Übersicht von Algorithmenentwurfsprinzipien

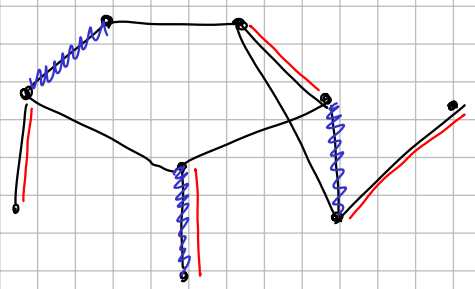
- Divide & Conquer: z.B. Mergesort, Dichtestes Punktepaar...
- Dynamische Programmierung: z.B. optimaler Suchbaum, TSP...
- Backtracking (mehr oder weniger Brute-Force): z.B. 8-Damen-Problem
- greedy Algorithmen: z.B. kürzeste Spannbäume: Algorithmus von Kruskal
  Maximum Matching (approximation)

Maximum Matching (Paarung)

gegg. $G=(V,E)$

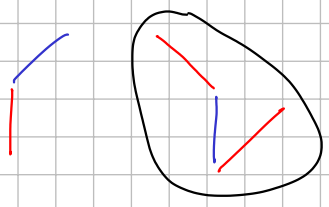
ges. $M \subseteq E$ mit $\forall e_1, e_2 \in M: e_1 \cap e_2 = \emptyset$ und $|M|$ so groß wie möglich

Bsp.:



maximal Matching (keine Kante kann hinzugefügt werden)

maximum Matching (Optimal)



greedy erzeugt immer ein maximal Matching (aber nicht unbedingt optimal)

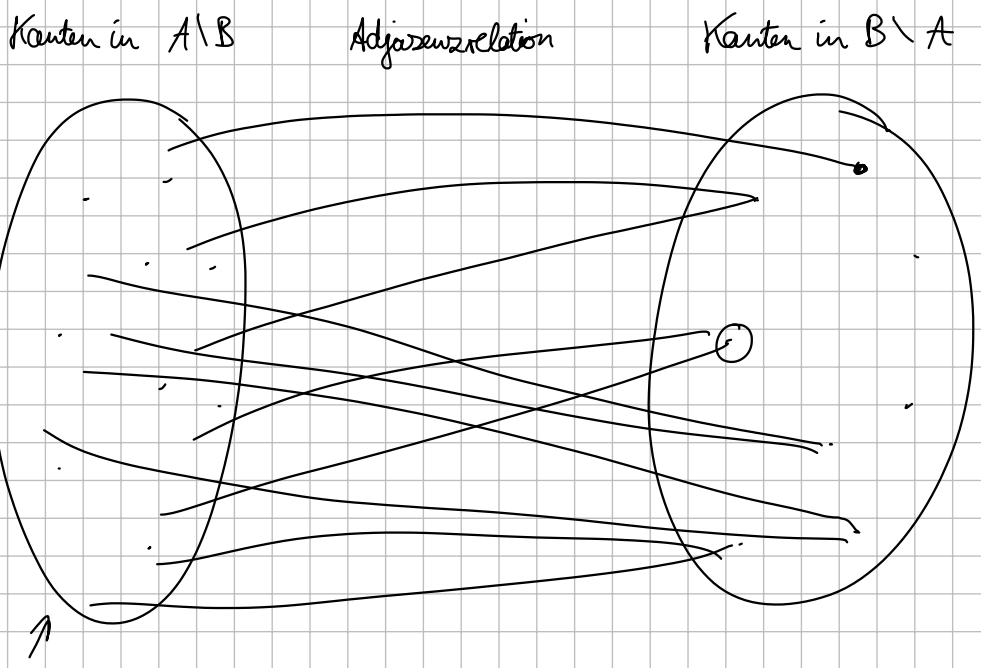
Approximationsalgorithmen: kein Algorithmus zur Berechnung einer Optimallösung verfügbar (z.B. NP-Schwer → zu hohe Laufzeit
in P aber Implementierung zu aufwändig / Konstanten zu hoch
nicht bekannt)

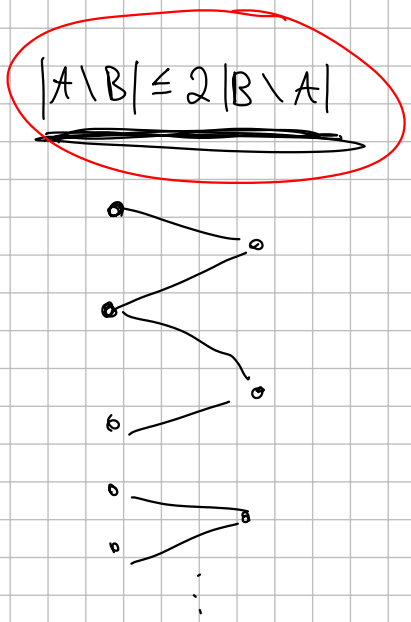
→ Entwicklung eines Algorithmus, der eine bestimmte Gütegarantie erfüllt (z.B. $\geq 0.5 OPT$)

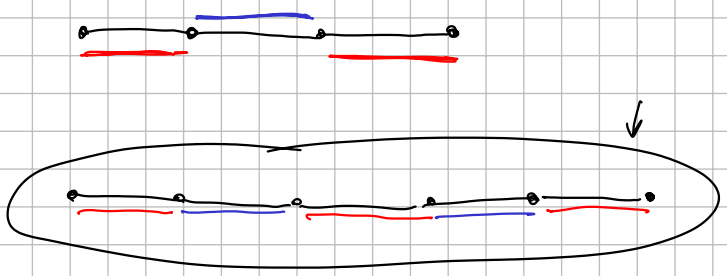
Greedy hat eine Garantie von $\geq 0.5 OPT$ für maximum Matching

Beweis: Betrachte zwei maximal Matchings A und B
Betrachte ferner $A \setminus B$ und $B \setminus A$
- Da A eine Paarung ist, kann jede Kante aus $B \setminus A$ zu höchstens 2 Kanten aus $A \setminus B$ sein.

- Da B ein maximal Matching ist, muss jede Kante aus $A \setminus B$ mind zu einer Kante aus $B \setminus A$ adjasent sein (sonst könnte man diese Kante zu B hinzufügen)







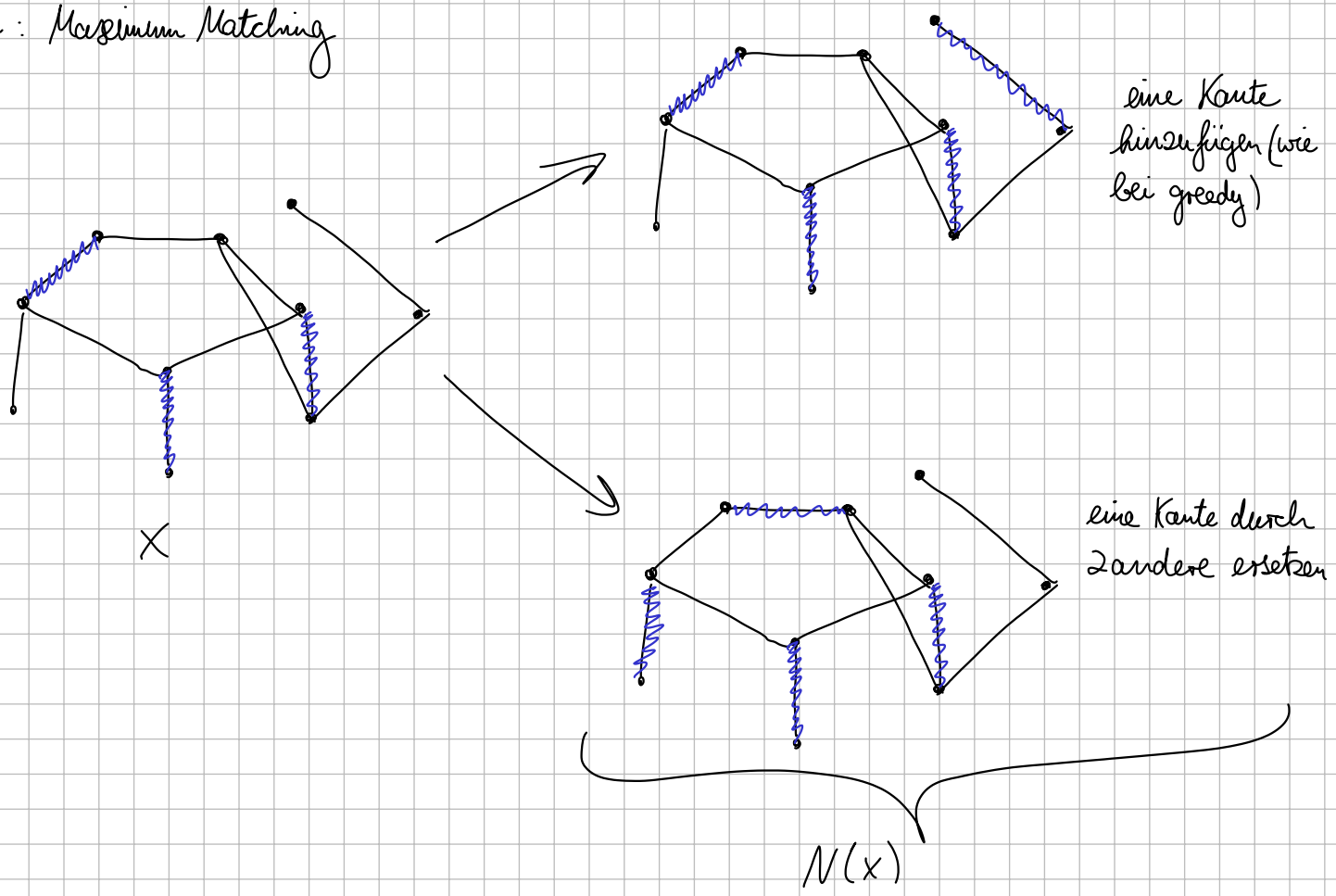
Da $|A \setminus B| \leq 2|B \setminus A|$ ist

$$\underbrace{|A|}_{\text{OPT}} = |A \cap B| + |A \setminus B| \leq 2|A \cap B| + 2|B \setminus A| = 2\big(|B \cap A| + |B \setminus A|\big) = \underbrace{2|B|}_{\text{greedy}}$$

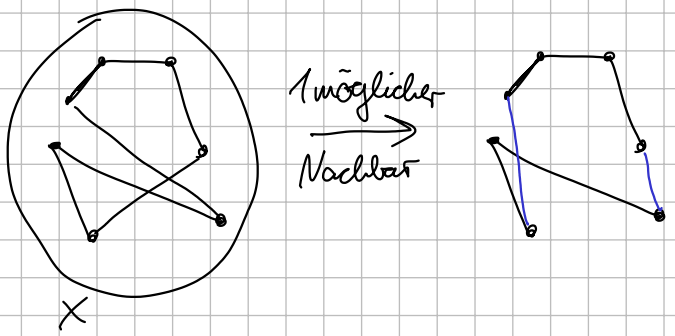
Heuristiken: keine Gütegarantie (im ggs zu Approximationsalgorithmen)

- lokale Optimierung: Versuche Lösung durch „kleine Änderung" zu Verbessern.
  Nachbarschaft: Zu jeder Lösung x ist eine Menge $N(x)$ von „benachbarten" Lösungen definiert

Bsp.: Maximum Matching



Bsp. 2 TSP



$\Rightarrow |N(x)| = \binom{n}{2}$ (alle möglichkeiten für kantenpaare)

Lokale Optimierung

- Menge S möglicher Lösungen
- Zielfunktion $c: S \to \mathbb{R}$

  Aufgabe: Finde x sodass c(x) so groß/klein wie möglich ist
- definiere dafür Nachbarschaft $N(x) \subseteq S$ für alle $x \in S$

Algorithmus:

fange mit einer Startlösung $x \leftarrow x_0 \in S$

Schleife:

falls $x' \in N(x)$ mit $c(x') \lesseqgtr c(x)$ existiert

$x \leftarrow x'$

sonst Abbruch und Ausgabe

Falls S endlich, dann terminiert Algorithmus, und zwar in einem lokalen Optimum.

Def.: ein lokales Optimum $x: \forall x' \in N(x): c(x') \gtreqless c(x)$

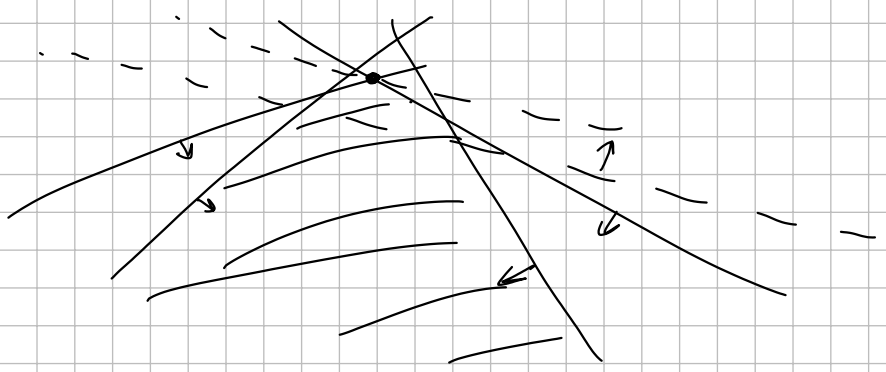
muss nicht ein globales Optimum sein!

[bei konvexen Zielfunktionen $c: \mathbb{R}^n \to \mathbb{R}$ sind lokale optima auch globale optima]

z.B.: lineare Programme: geg.: $A \in \mathbb{R}^{n \times n}$, $c \in \mathbb{R}^n$, $b \in \mathbb{R}^n$

$\min c^T x$

s.t $Ax \leq b$



Verbesserungen sind durch

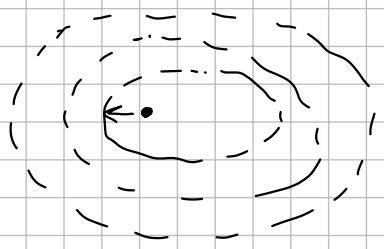
- bessere Wahl von x_0 (bzw. mehrere Durchläufe mit zufälligen x_0) und/oder
- vergrößerung der Nachbarschaft (führt zu weniger lokale optima)

möglich. Beide Verbesserungen verschlechtern Laufzeit

Je nach Problem muss man Güte vs. Laufzeit abwägen.

Variante: Steilster Abstieg (Gradientenverfahren)

wähle $x' \in N(x)$ mit kleinstem $c(x')$ (falls $c(x') < c(x)$)



Laufzeit/je nach feinheit/Problem
Güte

Simulated Annealing

- Akzeptiere Verschlechterungen der Zielfunktion mit gewisser Wsk. (hängt von "Temperatur" des Systems ab)

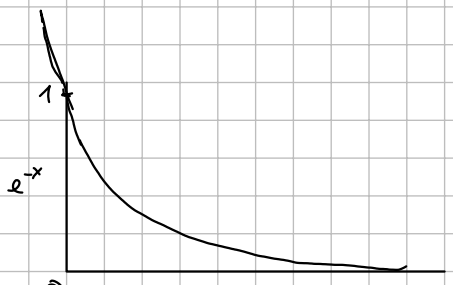
Schleife:

Wähle $x' \in N(x)$ aus

$\Delta c \leftarrow c(x') - c(x)$

Wenn $\Delta c > 0$, $x \leftarrow x'$

Sonst: $x \leftarrow x'$ mit Wsk $e^{-\frac{|\Delta c|}{T}}$



$\times$

Δc groß $\Rightarrow$ geringe Wsk

$\Delta c \to 0 \Rightarrow$ Wsk $\to 1$

T groß ($T \to \infty$): Wsk $\to 1$ (mehr Nachbarn werden akzeptiert)

T klein ($T \to 0$): Wsk $\to 0$ (keine Verschlechterungen werden akzeptiert $\Rightarrow$ lokale Optimierung)

[festes T: Verteilung strebt gegen die Boltzmannverteilung:
Wsk, dass Ausgabe $x \in S$ ist, ist proportional zu $e^{-\frac{c(x)}{T}}$]