max planck institut
informatik

Schnellste Wege, Teil II
Stable Marriages

Kurt Mehlhorn und Kosta Panagiotou
Max-Planck-Institut für Informatik

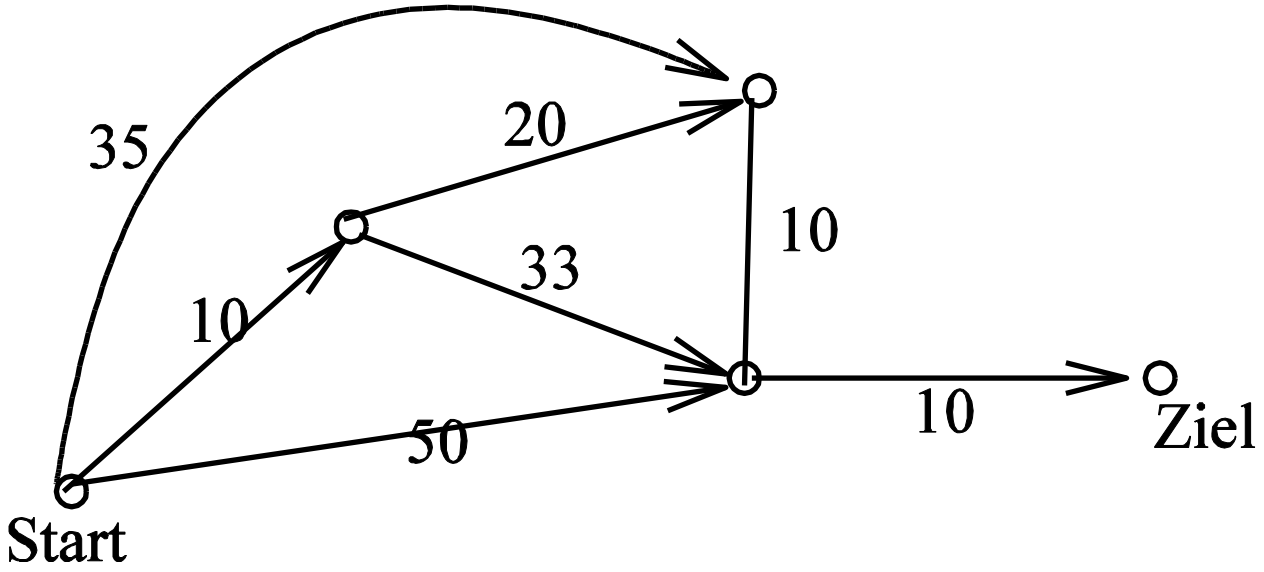
Vorlesung Computational Thinking

Übersicht

- Wiederholung: Grundalg und Korrektheit
- Dijkstras Algorithmus
- Vorberechnung für schnelle Anfragen
- Schleimpilzrechner
- Stabile Paarungen (stable marriages)
- Verwandte Probleme

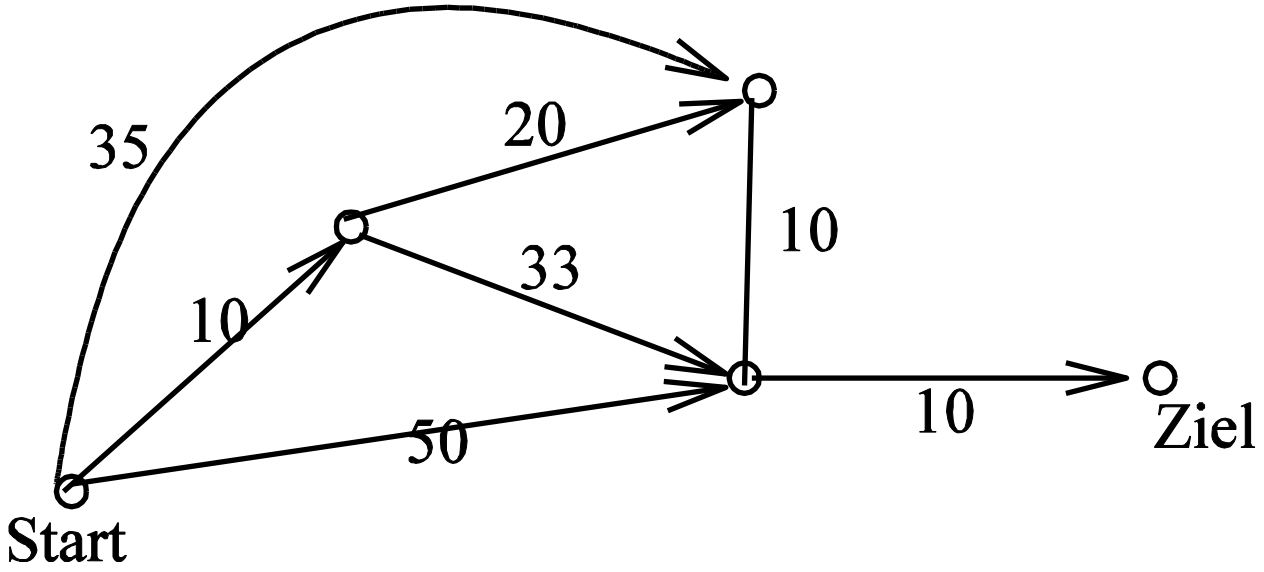
Regeln für schnellste Wege

1. Von Start nach Start in 0 Minuten
2. Wenn in X Min. nach A und Straße A→B braucht Y Min, dann in X+Y Min nach B
3. Wende an, bis stabil



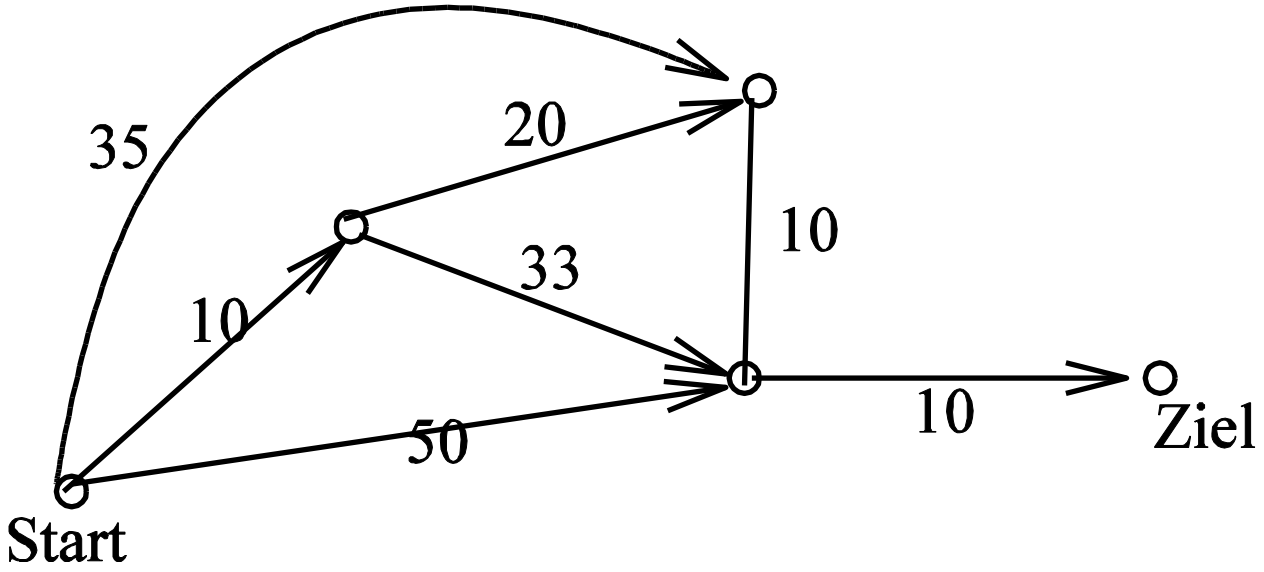
Korrektheit

1) Zu jeder Knotenmarkierung gibt es einen Pfad der entsprechenden Länge.
2) Solange schnellste Verbindung nicht gefunden, ist Situation nicht stabil.

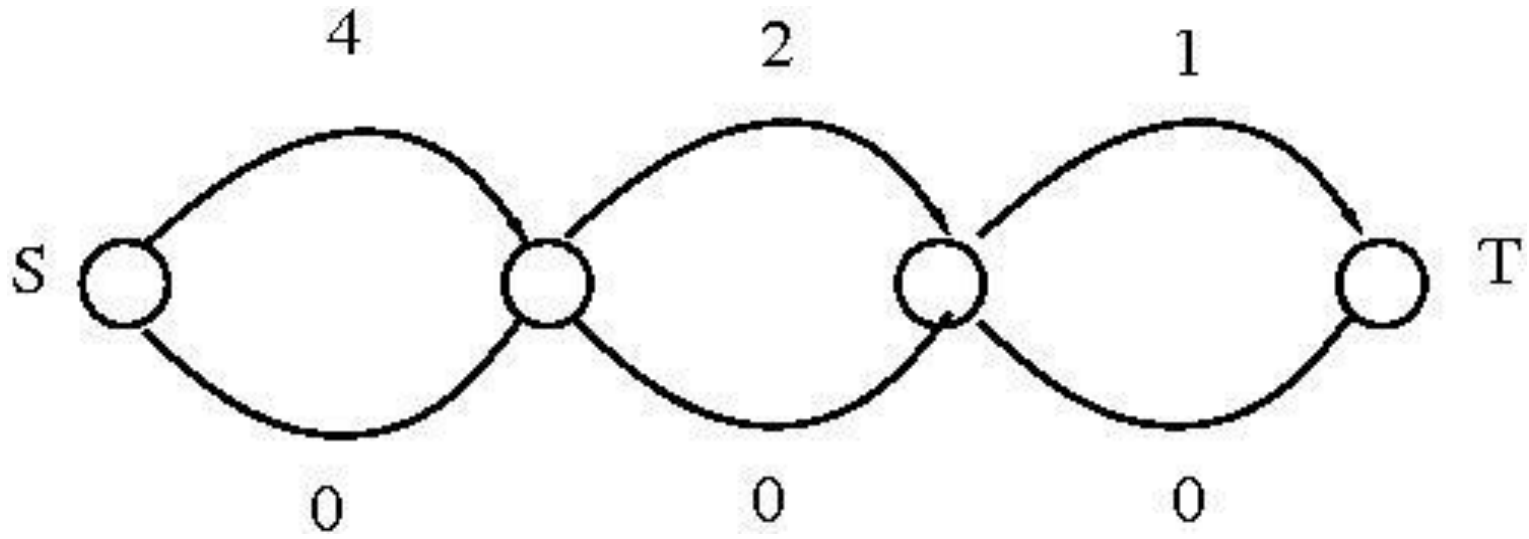


Dijkstras Algorithmus (1959)

- Wenn ich in X Minuten nach A komme und von A nach B in Y Minuten, dann komme ich in X+Y Minuten nach B.
- Wende die Regel immer auf den Knoten mit der kleinsten Fahrzeit an, auf den du sie noch nicht angewandt hast.



Und nun gemäß Dijkstra



Fahrzeit wird über jede Kante nur einmal propagiert

Strassennetzwerke

- Europa: 24 Millionen Knoten, 58 Millionen Kanten
- Dijkstra auf meinem Rechner braucht ein paar Sekunden für eine Anfrage:

Saarbrücken -- Lisabon

Dijkstra als Programm

dist[s] = 0;
dist[v] = unendlich für alle anderen Knoten v
markiere alle Knoten als aktiv;

Solange es aktiven Knoten gibt
    sei u der aktive Knoten mit kleinstem Wert dist[u]; markiere u als fertig;
    fuer alle Kanten (u,v) tue
        falls dist[u] + Zeit(u,v) < dist[v]
            ändere dist[v] auf dist[u] + Zeit(u,v)

Und nun aktuelle Forschung

- Kann man die Antworten vorberechnen?

	F	KA	M	SB
F		86	227	111
KA	86		172	106
M	227	172		268
SB	111	106	268	

10^4 Orte

1 sec pro Paar

10^4 x 10^4/2 = 500 x 10^6 = 16 Jahre

1 Jahr ist 31.5 x 10^6 sec

Verbesserungen

- Suche von Start vorwärts und von Ziel rückwärts
- Suche bevorzugt in die richtige Himmelsrichtung
- Strassenhierarchie: nur in der Nähe von Start und Ziel werden langsame Straßen benutzt
- Verbessert von Sekunden auf Millisekunden

Stand der Kunst

24 Millionen Knoten
58 Millionen Kanten

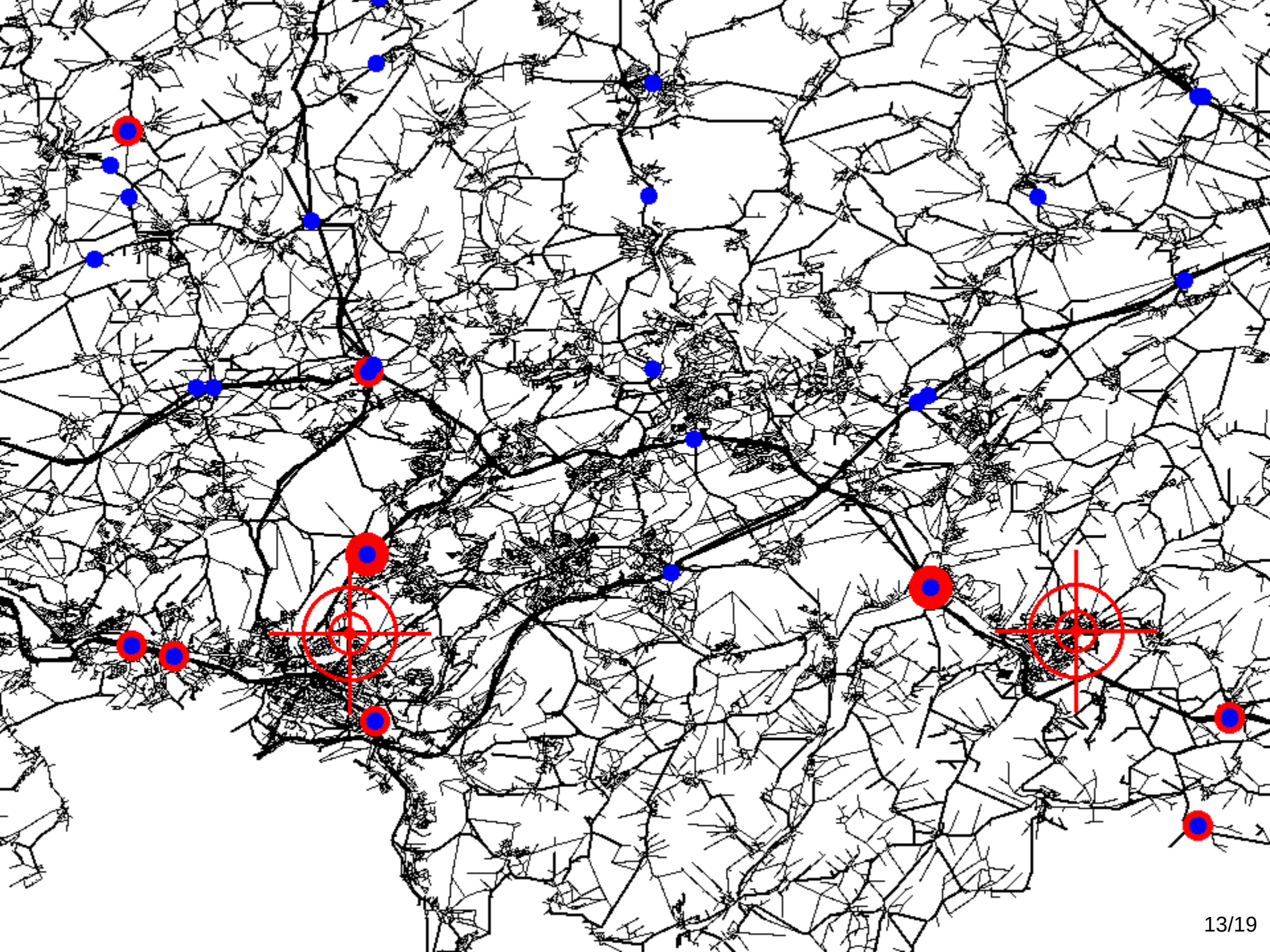
- Dijkstra's Algorithmus
  - **≈ 1 Sekunde** pro Anfrage, USA Straßennetzwerk
- Das bisher beste Verfahren
  - **≈ 1 Millisekunde** pro Anfrage, USA Netzwerk
- Bast und Funke (MPI Informatik 2007)
  - radikal neuer Ansatz: **Transitknoten**
  - **≈ 10 Mikrosekunden** pro Anfrage, USA Straßennetzwerk
  - SaarLB Preis Innovation 2008

Dijkstra '59
Luby and Ragde '85
⋮
Thorup and Zwick '01
Gutman '04
Goldberg et al '05/06
Sanders et al '04/05/06
Lauther et al '04
Möhring et al '05
Wagner et al '05/06
⋮

max planck institut informatik

Transitknoten

- Ich wohne in Scheidt, Schulstrasse
- Reisen über 100 km gehen immer entweder über St.Ingbert-West oder AS Sulzbach oder Westspange oder Kleinblittersdorf
- Das sind meine **Transitknoten**
- Alle Bewohner von Scheidt benutzen die gleichen Transitknoten
- Beobachtung gilt nicht nur für Scheidt



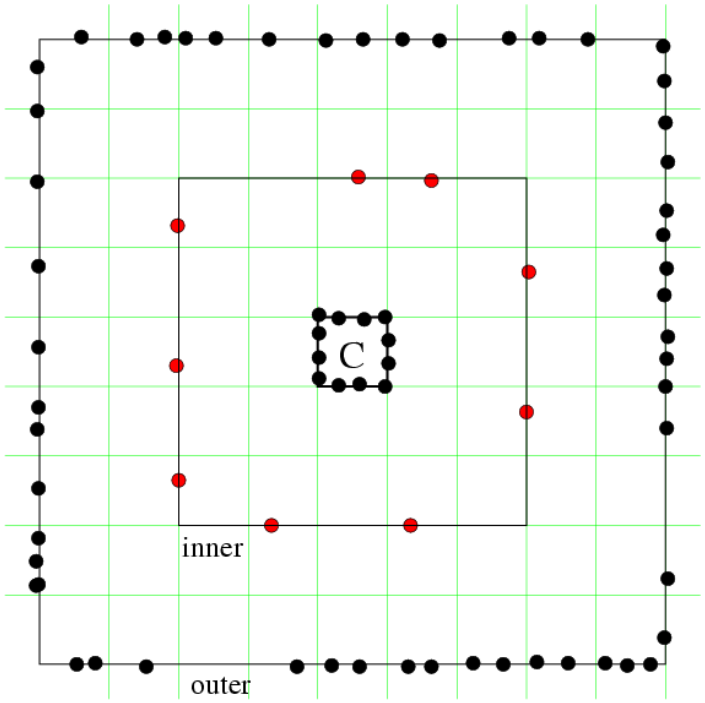
Die Transitknoten Idee

1. Vorberechnung weniger Transitknoten
   - mit der Eigenschaft, dass jeder kürzeste Pfad über eine gewisse Mindestdistanz durch einen Transitknoten geht
2. Vorberechnung der nächsten Transitknoten für jeden Knoten
   - mit der Eigenschaft, dass jeder kürzeste Pfad über eine gewisse Mindestdistanz von diesem Knoten aus durch eine dieser nächsten Transitknoten geht
3. Vorberechnung aller Distanzen
   - zwischen allen Paaren von Transitknoten und von jedem Knoten zu seinen nächsten Transitknoten

⇨ **Suchanfrage = wenige table lookups !**

Suchanfrage

Vorberechnung der Transitknoten

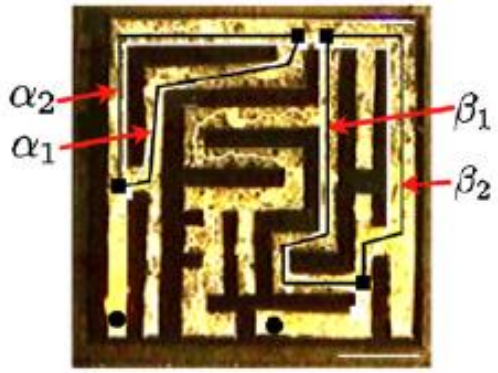


- Gitter, 10 km
- Bestimme schnellste Wege von jedem Punkt in C zu jedem Punkt am äußeren Rand

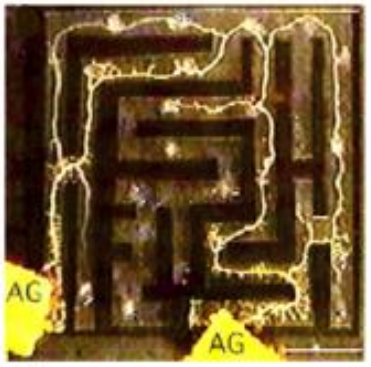
Schnittpunkte mit inner sind Transitknoten

Start

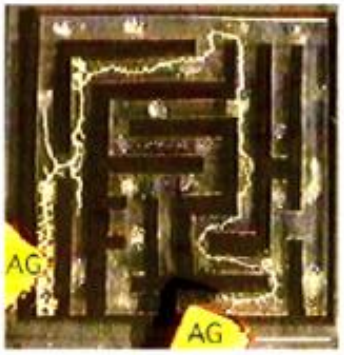
Schleimpilzcomputer



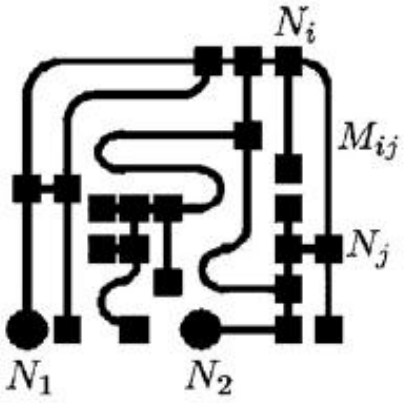
(a)



(b)



(c)



(d)

- Physarum
- Schleimpilz
- kann große Netzwerke bilden
- Nagaki, Yamada, Toth, Nature 2000
- Video

Mathematisches Model

- Slime wird als elektrisches Netzwerk modelliert
- Jede Kante e hat eine Länge L_e und einen Durchmesser $D_e(t)$
- Wir schicken einen 1 Einheit Strom von s_0 nach s_1
- Widerstand von e ist

$$R_e(t) = L_e / D_e(t)$$

- $Q_e(t)$ ist Fluss über e zum Zeitpunkt t
- Dynamik

$$\frac{D_e(t)}{dt} = |Q_e(t)| - D_e(t)$$

Tero et al., J. of Theoretical Biology, 553 -- 564, 2007

Satz (Bonifaci, Mehlhorn, Varma, 2011)

Das System konvergiert gegen den kürzesten Weg P, d.h.

D_e wird 1 für $e \in P$

D_e wird 0 für $e \notin P$

Stabile Paarungen

n Frauen und n Männer: jede Frau hat eine Reihung der Männer, jeder Mann hat eine Reihung der Frauen

Stabile Paarungen

n Frauen und n Männer: jede Frau hat eine Reihung der Männer, jeder Mann hat eine Reihung der Frauen

Algorithmus für Stable Marriages

Algorithmus arbeitet in Runden

- Jeder Mann ohne Partner wirbt um die erste Frau auf seiner Liste und streicht sie von der Liste
- Jede Frau wählt den besten Mann aus augenblicklichem Partner (falls sie schon einen hat) und Bewerbern in dieser Runde. Alle anderen weist sie zurück
- Algorithmus endet, wenn alle gepaart sind.

Eigenschaften

- Sobald ein Mann um eine Frau geworben hat, hat sie von da an immer einen Partner, sogar immer bessere in ihrem Sinn.
- Alg endet mit einer vollständigen Paarung: nimm an ein Mann (und damit eine Frau) geht leer aus; der Mann hat sich aber auch um diese Frau geworben.

Berechnete Paarung ist stabil

Nimm an, das wäre nicht so. M1 ist am Ende mit F1 gepaart.

Da er F2 vorzieht, hat er um F2 geworben

F2 hat ihn irgendwann verstoßen, weil ein besserer kam. Also ist F2 am Ende mit jemand gepaart, der ihrer Meinung nach besser ist als M1. M2 ist aber schlechter.

Weitere Eigenschaften

Frauen enden beim besten Bewerber, …

Sei X ein beliebiger Mann und Y die ihm vom Alg zugewiesene Partnerin. Dann gibt es **keine** stabile Paarung, die X eine bessere Partnerin zuordnet.

Paarung ist Mann-Optimal

Verwandte Probleme

- Maximale Flüsse
- Flüsse minimaler Kosten
- Optimale Zuordnungen