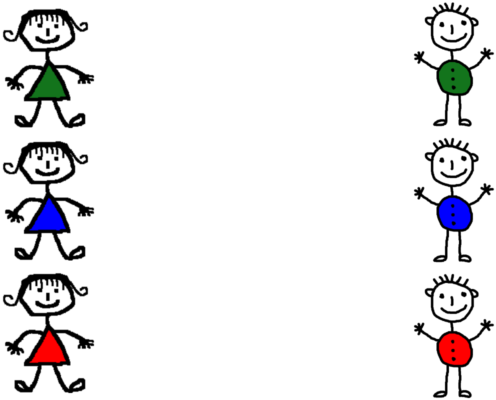
Dynamisches Matching mit Präferenzen

Martin Hoefer

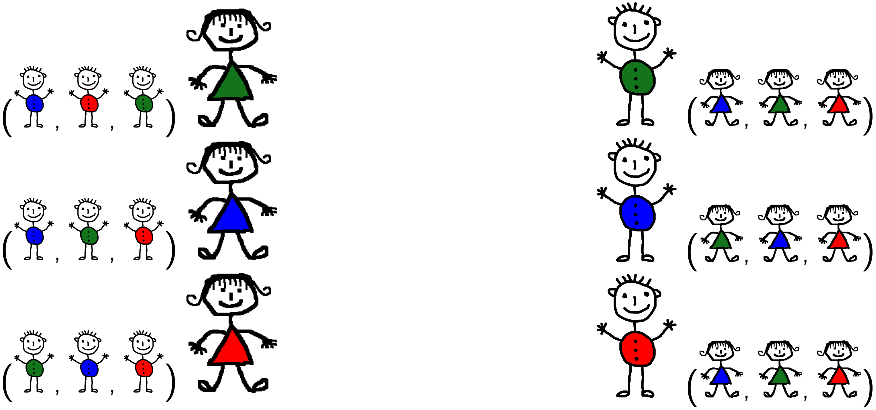
mhoefer@cs.uni-frankfurt.de

Wie findet man eine stabile Beziehung?

Stabiles Matching

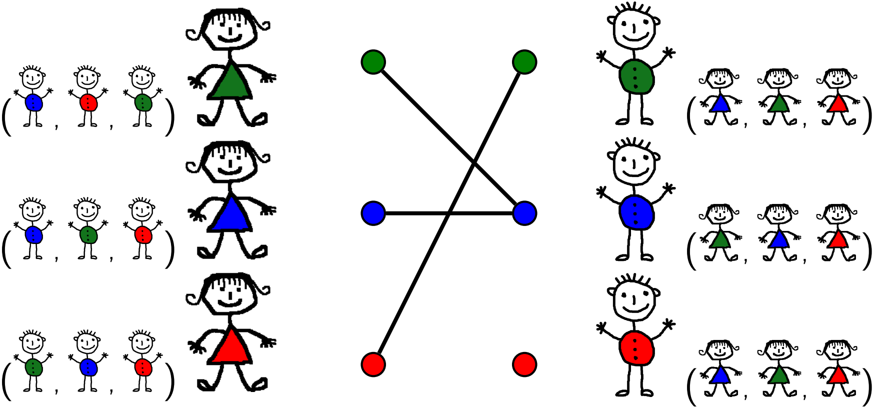


Stabiles Matching



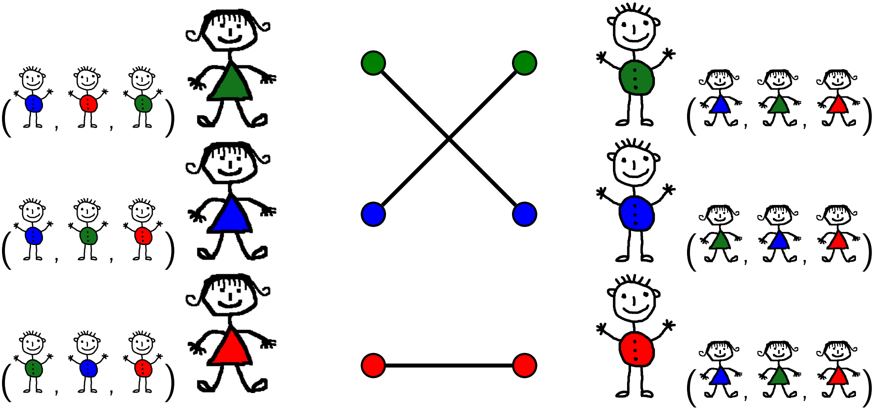
Jede Person hat eine Präferenzliste.

Stabiles Matching



Jede Person hat eine Präferenzliste.

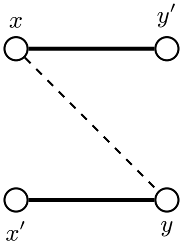
Stabiles Matching



Jede Person hat eine Präferenzliste.

Stabiles Matching

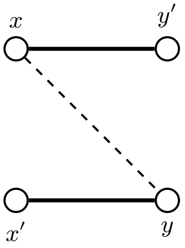
- $\{x, y\}$ ist Blocking Pair gdw x und y sich beide ihren momentanen Partnern vorziehen.
- Matching M ist ein stabiles Matching gdw es kein Blocking Pair erlaubt.



Stabiles Matching

- $\{x, y\}$ ist Blocking Pair gdw x und y sich beide ihren momentanen Partnern vorziehen.

- Matching M ist ein stabiles Matching gdw es kein Blocking Pair erlaubt.



Einige Resultate und Erweiterungen:

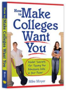
- Ein stabiles Matching existiert immer, und es gibt einen effizienten Algorithmus zur Berechnung. [Gale, Shapley 1962]

- Viele weitere Resultate seit den 60ern: "Mitbewohnerproblem", Ties, Unvollständige Listen, Multi-Matching, Netzwerkdesign...

Anwendungen

Ärzte/Krankenhäuser

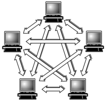
Uni-Zulassung



Arbeitsmarkt

Handel

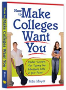
P2P Netzwerke



etc.

Ärzte/Krankenhäuser

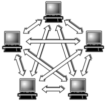
Uni-Zulassung



Arbeitsmarkt

Handel

P2P Netzwerke



etc.

Heute:

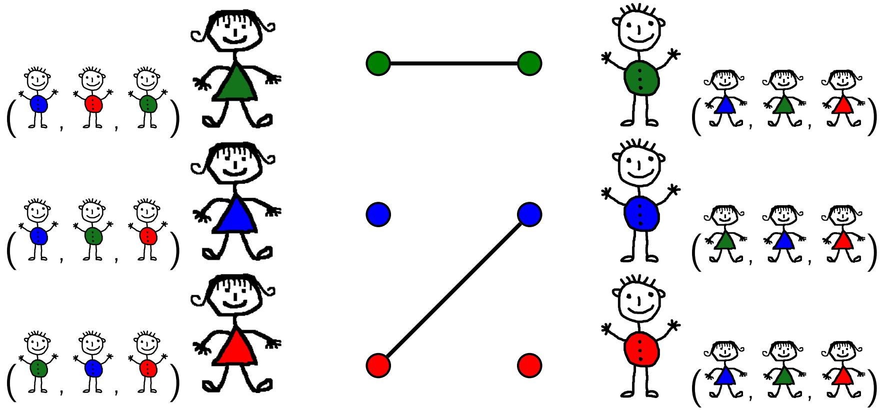
Was passiert, wenn ...

- ... das Matching nicht zentral vorgegeben wird, und...
- ... Teilnehmer nur beschränkte Information über mögliche Partner haben?

Können sie ein stabiles Matching erreichen? Wie lange dauert das?

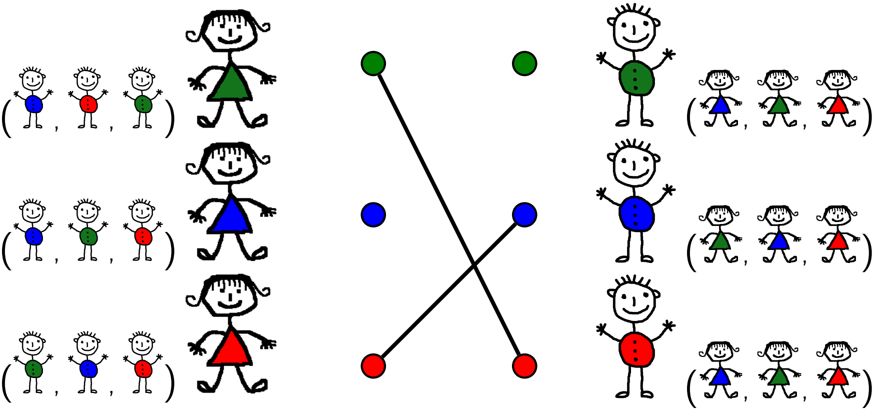
Blocking-Pair-Dynamik

Matching nicht stabil ⇒ Wähle Blocking Pair und löse es auf



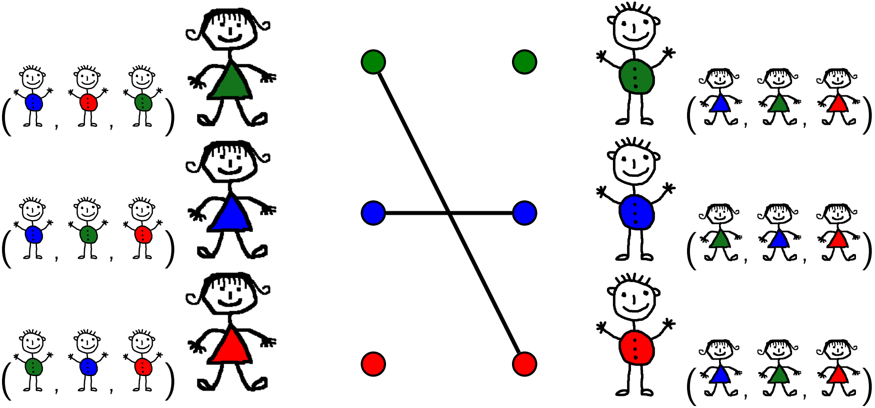
Blocking-Pair-Dynamik

Matching nicht stabil $\Rightarrow$ Wähle Blocking Pair und löse es auf



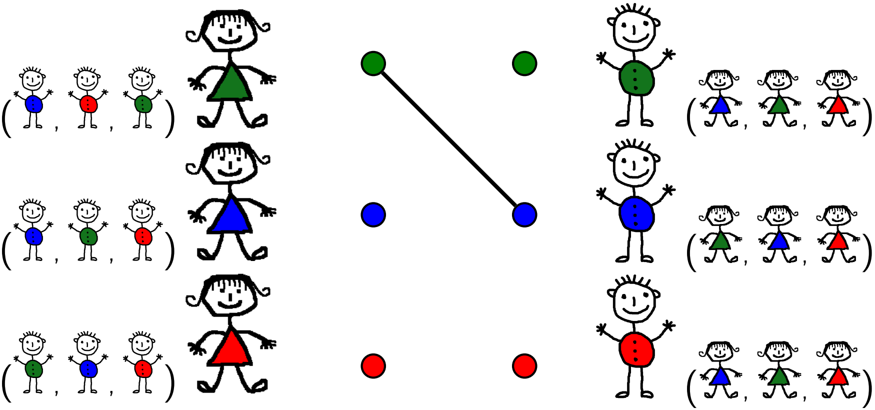
Blocking-Pair-Dynamik

Matching nicht stabil $\Rightarrow$ Wähle Blocking Pair und löse es auf



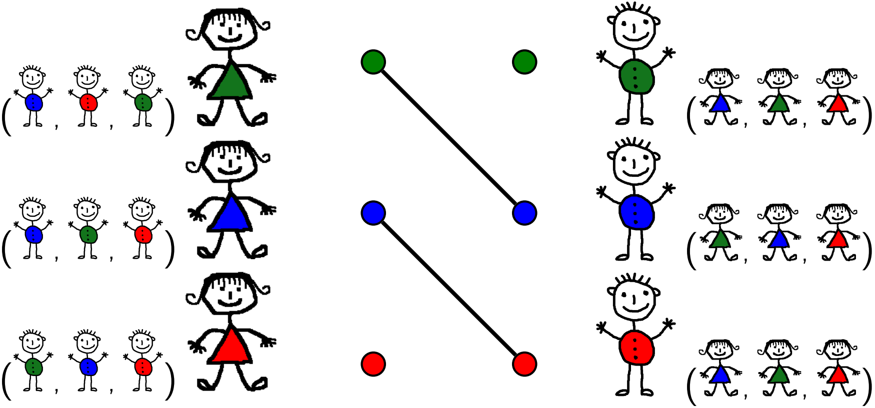
Blocking-Pair-Dynamik

Matching nicht stabil ⇒ Wähle Blocking Pair und löse es auf



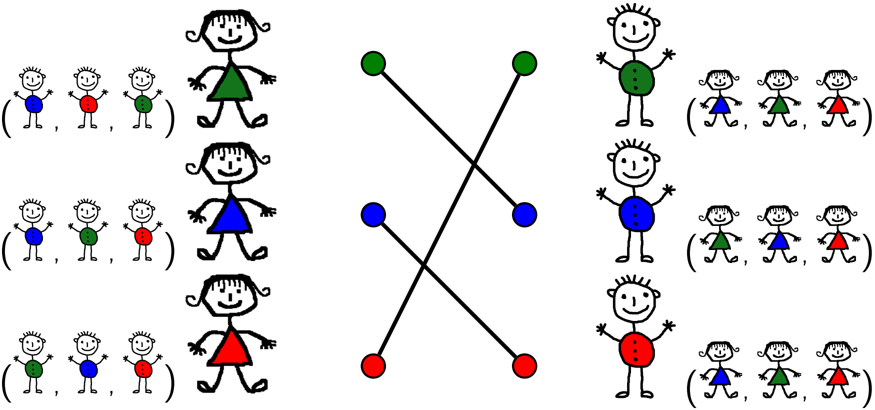
Blocking-Pair-Dynamik

Matching nicht stabil $\Rightarrow$ Wähle Blocking Pair und löse es auf



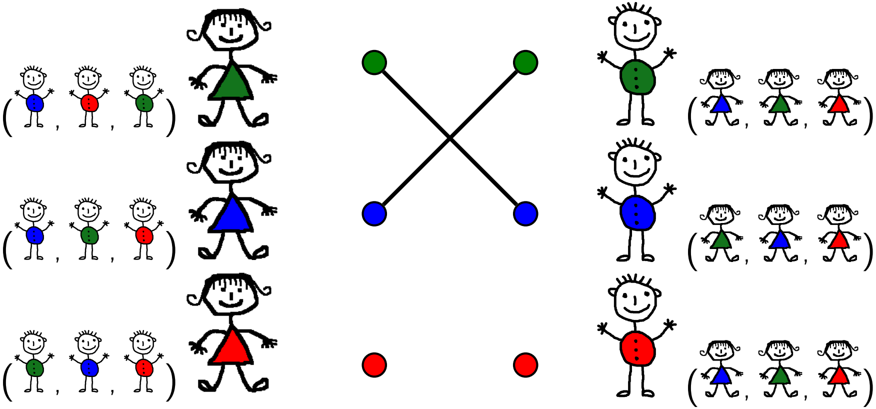
Blocking-Pair-Dynamik

Matching nicht stabil $\Rightarrow$ Wähle Blocking Pair und löse es auf



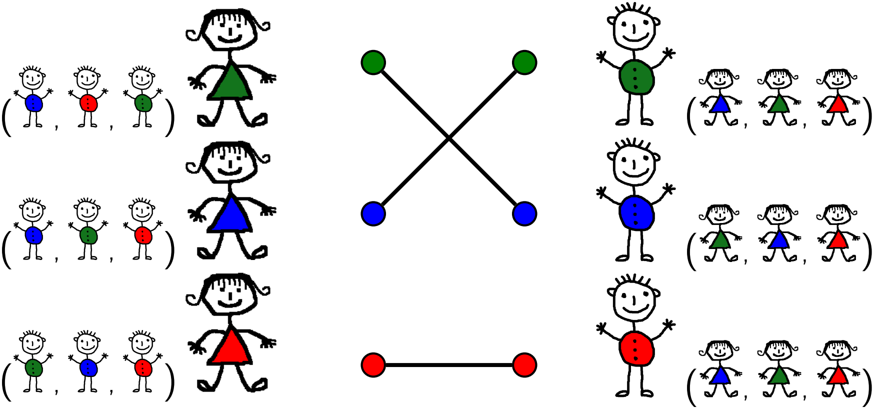
Blocking-Pair-Dynamik

Matching nicht stabil $\Rightarrow$ Wähle Blocking Pair und löse es auf



Blocking-Pair-Dynamik

Matching nicht stabil ⇒ Wähle Blocking Pair und löse es auf

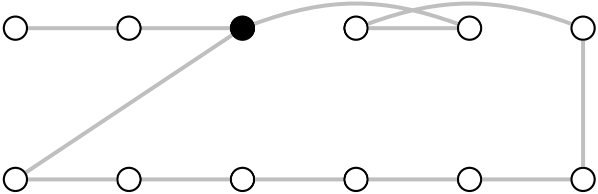


Resultate

- Blocking-Pair-Dynamik kann kreisen. [Knuth 1976]
- Von jedem initialen Matching gibt es eine Folge von Auflösungen, die in einem stabilen Matching endet. Die Folge hat polynomielle Länge. [Roth, Vande Vate 1990]
- Wenn das Blocking Pair in jedem Schritt uniform zufällig gewählt wird, konvergieren wir mit Wahrscheinlichkeit 1 zum stabilen Matching.
- Zufällige Dynamiken können mit hoher Wahrscheinlichkeit exponentielle Zeit benötigen, um ein stabiles Matching zu erreichen. [Ackermann, Goldberg, Mirrokni, Röglin, Vöcking 2011]

Lokal Stabiles Matching

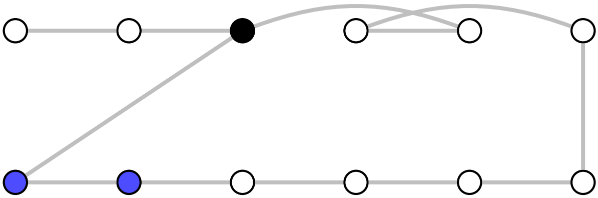
[Arcaute, Vassilvitskii 2009]



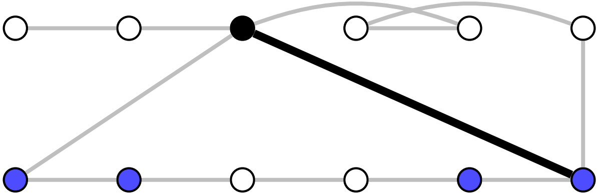
- Teilnehmer sind Knoten in einem statischen (sozialen) Netzwerk N mit ungerichteten Links L.
- Informationsstruktur mit Netzwerk und Triadic Closure:

Lokal Stabiles Matching

[Arcaute, Vassilvitskii 2009]



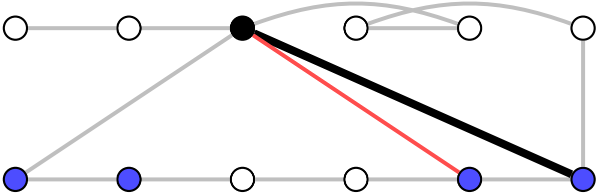
- Teilnehmer sind Knoten in einem statischen (sozialen) Netzwerk N mit ungerichteten Links L.
- Informationsstruktur mit Netzwerk und Triadic Closure:
- Jede/r Mann (Frau) kann zu jeder/m Frau (Mann) in der 2-Hop-Nachbarschaft in N verbinden.



- Teilnehmer sind Knoten in einem statischen (sozialen) Netzwerk N mit ungerichteten Links L.
- Informationsstruktur mit Netzwerk und Triadic Closure:
- Jede/r Mann (Frau) kann zu jeder/m Frau (Mann) in der 2-Hop-Nachbarschaft in N verbinden.
- Paar $\{x, y\} \Rightarrow x$ (y) kann sich zu einem direkten Nachbar von y (x) in N verbinden.

Lokal Stabiles Matching

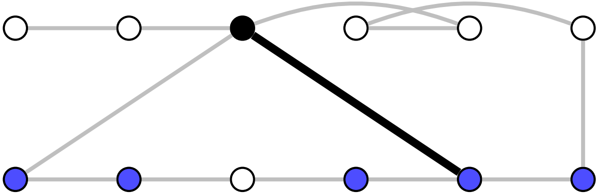
[Arcaute, Vassilvitskii 2009]



- In einem Matching M ist $\{x, y\}$ erreichbar wenn x und y Hop-Distanz 2 im Graphen $G = (V, L \cup M)$ haben.
- $\{x, y\}$ ist ein lokales Blocking Pair
  $\Leftrightarrow$ $\{x, y\}$ Blocking Pair und erreichbar.
- Matching M ist lokal stabiles Matching
  $\Leftrightarrow$ M erlaubt kein lokales Blocking Pair.

Lokal Stabiles Matching

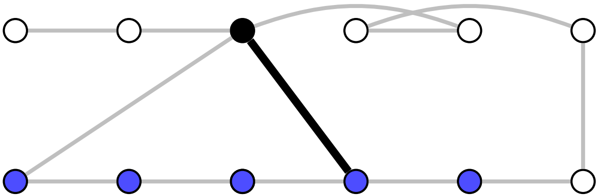
[Arcaute, Vassilvitskii 2009]



- In einem Matching M ist $\{x, y\}$ erreichbar wenn x und y Hop-Distanz 2 im Graphen $G = (V, L \cup M)$ haben.
- $\{x, y\}$ ist ein lokales Blocking Pair
  $\Leftrightarrow$ $\{x, y\}$ Blocking Pair und erreichbar.
- Matching M ist lokal stabiles Matching
  $\Leftrightarrow$ M erlaubt kein lokales Blocking Pair.

Lokal Stabiles Matching

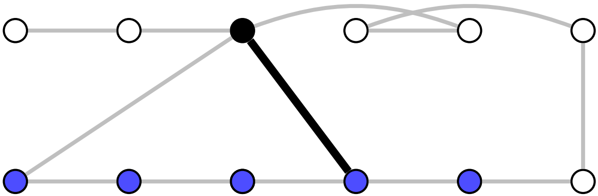
[Arcaute, Vassilvitskii 2009]



- In einem Matching M ist $\{x, y\}$ erreichbar wenn x und y Hop-Distanz 2 im Graphen $G = (V, L \cup M)$ haben.
- $\{x, y\}$ ist ein lokales Blocking Pair
  $\Leftrightarrow$ $\{x, y\}$ Blocking Pair und erreichbar.
- Matching M ist lokal stabiles Matching
  $\Leftrightarrow$ M erlaubt kein lokales Blocking Pair.

Lokal Stabiles Matching

[Arcaute, Vassilvitskii 2009]



Was passiert, wenn Teilnehmer iterativ lokale Blocking Pairs auflösen?

Lokal Stabiles Matching mit strikten Präferenzlisten [H., Wagner 2017]

Erreichbarkeit für lokal stabile Matchings mit beliebigen strikten Präferenzlisten

	Stabiles Matching	Lokal Stabiles Matching
Erreichbarkeit	Ja [RVV'90]	NP-hart

Lokal Stabiles Matching mit strikten Präferenzlisten [H., Wagner 2017]

Erreichbarkeit für lokal stabile Matchings mit beliebigen strikten Präferenzlisten

	Stabiles Matching	Lokal Stabiles Matching
Erreichbarkeit	Ja [RVV'90]	NP-hart
Kürzeste Sequenz	$O(n^2)$ [RVV'90]	$2^{\Omega(n)}$

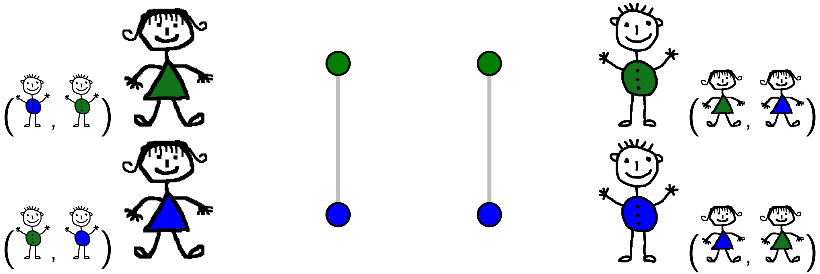
Lokal Stabiles Matching mit strikten Präferenzlisten [H., Wagner 2017]

Erreichbarkeit für lokal stabile Matchings mit beliebigen strikten Präferenzlisten

	Stabiles Matching	Lokal Stabiles Matching
Erreichbarkeit	Ja [RVV'90]	NP-hart (PSPACE-hart?)
Kürzeste Sequenz	$O(n^2)$ [RVV'90]	$2^{\Omega(n)}$

Unerreichbare LSMs

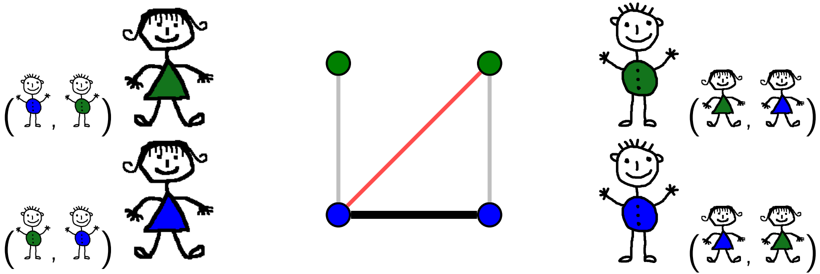
Es können nicht genug Paare erzeugt werden:



Zyklische Strukturen mit mehreren Paaren: [...]

Unerreichbare LSMs

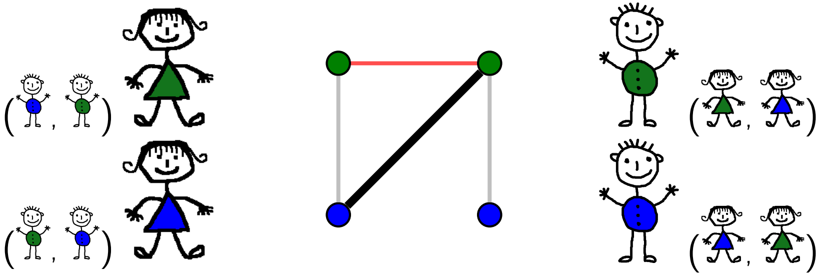
Es können nicht genug Paare erzeugt werden:



Zyklische Strukturen mit mehreren Paaren: [...]

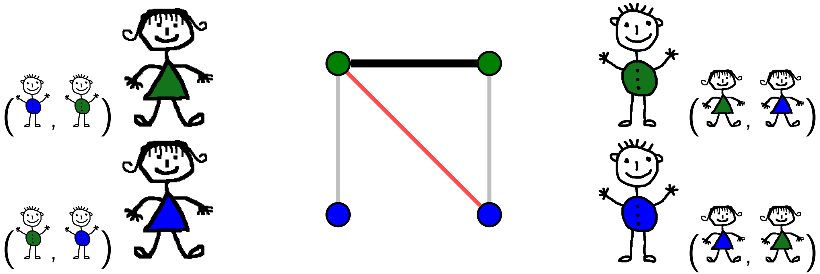
Unerreichbare LSMs

Es können nicht genug Paare erzeugt werden:



Zyklische Strukturen mit mehreren Paaren: [...]

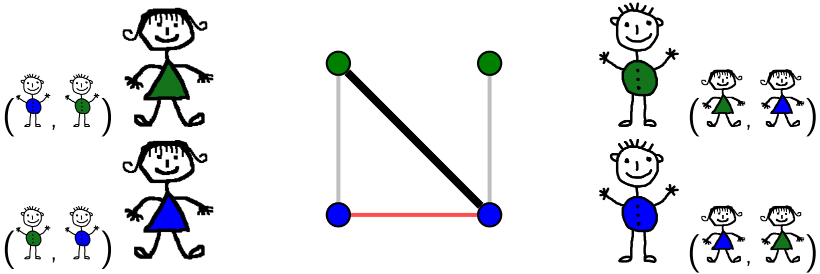
Es können nicht genug Paare erzeugt werden:



Zyklische Strukturen mit mehreren Paaren: [...]

Unerreichbare LSMs

Es können nicht genug Paare erzeugt werden:



Zyklische Strukturen mit mehreren Paaren: [...]

Memory

Konvergieren Dynamiken, wenn sich Teilnehmer an (einige) vorherige Partner erinnern?

Konvergieren Dynamiken, wenn sich Teilnehmer an (einige) vorherige Partner erinnern?

Zufalls-Memory

- Jeder Teilnehmer erinnert sich in jeder Runde uniform zufällig an einen Ex-Partner. Dieses Paar wird in der nächsten Runde erreichbar.

Memory

Konvergieren Dynamiken, wenn sich Teilnehmer an (einige) vorherige Partner erinnern?

Zufalls-Memory

- Jeder Teilnehmer erinnert sich in jeder Runde uniform zufällig an einen Ex-Partner. Dieses Paar wird in der nächsten Runde erreichbar.

Deterministisches Memory

- Jeder Teilnehmer erinnert sich deterministisch an einen Ex-Partner.
- Qualitäts-Memory: Erinnert sich an den besten Ex-Partner.
- Zeit-Memory: Erinnert sich an den letzten Ex-Partner.

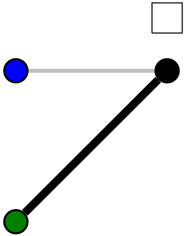
Lokal Stabiles Matching mit strikten Präferenzlisten [H., Wagner 2017]

Erreichbarkeit für lokal stabile Matchings mit beliebigen strikten Präferenzlisten

	Stabiles Matching	Lokal Stabiles Matching
Erreichbarkeit	Ja [RVV'90]	NP-hart
Kürzeste Sequenz	$O(n^2)$ [RVV'90]	$2^{\Omega(n)}$
Erreichbk. Qualität	Ja, $O(n^2)$	NP-hart, wenn eine Seite keine internen Links

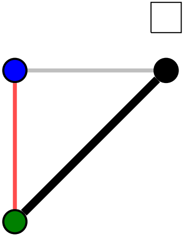
Qualitäts-Memory kann man leicht austricksen!

Zu einer Härte-Instanz für leeres Anfangsmatching fügen wir separate Gadgets zu ausgewählten Knoten hinzu, die zu Fixierung führen.



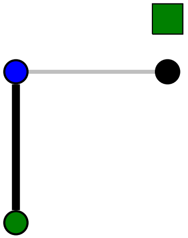
Qualitäts-Memory kann man leicht austricksen!

Zu einer Härte-Instanz für leeres Anfangsmatching fügen wir separate Gadgets zu ausgewählten Knoten hinzu, die zu Fixierung führen.



Qualitäts-Memory kann man leicht austricksen!

Zu einer Härte-Instanz für leeres Anfangsmatching fügen wir separate Gadgets zu ausgewählten Knoten hinzu, die zu Fixierung führen.



Lokal Stabiles Matching mit strikten Präferenzlisten [H., Wagner 2017]

Erreichbarkeit für lokal stabile Matchings mit beliebigen strikten Präferenzlisten

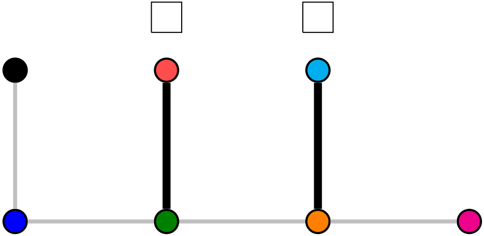
	Stabiles Matching	Lokal Stabiles Matching
Erreichbarkeit	Ja [RVV'90]	NP-hart
Kürzeste Sequenz	$O(n^2)$ [RVV'90]	$2^{\Omega(n)}$
Erreichbk. Qualität	Ja, $O(n^2)$	NP-hart, wenn eine Seite keine internen Links
Erreichbk. Zeit	Ja, $O(n^2)$	Ja, $O(n^3)$, wenn eine Seite keine internen Links

Erreichbarkeit mit Zeit-Memory

Konvergenz bei voller Information in zwei Phasen:

1. Verheiratete Männer lösen Blocking Pairs auf (männl. Präferenzanstieg)
2. Unverheiratete Männer lösen Blocking Pairs auf (weibl. Präferenzanstieg)

Keine Links zwischen Männern ⇒ Erweitere zweite Phase mit Zeit-Memory.

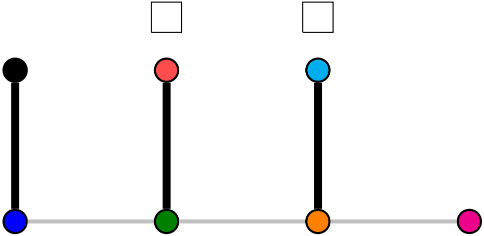


Erreichbarkeit mit Zeit-Memory

Konvergenz bei voller Information in zwei Phasen:

1. Verheiratete Männer lösen Blocking Pairs auf (männl. Präferenzanstieg)
2. Unverheiratete Männer lösen Blocking Pairs auf (weibl. Präferenzanstieg)

Keine Links zwischen Männern $\Rightarrow$ Erweitere zweite Phase mit Zeit-Memory.

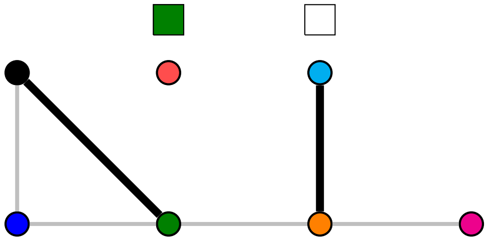


Erreichbarkeit mit Zeit-Memory

Konvergenz bei voller Information in zwei Phasen:

1. Verheiratete Männer lösen Blocking Pairs auf (männl. Präferenzanstieg)
2. Unverheiratete Männer lösen Blocking Pairs auf (weibl. Präferenzanstieg)

Keine Links zwischen Männern $\Rightarrow$ Erweitere zweite Phase mit Zeit-Memory.

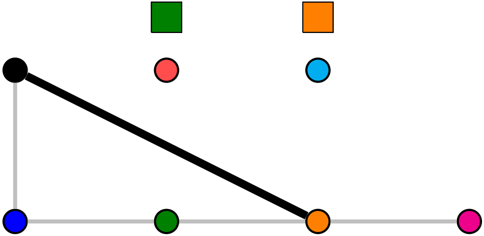


Erreichbarkeit mit Zeit-Memory

Konvergenz bei voller Information in zwei Phasen:

1. Verheiratete Männer lösen Blocking Pairs auf (männl. Präferenzanstieg)
2. Unverheiratete Männer lösen Blocking Pairs auf (weibl. Präferenzanstieg)

Keine Links zwischen Männern ⇒ Erweitere zweite Phase mit Zeit-Memory.

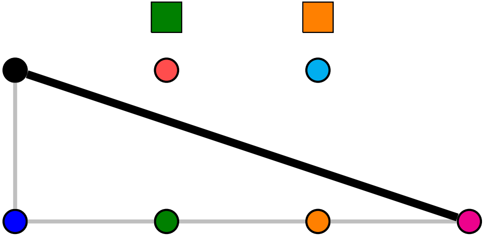


Erreichbarkeit mit Zeit-Memory

Konvergenz bei voller Information in zwei Phasen:

1. Verheiratete Männer lösen Blocking Pairs auf (männl. Präferenzanstieg)
2. Unverheiratete Männer lösen Blocking Pairs auf (weibl. Präferenzanstieg)

Keine Links zwischen Männern $\Rightarrow$ Erweitere zweite Phase mit Zeit-Memory.

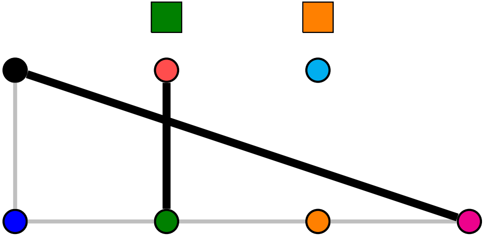


Erreichbarkeit mit Zeit-Memory

Konvergenz bei voller Information in zwei Phasen:

1. Verheiratete Männer lösen Blocking Pairs auf (männl. Präferenzanstieg)
2. Unverheiratete Männer lösen Blocking Pairs auf (weibl. Präferenzanstieg)

Keine Links zwischen Männern $\Rightarrow$ Erweitere zweite Phase mit Zeit-Memory.

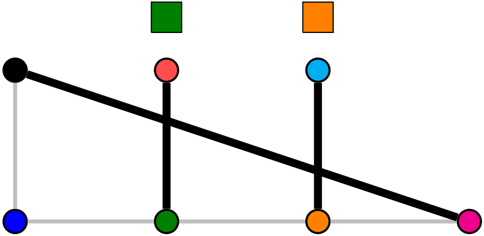


Erreichbarkeit mit Zeit-Memory

Konvergenz bei voller Information in zwei Phasen:

1. Verheiratete Männer lösen Blocking Pairs auf (männl. Präferenzanstieg)
2. Unverheiratete Männer lösen Blocking Pairs auf (weibl. Präferenzanstieg)

Keine Links zwischen Männern ⇒ Erweitere zweite Phase mit Zeit-Memory.



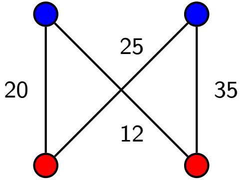
Lokal Stabiles Matching mit strikten Präferenzlisten [H., Wagner 2017]

Erreichbarkeit für lokal stabile Matchings mit beliebigen strikten Präferenzlisten

	Stabiles Matching	Lokal Stabiles Matching
Erreichbarkeit	Ja [RVV'90]	NP-hart
Kürzeste Sequenz	$O(n^2)$ [RVV'90]	$2^{\Omega(n)}$
Erreichbk. Qualität	Ja, $O(n^2)$	NP-hart, wenn eine Seite keine internen Links
Erreichbk. Zeit	Ja, $O(n^2)$	Ja, $O(n^3)$, wenn eine Seite keine internen Links
Erreichbk. Zufall	m. Wkeit 1	m. Wkeit 1

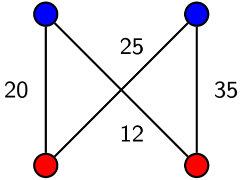
Gewichtetes Matching

Jede mögliche Kante $e = \{x, y\}$ liefert Profit $b_e > 0$.



Gewichtetes Matching

Jede mögliche Kante $e = \{x, y\}$ liefert Profit $b_e > 0$.



- Dynamik azyklisch – der sortierte Vektor von Kantenprofiten steigt lexikographisch an.

- Beste-Antwort-Dynamik löst immer das Blocking Pair mit größtem Profit auf. Diese Dynamik konvergiert in Zeit $O(n)$ zu einem stabilen Matching.

[Ackermann, Goldberg, Mirrokni, Röglin, Vöcking 2011]

Resultate für Gewichtetes Matching [H. 2013]

Parameter:

- Es gibt eine Teilmenge E von möglichen Kanten, sei $m = |E|$.

	Min-Folge	Zufäll. Dynamik	
Beste-Antwort			
	$2^{\Omega(n)}$	$2^{\Omega(n)}$	
Beliebig			
	$O(n \cdot m^2)$	$2^{\Omega(n)}$	

Resultate für Gewichtetes Matching [H. 2013]

Parameter:

- Es gibt eine Teilmenge E von möglichen Kanten, sei $m = |E|$.
- Jeder Agent kann bis zu $k \geq 1$ Kanten erstellen.
- Jeder Agent hat Sichtweite $\ell \geq 2$ im Graphen $G = (V, L \cup M)$.

	Min-Folge	Zufäll. Dynamik	
Beste-Antwort			
$k = 1$, $\ell = 2$	$2^{\Omega(n)}$	$2^{\Omega(n)}$	
Beliebig			
$k = 1$, $\ell = 2$	$O(n \cdot m^2)$	$2^{\Omega(n)}$	
$k > 1$ or $\ell > 2$	$2^{\Omega(n)}$	$2^{\Omega(n)}$	

Resultate für Gewichtetes Matching

[H. 2013]

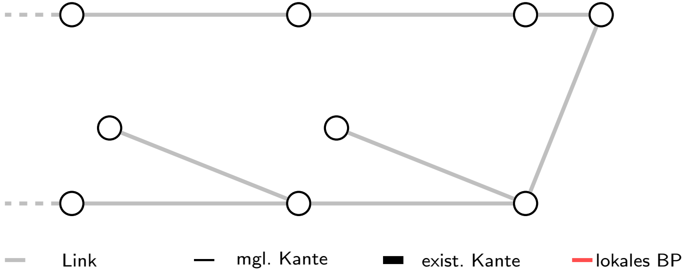
Parameter:

- Es gibt eine Teilmenge E von möglichen Kanten, sei $m = |E|$.
- Jeder Agent kann bis zu $k \geq 1$ Kanten erstellen.
- Jeder Agent hat Sichtweite $\ell \geq 2$ im Graphen $G = (V, L \cup M)$.

	Min-Folge	Zufäll. Dynamik	m. Zufalls-Memory
Beste-Antwort			
$k = 1,\ \ell = 2$	$2^{\Omega(n)}$	$2^{\Omega(n)}$	$O(n \cdot m^2)$
Beliebig			
$k = 1,\ \ell = 2$	$O(n \cdot m^2)$	$2^{\Omega(n)}$	$O(n \cdot m^2)$
$k > 1$ or $\ell > 2$	$2^{\Omega(n)}$	$2^{\Omega(n)}$	$O(n \cdot k \cdot m^2)$

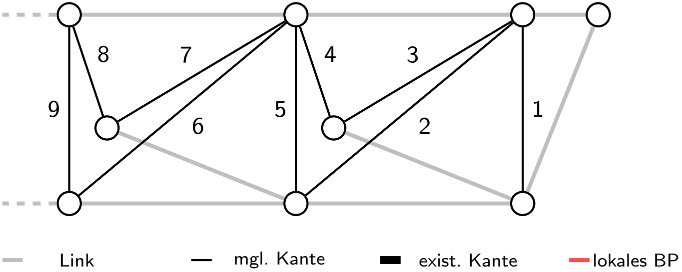
Zufällige Dynamik mit Zufalls-Memory konvergiert in polynomieller Zeit.

Bitzähler und Verbessungsschritte



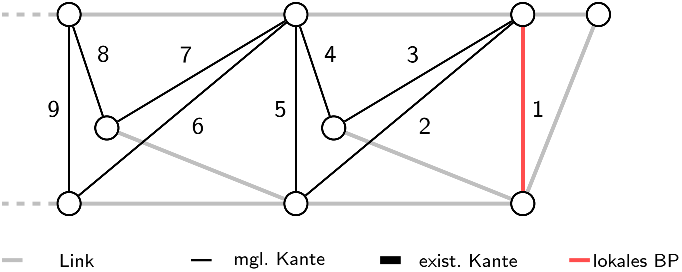
- Die eindeutige Beste-Antwort-Sequenz hat Länge $2^{\Omega(n)}$.
- Die zufällige Dynamik wählt eine beste Antwort mit Wkeit mind. 1/2, dadurch erwartete Anzahl von 1.5 Erzeugungen bis eine Kante zum nächsten Block vordringt.
- Bei b Blöcken ergibt sich eine Konvergenzzeit von mind. $1.5^b = 2^{\Omega(n)}$.

Bitzähler und Verbessungsschritte



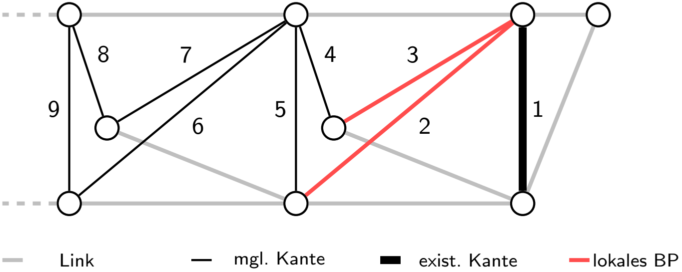
- Die eindeutige Beste-Antwort-Sequenz hat Länge $2^{\Omega(n)}$.
- Die zufällige Dynamik wählt eine beste Antwort mit Wkeit mind. 1/2, dadurch erwartete Anzahl von 1.5 Erzeugungen bis eine Kante zum nächsten Block vordringt.
- Bei b Blöcken ergibt sich eine Konvergenzzeit von mind. $1.5^b = 2^{\Omega(n)}$.

Bitzähler und Verbessungsschritte



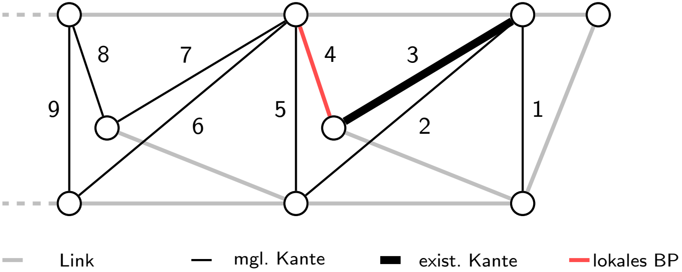
- Die eindeutige Beste-Antwort-Sequenz hat Länge $2^{\Omega(n)}$.
- Die zufällige Dynamik wählt eine beste Antwort mit Wkeit mind. 1/2, dadurch erwartete Anzahl von 1.5 Erzeugungen bis eine Kante zum nächsten Block vordringt.
- Bei b Blöcken ergibt sich eine Konvergenzzeit von mind. $1.5^b = 2^{\Omega(n)}$.

Bitzähler und Verbessungsschritte



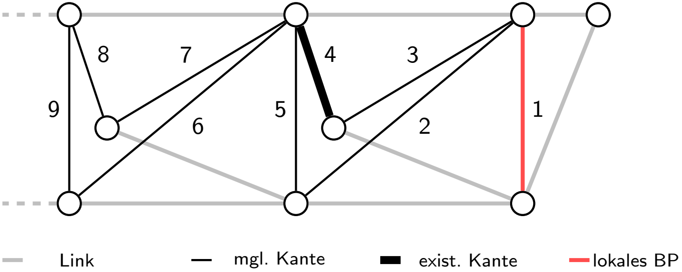
- Die eindeutige Beste-Antwort-Sequenz hat Länge $2^{\Omega(n)}$.
- Die zufällige Dynamik wählt eine beste Antwort mit Wkeit mind. 1/2, dadurch erwartete Anzahl von 1.5 Erzeugungen bis eine Kante zum nächsten Block vordringt.
- Bei b Blöcken ergibt sich eine Konvergenzzeit von mind. $1.5^b = 2^{\Omega(n)}$.

Bitzähler und Verbessungsschritte



- Die eindeutige Beste-Antwort-Sequenz hat Länge $2^{\Omega(n)}$.
- Die zufällige Dynamik wählt eine beste Antwort mit Wkeit mind. 1/2, dadurch erwartete Anzahl von 1.5 Erzeugungen bis eine Kante zum nächsten Block vordringt.
- Bei b Blöcken ergibt sich eine Konvergenzzeit von mind. $1.5^b = 2^{\Omega(n)}$.

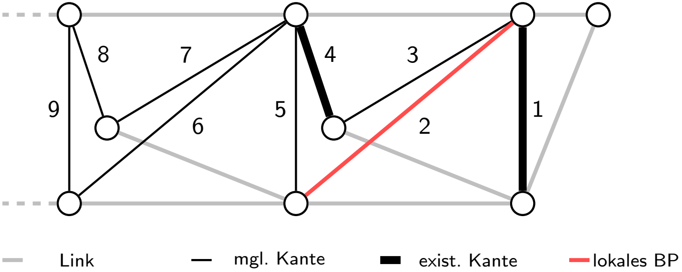
Bitzähler und Verbessungsschritte



- Die eindeutige Beste-Antwort-Sequenz hat Länge $2^{\Omega(n)}$.

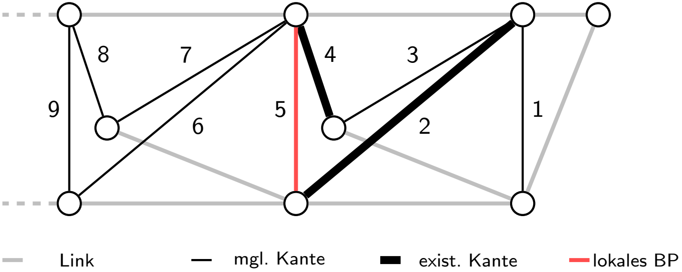
- Die zufällige Dynamik wählt eine beste Antwort mit Wkeit mind. 1/2, dadurch erwartete Anzahl von 1.5 Erzeugungen bis eine Kante zum nächsten Block vordringt.
- Bei b Blöcken ergibt sich eine Konvergenzzeit von mind. $1.5^b = 2^{\Omega(n)}$.

Bitzähler und Verbessungsschritte



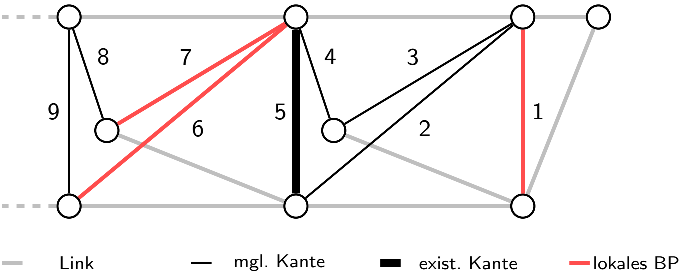
- Die eindeutige Beste-Antwort-Sequenz hat Länge $2^{\Omega(n)}$.
- Die zufällige Dynamik wählt eine beste Antwort mit Wkeit mind. 1/2, dadurch erwartete Anzahl von 1.5 Erzeugungen bis eine Kante zum nächsten Block vordringt.
- Bei b Blöcken ergibt sich eine Konvergenzzeit von mind. $1.5^b = 2^{\Omega(n)}$.

Bitzähler und Verbessungsschritte



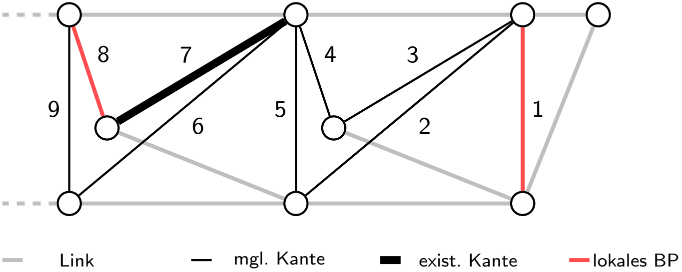
- Die eindeutige Beste-Antwort-Sequenz hat Länge $2^{\Omega(n)}$.
- Die zufällige Dynamik wählt eine beste Antwort mit Wkeit mind. 1/2, dadurch erwartete Anzahl von 1.5 Erzeugungen bis eine Kante zum nächsten Block vordringt.
- Bei b Blöcken ergibt sich eine Konvergenzzeit von mind. $1.5^b = 2^{\Omega(n)}$.

Bitzähler und Verbessungsschritte



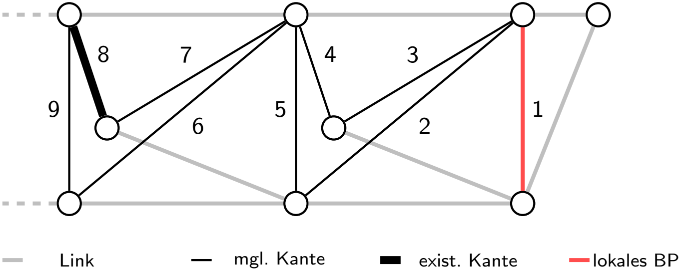
- Die eindeutige Beste-Antwort-Sequenz hat Länge $2^{\Omega(n)}$.
- Die zufällige Dynamik wählt eine beste Antwort mit Wkeit mind. 1/2, dadurch erwartete Anzahl von 1.5 Erzeugungen bis eine Kante zum nächsten Block vordringt.
- Bei b Blöcken ergibt sich eine Konvergenzzeit von mind. $1.5^b = 2^{\Omega(n)}$.

Bitzähler und Verbessungsschritte



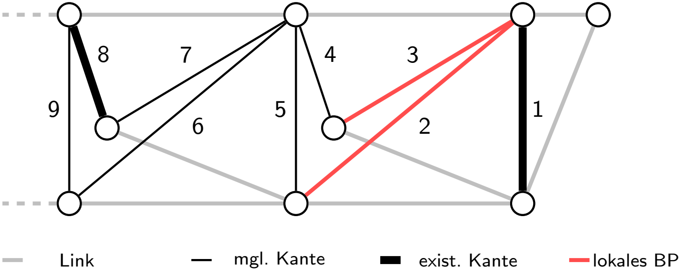
- Die eindeutige Beste-Antwort-Sequenz hat Länge $2^{\Omega(n)}$.
- Die zufällige Dynamik wählt eine beste Antwort mit Wkeit mind. 1/2, dadurch erwartete Anzahl von 1.5 Erzeugungen bis eine Kante zum nächsten Block vordringt.
- Bei b Blöcken ergibt sich eine Konvergenzzeit von mind. $1.5^b = 2^{\Omega(n)}$.

Bitzähler und Verbessungsschritte



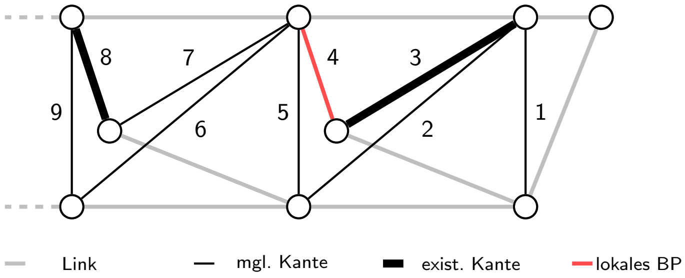
- Die eindeutige Beste-Antwort-Sequenz hat Länge $2^{\Omega(n)}$.
- Die zufällige Dynamik wählt eine beste Antwort mit Wkeit mind. 1/2, dadurch erwartete Anzahl von 1.5 Erzeugungen bis eine Kante zum nächsten Block vordringt.
- Bei b Blöcken ergibt sich eine Konvergenzzeit von mind. $1.5^b = 2^{\Omega(n)}$.

Bitzähler und Verbessungsschritte



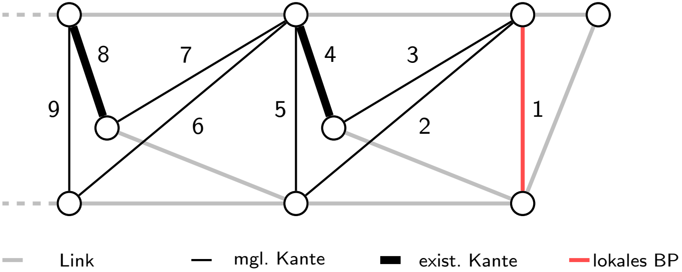
- Die eindeutige Beste-Antwort-Sequenz hat Länge $2^{\Omega(n)}$.
- Die zufällige Dynamik wählt eine beste Antwort mit Wkeit mind. 1/2, dadurch erwartete Anzahl von 1.5 Erzeugungen bis eine Kante zum nächsten Block vordringt.
- Bei b Blöcken ergibt sich eine Konvergenzzeit von mind. $1.5^b = 2^{\Omega(n)}$.

Bitzähler und Verbessungsschritte



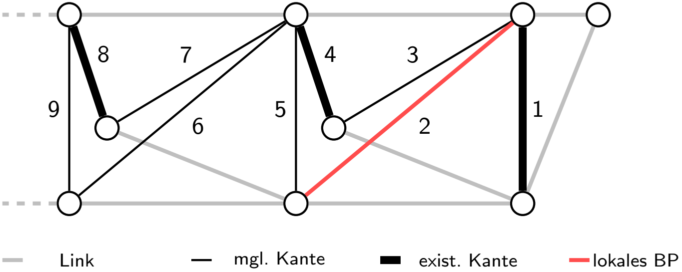
- Die eindeutige Beste-Antwort-Sequenz hat Länge $2^{\Omega(n)}$.
- Die zufällige Dynamik wählt eine beste Antwort mit Wkeit mind. 1/2, dadurch erwartete Anzahl von 1.5 Erzeugungen bis eine Kante zum nächsten Block vordringt.
- Bei b Blöcken ergibt sich eine Konvergenzzeit von mind. $1.5^b = 2^{\Omega(n)}$.

Bitzähler und Verbessungsschritte



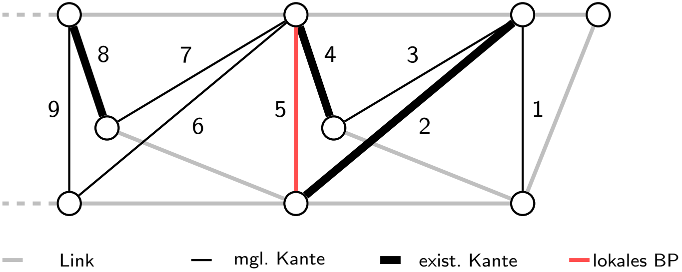
- Die eindeutige Beste-Antwort-Sequenz hat Länge $2^{\Omega(n)}$.
- Die zufällige Dynamik wählt eine beste Antwort mit Wkeit mind. 1/2, dadurch erwartete Anzahl von 1.5 Erzeugungen bis eine Kante zum nächsten Block vordringt.
- Bei b Blöcken ergibt sich eine Konvergenzzeit von mind. $1.5^b = 2^{\Omega(n)}$.

Bitzähler und Verbessungsschritte



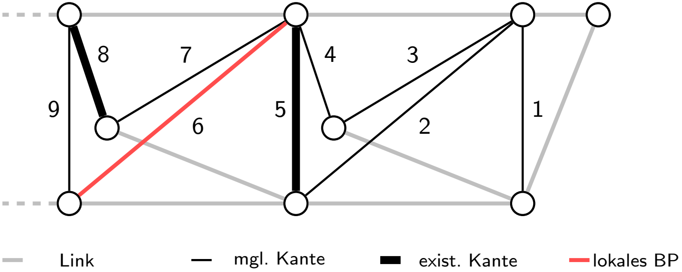
- Die eindeutige Beste-Antwort-Sequenz hat Länge $2^{\Omega(n)}$.
- Die zufällige Dynamik wählt eine beste Antwort mit Wkeit mind. 1/2, dadurch erwartete Anzahl von 1.5 Erzeugungen bis eine Kante zum nächsten Block vordringt.
- Bei b Blöcken ergibt sich eine Konvergenzzeit von mind. $1.5^b = 2^{\Omega(n)}$.

Bitzähler und Verbessungsschritte



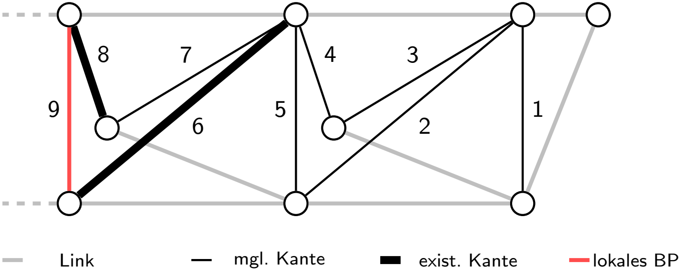
- Die eindeutige Beste-Antwort-Sequenz hat Länge $2^{\Omega(n)}$.
- Die zufällige Dynamik wählt eine beste Antwort mit Wkeit mind. $1/2$, dadurch erwartete Anzahl von 1.5 Erzeugungen bis eine Kante zum nächsten Block vordringt.
- Bei b Blöcken ergibt sich eine Konvergenzzeit von mind. $1.5^b = 2^{\Omega(n)}$.

Bitzähler und Verbessungsschritte



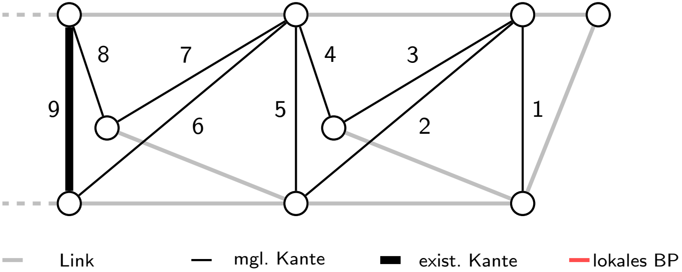
- Die eindeutige Beste-Antwort-Sequenz hat Länge $2^{\Omega(n)}$.
- Die zufällige Dynamik wählt eine beste Antwort mit Wkeit mind. 1/2, dadurch erwartete Anzahl von 1.5 Erzeugungen bis eine Kante zum nächsten Block vordringt.
- Bei b Blöcken ergibt sich eine Konvergenzzeit von mind. $1.5^b = 2^{\Omega(n)}$.

Bitzähler und Verbessungsschritte



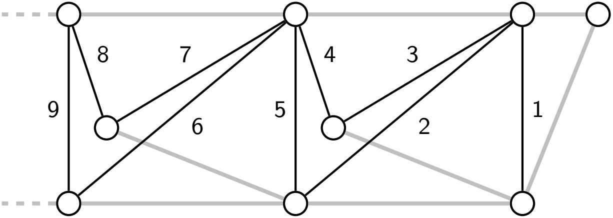
- Die eindeutige Beste-Antwort-Sequenz hat Länge $2^{\Omega(n)}$.
- Die zufällige Dynamik wählt eine beste Antwort mit Wkeit mind. 1/2, dadurch erwartete Anzahl von 1.5 Erzeugungen bis eine Kante zum nächsten Block vordringt.
- Bei b Blöcken ergibt sich eine Konvergenzzeit von mind. $1.5^b = 2^{\Omega(n)}$.

Bitzähler und Verbessungsschritte



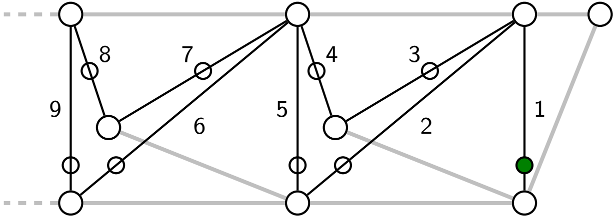
- Die eindeutige Beste-Antwort-Sequenz hat Länge $2^{\Omega(n)}$.
- Die zufällige Dynamik wählt eine beste Antwort mit Wkeit mind. 1/2, dadurch erwartete Anzahl von 1.5 Erzeugungen bis eine Kante zum nächsten Block vordringt.
- Bei b Blöcken ergibt sich eine Konvergenzzeit von mind. $1.5^b = 2^{\Omega(n)}$.

Existenz einer kurzen Sequenz



Graph der Kantenbewegungen

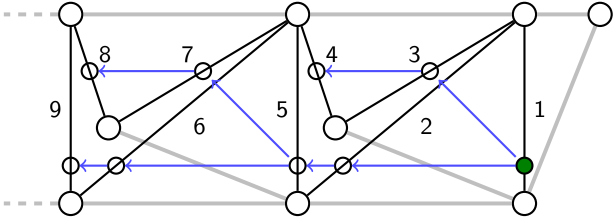
Existenz einer kurzen Sequenz



Graph der Kantenbewegungen

- Einen Knoten für jede mögliche Kante
- Startpunkt e: Lokales Blocking Pair für $M = \emptyset$

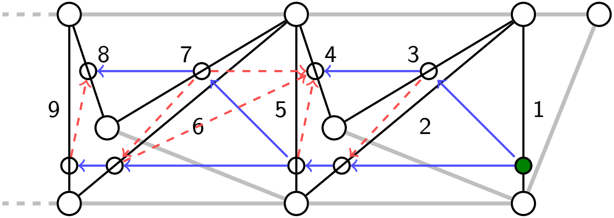
Existenz einer kurzen Sequenz



Graph der Kantenbewegungen

- Einen Knoten für jede mögliche Kante
- Startpunkt e: Lokales Blocking Pair für $M = \emptyset$
- Bewegungskante (e, e'): e' wird lokales BP wenn e existiert

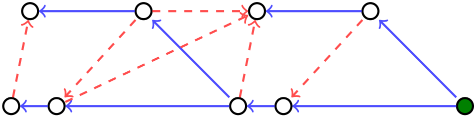
Existenz einer kurzen Sequenz



Graph der Kantenbewegungen

- Einen Knoten für jede mögliche Kante
- Startpunkt e: Lokales Blocking Pair für $M = \emptyset$
- Bewegungskante (e, e'): e' wird lokales BP wenn e existiert
- Dominierungskante (e, e'): e' kann kein BP sein wenn e existiert.

Existenz einer kurzen Sequenz

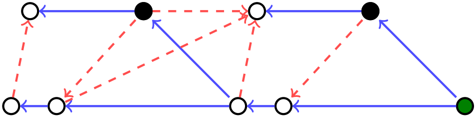


Existierende Kanten sind Knotenmarkierungen

Konstruiere Sequenz der Länge $O(nm^2)$ wie folgt:

1. Verbessere existierende Markierungen ohne neue zu erstellen. (in $O(nm)$)
2. Bewege Markierung zu einer dominiernden und weiter bei 1. (in $O(nm^2)$)
3. Erstelle neue stabile Markierungen über die Startpunkte. (in $O(nm)$)

Existenz einer kurzen Sequenz

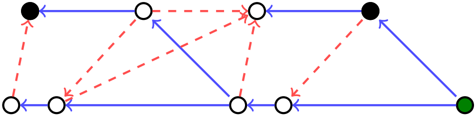


Existierende Kanten sind Knotenmarkierungen

Konstruiere Sequenz der Länge $O(nm^2)$ wie folgt:

1. Verbessere existierende Markierungen ohne neue zu erstellen. (in $O(nm)$)
2. Bewege Markierung zu einer dominiernden und weiter bei 1. (in $O(nm^2)$)
3. Erstelle neue stabile Markierungen über die Startpunkte. (in $O(nm)$)

Existenz einer kurzen Sequenz

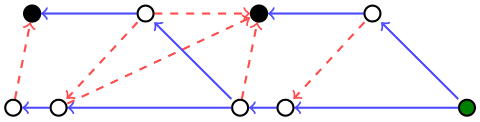


Existierende Kanten sind Knotenmarkierungen

Konstruiere Sequenz der Länge $O(nm^2)$ wie folgt:

1. Verbessere existierende Markierungen ohne neue zu erstellen. (in $O(nm)$)
2. Bewege Markierung zu einer dominiernden und weiter bei 1. (in $O(nm^2)$)
3. Erstelle neue stabile Markierungen über die Startpunkte. (in $O(nm)$)

Existenz einer kurzen Sequenz

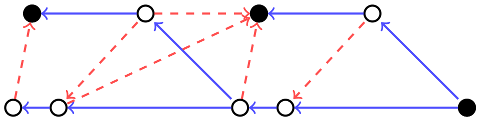


Existierende Kanten sind Knotenmarkierungen

Konstruiere Sequenz der Länge $O(nm^2)$ wie folgt:

1. Verbessere existierende Markierungen ohne neue zu erstellen. (in $O(nm)$)
2. Bewege Markierung zu einer dominierndden und weiter bei 1. (in $O(nm^2)$)
3. Erstelle neue stabile Markierungen über die Startpunkte. (in $O(nm)$)

Existenz einer kurzen Sequenz

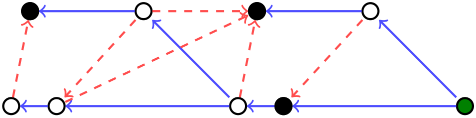


Existierende Kanten sind Knotenmarkierungen

Konstruiere Sequenz der Länge $O(nm^2)$ wie folgt:

1. Verbessere existierende Markierungen ohne neue zu erstellen. (in $O(nm)$)
2. Bewege Markierung zu einer dominiernden und weiter bei 1. (in $O(nm^2)$)
3. Erstelle neue stabile Markierungen über die Startpunkte. (in $O(nm)$)

Existenz einer kurzen Sequenz

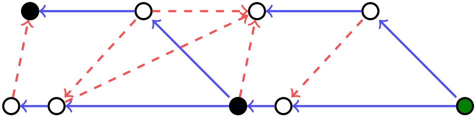


Existierende Kanten sind Knotenmarkierungen

Konstruiere Sequenz der Länge $O(nm^2)$ wie folgt:

1. Verbessere existierende Markierungen ohne neue zu erstellen. (in $O(nm)$)
2. Bewege Markierung zu einer dominiernden und weiter bei 1. (in $O(nm^2)$)
3. Erstelle neue stabile Markierungen über die Startpunkte. (in $O(nm)$)

Existenz einer kurzen Sequenz

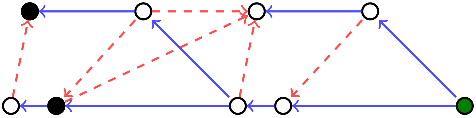


Existierende Kanten sind Knotenmarkierungen

Konstruiere Sequenz der Länge $O(nm^2)$ wie folgt:

1. Verbessere existierende Markierungen ohne neue zu erstellen. (in $O(nm)$)
2. Bewege Markierung zu einer dominierndne und weiter bei 1. (in $O(nm^2)$)
3. Erstelle neue stabile Markierungen über die Startpunkte. (in $O(nm)$)

Existenz einer kurzen Sequenz

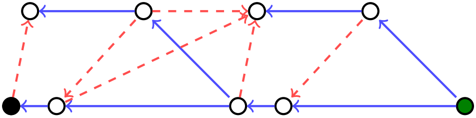


Existierende Kanten sind Knotenmarkierungen

Konstruiere Sequenz der Länge $O(nm^2)$ wie folgt:

1. Verbessere existierende Markierungen ohne neue zu erstellen. (in $O(nm)$)
2. Bewege Markierung zu einer dominiernden und weiter bei 1. (in $O(nm^2)$)
3. Erstelle neue stabile Markierungen über die Startpunkte. (in $O(nm)$)

Existenz einer kurzen Sequenz

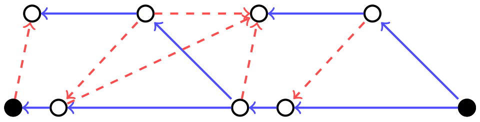


Existierende Kanten sind Knotenmarkierungen

Konstruiere Sequenz der Länge $O(nm^2)$ wie folgt:

1. Verbessere existierende Markierungen ohne neue zu erstellen. (in $O(nm)$)
2. Bewege Markierung zu einer dominiernden und weiter bei 1. (in $O(nm^2)$)
3. Erstelle neue stabile Markierungen über die Startpunkte. (in $O(nm)$)

Existenz einer kurzen Sequenz

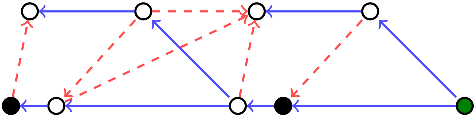


Existierende Kanten sind Knotenmarkierungen

Konstruiere Sequenz der Länge $O(nm^2)$ wie folgt:

1. Verbessere existierende Markierungen ohne neue zu erstellen. (in $O(nm)$)
2. Bewege Markierung zu einer dominiernden und weiter bei 1. (in $O(nm^2)$)
3. Erstelle neue stabile Markierungen über die Startpunkte. (in $O(nm)$)

Existenz einer kurzen Sequenz

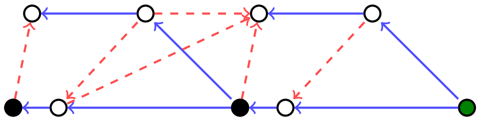


Existierende Kanten sind Knotenmarkierungen

Konstruiere Sequenz der Länge $O(nm^2)$ wie folgt:

1. Verbessere existierende Markierungen ohne neue zu erstellen. (in $O(nm)$)
2. Bewege Markierung zu einer dominiernden und weiter bei 1. (in $O(nm^2)$)
3. Erstelle neue stabile Markierungen über die Startpunkte. (in $O(nm)$)

Existenz einer kurzen Sequenz

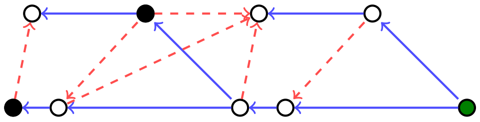


Existierende Kanten sind Knotenmarkierungen

Konstruiere Sequenz der Länge $O(nm^2)$ wie folgt:

1. Verbessere existierende Markierungen ohne neue zu erstellen. (in $O(nm)$)
2. Bewege Markierung zu einer dominiernden und weiter bei 1. (in $O(nm^2)$)
3. Erstelle neue stabile Markierungen über die Startpunkte. (in $O(nm)$)

Existenz einer kurzen Sequenz

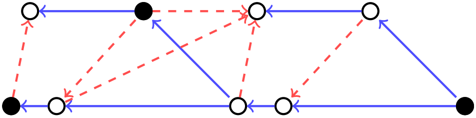


Existierende Kanten sind Knotenmarkierungen

Konstruiere Sequenz der Länge $O(nm^2)$ wie folgt:

1. Verbessere existierende Markierungen ohne neue zu erstellen. (in $O(nm)$)
2. Bewege Markierung zu einer dominiernden und weiter bei 1. (in $O(nm^2)$)
3. Erstelle neue stabile Markierungen über die Startpunkte. (in $O(nm)$)

Existenz einer kurzen Sequenz

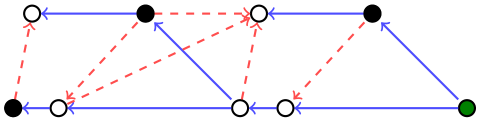


Existierende Kanten sind Knotenmarkierungen

Konstruiere Sequenz der Länge $O(nm^2)$ wie folgt:

1. Verbessere existierende Markierungen ohne neue zu erstellen. (in $O(nm)$)
2. Bewege Markierung zu einer dominiernden und weiter bei 1. (in $O(nm^2)$)
3. Erstelle neue stabile Markierungen über die Startpunkte. (in $O(nm)$)

Existenz einer kurzen Sequenz



Existierende Kanten sind Knotenmarkierungen

Konstruiere Sequenz der Länge $O(nm^2)$ wie folgt:

1. Verbessere existierende Markierungen ohne neue zu erstellen. (in $O(nm)$)
2. Bewege Markierung zu einer dominiernden und weiter bei 1. (in $O(nm^2)$)
3. Erstelle neue stabile Markierungen über die Startpunkte. (in $O(nm)$)

Zusammenfassung

Lokal Stabiles Matching

- Matching mit unvollständiger und dynamischer Information
- Exploration und Optimierung in Netzwerken

Strikte Präferenzen

- Erreichbarkeit ist NP-hart
- Bessere Resultate für (Zufalls-)Memory

Gewichtetes Matching

- Konvergenz in Poly-Zeit mit globaler Information
- Exponentielle Zeit mit lokalen Dynamiken
- Exponentieller Speed-Up mit Zufalls-Memory

Vielen Dank für die Aufmerksamkeit!