Optimierung des Decodierungsalgorithmus

Dana Woitas, Daniel Heinrich

19. Juli 2007

1 Wiederholung

Die Mathieu–Gruppe ist

$$M_{12} := \langle (1\,2), (3\,4), (5\,6), (7\,8), (9\,10), (11\,12), (1\,3\,2), (4\,7\,5), (8\,9\,11) \rangle .$$

M_{12} ist scharf 5-transitiv und 3-fehlerkorrigierend, ein $(1, 2, 3)$-Uncovering ist

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

2 Optimierung

Satz 2.1. *Wenn die Position von Fehlern beim empfangenen Wort bekannt ist, ist es ausreichend, kleinere Uncoverings zu benutzen.*

Beispiel: Es wird 1 2 3 4 5 6 7 8 9 10 11 12 gesendet

1) Empfangen: 1 1 4 3 5 6 7 8 9 10 11 12
   Ein (10,5,2)-Uncovering ist ausreichend.

2) Empfangen: 1 1 3 3 2 6 7 8 9 10 11 12
Ein (8,5,1)-Uncovering ist ausreichend.

3) Empfangen: 1 1 3 3 5 5 7 8 9 10 11 12
Sehe in der Liste der gesendeten Möglichkeiten nach

4) Empfangen: 1 1 1 1 5 6 7 8 9 10 11 12
Sehe in der Liste der gesendeten Möglichkeiten nach

5) Empfangen: 1 1 1 4 4 6 7 8 9 10 11 12
wie 3) und 4)

6) Empfangen: 1 1 1 4 4 6 7 8 9 10 11 12
Ein (9,5,1)-Uncovering ist ausreichend.

7) Empfangen: 1 1 3 3 5 6 7 8 9 10 11 12
Ein (8,5,1)-Uncovering ist ausreichend.

8) Empfangen: 1 1 1 4 5 6 7 8 9 10 11 12
Ein (9,5,1)-Uncovering ist ausreichend.

9) Empfangen: 1 1 3 4 5 6 7 8 9 10 11 12
Ein (10,5,2)-Uncovering ist ausreichen.

Ein (10,5,2)-Uncovering:

1	2	3	4	5
1	2	7	8	10
1	5	6	7	9
2	3	6	8	9
3	4	7	9	10
4	5	6	8	10

Ein (9,5,1)-Uncovering:

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

Ein (8,5,1)-Uncovering:

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

2.2 Verbesserung des Algorithmus durch Änderung der Reihenfolge des Uncoverings ...

2.2.1 ... durch Basisumordnung

Definition. $(B_1, B_2, ..., B_n)$ *seien Basen des (n,k,r)-Uncovering*
$a_j :=$ *durchschnittliche Anzahl der Basen, welche durchlaufen werden müssen, bis der Algorithmus zum Erfolg führt, wobei j die Anzahl der Fehler ist, die korrigiert werden sollen.* $j \leq r$

$a_j = \frac{\sum_{i=0}^{n} i s_{ij}}{\binom{n}{j}}$ *mit*
$s_{ij} :=$ *Anzahl der j-TM, welche sich nicht mit* B_i *schneiden, aber mit allen allen* B_{i-p} *mit* $1 \leq p \leq i$
$s_j = (s_{1j}, s_{2j}, ..., s_{nj})$

Algorithmus zur Umordnung der Basen
$(B_1, ..., B_n)$ Basen mit zugehöriger Folge $s_k 1 \leq k \leq n$
Es gelte $s_1 \geq s_2 \geq ... \geq s_\ell < s_{\ell+1}$
Dann erhält man die neue Basenreihenforlge $(B'_1, ..., B'_n)$ wie folgt:

$$\begin{array}{ccc} B'_1 = B_1 & B'_\ell = B_{\ell+1} & B'_{\ell+2} = B_{\ell+2} \\ B'_2 = B_2 & B'_{\ell+1} = B_\ell & B'_{\ell+3} = B_{\ell+3} \\ \vdots & & \vdots \\ B'_{\ell-1} = B_{\ell-1} & & B'_n = B_n \end{array}$$

Für die zugehörige Folge s'_k gilt

- $$\begin{array}{cc} s'_1 = s_1 & s'_{\ell+2} = s_{\ell+2} \\ s'_2 = s_2 & s'_{\ell+3} = s_{\ell+3} \\ \vdots & \vdots \\ s'_{\ell-1} = s_{\ell-1} & s'_n = s_n \end{array}$$
- $s'_\ell \geq s_{\ell+1} und s'_{\ell+1} \leq s_\ell$
- $s'_\ell + s'_{\ell+1} = s_\ell + s_{\ell+1}$

Beispiel: M_{12}

$s_3 = (35, 31, 30, 23, 17, 21, 20, 15, 17, 8, 3)$
$a_3 = \frac{1016}{220}$

(12,5,3)-Uncovering bzgl $(B'_1, ..., B'_n)$:

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

$s'_3 = (35, 31, 30, 23, 23, 15, 20, 15, 17, 8, 3)$
$a'_3 = \frac{1010}{220}$

Nach mehrfachem Anwenden des Algortihmus erhält man folgendes (12,5,3)-Uncovering:

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

und $s_3 = (35, 31, 30, 23, 23, 23, 20, 13, 11, 8, 3)$
$a_3 = \frac{988}{220}$. Bestmögliche Anordnung bietet ein $a_3 = \frac{959}{220}$

2.2.2 ... durch eingebettete Uncoverings

Wir wissen bereits, daß jedes (n, k, r)-Uncovering auch ein $(n, k, r-1)$-Uncovering ist, u. s. w. Daher wird nun versucht, durch eingebettete Uncoverings kleinerer Größe den Algorithmus für ein gegebenes (n, k, r)-Uncovering zu verbessern.

Bei der sogenannten „top-down"-Strategie sortiert man die Zeilen des gegebenen (n, k, r)-Uncoverings U folgendermaßen:
Finde zunächst das (nicht unbedingt eindeutig) kleinste $(n, k, r-1)$-Uncovering U' innerhalb von U. Sortiere nun die Zeilen von U' an den Anfang der Liste. Suche nun innerhalb von U' das (nicht unbedingt eindeutig) kleinste $(n, k, r-2)$-Uncovering U'' und sortiere die Zeilen von U' so, daß die Zeilen von U'' den Anfang bilden. Iteriere dieses Verfahren, bis die ersten Zeilen der Liste ein $(n, k, 1)$-Uncovering bilden.

Die „bottom-up"-Strategie funktioniert ähnlich. Man startet das untere Ende der Liste mit einem bestmöglichen $(n, k, 1)$-Uncovering und fügt nun darüber einige k-Tupel ein, so daß man ein $(n, k, 2)$-Uncovering erhält. So fährt man fort, bis man ein (n, k, r)-Uncovering gefunden hat, welches natürlich viel größer als notwendig geworden sein kann.

Beispiel: Betrachte die Gruppe M_{12} mit dem oben angegebenen $(12, 5, 3)$-Uncovering. Dann führt die Umordnung mit der „top-down"-Strategie zu folgendem Uncovering:

$$
\begin{array}{rl}
(12,5,1)\text{-Uncovering} & \left[\begin{array}{rrrrr} 1 & 4 & 6 & 7 & 10 \\ 2 & 4 & 8 & 9 & 12 \\ 3 & 5 & 6 & 10 & 12 \end{array}\right. \\
\text{Rest des } (12,5,2)\text{-Uncoverings} & \left[\begin{array}{rrrrr} 1 & 5 & 8 & 9 & 11 \\ 3 & 4 & 7 & 11 & 12 \\ 3 & 6 & 8 & 9 & 11 \end{array}\right. \\
\text{Rest des } (12,5,3)\text{-Uncoverings} & \left[\begin{array}{rrrrr} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 6 & 11 & 12 \\ 1 & 3 & 7 & 8 & 9 \\ 2 & 5 & 7 & 10 & 11 \\ 6 & 7 & 8 & 9 & 10 \end{array}\right.
\end{array}
$$

3 Erwartete Anzahl der Suchschritte des Algorithmus

Um nun herauszufinden, welche Umsortierung eines (n, k, r)-Uncoverings man als Grundlage zur Decodierung verwenden sollte, betrachten wir zunächst einmal die zu erwartende Anzahl der Suchvorgänge, die der Algorithmus bei der Decodierung durchführt.
Wir nehmen an, daß Fehler an den einzelnen Positionen im gesendeten Wort unabhänging und identisch verteilt mit fester Wahrscheinlichkeit p auftreten. Dann ist die Wahrscheinlichkeit des Ereignisses A, daß $j \leq r$ Fehler auftreten,

$$P(A) = \sum_{j \leq r} \binom{n}{j} p^j (1-p)^{n-j} \ .$$

Sei nun X die Zufallsvariable, die die Anzahl der Suchvorgänge des Algorithmus beschreibt. Dann suchen wir also $E(X|A)$.

Theorem. *Die erwartete Anzahl von Suchvorgängen bei einem Maximum von r Fehlern ist gegeben durch*

$$E(X|A) = \frac{\sum_{j \le r} T_j p^j (1-p)^{n-j}}{\sum_{j \le r} \binom{n}{j} p^j (1-p)^{n-j}},$$

wobei $T_j = \sum_i i s_{ij}$ *die Gesamtzahl der Suchvorgänge über alle Kombinationen mit j Fehlern ist.*

Somit kann man nun für jede Umordnung eines Uncoverings die Funktion

$$F(p) := E(X|A)$$

betrachten. Wir nennen nun für Umordnungen $U_1, \ldots, U_k$ des (n,k,r)-Uncoverings U diejenige optimal, für die

$$\forall p \in [0,1] : F_{U_i}(p) \le F_{U_j}(p)$$

für alle betrachteten $i = 1, \ldots, k$; $i \neq j$ gilt.

Beispiel: Betrachten wir nocheinmal die M_{12} mit unserem $(12,5,3)$-Uncovering. Dann ist $n = 12$, $r = 3$ und $T_0 = 1$, $T_1 = 22$, $T_2 = 189$, $T_3 = 1016$. Somit gilt für den Erwartungswert $E(X|A)$:

$$F(p) = \frac{(1-p)^3 + 22p(1-p)^2 + 189p^2(1-p) + 1016p^3}{(1-p)^3 + 12p(1-p)^2 + 66p^2(1-p) + 220p^3}.$$

Wir wollen nun sehen, wie die verschiedenen Umordnungen die Funktion $F(p)$ beeinflussen. Setze dazu

$$\begin{aligned} U_L &:= \text{„lexikographische Ordnung“} \\ U_B &:= \text{„Umordnung der Basen“} \\ U_E &:= \text{„Eingebettete Uncoverings“} \\ U_{T_3} &:= \text{„Eine Ordnung, für die } T_3 \text{ minimal ist.“} \end{aligned}$$

Dann ist für jedes $p \in [0,1]$

$$F_{U_{T_3}}(p) \le F_{U_E}(p) \le F_{U_B}(p) \le F_{U_L}(p).$$

Betrachten wir nun noch eine zufällige Umordnung U_Z des lexikographisch geordneten Uncoverings, so führt die „Umordnung der Basen“ zu folgender Situation: Für kleine p ist $F_{U_Z}(p) \le F_{U_L}(p)$, jedoch kehrt sich das Verhältnis für größere p ($p \ge 0,7$) um.

Allerdings arbeitet der Algorithmus nach Anwendung des Algorithmus zur Umordnung der Basen mit der verbesserten zufälligen Ordnung für alle $p \in [0,1]$ schneller.