

Mastermind

Bachelorarbeit im Fach Mathematik
an der Ruhr-Universität Bochum



von Michael Kußmann
aus Essen

Bochum, 21. April 2008

Inhaltsverzeichnis

1	Einleitung	3
1.1	Das Spiel und seine Regeln	3
1.2	Die Problemstellung	4
1.3	Herangehensweise und Darstellung	5
2	Der worst-case-Ansatz	7
2.1	Das 2^3 -Mastermind	7
2.2	Donald Knuths Strategie	9
2.3	Ist Knuths Strategie optimal im worst-case-Sinn?	10
3	Der average-case-Ansatz	17
3.1	Die expected-size-Strategie	18
3.2	Die most-parts-Strategie	19
3.3	Die untere Schranke bei allgemeiner Form	21
4	Der realistische Ansatz	25
4.1	Die Minimax-Strategie beim 2^3 -Mastermind	25
4.2	Die Minimax-Strategie beim 6^4 -Mastermind	28
5	Quellenverzeichnis	29

1 Einleitung

Mastermind ist eines der erfolgreichsten Spiele der Siebzigerjahre. 1973 erschienen, wurde es innerhalb weniger Jahre über zehn Millionen mal verkauft und erfreut sich noch heute einer großen Fangemeinde. Dies liegt vermutlich an der Mischung aus simplen Regeln und komplexen kombinatorischen Möglichkeiten.

Die vorliegende Arbeit ist aus einem mündlichen Vortrag über „Mastermind“ entstanden, der im Rahmen des Seminars „Mathematik und Spiel“ gehalten wurde. Das Seminar basierte auf dem Buch „Glück, Logik und Bluff“ von Jörg Bewersdorff¹ und fand unter der Leitung von Professor Dr. R. Verfürth an der Ruhr-Universität Bochum im Wintersemester 07/08 statt. Die Arbeit soll einen Überblick über die mathematischen und spieltheoretischen Untersuchungen geben, die zwischen 1976 und 2005 zu Mastermind angestellt wurden.

1.1 Das Spiel und seine Regeln

Mastermind ist ein Spiel für zwei Personen. Ein Spieler übernimmt die Rolle des Codierers und steckt einen vierstelligen Geheimcode, den er beliebig aus sechs Farben auswählen kann, hinter eine Sichtblende. Sein Gegenüber muss nun versuchen, diesen Code zu knacken. Dazu stehen ihm auf dem Spielbrett zwölf Steckreihen zur Verfügung. Nach jedem Rateversuch erhält er vom Codierer zwei Informationen: Zum einen die Anzahl der exakten Treffer, in Form von schwarzen Stiften, zum anderen die Anzahl der Farbstifte, die erst nach geeigneter Permutation zu zusätzlichen Treffern werden, in Form von weißen Stiften. Bleibt zu erwähnen, dass nicht ersichtlich ist, auf welche Farbstifte sich die Angaben beziehen.

Wäre der Geheimcode $ABBA$ und der Rateversuch $AAAB$, so bekäme man als Antwort einen schwarzen und zwei weiße Stifte. Im Folgenden schreiben wir diese Antwort in der Form $[1, 2]$.

¹BEWERSDORFF (2001)

Das Spiel endet, sobald der Decodierer als Antwort $[4, 0]$ erhält und der Code geknackt ist.

1.2 Die Problemstellung

Wir wollen untersuchen, wie der Decodierer den Code am schnellsten knacken kann. Dazu müssen wir uns überlegen, welchen Kategorien man Mastermind zuordnen kann, da sich je nach Kategorie unterschiedliche Kriterien ergeben, die es zu optimieren gilt. Bewersdorff untersucht drei unterschiedliche Ansätze:

- „Im Sinne eines worst-case-Ansatzes kann man nach einer Strategie suchen, die in möglichst wenigen Zügen jeden beliebigen Code garantiert entschlüsselt. Dazu kann man sich etwa vorstellen, der Codierer dürfe mogeln und seinen Code während der Partie noch ändern, allerdings nur in Übereinstimmung mit bereits gegebenen Antworten. Das bedeutet, dass der Codierer jede Antwort frei wählen kann, so weit sie nicht zu den zuvor bereits gegebenen Antworten im Widerspruch steht. In dieser Interpretation erscheint Mastermind als zufallsfreies Zweipersonenspiel mit perfekter Information, dessen Minimax-Wert² berechnet werden kann.“³
- Im Sinne eines average-case-Ansatzes kann man nach einer Strategie suchen, die im Durchschnitt am schnellsten zum Ziel führt, wobei jeder Geheimcode gleichwahrscheinlich ist.⁴ Hier spielen Methoden der Wahrscheinlichkeitstheorie eine Rolle. Es handelt sich in diesem Fall um ein Einpersonenspiel, da die einzige aktive Handlung des Codierers durch eine Zufallsentscheidung ersetzt wird.

²Der Begriff wird in Abschnitt 2 erläutert.

³BEWERSDORFF (2001) S. 228f

⁴Bewersdorff schreibt an dieser Stelle, dass es naheliegend aber nicht zwingend ist, von gleichwahrscheinlichen Codes auszugehen. Er behandelt aber ausschließlich diesen Fall.

- Bei einem realistischen Ansatz muss man die strategischen Überlegungen des Codierers mit einbeziehen. Wir werden hinterfragen, wie stark der Codierer die Gewinnaussichten des Spiels beeinflussen kann. Hierbei handelt es sich um ein Zweipersonenspiel ohne perfekte Information und mit Minimax-Wert.

Wir werden alle drei Ansätze betrachten.

1.3 Herangehensweise und Darstellung

Wer das Spiel kennt, liebt es vielleicht, aus zwei oder drei Antworten logische Schlüsse zu ziehen und sich so Schritt für Schritt der Lösung zu nähern. Von dieser eleganten Spielweise müssen wir uns nun verabschieden und Mastermind wie ein Computer spielen, d.h. wir prüfen vor jedem Rateversuch:

- Welche Codes sind noch möglich?
- Wie verkleinern die verschiedenen Rateversuche und deren mögliche Antworten die Menge der möglichen Codes?

An die zweite Frage schließt sich die Suche nach einem „besten“ Rateversuch an. Wir werden sehen, dass sich der beste Rateversuch beim worst-case-Ansatz vom besten Rateversuch beim average-case-Ansatz unterscheidet.

Bevor wir mit den ersten Untersuchungen beginnen, benötigen wir noch eine Kurzschreibweise:

- Darstellung eines vierstelligen Codes $q \in C_0$: z. B. *EBBA*.
 C_0 ist die Menge aller 6^4 Codes.
- Darstellung einer Antwort $a \in A$ aus schwarzen, weißen Stiften: $[s, w]$.
 A ist die Menge aller Antworten mit $0 \leq (s + w) \leq 4$, mit Ausnahme der Antwort $[3, 1]$.
- Darstellung einer Spielposition: Menge der noch möglichen Codes.

- C_0 charakterisiert die Position vor dem ersten Versuch.
- $C(q, a)$ ist die Menge der Codes, die auf den Rateversuch q die Antwort a liefern, z. B. $C(AAAB, [2, 2]) = \{AABA; ABAA; BAAA\}$.
- $C = \bigcap_{t=1}^s C(q_t, a_t)$ ist die Menge der möglichen Codes nach s Rateversuchen $q_1, \dots, q_s$ mit den Antworten $a_1, \dots, a_s$.

Wenn der Durchschnitt C nur noch ein Element enthält, kann man den Code im nächsten Zug sicher knacken.

Da ein bestimmter Geheimcode auf jeden Rateversuch genau eine Antwort liefert, sind bei einem beliebigen, aber festen $q \in C_0$ die Mengen $C(q, a_i)$ mit $1 \leq i \leq \alpha$, $\alpha = \#$ Antworten, paarweise disjunkt.

Weiter gilt: $\bigcup_{i=1}^{\alpha} C(q, a_i) = C_0$ für alle $q \in C_0$.

2 Der worst-case-Ansatz

Der Codierer darf bei diesem Ansatz den Geheimcode nach jedem Rateversuch und *vor* seiner Antwort ändern - in Einklang mit den bisher gegebenen Antworten. Bei dieser Variante gibt es eine feste Anzahl an Rateversuchen, nach der bei perfektem Spiel beider Spieler der Code geknackt ist. Diese Zahl, den sogenannten Minimax-Wert, und die dazugehörige Strategie veröffentlichte Donald Knuth drei Jahre nach dem Erscheinen von Mastermind.⁵

Bevor wir zu Knuths Strategie und zur Bestimmung des Minimax-Wertes des 6⁴-Masterminds kommen, schauen wir uns die Vorgehensweise an einem einfachen Beispiel an.

2.1 Das 2³-Mastermind

Bei dieser Variante gibt es nur zwei Farben und der Code hat die Länge 3. Wie sollte man vorgehen, um den Geheimcode aus der Menge der acht Codes möglichst schnell zu knacken? Bis auf Symmetrie gibt es lediglich zwei Eröffnungen, *AAA* und *AAB*, da sich alle anderen sechs Codes durch Permutation der Farben und/ oder Positionen aus *AAA* bzw. *AAB* ergeben. Zwei Codes, die durch eine solche Permutation ineinander zu überführen sind, sind äquivalent und gehören zur selben Codeklasse.

Wir spielen zur Untersuchung beide Eröffnungen komplett durch (s. Tabellen 1 und 2 S. 8). Die ersten drei Spalten zeigen den ersten Zug, alle möglichen Antworten und die jeweils noch möglichen Codes. Dann folgt der zweite Zug usw. In der letzten Spalte steht die Anzahl der benötigten Züge für den jeweiligen Geheimcode.

Ein Blick genügt, um zu sehen, dass die *AAB*-Eröffnung besser ist. Bei geschicktem Vorgehen sind wir nach dem dritten Rateversuch auf jeden Fall fertig. Hier sei kurz auf den Rateversuch *ABB* in Tabelle 2 in der dritten Zeile, vierte Spalte hingewiesen. Mit diesem gelingt es, anders als beim

⁵KNUTH (1976/77)

q_1	a_1	C_1	q_2	a_2	C_2	q_3	a_3	C_3	q_4	#
AAA	[3, 0]	{AAA}	-							1
	[2, 0]	{AAB, ABA, BAA}	AAB	[3, 0]	{AAB}	-				2
				[1, 2]	{ABA, BAA}	ABA	[3, 0]	{ABA}	-	3
							[1, 2]	{BAA}	BAA	4
	[1, 0]	{ABB, BAB, BBA}	ABB	[3, 0]	{ABB}	-				2
				[1, 2]	{BAB, BBA}	BAB	[3, 0]	{BAB}	-	3
							[1, 2]	{BBA}	BBA	4
	[0, 0]	{BBB}	BBB	[3, 0]	{BBB}	-				2

Tabelle 1: Die AAA-Eröffnung

q_1	a_1	$C_1 = C(q_1, a_1)$	q_2	a_2	$C_2 = C(q_2, a_2) \cap C_1$	q_3	#
AAB	[3, 0]	{AAB}	-				1
	[2, 0]	{ABB, BAB, AAA}	ABB	[3, 0]	{ABB}	-	2
				[1, 2]	{BAB}	BAB	3
				[1, 0]	{AAA}	AAA	3
	[1, 2]	{ABA, BAA}	ABA	[3, 0]	{ABA}	-	2
				[1, 2]	{BAA}	BAA	3
	[1, 0]	{BBB}	BBB	[1, 2]	{BBB}	-	2
	[0, 2]	{BBA}	BBA	[3, 0]	{BBA}	-	2

Tabelle 2: Die AAB-Eröffnung

Rateversuch AAA, die Menge der drei möglichen Codes in drei einelementige Mengen zu unterteilen.

Wir erhalten somit unter dem worst-case-Ansatz für das 2^3 -Mastermind den Minimax-Wert 3 und als Minimax-Strategie für den Decodierer die AAB-Eröffnung, gespielt wie in Tabelle 2. Betrachten wir die Tabellen nun aus dem entgegengesetzten Blickwinkel. Der Codierer gibt als Minimax-Strategie zunächst die Antwort $a_1 = [2, 0]$, da diese bei beiden Eröffnungen die meisten Codes übrig lässt. Gehen wir davon aus, dass der Decodierer optimal spielt, so kann der Codierer im zweiten Zug zwischen den Antworten [1, 2] und [1, 0] wählen und sieht sich dann gezwungen im dritten Zug [3, 0] zu antworten.

2.2 Donald Knuths Strategie

Kommen wir nun zurück zum 6^4 -Mastermind und zu Knuths Versuch, die Zahl der maximal benötigten Züge zu minimieren. Beim 6^4 -Mastermind gibt es fünf Codeklassen und somit fünf verschiedene Eröffnungen. Der linke Kasten von Tabelle 3 gibt zu jeder Eröffnung an, wie viele Codes bei jeder möglichen Antwort übrig bleiben.

	1. Rateversuch					2. Rateversuch	
Antwort	<i>AAAA</i>	<i>AAAB</i>	<i>AABB</i>	<i>AABC</i>	<i>ABCD</i>	nach <i>AABB</i>	
[0, 4]			1	2	9	<i>BBAA</i>	fertig
[0, 3]			16	44	136	<i>ABAC</i>	4
[0, 2]		61	96	222	312	<i>BCDD</i>	18
[0, 1]		308	256	276	152	<i>BCDD</i>	44
[0, 0]	625	256	256	81	16	<i>CCDE</i>	46
[1, 3]				4	8		
[1, 2]		27	36	84	132	<i>ABAC</i>	7
[1, 1]		156	208	230	252	<i>AACD</i>	38
[1, 0]	500	317	256	182	108	<i>ACDD</i>	44
[2, 2]		3	4	5	6	<i>ABAC</i>	1
[2, 1]		24	32	40	48	<i>ABBC</i>	6
[2, 0]	150	123	114	105	96	<i>ABCD</i>	20
[3, 0]	20	20	20	20	20	<i>ABBC</i>	5
[4, 0]	1	1	1	1	1	fertig	

Tabelle 3: *Nach den ersten beiden Zügen [maximal] übrig bleibende Codes*
(BEWERSDORFF (2001) S. 230)

Beim dick gedruckten Rateversuch *AABB* ist die Menge der verbleibenden Codes im worst case am kleinsten, maximal 256 Codes bleiben übrig, weshalb Knuth diese Eröffnung empfiehlt. Im rechten Kasten steht der nächste Rateversuch, der sich je nach Antwort anbietet. Dieser ist wieder so gewählt, dass sich die Anzahl der Codes im ungünstigsten Fall am stärksten verringert. Die rechte Spalte gibt an, wie viele Codes nach dem zweiten Versuch in eben diesem worst case übrig bleiben.

Anzumerken wäre noch der Rateversuch *ABAC* auf die Antwort $[2, 2]$. Im gesuchten Code kann das „C“ natürlich nicht vorkommen, doch nur mit diesem Trick gelingt die sichere Reduzierung auf nur noch einen Code, da auf *ABAC* in dieser Situation vier Antworten möglich sind, welche die Menge der vier möglichen Codes in vier einelementige Mengen unterteilen.

Knuth zeigte, dass auf diese Weise jeder Code spätestens im fünften Versuch geknackt ist, also mit vier schwarzen Stiften beantwortet wird. Der Beweis ist umfangreich, aber vom Prinzip her simpel. Man könnte zum Beispiel eine Fallunterscheidung wie in Tabelle 1 bzw. 2 vornehmen (vgl. S. 8), die Tabelle hierzu besäße jedoch 1296 Zeilen – für jeden Code eine.

Wie gut Knuths Strategie ist, wird besonders bei Betrachtung der naiven Strategie deutlich: Bei der naiven Strategie, auch lexikographische Strategie genannt, tippt man stets von den noch möglichen Codes denjenigen, der im Alphabet als erstes kommt. Beginnend mit der Eröffnung *AAAA* benötigt die naive Strategie im worst case neun Versuche.

2.3 Ist Knuths Strategie optimal im worst-case-Sinn?

Sobald man eine Strategie gefunden hat, die jeden Code sicher in fünf Zügen knackt, drängt sich die Frage auf, ob es auch eine Strategie gibt, die stets mit vier Zügen auskommt. Mit Hilfe der folgenden Sätze werden wir untersuchen, ob es eine solche Strategie geben kann. Die Sätze gehen auf eine Abschätzung zurück, die Viaud 1979 im Rahmen allgemeiner Untersuchungen von sequentiellen Suchspielen vorlegte.⁶

SATZ 2.1 (UNTERE SCHRANKE DER MAXIMALEN ZUGANZAHL I).

Wir nehmen an, es gäbe eine perfekte Strategie, die es schafft, die Menge der noch möglichen Codes in jedem Zug in die einelementige Teilmenge des Volltreffers und $(\alpha - 1)$ gleich große Teilmengen zu unterteilen.⁷ Sei $l(C)$ die Funktion, die zu einer Codemenge C und α möglichen Antworten die

⁶VIAUD (1979)

⁷Die Annahme einer solchen Strategie ist unrealistisch, da sich bei jeder Eröffnung die einzelnen Teilmengen in ihrer Größe mitunter deutlich unterscheiden (s. Tabelle 3 S. 9).

maximale Zuganzahl bei perfekter Strategie liefert. Dann ist $l(C)$ die untere Schranke der maximalen Zuganzahl, gegeben durch

$$l(C) = m + 1,$$

wobei m durch die folgende Ungleichung festgelegt wird:

$$\frac{(\alpha - 1)^m - 1}{\alpha - 2} < |C| \leq \frac{(\alpha - 1)^{m+1} - 1}{\alpha - 2}.$$

BEWEIS. Sei $|C_m|$ die größte Mächtigkeit an Codes, bei der es überhaupt noch eine Strategie zur sicheren Decodierung in m Versuchen geben kann – nämlich die perfekte Strategie.

$m = 1$: Der erste Versuch muss den Code knacken.

$$\Rightarrow |C_1| = 1.$$

$m = 2$: Der erste Versuch zerlegt in die einelementige Teilmenge des Volltreffers und $(\alpha - 1)$ Teilmengen, welche je ein Element enthalten.

$$\Rightarrow |C_2| = 1 + (\alpha - 1)|C_1| = 1 + (\alpha - 1).$$

$m > 2$: Der erste Versuch zerlegt in die einelementige Teilmenge des Volltreffers und $(\alpha - 1)$ gleich große Teilmengen, so dass bei jeder der $(\alpha - 1)$ Mengen $(m - 1)$ weitere Versuche reichen.

$$\Rightarrow |C_m| = 1 + (\alpha - 1)|C_{m-1}|.$$

Vollständige Induktion liefert:

$$\begin{aligned} |C_m| &= 1 + (\alpha - 1) + (\alpha - 1)^2 + \dots + (\alpha - 1)^{m-1} \\ &\stackrel{\text{geom.}}{=} \frac{(\alpha - 1)^m - 1}{(\alpha - 1) - 1} \\ &\stackrel{\text{Reihe}}{=} \frac{(\alpha - 1)^m - 1}{\alpha - 2}. \end{aligned}$$

Ist

$$|C| > |C_m| = \frac{(\alpha - 1)^m - 1}{\alpha - 2},$$

so folgt direkt, dass man mindestens $(m + 1)$ Versuche zur sicheren Dekodierung benötigt.

Gilt darüber hinaus

$$|C| \leq |C_{m+1}| = \frac{(\alpha - 1)^{m+1} - 1}{\alpha - 2},$$

so folgt weiter, dass man bei perfekter Strategie dazu genau $(m+1)$ Versuche benötigt.

Damit ist $l(C)$ die untere Schranke der maximalen Zuganzahl, da wir nicht schneller zum Ziel kommen können als mit der perfekten Strategie. $\square$

Bei dieser Version gelangen wir durch Einsetzen verschiedener Werte in die Ungleichung an unsere untere Schranke der maximalen Zuganzahl. Eine etwas elegantere Art zeigt uns der folgende Satz.

SATZ 2.2 (UNTERE SCHRANKE DER MAXIMALEN ZUGANZAHL II).

Seien $k > 3$ die Anzahl der Farben, $n > 0$ die Codelänge und $C(k^n)$ die Menge der daraus resultierenden k^n Codes. Dann sind im gesamten Spiel und bei einem geeigneten Rateversuch alle Antworten $[s, w]$ mit $0 \leq w \leq n$, $0 \leq s \leq (n - w)$ möglich, mit der einzigen Ausnahme $[n - 1, 1]$. Für die Summe α der möglichen Antworten gilt somit:

$$\alpha = \sum_{i=2}^{n+1} i. \quad (1)$$

Damit folgt für die untere Schranke der maximalen Zuganzahl in alleiniger Abhängigkeit von k und n :

$$l(C(k^n)) = \lceil \log_{-1 + \sum_{i=2}^{n+1} i} \{(-2 + \sum_{i=2}^{n+1} i) \cdot k^n + 1\} \rceil, \quad n > 1, \quad (2)$$

$$l(C(k^n)) = k, \quad n = 1. \quad (3)$$

BEWEIS. AD (1): Wir wollen zunächst zeigen, dass man im Falle $k > 3$ zu jeder Anzahl Steckplätze $n > 0$ für einen geeigneten Geheimcode Rateversuche finden kann, die alle Antworten $[s, w]$ mit $0 \leq w \leq n$, $0 \leq s \leq (n - w)$ abdecken, mit der einzigen Ausnahme $[n - 1, 1]$, und dass die Summe der Antworten $\alpha = \sum_{i=2}^{n+1} i$ beträgt. Hierzu machen wir eine Fallunterscheidung:

$n = 1$: Es gibt offensichtlich nur die beiden Antworten $[1, 0]$ und $[0, 0]$, bei richtiger bzw. falscher Farb-/ Codewahl. Somit gilt:

$$\alpha = 2 = \sum_{i=2}^{1+1} i.$$

$n = 2$: Als Geheimcode nehmen wir einen beliebigen zweifarbigen Code. Ohne Permutation ist $w = 0$. Durch Ersetzen keines, eines oder beider Farbstifte durch eine dritte Farbe erreicht man $0 \leq s \leq 2$, also die Antworten $[2, 0]$, $[1, 0]$ und $[0, 0]$. $w = 1$ erhält man, indem man beide Farbstifte vertauscht und einen durch eine dritte Farbe ersetzt. Dies ergibt die Antwort $[0, 1]$. Vertauscht man lediglich die Farbstifte des Geheimcodes, so bekommt man $[0, 2]$. Damit sind alle verlangten Antworten gezeigt. Darüber hinaus gilt:

$$\alpha = 3 + 1 + 1 = \sum_{i=2}^{2+1} i.$$

$n > 2$: Wir nehmen einen dreifarbigen Geheimcode, bei dem alle drei Farben so gut es geht gleichhäufig vorkommen. Ohne Permutation des Geheimcodes ist $w = 0$. Durch Vertauschen zweier unterschiedlicher Farbstifte des Geheimcodes und Ersetzen des einen durch eine vierte Farbe erhält man eine Antwort mit $w = 1$. Des weiteren führt eine geeignete Kombination aus Permutationen von zwei bzw. drei unterschiedlichen Farbstiften des Geheimcodes zu einer Antwort mit jedem gewünschten w mit $2 \leq w \leq n$.⁸ Die Anzahl s der schwarzen Stifte kann man anschließend durch Ersetzen exakter Farbstifte durch eine vierte Farbe zwischen 0 und $(n - w)$ variieren. Somit hat man in Abhängigkeit von w genau $(n + 1 - w)$ Möglichkeiten für s . Da für den Fall $w = 1$ ein Farbstift des Geheimcodes durch eine falsche Farbe ersetzt wird, kann es die Antwort $[n - 1, 1]$ nicht geben. Wir haben alle verlangten Antworten konstruiert.

Damit ergibt sich für die Anzahl der Antworten α die Summe der Möglichkeiten schwarzer Stifte bei jeweiliger Anzahl weißer Stifte abzüglich der nicht

⁸Zum Beispiel könnte man für $w = 8$ beim 10^{10} -Mastermind zweimal drei und einmal zwei unterschiedliche Farbstifte permutieren, wobei kein Farbstift in mehr als einer Permutation vorkommen darf.

möglichen Antwort $[n - 1, 1]$:

$$\begin{aligned}\alpha &= -1 + \sum_{w=0}^n n + 1 - w = -1 + \sum_{w=1}^{n+1} w \\ &= \sum_{w=2}^{n+1} w = \sum_{i=2}^{n+1} i.\end{aligned}$$

Aufgrund der Symmetrie des Spiels ergibt sich bei Vertauschung von Geheimcode und Rateversuch dieselbe Antwort.

Daraus folgt:

Wenn wir aus einem bestimmten Geheimcode Rateversuche konstruieren können, die auf diesen Geheimcode zusammen alle Antworten liefern, so sind auf diesen bestimmten (Geheim-)Code als *Rateversuch* alle Antworten möglich.

AD (2): Kommen wir zum Beweis von $l(C(k^n))$ in Abhängigkeit von k und n , zunächst für $n > 1$.

Dabei benutzen wir die Bedingung von $l(C)$ aus Satz 2.1:

$$\frac{(\alpha - 1)^m - 1}{\alpha - 2} < |C| \leq \frac{(\alpha - 1)^{m+1} - 1}{\alpha - 2}.$$

Wir setzen $|C| = k^n$ und machen aus der rechten Ungleichung eine Gleichung, indem wir $(m+1)$ durch $x \in \mathbb{R}$ ersetzen, und halten dabei fest, dass der nächst höhere ganze Wert der Lösung x die gesuchte untere Schranke ist.

$$\begin{aligned}k^n &= \frac{(\alpha - 1)^x - 1}{\alpha - 2} \\ \Leftrightarrow (\alpha - 2) \cdot k^n &= (\alpha - 1)^x - 1 \\ \Leftrightarrow (\alpha - 2) \cdot k^n + 1 &= (\alpha - 1)^x \\ \Leftrightarrow \log_{\alpha-1} \{(\alpha - 2) \cdot k^n + 1\} &= x \\ \Rightarrow \lceil \log_{-1+\sum_{i=2}^{n+1} i} \{(-2 + \sum_{i=2}^{n+1} i) \cdot k^n + 1\} \rceil &= l(C(k^n)).\end{aligned}$$

AD (3): Bleibt noch der Fall $n = 1$.

Da es nur die beiden Antworten $[1, 0]$ und $[0, 0]$ gibt, erhält man auf einen Rateversuch nur die Information, ob die getippte Farbe der Geheimcode ist oder nicht. Somit muss man im worst case alle k Farben durchprobieren. $\square$

Wir wollen noch kurz auf die Forderung $k > 3$ eingehen:

$k = 1$ macht offensichtlich keinen Sinn. $k = 2$ lässt nicht alle sonst möglichen Antworten zu, zum Beispiel kann es beim 2^3 -Mastermind nicht die Antworten $[0, 1]$, $[1, 1]$ und $[0, 3]$ geben (s. Tabelle 1 und 2, S. 8). Beim 3^3 -Mastermind sind diese zwar alle möglich, aber immer noch nicht bei demselben Rateversuch. So ist die Antwort $[0, 3]$ nur bei Rateversuchen mit 3 Farben möglich, dagegen kann die Antwort $[0, 0]$ nur bei Rateversuchen mit weniger als 3 Farben eintreten. Daher schließen wir auch diesen Fall aus.

Kommen wir zurück zu unserer Ausgangsfrage, ob es beim 6^4 -Mastermind eine Strategie geben kann, die sicher in vier Zügen löst.

Dazu bestimmen wir die untere Schranke mithilfe von Satz 2.2:

$$\begin{aligned}
 l(C(6^4)) &= \lceil \log_{-1+\sum_{i=2}^{4+1} i} \{(-2 + \sum_{i=2}^{4+1} i) \cdot 6^4 + 1\} \rceil \\
 &= \lceil \log_{-1+14} \{(-2 + 14) \cdot 1296 + 1\} \rceil \\
 &= \lceil \log_{13} \{12 \cdot 1296 + 1\} \rceil \\
 &= \lceil 3, 76 \rceil \\
 &= 4.
 \end{aligned}$$

Bei perfekter Strategie wäre ein sicheres Knacken des Codes in vier Zügen möglich. Dass dies nicht der Realität entspricht, können wir mit einer kleinen Überlegung zeigen: Mit dem ersten Zug lässt sich keine Reduktion auf weniger als 256 Codes erzwingen (s. Tabelle 3 S. 9). Mit diesem Wissen suchen wir die untere Schranke *nach* dem ersten Zug. Anders ausgedrückt suchen wir die untere Schranke des 6^4 -Masterminds mit nur 256 möglichen Codes, kurz $6^4|256$, wieder mit Satz 2.2:

$$\begin{aligned}
 l(C(6^4|256)) &= \lceil \log_{-1+\sum_{i=2}^{4+1} i} \{(-2 + \sum_{i=2}^{4+1} i) \cdot 256 + 1\} \rceil \\
 &= \lceil \log_{13} \{12 \cdot 256 + 1\} \rceil \\
 &= \lceil 3, 13 \rceil \\
 &= 4.
 \end{aligned}$$

Damit ist die untere Schranke für das ganze Spiel fünf.

Wir haben gezeigt, dass zur sicheren Erkennung (mindestens) fünf Züge nötig sind, d.h. dass Knuths Strategie optimal im worst-case-Sinn ist.

Wir erhalten somit unter dem worst-case-Ansatz für das 6^4 -Mastermind den Minimax-Wert 5 und Knuths Strategie als eine Minimax-Strategie für den Decodierer. Die Minimax-Strategie des Codierers gibt als Antwort stets den oder einen der worst cases aus Sicht des Decodierers.

3 Der average-case-Ansatz

Bei diesem Ansatz wird der Geheimcode per Zufall ausgewählt, wobei alle Codes gleichwahrscheinlich sind. Wir suchen die Strategie, mit der man im Durchschnitt am wenigsten Züge zum Knacken des Codes benötigt. Wie Knuth in seinem Artikel von 1976 bemerkte, ist „seine Strategie im Hinblick auf die durchschnittliche Zuganzahl zwar nicht optimal, wohl aber nahezu optimal.“⁹ Knuths worst-case-Strategie benötigt im Durchschnitt 4,478 (= 5804/1296) Züge.¹⁰ Diesen Wert erhält man, indem man bei fester Strategie alle Geheimcodes durchspielt, die benötigten Züge addiert und die Summe durch die Gesamtanzahl der Codes teilt.

Mit fester Strategie ist gemeint, dass man in einer bestimmten Situation immer exakt den gleichen Code wählt, d.h. man beginnt immer mit dem gleichen Rateversuch, in diesem Fall *AABB*, und tippt auf eine bestimmte Antwort a_1 immer gleich und nach zwei bestimmten Antworten a_1 und a_2 genauso usw.

Bereits 1978 wurde sein Wert von Irving verbessert.¹¹ Seine optimierte expected-size-Strategie benötigt im Durchschnitt 4,369 (= 5662/1296) Züge.

Wenige Jahre später suchte Neuwirth eine bessere Strategie mit Hilfe des Computers.¹² Aufgrund der damals geringen verfügbaren Rechenleistung untersuchte er nur Strategien, bei denen der Codierer stets Codes tippt, die noch möglich sind. Diese Einschränkung verkleinert die zu untersuchende Menge deutlich, wenn man auch Gefahr läuft, gute Strategien auszuschließen. Darüber hinaus musste er sich auf eine 5⁴-Variante beschränken. Trotzdem konnte er durch Kombination verschiedener Ansätze Irvings Ergebnis leicht verbessern, nämlich auf durchschnittlich 4,364 (= 5656/1296) Züge bis zum

⁹BEWERSDORFF (2001) S. 231

¹⁰Dieser Wert ist von Bewersdorff übernommen, welcher ihn wiederum aus Irvings Veröffentlichung hat. Barteld Kooi dagegen berechnete für Knuths Strategie den Erwartungswert 4,476 (= 5801/1296). Kooi glaubt, Irving habe bei seiner Berechnung einen Programmierfehler gemacht - denselben, den auch er erst gemacht hat.

¹¹IRVING (1978/79)

¹²NEUWIRTH (1982)

Knacken des Codes.

Die endgültig beste Strategie im average-case-Sinn wurde 1993 von Kenji Koyama und Tony Lai veröffentlicht und basiert auf vollständiger Optimierung.¹³ Bei dieser optimalen Strategie kommt der Codierer mit durchschnittlich 4,340 ($= 5625/1296$) Versuchen zum Ziel. Schlimmstenfalls sind es jedoch sechs Züge. Es bedarf einer kleinen Modifikation, um die Anzahl der Versuche auf fünf zu beschränken, wobei sich der Durchschnitt auf 4,341 ($= 5626/1296$) erhöht.

Die meines Wissens neuste Publikation stammt aus dem Jahre 2005 von Barteld Kooi, in der die most-parts-Strategie vorgestellt wird. Mit ihr benötigt der Codierer im Durchschnitt 4,373 ($= 5668/1296$) Züge.¹⁴

Die expected-size- sowie die most-parts-Strategie wollen wir näher betrachten.

3.1 Die expected-size-Strategie

Bei Irvings Strategie geht es darum, wie der Name andeutet, die zu erwartende Anzahl der nach dem Rateversuch übrigen Codes zu minimieren. Dies erreichen wir über die Berechnung des Erwartungswertes. Sehen wir uns noch einmal Tabelle 3 an (s. S. 9) und berechnen als Beispiel die zu erwartende Codemächtigkeit nach der Eröffnung AAAA:

$$\frac{625}{1296} \cdot 625 + \frac{500}{1296} \cdot 500 + \frac{150}{1296} \cdot 150 + \frac{20}{1296} \cdot 20 + \frac{1}{1296} \cdot 1 = 511,98.$$

Analog ergeben sich die Werte 204,54 für die AABB-Eröffnung, 185,27 für die AABC-Eröffnung und 188,19 für die ABCD-Eröffnung. Wieso die AABC- und ABCD-Eröffnungen im Durchschnitt die kleineren Codemengen liefern, lässt sich, wiederum mit einem Blick auf Tabelle 3, schnell einsehen. Es verwundert zunächst, da der höchste Wert der AABC- bzw. ABCD-Eröffnung

¹³KOYAMA UND LAI (1993)

¹⁴KOOI (2005)

mit 276 bzw. sogar 312 größer ist als 256 und beim Erwartungswert quadratisch eingeht. Jedoch ist auf die Frage *AABB* die Antwort $[1,3]$ nicht möglich, auf die Fragen *AABC* und *ABCD* dagegen schon. Aus diesem Grund teilt *AABB* die Menge in nur 13 nicht leere Teilmengen auf, *AABC* und *ABCD* dagegen in derer 14.

Irving rät zur *AABC* Eröffnung, da bei ihr die zu erwartende Codemenge am kleinsten ausfällt. Den zweiten Rateversuch bestimmte Irving nach dem gleichen Kriterium. Danach wird die Menge der Codes so überschaubar, dass er sie explizit untersuchen und optimieren konnte. Wenn wir die reine expected-size-Strategie spielen, d.h. wenn wir jeden Rateversuch dahingehend optimieren, dass die zu erwartende Codemenge möglichst klein ist, benötigen wir erwartungsgemäß 4,395 ($= 5696/1296$) Züge. Durch explizites Optimieren nach dem zweiten Rateversuch gelang es Irving, wie bereits erwähnt, in durchschnittlich 4,369 ($= 5662/1296$) Zügen den Code zu knacken.¹⁵

3.2 Die most-parts-Strategie

Bei dieser Strategie wollen wir die verbleibende Codemenge in möglichst viele Mengen unterteilen. Das heißt, wir wollen immer so fragen, dass wir möglichst viele verschiedene Antworten erhalten können.

Um besser verstehen zu können, wie Koois Strategie motiviert ist, schauen wir uns ein simples Beispiel eines sequentiellen Suchspiels an. Spieler 1 soll raten, welche Karte Spieler 2 zufällig aus einem Skatblatt mit 32 Karten gezogen hat. Bevor er seinen eigentlichen Tipp abgibt, darf Spieler 1 eine Frage stellen, die Spieler 2 mit „Ja“ oder „Nein“ beantworten kann. Intuitiv kommt einem die Frage „Ist es die Kreuzdame?“ schlecht, die Frage „Ist die Karte rot?“ dagegen gut vor. Man kann schnell zeigen, dass die Intuition trügt.

Die Frage nach der Kreuzdame unterteilt die 32 Karten in die Kreuzdame

¹⁵Kooi gelang es nicht Irvings Wert zu reproduzieren. Irvings zweiter Rateversuch auf die Antwort $[3,0]$ ist nach seiner eigenen Strategie schlicht falsch. Es scheint, als hätte er in diesem Fall nach der most-parts-Strategie (s. Abschnitt 3.2) optimiert, aber auch unter diesem Ansatz gelang es Kooi nicht die Ergebnisse zu reproduzieren.

und die restlichen 31 Karten, die Frage nach der Farbe unterteilt in zwei Teilmengen mit je 16 Karten. Der Erwartungswert, mit der einen oder anderen Frage zu gewinnen, ist in beiden Fällen gleich:

$$\begin{aligned}
 & \frac{1}{32} \cdot \frac{1}{1} + \frac{31}{32} \cdot \frac{1}{31} && \text{Ist es die Kreuzdame?} \\
 = & \frac{16}{32} \cdot \frac{1}{16} + \frac{16}{32} \cdot \frac{1}{16} && \text{Ist die Karte rot?} \\
 = & \frac{2}{32}.
 \end{aligned}$$

Egal mit welcher „Ja/ Nein“-Frage man die Menge teilt, die Gewinnchance bleibt immer

$$\frac{x}{32} \cdot \frac{1}{x} + \frac{y}{32} \cdot \frac{1}{y} = \frac{2}{32},$$

solange x und y größer null sind, d.h. solange beide Antworten möglich sind.

Dies kann man verallgemeinern. Wir wollen aus einer Menge M das gesuchte Element erraten. Bevor wir jedoch raten, können wir eine Frage stellen, die uns Aufschluss darüber gibt, in welcher der disjunkten, nicht-leeren Mengen M_i , $1 \leq i \leq n$ sich das gesuchte Element befindet. Die Wahrscheinlichkeit, das gesuchte Element zu erraten, beträgt dann

$$\sum_{i=1}^n \left\{ \frac{|M_i|}{|M|} \cdot \frac{1}{|M_i|} \right\} = \frac{n}{|M|}.$$

Somit spielt die Größe der einzelnen Teilmengen M_i keine Rolle, sondern einzig und allein die Anzahl n der Teilmengen, in welche unsere Frage die Menge M unterteilt.

Dieses Prinzip kann man auf Mastermind übertragen, mit dem einzigen Unterschied, dass jede Frage zugleich ein Rateversuch ist.

Nach Koois Strategie wählt der Decodierer als erstes diejenigen Rateversuche aus, auf die er zu diesem Zeitpunkt des Spiels die meisten Antworten erhalten kann. Von diesen nimmt er, falls möglich, diejenigen, die noch den Geheimcode darstellen können. Zuletzt wählt er den lexikographisch ersten Code. Beim ersten Rateversuch ist dies $AABC$, da er lexikographisch vor $ABCD$ kommt.

Kooi hat getestet, wie sich seine Strategie verändert, wenn man immer den lexikographisch letzten Code auswählt. Es macht einen Unterschied, wenn er auch verschwindend gering ist. So knackt man in der Summe einen einzigen Geheimcode eine Runde früher. Der Durchschnitt beträgt jedoch ebenfalls 4,373, da man in diesem Fall aufrundet, während beim lexikographisch ersten Code abgerundet wird.

Soweit mir bekannt ist, ist die most-parts-Strategie die beste reine Strategie. Anders ausgedrückt ist sie die beste aus der Gruppe der Strategien, die nur eine Antwort vorausschauen und immer nach dem gleichen Prinzip auswählen. Sie ist mit ihren durchschnittlich 4,373 (= 5668/1296) Zügen gerade mal 0,8% schlechter als Koyamas und Lais optimale Strategie.

3.3 Die untere Schranke bei allgemeiner Form

Abschließend wollen wir eine untere Schranke für die erwartete Zuganzahl setzen, und zwar für den allgemeinen Fall von $|C|$ Codes und α Antworten. Satz 3.1 geht wie schon Satz 2.1 und Satz 2.2 auf Viaud zurück. In Satz 3.2 werden wir sehen, wie Neuwirth aus diesen Überlegungen eine größere untere Schranke gewann, indem er einen Schritt weiter ging.¹⁶

SATZ 3.1 (UNTERE SCHRANKE DER ERWARTETEN ZUGANZAHL I).

Sei $e(C)$ die Funktion, die zu einer Codemenge C und α möglichen Antworten die erwartete Zuganzahl bei perfekter Strategie liefert. Dann ist $e(C)$ die untere Schranke der erwarteten Zuganzahl, gegeben durch

$$e(C) = \frac{1}{|C|} \cdot \left[\sum_{i=1}^m \{(\alpha - 1)^{i-1} \cdot i\} + (|C| - \frac{(\alpha - 1)^m - 1}{\alpha - 2}) \cdot (m + 1) \right],^{17}$$

¹⁶NEUWIRTH (1982)

¹⁷In dem mir vorliegenden Artikel von Neuwirth ist diese Funktion gegeben durch: $e(C) = \sum_{i=1}^{m-1} \{(\alpha - 1)^i \cdot i\} + (|C| - \frac{(\alpha-1)^m-1}{\alpha-2}) \cdot m$ (Zum besseren Vergleich habe ich Neuwirths Variablen durch meine Pendants ersetzt.). In Neuwirths Funktion wird jeder Code mit einem Zug zuwenig multipliziert. Darüber hinaus wird das Ergebnis nicht durch die

wobei m durch die folgende Ungleichung festgelegt wird:

$$\frac{(\alpha - 1)^m - 1}{\alpha - 2} < |C| \leq \frac{(\alpha - 1)^{m+1} - 1}{\alpha - 2}.$$

BEWEIS. Wir bestimmen für unsere Menge C zunächst das m , für das gilt:

$$\frac{(\alpha - 1)^m - 1}{\alpha - 2} < |C| \leq \frac{(\alpha - 1)^{m+1} - 1}{\alpha - 2}$$

Nach Satz 2.1 lassen sich bei perfekter Strategie ein Code in einem Zug, $(\alpha - 1)$ Codes in zwei Zügen, $(\alpha - 1)^2$ Codes in drei Zügen und so weiter und schließlich $(\alpha - 1)^{m-1}$ Codes in m Zügen knacken. Die verbleibenden Codes knackt man im $(m + 1)$ -ten und letzten Zug. Die Anzahl der verbleibenden Codes beträgt:

$$\begin{aligned} & |C| - 1 - (\alpha - 1) - (\alpha - 1)^2 - \dots - (\alpha - 1)^{m-1} \\ \stackrel{\text{geom.}}{=} & |C| - \frac{(\alpha - 1)^m - 1}{(\alpha - 1) - 1} \\ \text{Reihe} & \\ = & |C| - \frac{(\alpha - 1)^m - 1}{\alpha - 2}. \end{aligned}$$

Spielt man jeden Geheimcode bei fester Strategie einmal durch und addiert die benötigten Züge auf, so erhält man:

$$\begin{aligned} & 1 \cdot 1 + (\alpha - 1) \cdot 2 + (\alpha - 1)^2 \cdot 3 + \dots + (\alpha - 1)^{m-1} \cdot m \\ & + \left(|C| - \frac{(\alpha - 1)^m - 1}{\alpha - 2}\right) \cdot (m + 1) \\ = & \sum_{i=1}^m \{(\alpha - 1)^{i-1} \cdot i\} + \left(|C| - \frac{(\alpha - 1)^m - 1}{\alpha - 2}\right) \cdot (m + 1). \end{aligned}$$

Errechnet man den Erwartungswert, indem man diesen Wert durch die Gesamtanzahl der Codes teilt, erhält man $e(C)$.

Dies muss die untere Schranke der erwarteten Zuganzahl sein, da wir im Durchschnitt nicht schneller zum Ziel kommen können als mit der perfekten Strategie. $\square$

Gesamtanzahl der Codes geteilt, obwohl $e(C)$ auch bei Neuwirth die erwartete Zuganzahl liefern soll. Da Neuwirth in der folgenden Tabelle aber genau auf die Ergebnisse kommt, die meine Funktion $e(C)$ liefert, vermute ich hier einen Fehler des Textsetzers.

$e(C)$ lässt sich für Codemengen in der Größenordnung des 6^4 -Masterminds schnell berechnen. Die Berechnung der folgenden unteren Schranke in Satz 3.2 ist wesentlich aufwändiger und praktisch nur noch mit einem Computer durchführbar. Diese untere Schranke berechnet sich in zwei Schritten. Im ersten Schritt spielen wir einen echten ersten Rateversuch $q \in C_0$ und berechnen für jede der möglichen Antwortmengen $C(q, a_i)$ die untere Schranke $e(C(q, a_i))$, als würden wir ab dem zweiten Zug die perfekte Strategie spielen. Mit diesen berechnen wir den Erwartungswert der Zuganzahl bei der Eröffnung q , genannt $e_1(q)$. Im zweiten Schritt suchen wir denjenigen Code $q \in C_0$, für den $e_1(q)$ minimal wird, und erhalten so unsere untere Schranke. Dieser Mehraufwand rechtfertigt sich dadurch, dass diese untere Schranke größer ist als $e(C)$.

SATZ 3.2 (UNTERE SCHRANKE DER ERWARTETEN ZUGANZAHL II).

Sei $e(C)$ wie in Satz 3.1. Sei weiter $e_1(q)$ die Funktion, die zu einem ersten Rateversuch $q \in C_0$ die erwartete Zuganzahl liefert, sofern man ab dem zweiten Zug die perfekte Strategie verfolgt.

Sie ist gegeben durch

$$e_1(q) = 1 + \frac{1}{|C_0|} \cdot \sum_{a_i \in A \setminus \{[4,0]\}} \{|C(q, a_i)| \cdot e(C(q, a_i))\}. \quad (1)$$

Dann ist die untere Schranke der erwarteten Zuganzahl, sofern man einen realen ersten Rateversuch einbezieht, gegeben durch

$$\min_{q \in C_0} \{e_1(q)\}. \quad (2)$$

BEWEIS. AD (1): Wir bestimmen zunächst für einen beliebigen aber festen ersten Versuch $q \in C_0$ sämtliche Mengen $C(q, a_i)$ mit $\bigcup a_i = A \setminus \{[4,0]\}$. Für diese Mengen berechnen wir den Wert $e(C(q, a_i))$, die zu erwartende Zuganzahl nach dem ersten Versuch q und Antwort a_i bei perfekter Strategie. Anschließend multiplizieren wir jedes $e(C(q, a_i))$ mit der Wahrscheinlichkeit seines Eintretens $|C(q, a_i)|/|C_0|$ und summieren alle Produkte auf. Damit erhalten wir den Erwartungswert der Zuganzahl nach dem ersten Versuch q . Addieren wir noch den ersten Zug dazu, so erhalten wir $e_1(q)$.

AD (2): Die untere Schranke erhalten wir, wenn wir das Minimum von $e_1(q)$ über alle q suchen. $\square$

Unter der Annahme, dass man gänzlich mit perfekter Strategie spielt, erhält man für das 6^4 -Mastermind eine untere Schranke von:

$$e(C) = 3,847.$$

Spielt man als erstes einen „echten“ Rateversuch und geht erst ab dem zweiten Zug von der perfekten Strategie aus, so beträgt die untere Schranke:

$$\min_{q \in C_0} \{e_1(q)\} = 4,009.$$

Die starke Abweichung von Koyamas und Lais optimaler Strategie von 8% zeigt deutlich wie unrealistisch die Annahme einer perfekten Strategie ist.

4 Der realistische Ansatz

Nun wollen wir untersuchen, wie sich Situation und Vorgehensweise ändern, wenn der Codierer seinen Code *nicht* gleichwahrscheinlich auswählt. Es stellt sich die Frage, ob und wie stark der Codierer die Gewinnaussichten des Spiels beeinflussen kann. Der Decodierer kann seinerseits auf die taktischen Überlegungen des Codierers reagieren und wie beim worst-case-Ansatz lässt sich ein Minimax-Wert bestimmen. Diesen Minimax-Wert und die dazugehörigen Strategien wollen wir in diesem letzten Kapitel finden.

Bevor wir zum 6^4 -Mastermind kommen, schauen wir uns die Vorgehensweise erneut am einfachen Beispiel des 2^3 -Masterminds an.

4.1 Die Minimax-Strategie beim 2^3 -Mastermind

Erinnern wir uns an das 2^3 -Mastermind mit seinen zwei verschiedenen Eröffnungen AAA und AAB . Wir haben gezeigt, dass die AAB -Eröffnung beim worst-case-Ansatz besser ist. Sie ist auch besser, wenn man vom average-case-Ansatz ausgeht. In diesem Fall kommt der Decodierer in durchschnittlich 2,25 ($= 18/8$) Zügen ans Ziel, wogegen die AAA -Eröffnung 2,625 ($= 21/8$) Züge benötigt (vgl. Tabelle 1 und 2, letzte Spalte, S. 8).

Geht man davon aus, dass sämtliche Codes gleichwahrscheinlich ausgewählt werden, kann man sich bei seinen Untersuchungen auf genau diese beiden Strategien¹⁸ beschränken und beim optimalen Spiel sogar allein auf die AAB -Strategie.

Es leuchtet aber sofort ein, dass man gegen einen taktierenden Codierer nicht immer gleich spielen sollte. Deshalb ist es beim realistischen Ansatz wichtig, die Strategien mit ihren Spiegelbildern zu mixen.

Es reicht jedoch nicht aus, abwechselnd die Eröffnungen der Codeklasse mit

¹⁸Die Begriffe Eröffnung und Strategie sind in diesem Abschnitt gleichbedeutend benutzt, da mit der Wahl der Eröffnung die Strategie festgelegt ist, möchte man im Durchschnitt optimal spielen.

zwei Farben zu spielen. Ebenso wichtig ist es, die *AAA*-Strategie nicht zu verwerfen. Wir werden gleich sehen warum.

Im Folgenden meinen wir, wenn wir von der *AAA*-Strategie sprechen, die Eröffnungen *AAA* und *BBB* und analog alle Eröffnungen mit zwei Farben, wenn wir von der *AAB*-Strategie sprechen.

Schauen wir uns die zwei Eröffnungen noch einmal genauer an. Ist der Geheimcode einfarbig, so knackt die *AAA*-Strategie ihn in 1,5 Zügen. Die erste Antwort ist entweder $[3, 0]$ oder aber $[0, 0]$ und man löst im zweiten Versuch. Die *AAB*-Strategie benötigt für einfarbige Geheimcodes 2,5 Züge. Dafür ist die *AAB*-Strategie bei zweifarbigen Geheimcodes die bessere Wahl - sonst könnte sie auch im average-case-Sinn nicht besser sein.

Tabelle 4 fasst alle vier Fälle zusammen:

$\emptyset$ Züge	Geheimcode einfarbig	Geheimcode zweifarbig
<i>AAA</i> -Strategie	1,5	3
<i>AAB</i> -Strategie	2,5	$2\frac{1}{6}$

Tabelle 4: Die erwartete Zuganzahl je nach Geheimcode und Eröffnung

Daraus lassen sich zwei Schlüsse ziehen:

- Der Codierer sollte mit größerer Wahrscheinlichkeit aus der Menge der einfarbigen Codes wählen als nur mit $1/4$ wie der Zufallsgenerator, um die *AAB*-Strategie unattraktiver zu machen.
- Der Decodierer sollte mal die *AAA*-, mal die *AAB*-Strategie spielen, um auf die Strategie des Codierers zu reagieren.

Kommen wir nun zu der Frage, wie die Minimax-Strategie für beide aussieht. Mit anderen Worten, wie sollten beide spielen, damit der Gegner seine Position durch keine Taktik verbessern kann?

Dazu stellen wir eine Funktion $\emptyset Züge(p(AAA), p(1F))$ auf, welche in Abhängigkeit der Wahrscheinlichkeiten $p(AAA)$ und $p(1F)$ die zu erwartende

Zuganzahl liefert. Dabei ist $p(AAA)$ die Wahrscheinlichkeit, mit welcher der Decodierer die AAA-Strategie wählt, und $p(1F)$ die Wahrscheinlichkeit, mit welcher der Codierer einen einfarbigen Geheimcode steckt.

Die Funktion ist gegeben durch

$$\begin{aligned} \varnothing Z\ddot{u}ge(p(AAA), p(1F)) &= p(AAA) \cdot p(1F) \cdot 1\frac{1}{2} \\ &\quad + p(AAA) \cdot (1 - p(1F)) \cdot 3 \\ &\quad + (1 - p(AAA)) \cdot p(1F) \cdot 2\frac{1}{2} \\ &\quad + (1 - p(AAA)) \cdot (1 - p(1F)) \cdot 2\frac{1}{6}. \end{aligned}$$

Das Ziel des Codierers ist es, dem Decodierer keine Möglichkeit zu lassen, seine Position zu verbessern:

$$\begin{aligned} \varnothing Z\ddot{u}ge(p(AAA), p(1F)) &= p(AAA) \cdot [p(1F) \cdot 1\frac{1}{2} + (1 - p(1F)) \cdot 3] + \\ &\quad (1 - p(AAA)) \cdot [p(1F) \cdot 2\frac{1}{2} + (1 - p(1F)) \cdot 2\frac{1}{6}] \\ &\stackrel{!}{=} \varnothing Z\ddot{u}ge(p(1F)) \end{aligned}$$

Damit der Wert der Funktion $\varnothing Z\ddot{u}ge$ für alle Werte von $p(AAA)$ gleich bleibt und der Decodierer damit keine Handlungsmöglichkeit hat, müssen die eckigen Klammern gleich sein:

$$\begin{aligned} p(1F) \cdot 1\frac{1}{2} + (1 - p(1F)) \cdot 3 &= p(1F) \cdot 2\frac{1}{2} + (1 - p(1F)) \cdot 2\frac{1}{6} \\ \Leftrightarrow \frac{5}{6} \cdot (1 - p(1F)) &= p(1F) \\ \Leftrightarrow \frac{5}{6} &= 1\frac{5}{6}p(1F) \\ \Leftrightarrow \frac{5}{11} &= p(1F). \end{aligned}$$

Somit sollte der Codierer im Verhältnis 5 zu 6 einfarbige und zweifarbige Codes stecken.

Analog ergibt sich als Minimax-Strategie für den Decodierer, dass er im Verhältnis 2 zu 9 zwischen AAA- und AAB-Strategie auswählen sollte, um es seinerseits dem Codierer unmöglich zu machen, die zu erwartende Zuganzahl zu seinen Gunsten zu verbessern.

Setzt man mindestens eine der Wahrscheinlichkeiten in die Ausgangsgleichung ein, erhält man den Minimax-Wert des 2³-Masterminds von 2,318 Zügen.

Wir erinnern uns, dass man bei Gleichverteilung in 2,25 Zügen gewinnen kann. Somit kann der Codierer die Gewinnaussichten um rund 3% zu seinen Gunsten beeinflussen.

4.2 Die Minimax-Strategie beim 6^4 -Mastermind

Im Jahre 1995 lieferte Michael Wiener in einer Newsgroup-Mitteilung den Minimax-Wert von 4,341 ($= 5600/1290$), den er aus mehrmonatiger Computerberechnung erhalten hatte. Ebenfalls aus der Computerberechnung ergaben sich die folgenden Minimax-Strategien:

- Der Codierer sollte seinen Code gleichwahrscheinlich unter den 1290 Codes auswählen, die mindestens 2 Farben enthalten.
- Der Decodierer sollte genau nach Koyama und Lai spielen, so als würde der Codierer seinen Code gleichwahrscheinlich unter allen Codes auswählen.

Die Strategie des Codierers lässt sich damit begründen, dass einfarbige Codes erwartungsgemäß schneller geknackt werden. Das Festhalten des Decodierers an der im average-case-optimalen Strategie von Koyama und Lai folgt daraus, dass die Strategie des Codierers die Situation nur minimal verändert.

Den Minimax-Wert kann man leicht ausrechnen, wenn man weiß, dass einfarbige Codes bei Koyama und Lai in durchschnittlich $25/6$ Zügen geknackt werden. Zur Erinnerung: Für sämtliche Codes benötigt man im Durchschnitt $5625/1296$ Züge.

Damit folgt für alle Codes mit mindestens zwei Farben die zu erwartende Zuganzahl x :

$$\frac{6}{1296} \cdot \frac{25}{6} + \frac{1290}{1296} \cdot x = \frac{5625}{1296} \Rightarrow x = \frac{5600}{1290}.$$

Aus Sicht des Codierers ist die Minimax-Strategie um gerade mal 0,02% besser gegenüber der zufälligen Codewahl, also gegenüber gar keiner Strategie.

5 Quellenverzeichnis

- BEWERSDORFF, J. (2001): *Glück, Logik und Bluff – Mathematik im Spiel: Methoden, Ergebnisse und Grenzen*. Braunschweig/Wiesbaden, 2. Auflage.
- IRVING, R. (1978/79): *Towards an Optimum Mastermind Strategy*. Journal of Recreational Mathematics, Ausg. 1, S. 81-87.
- KNUTH, D. (1976/77): *The Computer as Master Mind*. Journal of Recreational Mathematics, Ausg. 9, S. 1-6.
- KOOI, B. (2005): *Yet Another Mastermind Strategy*. ICGA Journal, Ausg. 28(1), S. 13-20.
- KOYAMA, K. UND LAI, T. (1993): *An Optimal Mastermind Strategy*. Journal of Recreational Mathematics, Ausg. 25, Nr. 4, S. 251-256.
- NEUWIRTH, E. (1982): *Some Strategies for Mastermind*. Zeitschrift für Operations Research, Ausg. 26, S. B257-B278.
- VIAUD, D. (1979): *Une formalisation du jeu de Mastermind*. R.A.I.R.O. Recherche opérationnelle/Operations Research, Ausg. 13, S. 307-321.

Selbstständigkeitserklärung

Hiermit erkläre ich, Michael Kußmann, dass ich die vorliegende Bachelorarbeit mit dem Titel

Mastermind

selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Essen, den _____
