On Order Types, Projective Classes, and Realizations

Manfred Scheucher

20. November 2014

Motivation

- Counting Convex 5-Holes
- abstrakte und realisierte Ergebnisse
- abstrakt $\xrightarrow{?}$ realisierbar

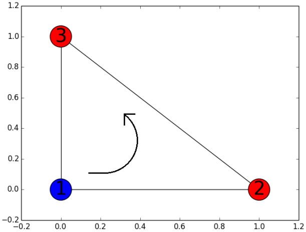
Order Types

Punktmenge

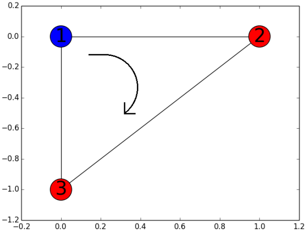
- $P = \{p_1, \ldots, p_n\} \subset \mathbb{R}^2$
- keine kollinearen Punkte

Order Types

Beispiel: 3 Punkte



(a) positiv orientiert



(b) negativ orientiert

Order Types

Λ-Matrix

- Je 3 Punkte positiv oder negativ orientiert:

$$\Lambda_{ijk} := \operatorname{sgn}\det\begin{pmatrix} 1 & 1 & 1 \\ x_i & x_j & x_k \\ y_i & y_j & y_k \end{pmatrix} \in \{-1, +1\}$$

- Λ-Matrix: $\Theta(n^3)$ Einträge
- ähnliche Λ-Matrizen durch Permutation der Indizes und durch Spiegelung $\Rightarrow$ Order Type

Order Types

Λ-Matrix

- Je 3 Punkte positiv oder negativ orientiert:

$$\Lambda_{ijk} := \operatorname{sgn} \det \begin{pmatrix} 1 & 1 & 1 \\ x_i & x_j & x_k \\ y_i & y_j & y_k \end{pmatrix} \in \{-1, +1\}$$

- Λ-Matrix: $\Theta(n^3)$ Einträge
- ähnliche Λ-Matrizen durch Permutation der Indizes und durch Spiegelung $\Rightarrow$ Order Type

Order Types

Λ-Matrix

- Je 3 Punkte positiv oder negativ orientiert:

$$\Lambda_{ijk} := \operatorname{sgn}\det\begin{pmatrix} 1 & 1 & 1 \\ x_i & x_j & x_k \\ y_i & y_j & y_k \end{pmatrix} \in \{-1, +1\}$$

- Λ-Matrix: $\Theta(n^3)$ Einträge
- ähnliche Λ-Matrizen durch Permutation der Indizes und durch Spiegelung $\Rightarrow$ Order Type

Order Types

λ-Matrix

- $\lambda_{ij} := |\{k \mid \Lambda_{ijk} > 0\}|$
- λ-Matrix: $\Theta(n^2)$ Einträge
- Äquivalent!

Order Types

λ-Matrix

- $\lambda_{ij} := |\{k \mid \Lambda_{ijk} > 0\}|$
- λ-Matrix: $\Theta(n^2)$ Einträge
- Äquivalent!

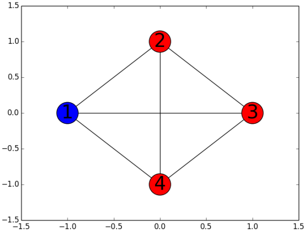
Order Types

λ-Matrix

- $\lambda_{ij} := |\{k \mid \Lambda_{ijk} > 0\}|$
- λ-Matrix: $\Theta(n^2)$ Einträge
- Äquivalent!

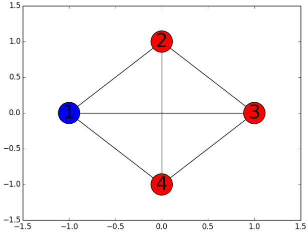
Order Types

Beispiel



Order Types

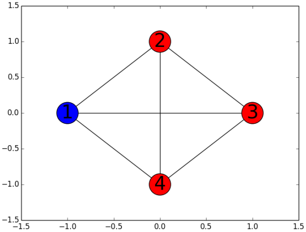
Beispiel



$$\Lambda_{123} = -1$$
$$\Lambda_{124} = -1$$
$$\Lambda_{134} = -1$$
$$\Lambda_{234} = -1$$

Order Types

Beispiel

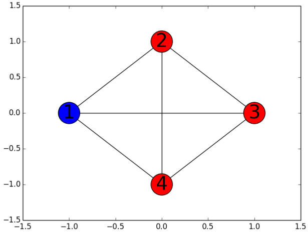


$$\lambda_{12} = 0$$

$$\Lambda_{123} = -1$$
$$\Lambda_{124} = -1$$
$$\Lambda_{134} = -1$$
$$\Lambda_{234} = -1$$

Order Types

Beispiel



$\Lambda_{123} = -1$

$\Lambda_{124} = -1$

$\Lambda_{134} = -1$

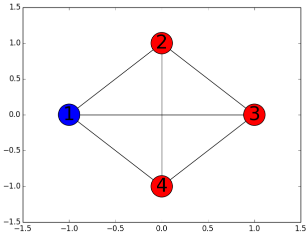
$\Lambda_{234} = -1$

$\lambda_{12} = 0$

$\lambda_{13} = 1$

Order Types

Beispiel

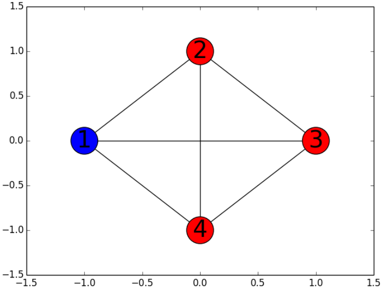


$$\Lambda_{123} = -1 \qquad \Lambda_{124} = -1 \qquad \Lambda_{134} = -1 \qquad \Lambda_{234} = -1$$

$$\lambda_{12} = 0 \qquad \lambda_{13} = 1 \qquad \lambda_{14} = 2 \qquad \lambda_{23} = 0 \qquad \lambda_{24} = 1 \qquad \lambda_{34} = 0$$

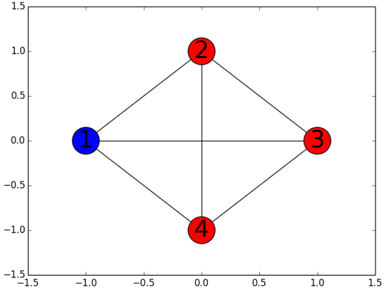
Order Types

Beispiel



Order Types

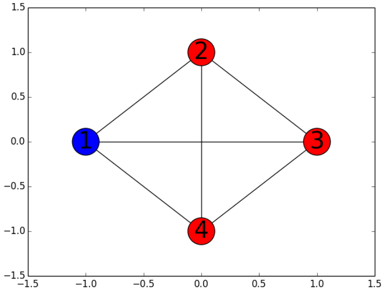
Beispiel



0	0	1	2
2	0	0	1
1	2	0	0
0	1	2	0

Order Types

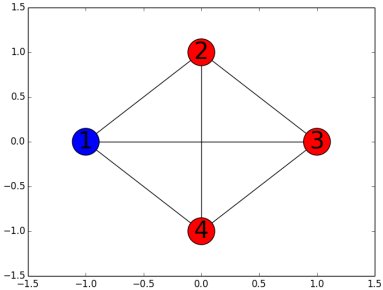
Beispiel



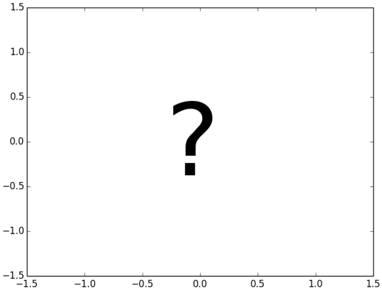
0	0	1	2		0	1	2	0
2	0	0	1		1	0	1	1
1	2	0	0		0	1	0	2
0	1	2	0		2	1	0	0

Order Types

Beispiel



$$\begin{matrix} 0 & 0 & 1 & 2 \\ 2 & 0 & 0 & 1 \\ 1 & 2 & 0 & 0 \\ 0 & 1 & 2 & 0 \end{matrix}$$



$$\begin{matrix} 0 & 1 & 2 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 2 \\ 2 & 1 & 0 & 0 \end{matrix}$$

Realisierbarkeit

Realisierungsproblem

- Gegeben: Λ-Matrix
- Gesucht: Punktmenge, die Λ induziert, oder Beweis für Nichtrealisierbarkeit

Realisierbarkeit

Geometrisches Einfügen: Beispiel

$$\begin{array}{cccc} 0 & \color{red}{1} & \color{red}{2} & \color{red}{0} \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 2 \\ 2 & 1 & 0 & 0 \end{array}$$

Realisierbarkeit

Geometrisches Einfügen: Beispiel

$$\begin{matrix} 0 & 1 & 2 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 2 \\ 2 & 1 & 0 & 0 \end{matrix}$$

$$\Lambda_{123} = -$$
$$\Lambda_{124} = +$$
$$\Lambda_{134} = +$$
$$\Lambda_{234} = +$$

Realisierbarkeit

Geometrisches Einfügen: Beispiel

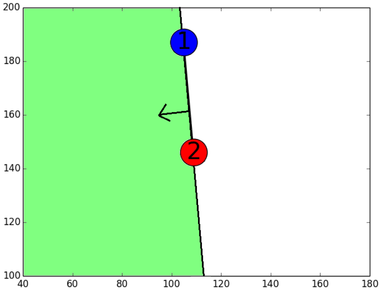
$$\begin{matrix} 0 & 1 & 2 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 2 \\ 2 & 1 & 0 & 0 \end{matrix}$$

$\Lambda_{123} = -$

$\Lambda_{124} = +$

$\Lambda_{134} = +$

$\Lambda_{234} = +$

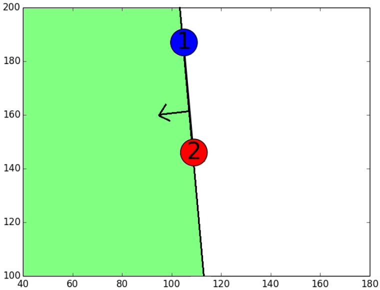


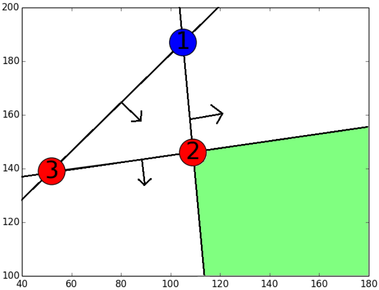
Realisierbarkeit

Geometrisches Einfügen: Beispiel

$$\begin{array}{cccc} 0 & 1 & 2 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 2 \\ 2 & 1 & 0 & 0 \end{array}$$

$$\Lambda_{123} = -$$
$$\color{red}{\Lambda_{124} = +}$$
$$\color{red}{\Lambda_{134} = +}$$
$$\color{red}{\Lambda_{234} = +}$$

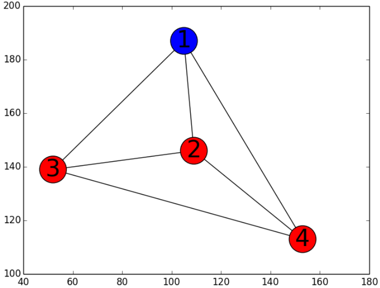




Realisierbarkeit

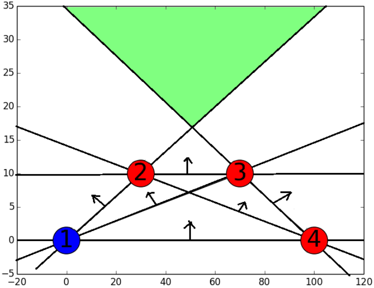
Geometrisches Einfügen: Beispiel

$$\begin{matrix} 0 & 1 & 2 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 2 \\ 2 & 1 & 0 & 0 \end{matrix} \longrightarrow$$



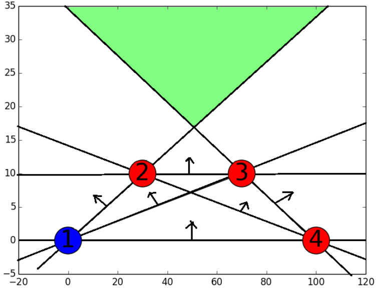
Realisierbarkeit

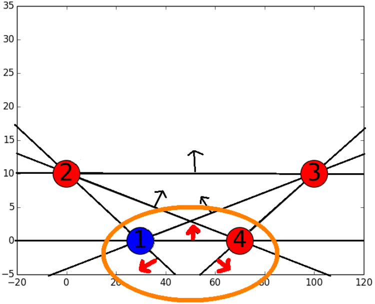
Geometrisches Einfügen: Das Problem



Realisierbarkeit

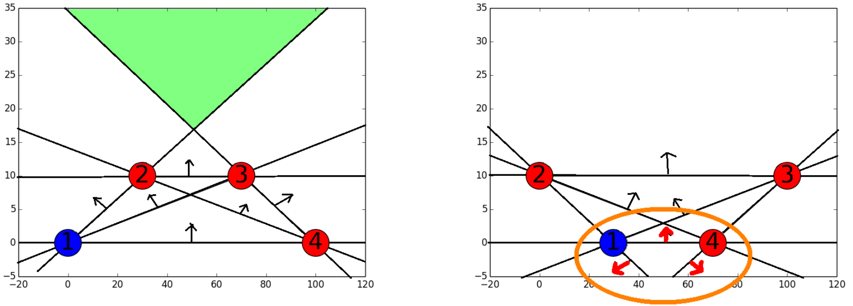
Geometrisches Einfügen: Das Problem





Realisierbarkeit

Geometrisches Einfügen: Das Problem



Das Realisierungsproblem ist i.A. NP-schwer!

Realisierbarkeit

Geometrisches Einfügen: Anmerkung

Speedup möglich durch

- Forward Checking und Backtracking
- Quadtree-Struktur beim Suchen

Realisierbarkeit

Geometrisches Einfügen: Anmerkung

Speedup möglich durch

- Forward Checking und Backtracking
- Quadtree-Struktur beim Suchen

Realisierbarkeit

Realisierung mittels Quadratically Constrained Programming (QCP)

- Äquivalent:
  - $\Lambda_{ijk} = +1$
  - $\det \begin{pmatrix} 1 & 1 & 1 \\ x_i & x_j & x_k \\ y_i & y_j & y_k \end{pmatrix} > 0$
  - $x_j y_k - x_k y_j + x_k y_i - x_i y_k + x_i y_j - x_j y_i > 0$
  - $x_j y_k - x_k y_j + x_k y_i - x_i y_k + x_i y_j - x_j y_i \geq \epsilon > 0$
  - $x_j y_k - x_k y_j + x_k y_i - x_i y_k + x_i y_j - x_j y_i \geq 1$
- QCP mit $\binom{n}{3}$ Nebenbedingungen (NB)

Realisierbarkeit

Realisierung mittels Quadratically Constrained Programming (QCP)

- Äquivalent:
  - $\Lambda_{ijk} = +1$
  - $\det \begin{pmatrix} 1 & 1 & 1 \\ x_i & x_j & x_k \\ y_i & y_j & y_k \end{pmatrix} > 0$
  - $x_j y_k - x_k y_j + x_k y_i - x_i y_k + x_i y_j - x_j y_i > 0$
  - $x_j y_k - x_k y_j + x_k y_i - x_i y_k + x_i y_j - x_j y_i \geq \epsilon > 0$
  - $x_j y_k - x_k y_j + x_k y_i - x_i y_k + x_i y_j - x_j y_i \geq 1$
- QCP mit $\binom{n}{3}$ Nebenbedingungen (NB)

Realisierbarkeit

Realisierung mittels Quadratically Constrained Programming (QCP)

- Äquivalent:
  - $\Lambda_{ijk} = +1$
  - $\det \begin{pmatrix} 1 & 1 & 1 \\ x_i & x_j & x_k \\ y_i & y_j & y_k \end{pmatrix} > 0$
  - $x_j y_k - x_k y_j + x_k y_i - x_i y_k + x_i y_j - x_j y_i > 0$
  - $x_j y_k - x_k y_j + x_k y_i - x_i y_k + x_i y_j - x_j y_i \geq \epsilon > 0$
  - $x_j y_k - x_k y_j + x_k y_i - x_i y_k + x_i y_j - x_j y_i \geq 1$
- QCP mit $\binom{n}{3}$ Nebenbedingungen (NB)

Realisierbarkeit

Realisierung mittels Quadratically Constrained Programming (QCP)

- Äquivalent:
  - $\Lambda_{ijk} = +1$
  - $\det \begin{pmatrix} 1 & 1 & 1 \\ x_i & x_j & x_k \\ y_i & y_j & y_k \end{pmatrix} > 0$
  - $x_j y_k - x_k y_j + x_k y_i - x_i y_k + x_i y_j - x_j y_i > 0$
  - $x_j y_k - x_k y_j + x_k y_i - x_i y_k + x_i y_j - x_j y_i \geq \epsilon > 0$
  - $x_j y_k - x_k y_j + x_k y_i - x_i y_k + x_i y_j - x_j y_i \geq 1$
- QCP mit $\binom{n}{3}$ Nebenbedingungen (NB)

Realisierbarkeit

Realisierung mittels Quadratically Constrained Programming (QCP)

- Äquivalent:
  - $\Lambda_{ijk} = +1$
  - $\det \begin{pmatrix} 1 & 1 & 1 \\ x_i & x_j & x_k \\ y_i & y_j & y_k \end{pmatrix} > 0$
  - $x_j y_k - x_k y_j + x_k y_i - x_i y_k + x_i y_j - x_j y_i > 0$
  - $x_j y_k - x_k y_j + x_k y_i - x_i y_k + x_i y_j - x_j y_i \geq \epsilon > 0$
  - $x_j y_k - x_k y_j + x_k y_i - x_i y_k + x_i y_j - x_j y_i \geq 1$
- QCP mit $\binom{n}{3}$ Nebenbedingungen (NB)

Realisierbarkeit

Realisierung mittels Quadratically Constrained Programming (QCP)

- Äquivalent:
  - $\Lambda_{ijk} = +1$
  - $\det \begin{pmatrix} 1 & 1 & 1 \\ x_i & x_j & x_k \\ y_i & y_j & y_k \end{pmatrix} > 0$
  - $x_j y_k - x_k y_j + x_k y_i - x_i y_k + x_i y_j - x_j y_i > 0$
  - $x_j y_k - x_k y_j + x_k y_i - x_i y_k + x_i y_j - x_j y_i \geq \epsilon > 0$
  - $x_j y_k - x_k y_j + x_k y_i - x_i y_k + x_i y_j - x_j y_i \geq 1$
- QCP mit $\binom{n}{3}$ Nebenbedingungen (NB)

Realisierbarkeit

Realisierung mittels Minimierung

- Formulierung als Minimierungsproblem (ohne NB):

$$\sum_{\substack{i,j,k \\ \Lambda_{ijk}=+1}} \phi(x_j y_k - x_k y_j + x_k y_i - x_i y_k + x_i y_j - x_j y_i - 1)$$

mit $\phi : \mathbb{R} \to [0, \infty)$, $\phi(x) = 0 \Leftrightarrow x \geq 0$

- Beispiel: $\phi(x) := (|x| - x)^2$, stetig differenzierbar

Realisierbarkeit

Realisierung mittels Minimierung

- Formulierung als Minimierungsproblem (ohne NB):

$$\sum_{\substack{i,j,k \\ \Lambda_{ijk}=+1}} \phi(x_j y_k - x_k y_j + x_k y_i - x_i y_k + x_i y_j - x_j y_i - 1)$$

  mit $\phi : \mathbb{R} \to [0, \infty)$, $\phi(x) = 0 \Leftrightarrow x \geq 0$
- Beispiel: $\phi(x) := (|x| - x)^2$, stetig differenzierbar

Beweis für Nichtrealisierbarkeit

Grassmann-Plücker-Heuristik

- Grassmann-Plücker-Gleichung:

$$X_{abc} \cdot X_{ade} + X_{abd} \cdot X_{aec} + X_{abe} \cdot X_{acd} = 0$$

wobei

$$X_{ijk} := \det \begin{pmatrix} 1 & 1 & 1 \\ x_i & x_j & x_k \\ y_i & y_j & y_k \end{pmatrix}$$

- (abgeschwächte) Formulierung als LP:

$$L_{abc} + L_{ade} - L_{abd} - L_{ace} < 0$$
$$L_{abe} + L_{acd} - L_{abd} - L_{ace} < 0$$

Beweis für Nichtrealisierbarkeit

Grassmann-Plücker-Heuristik

- Grassmann-Plücker-Gleichung:

$$X_{abc} \cdot X_{ade} + X_{abd} \cdot X_{aec} + X_{abe} \cdot X_{acd} = 0$$

  wobei

$$X_{ijk} := \det \begin{pmatrix} 1 & 1 & 1 \\ x_i & x_j & x_k \\ y_i & y_j & y_k \end{pmatrix}$$

- (abgeschwächte) Formulierung als LP:

$$\begin{aligned} L_{abc} + L_{ade} - L_{abd} - L_{ace} &< 0 \\ L_{abe} + L_{acd} - L_{abd} - L_{ace} &< 0 \end{aligned}$$

Beweis für Nichtrealisierbarkeit

- Grassmann-Plücker-Gleichung:

$$X_{abc} \cdot X_{ade} + X_{abd} \cdot X_{aec} + X_{abe} \cdot X_{acd} = 0$$

  wobei

$$X_{ijk} := \det \begin{pmatrix} 1 & 1 & 1 \\ x_i & x_j & x_k \\ y_i & y_j & y_k \end{pmatrix}$$

- (abgeschwächte) Formulierung als LP:

$$\begin{aligned} L_{abc} + L_{ade} - L_{abd} - L_{ace} &\leq -1 \\ L_{abe} + L_{acd} - L_{abd} - L_{ace} &\leq -1 \end{aligned}$$

Beweis für Nichtrealisierbarkeit

Grassmann-Plücker-Heuristik

- $10\binom{n}{5}$ Ungleichungen in $\binom{n}{3}$ Variablen
- Algorithmen: Simplex- und Innere-Punkte-Verfahren
- Verifizieren einer Lösung: polynomiell
- $\Rightarrow$ Zertifikat für Nichtrealisierbarkeit

Beweis für Nichtrealisierbarkeit

Grassmann-Plücker-Heuristik

- $10\binom{n}{5}$ Ungleichungen in $\binom{n}{3}$ Variablen
- Algorithmen: Simplex- und Innere-Punkte-Verfahren
- Verifizieren einer Lösung: polynomiell
- $\Rightarrow$ Zertifikat für Nichtrealisierbarkeit

Beweis für Nichtrealisierbarkeit

Grassmann-Plücker-Heuristik

- $10\binom{n}{5}$ Ungleichungen in $\binom{n}{3}$ Variablen
- Algorithmen: Simplex- und Innere-Punkte-Verfahren
- Verifizieren einer Lösung: polynomiell
- $\Rightarrow$ Zertifikat für Nichtrealisierbarkeit

Beweis für Nichtrealisierbarkeit

Grassmann-Plücker-Heuristik

- $10\binom{n}{5}$ Ungleichungen in $\binom{n}{3}$ Variablen
- Algorithmen: Simplex- und Innere-Punkte-Verfahren
- Verifizieren einer Lösung: polynomiell
- $\Rightarrow$ Zertifikat für Nichtrealisierbarkeit

Sub Order Types

- Induzierte Sub Λ-Matrix Λ':

$$\Lambda'_{i,j,k} := \Lambda_{a_i,a_j,a_k}$$

  für $A = \{a_1, \ldots, a_m\} \subseteq \{1, \ldots, n\}$
- Induzierter Sub Order Type analog...

Sub Order Types

- Induzierte Sub Λ-Matrix Λ':

$$\Lambda'_{i,j,k} := \Lambda_{a_i,a_j,a_k}$$

  für $A = \{a_1, \ldots, a_m\} \subseteq \{1, \ldots, n\}$

- Induzierter Sub Order Type analog...

Sub Order Types

- Order Type realisierbar
  $\Rightarrow$ alle Sub Order Types realisierbar
- ein Sub Order Type nicht realisierbar
  $\Rightarrow$ Order Type nicht realisierbar

Sub Order Types

- Order Type realisierbar
  $\Rightarrow$ alle Sub Order Types realisierbar
- ein Sub Order Type nicht realisierbar
  $\Rightarrow$ Order Type nicht realisierbar

Iterative Realisierungsmethode

1. Wähle bel. Permutation der Indizes $\pi \in S_n$
2. Für $k = 1, \ldots, n$:
   3. $A^{(k)} := \{\pi_1, \ldots, \pi_k\}$
   4. $\Lambda^{(k)}$, der durch $A^{(k)}$ induzierte Sub Order Type
   5. Teste $\Lambda^{(k)}$ auf Realisierbarkeit
   6. Wenn $\Lambda^{(k)}$ nicht realisierbar
      $\rightarrow$ STOP (Λ nicht realisierbar)
7. Realisierung von $\Lambda = \Lambda^{(n)}$ gefunden!

Iterative Realisierungsmethode

1. Wähle bel. Permutation der Indizes $\pi \in S_n$
2. Für $k = 1, \ldots, n$:
    3. $A^{(k)} := \{\pi_1, \ldots, \pi_k\}$
    4. $\Lambda^{(k)}$, der durch $A^{(k)}$ induzierte Sub Order Type
    5. Teste $\Lambda^{(k)}$ auf Realisierbarkeit
    6. Wenn $\Lambda^{(k)}$ nicht realisierbar
       $\rightarrow$ STOP (Λ nicht realisierbar)
7. Realisierung von $\Lambda = \Lambda^{(n)}$ gefunden!

Iterative Realisierungsmethode

1. Wähle bel. Permutation der Indizes $\pi \in S_n$
2. Für $k = 1, \ldots, n$:
    3. $A^{(k)} := \{\pi_1, \ldots, \pi_k\}$
    4. $\Lambda^{(k)}$, der durch $A^{(k)}$ induzierte Sub Order Type
    5. Teste $\Lambda^{(k)}$ auf Realisierbarkeit
    6. Wenn $\Lambda^{(k)}$ nicht realisierbar
       $\rightarrow$ STOP (Λ nicht realisierbar)
7. Realisierung von $\Lambda = \Lambda^{(n)}$ gefunden!

Iterative Realisierungsmethode

1. Wähle bel. Permutation der Indizes $\pi \in S_n$
2. Für $k = 1, \dots, n$:
    3. $A^{(k)} := \{\pi_1, \dots, \pi_k\}$
    4. $\Lambda^{(k)}$, der durch $A^{(k)}$ induzierte Sub Order Type
    5. Teste $\Lambda^{(k)}$ auf Realisierbarkeit
    6. Wenn $\Lambda^{(k)}$ nicht realisierbar
       $\rightarrow$ STOP (Λ nicht realisierbar)
7. Realisierung von $\Lambda = \Lambda^{(n)}$ gefunden!

Iterative Realisierungsmethode

1. Wähle bel. Permutation der Indizes $\pi \in S_n$
2. Für $k = 1, \ldots, n$:
   3. $A^{(k)} := \{\pi_1, \ldots, \pi_k\}$
   4. $\Lambda^{(k)}$, der durch $A^{(k)}$ induzierte Sub Order Type
   5. Teste $\Lambda^{(k)}$ auf Realisierbarkeit
   6. Wenn $\Lambda^{(k)}$ nicht realisierbar
      $\rightarrow$ STOP (Λ nicht realisierbar)
7. Realisierung von $\Lambda = \Lambda^{(n)}$ gefunden!

Iterative Realisierungsmethode

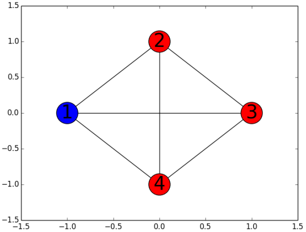
1. Wähle bel. Permutation der Indizes $\pi \in S_n$
2. Für $k = 1, \ldots, n$:
    3. $A^{(k)} := \{\pi_1, \ldots, \pi_k\}$
    4. $\Lambda^{(k)}$, der durch $A^{(k)}$ induzierte Sub Order Type
    5. Teste $\Lambda^{(k)}$ auf Realisierbarkeit
    6. Wenn $\Lambda^{(k)}$ nicht realisierbar
       $\rightarrow$ STOP (Λ nicht realisierbar)
7. Realisierung von $\Lambda = \Lambda^{(n)}$ gefunden!

Beobachtung

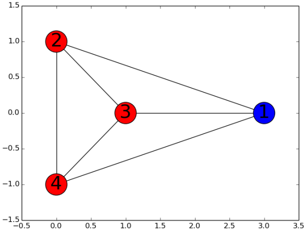
Die Wahl der Permutation ist bei nicht-realisierbaren Order Types entscheident über die Anzahl an Iterationen!

Projective Classes

Beispiel 1



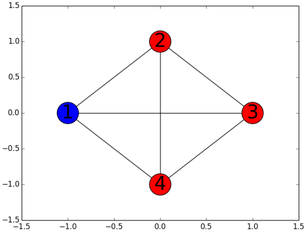
$\xleftrightarrow{1}$



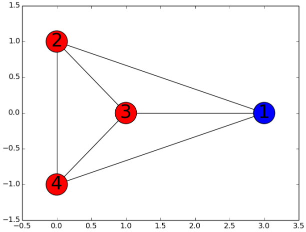
- Orientierungstripel werden invertiert
- Welcher OT in PC ist am “einfachsten”?
- (Rück-)Transformation der Realisierung?

Projective Classes

Beispiel 1



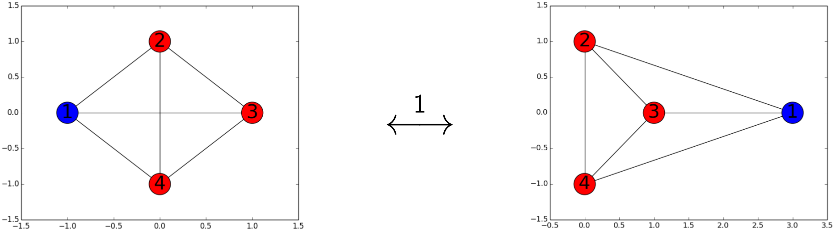
$$\overset{1}{\longleftrightarrow}$$



- Orientierungstripel werden invertiert
- Welcher OT in PC ist am “einfachsten”?
- (Rück-)Transformation der Realisierung?

Projective Classes

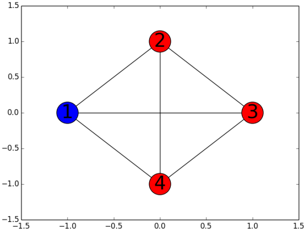
Beispiel 1



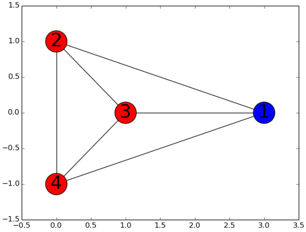
- Orientierungstripel werden invertiert
- Welcher OT in PC ist am "einfachsten"?
- (Rück-)Transformation der Realisierung?

Projective Classes

Beispiel 1



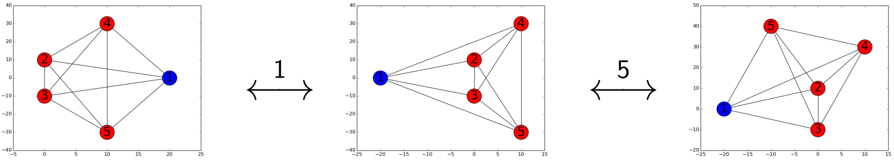
$$\xleftrightarrow{1}$$



- Orientierungstripel werden invertiert
- Welcher OT in PC ist am “einfachsten”?
- (Rück-)Transformation der Realisierung?

Projective Classes

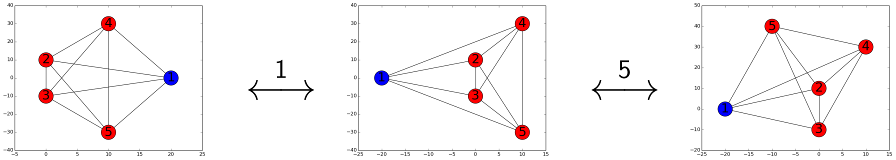
Beispiel 2



- Jeden Punkt einzeln transformieren (rotieren)
- $\Rightarrow$ i.A. sehr große Koordinaten

Projective Classes

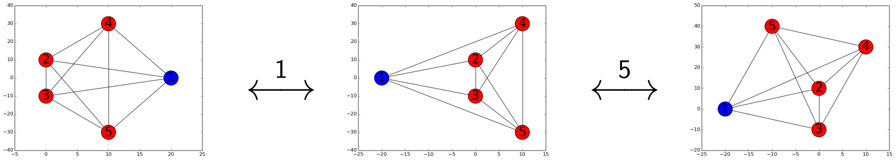
Beispiel 2



- Jeden Punkt einzeln transformieren (rotieren)
- $\Rightarrow$ i.A. sehr große Koordinaten

Projective Classes

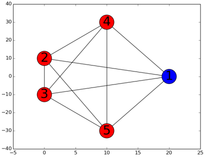
Beispiel 2



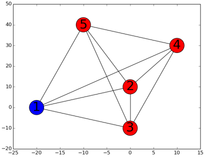
- Jeden Punkt einzeln transformieren (rotieren)
- $\Rightarrow$ i.A. sehr große Koordinaten

Projective Classes

Beispiel 2



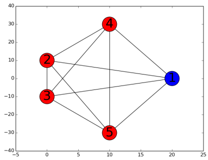
$\xleftrightarrow{1,5}$



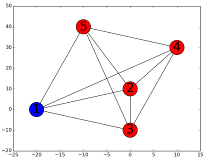
- Punkte gemeinsam transformieren (rotieren)
- B Bit Durchmesser $\rightarrow$ nB Bit Durchmesser

Projective Classes

Beispiel 2



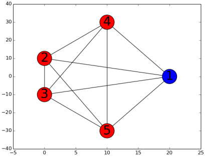
$\overset{1,5}{\longleftrightarrow}$



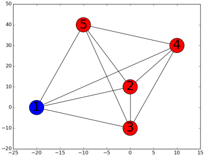
- Punkte gemeinsam transformieren (rotieren)
- B Bit Durchmesser $\rightarrow$ nB Bit Durchmesser

Projective Classes

Beispiel 2



$$\overset{1,5}{\longleftrightarrow}$$



- Punkte gemeinsam transformieren (rotieren)
- B Bit Durchmesser $\rightarrow$ nB Bit Durchmesser

pyotlib

- Implementierung aller genannten Methoden
- RealizationTester-Verschachtelung möglich:
  - Versuche OT zu realisieren (Nichtrealisierbarkeit zu beweisen)
  - Falls erfolgreich → STOP
  - Sonst übergib an inneren RealizationTester (sofern vorhanden)
- Beispiel:

```
tester = GPRealizationTester(
           PCRealizationTester(
             MatlabRealizationTester()
           )
         )
```

pyotlib

- Implementierung aller genannten Methoden
- RealizationTester-Verschachtelung möglich:
  - Versuche OT zu realisieren (Nichtrealisierbarkeit zu beweisen)
  - Falls erfolgreich $\rightarrow$ STOP
  - Sonst übergib an inneren RealizationTester (sofern vorhanden)
- Beispiel:

```
tester = GPRealizationTester(
           PCRealizationTester(
             MatlabRealizationTester()
           )
         )
```

pyotlib

- Implementierung aller genannten Methoden
- RealizationTester-Verschachtelung möglich:
  - Versuche OT zu realisieren (Nichtrealisierbarkeit zu beweisen)
  - Falls erfolgreich → STOP
  - Sonst übergib an inneren RealizationTester (sofern vorhanden)
- Beispiel:

```
tester = GPRealizationTester(
           PCRealizationTester(
             MatlabRealizationTester()
           )
         )
```

pyotlib

- Implementierung aller genannten Methoden
- RealizationTester-Verschachtelung möglich:
  - Versuche OT zu realisieren (Nichtrealisierbarkeit zu beweisen)
  - Falls erfolgreich $\rightarrow$ STOP
  - Sonst übergib an inneren RealizationTester (sofern vorhanden)
- Beispiel:

```
tester = GPRealizationTester(
           PCRealizationTester(
             MatlabRealizationTester()
           )
         )
```

pyotlib

- Implementierung aller genannten Methoden
- RealizationTester-Verschachtelung möglich:
  - Versuche OT zu realisieren (Nichtrealisierbarkeit zu beweisen)
  - Falls erfolgreich → STOP
  - Sonst übergib an inneren RealizationTester (sofern vorhanden)
- Beispiel:

```
tester = GPRealizationTester(
           PCRealizationTester(
             MatlabRealizationTester()
           )
         )
```

pyotlib

- Implementierung aller genannten Methoden
- RealizationTester-Verschachtelung möglich:
  - Versuche OT zu realisieren (Nichtrealisierbarkeit zu beweisen)
  - Falls erfolgreich → STOP
  - Sonst übergib an inneren RealizationTester (sofern vorhanden)
- Beispiel:

```
tester = GPRealizationTester(
           PCRealizationTester(
             MatlabRealizationTester()
           )
         )
```

Resultate

Convex 5-Holes

n	$h_5(n)$ (alte Schranke)	$h_5(n)$ (neue Schranke)
17	≤ 16	
18	≤ 21	
19	≤ 28	≤ 27[1]
20	≤ 36	≤ 33[1]
21	≤ 43	≤ 41
22	≤ 52	≤ 48
23	≤ 63	≤ 59
24	≤ 71	≤ 67
25	≤ 82	≤ 75
26	≤ 90	≤ 88
27	≤ 103	≤ 100
28	≤ 117	≤ 115
29	≤ 134	≤ 131
30	≤ 155	≤ 150
...	...	...

[1] Realisierung eines abstrakten Order Types

Resultate

Rectilinear Crossing Numbers

n	$\bar{cr}(n)$
40	≤ 33070[1]
52	≤ 99161
57	≤ 145170
58	≤ 156042
60	≤ 179523
61	≤ 192267
98	≤ 1347651
99	≤ 1404666

[1] Realisierung eines abstrakten Order Types

Resultate

Realisierungen minimaler Größe

n	3	4	5	6	7	8	9	10
# bits needed (with properties)	2	2	3	3	4	5	≤ 7	≤ 13
# bits needed (without properties)	1	2	2	3	4	5	≤ 6	≤ 9

Resultate

Universal Order Types

m	3	4	5	6	7	8	9
OT	3	5	6	9	12	≤ 22	?
PC	3	4	5	7	9	≤ 14	≤ 30

Vielen Dank für Ihre Aufmerksamkeit!