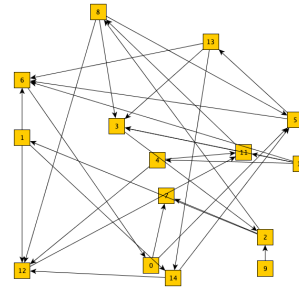


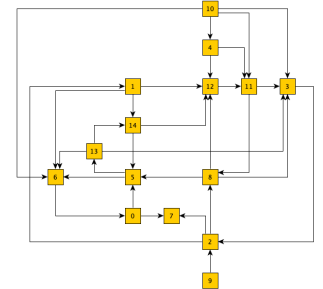
## Kombinatorische Optimierung mit Maschinellem Lernen

Zu einem Graphen gibt es sehr viele mögliche Zeichnungen. In manchen davon lässt sich die Struktur des Graphen und die Beziehungen der Knoten deutlich besser erkennen als in anderen.

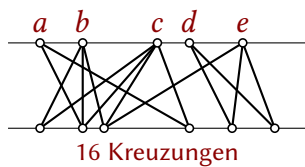
Ein Qualitätsmerkmal guter Zeichnungen sind wenige Kantenkreuzungen. Das entsprechende Optimierungsproblem ist das *Kreuzungsminimierungsproblem* (KMP).



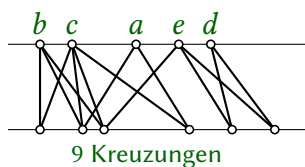
Eine ziemlich schlechte Zeichnung.  
( $> 50$  Kreuzungen)



Eine ziemlich gute Zeichnung.  
(4 Kreuzungen)



16 Kreuzungen



9 Kreuzungen

Das KMP wird häufig auf kleinere Instanzen des *einseitigen Kreuzungsminimierungsproblems* (EKMP) reduziert. Dabei sind die Knoten auf zwei parallele horizontale Linien verteilt. Die Knoten auf der oberen Linie sollen so umsortiert werden, dass möglichst wenige Kantenkreuzungen entstehen. Die Knoten auf der unteren Linie dürfen nicht umsortiert werden.

**Das EKMP ist NP-schwer**, es gibt also wahrscheinlich keinen effizienten exakten Algorithmus. **Kann man mit Maschinellem Lernen (ML) einen Algorithmus konstruieren, der in der Praxis gute Ergebnisse liefert?**

### Idee 1: Das EKMP als Spiel

Betrachte das Problem als ein Spiel, dessen Ziel es ist, eine Permutation zu finden, die minimal viele Kreuzungen verursacht. Ein Spielzug könnte darin bestehen, zu entscheiden, ob zwei benachbarte Knoten vertauscht werden sollen oder nicht.

- Welchen Teil vom Spielfeld sollte man bei dieser Entscheidung ansehen?
- Kann man eine KI entwickeln, die dieses Spiel gut spielt?

### Idee 2: Den Graphen auf „das Wichtigste“ reduzieren

Viele ML-Ansätze sind für Eingaben fester Größe ausgelegt (z.B. Erkennen von Katzen in Bildern der Größe  $512 \times 512$ ). Wir wollen aber beliebig große Graphen zeichnen. Ein möglicher Ansatz ist es zu versuchen, einen Graphen beliebiger Größe auf ein sogenanntes *embedding* konstanter Größe zu reduzieren, das „das Wichtigste“ des Graphen kodiert, und dann mit klassischen ML-Ansätzen weiterzuarbeiten.

- Was ist das „Wichtigste“ eines Graphen bzgl. des EKMPs?
- Helfen embeddings beim Lösen des EKMPs?
- Gibt es für das EKMP neue, bessere embeddings?



Silver et al. *A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play.*

<https://doi.org/10.1126/science.aar6404>



Bello et al. *Neural Combinatorial Optimization with Reinforcement Learning.*

<https://arxiv.org/abs/1611.09940>



Khalil et al. *Learning Combinatorial Optimization Algorithms over Graphs.*

**Was Du mitbringen solltest:** gute Programmierkenntnisse und Interesse am Thema!  
Vorkenntnisse zu Graphen und Maschinellem Lernen sind hilfreich, aber nicht notwendig.

(unverbindlich) **Interesse?** Schreibe mir eine E-Mail oder komme bei mir vorbei!

Guido Brückner

[brueckner@kit.edu](mailto:brueckner@kit.edu)

Infobau Raum 317