

Kap. 8: Travelling Salesman Problem

Professor Dr. Petra Mutzel

Lehrstuhl für Algorithm Engineering, LS11

15. VO

5.2.07

Überblick

- 8.1 Einführung
 - Einführung in TSP
- 8.2 ILP-Formulierung für TSP
- 8.3 Branch-and-Cut Algorithmus
 - Separierung der Subtour Bedingungen
 - Exkurs: Polyedertheorie
 - Zusätzliche Ungleichungen
 - Branch & Cut
 - Spaltengenerierung

Literatur

- TSP-Web Site: www.tsp.gatech.edu
- M. Jünger, G. Reinelt und G. Rinaldi: The Traveling Salesman Problem, 1995, Elsevier Science in the collection "Network Models, Handbook on Operations Research and Management Science" (Eds. M. Ball, T. Magnanti, C. Monma, G. Nemhauser), vol. 7, 225-230.
- NEUES BUCH: D. Applegate, R.E. Bixby, V. Chvátal und W.J. Cook: **The Traveling Salesman Problem: A Computational Study**, [Princeton University Press](http://www.princeton.edu/~applegate/) 2006, 606 pp.
- Nachschlagewerk bei Interesse: M. Jünger und D. Naddef (Eds.): Computational Combinatorial Optimization, Optimal or Provably Near-Optimal Solutions, LNCS 2241, Springer, 2001, i.e. 157-223

8.1 Einführung: Das Travelling Salesman Problem

Das Handlungsreisendenproblem, TSP

- Gegeben: Vollständiger ungerichteter Graph $G=(V,E)$ mit Kantenkosten c_e
- Gesucht: Tour T (Kreis, der jeden Knoten genau einmal enthält) mit minimalen Kosten $c(T)=\sum c_e$

Buch von F. Voigt, Ilmenau 1831

Der Handlungsreisende, wie er sein soll und was er zu thun hat, um Aufträge zu erhalten und eines glücklichen Erfolgs in seinen Geschäften gewiß zu sein. Von einem alten Commis=Voyageur.

„... Durch geeignete Auswahl und Planung der Tour kann man oft so viel Zeit sparen, daß wir einige Vorschläge zu machen haben. ... Der Wichtigste Aspekt ist, so viele Orte wie möglich zu erreichen, ohne einen Ort zweimal zu besuchen. ... “

Das Travelling Salesman Problem

Ann.: Rechner schafft 40 Mio. Touren pro Sekunde

Anzahl der verschiedenen Touren: $(|V|-1)! / 2$

n	#Tour	Zeit
10	181440	0.0045 Sek.
17	ca. 10^{13}	3 Tage
19	ca. 10^{15}	2,5 Jahre
20	ca. 10^{17}	48 Jahre
25	ca. 10^{23}	10^8 Jahre
60	ca. 10^{80}	10^{64} Jahre

Anzahl der Atome im Weltall: ca. 10^{80}

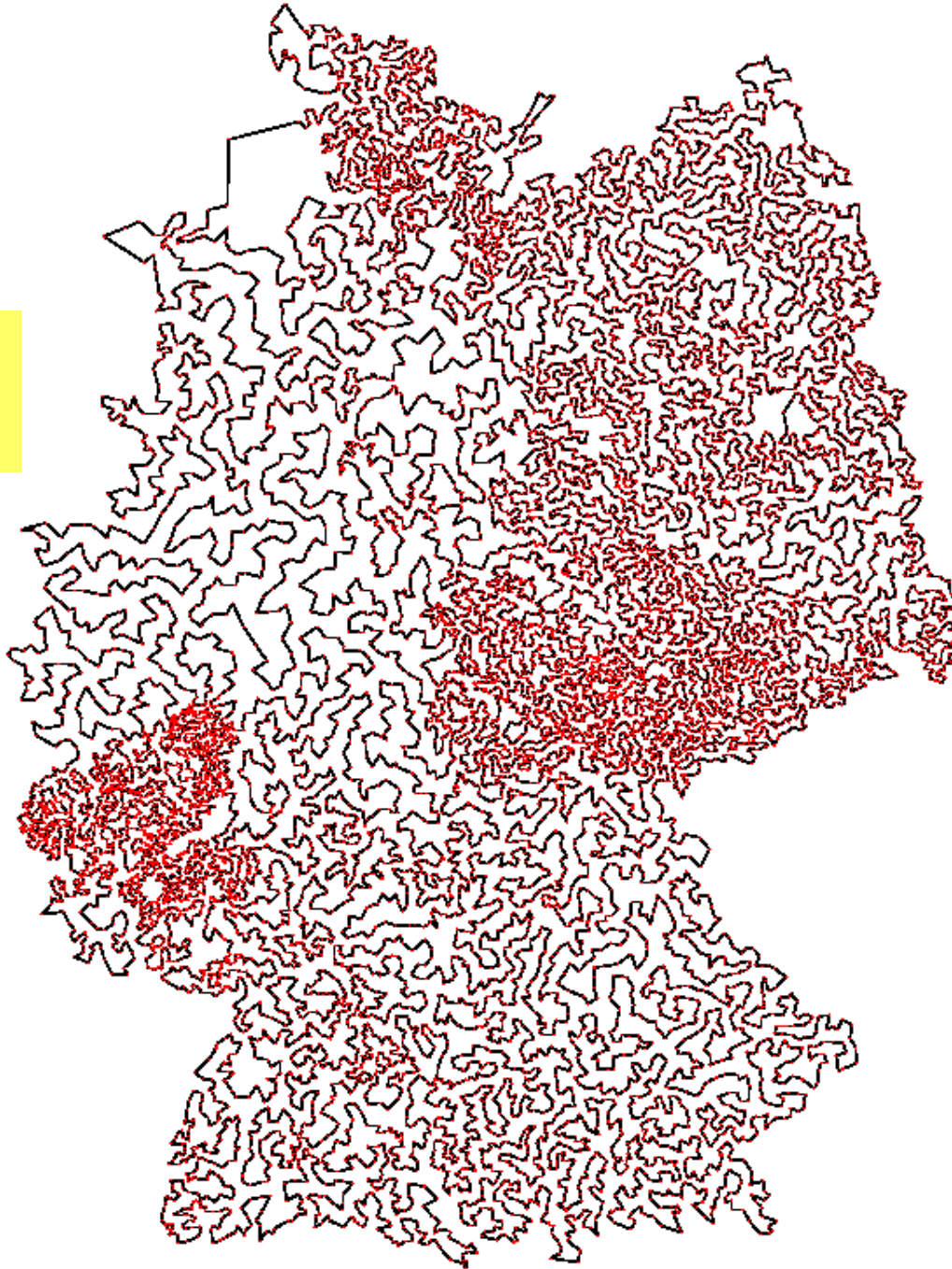
Meilensteine für TSP Lösungen

Jahr	Weltmeister	Städte
1954	Dantzig, Fulkerson, Johnson	49
1971	Held, Karp	64
1977	Grötschel	120
1980	Crowder, Padberg	318
1987	Padberg, Rinaldi	532
1987	Grötschel, Holland	666
1987	Padberg, Rinaldi	2392
1994	Applegate, Bixby, Chvátal, Cook	7397
1998	Applegate, Bixby, Chvátal, Cook	13509
2001	Applegate, Bixby, Chvátal, Cook	15112
2004	Applegate, Bixby, Chvátal, Cook, Helsgaun	24978

aktuelle Challenge: Worldtour 1.904.711 Städte

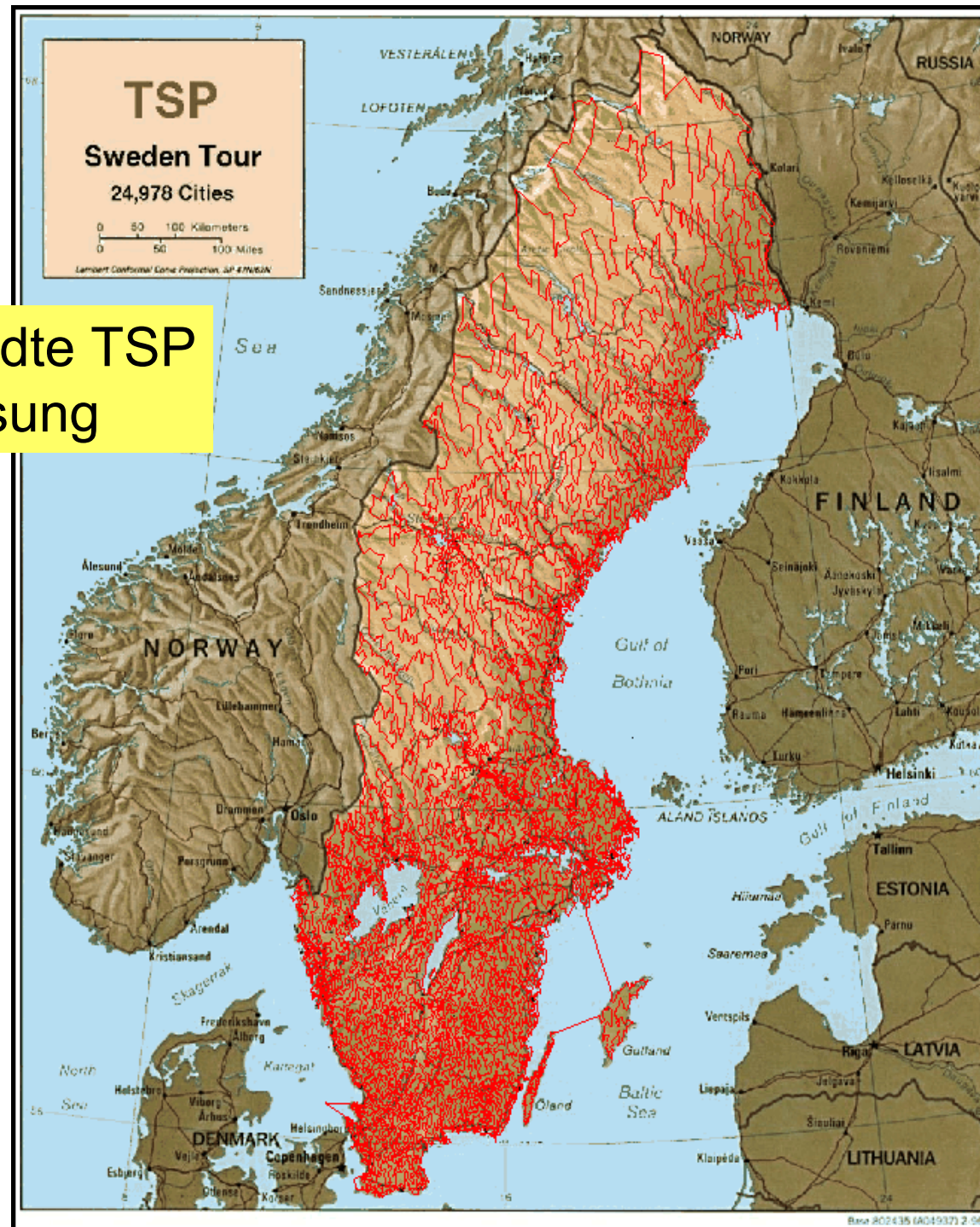
15112 Städte TSP

22,6 CPU Jahre
auf 110 Prozessoren

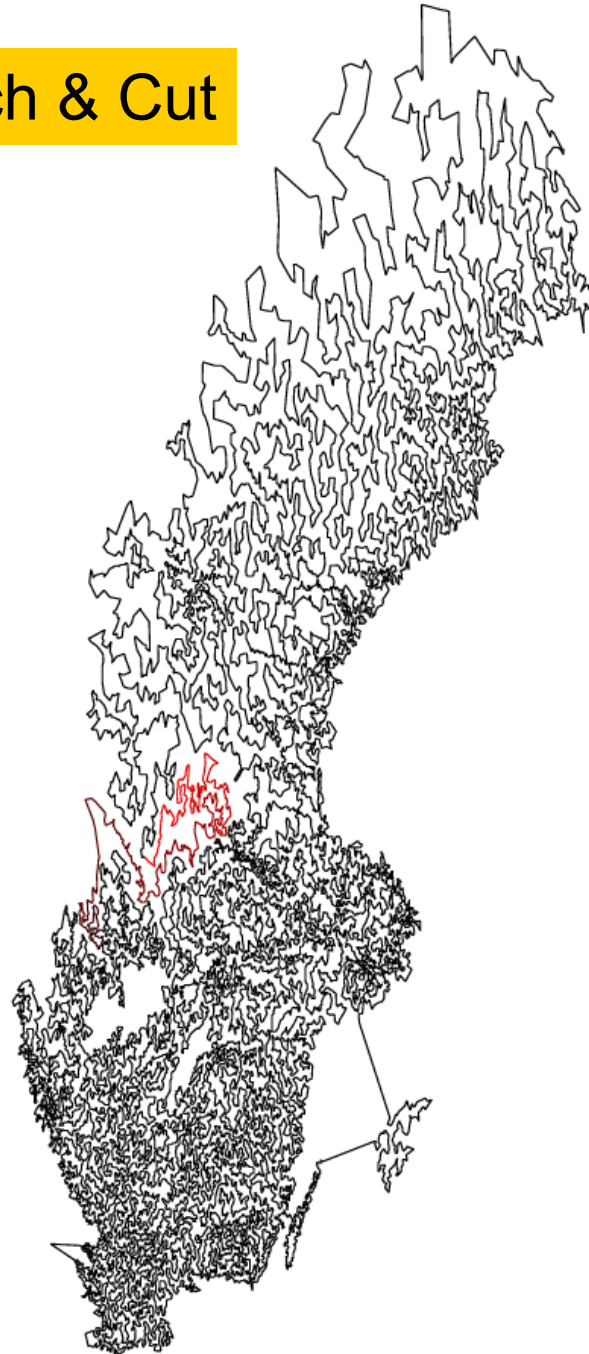


24978 Städte TSP
Optimallösung

8 Jahre



Lösung mittels Branch & Cut

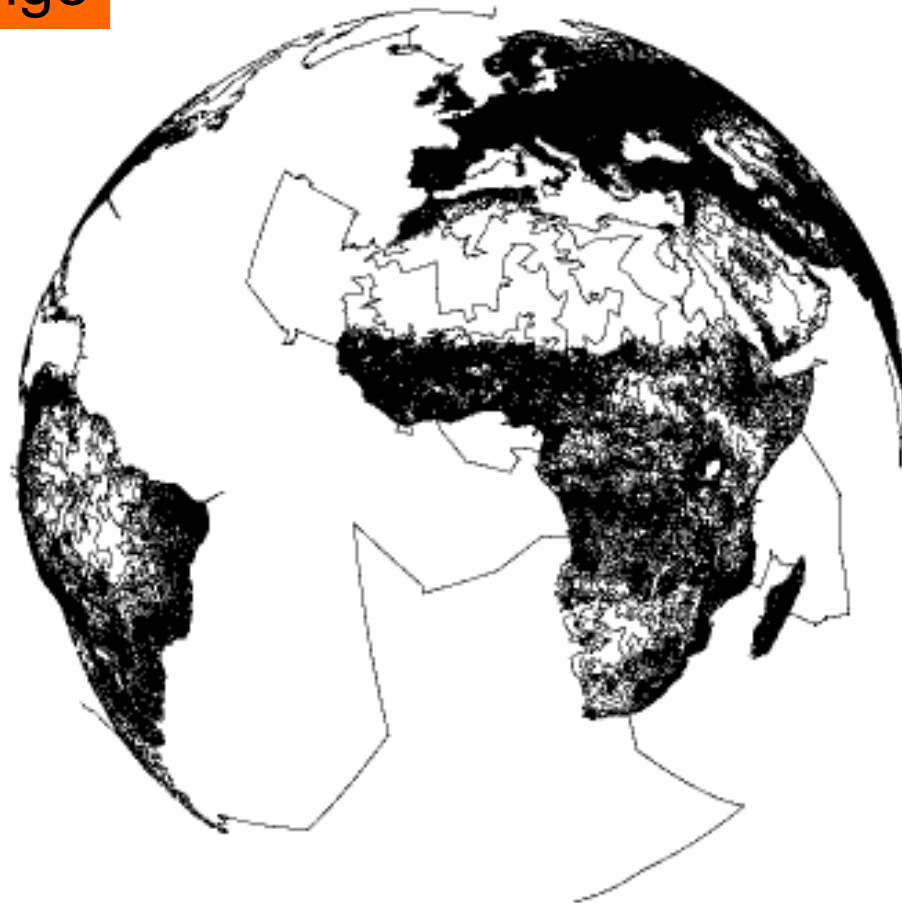


World Tour: 1.904.711 Städte

Tour der Länge **7,516,353,779** ([Keld Helsgaun](#), 2003)

Garantie: Tourlänge **0.076%** nahe an OPT(Concorde TSP code)

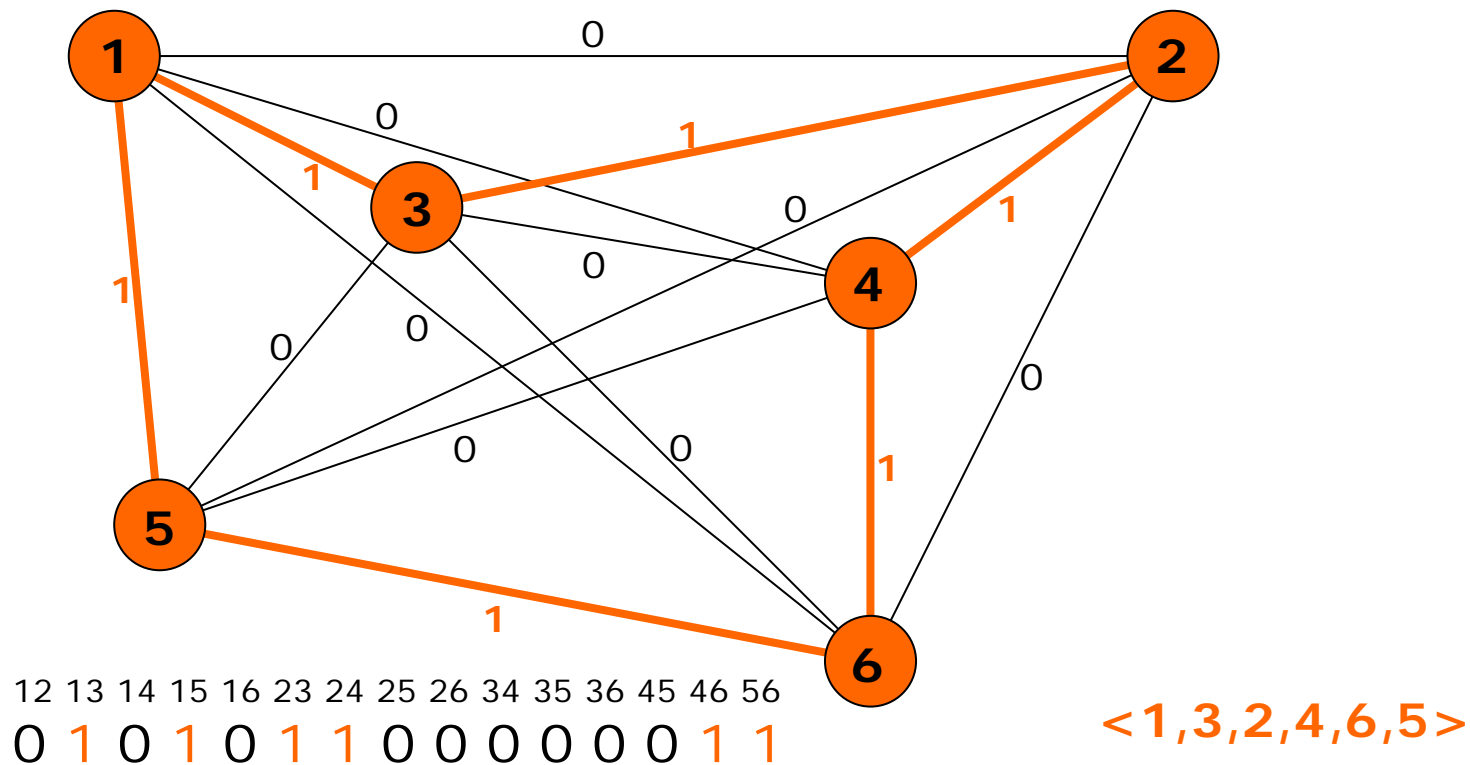
aktuelle Challenge



8.2 ILP-Formulierung für TSP

Binär-Variable: $x_{uv}=1$, wenn Kante uv in Tour T , und $x_{uv}=0$ sonst

Beispiel:



ILP-Formulierung für TSP

Geg.: vollständiger Graph $G=(V,E)$ mit c_e für alle Kanten e

$$\min \quad c^T x \quad (1)$$

$$\sum_{e \in \delta(v)} x_e = 2 \quad \forall v \in V \quad \text{Grad-Gl.} \quad (2)$$

$$\sum_{e \in \delta(S)} x_e \geq 2 \quad \forall S \subset V \quad \text{Subtour-Ungl.} \quad (3)$$

$$0 \leq x_e \leq 1 \quad \forall e \in E \quad (4)$$

$$x_e \in \{0, 1\} \quad \forall e \in E \quad (5)$$

Problem: Ganzzahlige Optimierung ist NP-schwer

→ Löst eine Relaxierung des Problems

Branch & Cut für TSP

LP-Relaxierung: Entferne Ganzzahligkeitsbedingungen

$$\begin{array}{ll} \text{minimiere} & \sum_{e \in E} c_e x_e \\ \text{so dass} & 0 \leq x_e \leq 1 \quad \text{für alle } e \in E \\ & \text{ ~~} x_e \text{ ganzzahlig für alle } e \in E \text{~~ } \\ & \sum_{e \in \delta(v)} x_e = 2 \quad \text{für alle } v \in V \\ & \sum_{e \in \delta(W)} x_e \geq 2 \quad \text{für alle } \emptyset \neq W \subset V \end{array}$$

Subtour-Relaxierung: man relaxiert Ganzzahligkeit

Branch & Cut für TSP

Auch die Subtour-Relaxierung hat bereits
exponentiell viele ($\Theta(2^{n-1})$) Ungleichungen

- hoffnungslos?
- Lösung: Schnittebenenverfahren

Können wir die Subtour-Relaxierung beim TSP lösen?

Satz von Grötschel, Lovasz, Schrijver

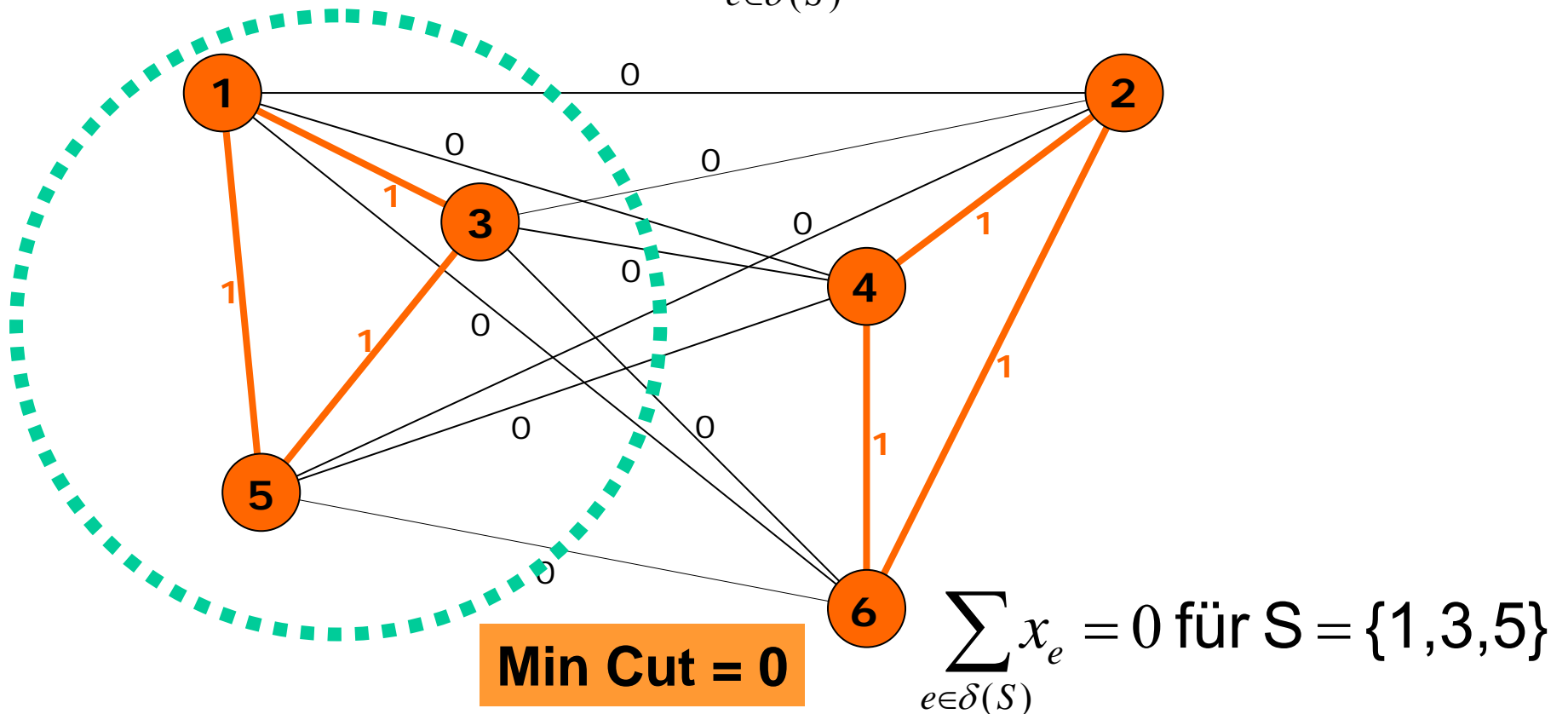
Ein Optimierungsproblem ist in polynomieller Zeit lösbar genau dann wenn das zugehörige Separationsproblem in polynomieller Zeit lösbar ist.

Frage: Können wir Separationsproblem für die Subtour-Ungleichungen des TSP lösen?

Lösung des kleinsten Schnittproblems

Separierung TSP

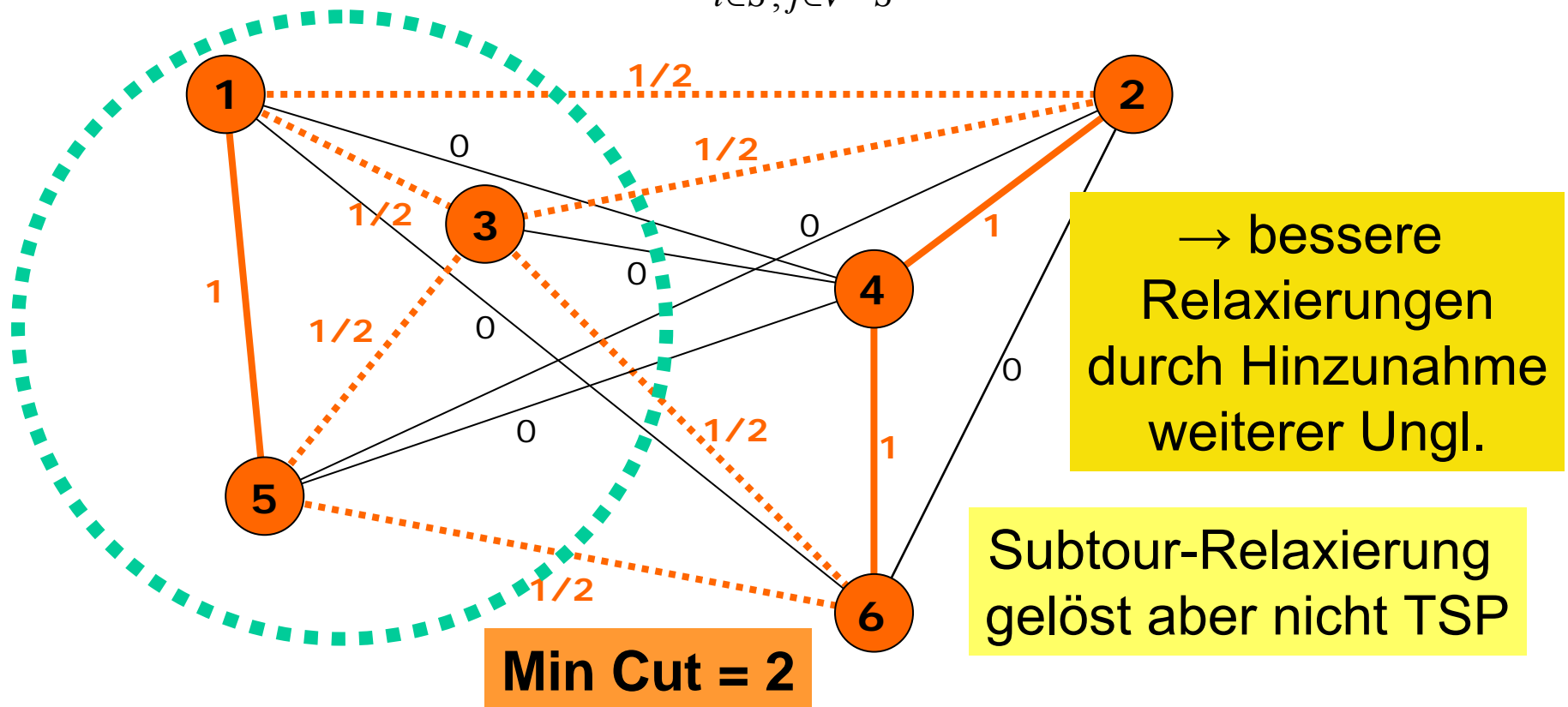
- Subtour Bedingungen $\sum_{e \in \delta(S)} x_e \geq 2$ für alle $\emptyset \neq S \subset V$



Verletzte Subtour-Ungleichung gefunden

Separierung TSP

- Subtour Bedingungen $\sum_{i \in S, j \in V-S} x_{ij} \geq 2$ für alle $\emptyset \neq S \subset V$



Es existiert keine verletzte Subtour-Ungleichung

Bekannte Ungleichungen beim TSP

- Grad Gleichungen
- Subtour Bedingungen
- 2-Matching Ungleichungen
- Blossom Ungleichungen
- Comb Ungleichungen
- Cliquenbaum Ungleichungen
- Hypergraph Ungleichungen
- ...



Separierungs-
problem
polynomiell

Separierungs-
problem
offen

Wie kommt man auf diese Ungleichungen: Polyedertheorie

oder: Rechnen und dann Studium von nicht-zulässigen Lösungen

Exkurs: Polyedertheorie (Kurzeinführung)

Satz [Minkowski 1896, Weyl 1935]: Jedes Polyeder $P \in R^n$ besitzt eine Darstellung der Form

$$P = \text{conv}(V) + \text{cone}(E),$$

wobei V und E endlichen Teilmengen des R^n entsprechen und umgekehrt.

Deshalb existieren immer zwei Darstellungen von Polyedern

$$P = \{x \in R^n \mid Ax \leq b\} = \text{conv}(V) + \text{cone}(E)$$

Polytope lassen sich durch $P = \text{conv}(V)$ darstellen.

Definition Kombinatorisches Optimierungsproblem

Gegeben sind:

- endliche Menge E (Grundmenge)
- Teilmenge I der Potenzmenge 2^E von E (zulässige Mengen)
- Kostenfunktion $c: E \rightarrow K$

Gesucht ist:

eine Menge $I^* \in I$, so dass

$c(I^*) = \sum_{e \in I^*} c(e)$ so groß (klein) wie möglich ist.

Zusammenhang zu Kombinatorischer Optimierung

Jedes k. OP kann als 0/1 LP formuliert werden und vice versa

Ist E eine endliche Menge und $F \subseteq E$, dann ist der charakteristische Vektor $\chi^F \in \mathbb{R}^E$ für F definiert als

$$\begin{aligned}\chi_e^F &= 1 \quad \Leftrightarrow \quad e \in F \\ \chi_e^F &= 0 \quad \Leftrightarrow \quad e \notin F\end{aligned}$$

Wir assoziieren zu jedem Element $e \in E$ eine Komponente des Vektors χ^F .

Umgekehrt, ist jeder 0/1-Vektor $x \in \{0,1\}^E$ charakteristischer Vektor einer Teilmenge F_x von E , und zwar gilt:

$$F_x = \{e \in E \mid x_e = 1\}.$$

Kombinatorische Optimierung vs. 0/1-IP

Gegeben ist kombinatorisches OP: (E, I, c)

Assoziiertes 0/1-IP:

$$P_F = \text{conv}\{\chi^F \in \{0, 1\}^E \mid F \in I\}$$

$$\max\{c^T x \mid x \in P_F\}$$

Jedes Polyeder hat Beschreibung durch Ungleichungen

Wir können also jedes komb. OP als LP formulieren

Probleme:

- Berechnung der LP-Darstellung nicht in pol.- Zeit möglich
- i.A. exponentiell viele Ungleichungen
- Ungleichungen besitzen Koeffizienten exponentieller Größe

Branch & Cut

- Identifiziere die zulässigen Lösungen mit 0/1-Vektoren
- Beschreibe die konvexe Hülle dieser Vektoren möglichst gut durch lineare Ungleichungen

Branch & Cut für TSP

Vollständige lineare Beschreibungen sind sehr gross

Städte	Touren	Gleichungen	Ungleichungen
3	1	3	0
4	3	4	3
5	12	5	20
6	60	6	100
7	360	7	3.437
8	2.520	8	194.187
9	20.160	9	42.104.442
10	181.440	10	51.043.900.866

Praxis: ILP Formulierungen

Branch-and-Cut Verfahren

Verbindung von Schnittebenenverfahren
mit Branch-and-Bound





Versuche, jeweils die Teilprobleme (LP-Relaxierungen)
mittels Schnittebenenverfahren zu lösen

Falls die Lösung nicht ganzzahlig ist, dann wähle
nicht-ganzzahlige Variable und generiere zwei neue
Teilprobleme:

P1 mit zusätzlichen Restriktionen $x_e=0$

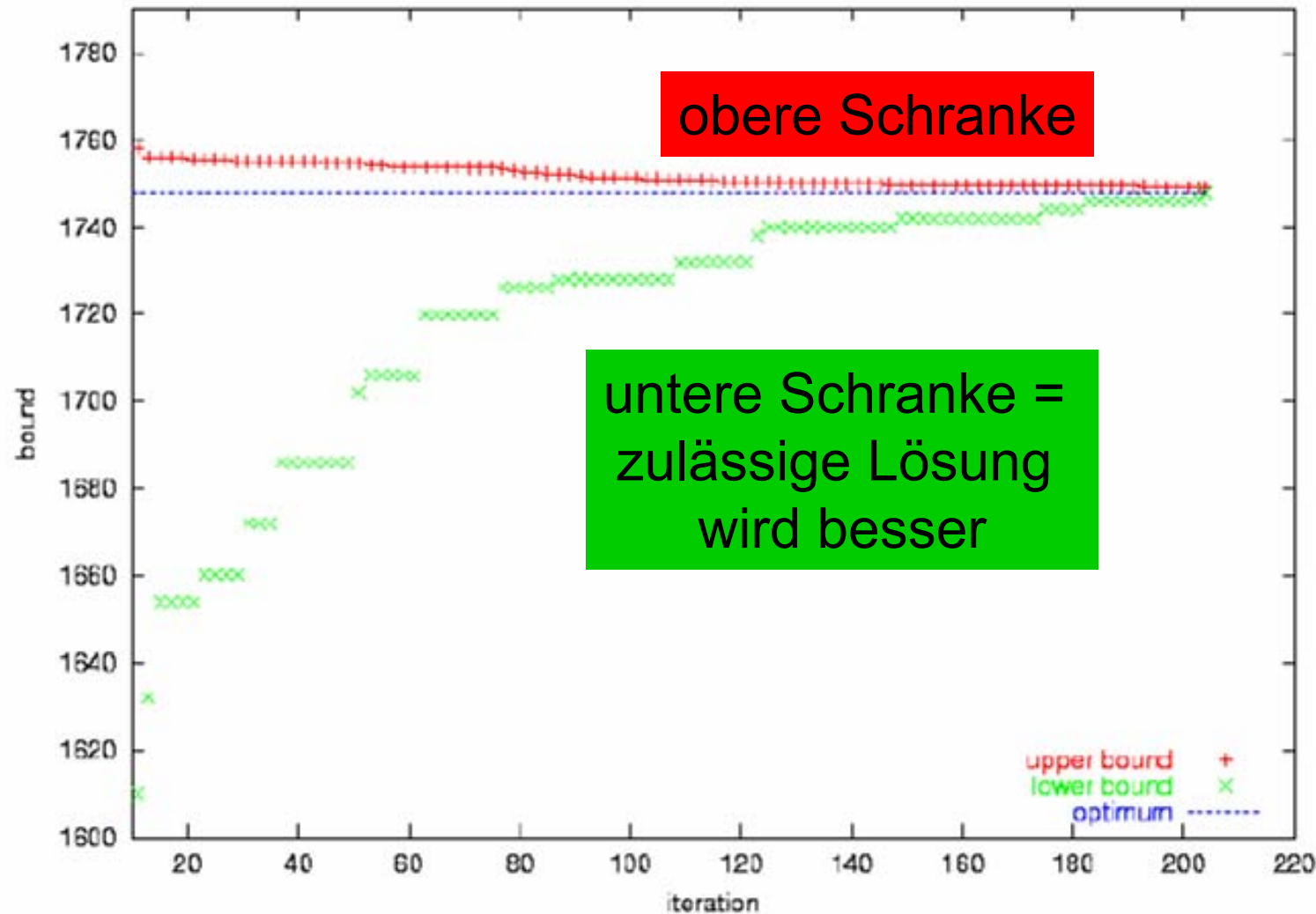
P2 mit zusätzlichen Restriktionen $x_e=1$

Branch-and-Cut (Grundidee)

- Löse Relaxierungen mit Schnittebenenverfahren.  **Cut**
- Verzweige, wenn keine Schnittebenen gefunden werden, aber die Lösung nicht ganzzahlig ist.  **Branch**
- Finde gute zulässige Lösungen (hier: Touren) basierend auf fraktionellen Lösungen.  **Exploit**
- Schneide nichtprofitable Teilbäume weg.  **Bound**

Eine typische B&C Berechnung

Maximierungsproblem



Bekannte Ungleichungen beim TSP

- Grad Gleichungen
- Subtour Bedingungen
- 2-Matching Ungleichungen
- Blossom Ungleichungen
- Comb Ungleichungen
- Cliquenbaum Ungleichungen
- Hypergraph Ungleichungen
- ...



Separierungs-
problem
polynomiell

Separierungs-
problem
offen

Wie kommt man auf diese Ungleichungen: Polyedertheorie

oder: Rechnen und dann Studium von nicht-zulässigen Lösungen

Comb-Ungleichungen

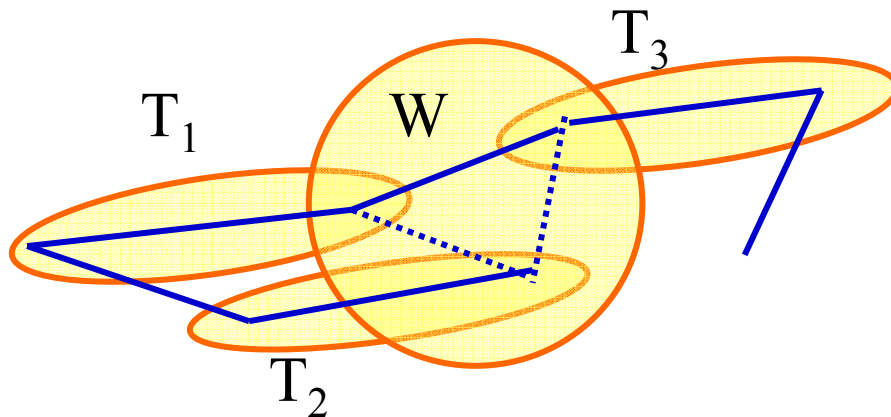
- Sei $S = \{W, T_1, \dots, T_k\}$, wobei W der “Griff” (handle) und T_i die “Zinken” (teeth) sind. Es muss gelten:

$$|T_i \cap W| \geq 1 \quad \text{fuer } i = 1, \dots, k, \quad (1)$$

$$|T_i \setminus W| \geq 1 \quad \text{fuer } i = 1, \dots, k \quad (2)$$

$$T_i \cap T_j = \emptyset \quad \text{fuer } 1 \leq i < j \leq k \quad (3)$$

$$k \geq 3 \quad \text{und ungerade.} \quad (4)$$



- “2-Matching Ungleichungen”, wenn für alle i gilt: $|T_i| = 2$

Comb-Ungleichungen

- Sei $S = \{W, T_1, \dots, T_k\}$, wobei W der “Griff” (handle) und T_i die “Zinken” (teeth) sind. Es muss gelten:

$$|T_i \cap W| \geq 1 \quad \text{fuer } i = 1, \dots, k, \quad (1)$$

$$|T_i \setminus W| \geq 1 \quad \text{fuer } i = 1, \dots, k \quad (2)$$

$$T_i \cap T_j = \emptyset \quad \text{fuer } 1 \leq i < j \leq k \quad (3)$$

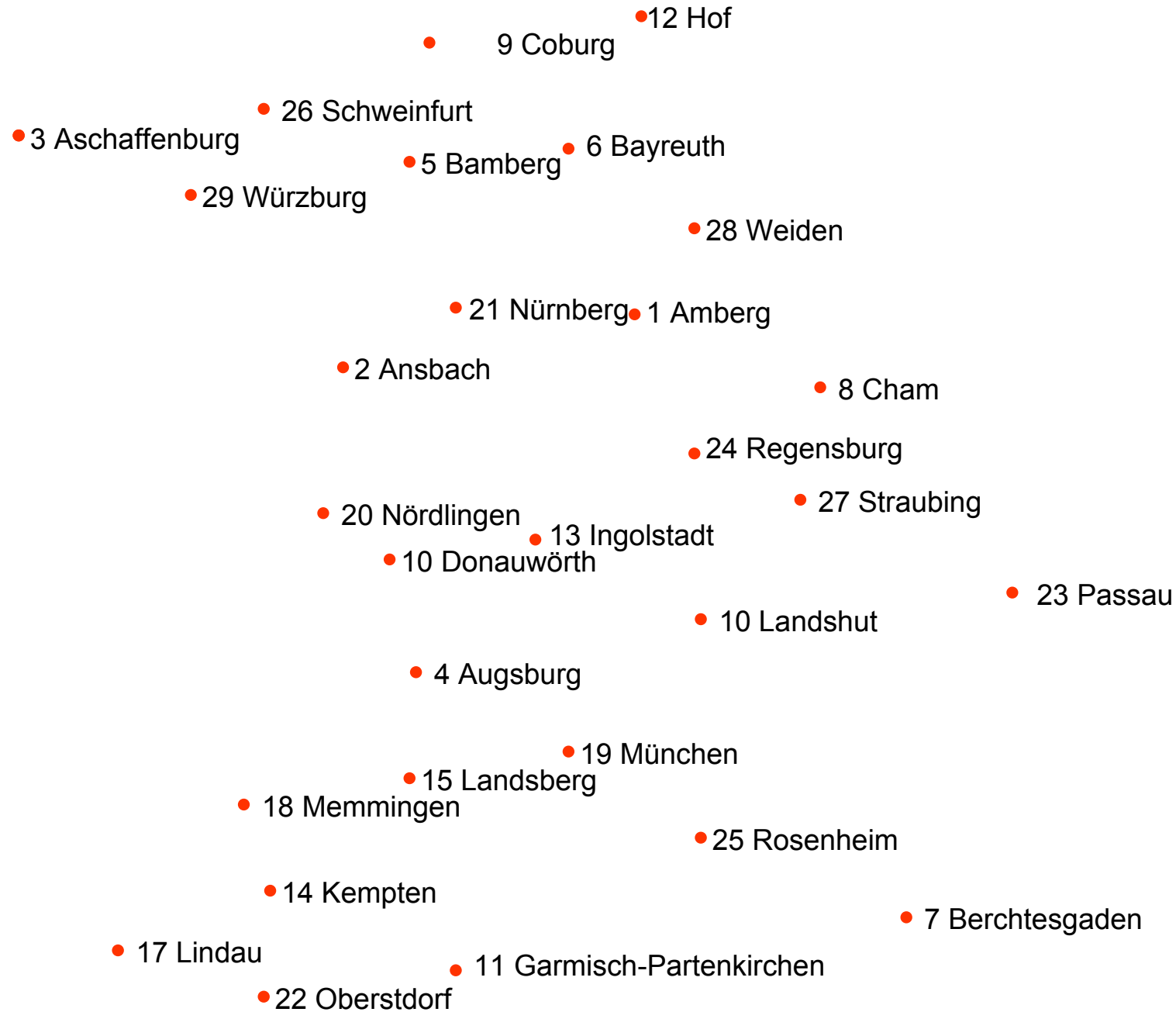
$$k \geq 3 \quad \text{und ungerade.} \quad (4)$$

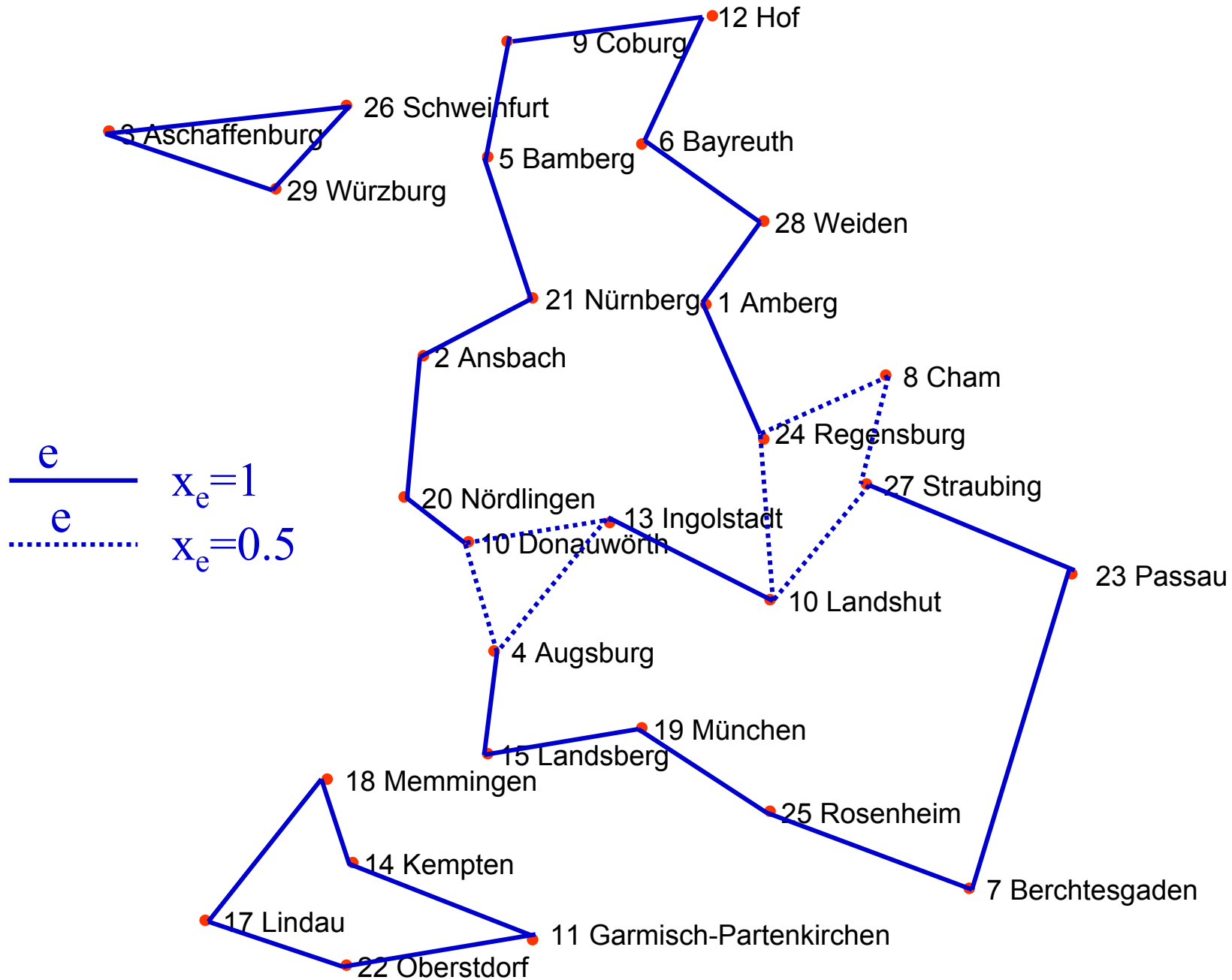
$$x(E(W)) + \sum_{i=1}^k x(E(T_i)) \leq |W| + \frac{k-1}{2}$$

- “2-Matching Ungleichungen”, wenn für alle i gilt: $|T_i|=2$

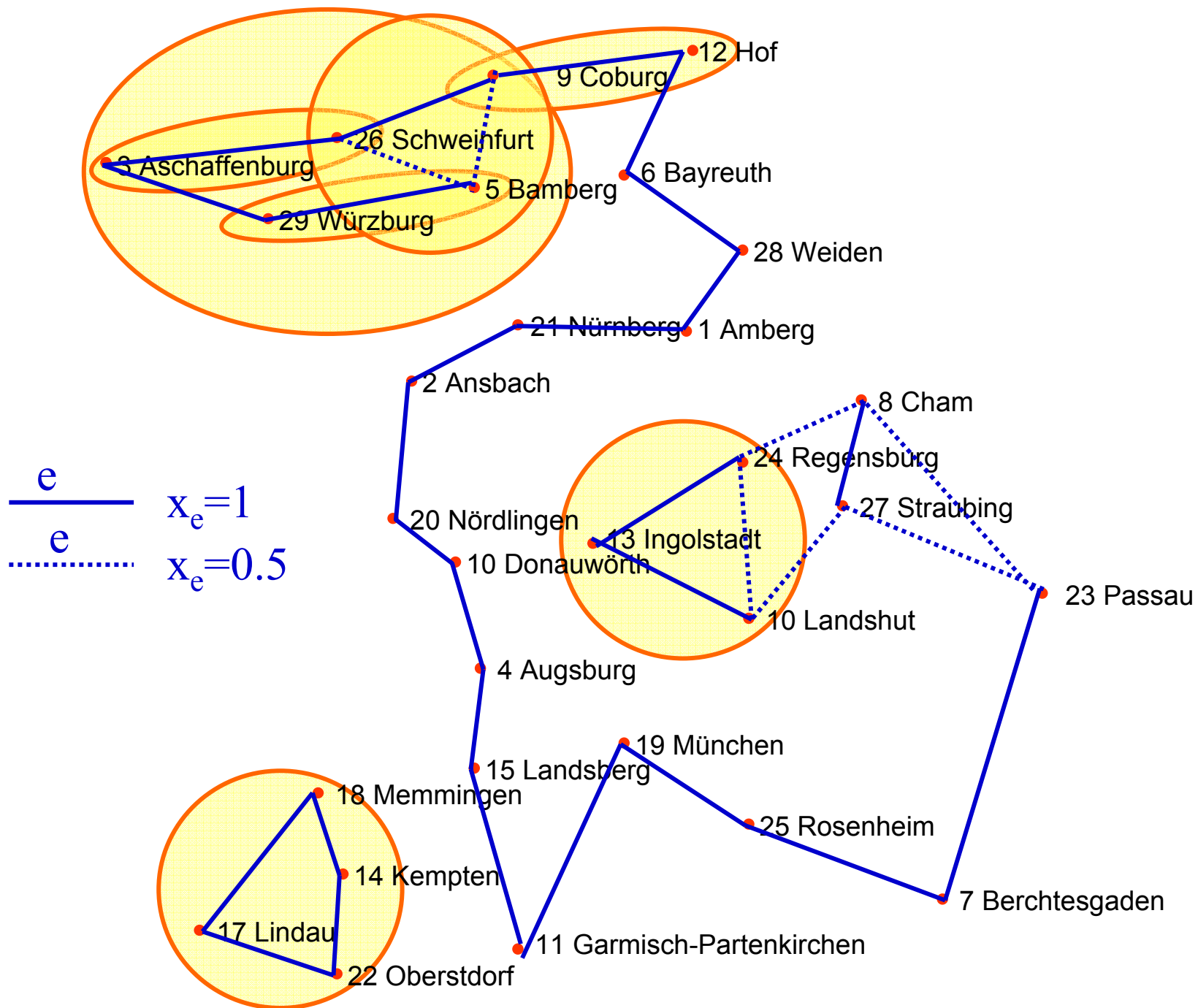
Beispiel: Bayern

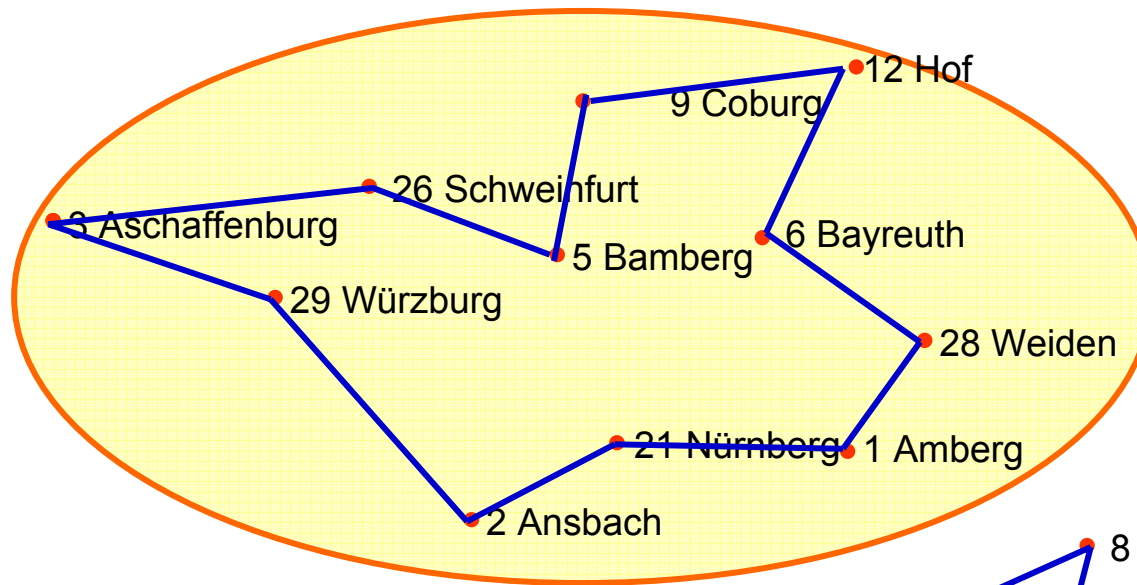
- 29 Städte in Bayern
- TSPLIB/bayg29.tsp



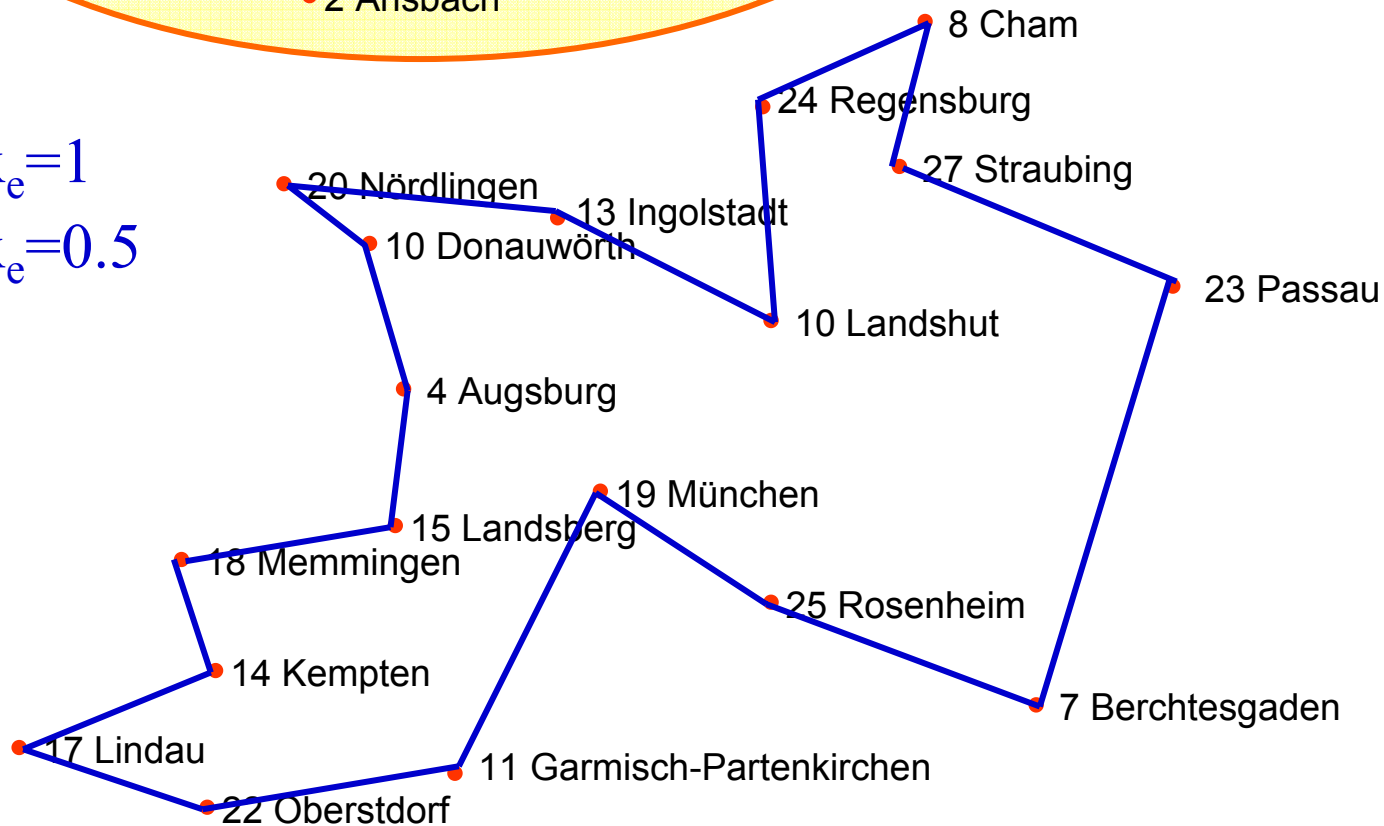


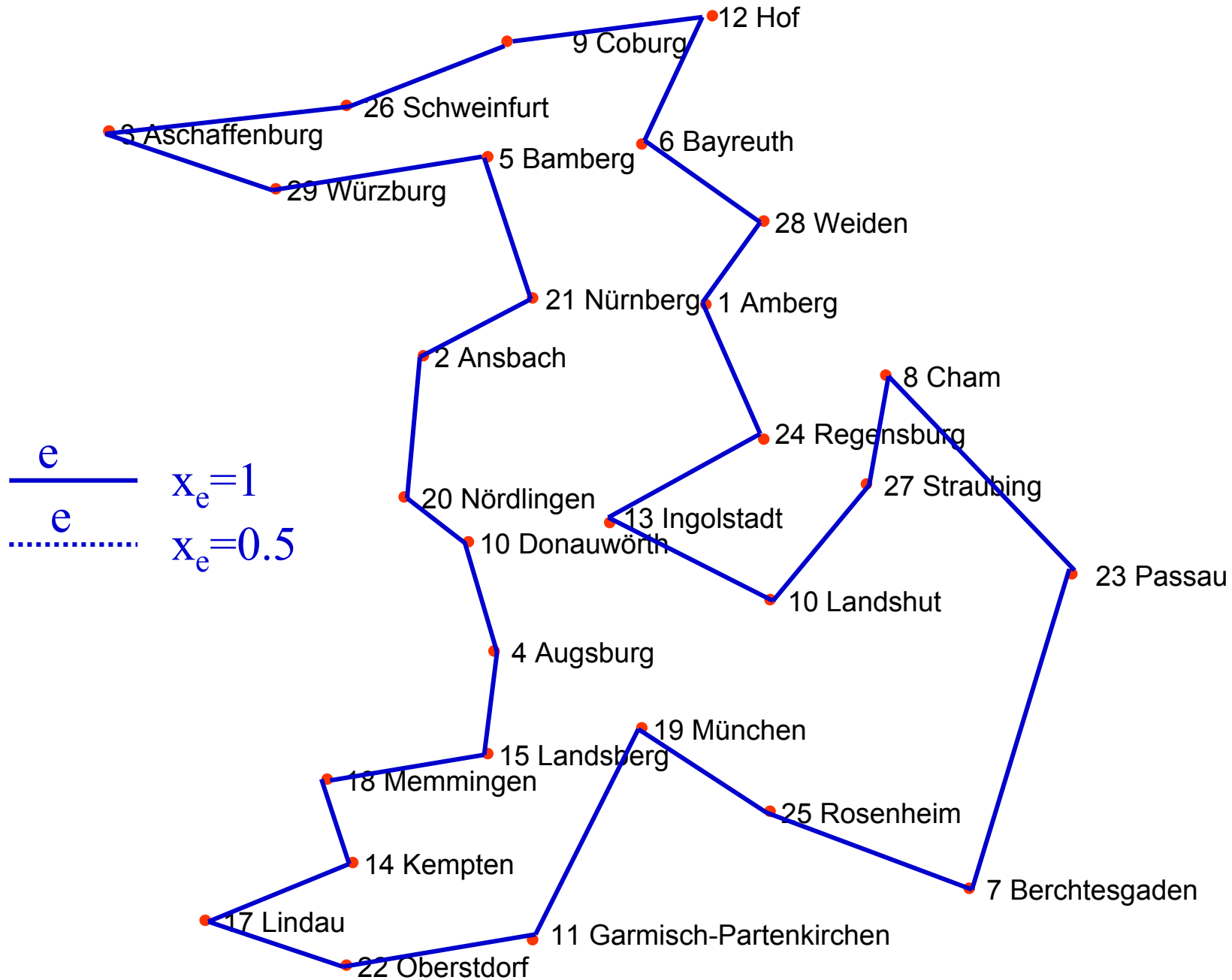






$\frac{e}{e} \quad x_e = 1$
 $\frac{e}{e} \quad x_e = 0.5$





Spaltengenerierung für TSP

Idee:

1. Starte nicht mit allen Variablen, sondern nur mit einer Teilmenge $E' \subseteq E$ (dies entspricht einer Teilmenge der Spalten des ILPs).
2. Bewährt hat sich $\text{Sparse Graph} \subseteq \text{Reserve Graph} \subseteq \text{ganz } G$.
3. z.B. als Sparse Graph den 5-Nächsten Nachbar Graphen, als Reserve Graph den 10-Nächsten Nachbarn
4. Löse die LP-Relaxierung LPRel bzgl. E'
5. Nun muss man testen, ob diese Lösung auch eine gültige Lösung der LP-Relaxierung bzgl. E ist.

Spaltengenerierung für TSP

Idee:

1. Starte nicht mit allen Variablen, sondern nur mit einer Teilmenge $E' \subseteq E$ (dies entspricht einer Teilmenge der Spalten des ILPs).
2. Löse die LP-Relaxierung LPRel bzgl. E'
3. Nun muss man testen, ob diese Lösung auch eine gültige Lösung der LP-Relaxierung bzgl. E ist.
4. Falls dies nicht der Fall ist, müssen zusätzliche Spalten hinzugefügt werden → Re-Optimierung über dem neuen $E' \rightarrow$ gehe zu 3.
5. Sonst: Falls die Lösung von LPRel ganzzahlig ist → Optimallösung gefunden
6. Sonst: Branching

„Branch&Cut&Price“

Pricing-Problem

Geg. ist eine Lösung x^* von LPRel bzgl. $E' \subseteq E$. Ist diese optimal auch bzgl. E ?

Eine neue Spalte $t \in E \setminus E'$ kann den Lösungswert nur dann verbessern, wenn die reduzierten Kosten der dazugehörigen Variablen kleiner als 0 sind.

Reduzierte Kosten für neue Variable t : $c_t - \pi A_t$, wobei

- A_t die Spalte t der Matrix bezeichnet
- c_t der dazugehörige Kostenkoeffizient, und
- π der Lösungswert der dualen Variablen zu Zeile i .

Suche also eine Spalte t , für die der Wert $c_t - \pi A_t < 0$.

Bei TSP: probiere zunächst alle Kanten aus Reserve Graph, wenn nicht erfolgreich, dann alle restlichen Kanten aus G .

Pricing-Problem

Bemerkung: Für alle Spalten, die bereits in E' enthalten sind gilt: die reduzierten Kosten von Lösung x^* sind ≥ 0 , sie werden also nicht wieder ausgewählt.

und jetzt: auf zur neuen Challenge:

World Tour: 1.904.711 Städte

aktuelle Challenge

