

Bachelorarbeit

Zeichnung von Hypergraphen in 3D mithilfe des Inzidenzposets

Leon Füger

Abgabedatum: 9. Juli 2020

Betreuer: Prof. Dr. Alexander Wolff



Julius-Maximilians-Universität Würzburg
Lehrstuhl für Informatik I
Algorithmen, Komplexität und wissensbasierte Systeme

Zusammenfassung

Ossona de Mendez Artikel *Realization of Posets* beweist, dass es für Graphen, deren Inzidenzposet einen Realisierer von Dimension d besitzt, eine Zeichnung des Graphen in $d - 1$ Dimensionen gibt. In dieser Arbeit wird dieser Beweis angewendet, um Hypergraphen, konkret Steiner-Tripel-Systeme mit maximal vierdimensionalen Realisierer, in 3D zu zeichnen. Mein Beitrag besteht darin, den Satz von Ossona de Mendez anzuwenden, um damit den K_4 in 2D und die beiden kleinsten Steiner-Systeme $S(2, 3, 7)$ und $S(2, 3, 9)$ in 3D zu zeichnen. In Abschnitt 2 wird der Beweis des Satzes von Ossona de Mendez anhand des K_4 erklärt und infolgedessen wird der Graph mit dieser Methode gezeichnet. Anschließend werden in Abschnitt 3 die zwei Hypergraphen mit dieser Methode gezeichnet, wobei deren Hyperkanten als Polygone dargestellt werden. Zum Schluss werden noch Vorteile und Nachteile des Verfahrens gegenübergestellt und ein kurzer Ausblick auf weiterführende Problemstellungen gegeben.

Inhaltsverzeichnis

1. Einleitung	4
2. Der Beweis des Satzes von Ossona de Mendez am Beispiel des K_4	8
2.1. Die Zeichnung von Graphen in d Dimensionen	9
2.2. Beweis der Trennungseigenschaft für die Ursprungszeichnung	10
2.3. Die Übertragung der Trennungseigenschaft auf die Projektion der Ursprungszeichnung	11
2.4. Zeichnung des K_4	13
3. Berechnung und Zeichnung der Hypergraphen	17
3.1. Zeichnung der Fano-Ebene $S(2, 3, 7)$	17
3.2. Zeichnung des Steiner-Tripel-Systems $S(2, 3, 9)$	20
4. Fazit und Ausblick	24
Literaturverzeichnis	25
A. Code zur Berechnung von $f(X)$	27
B. Realisierer	28
B.1. K_4	28
B.2. $S(2, 3, 7)$	28
B.3. $S(2, 3, 9)$	28

1. Einleitung

Das Thema zu dem diese Arbeit einen Beitrag liefert, ist die graphische Darstellung von Graphen, genauer gesagt von *Hypergraphen*. Hierbei handelt es sich um eine Generalisierung von Graphen, wobei eine Kante eine beliebige Anzahl von Knoten miteinander verbinden kann. Solche Kanten bezeichnet man als *Hyperkanten*. Das Zeichnen von Graphen wurde bereits eingehend erforscht. Einen Überblick hierüber findet man in [DETT99].

Zur Darstellung von Hypergraphen gibt es einige Verfahren. So haben Brandes et al. [BCPS12] Hypergraphen als Graphen mit der selben Knotenmenge wie der des Ursprungsgraphen dargestellt, wobei nun jede Hyperkante einen Subgraphen induziert. Solche Graphen bezeichnet man als *Stütze* eines Hypergraphen; siehe Abbildung 1.1. Buchin et al. [BvKM⁺11] haben einen Algorithmus entwickelt, welcher in Polynomialzeit ermittelt, ob ein Hypergraph eine planare Stütze besitzt und diese gegebenenfalls berechnet.

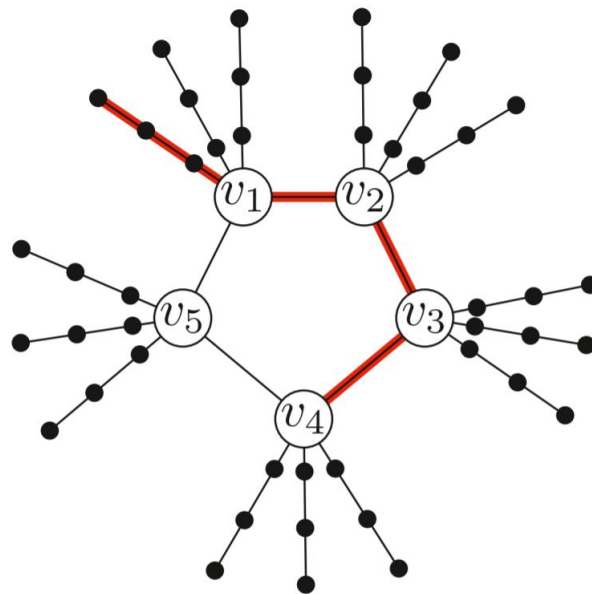


Abb. 1.1.: Stütze-Darstellung von Hypergraphen [BCPS12]

Valdivia et al. [VBP⁺19] haben die sogenannte *PAOH*-Darstellung von Hypergraphen eingeführt, wobei Kanten durch vertikale Geraden dargestellt werden und Knoten links ausgerichtet sind; siehe Abbildung 1.2.

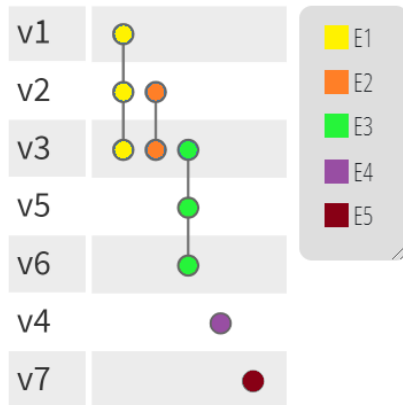


Abb. 1.2.: PAOH-Darstellung eines Graphen [pao]

Weiterhin lassen sich Graphen auch mit Polygonen darstellen. Arseneva et al. [AKK⁺20] haben sogenannte *Seitenkontakt-Darstellungen* untersucht. Hierbei werden die Knoten des Graphen durch Polygone dargestellt, wobei zwei Polygone im Inneren disjunkt sind und genau dann eine Seite teilen, wenn die Knoten adjazent sind. Sie bewiesen, dass *jeder* Graph eine solche Darstellung besitzt, wenn die Polygone nicht konvex sein müssen; siehe Abbildung 1.3. Weiterhin zeigen sie, dass es nicht für jeden Graphen solch eine Darstellung gibt, falls die Polygone konvex sein müssen.

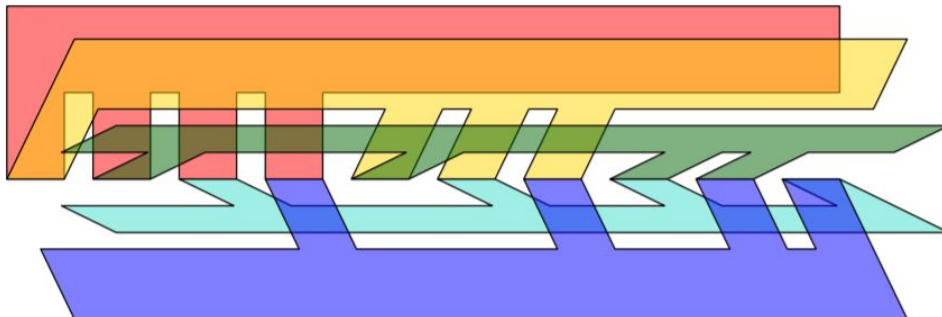


Abb. 1.3.: Eine Seitenkontakt-Darstellung des K_5 von Arseneva et al. [AKK⁺20]

Evans et al. [ERS⁺19] haben ebenfalls versucht, Graphen und Hypergraphen durch sich berührende Polygone in 3D darzustellen. Als *Kontakt-Repräsentation* bezeichnet man eine Menge von sich nicht schneidenden Polygonen, welche die Knoten eines Graphen darstellen. Diese Polygone berühren sich nur an den Ecken, wenn die entsprechenden Knoten adjazent sind. Wir befassen uns mit der Dualität einer solchen Darstellung, also einer Kontakt-Repräsentation, wobei die (Hyper-)Kanten als Polygone dargestellt werden. Evans et al. haben versucht, sogenannte Steiner-Systeme auf diese Weise in 3D darzustellen.

Ein *Steiner-System* $S(t, k, n)$ ist eine n -elementige Menge S mit einer Menge von k -

elementigen Teilmengen von S (auch *Blöcke* genannt), sodass jede t -elementige Teilmenge von S in genau einem Block enthalten ist. Steiner-Systeme mit $k = 3$ werden auch *drei-uniform* oder *Steiner-Tripel-Systeme* genannt. Es gibt solche Tripel-Systeme für jede Anzahl von Knoten in $\{6k + 1; 6k + 3 : k \in \mathbb{N}\}$.

Evans et al. fanden Zeichnungen in 3D für die zwei kleinsten Tripel-Systeme $S(2, 3, 7)$ und $S(2, 3, 9)$, ersteres wird auch *Fano-Ebene* genannt. Für $k = 4$, also *Steiner-Quadrupel-Systeme*, bewiesen sie, dass es keine Zeichnungen ohne Kantenkreuzungen in 3D gibt. Die in dieser Arbeit verwendete Methode von Ossona de Mendez [OdM02] liefert für diese Klasse von Graphen ebenfalls keine Zeichnungen in 3D.

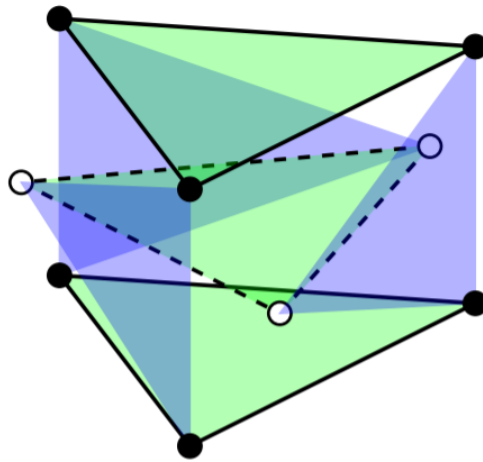


Abb. 1.4.: Eine Kontakt-Repräsentation des $K_{3,3}$ von Evans et al. [ERS⁺19]

Ein *Inzidenzposet* ist eine Darstellung eines Graphen als Halbordnung. Hierbei werden die Knoten als die Menge der Elemente A der Halbordnung bezeichnet und die Kanten als Menge der Elemente B der Halbordnung. Zwischen zwei Elementen $a \in A$ und $b \in B$ besteht eine Relation $a < b$ genau dann, wenn im Graphen die Kante b zum Knoten a inzident ist [Sch89]. Man kann Halbordnungen als *Hasse-Diagramm* darstellen, wobei die höher-geordneten Elemente weiter oben stehen; siehe Abbildung 1.5.

Eine Halbordnung einer Menge X lässt sich als die Schnittmenge einer Menge linearer Ordnungen auf X auffassen, wobei zwei Elemente $a, b \in X$ in der Halbordnung vergleichbar sind, falls in allen linearen Ordnungen entweder $a < b$ oder $b < a$ gilt. Die Menge dieser linearen Ordnungen bezeichnet man als *Realisierer* [DM41]. Entsprechend bezeichnet die *Posetdimension* die kleinste Anzahl der linearen Ordnungen eines Realisierers. Der Realisierer des K_4 , welcher die Halbordnung in Abbildung 1.5 erzeugt, wird in Abschnitt B.1 des Anhangs aufgelistet. Der K_4 hat Posetdimension 3.

Hier kommt das Resultat von Ossona de Mendez [OdM02] ins Bild. Er beweist, dass es für Graphen, deren Inzidenzposet einen Realisierer von Posetdimension d hat, eine Zeichnung in $d - 1$ Dimensionen gibt, wobei die Knoten als Punkte und die Hyperkanten als konvexe Hüllen dieser Punkte dargestellt werden.

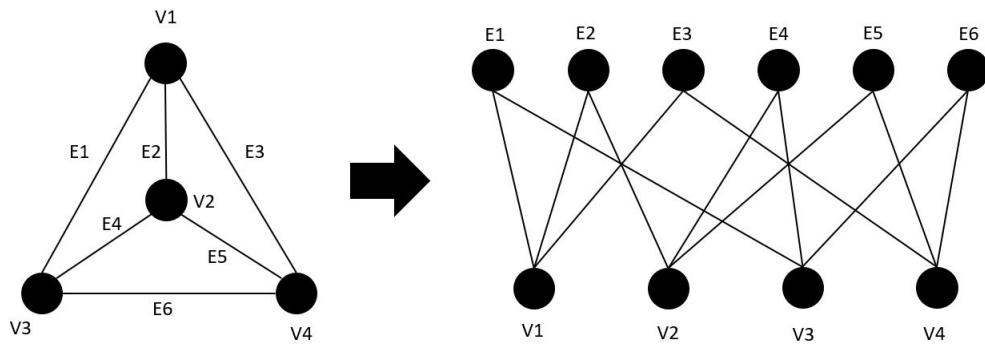


Abb. 1.5.: Ein Hasse-Diagramm des Inzidenzposets des K_4

Mein Beitrag. Das Ziel dieser Arbeit ist es, den Satz von Ossona de Mendez anzuwenden, um drei-uniforme Hypergraphen zu zeichnen, also Graphen, bei denen jede Kante drei Knoten miteinander verbindet. Exemplarisch sollen die beiden Steiner-Systeme $S(2, 3, 7)$ und $S(2, 3, 9)$ in 3D gezeichnet werden, wobei die Hyperkanten als Dreiecke dargestellt werden, wie in der Arbeit von Evans et al. [ERS⁺19]. Der Unterschied zu [ERS⁺19] besteht darin, dass wir mit dem Satz von Ossona de Mendez direkt eine Zeichnung aus dem Realisierer ableiten können und somit kein manuelles Zeichnen notwendig ist. Da die beiden Steiner-Tripel-Systeme jeweils einen Realisierer mit Posetdimension vier besitzen, liefert der Satz eine Zeichnung in 3D und da die konvexe Hülle von drei Punkten ein Dreieck ist, erhalten wir eine Zeichnung mit den Hyperkanten als Dreiecke. In Abschnitt 2 wird zunächst der Satz von Ossona de Mendez anhand des Beispiels des K_4 erklärt und infolgedessen wird der K_4 mit Posetdimension drei in 2D gezeichnet. In Abschnitt 3 werden dann die beiden Steiner-Systeme $S(2, 3, 7)$ und $S(2, 3, 9)$ in 3D gezeichnet und mit den Resultaten aus [ERS⁺19] verglichen. Zuletzt wird in Abschnitt 4 ein Fazit gezogen und ein Ausblick auf weiterführende Problemstellungen gegeben.

2. Der Beweis des Satzes von Ossona de Mendez am Beispiel des K_4

Zunächst müssen noch mehrere Begriffe erläutert werden, die des Down-Sets und der Hyperebene. Die Kenntnis dieser beiden Konzepte lässt uns die Trennungseigenschaft verstehen, welche die Basis für den Beweis von Ossona de Mendez darstellt.

Eine Teilmenge L von einer Menge X bezeichnet man als *Down-Set*, falls für alle $x \in L$ und alle $y \in X$ mit $y \leq x$ gilt: $y \in L$. Ein Down-Set der Menge A nennen wir $\text{Inf}(A)$.

Eine *Hyperebene* im n -dimensionalen Raum lässt sich als Lösungsmenge einer Linearen Gleichung mit n Unbekannten darstellen. Solch eine Hyperebene besitzt stets die Dimension $n - 1$, so ist z.B. eine Hyperebene in 3D eine Ebene, eine Hyperebene in 2D ist eine Gerade usw.

Hiermit lässt sich der Begriff der Trennungseigenschaft verstehen. Die *Trennungseigenschaft* besitzt eine Funktion $f: X \rightarrow \mathbb{R}$ auf den Elementen X einer Halbordnung, wenn für alle $A, B \subseteq X$ eine Hyperebene von \mathbb{R}^n existiert, welche die Punkte von $f(\text{Inf}(A) \setminus \text{Inf}(B))$ und von $f(\text{Inf}(B) \setminus \text{Inf}(A))$ voneinander trennt. Dadurch kommen wir auch schon zum Satz von Ossona de Mendez [OdM02].

Satz 2.1 (Ossona de Mendez). *Sei $\mathbf{P} = (X, P)$ eine endliche Halbordnung und sei $d = \text{Dim } \mathbf{P}$ dessen Posetdimension. Dann existiert eine Funktion $f: X \rightarrow \mathbb{R}^{d-1}$, welche die Trennungseigenschaft erfüllt.*

Der Beweis von Ossona de Mendez funktioniert folgendermaßen: Er zeigt, dass sich die Trennungseigenschaft einer d -dimensionalen Zeichnung eines Graphen auf die Projektion dieser Zeichnung auf eine Hyperebene vererbt. Da eine Hyperebene immer die Dimension $d - 1$ relativ zum Raum hat, in der diese liegt, erhält man somit eine Zeichnung des Graphen auf der Hyperebene. Wenn man nun jeweils eine der Koordinaten der neuen Zeichnung weglässt, bekommt man eine orthogonale Projektion der Hyperebene, also eine Dimension niedriger, und da dabei die Zeichnung nur gestaucht wird, bleibt die Trennungseigenschaft erhalten. Als Beispiel verwenden wir den K_4 .

Zunächst wird in Abschnitt 2.1 eine Funktion festgelegt, mit der man eine Zeichnung des Graphen in d Dimensionen erhält. Dann wird in Abschnitt 2.2 bewiesen, dass die Trennungseigenschaft für solch eine Zeichnung gilt. Diese Zeichnung wird dann in Abschnitt 2.3 auf eine Hyperebene projiziert und es wird gezeigt, dass die Trennungseigenschaft dabei erhalten bleibt. Anschließend wird dieses Verfahren in Abschnitt 2.4 noch auf den K_4 angewendet.

2.1. Die Zeichnung von Graphen in d Dimensionen

Sei $R = \{<_1, \dots, <_d\}$ ein Realisierer von \mathbf{P} und sei $\min(X, <_i)$ das minimale Element der Menge X in der linearen Ordnung $<_i$. Seien F_1, \dots, F_d Funktionen von X zu $]1; +\infty[$, wobei jedes F_i schnell anwächst relativ zu $<_i$, was bedeutet, dass für alle $x <_i y$ gelten muss, dass $F_i(y)$ *mehr als d mal so groß* wie $F_i(x)$ sein muss. Wir definieren also die Funktion $F: X \mapsto \mathbb{R}^d$ mit $F(x) = (F_1(x), \dots, F_d(x))$. Wir nehmen den Realisierer des K_4 als Beispiel, also $d = 3$.

Für alle $A, B \subseteq X$, sodass $\text{Inf}(B) \not\subseteq \text{Inf}(A)$, wird die lineare Form $L_{A,B}: \mathbb{R}^{d-1} \mapsto \mathbb{R}$ definiert:

$$\forall \pi = (\pi_1, \dots, \pi_d) \in \mathbb{R}^d, \quad L_{A,B}(\pi) = \sum_{\substack{1 \leq i \leq d \\ \min(A, <_i) <_i \min(B, <_i)}} \frac{\pi_i}{\min_{a \in A} F_i(a)}$$

Wir setzen z.B.

$$A = V1, E2, E3$$

$$B = V3, E4, E6$$

mit $\text{Inf}(A) = V1$ und $\text{Inf}(B) = V3$. Das Summenzeichen der Linearform hat den Index i von 1 zu d , es werden also die Werte für die verschiedenen linearen Ordnungen aufsummiert, die zweite Zeile darunter ist die Ungleichung $\min(A, <_i) <_i \min(B, <_i)$, die nur für manche der linearen Ordnungen stimmt, man summiert also jeweils nur die Funktionswerte auf, welche diese Bedingung erfüllen. In dem Fall erhält man

$$L_{A,B}(\pi) = \frac{\pi_1}{F_1(V1)} + \frac{\pi_3}{F_3(V1)}$$

$$L_{B,A}(\pi) = \frac{\pi_2}{F_2(V3)}$$

Für $L_{B,A}$ werden also genau die Elemente aufsummiert, welche für $L_{A,B}$ ignoriert werden. Wir verwenden als Funktion für die erste Zeichnung des Graphen

$$F(z) = (d+1)^{\text{rang}(z)}$$

wobei d die Dimension des Realisierers ist und $\text{rang}(z)$ die Position von z in der jeweiligen linearen Ordnung, beginnend mit 0 für das kleinste Element. Diese Funktion erfüllt die Bedingung $x <_i y: d \cdot F_i(x) < F_i(y)$. Entsprechend erhalten wir für den K_4

$$F_{K_4}(z) = 4^{\text{rang}(z)}$$

2.2. Beweis der Trennungseigenschaft für die Ursprungszeichnung

Für jedes $z \in \text{Inf}(B) \setminus \text{Inf}(A)$ ¹ existiert ein $a \in A$ und ein $1 \leq i_0 \leq d$, mit $z >_{i_0} a$. Dann gilt $F_{i_0}(z) > d \cdot F_{i_0}(a)$. Da $\min(b, <_{i_0}) \geq_{i_0} z >_{i_0} \min(A, <_{i_0})$, erhalten wir: $L_{A,B}(F(z)) > d$. Für $z = 3$ wählen wir $a = 1$ und $i_0 = 1$, dann gilt $3 >_1 1$ und $F_1(3) > 3 \cdot F_1(1)$. Weiterhin gilt $3 \geq_1 3 >_1 1$, dadurch erhalten wir $L_{A,B}(F(3)) > 3$.

Für jedes $z \in \text{Inf}(A)$, gilt $F_i(z) \leq F_i(a)$ für alle $i \in [d]$ und alle $a \in A$. Also gilt $L_{A,B}(F(z)) \leq d$. Für $z = 1$ gilt $F_1(1) \leq F_1(1)$, $F_1(1) \leq F_1(6)$, $F_1(1) \leq F_1(7)$ und analog für F_2 und F_3 . Entsprechend gilt:

$$\begin{aligned} L_{A,B}(F(1)) &\leq 3 \\ \frac{F_1(1)}{F_1(1)} + \frac{F_3(1)}{F_3(1)} &\leq 3 \\ 2 &\leq 3 \end{aligned}$$

Für alle $A, B \subseteq X$, sodass $A \cap B = \emptyset$, separiert die Hyperebene $H_{A,B}$ mit Gleichung $L_{A,B}(\pi) - L_{B,A}(\pi) = 0$ die Punkte von $F(\text{Inf}(B) \setminus \text{Inf}(A))$ (für die gilt: $L_{A,B}(F(z)) > d \geq L_{B,A}(F(z))$) und von $F(\text{Inf}(A) \setminus \text{Inf}(B))$ (für die gilt: $L_{A,B}(F(z)) \leq d < L_{B,A}(F(z))$) voneinander. Der Ursprung O gehört zu allen so konstruierten Hyperebenen. Wir erhalten die Hyperebene $H_{A,B}$ mit Gleichung:

$$\frac{\pi_1}{F_1(1)} - \frac{\pi_2}{F_2(3)} + \frac{\pi_3}{F_3(1)} = 0$$

Weiterhin gilt für $z = 1$:

$$\begin{aligned} L_{A,B}(F(z)) &\leq d < L_{B,A}(F(z)) \\ \frac{F_1(1)}{F_1(1)} + \frac{F_3(1)}{F_3(1)} &\leq 3 < \frac{F_2(1)}{F_2(3)} \\ 2 &\leq 3 < \frac{F_2(1)}{F_2(3)} \\ 2 &\leq 3 < \frac{4096}{64} \\ 2 &\leq 3 < 64 \end{aligned}$$

Und für $z = 3$:

¹Da es für den Beweis notwendig ist, notiere ich hier die Knoten und Kanten statt $V1, V2, V3, V4, E1, E2, \dots$ mit $1, 2, 3, 4, 5, 6, \dots$

$$\begin{aligned}
L_{A,B}(F(z)) &> d \geq L_{B,A}(F(z)) \\
\frac{F_1(3)}{F_1(1)} + \frac{F_3(3)}{F_3(1)} &> 3 \geq \frac{F_2(3)}{F_2(3)} \\
\frac{F_1(3)}{F_1(1)} + \frac{F_3(3)}{F_3(1)} &> 3 \geq 1 \\
\frac{64}{1} + \frac{64}{4} &> 3 \geq 1 \\
64 + 16 &> 3 \geq 1 \\
80 &> 3 \geq 1
\end{aligned}$$

Die Gleichung einer Hyperebene hat die Form $a_1x_1 + \dots + a_nx_n = p$, bzw. π für x . Für unser $F(z)$ erhalten wir

$$\frac{1}{1}\pi_1 - \frac{1}{64}\pi_2 + \frac{1}{4}\pi_3 = 0$$

Die Hyperebene H_{AB} teilt die Down-sets von A und B , also die Punkte, erfüllt also die Trennungseigenschaft. Der Ursprung O ist der Punkt $0, 0, 0$.

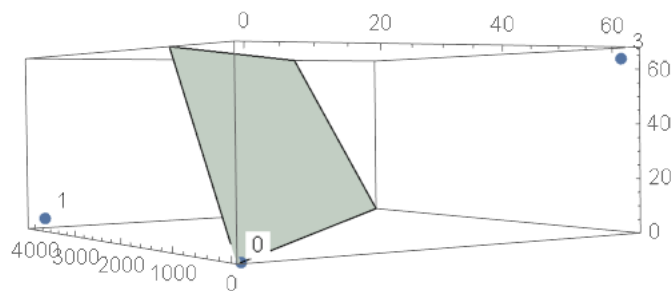


Abb. 2.1.: Die Hyperebene für K_4 trennt 1 von 3

Entsprechend erhält man für jede Wahl von A und B eine Hyperebene, welche die Down-Sets der beiden Mengen voneinander trennt. Entsprechend gilt die Trennungseigenschaft.

2.3. Die Übertragung der Trennungseigenschaft auf die Projektion der Ursprungszeichnung

Nun betrachten wir eine Hyperebene H_0 mit Gleichung $\sum_{1 \leq i \leq d} \pi_i = 1$, welche den Ursprung O und die Bilder von X unter F voneinander trennt. Mit jedem Element z von X wird der Punkt $f(z)$ von H_0 assoziiert, welcher den Schnittpunkt von H_0 mit der Gerade $(O, F(z))$ darstellt. Wir bekommen für H_0 :

$$\pi_1 + \pi_2 + \pi_3 = 1$$

Abbildung 2.2 zeigt diese Hyperebene mit den dazugehörigen Geraden (hier nur für den Punkt 3, es wird mit allen Punkten sehr unübersichtlich). Der Schnittpunkt der jeweiligen Geraden mit der Ebene ist der jeweilige Punkt $f(z)$ von H_0 .

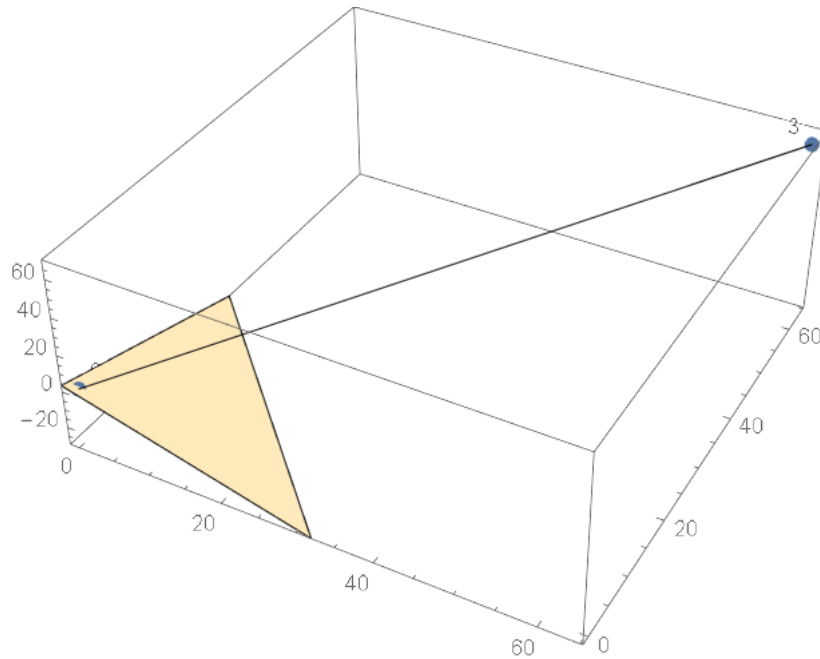


Abb. 2.2.: Die Hyperebene H_0 für K_4 mit der Gerade von O zu 3

Für alle $A, B \subset X$, sodass $A \cap B = \emptyset$, da $H_{A,B}$ O beinhaltet, teilt die Hyperebene $H_{A,B} \cap H_0$ von H_0 die Punkte von $F(\text{Inf}(B) \setminus \text{Inf}(A))$ und von $F(\text{Inf}(A) \setminus \text{Inf}(B))$ voneinander. Da die Hyperebene H_0 stets die Dimension $d - 1$ besitzt, erhält man eine Zeichnung des Graphen auf der Hyperebene. Abbildung 2.3 zeigt H_{AB} und H_0 . Da der Schnittpunkt der Ebenen davon eine Dimension niedriger ist (also hier eine Gerade), kann man die Punkte 1, 3, bzw. die Down-sets von A und B auf der Fläche H_0 darstellen, wobei die Gerade $H_{AB} \cap H_0$ diese Down-sets voneinander trennt. Die Punkte lassen sich auf der Ebene H_0 zeichnen mit $f(z)$ also dem Schnittpunkt der Ursprungsgeraden der einzelnen Punkte. Man erhält also eine Projektion.

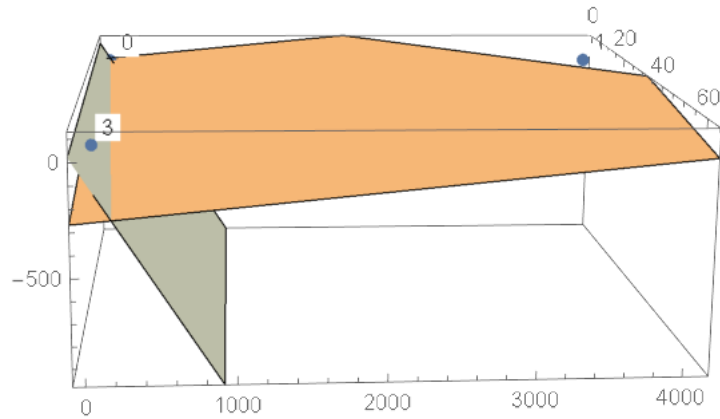


Abb. 2.3.: H_0 und $H_{A,B}$

2.4. Zeichnung des K_4

Mithilfe eines Pythonskripts *dim3.py* (siehe Anhang A) wurden die Schnittpunkte der Ursprungsgeraden mit H_0 berechnet. Dafür muss zunächst H_0 von Koordinatenform in Parameterdarstellung umgewandelt werden.

$$\pi_1 + \pi_2 + \pi_3 = 1$$

$$H_0(\pi) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + r \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} + s \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}$$

Da wir $F = (d+1)^{\text{rang}(z)}$ definiert haben, bekommen wir hier $F_{K_4} = 4^{\text{rang}(z)}$, entsprechend erhalten wir die Funktionswerte der Punkte:

	F_1	F_2	F_3
$F(1)$	4^0	4^6	4^1
$F(2)$	4^1	4^0	4^6
$F(3)$	4^3	4^3	4^3
$F(4)$	4^6	4^1	4^0

Mit diesen Funktionswerten erhält man eine Zeichnung des K_4 in 3D:

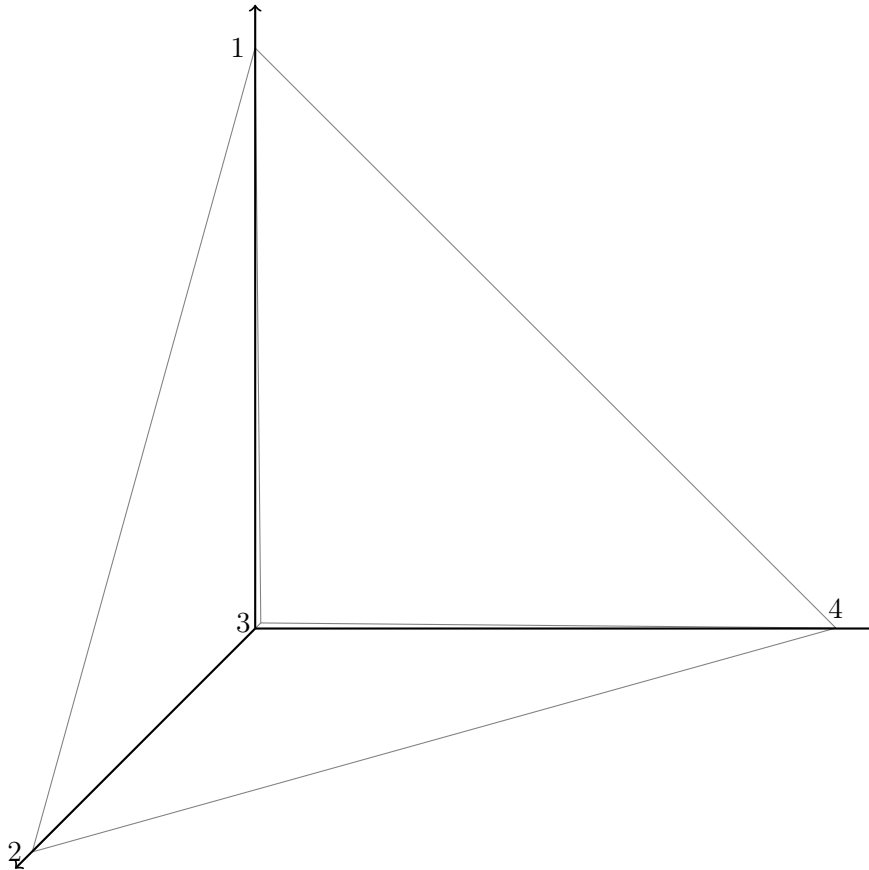


Abb. 2.4.: Eine Zeichnung des K_4 in 3D

Die Vorgehensweise zum Finden des Schnittpunktes wird zunächst für $z = 1$ erklärt. Die Gleichung für die Ursprungsgerade von $z = 1$ ist

$$l_1 = v \begin{bmatrix} 1 \\ 4096 \\ 4 \end{bmatrix}$$

Um den Schnittpunkt zu finden, muss man die Gleichungen gleichsetzen und auflösen.

$$v \begin{bmatrix} 1 \\ 4096 \\ 4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + r \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} + s \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}$$

Das resultierende Gleichungssystem wird aufgelöst, man erhält $s = 0.9987807851743478$ und $r = 0.000243842965130456$. Nun werden r und s in die Ebenengleichung eingesetzt. Dadurch erhält man $f(1)$. Analog für die anderen Punkte. Mithilfe des Pythonskripts `dim3.py` wurden die folgenden Koordinaten berechnet:

	x -Koordinate	y -Koordinate	z -Koordinate
$f(1)$	0.000243842965130456	0.9987807851743478	0.0009753718605217676
$f(2)$	0.0009753718605218239	0.000243842965130456	0.9987807851743477
$f(3)$	0.3333333333333333	0.3333333333333333	0.3333333333333333
$f(4)$	0.9987807851743477	0.000975371860521824	0.0002438429651305243

Man erhält so eine Zeichnung auf H_0 , also eine 2D-Zeichnung im 3D-Raum:

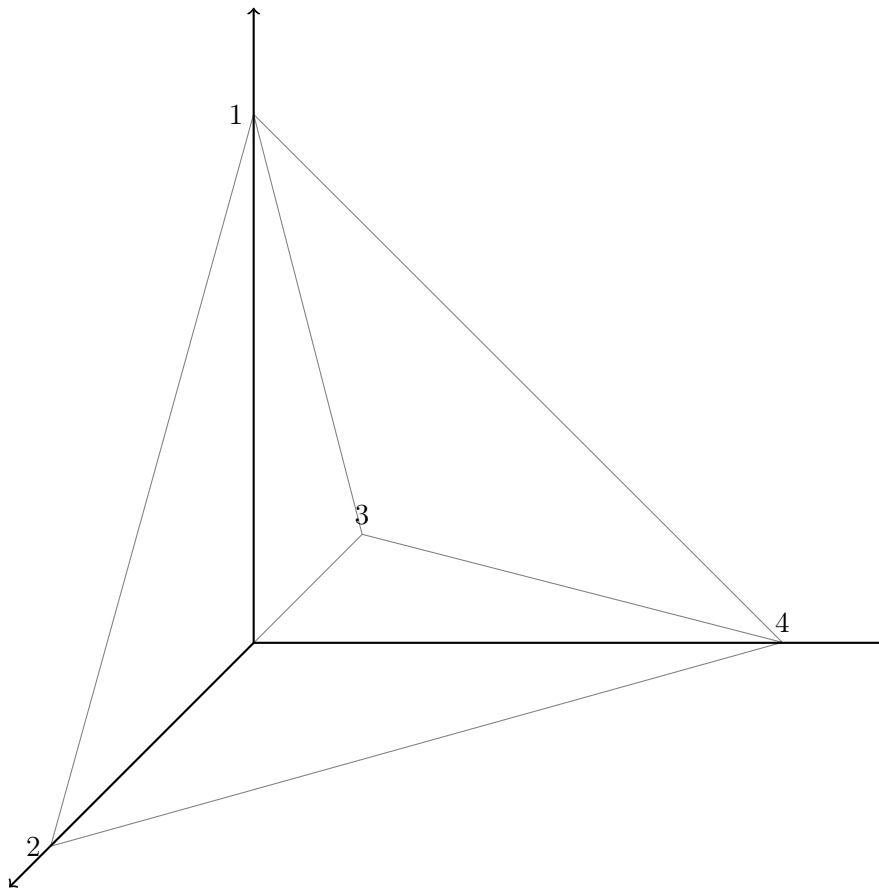


Abb. 2.5.: Der fertige K_4 in 3D

Wenn man jeweils die z -Koordinate weglässt, bekommt man eine orthogonale Projektion auf die xy -Ebene. Die Trennungseigenschaft geht hierdurch nicht verloren, da die Zeichnung nur “gestaucht” wird.

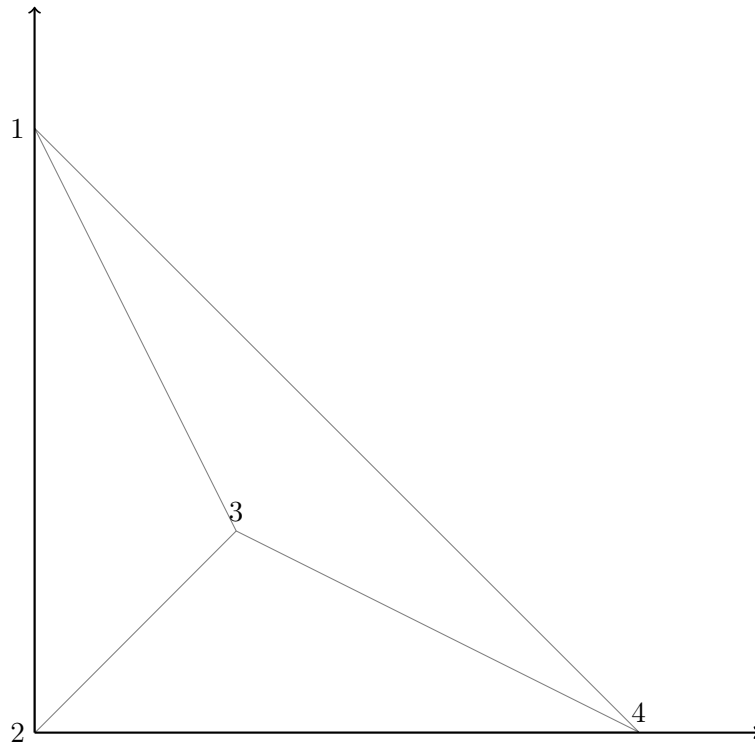


Abb. 2.6.: Der fertige K_4 in 2D

Hier lässt sich noch erwähnen, dass man stets eine planare Zeichnung bekommt, wenn die Dimension des Realisierers maximal 3 beträgt [Sch89].

3. Berechnung und Zeichnung der Hypergraphen

Wie bereits erwähnt betrachten wir drei-uniforme Hypergraphen, d.h. jede Kante verbindet drei Knoten miteinander. Die betrachteten Graphen sind auch *linear*, was bedeutet, das sich zwei Kanten in maximal einem Knoten schneiden. Ein Beispiel für solche Hypergraphen sind drei-uniforme Steiner-Systeme. Wir betrachten die beiden kleinsten dieser Systeme, nämlich $S(2, 3, 7)$, auch *Fano-Ebene* genannt, sowie $S(2, 3, 9)$. Höhere Steiner-Systeme, z.B. Quadrupel-Systeme, besitzen keine vierdimensionalen Realisierer und sind daher für uns uninteressant und können, wie schon von [ERS⁺19] bewiesen, nicht kreuzungsfrei in 3D gezeichnet werden.

3.1. Zeichnung der Fano-Ebene $S(2, 3, 7)$

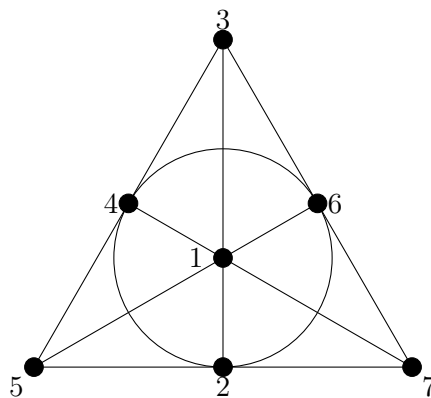


Abb. 3.1.: Die Fano-Ebene $S(2, 3, 7)$

Die Fano-Ebene hat einen Realisierer mit Dimension 4; siehe Anhang A. Also verwenden wir für die Koordinaten der ersten Zeichnung in 4D die Funktion

$$F_{S(2,3,7)}(z) = 5^{\text{rang}(z)}$$

Man erhält die Punkte:

	F_1	F_2	F_3	F_4
$F(1)$	1	625	78125	625
$F(2)$	78125	1	625	3125
$F(3)$	1953125	3125	1	1
$F(4)$	5	5	9765625	5
$F(5)$	25	1953125	3125	25
$F(6)$	125	25	5	125
$F(7)$	3125	15625	25	78125

Analog zum K_4 erhält man die Gleichung der Hyperebene H_0 , diesmal aber in 4D, da die Fano-Ebene keinen dreidimensionalen Realisierer besitzt:

$$H_0(\pi) = \pi_1 + \pi_2 + \pi_3 + \pi_4 = 1$$

Als Beispiel für die Berechnung der Schnittpunkte zwischen H_0 und den Ursprungsgeraden der Koordinaten verwenden wir wieder $z = 1$. Die Gleichung der Ursprungsgerade:

$$l_1 = v \begin{bmatrix} 1 \\ 625 \\ 78125 \\ 625 \end{bmatrix}$$

Die Parameterform der Hyperebene:

$$H_0(\pi) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} + r \begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \end{bmatrix} + s \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix} + t \begin{bmatrix} 0 \\ 0 \\ 1 \\ -1 \end{bmatrix}$$

Für den Schnittpunkt erhält man also folgende Gleichung:

$$v \begin{bmatrix} 1 \\ 625 \\ 78125 \\ 625 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} + r \begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \end{bmatrix} + s \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix} + t \begin{bmatrix} 0 \\ 0 \\ 1 \\ -1 \end{bmatrix}$$

Mithilfe von `4dim.py` wurden die Schnittpunkte ausgerechnet (der Übersicht zugunsten wird die vierte Koordinate hier weggelassen):

	x -Koordinate	y -Koordinate	z -Koordinate
$f(1)$	0.000012598266478532	0.007873916549082847	0.9842395686353559
$f(2)$	0.9541868190922859	0.000012213591284381	0.007633494552738287
$f(3)$	0.9984015351805391	0.001597442456288862	0.000000511181586012
$f(4)$	0.000000511999213569	0.000000511999213569	0.9999984640023595
$f(5)$	0.000012779226090067	0.9983770382865615	0.001597403261258498
$f(6)$	0.4464285714285714	0.08928571428571429	0.017857142857142853
$f(7)$	0.032249742002063	0.1612487100103199	0.000257997936016511

Ohne die vierte Koordinate bekommt man wieder eine orthogonale Projektion der Hyperebene, diesmal in 3D; siehe Abbildung 3.2. Wie man hier erkennt, drängen sich aufgrund der exponentiell steigenden Funktionswerte der Originalzeichnung in der resultierenden Zeichnung die Knoten teils sehr nah beieinander. Entsprechend liefert die Methode für eine große Anzahl an Knoten und Kanten zwar eine gültige (die Trennungseigenschaft erfüllende) Zeichnung, aber keine besonders übersichtliche, weshalb sich die Methode eher für kleinere Graphen eignet.

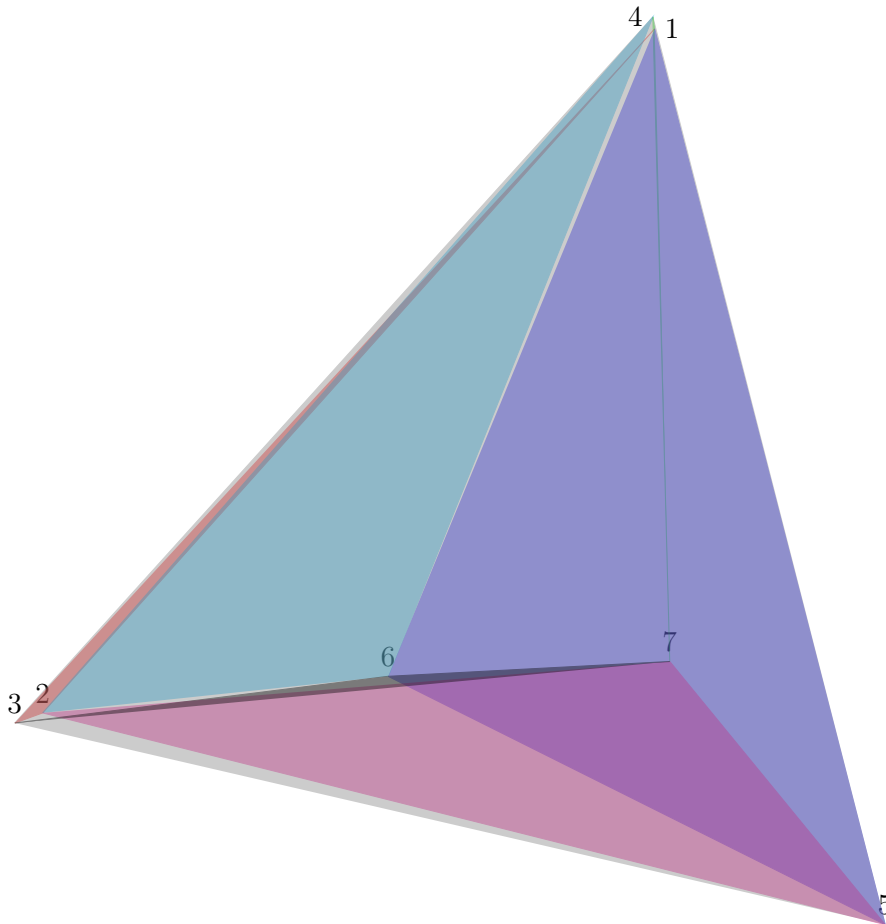


Abb. 3.2.: Zeichnung der Fano-Ebene, die mithilfe des Satzes von Ossona de Mendez [OdM02] konstruiert wurde.

Hier wird noch das Ergebnis von Evans et al. meinem Ergebnis gegenübergestellt; siehe Abbildung 3.3.

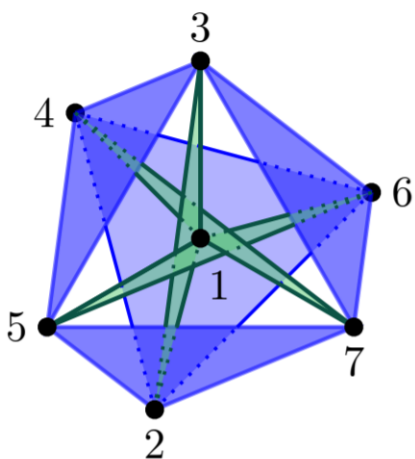


Abb. 3.3.: Zeichnung der Fano-Ebene aus [ERS⁺19]

3.2. Zeichnung des Steiner-Tripel-Systems $S(2, 3, 9)$

Die Zeichnung des Steiner-Systems $S(2, 3, 9)$ läuft ähnlich ab, wie die der Fano-Ebene. Da es sich wieder um einen vierdimensionalen Realisierer handelt, verwenden wir wieder

$$F_{S(2,3,9)}(z) = 5^{\text{rang}(z)}$$

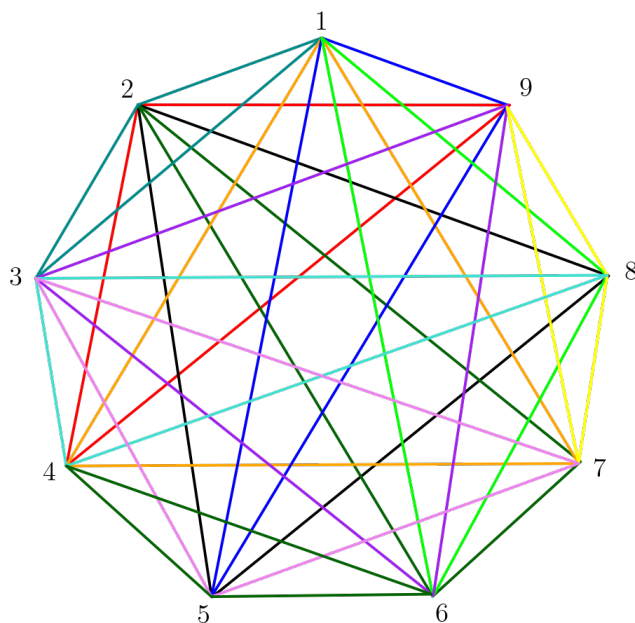


Abb. 3.4.: Das Steiner-System $S(2, 3, 9)$; jedes Tripel entspricht einem einfarbigen Dreieck.

Man erhält folgende Funktionswerte:

	F_1	F_2	F_3	F_4
$F(1)$	1953125	78125	1953125	5
$F(2)$	3125	30517578125	25	25
$F(3)$	25	1	152587890625	125
$F(4)$	1	5	5	152587890625
$F(5)$	625	25	48828125	48828125
$F(6)$	30517578125	125	3125	1
$F(7)$	78125	1953125	625	78125
$F(8)$	5	244140625	78125	3125
$F(9)$	244140625	3125	1	1953125

Analog zur Fano-Ebene erhält man durch die Gleichsetzung der Geradengleichung zur Ebenengleichung die Schnittpunkte.

	x -Koordinate	y -Koordinate	z -Koordinate
$f(1)$	0.49019546328412456	0.019607818531364982	0.49019546328412444
$f(2)$	0.00000010239998934646	0.9999998959616108	0.00000000081919991477
$f(3)$	0.00000000016383999983	0.00000000000655359999	0.9999999990104065
$f(4)$	0.00000000000655359999	0.00000000003276799999	0.00000000003276799999
$f(5)$	0.00000639995740188353	0.00000025599829607534	0.499996672022151
$f(6)$	0.999998934712434	0.00000000409599956365	0.00000010239998909145
$f(7)$	0.0370260663507109	0.9256516587677724	0.0002962085308056872
$f(8)$	0.00000002047318610437	0.9996672902526177	0.0003198935328808376
$f(9)$	0.992050890560675	0.00001269825139917664	0.00000000406344044773

Wir erhalten wieder eine Zeichnung in 3D, siehe Abbildung 3.5, und ähnlich zur Fano-Ebene drängen sich manche der Knoten aneinander.

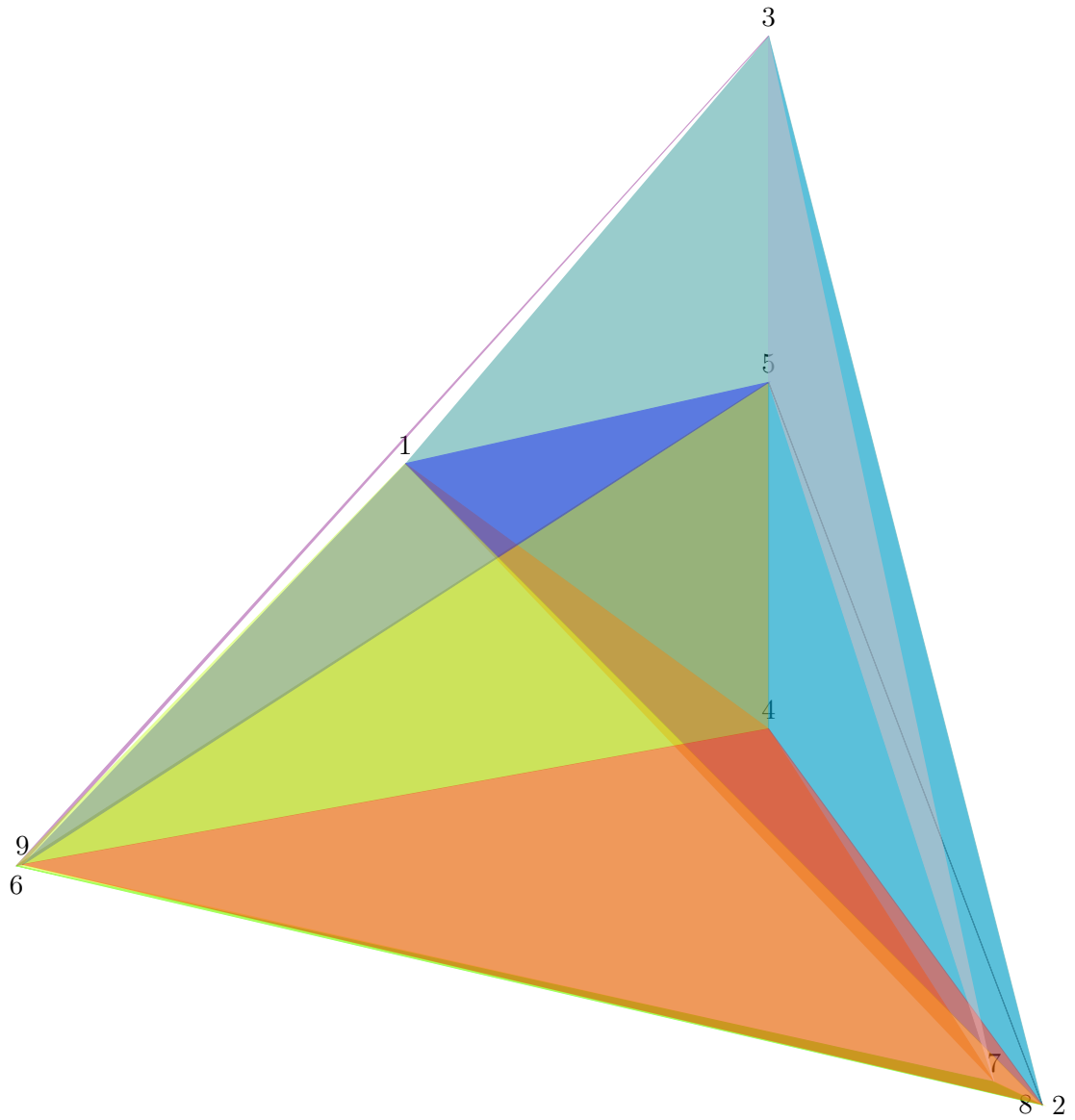


Abb. 3.5.: Zeichnung des Steiner-Systems $S(2,3,9)$, das mithilfe des Satzes von Ossona de Mendez [OdM02] konstruiert wurde.

Und auch hier nochmal das Ergebnis von Evans et al. für diesen Hypergraphen; siehe Abbildung 3.6.

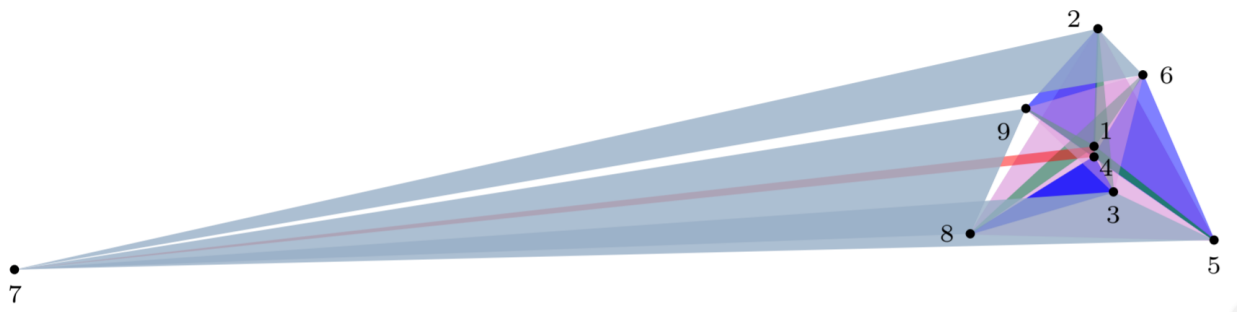


Abb. 3.6.: Zeichnung des $S(2, 3, 7)$ aus [ERS⁺19]

4. Fazit und Ausblick

Die hier angewandte Methode erlaubt es, Hypergraphen mit wenigen Knoten und Kanten in 3D im Prinzip übersichtlich zu zeichnen, sofern sie einen maximal vierdimensionalen Realisierer besitzen. Bei den Steiner-Systemen kommt das Verfahren jedoch bereits an seine Grenzen, da die exponentiell ansteigenden Funktionswerte des Verfahrens von Ossona de Mendez dazu führen, dass Knoten in der resultierenden Zeichnung direkt nebeneinander liegen und somit die Zeichnung in diesen Bereichen unübersichtlich wird. Ein Vorteil dieser Methode ist allerdings, dass die Zeichnungen direkt aus dem Realisierer hergeleitet werden können und somit kein manuelles Zeichnen notwendig ist. Anreiz für zukünftige Arbeiten könnte die Suche nach einem Verfahren sein, das keine exponentiell steigenden Funktionswerte benötigt und entsprechend für Graphen mit mehr als einer einstelligen Anzahl an Knoten und Kanten noch übersichtliche Zeichnungen liefern kann. Hier wäre der Ansatz des *kräftebasierten* Zeichnens von Graphen interessant; siehe Eades [Ead84] und Fruchterman et al. [FR91]. Eventuell wäre es damit möglich, die extrem unterschiedlichen Kantenlängen anzugleichen, ohne dass sich dadurch Kanten schneiden.

Literaturverzeichnis

- [AKK⁺20] Elena Arseneva, Linda Kleist, Boris Klemz, Maarten Löffler, André Schulz, Birgit Vogtenhuber und Alexander Wolff: Representing Graphs by Polygons with Edge Contacts in 3D. In: Steven Chaplick, Philipp Kindermann und Alexander Wolff (Herausgeber): *Proc. 36th European Workshop on Computational Geometry (EuroCG'20)*, Seiten 53:1–8, 2020.
- [BCPS12] Ulrik Brandes, Sabine Cornelsen, Barbara Pampel und Arnaud Sallaberry: Path-based supports for hypergraphs. *J. Discrete Algorithms*, 14:248–261, 2012.
- [BvKM⁺11] Kevin Buchin, Marc J. van Kreveld, Henk Meijer, Bettina Speckmann und Kevin Verbeek: On Planar Supports for Hypergraphs. *J. Graph Algorithms Appl.*, 15(4):533–549, 2011.
- [DETT99] Guiseppe Di Battista, Peter Eades, Roberto Tamassia und Ioannis G. Tollis: *Graph Drawing*. Prentice Hall, Upper Saddle River, NJ, 1999.
- [DM41] Ben Dushnik und Edwin W. Miller: Partially ordered sets. *American Journal of Mathematics*, 63(3):600–610, 1941.
- [Ead84] Peter Eades: A heuristics for graph drawing. *Congressus numerantium*, 42:146–160, 1984.
- [ERS⁺19] William Evans, Paweł Rzażewski, Noushin Saeedi, Chan Su Shin und Alexander Wolff: Representing Graphs and Hypergraphs by Touching Polygons in 3D. In: Daniel Archambault und Csaba D. Tóth (Herausgeber): *Proc. 27th Int. Symp. Graph Drawing & Network Vis. (GD'19)*, Band 11904 der Reihe *Lect. Notes Comput. Sci.*, Seiten 18–32. Springer-Verlag, 2019. <http://arxiv.org/abs/1908.08273>.
- [FR91] Thomas M. J. Fruchterman und Edward M. Reingold: Graph drawing by force-directed placement. *Software – Practice & Experience*, 21(11):1129–1164, 1991.
- [OdM02] Patrice Ossona de Mendez: Realization of Posets. *J. Graph Algorithms Appl.*, 6(1):149–153, 2002.
- [pao] https://commons.wikimedia.org/wiki/File:PAOH_representation_of_the_hypergraph.png. Stand: 1.7.2020.

- [Sch89] Walter Schnyder: Planar graphs and poset dimension. *Order*, 5(4):323–343, 1989.
- [VBP⁺19] Paola Valdivia, Paolo Buono, Catherine Plaisant, Nicole Dufournaud und Jean Daniel Fekete: Analyzing Dynamic Hypergraphs with Parallel Aggregated Ordered Hypergraph Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 2019.

A. Code zur Berechnung von $f(X)$

Kurze Python-Skripte zum Berechnung der Schnittpunkte der Geraden, welche jeweils durch den Ursprung und die jeweiligen Punkte verlaufen, mit der Hyperebene H_0 . Jeweils für drei- und vierdimensionale Realisierer.

```
1 import numpy as np
2
3 val = (0, 0, 0) # Die Koordinaten des jeweiligen Punktes der ersten
   Zeichnung in 3D
4
5 A = np.array([[ -1 * val[0], 1, 0], [ -1 * val[1], 0, 1], [ -1 * val[2], -1,
   -1]])
6 b = np.array([0, 0, -1])
7 z = np.linalg.solve(A, b) # Die Gleichung von H0 und die der
   Ursprungsgerade gleichgesetzt
8
9 x1 = z[1] # Die resultierenden Koordinaten des Schnittpunktes
10 x2 = z[2]
11 x3 = 1 - z[1] - z[2]
12
13 print(x1, x2, x3)
```

Listing A.1: dim3.py

```
1 import numpy as np
2
3 val = (0, 0, 0, 0)
4
5 A = np.array([[ -1 * val[0], 1, 0, 0], [ -1 * val[1], 0, 1, 0], [ -1 * val
   [2], 0, 0, 1], [ -1 * val[3], -1, -1, -1]])
6 b = np.array([0, 0, 0, -1])
7 z = np.linalg.solve(A, b)
8
9 x1 = z[1]
10 x2 = z[2]
11 x3 = z[3]
12 x4 = 1 - z[1] - z[2] - z[3]
13
14 print(x1, x2, x3, x4)
```

Listing A.2: dim4.py

B. Realisierer

Hier werden noch die Realisierer der Graphen aufgelistet. Den Realisierer des K_4 habe ich selbst ermittelt, die der Steiner-Systeme wurden mit einem ILP von Oksana Firman berechnet.

B.1. K_4

$$E1[1, 3]; E2[1, 2]; E3[1, 4]; E4[2, 3]; E5[2, 4]; E6[3, 4]$$

$$R1: E6 > E5 > E3 > V4 > E4 > E1 > V3 > E2 > V2 > V1$$

$$R2: E3 > E2 > E1 > V1 > E6 > E4 > V3 > E5 > V4 > V2$$

$$R3: E5 > E4 > E2 > V2 > E6 > E1 > V3 > E3 > V1 > V4$$

B.2. $S(2, 3, 7)$

$$E1[123]; E2[147]; E3[156]; E4[246]; E5[257]; E6[345]; E7[367]$$

$$R1: E7 > E6 > E4 > E1 > V3 > E5 > V2 > E2 > V7 > E3 > V6 > V5 > V4 > V1$$

$$R2: E7 > E6 > E5 > E3 > V5 > E2 > E1 > V7 > V3 > V1 > E4 > V6 > V4 > V2$$

$$R3: E6 > E4 > E2 > V4 > E3 > E1 > V1 > E5 > V5 > V2 > E7 > V7 > V6 > V3$$

$$R4: E7 > E4 > E3 > V6 > E5 > E2 > V7 > E1 > V2 > V1 > E6 > V5 > V4 > V3$$

B.3. $S(2, 3, 9)$

$$E1[123]; E2[456]; E3[789]; E4[147]; E5[258]; E6[369];
E7[159]; E8[267]; E9[348]; E10[168]; E11[249]; E12[357]$$

R1: $E6 > E10 > E8 > E11 > E2 > V6 > E3 > E7 > V9 > E1 > E4 > V1 > E12$
 $> V7 > E5 > V2 > V5 > E9 > V3 > V8 > V4$

R2: $E9 > E5 > E8 > E11 > E1 > V2 > E10 > E3 > V8 > E12 > E4 > V7 > E7$
 $> V1 > E6 > V9 > E2 > V6 > V5 > V4 > V3$

R3: $E12 > E9 > E6 > E1 > V3 > E5 > E7 > E10 > E2 > V5 > E4 > V1 > E3$
 $> V8 > E8 > V6 > V7 > E11 > V2 > V4 > V9$

R4: $E2 > E11 > E9 > E4 > V4 > E3 > E12 > E7 > E5 > V5 > E6 > V9 > E8$
 $> V7 > E10 > V8 > E1 > V3 > V2 > V1 > V6$

Erklärung

Hiermit versichere ich die vorliegende Abschlussarbeit selbstständig verfasst zu haben, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt zu haben, und die Arbeit bisher oder gleichzeitig keiner anderen Prüfungsbehörde unter Erlangung eines akademischen Grades vorgelegt zu haben.

Würzburg, den 9. Juli 2020

.....
Leon Füger