes Zeichen durch seinen inversen Gegenpart ersetzt. Es gilt dann $w(r^{-1}r) = id_{\mathbb{B}}$.

2.2 Rochaden

Mit Hilfe der Rochaden ist es uns nun möglich, längere Zugfolgen eindeutig zu notieren. Dabei haben wir die Wirkung und die Notation der Rochaden gerade so gewählt, dass sie dem in der Brainball-Anleitung umgangssprachlich eingeführten Begriff der Rochade entsprechen. Dort findet man auch die Unterscheidung in *Farbrochaden* und *Zahlenrochaden*. Demnach zeichnen sich Farbrochaden dadurch aus, dass ihre Wirkung nur die Farben von Steinen verändert, nicht aber ihre Anordnung. Alle anderen Rochaden werden in der Spielanleitung als Zahlenrochaden bezeichnet. Abweichend davon wollen wir hier nur diejenigen Rochaden als Zahlenrochaden bezeichnen, deren Wirkung die Farben der Steine nicht beeinflusst. Den Rest nennen wir *gemischte Rochaden*. Ferner bezeichnen wir eine Rochade genau dann als *in Standardform*, wenn maximal ein Umdrehzeichen „D“ und zwar genau als letztes Zeichen in der Rochade vorkommt und sich ansonsten immer ein Kippzeichen „/“ und ein Schiebezeichen abwechseln.

Es folgen ein paar der in der Spielanleitung und auf der Brainball-Homepage angegebenen Rochaden mit ihren Namen und ihrer (graphisch dargestellten) Wirkung auf einen Brainball mit einfarbigem Zustand:

Venus-Rochade (Farbe)	Saturn-Rochade (Farbe)	Odysseus-Rochade (gemischt)	Kleopatra-Rochade (gemischt)	Stellinger-Rochade (Zahlen)
				
/-2/2/-2 /2/-2/2	/-2/-2/-5/-5 /3/6/-5/5D	/-1/-1/-5/-1 /-4/-6/6/	/-1/1/4/ 1/-1/-4	/1/-6/ 6/6/6/
				

3 Die Aufgabenstellungen

Der Hersteller des Brainballs hat sich verschiedene Spielvarianten ausgedacht, um dem Brainball-Spieler die Möglichkeit zu geben, den Schwierigkeitsgrad – je nach eigener Erfahrung – Stufe um Stufe steigern zu können.

3.1 Logikspiel

Das Logikspiel selbst ist in 12 Schwierigkeitsstufen unterteilt. Die jeweiligen Ziele sind:

Stufe 1: Bringen Sie alle weissen Spielsteine auf eine Seite.

Stufe 2: Bringen Sie alle dunklen Spielsteine auf eine Seite, so dass auf dieser Seite die Zahlen 1 - 2 im Uhrzeigersinn aufsteigend aufeinanderfolgen.

...

Stufe 11: Bringen Sie alle weissen Spielsteine auf eine Seite, so dass auf dieser Seite die Zahlen 1 - 11 im Uhrzeigersinn aufsteigend aufeinanderfolgen.

Stufe 12: Bringen Sie alle dunklen Spielsteine auf eine Seite, so dass auf dieser Seite alle 13 Zahlen im Uhrzeigersinn aufsteigend sortiert sind.

Das Logikspiel ist gut als Einstiegsaufgabe geeignet. Anfangs hat man sehr große Schwierigkeiten, überhaupt Stufe 1 zu erreichen. Mit der Zeit entdeckt man spezielle Rochaden, mit denen man bestimmte Teilaufgaben, wie zum Beispiel das Sortieren von Zahlen, lösen kann. Mit den richtigen Rochaden kann man schliesslich relativ leicht alle 12 Stufen meistern.

Wie man dabei vorgehen kann, wird im nächsten Kapitel erläutert, wenn es darum geht, zu beweisen, dass alle denkbaren Zustände von einem beliebigen Startzustand aus erreichbar sind.

3.2 Strategiespiel

Das Ziel des Strategiespiels ist, durch geschickte Kombination von bestimmten Rochaden den Brainball von einem vorgegebenen Ausgangszustand in einen ebenso vorgegebenen Endzustand zu überführen. Beispielsweise könnte die Aufgabenstellung lauten:

Überführen Sie den Brainball vom Zustand $(\bar{5}, \bar{6}, 7, 8, 9, 10, 11, 12, 13, 1, 2, 3, 4)$ in den Zustand $(\bar{5}, \bar{6}, 7, 8, 11, \bar{9}, \bar{13}, 10, \bar{12}, \bar{1}, 2, \bar{3}, \bar{4})$. Benutzen Sie dabei 2 mal die Kleopatra-Rochade und 1 mal die Venus-Rochade. Eine weitere Verwendung des grundlegenden Spielzuges „Kippen“ ist nicht erlaubt.

Eine Strategiespiel-Aufgabe ist gut geeignet, um die effektive und vorausplanende Verwendung von Rochaden zu üben. Wir werden diese Aufgabe an dieser Stelle jedoch nicht eingehender betrachten, da sie aus Sicht der Informatik keine Herausforderung darstellt. Schliesslich muss man nur eine sehr geringe Zahl an Möglichkeiten überprüfen (Beim obigen Beispiel $3 \cdot 13^3 = 6591$), welches durch eine informatische Simulation innerhalb eines Bruchteils einer Sekunde gelöst werden kann. Interessant wird eine solche Aufgabe erst, wenn die Zahl der zu kombinierenden Rochaden deutlich grösser ist. Dann wäre eine Simulation durch simples Ausprobieren aller Lösungen zeitlich nicht mehr vertretbar und es stellt sich unter anderem die Frage, ob man – abhängig von der Art der erlaubten Rochaden – intelligentere Simulationsalgorithmen entwickeln kann, die zielgerichtet arbeiten. In dieser Arbeit wollen wir auf diese Frage allerdings nicht eingehen.

3.3 Wettkampfspiel

Wie bei dem Strategiespiel geht es bei dem Wettkampfspiel darum, von einem vorgegebenen Ausgangszustand in einen vorgegebenen Zielzustand zu gelangen. Diesmal gilt es im Spiel gegen die Zeit oder gegen andere Mitspieler die Zielstellung so schnell wie möglich zu erreichen. Dabei ist den Spielern vollkommen freigestellt, welche Spielzüge sie verwenden.

Diese Spielvariante ist besonders für Spieler geeignet, die eine hinreichende Menge von Rochaden beherrschen. Wir wollen diese Aufgabe hier nicht weiter behandeln, da es sich um eine reine Wettkampfaufgabe zwischen realen Spielern handelt.

3.4 Große Strategie-Aufgabe

Bringen Sie den Brainball mit möglichst wenigen „Kipp“-Zügen von dem Zustand $(\bar{5}, \bar{6}, 7, 8, 9, 10, 11, 12, 13, 1, 2, 3, 4)$ in den Zustand $(9, 11, 13, \bar{2}, \bar{4}, \bar{6}, \bar{8}, \bar{10}, \bar{12}, 1, 3, 5, 7)$.

Definiert man die Länge l einer Rochade r als Anzahl des Auftretens des Spielzuges „/“ in der Rochade, also $l(r) := \#\{i \mid r[i] = /\}$, dann lautet die obige Aufgabe, zu einer gegebenen Wirkung eine Rochade minimaler Länge zu

ermitteln. Damit handelt es sich um ein Optimierungsproblem, welches wir in einem späteren Kapitel noch sehr ausführlich behandeln werden. Wir wollen das Ergebnis der hier vorgegebenen Problem Instanz bereits vorweg nehmen: Es gibt genau 14 verschiedene Rochaden der Länge 13, die die gewünschte Wirkung haben. Außerdem existiert keine kürzere Rochade mit dieser Lösung. Die optimalen Rochaden sind:

-6/-1/2/-2/-1/-1/2/4/-5/-4/4/-6/-5/2D

-6/-1/2/-2/-1/-1/5/-4/-2/-4/-6/6/5/2D

-6/4/2/-1/-3/-4/4/2/-5/-2/-1/-4/1/1D

-6/4/2/2/3/-1/4/2/-5/-2/-1/-4/1/1D

2/-4/-2/-2/-3/1/-4/-2/5/2/1/4/-1/1D

2/-4/-2/1/3/4/-4/-2/5/2/1/4/-1/1D

2/1/-2/2/1/1/-2/-4/5/4/-4/6/5/D

2/1/-2/2/1/1/-5/4/2/4/6/-6/-5/D

3/-3/-5/-1/5/1/2/-5/-4/-2/1/4/4/2D

3/-3/2/1/-5/-1/-2/5/4/2/-1/-4/-4/D

4/1/4/2/-1/-2/-1/5/2/6/-1/5/-5/1D

5/-1/-4/-2/1/2/1/-5/-2/-6/1/-5/5/1D

6/3/-2/-1/5/1/2/-5/-4/-2/1/4/4/2D

6/3/5/1/-5/-1/-2/5/4/2/-1/-4/-4/D

4 Kombinatorik

4.1 Wie viele Zustände gibt es?

Es gibt $13! = 6.227.020.800$ unterscheidbare Permutationen über \mathbb{N}_{13} . Ausserdem gibt es $2^{13} = 8.192$ unterscheidbare Abbildungen γ von \mathbb{N}_{13} in Γ . Also gibt es insgesamt $13! \cdot 2^{13} = 51.011.754.393.600$ unterscheidbare Zustände.

Um nicht mit dieser sehr großen Zustandsmenge arbeiten zu müssen, nennen wir zwei Zustände *äquivalent*, wenn sie ohne „Kipp“-Züge ineinander überführbar sind, also wenn sich zwei Brainbälle, die sich in diesen Zuständen befinden, höchstens in der Drehposition der Brainballkugeln und in der betrachteten Zahlenkranzseite unterscheiden (Weiterhin nennen wir zwei Äquivalenzklassen B_1, B_2 genau dann *benachbart*, wenn $(\exists b_1 \in B_1 : w_\gamma(b_1) \in B_2) \wedge (\exists b_2 \in B_2 : w_\gamma(b_2) \in B_1)$ gilt, oder – was in *diesem* Fall dasselbe bedeutet – wenn zwei Zustände $b_1 \in B_1, b_2 \in B_2$ existieren mit $w_\gamma(b_1) = b_2$). Hat man nun eine Rochade gefunden, die zum Beispiel einen zum Startzustand der grossen Strategieaufgabe äquivalenten Zustand in einen zum Endzustand äquivalenten Zustand überführt, so ist tatsächliche Lösung trivial ermittelbar. Betrachtet man nur bezüglich dieser Äquivalenz unterscheidbare Zustände, so bleiben immernoch $12! \cdot 2^{12} = 1.961.990.553.600$ Möglichkeiten. Insbesondere beträgt die Wahrscheinlichkeit $\frac{1}{1.961.990.553.600}$, dass ein zufällig gewählter Zustand der Lösung der 12. Logikaufgabe entspricht.

Entsprechend kann man zum Beispiel fragen, wie viele bezüglich der 1. Logikstufe unterscheidbare Zustandsklassen es gibt, also, wie viele Situationen man unterscheiden muß, wenn es darum geht, alle gleichfarbigen Spielsteine auf eine Brainballseite zu bringen. Wieder beschränken wir uns darauf, nur eine Brainballseite zu betrachten. Es bietet sich diejenige an, auf der mehr weiße Spielsteine sind: Es gibt nur eine Zustandsklasse mit 13 weißen Steinen. Die Anzahl der Zustandsklassen mit $n = 12, 11, \dots, 7$ weißen Steinen (6, ..., 1 müssen nicht beachtet werden, da diese die Rückseiten der anderen sind) ergibt sich aus der Formel $\frac{\binom{13}{n}}{13}$. Das ergibt folgende Tabelle:

Anzahl an weißen Steinen	13	12	11	10	9	8	7	\sum
Anzahl an Zustandsklassen	1	1	6	22	55	99	132	316

Es gibt also insgesamt 316 zu unterscheidene Situationen bei der Lösung der 1. Logikstufe. Dabei wurde noch nicht beachtet, dass viele Zustandsklassen Spiegelsymmetrien untereinander aufweisen, so dass ein geübter Spieler sich noch weniger merken muss (nämlich 190 Situationen), um aus jedem Startzustand in

einen Zielzustand zu gelangen, bei dem jede Brainballseite nur eine Farbe aufweist.

Für die 2. Logikstufe gilt:

weiße Steine	13	12	11	10	9	8	7	Σ
Zustandsklassen	12	156	936	3.432	8.580	15.444	20.592	49.152

Entsprechend kann man die Zahl der unterscheidbaren Situationen für jede Logikstufe berechnen und kommt zu folgendem Ergebnis:

Logikstufe	Anzahl an Zustandsklassen
1	316
2	49.152
3	540.672
4	5.406.720
5	48.660.480
6	389.283.840
7	2.724.986.880
8	16.349.921.280
9	81.749.606.400
10	326.998.425.600
11	980.995.276.800
12	1.961.990.553.600

4.2 Erreichbarkeit

Bei den Berechnungen im letzten Abschnitt wurde stillschweigend vorausgesetzt, dass alle theoretisch denkbaren Zustände von jedem beliebigen Startzustand aus erreichbar sind. Den Beweis hierfür holen wir nun nach. Dazu zeigen wir, wie man einen beliebigen Zustand in den Zustand $b_0 = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13)$ überführen kann. Da jede Rochade invertierbar ist, können wir so jeden Zustand über b_0 in jeden anderen Zustand überführen.

Um zu zeigen, wie wir von jedem Zustand aus zu b_0 gelangen, gehen wir in zwei Schritten vor:

1. Farben sortieren

Dies entspricht der Aufgabenstellung der 1. Logikstufe. Zuerst überprüfen wir, ob auf der betrachteten Brainballseite eine gerade Zahl an gelben Steinen ist. Wenn nicht, dann drehen wir ihn mittels „D“ um. Nun gehen wir folgendermaßen vor: Solange noch gelbe Steine auf der betrachteten Seite vorkommen, schieben wir den Zahlenkranz im Uhrzeigersinn soweit, dass der nächsten gelbe Stein in der Mitte des kleinen Kippsegments liegt und führen die Venus-Rochade durch. Wenn man den Zahlenkranz auf diese Weise um insgesamt 11 Schritte verschoben hat, sind nur noch zwei Fälle möglich: Entweder es sind keine gelben Steine mehr auf der betrachteten Seite, oder es sind genau zwei direkt nebeneinander. Bei dem letzteren Fall zeigt sich nun, dass das obige Verfahren – wenn man es an der Position des zweiten gelben Steines wieder ansetzt – nach 7 Schritten zum Ziel führt. Insgesamt terminiert der Algorithmus also nach maximal 18 Schritten. Da die Venus-Rochade 6 „Kipp“-Züge benötigt, brauchen wir also eine Rochade mit einer maximalen Länge von 108 mal Kippen, um die Farben zu sortieren.

2. Zahlen sortieren

Um die Zahlen zu sortieren, ohne die gerade hergestellte Ordnung der Farben zu zerstören, verwenden wir ausschliesslich Zahlenrochaden. Damit ist bei einem Zustand (π, γ) nur die Zahlenpermutation π von Interesse. Wir werden gleich einen Algorithmus konstruieren, der einen solchen Zustand mit gerader Permutation in den Zielzustand b_0 überführt. Wenn wir also einen Zustand mit ungerader Permutation haben, so müssen wir auf ihn zuerst eine Zahlenrochade, deren Wirkung selbst eine ungerade Permutation beschreibt, anwenden (Zur Erinnerung: Wendet man zwei Rochaden hintereinander an, so ergibt sich das Vorzeichen der dadurch entstehenden Rochade als Produkt der beiden Vorzeichen). Eine Rochade mit ungerader Permutation ist zum Beispiel „/5/5/D“, wie sich leicht überprüfen lässt. Um einen Zustand von den nun garantierbaren Eigenschaften (alle weißen

Steine auf einer Seite in gerader Permutation) zu b_0 zu gelangen, benötigen wir nur noch die Steldinger-Rochade (Diese ist eine Zahlenrochade mit gerader Permutationswirkung). Dabei gehen wir wie folgt vor: Solange es noch drei benachbarte Steine gibt, die nicht bezüglich ihrer Grösse geordnet sind und von denen kein Stein die Zahl 13 trägt, vertausche ihre Reihenfolge durch

- Verschieben des Zahlenkranzes, so dass der Stein rechts neben dieser Dreiergruppe am mittleren Feld des kleinen Kippsegments liegt, und
- Ausführen der Steldinger-Rochade.

Wenn der Algorithmus terminiert, dann im gewünschten Zustand b_0 . Dass der Algorithmus bei einem Eingabezustand mit gerader Permutation terminiert lässt sich leicht zeigen. Allerdings könnte man noch untersuchen, nach welchem Verfahren eine von evtl. mehreren vorhandenen unsortierten Dreiergruppen ausgewählt werden sollte, damit der Algorithmus eine kleinstmögliche Worst-Case-Schrittzahl benötigt. Da es sich um ein spezielles Sortierproblem handelt, stellt sich die Frage, ob die untere Schranke $O(n \cdot \log n)$ (bei n zu sortierenden Steinen) erreichbar ist.

Es lassen sich auch mit Hilfe von anderen Rochaden Beweise aufstellen, dass alle Zustände erreichbar sind – insbesondere lassen sich so Algorithmen aufstellen, die dies in weniger Schritten schaffen. Hier wurde mit Absicht eine Variante gewählt, die durch die ausschliessliche Verwendung von nur einer Zahlenrochade und einer Farbrochade – also ohne die Verwendung von gemischten Rochaden – eine gewisse Eleganz aufweist.

5 Berechnen von Rochaden

Wir haben schon verschiedene Rochaden kennengelernt. Doch tatsächlich gibt es unendlich viele. Allerdings kann man bezüglich ihrer Wirkung nur so viele Rochaden unterscheiden, wie es unterscheidbare Zustände gibt. Besonders interessant, weil nützlich, sind Farb- und Zahlenrochaden. Für viele Aufgaben ist es von Vorteil, viele solche Rochaden zu kennen. Um solche Rochaden zu suchen bietet sich die Zuhilfenahme eines Computers an. Auf welche Weise eine solche Suche abläuft, kann man dem folgenden Kapitel entnehmen. Hier wollen wir Ergebnisse der Suche nach interessanten Rochaden vorstellen.

5.1 Farbrochaden

Bezüglich ihrer Wirkung lassen sich nur so viele Farbrochaden unterscheiden, wie es Farbbelegungen eines Brainballs mit gleichbleibender Zahlenanordnung gibt. Also gibt es nur $2^{13} = 8192$ in Wirkung unterscheidbare Farbrochadenklassen. Ein Brainballspieler braucht aber nur $2 \cdot 316$ (wegen der zwei Seiten; siehe Kombinatorik-Kapitel) Farbrochaden zu unterscheiden, um jede Farbpermutation durch Anwenden von nur einer Rochade (und davor, bzw. danach evtl. noch grundlegenden Zügen der Art „Zahlenkranz verschieben“) erreichen zu können.

Wenn man Spiegelsymmetrien ausnutzt (Der Spieler müsste einfach die entsprechende Rochade in umgekehrter Reihenfolge abarbeiten), so bleiben noch $2 \cdot 190 = 380$ Farbrochadenklassen. Sucht man entsprechende Rochaden systematisch, so ergibt sich, dass es keine Farbrochaden r mit einer Länge $l(r) < 6$ gibt und nur 14 dieser Farbrochadenklassen durch Rochaden der Länge $l(r) \leq 8$ repräsentiert werden können. Zu Vieren sind Repräsentanten in der Brainball-Spielanleitung angegeben. Zu den restlichen 10 stehen hier die kürzesten. Die Wirkung bezieht sich immer auf den Zustand $b_0 = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13)$:

Venus: $/-2/2/-2/2/-2/2 :$ $(\bar{1}, 2, \bar{3}, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13)^*$
Jupiter: $/-5/5/2/5/-5/-2 :$ $(1, 2, 3, \bar{4}, 5, 6, 7, 8, \bar{9}, 10, 11, 12, 13)^*$
Mars: $/2/6/5/2/5/6/2/-2 :$ $(\bar{1}, 2, 3, 4, \bar{5}, 6, 7, \bar{8}, 9, 10, 11, \bar{12}, 13)^*$
Saturn: $/-2/-2/-5/-5/3/6/-5/5D :$ $(1, \bar{2}, 3, 4, 5, 6, 7, 8, \bar{9}, 10, \bar{11}, 12, 13)$
 $/5/-3/5/5/-3/5D :$ $(\bar{1}, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13)$
 $/4/2/-2/2/-2/2/-6/ :$ $(1, 2, 3, 4, 5, 6, \bar{7}, 8, 9, \bar{10}, 11, 12, 13)$
 $/1/2/-2/2/-2/2/-3/ :$ $(1, \bar{2}, 3, 4, 5, 6, 7, 8, 9, 10, \bar{11}, 12, 13)$
 $/5/2/2/5/-5/3/-3/5D :$ $(\bar{1}, 2, 3, 4, 5, 6, \bar{7}, 8, 9, 10, 11, \bar{12}, 13)$
 $/5/-5/-2/-2/-5/5/-3/-3 :$ $(1, 2, \bar{3}, 4, 5, \bar{6}, 7, 8, 9, 10, \bar{11}, 12, 13)$
 $/5/6/2/-2/2/6/5/2 :$ $(1, 2, 3, \bar{4}, 5, \bar{6}, 7, 8, 9, 10, \bar{11}, 12, \bar{13})^*$

$$\begin{aligned}
/5/2/5/6/2/-2/2/6 &: (\bar{1}, 2, 3, 4, 5, 6, \bar{7}, 8, \bar{9}, 10, 11, \bar{12}, 13) \\
/5/-5/6/-2/-5/5/-6/2 &: (1, 2, 3, 4, \bar{5}, \bar{6}, 7, 8, 9, 10, \bar{11}, 12, 13) \\
/5/-5/-2/6/-5/5/2/-6 &: (1, 2, \bar{3}, 4, \bar{5}, \bar{6}, 7, 8, 9, 10, \bar{11}, 12, 13) \\
/6/-6/6/-6/6/-6/6/-6 &: (\bar{1}, \bar{2}, \bar{3}, 4, 5, \bar{6}, \bar{7}, \bar{8}, \bar{9}, 10, 11, 12, \bar{13})^*
\end{aligned}$$

Auffallend ist, dass es keine Farbrochaden ungerader Länge zu geben scheint. Dass und warum dies so ist, werden wir später begründen. Ebenso werden wir sehen, warum genau diejenigen Rochaden, bei denen sich die Farbe einer ungeraden Zahl an Steinen ändert, den Spielzug „D“ benötigen. Auch auffallend ist, dass es zu einer Farbrochadenklasse mit symmetrischer Wirkung oft repräsentierende Rochaden zu geben scheint, die Palindrome darstellen. Dies ist zumindest bei den mit einem * markierten Rochaden der Fall. Ob dies immer gilt, beziehungsweise wann dies gilt, wird allerdings in dieser Arbeit nicht untersucht.

5.2 Zahlenrochaden

Es gibt weitaus mehr Zahlenrochaden als Farbrochaden. Wenn man anfängt, sich Zahlenrochaden berechnen zu lassen, so stellt man fest, dass es allein über 63.000 in ihrer Wirkung unterscheidbare Zahlenrochaden r der Länge $l(r) \leq 8$ gibt. Eine davon ist die bereits vorgestellte Steldinger-Rochade. In der Brainball-Spielanleitung steht nicht eine Zahlenrochade – dies ist um so verwunderlicher, wenn man bedenkt, wie viele es gibt. Hier wollen wir nur die kürzesten nennen (Wieder mit exemplarischer Wirkung auf b_0):

$$\begin{aligned}
/5/5/D &: (1, 5, 3, 4, 2, 6, 7, 11, 12, 10, 8, 9, 13) \\
/5/-3/5/ &: (1, 5, 6, 7, 2, 3, 4, 11, 12, 13, 8, 9, 10) \\
/4/-3/4/-3 &: (1, 2, 3, 4, 5, 12, 7, 8, 9, 10, 11, 13, 6) \\
/1/3/1/-3 &: (1, 2, 3, 4, 7, 12, 13, 8, 9, 10, 11, 5, 6)
\end{aligned}$$

5.3 Kürzungsregeln

Ein interessanter Ansatz, um möglichst kurze Rochaden einer gegebenen Wirkung zu ermitteln ist, zu versuchen, eine gegebene Rochade dieser Wirkung zu einer kürzeren umzuformen. Dabei geht man so vor, dass man bei der Rochade Teilrochaden (Teilworte) sucht, zu denen man bezüglich der Wirkung äquivalente, aber kürzere Rochaden kennt. Ersetzt man damit den entsprechenden Teil der Ausgangsrochade, so ergibt sich eine kürzere mit derselben Wirkung.

Da die betrachtete Teilrochade auch die ganze Rochade sein kann, führt das Verfahren prinzipiell unmöglich in eine Sackgasse, es führt also stets zu einer Rochade optimaler Länge. Allerdings ist der Aufwand dafür im Worst-Case

deutlich größer als bei einer direkten Suche der optimalen Lösung (wie sie im nächsten Kapitel beschrieben wird), da dies hier als Teilaufgabe (die Suche einer zur größten Teilrochade äquivalenten Rochade) auftritt. Außerdem kann ein Ende des Algorithmusses nicht erkannt werden, ohne alle Rochaden auf Äquivalenz zu überprüfen, die maximal die gleiche Länge besitzen. Beschränkt man sich darauf, nur Teilrochaden bis zu einer bestimmten Länge k zu untersuchen, so ist der Algorithmus zwar bedeutend schneller, aber eine gefundene Lösung nicht unbedingt optimal, da es diesmal nicht ausgeschlossen ist, in „Sackgassen“ zu landen. Solche *k-optimalen Lösungen* sind es vielleicht wert, genauer untersucht zu werden.

Überraschenderweise wird hier eine Klasse von Rochaden besonders interessant, die sonst eher sehr unbedeutend wirkt: Die Rochaden mit der Identität als Wirkung, also diejenigen Rochaden, die einen Zustand wieder in sich selbst überführen (im Folgenden als *identische Rochaden* bezeichnet), haben nämlich die Eigenschaft, dass man aus ihnen (durch Zerlegung in zwei Teile) alle Paare von äquivalenten Rochaden ermitteln kann, denn es gilt:

$$w(u) = w(v^{-1}) \Leftrightarrow w(uv) = id_{\mathbb{B}} \Leftrightarrow w(v) = w(u^{-1})$$

Zwar bringt dies keinen wirklichen Vorteil bei der Berechnung von kürzeren Rochaden, da identische Rochaden genau so aufwendig zu ermitteln sind, aber mit der Kenntnis von zwei äquivalenten Rochaden kennt man nun gleich eine ganze Menge von äquivalenten Rochadenpaaren, da sich die daraus gewonnene identische Rochade auf viele verschiedene Weisen in zwei Teilrochaden u und v aufteilen lässt.

Im Übrigen ist es nicht nötig, dass man eine Teilrochade durch eine kürzere ersetzt; auch gleich lange Rochaden können sinnvoll sein, wie das folgende Beispiel darlegt. Eine sehr kurze identische Rochade ist $/3/3/3/3$. Daraus folgen unter anderem die Äquivalenzen $3/3/3 \simeq /-3/$ und $-3/-3/-3 \simeq /3/$. Man beachte, dass diese beiden Paare von Rochaden alle von derselben Länge drei sind. Will man nun zum Beispiel die Rochade $/3/3/6/3/$ kürzen, so kann man zum Beispiel das erste Auftreten von $/3/$ durch $-3/-3/-3$ ersetzen. Das Ergebnis ist $-3/-3/-33/6/3/$. Nun kann man die Folge -33 weglassen, da die beiden Zeichen invers zueinander sind. Ebenso kann man aus der übrig bleibenden Rochade $-3/-3//6/3/$ die aufeinander folgenden Kippzeichen entfernen. Man erhält die Rochade $-3/-36/3/$. Jetzt kann man nach Definition der Definition der Schiebzeichen $-3\ 6$ durch 3 ersetzen, was $-3/3/3/$ ergibt. Entsprechend fortgesetzt ergibt sich schliesslich die Rochade $-6/-3$, welche um 4 Kippzeichen kürzer ist, als die Ausgangsrochade.

Damit haben wir eine sehr oft verwendbare Teilrochaden-Ersetzungsregel entdeckt, die in vielen Fällen auch zu kürzeren Rochaden führt. Man kann sie (für Rochaden in Standardform) folgendermaßen formulieren:

Wenn in einer Rochade irgendwo 3 (oder -3) steht, kann man diese durch -3 (bzw. 3) ersetzen, solange man von beiden „Nachbarzahlen“ 3 abzieht (bzw. addiert). Trat eine 3 (bzw. -3) als Nachbar auf, so heben sich zwei Kippungen auf. Die entstehende Rochade besitzt dieselbe Wirkung wie die Ausgangsrochade.

Beispiel:

$$\begin{aligned} 1/3/-3/-3/4 &\simeq -2/-3/-6/-3/4 \simeq -2/-3/-3/3/-6 \\ &\simeq -2/-3/-6/-3/-3 \simeq 1/3/-3/-3/-3 \simeq 1/6/-6 \end{aligned}$$

Diese einzelne Regel zeigt eindrucksvoll, wie mächtig ein Algorithmus zum Kürzen von Rochaden sein kann, selbst wenn sich dieser auf das Finden von k -optimalen Lösungen mit einem relativ kleinen $k \in \mathbb{N}$ (zum Beispiel $k = 6$) beschränkt. Dies ist vor allem deshalb bereits sehr effektiv, weil die Wahrscheinlichkeit, sehr kurze Kürzungsregeln (wie die oben genannte) anwenden zu können, ungleich größer ist, als die für längere Regeln.

6 Die große Strategie-Aufgabe

Die Aufgabenstellung der großen Strategieraufgabe wurde bereits genannt. In diesem Kapitel wollen wir verschiedene Verfahren zur Suche einer optimalen Lösung diskutieren. Dabei betrachten wir hier mehrere Varianten der Suche eines kürzesten Pfades zwischen zwei Ecken in einem ungerichteten, ungewichteten Graphen. Der zu durchsuchende Graph G enthält die Äquivalenzklassen der Zustandsmenge als Ecken, und die durch die „benachbart“-Relation definierten Kanten: $G = (V, E)$ mit $V = \{B \mid \exists b \in \mathbb{B} : B = [b]_{\sim}\}$ und $E = \{(B_1, B_2) \mid B_1 \text{ ist benachbart zu } B_2\}$.

Der Graph hat 1.961.990.553.600 Ecken des Grades 13. Die Größe des Graphen macht eine besonders sorgfältige Wahl des Suchverfahrens nötig, da die Unterschiede in den Laufzeiten und im benötigten Speicherplatz zum Teil gravierend sind. Alle hier diskutierten Suchverfahren arbeiten nach dem selben Prinzip:

Die Menge der Ecken wird aufgeteilt in bereits untersuchte, bereits entdeckte (aber noch nicht untersuchte) und noch nicht entdeckte Ecken. Während der Suche wird ständig eine bereits entdeckte Ecke ausgewählt und ihre Nachbarecken ermittelt (die Ecke wurde somit untersucht). Diese werden dann als entdeckt klassifiziert. Ist eine dieser Ecken eine Zielecke, so stoppt der Algorithmus mit der Ausgabe des Lösungspfades (dieser muß ständig mitgespeichert werden). Die folgenden Varianten unterscheiden sich unter anderem in der Definition der Starteckenmenge (die Ecken, die am Anfang als entdeckt klassifiziert werden) und der Zieleckenmenge, sowie in der Wahl des Verfahrens zur Auswahl einer entdeckten Ecke.

Wenn im Folgenden Laufzeiten von Programmen genannt werden, so beziehen sie sich auf die Ausführung von in C++ geschriebenen und mit „Borland C++ 5.0 for Windows“ kompilierten Code auf einem Intel Pentium II mit einer Taktfrequenz von 400 MHz mit 192 MByte PC100-RAM Arbeitsspeicher.

6.1 Breitensuche / Dijkstras Algorithmus

Die Breitensuche läuft in Schritten ab: In einem Schritt werden alle bisher entdeckten (aber noch nicht untersuchten) Ecken untersucht, bevor (im nächsten Schritt) die nun neu entdeckten Ecken untersucht werden (Da wir es mit einem ungewichteten Graphen zu tun haben, entspricht dies genau dem „äquidistante Welle“-Suchalgorithmus von Dijkstra). Dabei enthalten Startecken- und Zieleckenmenge jeweils nur ein Element (die beiden Ecken, zwischen denen wir einen möglichst kurzen Pfad suchen). Breitensuche hat den großen Vorteil, dass ei-

ne gefundene Lösung immer optimal ist und der Algorithmus garantiert nach endlicher Zeit terminiert (schließlich ist jede der endlich vielen Ecken mit endlich vielen Schritten erreichbar). Allerdings braucht dieser Algorithmus sehr viel Speicherplatz (exponentiell von der Schritttiefe abhängig), da die gesamte Menge der am Anfang eines jeweiligen Schrittes erreichten, noch nicht untersuchten Ecken gespeichert werden muss. Man beachte, dass zu jeder zu speichernden Ecke auch der Pfad (eine Rochade), durch den diese Ecke erreicht wurde, mitgespeichert werden muß. Die folgende Tabelle zeigt, wie viele Ecken in jedem der ersten 7 Schritte bereits entdeckt oder schon untersucht wurden, beziehungsweise wieviele gerade neu hinzukommen sind:

Rochadenlänge i	entdeckt/untersucht Def.: $f(i)$	neu entdeckt $f(i) - f(i - 1)$	Wachstumsfaktor $f(i)/f(i - 1)$
0	1		
1	14	13	14,000
2	157	143	11,214
3	1.704	1.547	10,854
4	18.344	16.640	10,765
5	195.131	176.787	10,637
6	2.045.511	1.850.380	10,483
⋮	⋮	⋮	⋮
n	1.961.990.553.600	> 0	> 1
$n+1$	1.961.990.553.600	0	1

Dafür wurde für jede Schritttiefe n eine Liste aller mit bis zu n Schritten erreichbaren Ecken in einer Datei gespeichert und in dieser anschliessend alle Mehrfachnennungen gelöscht (dazu wurde die Datei sortiert). Allein bei dem letzten berechneten Schritt (Länge 6) war die unsortierte Datei 37MB groß. Die Datei zu sortieren und die Anzahl der unterschiedlichen Einträge zu ermitteln brauchte über 17 Minuten. Hätte man noch eine weitere Zeile ausrechnen wollen, so wäre die dafür nötige Datei fast 0,5GB groß und man bräuchte über 4 Stunden (bei einem Sortieralgorithmus mit Komplexität $O(n \log n)$). Noch ein weiterer Schritt würde zu über 6GB und über 2 Tagen Rechenzeit führen, usw. . Ab einem gewissen n hat man schliesslich alle Ecken erreicht und es kommen in den weiteren Schritten keine neuen hinzu. Diese Grenze (von der wir noch nicht wissen, wo genau sie liegt) ist insofern von Bedeutung, dass sie die maximal nötige Suchtiefe für jeden Suchalgorithmus angibt.

6.2 Tiefensuche

Bei der Tiefensuche wird zu einer zu untersuchenden Ecke nur ein Nachfolger ermittelt und sofort dieser nach demselben Verfahren rekursiv untersucht. Erst,

wenn bei diesem sämtliche Nachfolger untersucht worden sind, wird der nächste Nachfolger der ursprünglichen Ecke betrachtet. Das hat den Vorteil, dass man mit sehr wenig Speicher auskommt (linear abhängig von der Schritttiefe). Allerdings ist dieser Algorithmus nicht ohne weiteres auf unseren Suchgraphen anwendbar, da er leicht in einen Zyklus gelangt und somit unendlich lange läuft. Selbst wenn man Zyklen vermeidet, so eignet sich die Tiefensuche kaum für unsere Zwecke, da eine gefundene Lösung in der Regel nicht optimal ist. Es kann sogar sein, dass der Pfad der ersten gefundenen Lösung nahezu alle 1.961.990.553.600 Ecken durchläuft.

6.3 Iterativ begrenzte Tiefensuche (Iterative Deepening)

Wir können das Prinzip der Tiefensuche erheblich verbessern, indem wir eine maximale Schritttiefe einführen. So kommt der Algorithmus nach endlicher Zeit zu einer Lösung, wenn es eine Lösung mit dieser maximalen Schritttiefe gibt. Wir wissen bereits, dass diese Grenze so wählen können, dass immer eine Lösung gefunden werden kann – Leider wissen wir dabei nicht, wo genau wir diese Grenze setzen müssen. Ausserdem haben wir so immer noch nicht gewährleistet, dass die gefundene Lösung optimal ist. Dies können wir aber durch das so genannte *Iterative Deepening* erreichen. Wir setzen die maximale Schritttiefe iterativ immer höher und überprüfen jeweils, ob eine Lösung existiert. Sobald wir so eine Lösung finden, ist sie optimal und wir brechen das Verfahren ab. Dabei wird zwar in jedem Schritt unter anderem alles das wieder neu berechnet, was im vorherigen Schritt bereits vorlag, dafür kommt dieser Algorithmus aber mit sehr wenig Speicher aus. Genauere Untersuchungen zeigen, dass der Rechenmehraufwand gegenüber einer Breitensuche sich in einem sehr verträglichen Rahmen hält. Er ist nämlich durch den konstanten Faktor $(\frac{13}{12})^2 = 1,174$ (siehe [6]) gegeben. Also verbindet das Iterative Deepening mit niedrigem Platzaufwand und Optimalitätsgarantie die Vorteile von Tiefen- und Breitensuche. Sowohl bei der Berechnung der letzten Tabelle, als auch bei dem systematischen Berechnen von Rochaden wurde Iterative Deepening verwendet. Allerdings hätte man mit diesem Verfahren die 13 optimalen Lösungen der großen Strategiegabe nicht in vertretbarer Zeit ermitteln können. In Schritttiefe 7 gilt es zum Beispiel $13 \cdot 12^6 = 3.234.816$ Ecken zu berechnen und zu prüfen (dies dauert ca. 33 Sekunden; man beachte, dass hier mehrfach gefundene Ecken auch mehrfach geprüft werden – dadurch entfällt insbesondere der große Aufwand der Zyklenerkennung). Eine Hochrechnung zeigt, dass für Schritttiefe 13 (soviel ist ja nötig, um die große Strategiegabe zu lösen) über 35 Jahre benötigt werden würde.

6.4 A^* -Algorithmus

Bei dem A^* -Algorithmus wird wie bei der Breitensuche eine zu untersuchende Ecke ausgewählt und deren Folgeecken in der Menge der bereits erreichten Ecken eingefügt. Allerdings wird immer genau die Ecke aus dieser Menge als nächstes untersucht, die den kürzesten Weg zum Ziel verspricht. Dies wird mittels einer Heuristik ermittelt, die zu jeder Ecke (durch eine möglichst einfache Rechnung) eine Abschätzung des Abstandes zum Ziel berechnet. Die geschätzte Weglänge über eine Ecke berechnet sich nun durch Addition ihrer (bekannten) Entfernung zur Startecke und ihrer Heuristik. Damit der Algorithmus korrekt funktioniert und eine optimale Lösung findet, darf die Heuristik nicht überschätzen, also muss die berechnete Entfernung zum Ziel niedriger oder genauso lang sein, wie die tatsächliche. Liefert die Heuristik für jede Ecke 0, so verhält sich der Algorithmus genauso, wie eine Breitensuche. Liefert sie immer den tatsächlichen Abstand zum Ziel, so verhält sich der Algorithmus wie ein Greedy-Verfahren und liefert eine optimale Lösung der Länge k nach genau k Schritten. Weiteres wäre ideal, ersteres zeigt aber, dass der A^* -Algorithmus – genauso, wie die Breitensuche – droht, wegen zu hoher Speicherplatzanforderung für unsere Zwecke ungeeignet zu sein.

Es gilt also, eine möglichst gute Heuristik zu ermitteln. Denkbar wäre zum Beispiel die Zahlenverteilung der Zustandsklasse einer Ecke zu vernachlässigen und bezüglich des dann entstehenden Graphen mit den 316 Farbverteilungsabhängigen Ecken die kürzeste Lösung als Heuristik für die kürzeste Lösung im eigentlich betrachteten Suchgraphen zu nehmen. Auf diese Weise kann man verschiedenste durch solche Projektionen entstehende Graphen als Heuristik verwenden. Allerdings erfüllen nur die Graphen mit kleiner Eckenzahl die wichtige Bedingung, dass die Berechnung der Heuristik hinreichend einfach sein soll. In diesem Fall kann man insbesondere den Entfernungsvektor aller Ecken zu der Ecke, die durch Projektion des Zielzustandes entsteht im vornherein berechnen. Doch derartig kleine Projektionsgraphen bringen meist nur eine schlechte Heuristik. Insbesondere ist es dem Autor bisher nicht gelungen, eine effektive Heuristik zu entwickeln, mit der das Speicherplatzproblem, welches bei dem A^* -Algorithmus im Prinzip genauso gegeben ist, wie bei der Breitensuche, ausreichend gelöst wurde. Insbesondere war es damit bisher erst recht nicht möglich, optimale Suchalgorithmen zu entwickeln, die durch Erkennen und Nichtbeachtung von suboptimalen Pfaden den Suchbaum effektiv einschränken.

6.5 Bidirektionale Suche

Wir können jedes der bisher verwendeten Verfahren so verändern, dass es bidirektional arbeitet, also sowohl von der Startecke, also auch von der Zielecke aus jeweils einen Suchbaum expandiert und stoppt, wenn beide Bäume eine gemeinsame Ecke erreichen. Auf diese Weise kann man den Zeitaufwand erheblich reduzieren – allerdings auf Kosten des nötigen Speicherplatzes. Die vom Autor gewählte Implementation verfährt folgendermaßen: Zuerst wird die Menge aller von der Startecke in maximal 7 Schritten erreichbaren Ecken (zusammen mit den jeweiligen Pfaden) mit Iterative Deepening berechnet und in einer Datei gespeichert und sortiert (Größe: ca. 750MB; Zeitaufwand: ca. 6 Stunden). Anschliessend werden per Iterative Deepening die von der Zielecke erreichbaren Ecken ermittelt und per „Binary Search“ auf Existenz in der Datei geprüft. Bei Erfolg kann sofort mit einer garantiert optimalen Lösung abgebrochen werden, oder auch weitere Lösungen gesucht werden.

Auf diese Weise wurden die 14 optimalen Lösungen der großen Strategiegabe ermittelt, wobei dafür 10 Stunden Rechenzeit und 750MB freier Speicherplatz nötig waren.

6.6 Inselbasierte Suche

Die Idee, die zur bidirektionalen Suche führte, kann man auch noch erweitern, indem man nicht nur von Start- und Endecke aus sucht, sondern auch von weiteren Zwischenzielen und dadurch mehrere „Inseln“ zusammenwachsen lässt. Dies führt allerdings nur dann zu einem effektiven und optimalen Ergebnis, wenn man weiß, dass es eine optimale Lösung gibt, die auch tatsächlich die Zwischenziele durchläuft. Dies ist uns bei unserem Suchgraphen (mit bisherigem Wissen) aber leider nicht möglich.

7 Eigenschaften des Suchgraphen

Wir wollen nun den im letzten Kapitel eingeführten Suchgraphen $G = (V, E)$ betrachten. Wie wir wissen, hat er 1.961.990.553.600 Ecken des Grades 13, ist also 13-regulär und besitzt eine sehr geringe Kantendichte. Des weiteren fällt sofort auf, dass der Graph extrem symmetrisch ist, d.h. er besitzt sehr viele Automorphismen. Auch lässt sich leicht zeigen, dass der Radius des Graphen gleich seinem Durchmesser ist (d.h. der minimale Abstand der zwei am weitesten voneinander entfernten Ecken ist gleich dem Maximum der minimalen Abstände aller Ecken zu einer zentralen Ecke), da alle Ecken in G zentral sind. Wir wissen auch bereits, dass die Taillenweite des Graphen $g(G) = 4$ ist. Außerdem handelt es sich um einen bipartiten Graphen.

Um dies zu beweisen, definieren wir eine neue Normalform für Rochaden: Eine Rochade sei in 1. Normalform, wenn maximal ein Umdrehzeichen „D“ und zwar genau als letztes Zeichen in der Rochade vorkommt und sich ansonsten immer ein Dreh-Kippzeichen „\“ und ein Schiebezeichen abwechseln. Dabei entspreche „\“ dem Kippen des statischen Kugelelements, also „\“:=„D/“ (Im übrigen entspricht für jede Rochade in 1. Normalform die bisherige Definition der Länge dieser Rochade der Zahl an enthaltenden \-Zeichen). Damit gibt es zu jeder Rochade eine äquivalente in 1. Normalform, zum Beispiel sind /1/2/3/4/D und \-1\2\-3\4\ äquivalent. Rochaden in 1. Normalform haben ein paar sehr schöne Eigenschaften bezüglich der Permutations- und Verfärbungswirkung:

- Jede Rochade, die in 1. Normalform ist und nicht mit einem „D“ endet führt einen Zustand mit einer geraden Zahl an gelben Steinen auf der betrachteten Seite wieder in einen solchen über.
- Jede Rochade, die in 1. Normalform ist und mit einem „D“ endet führt einen Zustand mit einer geraden Zahl an gelben Steinen auf der betrachteten Seite in einen mit ungerader Zahl an gelben Steinen über.
- Jede Rochade, die in 1. Normalform ist und nicht mit einem „D“ endet führt einen Zustand mit einer geraden Permutation der Zahlen auf dem Zahlenkranz genau dann wieder in einen Zustand mit gerader Zahlenkranz-Permutation über, wenn ihre Länge gerade ist.
- Jede Rochade, die in 1. Normalform ist und mit einem „D“ endet führt einen Zustand mit einer geraden Permutation der Zahlen auf dem Zahlenkranz genau dann wieder in einen Zustand mit gerader Zahlenkranz-Permutation über, wenn ihre Länge nicht gerade ist.

Diese Eigenschaften folgen direkt daraus, dass das neu eingeführte Dreh-Kippzeichen eine gerade Zahl an Farbänderungen bewirkt und dass es – im Gegensatz zu den Schiebezeichen – eine ungerade Permutationswirkung hat. Damit lassen sich die Ecken des Graphen G so in zwei gleich mächtige disjunkte Klassen aufteilen, dass keine Kante in E zwei Ecken derselben Klasse verbindet. Man braucht als Aufteilungskriterium für jede Ecke nur zu entscheiden, ob der Zahlenkranz eines entsprechenden Zustandes – von der Seite betrachtet, die eine gerade Zahl an gelben Steinen aufweist – eine gerade oder eine ungerade Permutationsanordnung besitzt. Damit ist bewiesen, dass der Graph bipartit ist.

Ausserdem kann man für ein Paar von Start- und Zielzustand direkt ablesen, ob eine Rochade der gewünschten Wirkung eine gerade oder ungerade Länge besitzt, bzw. ob sie, wenn sie denn in 1. Normalform ist, mit einem „D“ endet, oder nicht:

- Ist die Zahl der nötigen Farbänderungen gerade, so endet eine passende Rochade in 1. Normalform nicht mit einem „D“.
- Ist die Zahl der nötigen Farbänderungen ungerade, so endet eine passende Rochade in 1. Normalform mit einem „D“.
- Ist die gewünschte Permutationswirkung gerade, so hat eine passende Rochade eine gerade Länge.
- Ist die gewünschte Permutationswirkung ungerade, so hat eine passende Rochade eine ungerade Länge.

Auf Rochaden in Standardform übertragen kann man die ersten beiden Regeln ersetzen durch:

- Ist die Zahl der nötigen Farbänderungen gerade, so endet eine passende Rochade in Standardform genau dann mit einem „D“, wenn sie von ungerader Länge ist.
- Ist die Zahl der nötigen Farbänderungen ungerade, so endet eine passende Rochade in Standardform genau dann mit einem „D“, wenn sie von gerader Länge ist.

Damit können wir zum Beispiel direkt nachweisen, dass es keine Zahlenrochaden ungerader Länge geben kann. Außerdem ist nun klar, dass jede Lösung der großen Strategiegabe eine Rochade ungerader Länge sein muß, die – in Standardform – mit einem „D“ endet.

Allerdings wäre es sehr interessant, zu wissen, wie viele Schritte man mindestens braucht, um von einem Zustand jeden anderen Zustand erreichen zu können. Die Frage ist also, wie groß der Durchmesser des Graphen G ist. Die Bedeutung dieser Zahl haben wir bereits bei der Betrachtung der verschiedenen Suchverfahren kennen gelernt. Auf Grund der dort berechneten Tabelle, liegt es nahe, dass der Durchmesser nicht viel größer als 13 ist (man setze die Tabelle einfach mit dem Wachstumsfaktor 10 fort). Kleiner kann er nicht sein, da wir mit der großen Strategieraufgabe bereits ein Beispiel kennen, bei dem passende Rochaden mindestens 13 Schritte lang sein müssen.

Der Autor hat sich lange vergeblich bemüht, seine ursprüngliche Vermutung, der Durchmesser betrage exakt 13, zu beweisen. Allerdings lässt sich leicht zeigen, dass es Zustände gibt, die sich nicht mit Rochaden ineinander überführen lassen, deren Länge weniger als 14 beträgt. Dazu betrachten wir die optimalen Lösungsrochaden der großen Strategieraufgabe etwas genauer. Es fällt auf, dass alle 14 Rochaden mit „/2D“, „/1D“ oder „/D“ enden. Erweitert man diese Rochaden nun jeweils durch Anhängen der Rochade „-3/“ so beschreiben alle 14 entstehenden Rochaden wieder dieselbe Wirkung. Diese Wirkung ist nur mit einer Rochade mit gerader Länge erreichbar. Dies kann aber unmöglich eine Rochade der Länge 12, oder weniger sein! Denn wäre es so, könnte man aus dieser Rochade durch Anhängen von „/3“ eine optimale Rochade für die große Strategieraufgabe gewinnen, die – in Standardform gebracht – mit „/-3D“ endet. So eine Rochade kann es aber nicht geben, da wir alle 14 optimalen Lösungen berechnet haben und keine von ihnen diese Form hat.

Also beträgt der Durchmesser, bzw. Radius des Suchgraphen mindestens 14.

8 Ausblick

Bei dem Brainball handelt es sich um ein sehr junges Produkt. Seit Markteinführung im Frühjahr 1999 wurden ca. 100.000 Stück vertrieben. Daher stellt diese Arbeit eine der ersten, wenn nicht die erste wissenschaftliche Auseinandersetzung mit diesem Spiel dar. Daher ist es nicht verwunderlich, dass noch viele Fragen unbeantwortet sind. Dazu zählt insbesondere die Frage nach dem Radius des Suchgraphen.

Außerdem ist es auch interessant, zu wissen, ob der Suchgraph hamiltonsch ist, beziehungsweise, wie lang ein einfacher Kreis maximaler Länge ist. Je länger dieser ist, desto tiefer kann eine unbeschränkte Teifensuche mit Zyklenerkennung im Suchgraphen vordringen, bevor zum ersten mal ein Zyklus erkannt wird.

Es ist im Übrigen nicht ausgeschlossen, dass es ein effektives Verfahren gibt, das – ohne den Lösungsraum zu durchsuchen – erlaubt, zu einer gegebenen Wirkung eine Rochade optimaler Länge direkt zu konstruieren oder zumindest diese Länge zu berechnen.

Zur weiteren Verfolgung dieser und ähnlicher Fragen ist möglicherweise eine Betrachtung mittels der Theorie endlicher Gruppen erfolgversprechender als der bisher verfolgte graphentheoretische Ansatz.

Literatur

- [1] L. Ammeraal : *Programmdesign und Algorithmen in C*. Hanser-Verlag (1989)
- [2] N. Biggs : *Discrete Mathematics*. Oxford University Press (1985)
- [3] R. Diestel : *Graphentheorie*, Springer-Verlag (1996).
- [4] D. Jungnickel : *Graphen, Netzwerke und Algorithmen*, 3. Auflage, BI-Wissenschaftsverlag (1994).
- [5] K. Königsberger : *Analysis 1*, 3. Auflage, Springer-Verlag (1995).
- [6] D. Poole, A. Mackworth & R. Goebel : *Computational Intelligence - A Logical Approach*, Oxford University Press (1998).
- [7] A. Unsicker : *Spielanleitung zum Brainball*, (1999).
- [8] www.Brainball.de, die offizielle Brainball-Homepage.
- [9] www.org2.com/jaap/puzzles/brain.htm, Informationen zum Brainball auf der Jaap-Page, einer Sammlung von Informationen und Lösungen zu den verschiedensten Puzzle-Spielen.