Industrielle Anwendungen der Computeralgebra

1. Statistik: "Design of Experiments"

In dem Teilbereich "Design of Experiments" der Statistik geht es um folgendes Problem. Zur vollständigen Beantwortung einer Frage ist es notwendig, eine sehr große Zahl von Datenpunkten ("full factorial design") zu kennen. Dies ist aber in der Praxis entweder unmöglich oder zu teuer. Darum soll eine Teilmenge so bestimmt werden, daß aus der Kenntnis der Teildaten die gesuchten Erkenntnisse über die Gesamtmenge erschlossen werden können.

Erstes Beispiel: (Marketing)
Ein neues Handy soll am Markt eingeführt werden. Für die von den Kunden gewünschten Eigenschaften wird ein mathematisches Modell aufgestellt.

Eigenschaft	*Werte*	*Modell*
F*arbe*	*blau, gelb, rot*	$0,1,2$
G*ewicht*	100 *Gramm Schritte*	$0,1,2,3,4$
K*anten*	*eckig, rund, abgeflacht*	$0,1,2$
P*reis*	5 *Preisbereiche*	$0,1,2,3,4$

Die Menge aller möglichen Tupel $(\mathbf{F},\mathbf{G},\mathbf{K},\mathbf{P}) \in \mathbb{Q}^4$ bildet das "full design" $\mathbb{D}$. Die (gesuchten) Kundenpräferenzen entsprechen einer (unbekannten) Funktion $f : \mathbb{D} \longrightarrow \mathbb{Q}$, die man das **Modell** nennt.

Bemerkung: Die Funktion $f : \mathbb{D} \longrightarrow \mathbb{Q}$ ist auf einer endlichen Menge definiert. Dann gibt es ein Polynom in $\mathbb{Q}[x_1, x_2, x_3, x_4]$, das auf $\mathbb{D}$ die gleichen Werte annimmt wie f.

Problem: Wir können bei einer Kundenumfrage nicht verlangen, daß die Testpersonen eine Rangfolge für alle 225 möglichen Kombinationen der Eigenschaften aufstellen.

Lösungsansatz:
• Finde eine geeignete "kleine" Teilmenge $\mathbb{F} \subset \mathbb{D}$ ("fraction").
• Erfrage die Bewertung der Elemente von $\mathbb{F}$ durch die Testpersonen.
• Versuche die Funktion $f : \mathbb{D} \longrightarrow \mathbb{Q}$ aus diesen Daten zu rekonstruieren ("identify the model").

Mathematische Durchführung:
Sie $I(\mathbb{D})$ die Menge aller Polynome in $P = \mathbb{Q}[x_1, x_2, x_3, x_4]$, die auf $\mathbb{D}$ verschwinden. Dies ist ein Ideal in P. Es wird erzeugt von den Polynomen $\{x_1(x_1 - 1)(x_1 - 2), x_2(x_2 - 1) \cdots (x_2 - 4), x_3(x_3 - 1)(x_3 - 2), x_4(x_4 - 1) \cdots (x_4 - 4)\}$. Wir müssen das Ideal $I(\mathbb{F})$ aller Polynome, die auf $\mathbb{F}$ verschwinden, berechnen und die Menge der "confounding polynomials" $I(\mathbb{F}) \setminus I(\mathbb{D})$.

Wenn wir dann etwas über die Monome wissen, die in dem gesuchten Polynom f wirklich vorkommen (z.B. "der Einfluß von Farbe und Preis auf f ist unkorrelliert"), dann können wir versuchen $\mathbb{F}$ so zu wählen, dass $f|_{\mathbb{F}}$ bereits das Polynom $f \in P$ bestimmt. Allerdings können wir f stets um Elemente aus $I(\mathbb{D})$ abändern, ohne seine Werte auf $\mathbb{D}$ zu beeinflussen.

Zweites Beispiel: (Umzug einer Chemiefabrik)
Eine Chemiefabrik zog in ein neues Gebäude in einer anderen Gegend um. Plötzlich funktionierte einer der Filtrierschritte nicht mehr. Für die möglichen Ursachen gab es eine Reihe von Kandidaten:

- Die Lufttemperatur in der neuen Fabrikhalle war anders.
- Das Wasser in der neuen Gegend war anders.
- Eine der Maschinen hatte beim Transport einen Schaden erlitten.
- Die verwendeten Chemikalien hatten eine etwas andere Zusammensetzung (neue Zulieferer).
- Die Prozedur der Vorbehandlung hatte sich geändert (neue Angestellte).

Es war zu teuer, alle Kombinationen aller Werte dieser Faktoren in Testläufen durchzuspielen. Daher mußten einige wenige Testläufe so konfiguriert werden, daß sich aus ihnen der entscheidende Faktor ablesen ließ.

Wieder ist ein wichtiger Schritt in der Lösung die Berechnung des Ideals $I(\mathbb{F}) \subset P$ aller Polynome in $\mathbb{Q}[x_1, \ldots, x_n]$ die auf der (geplanten) Teilmenge $\mathbb{F} \subset \mathbb{D}$ verschwinden. Sei $\mathbb{F} = \{P_1, \ldots, P_s\}$ mit $P_i = (p_{i1}, \ldots, p_{in}) \in \mathbb{Q}^n$. Das Ideal aller Polynome, die in P_i verschwinden, heiße $I(P_i)$. Zu berechnen ist

$$I(\mathbb{F}) = \bigcap_{i=1}^{s} I(P_i) = \bigcap_{i=1}^{s} (x_1 - p_{i1}, \ldots, x_n - p_{in}).$$

Der effektivste Algorithmus in dieser Situation ist der Buchberger-Möller-Algorithmus.

Der Buchberger-Möller Algorithmus

Sei K ein Körper, σ eine Termordnung auf den Termen von $K[x_1, \ldots, x_n]$, und seien $P_1, \ldots, P_s \in K^n$. Betrachte die folgenden Instruktionen:

1) Starte mit Listen $G = [\,]$, $Q = [\,]$, $S = [\,]$, $L = [1]$ und einer Matrix $M = (m_{ij})$ über K mit s Spalten und anfangs 0 Zeilen.

2) Ist $L = [\,]$ so gib $[G, Q]$ aus und stoppe. Andernfalls wähle $t = \min_\sigma(L)$ und streiche es aus L.

3) Berechne $(t(P_1), \ldots, t(P_s)) \in K^s$. Reduziere diesen Vektor gegen die Zeilen von M und erhalte

$$(v_1, \ldots, v_s) = (t(P_1), \ldots, t(P_s)) - \sum_i a_i\,(m_{i1}, \ldots, m_{is})$$

mit $a_i \in K$.

4) Ist $(v_1, \ldots, v_s) = 0$, so füge das Polynom $t - \sum_i a_i s_i$ zu G hinzu, wobei s_i das i-te Element von S ist. Streiche aus L alle Vielfachen von t und fahre mit 2) fort.

5) Ist $(v_1, \ldots, v_s) \neq 0$, so füge $(v_1, \ldots, v_s)$ als neue Zeile zu M hinzu, $t - \sum_i a_i s_i$ als neues Element zu S, t zur Liste Q, und zu L alle Terme in $\{x_1 t, \ldots, x_n t\}$, die weder Vielfaches eines Elements von L noch von $\{\mathrm{LT}_\sigma(g) \mid g \in G\}$ sind. Fahre dann mit 2) fort.

Dies ist ein Algorithmus, der die reduzierte σ-Gröbner Basis G des Ideals $I = I(P_1) \cap \cdots \cap I(P_s)$ berechnet sowie die Liste Q derjenigen Terme, die nicht in $\mathrm{LT}_\sigma(I)$ liegen.

Über $K = \mathbb{Q}$ stellt sich das Problem des Wachstums der Koeffizienten der Polynome. Man braucht eine *modulare Version* des Algorithmus.

Beispiel: Gegeben seien 100 Punkte in $\mathbb{Q}^{10}$ mit Koordinaten in $\{0,1,2,\ldots,9\}^{10}$. Wir suchen ein Erzeugendensystem für das Verschwindungsideal dieser Punkte.

Das minimale Erzeugendensystem hat 286 Polynome mit je ca. 100 Termen. Jeder Koeffizient hat ca. 160 stellige Zähler und Nenner. Das Erzeugendensystem füllt ca. 4000 Bildschirmseiten.

Die modulare Version des BM-Algorithmus

In obiger Situation betrachte folgende Instruktionen:

1) Setze $Q = [\,]$, $G = [\,]$, $m = 1$ und $G_m = [\,]$.

2) Wähle eine neue Primzahl p, so daß die Reduktionen $\overline{P}_1, \ldots, \overline{P}_s$ paarweise verschieden sind.

3) Wende den BM-Algorithmus mit $K = \mathbb{F}_p$ auf die Punktmenge $\{\overline{P}_1, \ldots, \overline{P}_s\}$ an und erhalte $[G_p, Q_p]$.

4) Vergleiche $Q_p = [q_1', q_2', \ldots]$ mit $Q = [q_1, q_2, \ldots]$ wie folgt: Q ist *besser* als Q_p wenn es ein j gibt mit $q_1 = q_1', \ldots,$ $q_{j-1} = q_{j-1}'$, $q_j <_\sigma q_j'$. Ist Q besser als Q_p so fahre mit 2) fort. Ist Q_p besser als Q, so ersetze Q durch Q_p, m durch p, G_m durch G_p und fahre mit 2) fort.

5) Jetzt gilt $Q = Q_p$. Ersetze m durch mp und kombiniere G_m mit G_p mit Hilfe des *Chinesischen Restsatz Algorithmus*. Nenne das Ergebnis G'.

6) Ist $G' \neq G$, so ersetze G durch G'. Fahre mit 2) fort.

7) Nun gilt $G' = G$. Prüfe, ob alle Polynome aus G an den Punkten $P_1, \ldots, P_s$ verschwinden. Wenn nicht, fahre mit 2) fort. Wenn ja, gib $[G, Q]$ aus und stoppe.

Dies ist ein Algorithmus, der die reduzierte σ-Gröbner Basis G von $I = I(P_1) \cap \cdots \cap I(P_s)$ und die Liste Q der Terme außerhalb $\mathrm{LT}_\sigma(I)$ berechnet.

Weitere Fragen:

1) Sei eine Teilmenge $\mathbb{F} \subset \mathbb{D}$ vorgegeben. Welche Modelle $f : \mathbb{F} \longrightarrow \mathbb{D}$ können mit Hilfe von $f|_{\mathbb{F}}$ identifiziert werden?

2) Die Koeffizienten von $\{1, x_1, \ldots, x_n\}$ in f heißen die **Haupteffekte**, diejenigen von $\{x_i x_j \mid i, j = 1, \ldots, n\}$ heißen die **Interaktionen**. Welche Teilmengen $\mathbb{F}$ von $\mathbb{D}$ sind zu wählen, wenn man nur die Haupteffekte identifizieren will?

3) Welche Teilmengen $\mathbb{F}$ von s Punkten aus $\mathbb{D}$ identifizieren die größte Anzahl linearer polynomialer Modelle?

Literatur:

1) L. Robbiano, Gröbner bases and statistics, in: B. Buchberger und F. Winkler (Hrsg.), Gröbner Bases and Applications, Proc. Conf. Linz 1998, London Math. Soc. Lect. Notes **251**, Cambridge University Press, S. 179–204

2) J. Abbott, A. Bigatti, M. Kreuzer und L. Robbiano, Computing ideals of points, J. Symb. Comput. **30** (2000), S. 341–356

3) J. Abbott, M. Kreuzer und L. Robbiano, Computing zero-dimensional schemes, Preprint 2000

2. Computeralgebra in der Warteschlangentheorie

in Zusammenarbeit mit

Dr. H. Gold (Infineon Technologies)

Bei der Ablaufplanung in der Chipfertigung spielen Warteschlangen mit Kontrolle eine große Rolle. Hat man eine Anzahl von verschiedenen Maschinen (z.B. Stepper, Plasmaätzer, etc.) und Lots (d.h. Fertigungsgruppen) unterschiedlicher Priorität und Größe, so ist die Durchlaufzeit eines einzelnen Lots durch die Maschinengruppe annähernd Poisson-verteilt. Ferner nehmen wir an, daß die Rohmaterialanlieferung und die Chipnachfrage durch Poisson-Prozesse gegeben sind. Wir erhalten folgende Situation.

$$
\begin{tikzcd}
 & & & {} \arrow[d, "\lambda_2"] \\
 & & & \boxed{\text{Schlange } Q_2} \arrow[d] \\
{} \arrow[r, "\lambda_1"] & \boxed{\text{Schlange } Q_1} \arrow[r, "\mu"] & \boxed{\text{Schlange } Q_3} \arrow[r] & \text{Output}
\end{tikzcd}
$$

Hierbei ist $\xrightarrow{\lambda_1}$ der Anlieferungsprozeß,
$\xrightarrow{\mu}$ der Fertigungsprozeß, $\xrightarrow{\lambda_2}$ der Chipnachfrageprozeß,
Schlange Q_1 enthält die noch nicht abgearbeiteten Lots,
Schlange Q_2 enthält die noch nicht befriedigte Nachfrage,
und Schlange Q_3 ist das Lager.

Alle drei Warteschlangen sollen eine endliche Kapazität haben. Wir nehmen außerdem an, daß Q_2 oder Q_3 stets leer ist, d.h. daß alle Aufträge sofort erledigt werden, solange Chips im Lager sind.

Für $i = 1, 2, 3$ sei s_i die Kapazität der Warteschlange Q_i. Wir können voraussetzen, daß das System so dimensioniert ist, daß $s_1 = s_2 + s_3$ gilt, und diese Zahl nennen wir ab jetzt s. Wir setzen

$$i = \begin{cases} 1 + \text{ Jobs in } Q_1 & \text{falls gerade gefertigt wird,} \\ 0 & \text{sonst, und} \end{cases}$$

$$j = \begin{cases} s_3 + \text{ Jobs in } Q_2 & \text{falls } Q_2 \text{ nichtleer ist,} \\ \text{freie Plätze in } Q_3 & \text{sonst} \end{cases}$$

Dann erfüllen die Wahrscheinlichkeiten p_{ij} dafür, daß sich das System im Zustand (i, j) befindet, die folgenden Rekursionsgleichungen:

$$\begin{aligned} p_{i+1\, j+1}\, \chi_{i<s,\, j<s}\, \mu = {} & \\ & p_{ij}[\chi_{i<s}\lambda_1 + \chi_{j<s}\lambda_2 + \chi_{i>0,\, j>0}\mu] \\ & - p_{i-1\, j}\, \chi_{i>0}\, \lambda_1 \ - \ p_{i\, j-1}\, \chi_{j>0}\, \lambda_2 \end{aligned}$$

1) Für spezielle Werte von λ_1, λ_2 und μ kann man diese Rekursionsgleichungen mit Hilfe eines expliziten Algorithmus lösen, vgl.

H. Gold, A Markovian single server with upstream job and downstream demand arrival system, Queuing Systems **30** (1998), 435 – 455

2) Eigentlich möchte man die Lösungen in Abhängigkeit von λ_1, λ_2 und μ studieren.

Dazu betrachten wir die Rekursionsgleichungen als ein algebraisches Gleichungssystem über dem Funktionenkörper $\mathbb{Q}(\lambda_1, \lambda_2, \mu)$. Wir versuchen es mit Mitteln der Computeralgebra symbolisch zu lösen. Im Fall $s = 3$ ergibt sich nach der Substitution $x_{ij} := p_{ij}$, $y := \frac{\lambda_1}{\mu}$, $z := \frac{\lambda_2}{\mu}$ zum Beispiel

(1) $x_{00}(y+z) - x_{11} = 0$

(2) $x_{01}(y+z) - x_{12} - x_{00}z = 0$

(3) $x_{02}(y+z) - x_{13} - x_{01}z = 0$

(4) $x_{03}y - x_{02}z = 0$

(5) $x_{10}(y+z) - x_{21} - x_{00}y = 0$

(6) $x_{11}(y+z+1) - x_{22} - x_{01}y - x_{10}z = 0$

(7) $x_{12}(y+z+1) - x_{23} - x_{02}y - x_{11}z = 0$

(8) $x_{13}(y+1) - x_{03}y - x_{12}z = 0$

(9) $x_{20}(y+z) - x_{31} - x_{10}y = 0$

(10) $x_{21}(y+z+1) - x_{32} - x_{11}y - x_{20}z = 0$

(11) $x_{22}(y+z+1) - x_{33} - x_{12}y - x_{21}z = 0$

(12) $x_{23}(y+1) - x_{13}y - x_{22}z = 0$

(13) $x_{30}z - x_{20}y = 0$

(14) $x_{31}(z+1) - x_{21}y - x_{30}z = 0$

(15) $x_{32}(z+1) - x_{22}y - x_{31}z = 0$

(16) $x_{33} - x_{23}y - x_{32}z = 0$

Man hat also ein lineares Gleichungssystem von $(s+1)^2$ Gleichungen in $(s+1)^2$ Unbestimmten x_{ij} über dem Körper $\mathbb{Q}(y,z)$.

Man kann die Koeffizientenmatrix M_s aufstellen und die Cramersche Regel anwenden. Es gibt einen 1-dimensionalen Lösungsraum, in dem die korrekte Lösung die Normierungsgleichung

$$\sum_{i,j} x_{ij} = 1$$

erfüllen muß.

Der Lösungsvektorraum von M_s wird also von einem Tupel von Polynomen $(f_{00}, \ldots, f_{ss})$ aufgespannt, die bis auf einen Normierungsfaktor eindeutig bestimmt sind. Folglich können wir annehmen, daß sie keinen gemeinsamen Faktor haben.

Die folgenden **Fragen** sind für die Anwendungen interessant:

1) Welche Grade haben die Polynome f_{ij} ?

2) Was ist ihr Leitterm (bzgl. der gradkompatiblen Termordnung σ mit $y > z$) ?

3) Bis zu welchem s kann man die f_{ij} effektiv berechnen? (Praktisch relevant ist $s \leq 25$.)

Einige Antworten: Um die Matrix M_s zu lösen, führt man zuerst einige Vereinfachungsschritte durch.

1) Wende die Substitution $v = \frac{z}{y}$, $w = \frac{1}{y}$ an und beseitige alle Nenner. Aus der $(-y)$-Diagonalen wird dadurch eine (-1)-Diagonale.

2) Verwende die (-1)-Diagonale, um die Matrix M_s auszuräumen. Erhalte in der rechten unteren Ecke eine $(s+1) \times (s+1)$-Matrix N_s, die jedoch jetzt dicht mit Polynomen höheren Grades besetzt ist.

Algorithmus 1: Wende eine geeignete Version des Bareiss-Algorithmus oder des Malashonok-Algorithmus auf N_s an.

Satz 2.1. *a) Die Matrix N_s hat den linearen Anteil*

$$N_s^{\text{lin}} = \begin{pmatrix} -v & w & 0 & \cdots & \cdots & 0 \\ v & -v-w & w & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & -v-w & w \\ 0 & \cdots & \cdots & 0 & v & -w \end{pmatrix}$$

b) Die Matrix N_s^{lin} hat den (bis auf einen Faktor eindeutigen) Lösungsvektor $(w^s, vw^{s-1}, \ldots, v^s)$.

c) Die Matrix N_s hat einen (bis auf einen Faktor eindeutigen) Lösungsvektor $(g_0, \ldots, g_s)$ mit

$$g_i = v^i w^{s-i} + (\textit{Terme höheren Grades})$$

Korollar 2.2. *Für die Leitterme der Polynome* $f_{s0}, \ldots, f_{ss}$ *gilt*

$$\mathrm{LT}_\sigma(f_{si}) = y^{s^2+s} z^i \qquad \text{für } i = 0, \ldots, s.$$

Insbesondere gilt $\deg(f_{si}) = s^2 + s + i$ *für* $i = 0, \ldots, s$.

Vermutung: Der Leitterm von f_{0i} ist $y^{s^2-i} z^{s+i}$ für $i = 0, \ldots, s$. Ist die Vermutung richtig, so gilt

$$\deg(f_{ij}) = s^2 + s + \min\{i, j\} \qquad \text{für } i, j = 0, \ldots, s.$$

Satz 2.3. *Die Polynome des Lösungsvektors* $(g_0, \ldots, g_s)$ *von* N_s *sind von der Form* $g_i = v^i w^{s-i} h_i$ *mit Polynomen* h_i *vom Grad* $\deg(h_i) = s^2 - 2s + i$ *und* $h_i(0,0) \neq 0$ *für* $i = 0, \ldots, s$.

Man kann eine weitere Matrix L_s *aufstellen, deren Lösungsvektor gerade* $(h_0, \ldots, h_s)$ *ist.*

Algorithmus 2: Man kann die Matrix L_s wie folgt effizient lösen:

1) Für verschiedene Primzahlen p betrachte die *Reduktion* $L_s(\mathrm{mod}\, p)$ der Matrix L_s modulo p, d.h. das Bild von L_s in $\mathbb{Z}/(p)[v, w]$.

2) Löse $L_s(\mathrm{mod}\, p)$ mit Hilfe des Malashonok-Verfahrens.

3) Rekombiniere die modularen Lösungen mit Hilfe des Chinesichen Restsatzes.

4) Ist das Produkt der verwendeten Primzahlen groß genug, erhält man die korrekte Lösung über $\mathbb{Q}$. Dies kann man durch Einsetzen leicht bestätigen.

In der Praxis erhalten wir folgende **Ergebnisse:**

1) Cramer's Regel funktioniert für $s \leq 4$.

2) Das Malashonok-Verfahren löst N_s für $s \leq 8$.

3) Algorithmus 2 löst L_s für $s \leq 12$.

Die Größe der Lösung von N_s für $s = 7$ (d.h. die 8 Polynome $g_0, \ldots, g_7$) ist ca. 400 kB.

Literatur:

1) H. Gold, A Markovian single server with upstream job and downstream demand arrival system, Queuing Systems **30** (1998), 435 – 455

2) H. Gold, Performance evaluation of and decision making tools for semiconductor wafer probing facilities, in: Proc. Conf. MASM 2000 (erscheint)

3) B. Krammer, Effektive lineare Algebra für Polynommatrizen, Diplomarbeit, Universität Regensburg 2000

4) G.I. Malashonok, Solution methods for linear equation systems in a commutative ring, Proc. Conf. MEGA '90 (Castiglioncello 1990), Progress in Math. **94**, Birkhäuser, Basel 1992

5) A. Wills, Algebraische Probleme in der Warteschlangentheorie, Diplomarbeit, Universität Regensburg (in Vorbereitung)

3. Computeralgebra in der Codierungstheorie

Sei p eine Primzahl, sei $e > 0$, sei $q = p^e$, und sei $\mathbb{F}_q$ der Körper mit q Elementen. Ein *linearer Code* C ist ein $\mathbb{F}_q$-Untervektorraum von $\mathbb{F}_q^s$. Dabei heißt s die *Länge* von c, die Zahl $d = \dim_{\mathbb{F}_q}(C)$ heißt die *Dimension* von C, und

$$m(C) = \min\{\#\{i \mid c_i \neq 0\} \mid (c_1, \ldots, c_s) \in C \setminus \{0\}\}$$

heißt der *Minimalabstand* von C.

Ein guter linearer Code hat einen großen Minimalabstand bei kleiner Länge und großer Dimension. Es gilt die

Singleton-Schranke: $m \leq s - d + 1$

Ziele: 1) Konstruiere gute lineare Codes, also Codes, die der Singleton-Schranke nahe kommen.

2) Ist ein linearer Code $C \subseteq \mathbb{F}_q^s$ gegeben, so berechne seinen Minimalabstand.

Hanson's Konstruktion: Gegeben sei eine endliche Punktmenge $\mathbb{X} = \{P_1, \ldots, P_s\}$ in $\mathbb{P}^n(\mathbb{F}_q)$. Ihr homogenes Verschwindungsideal in $\mathbb{F}_q[x_0, \ldots, x_n]$ werde mit $I_{\mathbb{X}}$ bezeichnet. Ferner sei $R = \mathbb{F}_q[x_0, \ldots, x_n]/I_{\mathbb{X}}$ der homogene Koordinatenring von $\mathbb{X}$ und $H_{\mathbb{X}} : \mathbb{N} \longrightarrow \mathbb{N}$ $(i \mapsto \dim_{\mathbb{F}_q}(R_i))$ die Hilbert-Funktion von $\mathbb{X}$.

Bekanntlich steigt die Funktion $H_{\mathbb{X}}$ streng monoton an, bis sie die Punktezahl s erreicht. Dann bleibt sie konstant gleich s. Wir setzen $\sigma_{\mathbb{X}} = \max\{i \in \mathbb{N} \mid H_{\mathbb{X}}(i) < s\}$. Im folgenden nehmen wir an, daß die erste Koordinate aller Punkte in $\mathbb{X}$ von Null verschieden ist, d.h. daß $\mathbb{X}$ im affinen Raum $D_+(x_0)$ liegt.

Für $j = 1, \ldots, \sigma_{\mathbb{X}}$ betrachte die Abbildung

$$\begin{array}{rccc} \Phi_j : & R_j & \longrightarrow & \mathbb{F}_q^s \\ & f & \longmapsto & (f(P_1), \ldots, f(P_s)) \end{array}$$

wobei die Funktionswerte in $D_+(x_0)$ zu bilden sind.

Satz 3.1. *(Hansen)*

a) Für jedes $j \in \{1, \ldots, \sigma_{\mathbb{X}}\}$ ist die Abbildung Φ_j injektiv. Ihr Bild ist ein linearer Code $C_j(\mathbb{X})$ der Länge s und der Dimension $H_{\mathbb{X}}(j)$.

b) Sei $i \geq 1$ und $j \in \{1, \ldots, \sigma_{\mathbb{X}}\}$. Genau dann gilt $m(C_j(\mathbb{X})) \geq s - i + 1$, wenn für jede Teilmenge $\mathbb{Y} \subseteq \mathbb{X}$ bestehend aus i Punkten gilt, daß $H_{\mathbb{Y}}(j) = H_{\mathbb{X}}(j)$ ist.

*In diesem Fall sagen wir, $\mathbb{X}$ sei $(s - i, j)$-*uniform.

Wie man leicht sieht, kann jeder lineare Code durch Hansen's Konstruktion erhalten werden. Um gute lineare Codes zu konstruieren, braucht man also möglichst große, uniforme Teilmengen $\mathbb{X} \subseteq \mathbb{P}^n(\mathbb{F}_q)$.

Algorithmen zur Uniformitätsberechnung

1) Die $(1, \sigma_{\mathbb{X}})$-Uniformität von $\mathbb{X}$ heißt klassisch auch die *Cayley-Bacharach Eigenschaft.* Man kann sie mit Hilfe einer Liaison-Technik effizient prüfen.

2) Für jedes $j \in \{1, \ldots, \sigma_{\mathbb{X}}\}$ ist die $(s - H_{\mathbb{X}}(j), j)$-Uniformität die maximal mögliche. Um sie zu prüfen, muß man zuerst den *kanonischen Modul* ω_R des Rings R mit Hilfe der *Adjunktionsformel*

$$\omega_R \cong \underline{\mathrm{Hom}}_{K[x_0]}(R, K[x_0])(-1)$$

berechnen und dann gewisse Multiplikationsabbildungen in diesem Modul auf Biinjektivität testen.

3) Um die $(i, \sigma_{\mathbb{X}})$-Uniformität für $i \in \{1, \ldots, s - H_{\mathbb{X}}(\sigma_{\mathbb{X}})\}$ zu prüfen, kann man mit Hilfe einer Variante des Buchberger-Möller Algorithmus die *Separatoren* von $\mathbb{X}$ bestimmen. Dies sind homogene Polynome $f_1, \ldots, f_s$ vom Grad $\sigma_{\mathbb{X}} + 1$ mit $f_i(P_j) = \delta_{ij}$ für $i, j = 1, \ldots, s$. An diesen Separatoren kann man die gewünschte Uniformität erkennen.

Literatur:

1) S. Beck und M. Kreuzer, How to compute the canonical module of a set of points, in: Proc. Conf. MEGA '94 (Santander 1994), Progress in Math. **143**, Birkhäuser, Basel 1996

2) J.P. Hansen, Points in uniform position and maximum distance separable codes, in: Proc. Conf. Zero-Dimensional Schemes (Ravello 1992), de Gruyter, Berlin 1994

3) M. Kreuzer, On the canonical module of a set of points, Boll. Un. Mat. Ital. (8) 1-B (2000), 221 – 261