Evolutionary Algorithms for Sorting and Shortest Path Computation

Ferienakademie im Sarntal — Kurs 1
Moderne Suchmethoden der Informatik: Trends und Potenzial

Dominik Schmid

Fakultät für Mathematik
TU München

24. September 2014

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

Technische Universität München

Universität Stuttgart

Agenda

Bubblesort

Array von Keys:

3	6	5	1	4	2

Gesucht: Methode beliebiges Array zu sortieren

Bubblesort

Array von Keys:

3	6	5	1	4	2

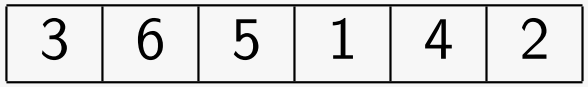
Gesucht: Methode beliebiges Array zu sortieren

Wiederhole solange bis Sortiert:

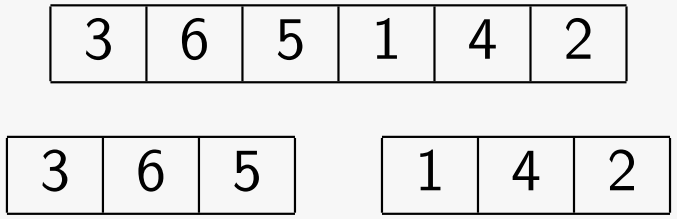
1. Vergleiche zwei benachbarte Zahlen
2. Tausche diese wenn links $>$ rechts
3. Betrachte nun die nächsten zwei Zahlen

$\Rightarrow$ Worst-Case Laufzeit in $\Theta(n^2)$

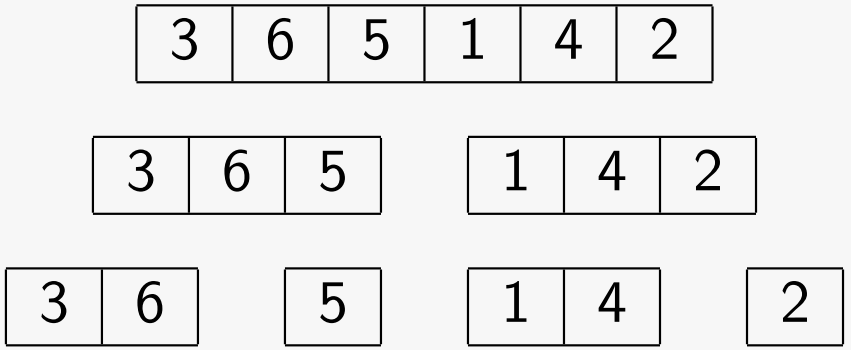
Mergesort



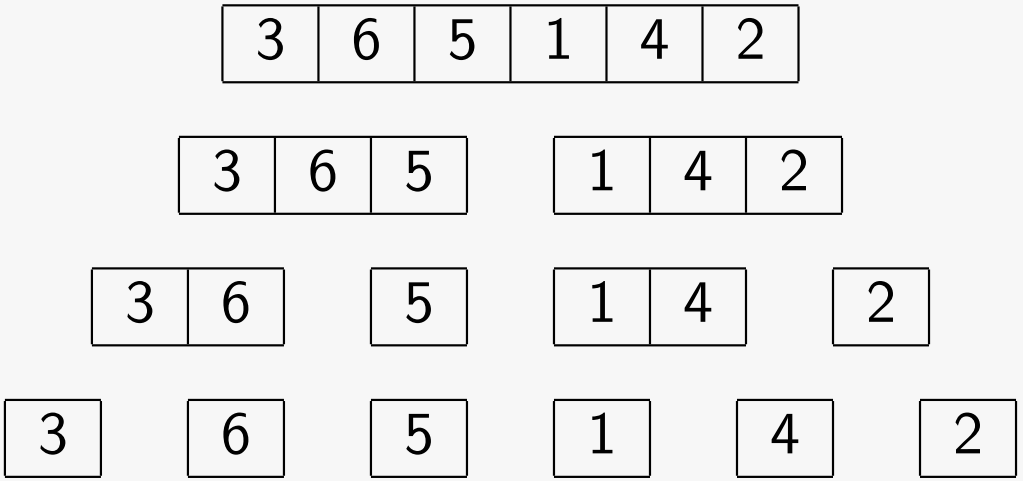
Mergesort



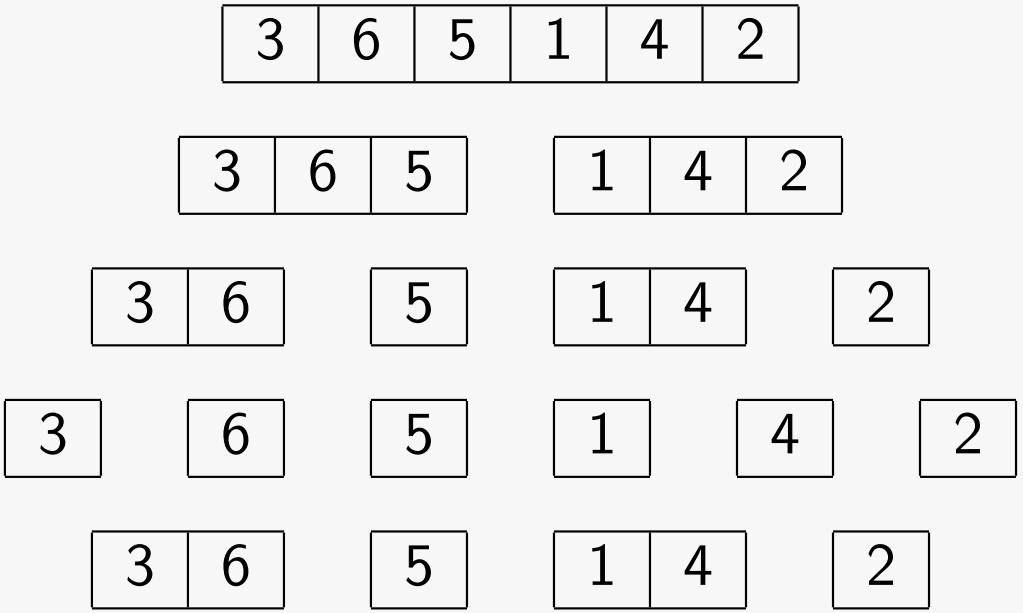
Mergesort



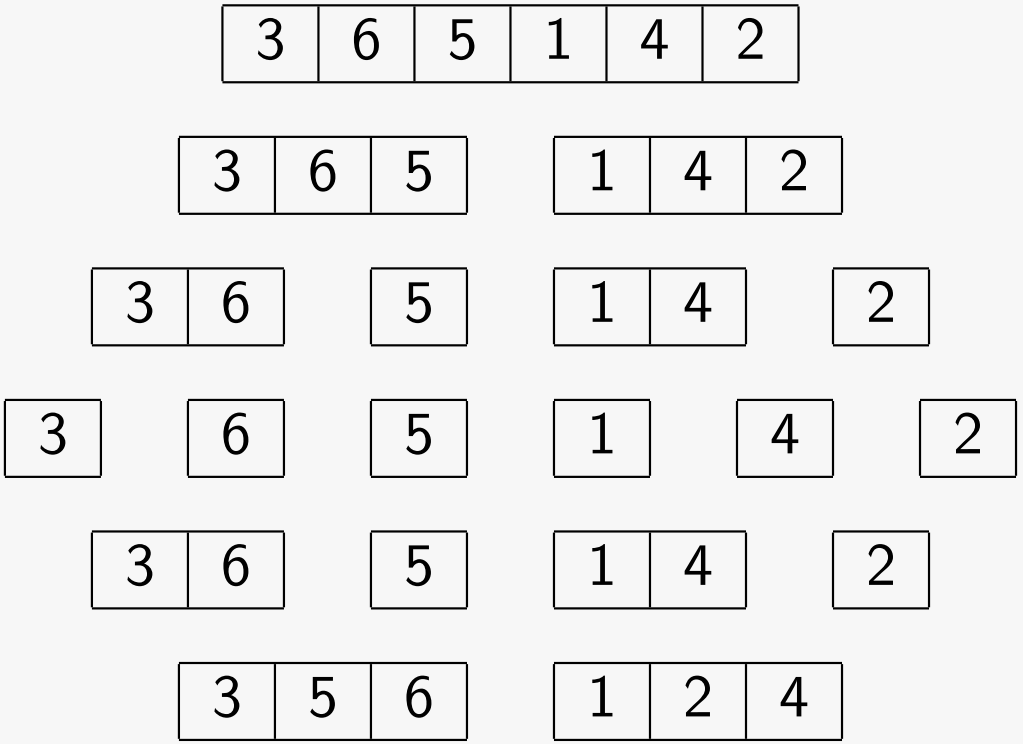
Mergesort



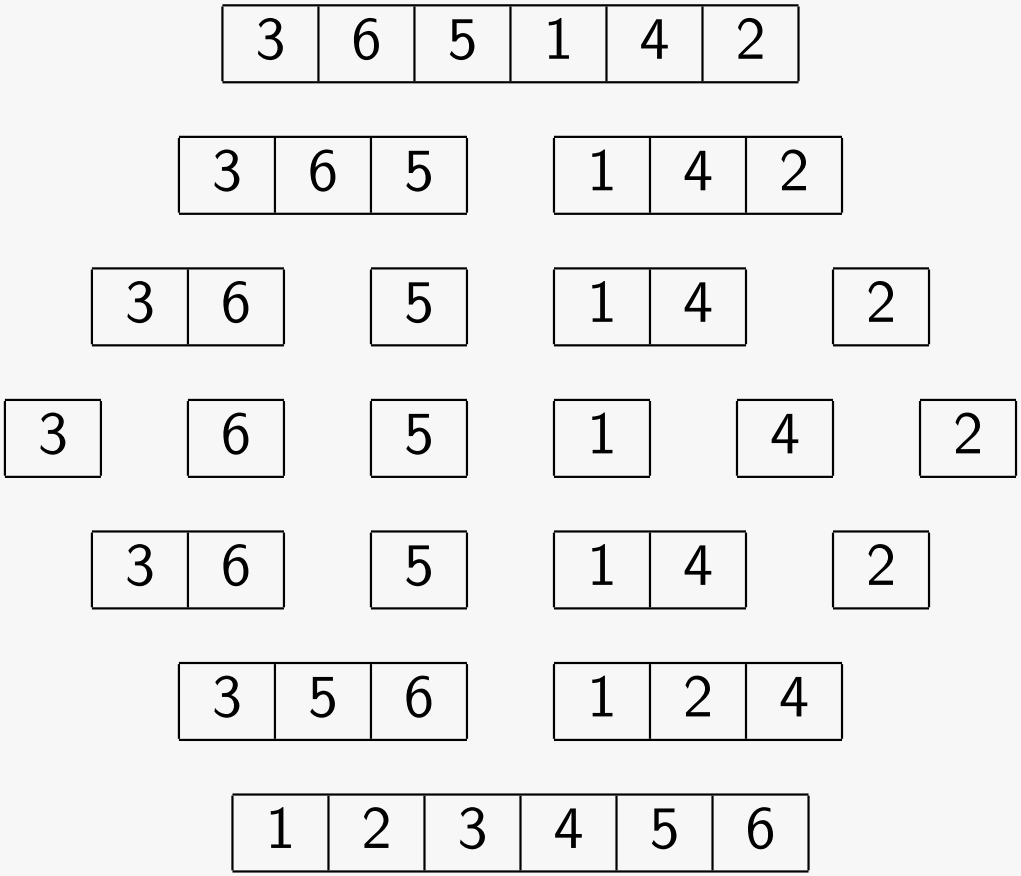
Mergesort



Mergesort

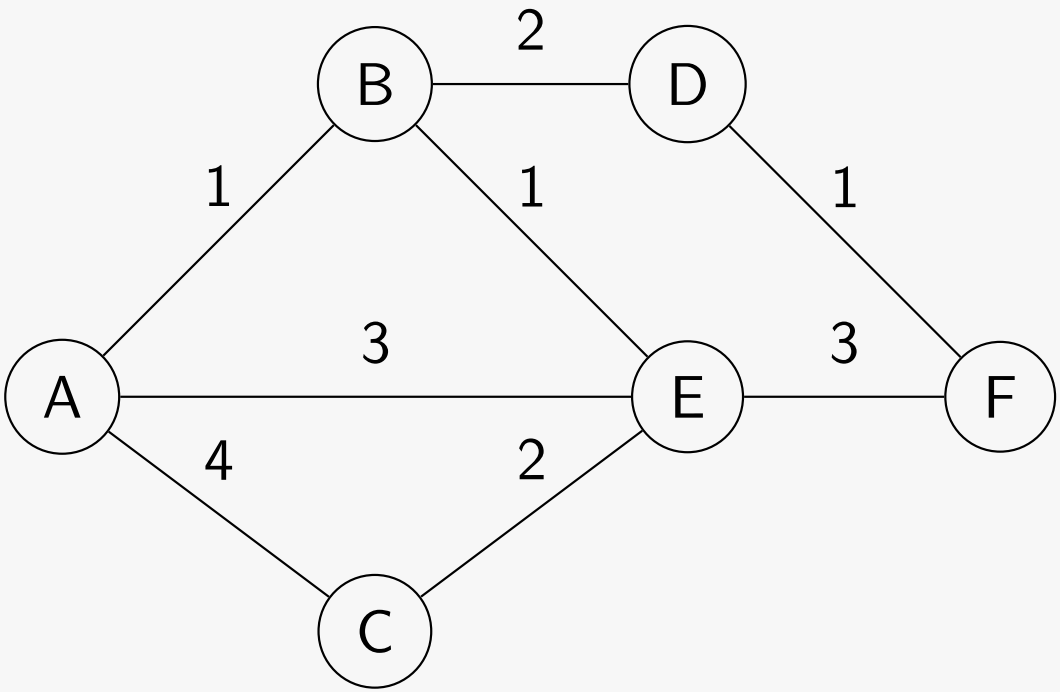


Mergesort



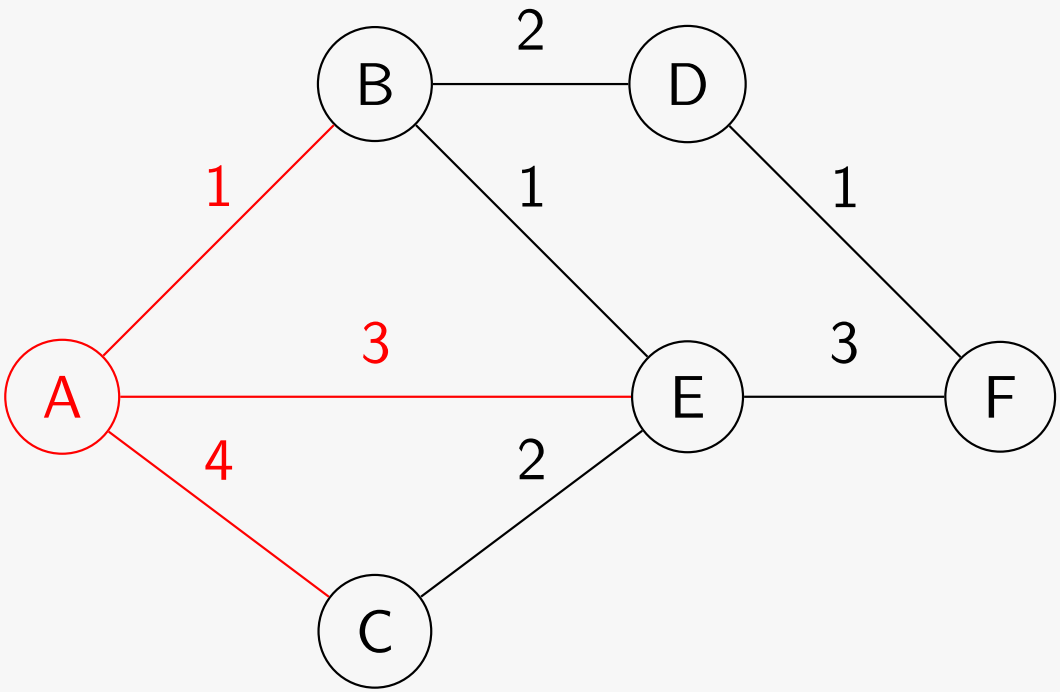
$\Rightarrow$ Worst-Case Laufzeit in $\Theta(n \log n)$

Dijkstra



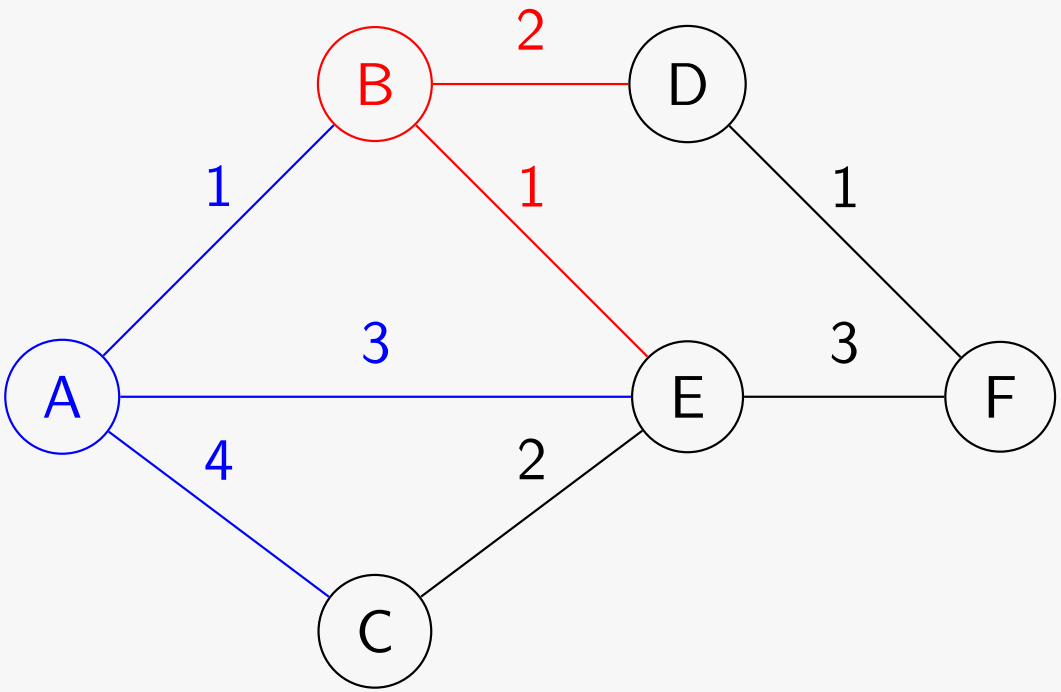
A B C D E F

Dijkstra



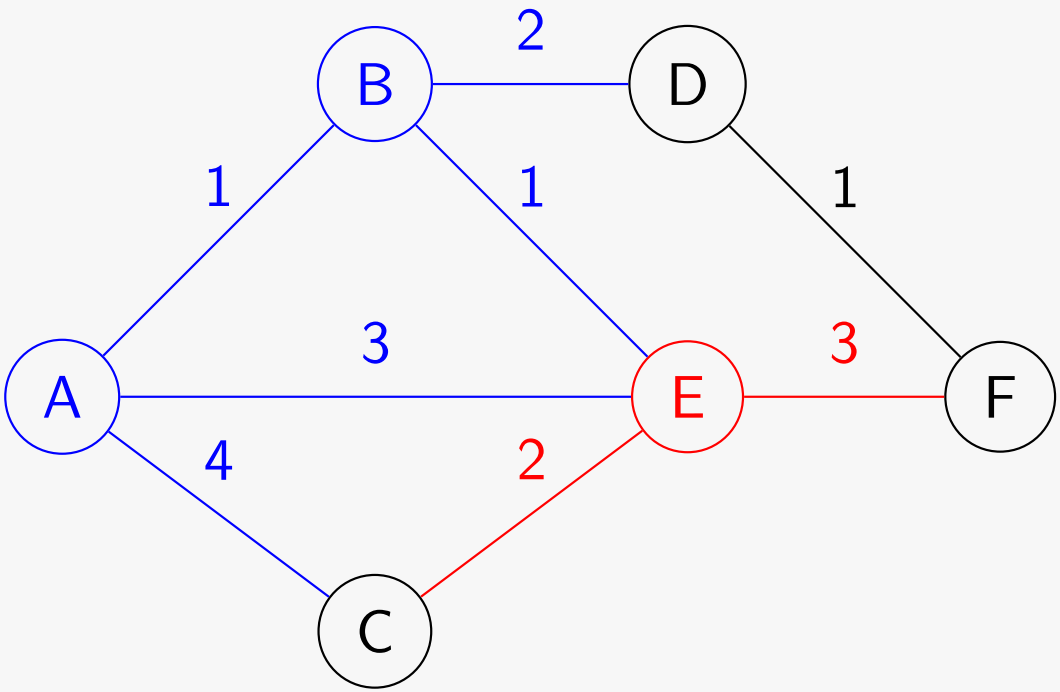
	A	B	C	D	E	F
Update A	A(0)	A(1)	A(4)		A(3)	

Dijkstra



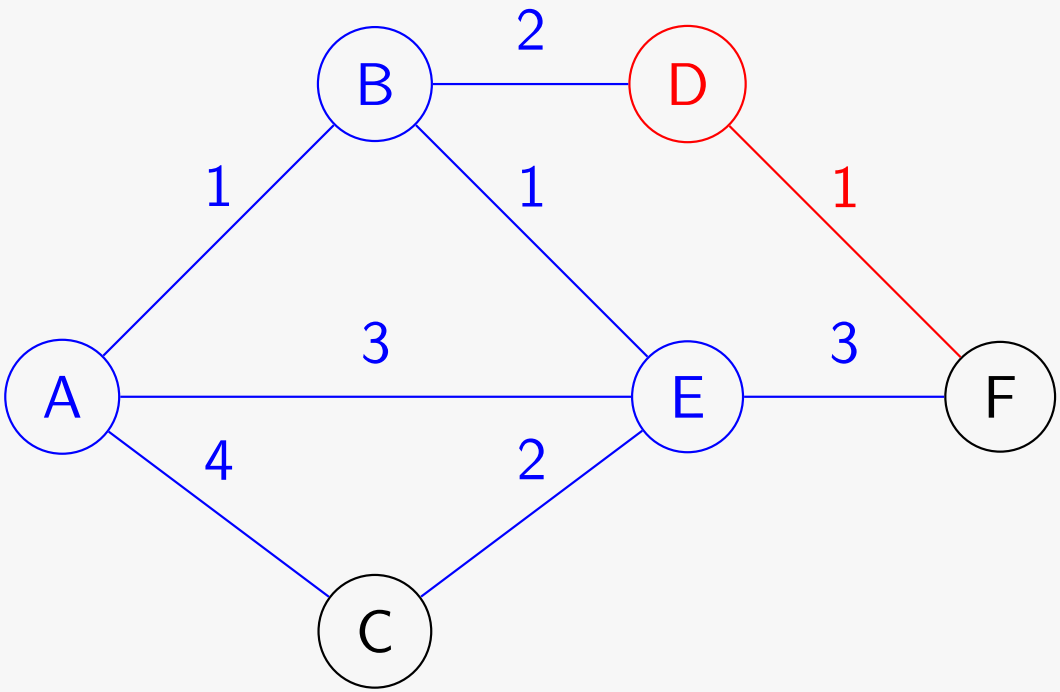
	A	B	C	D	E	F
Update A	A(0)	A(1)	A(4)		A(3)	
Update B	A(0)	A(1)	A(4)	B(3)	B(2)	

Dijkstra



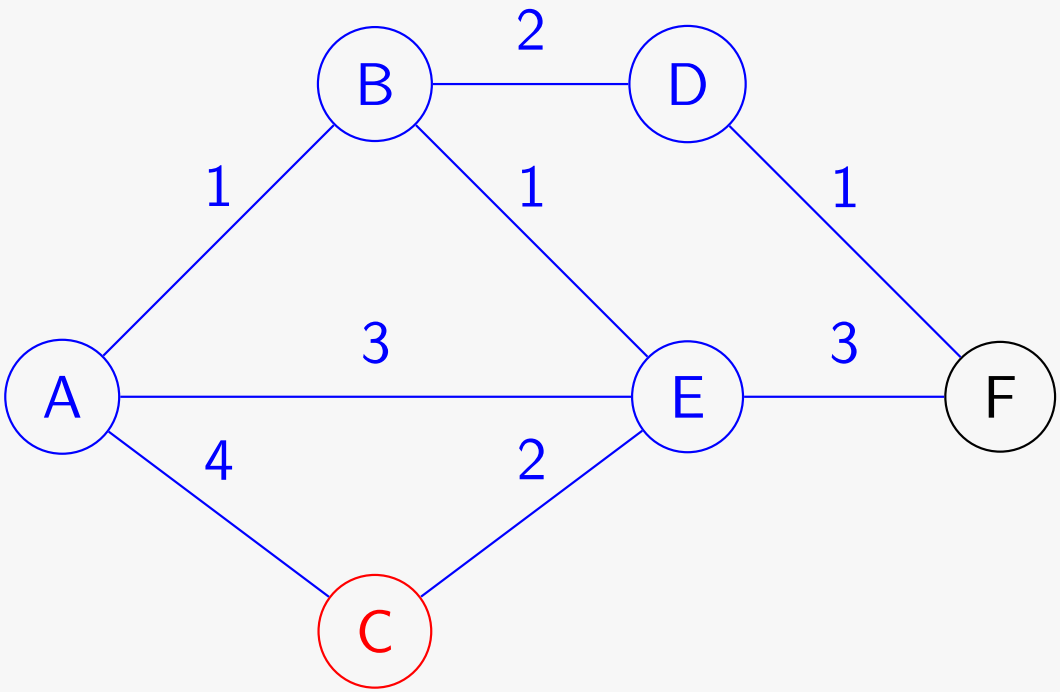
	A	B	C	D	E	F
Update A	A(0)	A(1)	A(4)		A(3)	
Update B	A(0)	A(1)	A(4)	B(3)	B(2)	
Update E	A(0)	A(1)	A(4)	B(3)	B(2)	E(5)

Dijkstra



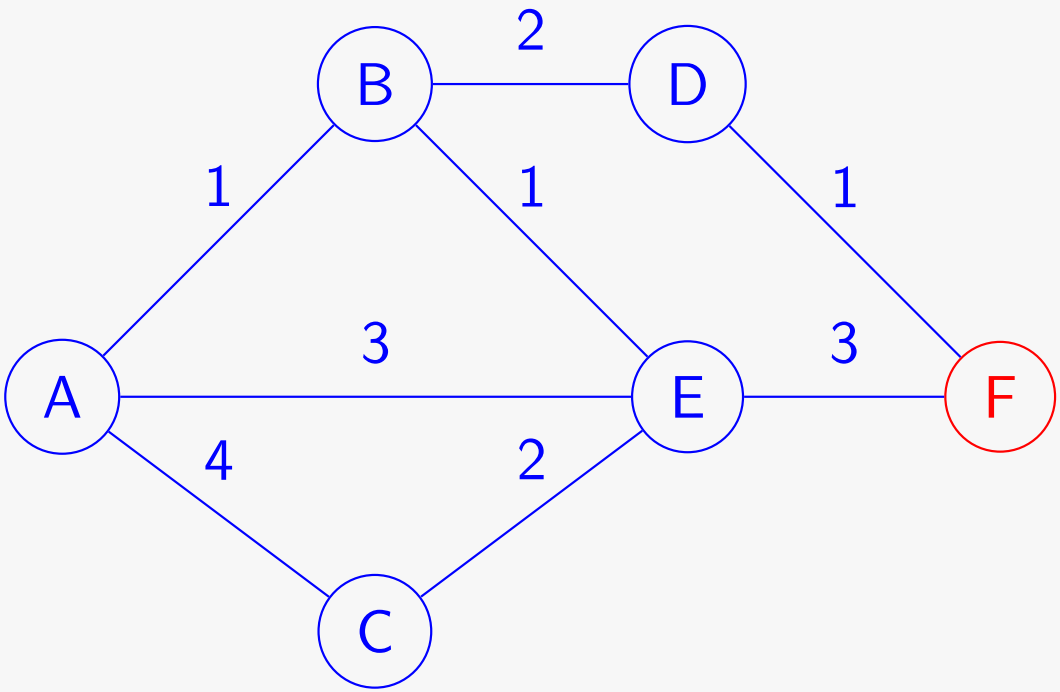
	A	B	C	D	E	F
Update A	A(0)	A(1)	A(4)		A(3)	
Update B	A(0)	A(1)	A(4)	B(3)	B(2)	
Update E	A(0)	A(1)	A(4)	B(3)	B(2)	E(5)
Update D	A(0)	A(1)	A(4)	B(3)	B(2)	D(4)

Dijkstra



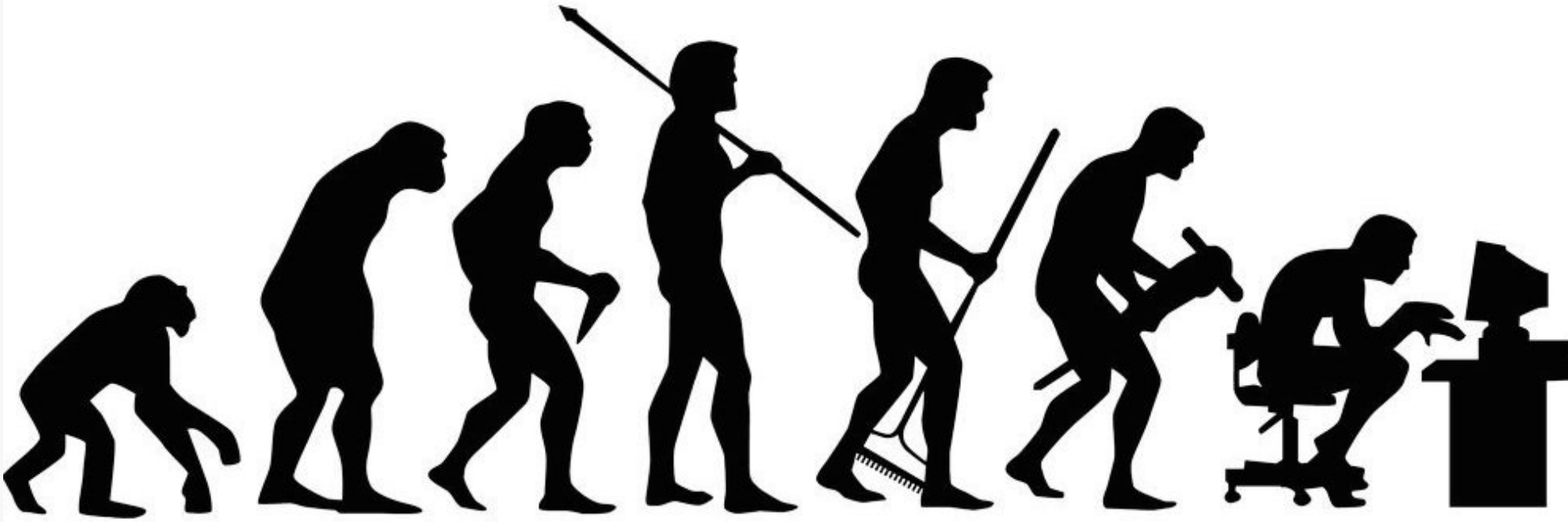
	A	B	C	D	E	F
Update A	A(0)	A(1)	A(4)		A(3)	
Update B	A(0)	A(1)	A(4)	B(3)	B(2)	
Update E	A(0)	A(1)	A(4)	B(3)	B(2)	E(5)
Update D	A(0)	A(1)	A(4)	B(3)	B(2)	D(4)
Update C	A(0)	A(1)	A(4)	B(3)	B(2)	D(4)

Dijkstra

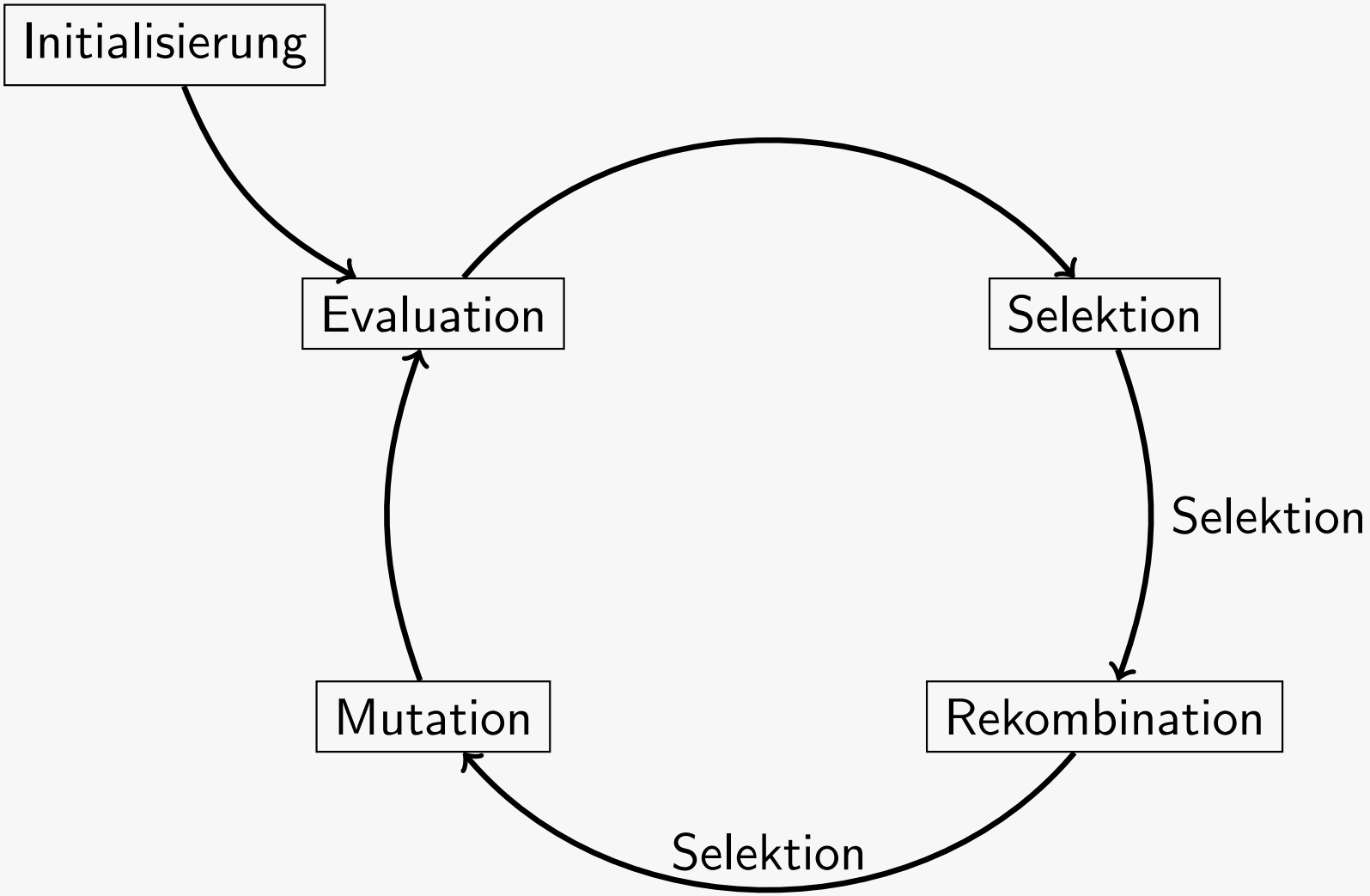


	A	B	C	D	E	F
Update A	A(0)	A(1)	A(4)		A(3)	
Update B	A(0)	A(1)	A(4)	B(3)	B(2)	
Update E	A(0)	A(1)	A(4)	B(3)	B(2)	E(5)
Update D	A(0)	A(1)	A(4)	B(3)	B(2)	D(4)
Update C	A(0)	A(1)	A(4)	B(3)	B(2)	D(4)
Update F	A(0)	A(1)	A(4)	B(3)	B(2)	D(4)

Agenda



Evolutionäre Algorithmen



EA als Sortierverfahren

Ziel: Maximiere den Grad der Sortiertheit bzgl. $\boldsymbol{\pi_{\mathbf{opt}}}$

EA als Sortierverfahren

Ziel: Maximiere den Grad der Sortiertheit bzgl. $\boldsymbol{\pi_{opt}}$

Ausgangssituation:
Gegebene zufällige Permutation $\boldsymbol{\pi_1}$ der Elemente 1 bis n (Set)

Welche Möglichkeiten für

- Mutationsoperatoren
- Fitnessfunktionen

Mutationsoperatoren

Nur Mutationen für Sortieren

Mutationsoperatoren

Nur Mutationen für Sortieren

Bisher:

Bubblesort mit $swap(i)$ Operation

Mutationsoperatoren

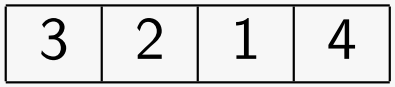
Nur Mutationen für Sortieren

Bisher:

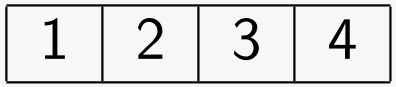
Bubblesort mit *swap*(i) Operation

Jetzt:

exchange(i,j)



exchange(3,1) $\rightarrow$



Mutationsoperatoren

Nur Mutationen für Sortieren

Bisher:

Bubblesort mit *swap*(i) Operation

Jetzt:

exchange(i,j)

3	2	1	4

exchange(3,1) →

1	2	3	4

jump(i,j)

1	3	4	2

jump(4,2) →

1	2	3	4

Mutationsoperatoren

Nur Mutationen für Sortieren

Bisher:

Bubblesort mit *swap*(i) Operation

Jetzt:

exchange(i,j)

3	2	1	4

exchange(3,1) →

1	2	3	4

jump(i,j)

1	3	4	2

jump(4,2) →

1	2	3	4

reverse(i,j)

4	3	2	1

reverse(1,4) →

1	2	3	4

Mutationsoperatoren

Nur Mutationen für Sortieren

Bisher:

Bubblesort mit *swap*(i) Operation

Jetzt:

exchange(i,j)

3	2	1	4

exchange(3,1) →

1	2	3	4

jump(i,j)

1	3	4	2

jump(4,2) →

1	2	3	4

reverse(i,j)

4	3	2	1

reverse(1,4) →

1	2	3	4

Analyse nur für exchange und jump Operatoren!

Fitnessfunktionen

Fitnessfunktion $f_{\pi_{opt}}(\pi)$ zu $f(\pi) := f_{id}(\pi)$ vereinfachbar

Fitnessfunktionen

Fitnessfunktion $f_{\pi_{opt}}(\pi)$ zu $f(\pi) := f_{id}(\pi)$ vereinfachbar

Fitnessfunktion	Beschreibung
INV	$\#(i,j) : \pi(i) < \pi(j)$, wobei $1 \leq i < j \leq n$

Fitnessfunktionen

Fitnessfunktion $f_{\pi_{opt}}(\pi)$ zu $f(\pi) := f_{id}(\pi)$ vereinfachbar

Fitnessfunktion	Beschreibung
INV	$\#(i,j) : \pi(i) < \pi(j)$, wobei $1 \leq i < j \leq n$
HAM	$\#i : \pi(i) = i$, $1 \leq i \leq n$

Fitnessfunktionen

Fitnessfunktion $f_{\pi_{opt}}(\pi)$ zu $f(\pi) := f_{id}(\pi)$ vereinfachbar

Fitnessfunktion	Beschreibung
INV	$\#(i,j) : \pi(i) < \pi(j)$, wobei $1 \leq i < j \leq n$
HAM	$\#i : \pi(i) = i$, $1 \leq i \leq n$
RUN	$1 + (\#i : \pi(i+1) < \pi(i))$

Fitnessfunktionen

Fitnessfunktion $f_{\pi_{opt}}(\pi)$ zu $f(\pi) := f_{id}(\pi)$ vereinfachbar

Fitnessfunktion	Beschreibung
INV	$\#(i,j) : \pi(i) < \pi(j)$, wobei $1 \leq i < j \leq n$
HAM	$\#i : \pi(i) = i$, $1 \leq i \leq n$
RUN	$1 + (\#i : \pi(i+1) < \pi(i))$
LAS	$\max(k : \pi(i_1) < \cdots < \pi(i_k)$ mit $i_1 < \cdots < i_k)$

Fitnessfunktionen

Fitnessfunktion $f_{\pi_{opt}}(\pi)$ zu $f(\pi) := f_{id}(\pi)$ vereinfachbar

Fitnessfunktion	Beschreibung
INV	$\#(i,j) : \pi(i) < \pi(j)$, wobei $1 \leq i < j \leq n$
HAM	$\#i : \pi(i) = i$, $1 \leq i \leq n$
RUN	$1 + (\#i : \pi(i+1) < \pi(i))$
LAS	$\max(k : \pi(i_1) < \cdots < \pi(i_k)$ mit $i_1 < \cdots < i_k)$
EXC	$\min(\ \#\ exchange(i,j)\ : \pi = id)$

(1+1) EA für Permutationen

Mutationsoperator

Wähle k gemäß der Poissonverteilung $Poi_\lambda(k) = \frac{\lambda^k}{k!}e^{-\lambda}$ mit $\lambda = 1$ und führe sequentiell $k+1$ *jump/exchange* Operationen unter zufällig ausgewählten Paaren $(i,j) : 1 \leq i,j \leq n$ und $i \neq j$ aus.

(1+1) EA für Permutationen

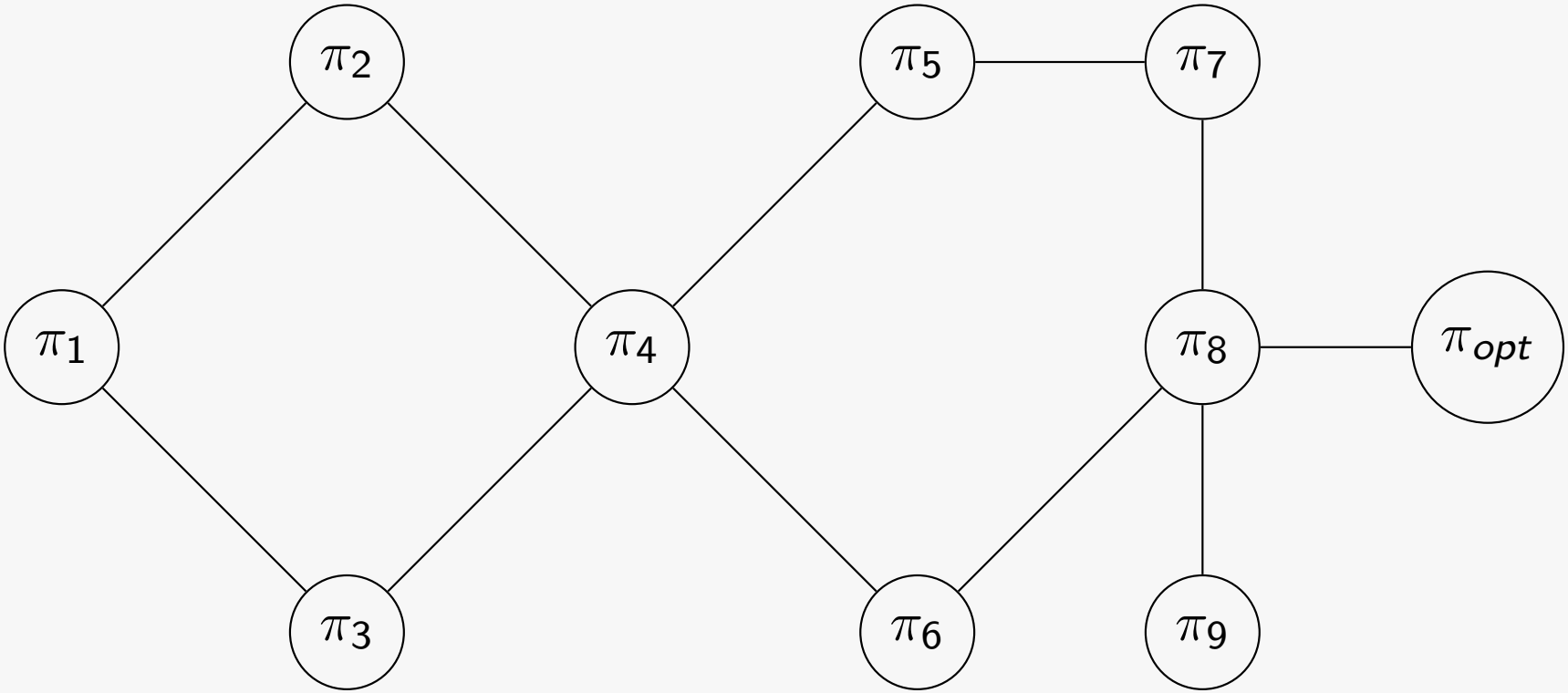
Mutationsoperator

Wähle k gemäß der Poissonverteilung $Poi_\lambda(k) = \frac{\lambda^k}{k!}e^{-\lambda}$ mit $\lambda = 1$ und führe sequentiell $k+1$ *jump/exchange* Operationen unter zufällig ausgewählten Paaren $(i,j) : 1 \leq i,j \leq n$ und $i \neq j$ aus.

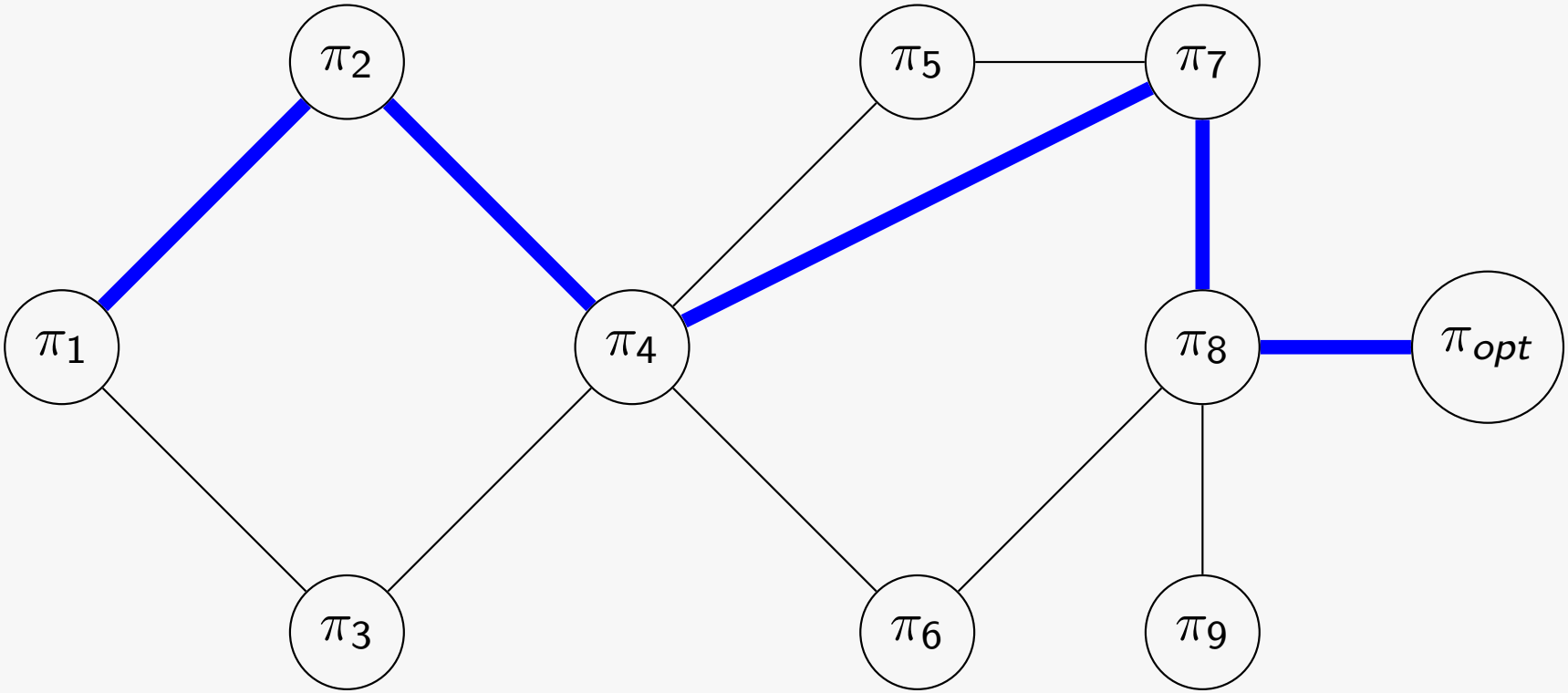
Algorithmus des (1+1) EA

- Initialisiere zufällige Permutation π
- Wiederhole:
  1. Erstelle π' durch Mutation von π
  2. Ersetze π durch π' falls $f(\pi) \leq f(\pi')$ bei Maximierungsproblem und falls $f(\pi) \geq f(\pi')$ bei Minimierung
  3. Falls $\pi = \pi_{opt}$ Schleife abbrechen

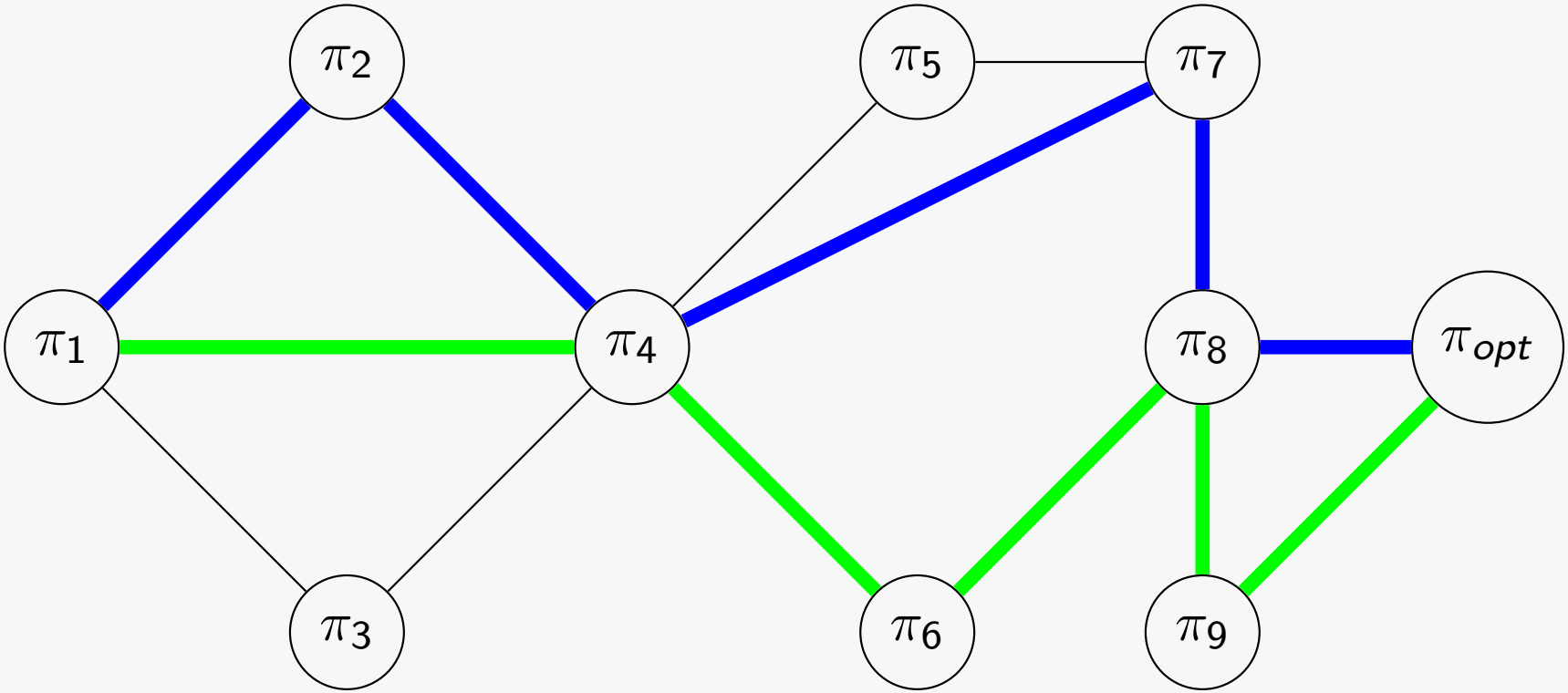
Visualisierung



Visualisierung



Visualisierung



\pause

Agenda

Theorem 1:

Die erwartete Laufzeit des (1+1) EA mit Fitnessfunktionen INV(π), HAM(π), RUN(π), LAS(π), EXC(π) liegt in $\Omega(n^2)$

Theorem 2:

Erwartete Laufzeit des (1+1) EA mit Fitnessfunktionen INV(π), HAM(π), LAS(π), EXC(π) liegt in $\mathcal{O}(n^2 \log n)$

Theorem 1: Untere Schranke der erwarteten Laufzeit des (1+1) EA

Beweis:

Annahme: Es existiert nur ein globales Maximum/Minimum

- $P_{\pi_{opt}} = \frac{1}{n!}$
- Betrachte nur den letzten Permutationsschritt: Maximal zwei "gute" *jump* und zwei "gute" *exchange* Operationen

$\rightarrow P_{opt} = \frac{1}{2} \cdot \frac{2}{n(n-1)} + \frac{1}{2} \cdot \frac{2}{n(n-1)} = \frac{2}{n(n-1)}$

$\rightarrow$ Waiting time nach unten durch $\frac{n(n-1)}{2}$ beschränkt

□

Theorem 2: Obere Schranke der erwarteten Laufzeit des (1+1) EA

Beweis für INV:

Sei (i, j) ein ïnkorrektes Paar"($\pi(j) < \pi(i)$, wobei $i < j$)

- $a := \#k : \pi(k) < \pi(j)$ für $k \in \{i+1, ..., j-1\}$
- $b := \#k : \pi(j) < \pi(k) < \pi(i)$ für $k \in \{i+1, ..., j-1\}$
- $c := \#k : \pi(i) < \pi(k)$ für $k \in \{i+1, ..., j-1\}$

Theorem 2: Obere Schranke der erwarteten Laufzeit des (1+1) EA

Beweis für INV:

Sei (i, j) ein ïnkorrektes Paar"($\pi(j) < \pi(i)$, wobei $i < j$)

- $a := \#k : \pi(k) < \pi(j)$ für $k \in \{i+1, ..., j-1\}$
- $b := \#k : \pi(j) < \pi(k) < \pi(i)$ für $k \in \{i+1, ..., j-1\}$
- $c := \#k : \pi(i) < \pi(k)$ für $k \in \{i+1, ..., j-1\}$

Fitnessveränderungen:

- exchange(i,j) = exchange(j,i) $\rightarrow 2b+1$
- jump(i,j) $\rightarrow a+b-c+1$
- jump(j,i) $\rightarrow -a+b+c+1$

Theorem 2: Obere Schranke der erwarteten Laufzeit des (1+1) EA

Beweis für INV (Fortsetzung):

$\Rightarrow$ Mindestens 3 Möglichkeiten um Fitness zu verbessern, wenn genau ein Increase durchgeführt wird (Wahrscheinlichkeit: $\frac{1}{e}$)

- $P_{exchange} = \frac{1}{n(n-1)}$
- $P_{jump} \geq \frac{1}{2n(n-1)}$

$\Rightarrow P_{increase} \geq \frac{3m}{2en(n-1)}$, wobei $m = \#$ inkorrekte Paare

Theorem 2: Obere Schranke der erwarteten Laufzeit des (1+1) EA

Beweis für INV (Fortsetzung):

$\Rightarrow$ Mindestens 3 Möglichkeiten um Fitness zu verbessern, wenn genau ein Increase durchgeführt wird (Wahrscheinlichkeit: $\frac{1}{e}$)

- $P_{exchange} = \frac{1}{n(n-1)}$
- $P_{jump} \geq \frac{1}{2n(n-1)}$

$\Rightarrow P_{increase} \geq \frac{3m}{2en(n-1)}$, wobei $m = \#$ inkorrekte Paare

Für die erwartete Laufzeit von INV folgt somit als obere Schranke

$$\frac{2}{3}en^2 \sum_{1 \leq m \leq n(n-1)/2} 1/m = \frac{2}{3}en^2 H(\frac{n(n-1)}{2}) = \mathcal{O}(n^2 \log n)$$

Theorem 2: Obere Schranke der erwarteten Laufzeit des (1+1) EA

Beweis für HAM:

Sei $\mathrm{HAM}(\pi) = k$, so gibt es $n - k$ Elemente an falscher Position

$\rightarrow$ Für jedes solche Element existiert passende *exchange* Operation

$\rightarrow$ Analog zum Beweis von INV folgt obere Schranke $2en^2H(n)$

Theorem 2: Obere Schranke der erwarteten Laufzeit des (1+1) EA

Beweis für HAM:

Sei $\text{HAM}(\pi) = k$, so gibt es $n - k$ Elemente an falscher Position

$\rightarrow$ Für jedes solche Element existiert passende *exchange* Operation

$\rightarrow$ Analog zum Beweis von INV folgt obere Schranke $2en^2H(n)$

Beweis für LAS:

Sei $\text{LAS}(\pi) = k$, so gibt es $n - k$ Elemente, die nicht in einer der längsten möglichen Sequenzen liegen

$\rightarrow$ Für jedes solche Element existiert passende *jump* Operation

$\rightarrow$ Analog zum Beweis von INV folgt obere Schranke $2en^2H(n)$

Theorem 2: Obere Schranke der erwarteten Laufzeit des (1+1) EA

Beweis für EXC:

Sei $\mathrm{EXC}(\pi) = k$, so gibt es $n - k$ Zyklen innerhalb der Permutation π

$\rightarrow$ Maximal $n - k - 1$ Zyklen haben Länge 1 bzw. mind. $k + 1$ Elemente in Zyklen mit Länge $l \geq 2$

$\rightarrow$ Für jedes dieser $k + 1$ Elemente existiert *exchange* Operation mit korrekter Position (Fitnesssteigerung!)

$\rightarrow$ Analog zum Beweis von INV folgt obere Schranke $2en^2 H(n)$

$\Rightarrow$ Für INV, HAM, LAS, EXC obere Schranke $\mathcal{O}(n^2 \log n)$ gezeigt

□

Theorem 3:

Die erwartete Laufzeit des (1+1) EA nur mit *exchange* Operationen liegt für Fitnessfunktionen HAM(π) und EXC(π) in $\Theta(n^2 \log n)$

Theorem 3: Erwartete Laufzeit des (1+1) EA mit HAM(π) und EXC(π)

Vorgehen für untere Grenze:

1. Lokale Operation ändert Fitness maximal um Konstante c
2. $\exists k_0(n) : \forall k \leq k_0(n)$ gilt:
   Die Wahrscheinlichkeit für eine Fitnesserhöhung liegt in $\mathcal{O}(\frac{k}{n^2})$

$\Rightarrow$ Wenn $k \in \Omega(n^\epsilon)$ dann folgt die untere Grenze $\Omega(n^2 \log n)$

Theorem 3: Erwartete Laufzeit des (1+1) EA mit HAM(π) und EXC(π)

Beweis:

Zunächst nur eine Fitnessänderung pro Mutationsschritt

Sei c' die Konstante der $\mathcal{O}(\frac{k}{n^2})$ Grenze und $f(\pi) \geq f(\pi_{opt}) - j$

$\Rightarrow$ Erwartete Laufzeit für eine Fitnesserhöhung $\frac{n^2}{(c'j)}$

Sei $k \in \Omega(n^\epsilon)$, dann folgt für die erwartete Laufzeit als untere Grenze

$$\frac{n^2}{c'} \cdot (\frac{1}{c} + \frac{1}{2c} + ... + \frac{1}{\lfloor k/c \rfloor c}) = \Omega(n^2 \log n)$$

Theorem 3: Erwartete Laufzeit des (1+1) EA mit HAM(π) und EXC(π)

Jetzt: (1+1) EA

Exponentiell kleine Wahrscheinlichkeit für mehr als $3t$ Operationen in t Schritten

- $X + 1$ Schritte, wobei X poisson-verteilt ist
- t Schritte als $t \cdot m$ Bernoulliereignisse mit Wahrscheinlichkeit $\frac{1}{m}$
- Mit den Chernoff-Grenzen folgt, dass $Pr(2t)$ exponentiell klein ist für alle $m \in \mathbb{N}$ und damit auch $m \to \infty$

$\Rightarrow$ Mit Vorfaktor $\frac{1}{3}(1 - o(1))$ folgt analog die untere Grenze $\Omega(n^2 \log n)$

Theorem 3: Erwartete Laufzeit des (1+1) EA mit HAM(π) und EXC(π)

Beweis für HAM(π):

- Obere Grenze $\mathcal{O}(n^2 \log n)$ siehe *Theorem 2*
- Erwartete Fitness von HAM bei Initialisierung nahezu 1 (für $n\epsilon$ exponentiell klein)
- Anwendung des Vorgehens für untere Grenze, wobei HAM(π) = k:
  - *Exchange* verbessert Fitness um maximal 2
  - Wahrscheinlichkeit für Fitnessverbesserung $\frac{k}{n} \cdot \frac{2}{(n-1)}$

$\Rightarrow$ Insgesamt somit erwartete Laufzeit in $\Theta(n^2 \log n)$

□

Theorem 3: Erwartete Laufzeit des (1+1) EA mit HAM(π) und EXC(π)

Präposition 1:

Wenn $\text{EXC}(\pi) = k > 0$, *dann ist* $n - 2k \leq \text{HAM}(\pi) \leq n - k - 1$

Beweis:

- Permutation π besteht aus $n - k$ Zyklen
- Maximal $n - k - 1$ Zyklen der Länge 1

$\Rightarrow \text{HAM}(\pi) \leq n - k - 1$

Theorem 3: Erwartete Laufzeit des (1+1) EA mit HAM(π) und EXC(π)

Präposition 1:

Wenn $\text{EXC}(\pi) = k > 0$, *dann ist* $n - 2k \leq \text{HAM}(\pi) \leq n - k - 1$

Beweis:

- Permutation π besteht aus $n - k$ Zyklen
- Maximal $n - k - 1$ Zyklen der Länge 1

$\Rightarrow \text{HAM}(\pi) \leq n - k - 1$

- Wenn $k \geq \frac{n}{2}$ Grenze trivial, daher $n - k > \frac{n}{2}$
- Summe der Längen der $n - k$ Zyklen $= n$
- HAM minimal, wenn alle anderen Zyklen Länge 2 haben

$\Rightarrow \text{HAM}(\pi) \geq n - 2k$

Theorem 4: Erwartete Laufzeit von (1+1) EA nur mit *exchange* und EXC in $\Theta(n^2 \log n)$

Beweis für EXC(π):

- Obere Grenze $\mathcal{O}(n^2 \log n)$ siehe *Theorem 2*
- Da erwartete Fitness von HAM bei Initialisierung für $n\epsilon$ exponentiell klein, folgt mit *Präposition 1*, dass für EXC bei Initialisierung mit großer Wahrscheinlichkeit gilt:

$$\text{EXC}(\pi) \geq (n - n\epsilon)/2 = \Omega(n)$$

- HAM(π) gibt "Fortschritt"des Algorithmus mit EXC(π) an

$\Rightarrow$ Analog zu *Theorem 3* erwartete Laufzeit in $\Theta(n^2 \log n)$

□

Theorem 4:

Erwartete Laufzeit des (1+1) EA mit Fitnessfunktion LAS(π) liegt in $\Theta(n^2 \log n)$

Theorem 4: Erwartete Laufzeit von (1+1) EA mit LAS

Präposition 2:

In jeder längsten absteigenden Subsequenz bleibt die Position eines Elementes i gleich

Theorem 4: Erwartete Laufzeit von (1+1) EA mit LAS

Beweis:

- Obere Grenze $\mathcal{O}(n^2 \log n)$ siehe *Theorem 2*
- Anwendung des Vorgehens für untere Grenze, wobei $k_0 = n^\epsilon$ für $\epsilon \in (0, \frac{1}{3})$
  - *Exchange* oder *jump* verbessern Fitness um maximal 2
  - **ABER:** Fitnessverbesserung i.A. nur in $\mathcal{O}(\frac{k^2}{n^2})$

$\Rightarrow$ Finde in geringer Zeit Permutation, die $\mathcal{O}(\frac{k}{n^2})$ erfüllt!

Theorem 4: Erwartete Laufzeit von (1+1) EA mit LAS

Sei $\text{LAS}(\pi) = n - k$

- $\text{pos}(i, \pi)$ eindeutige Position in LAS gemäß *Präposition 2* (wenn nicht in LAS dann $\text{pos}(i, \pi) = 0$)
- $\text{pos}(i, \pi)$ mit $n - k$ Funktionswerten ungleich 0 und $n - 2k$ nur einmal vergebenen Werten

$\Rightarrow 2k$ öutsiders", die Fitness durch *jump/exchange* erhöhen können

Theorem 4: Erwartete Laufzeit von (1+1) EA mit LAS

Definition

Seien $a_1, \cdots, a_m$ die Elemente, die in jeder LAS vorkommen. Bezeichne die Elemente zwischen a_i und a_{i+1} als spaces

$\Rightarrow$ Ziel: Jeder space enthält $\mathcal{O}(1)$ outsiders

Theorem 4: Erwartete Laufzeit von (1+1) EA mit LAS

Behauptung 1:

Nach $\Theta(n \log^2 n)$ Schritten enthält jeder *space* mit Wahrscheinlichkeit $1 - o(n)$ höchstens einen *outsider* und die Fitness wurde nicht erhöht

Behauptung 2:

Enthält jeder *space* von π höchstens einen *outsider*, so enthält er nach $\Theta(n^2 \log n)$ Schritten mit Wahrscheinlichkeit $1 - o(1)$ maximal 6 *outsiders*

Theorem 4: Erwartete Laufzeit von (1+1) EA mit LAS

Beweis Behauptung 1:

Erinnerung: $n-$ LAS$(\pi) = k \leq k_0 = n^\epsilon \qquad 0 < \epsilon < \frac{1}{3}$

Sammelbildproblem: Mit Wahrscheinlichkeit $1 - o(n)$ für jedes Element Operation ausgeführt

Theorem 4: Erwartete Laufzeit von (1+1) EA mit LAS

Beweis Behauptung 1:

Erinnerung: $n-$ LAS$(\pi) = k \leq k_0 = n^\epsilon \qquad 0 < \epsilon < \frac{1}{3}$

Sammelbildproblem: Mit Wahrscheinlichkeit $1 - o(n)$ für jedes Element Operation ausgeführt

Fitnesserhöhung nur wenn einer der $\mathcal{O}(k)$ *outsider* "richtig" springt in $\mathcal{O}(n^{2\epsilon-2})$

$\Rightarrow$ Wahrscheinlichkeit für alle Schritte in $\mathcal{O}(n^{2\epsilon-1}\log^2 n)$ und damit $o(1)$

□

Theorem 4: Erwartete Laufzeit von (1+1) EA mit LAS

Beweis Behauptung 2:

Betrachte zunächst nur einen (nicht leeren) *space*

- Wahrscheinlichkeit $\Omega(\frac{1}{n})$, dass sich *space* Länge verkleinert
- Wahrscheinlichkeit $\mathcal{O}(n^{\epsilon-2})$, dass sich *space* Länge vergrößert
- Wahrscheinlichkeit für 2 oder mehr Verlängerungen in $\mathcal{O}(n^{2\epsilon-4})$

$\Rightarrow$ Bei $\mathcal{O}(n^2 \log n)$ Schritten Wahrscheinlichkeit in $\mathcal{O}(n^{3\epsilon-2} \log n) = o(1)$

$\Rightarrow$ Wahrscheinlichkeit für mehr als 2 Längenerhöhungen vernachlässigbar

Theorem 4: Erwartete Laufzeit von (1+1) EA mit LAS

Beweis Behauptung 2 (Fortsetzung):

Bedingte Wahrscheinlichkeit, dass Längenänderung für *space* eine Erhöhung ist: $p := \mathcal{O}(n^{\epsilon-1})$

Frage: Mit welcher Wahrscheinlichkeit erreichen wir Länge 6 bevor der *space* leer wird ?

Theorem 4: Erwartete Laufzeit von (1+1) EA mit LAS

Beweis Behauptung 2 (Fortsetzung):

Bedingte Wahrscheinlichkeit, dass Längenänderung für *space* eine Erhöhung ist: $p := \mathcal{O}(n^{\epsilon-1})$

Frage: Mit welcher Wahrscheinlichkeit erreichen wir Länge 6 bevor der *space* leer wird ?

$\Rightarrow$ "Gambler's ruin problem" mit $t = \frac{1-p}{p} = \Omega(n^{1-\epsilon})$ und damit

$$q = \frac{(t-1)}{t^6 - 1} = \mathcal{O}(n^{5\epsilon-5})$$

$\Rightarrow$ Bei $\mathcal{O}(n^2 \log n)$ Schritten Wahrscheinlichkeit nach oben beschränkt durch $\mathcal{O}(n^{6\epsilon-3} \log n) = o(1)$

Analyse: Evolutionäre Algorithmen für Sortierprobleme

Haben bereits für INV, HAM, LAS, EXC gezeigt:

- Untere Schranke $\Omega(n^2)$
- Obere Schranke $\mathcal{O}(n^2 \log n)$
- HAM nur mit *exchange* in $\Theta(n^2 \log n)$
- EXC nur mit *exchange* in $\Theta(n^2 \log n)$
- LAS in $\Theta(n^2 \log n)$

$\Rightarrow$ Für INV keine genaueren Aussagen möglich

Analyse: Evolutionäre Algorithmen für Sortierprobleme

Haben bereits für INV, HAM, LAS, EXC gezeigt:

- Untere Schranke $\Omega(n^2)$
- Obere Schranke $\mathcal{O}(n^2 \log n)$
- HAM nur mit *exchange* in $\Theta(n^2 \log n)$
- EXC nur mit *exchange* in $\Theta(n^2 \log n)$
- LAS in $\Theta(n^2 \log n)$

$\Rightarrow$ Für INV keine genaueren Aussagen möglich

Erwartete Laufzeit von RUN(π) ?

Beispiel:

2 5 6 7 14 15 | 1 3 4 12 | 10 11 13 | 8 9

- keine Längenveränderung
- *Run* Länge steigt/sinkt
- Zwei *Runs* verschmelzen

$\Rightarrow$ Analyse nur für *jump* Operation

Annahme: *Run* einer Größe von mindestens $\frac{3}{8}n$ und beliebiger anderer *Run* verschmelzen nicht

Theorem 5:

Die erwartete Laufzeit des (1+1) EA nur mit *jump* Operationen und Fitnessfunktionen RUN(π) ist exponentiell groß

Theorem 5: Erwartete Laufzeit von RUN(π)

Beweis:

Anfang: Zufällige Permutation

$\Rightarrow$ Wahrscheinlichkeit für *Run* der Länge $n^{\frac{1}{2}}$ exponentiell klein

Betrachte Zeitpunkt π wenn für den längsten *Run* der Länge l_1 gilt:

$$\frac{5}{8}n \leq l_1 \leq \frac{3}{4}n$$

Allgemein: *Runs* der Länge nach geordnet mit $l_1 \geq l_2 \geq ... \geq l_k$, wobei RUN($\pi$) $= k$

Theorem 5: Erwartete Laufzeit von RUN(π)

Zufällige jump Operation:

- l_1 wird erhöht mit Wahrscheinlichkeit:

$$\frac{n-l_1}{n} \cdot \frac{1}{n-1} = \frac{1}{n-1} - \frac{l_1}{n(n-1)} \leq \frac{3}{8} \cdot \frac{1}{n-1}$$

- l_1 wird erniedrigt mit Wahrscheinlichkeit:

$$\frac{l_1}{n} \cdot \frac{k-1}{n-1} \geq \frac{l_1}{n(n-1)} \geq \frac{5}{8} \cdot \frac{1}{n-1}$$

$\Rightarrow$ Nur Schritte mit Änderung von l_1 (Untere Grenze)

Theorem 5: Erwartete Laufzeit von RUN(π)

Annahme: Immer nur eine *jump* Operation mit Startwert $l_1 = \frac{3}{4}n$

Frage: Mit welcher Wahrscheinlichkeit erreichen wir Länge $\frac{7}{8}n$ bevor der *Run* auf Länge $\frac{5}{8}n$ schrumpft?

Theorem 5: Erwartete Laufzeit von RUN(π)

Annahme: Immer nur eine *jump* Operation mit Startwert $l_1 = \frac{3}{4}n$

Frage: Mit welcher Wahrscheinlichkeit erreichen wir Länge $\frac{7}{8}n$ bevor der *Run* auf Länge $\frac{5}{8}n$ schrumpft?

$\Rightarrow$ "Gambler's ruin problem" mit $t = \frac{1-p}{p} = \frac{5}{3}$ und damit

$$q = \frac{(5/3)^{n/8} - 1}{(5/3)^{n/4} - 1} = (5/3)^{-n/8}(1 - o(1))$$

$\Rightarrow$ Finde passende Phasenlänge

Theorem 5: Erwartete Laufzeit von RUN(π)

Jetzt: Mehrere *jump* Operatoren innnerhalb eines Schrittes

Betrachte ab dem Zeitpunkt an dem $n - 2n^{1/2} \leq l_1 \leq n - n^{1/2}$ nur noch Phasen der Länge $n^{3/2}$

- $\Omega(n^{1/2})$ Schritte verkürzen den *Run* (mit großer Wahrscheinlichkeit)
- $\mathcal{O}(n^{\epsilon})$ Schritte verlängern den *Run* (mit großer Wahrscheinlichkeit)

$\Rightarrow$ Wähle $\epsilon < \frac{1}{2}$ und n groß

Theorem 5: Erwartete Laufzeit von RUN(π)

Run- Verkürzung

Wahrscheinlichkeit in $\Omega(\frac{1}{n})$

- $\frac{1}{e}$ nur eine *jump* Operation
- $\Omega(1)$ Element des längsten *Runs*
- $\Omega(\frac{1}{n})$ passendes Ziel

$\Rightarrow$ Mit Chernoff Grenzen: # Verkürzungen in $\Omega(n^{1/2})$ für Phase der Länge $n^{3/2}$ mit großer Wahrscheinlichkeit

$\Rightarrow$ Beweis für RUN-Verlängerung technisch

□

Analyse: Evolutionäre Algorithmen für Sortierprobleme

Haben für INV, HAM, LAS, EXC, RUN gezeigt:

- Untere Schranke $\Omega(n^2)$

Analyse: Evolutionäre Algorithmen für Sortierprobleme

Haben für INV, HAM, LAS, EXC, RUN gezeigt:

- Untere Schranke $\Omega(n^2)$

Haben für INV, HAM, LAS, EXC, gezeigt:

- Obere Schranke $\mathcal{O}(n^2 \log n)$
- HAM nur mit *exchange* in $\Theta(n^2 \log n)$
- EXC nur mit *exchange* in $\Theta(n^2 \log n)$
- LAS in $\Theta(n^2 \log n)$

Analyse: Evolutionäre Algorithmen für Sortierprobleme

Haben für INV, HAM, LAS, EXC, RUN gezeigt:

- Untere Schranke $\Omega(n^2)$

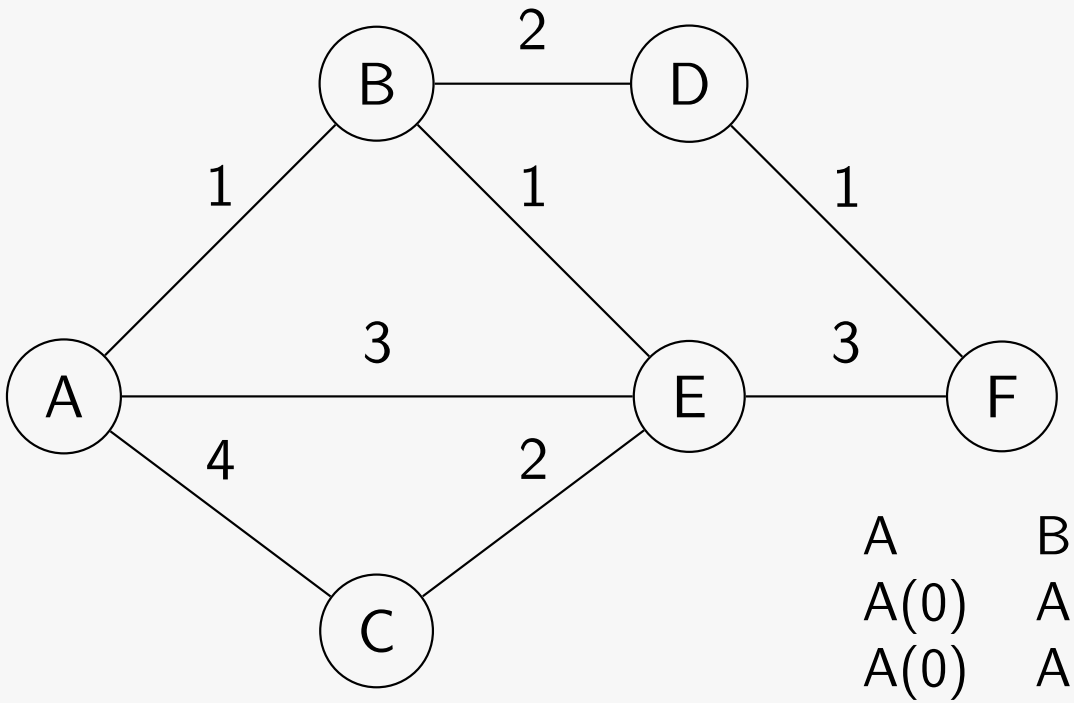
Haben für INV, HAM, LAS, EXC, gezeigt:

- Obere Schranke $\mathcal{O}(n^2 \log n)$
- HAM nur mit *exchange* in $\Theta(n^2 \log n)$
- EXC nur mit *exchange* in $\Theta(n^2 \log n)$
- LAS in $\Theta(n^2 \log n)$

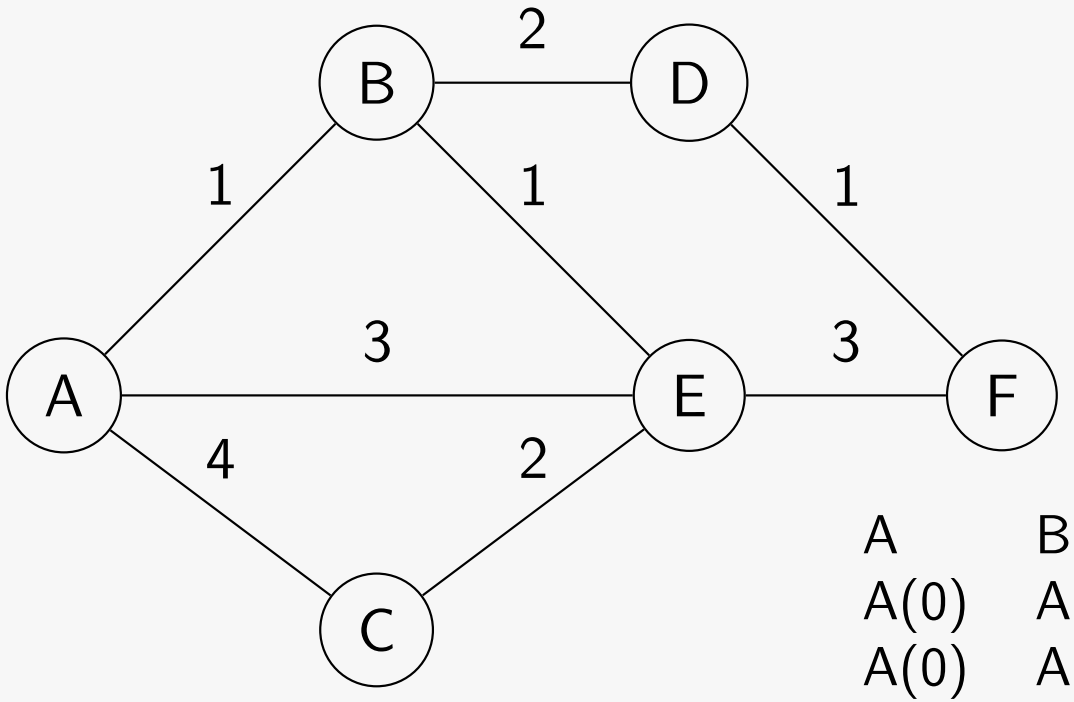
Haben für RUN gezeigt:

- Laufzeit unter bestimmten Voraussetzungen exponentiell

Agenda



A	B	C	D	E	F
A(0)	A(1)	A(4)		A(3)	
A(0)	A(1)	A(4)	B(3)	B(2)	
A(0)	A(1)	A(4)	B(3)	B(2)	E(5)
A(0)	A(1)	A(4)	B(3)	B(2)	D(4)
A(0)	A(1)	A(4)	B(3)	B(2)	D(4)
A(0)	A(1)	A(4)	B(3)	B(2)	D(4)



A	B	C	D	E	F
A(0)	A(1)	A(4)		A(3)	
A(0)	A(1)	A(4)	B(3)	B(2)	
A(0)	A(1)	A(4)	B(3)	B(2)	E(5)
A(0)	A(1)	A(4)	B(3)	B(2)	D(4)
A(0)	A(1)	A(4)	B(3)	B(2)	D(4)
A(0)	A(1)	A(4)	B(3)	B(2)	D(4)

$\rightarrow$ Speicher Dijkstra in $\mathcal{O}(n)$

$\rightarrow$ Laufzeit Dijkstra in $\mathcal{O}(n^2)$ (Fibonacci Heap: $\mathcal{O}(n \log n + m)$)

Evolutionäre Algorithmen für Kürzeste-Wege-Problem:

Ausgangssituation:

Distanzwerte in Distanzmatrix $D = (d_{ij})_{1 \leq i,j \leq n}$ mit $d_{ij} \in \mathbb{N} \cup \{\infty\}$

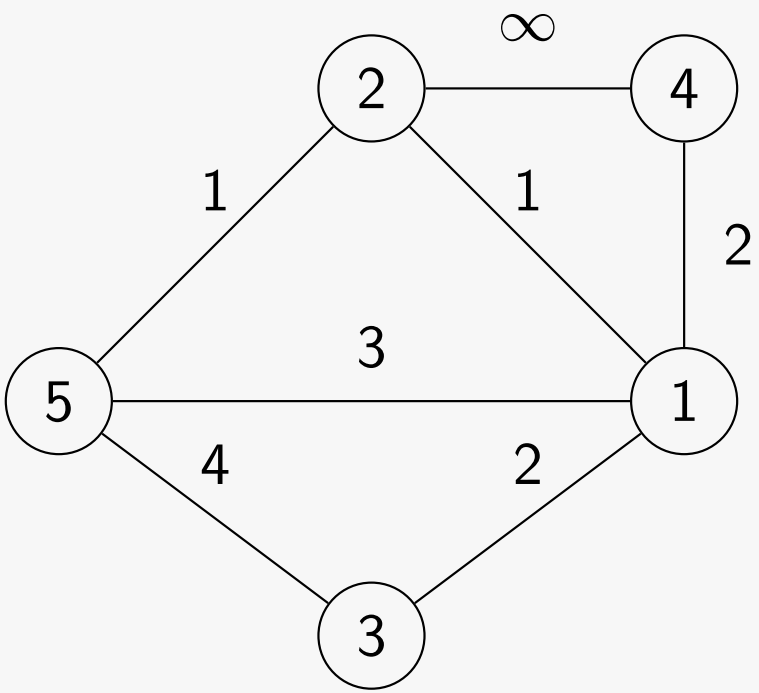
$\Rightarrow$ Speicherbedarf in $\Theta(n^2)$

Evolutionäre Algorithmen für Kürzeste-Wege-Problem:

Ausgangssituation:

Distanzwerte in Distanzmatrix $D = (d_{ij})_{1 \leq i,j \leq n}$ mit $d_{ij} \in \mathbb{N} \cup \{\infty\}$

$\Rightarrow$ Speicherbedarf in $\Theta(n^2)$



$$D = \begin{pmatrix} - & 1 & 2 & 2 & 3 \\ 1 & - & - & \infty & 1 \\ 2 & - & - & - & 4 \\ 2 & \infty & - & - & - \\ 3 & 1 & 4 & - & - \end{pmatrix}$$

Evolutionäre Algorithmen für Kürzeste-Wege-Problem:

Suchraum:

Vektoren $v = (v_1, \cdots, v_{n-1}) \in \{1, \cdots, n\}^{n-1}$ mit $v_i \neq i$ und $s = n$

$\Rightarrow$ Forderung an durch v beschriebenen gerichteten Graph:

$$deg^-(n) = 0 \text{ und } deg^-(x) = 1 \ \forall x \in [n-1]$$

Evolutionäre Algorithmen für Kürzeste-Wege-Problem:

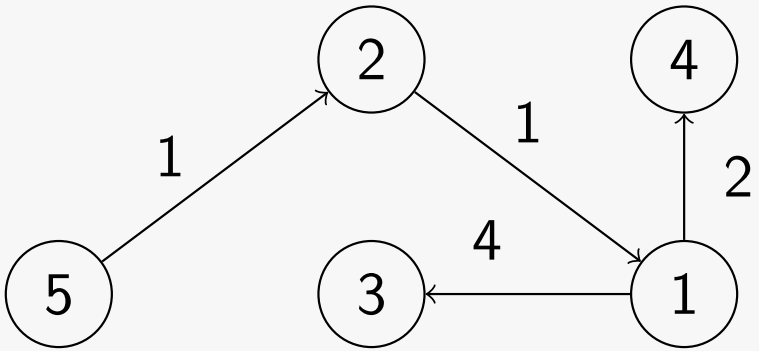
Suchraum:

Vektoren $v = (v_1, \cdots, v_{n-1}) \in \{1, \cdots, n\}^{n-1}$ mit $v_i \neq i$ und $s = n$

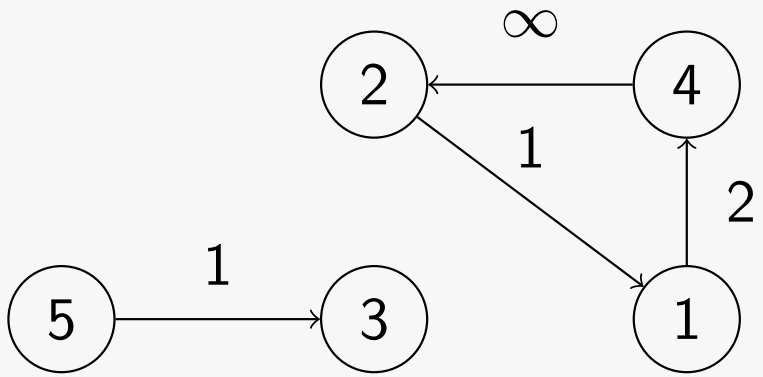
$\Rightarrow$ Forderung an durch v beschriebenen gerichteten Graph:

$$deg^-(n) = 0 \text{ und } deg^-(x) = 1 \ \forall x \in [n-1]$$

$v = (2, 5, 1, 1)$



$v = (2, 4, 5, 1)$



Evolutionäre Algorithmen für Kürzeste-Wege-Problem:

Lokale Operation:

Ändere den Wert einer Komponente v_i des Vektors v in $v_i' \in \{1, \cdots, n\} - \{i, v_i\}$

$\Rightarrow (n-1)(n-2)$ Möglichkeiten

Mutationsoperator:

Wähle k gemäß der Poissonverteilung mit $Poi_1(k) = \frac{1}{k!}e^{-1}$ aus und führe sequentiell $k+1$ zufällig ausgewählte lokale Operationen durch

Evolutionäre Algorithmen für Kürzeste-Wege-Problem:

Algorithmus des (1+1) EA für Kürzeste-Wege-Problem

- Initialisiere zufälligen Vektor $v \in \{1, \cdots, n\}^{n-1}$
- Wiederhole:
  1. Erstelle v' durch Mutation von v
  2. Ersetze v durch v' falls $f(v) \geq f(v')$
  3. Falls v die kürzesten Wege $s - i \quad \forall i \in [n-1]$ beschreibt, dann Schleife abbrechen

Evolutionäre Algorithmen für Kürzeste-Wege-Problem:

Algorithmus des (1+1) EA für Kürzeste-Wege-Problem

- Initialisiere zufälligen Vektor $v \in \{1, \cdots, n\}^{n-1}$
- Wiederhole:
  1. Erstelle v' durch Mutation von v
  2. Ersetze v durch v' falls $f(v) \geq f(v')$
  3. Falls v die kürzesten Wege $s - i \quad \forall i \in [n-1]$ beschreibt, dann Schleife abbrechen

$\Rightarrow$ Wie können wir die Fitness messen?

Fitnessfunktion für (1+1) EA: Versuch 1

Setze $f(v) = \infty$ wenn f ungültig ist und $f = \sum_{i=1}^{n-1} l_i$ sonst, mit l_i als Gewicht des durch v beschriebenen $s - i$ -Weges

Fitnessfunktion für (1+1) EA: Versuch 1

Setze $f(v) = \infty$ wenn f ungültig ist und $f = \sum_{i=1}^{n-1} l_i$ sonst, mit l_i als Gewicht des durch v beschriebenen $s - i$ -Weges

Aber: Problem der Nadel im Heuhaufen

Sei $d_{i,i-1} < \infty$ und $d_{i,j} = \infty$ sonst

$\Rightarrow$ Vektor $v_{opt} = (2, 3, ..., n-1, n)$ als einziger $\neq \infty$

$\Rightarrow$ Komplett zufällige Suche in erwarteter exponentieller Laufzeit

$\Rightarrow$ **Mehr Information in Fitnessfunktion**

Fitnessfunktion für (1+1) EA: Versuch 2

Schränke das Problem auf Distanzmatrizen mit $d_{ij} \in \{1, ..., d^*\} \cup \{\infty\}$ ein

Fitnessfunktion für (1+1) EA: Versuch 2

Schränke das Problem auf Distanzmatrizen mit $d_{ij} \in \{1, ..., d^*\} \cup \{\infty\}$ ein

Definiere $f(v)$ als Summe von:

- Der Summe der Gewichte aller j endlichen $s - i$ Wege
- $c \cdot nd^*$, wobei c die Anzahl der $s - i$ Wege mit Gewicht ∞
- $((n-1) - j - c) \cdot n^2 d^*$ für nicht vorhandene $s - i$ Wege

$\Rightarrow$ **SZu komplex"für präzise Laufzeitanalyse**

Fitnessfunktion für (1+1) EA: Versuch 3

Single-objective $\rightarrow$ *Multi-objective*:

Sei $f(v) = (f_1(v), ..., f_n(v))$ mit f_i als Gewicht des $s - i$-Weges (∞ sonst)

Definiere partielle Ordnung auf $\mathbb{R}^{n-1}$:

$$f(v) \preccurlyeq f(v') \Leftrightarrow f_i(v) \leq f_i(v') \quad \forall i \in [n-1]$$

Fitnessfunktion für (1+1) EA: Versuch 3

Single-objective $\rightarrow$ *Multi-objective*:

Sei $f(v) = (f_1(v), ..., f_n(v))$ mit f_i als Gewicht des $s - i$-Weges (∞ sonst)

Definiere partielle Ordnung auf $\mathbb{R}^{n-1}$:

$$f(v) \preccurlyeq f(v') \Leftrightarrow f_i(v) \leq f_i(v') \quad \forall i \in [n-1]$$

$\Rightarrow$ Es existiert genau ein *Pareto* optimaler Vektor $l^* = (l_1^*, ..., l_{n-1}^*)$

$\Rightarrow$ Finde Suchpunkt, sodass der dazugehörige Vektor l^* *Pareto* optimal ist

Evolutionäre Algorithmen für Kürzeste-Wege-Problem:

Algorithmus des *multi-objective* (1+1) EA für Kürzeste-Wege-Problem

- Initialisiere zufälligen Vektor $v \in \{1, \cdots, n\}^{n-1}$
- Wiederhole:
  1. Erstelle v' durch Mutation von v
  2. Ersetze v durch v' falls $f(v') \preccurlyeq f(v)$
  3. Falls v einen *Pareto* optimalen Gewichtsvektor beschreibt, dann Schleife abbrechen

Theorem 6:

Die erwartete Laufzeit des *multi-objective* (1+1) EA für SSSP liegt in $\mathcal{O}(n^3)$

Theorem 6: Erwartete Laufzeit des multi-objective (1+1) EA

Präposition 3:

Erwartete Laufzeit des multi-objective (1+1) EA *für ein globales Maximum in* $\Omega(n^2)$

Theorem 6: Erwartete Laufzeit des multi-objective (1+1) EA

Präposition 3:

Erwartete Laufzeit des multi-objective (1+1) EA *für ein globales Maximum in* $\Omega(n^2)$

Beweis:

- $P_{v_{opt}} = \frac{1}{n!}$
- Betrachte nur letzten Mutationsschritt: Maximal ein „passender Vorgänger “

$\rightarrow P_{Last} = \frac{1}{(n-1)(n-2)}$ und damit die *Waiting time* in $\Omega(n^2)$

□

Theorem 6: Erwartete Laufzeit des multi-objective (1+1) EA

Beweis:

Seien

- t_i Minimum Kanten in kürzestem $s - i$-Weg
- $m_j := \#\{\ i \mid t_i = j\}$
- $T = \max(\ j \mid m_j > 0)$

Wir zeigen als obere Grenze

$$en^2 \sum_{1 \leq j \leq T} (\ln m_j + 1)$$

$\Rightarrow$ Maximal für $m_1 = \ldots = m_{n-1} = 1$

Theorem 6: Erwartete Laufzeit des multi-objective (1+1) EA

Annahme: $f_i(v) = l_i^* \ \forall i$ mit $t_i < t$

$\rightarrow$ Warte bis $f_i(v) = l_i^* \ \forall i$ mit $t_i \leq t$

Für r Positionen i mit $t_i = t$ und $f_i(v) > l_i^*$ existiert j mit $t_j = t - 1$

$\rightarrow$ Wahrscheinlichkeit $\frac{r}{e(n-1)(n-2)}$ für „gute“ Mutationen

Theorem 6: Erwartete Laufzeit des multi-objective (1+1) EA

Annahme: $f_i(v) = l_i^* \ \forall i$ mit $t_i < t$

$\rightarrow$ Warte bis $f_i(v) = l_i^* \ \forall i$ mit $t_i \leq t$

Für r Positionen i mit $t_i = t$ und $f_i(v) > l_i^*$ existiert j mit $t_j = t - 1$

$\rightarrow$ Wahrscheinlichkeit $\frac{r}{e(n-1)(n-2)}$ für „gute“ Mutationen

$\Rightarrow$ Insgesamt erwartete Laufzeit für Phase durch $en^2(\ln m_t + 1)$ beschränkt

$\Rightarrow$ Da $1 \leq t \leq T$ folgt die obere Grenze:

$$en^2 \sum_{1 \leq t \leq T} (\ln m_t + 1)$$

□

Theorem 6: Erwartete Laufzeit des multi-objective (1+1) EA

Theorem 6 auch für *Problem der Nadel im Heuhaufen*?

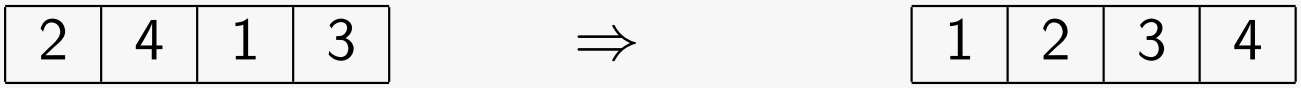
Sei $d_{i,i-1} = 1$ und $d_{i,j} = \infty$ sonst

- Solange $v_{n-1} = n$ gilt $f(v) = (\infty, ..., \infty)$
- Wahrscheinlichkeit $\frac{1}{n-1}$ für $v_{n-1} = n$ von Anfang an
- Erwartete Laufzeit bis v_{n-1} in $\Theta(n^2)$
- Wiederhole Vorgehen für $v_{n-2}, ..., v_1$

$\Rightarrow$ Insgesamt erwartete Laufzeit in $\Theta(n^3)$

Zusammenfassung

Analyse von EA für Sortierprobleme:



(1+1) EA für Fitnessfunktionen INV, HAM, EXC, LAS, RUN

Zusammenfassung

Analyse von EA für Sortierprobleme:

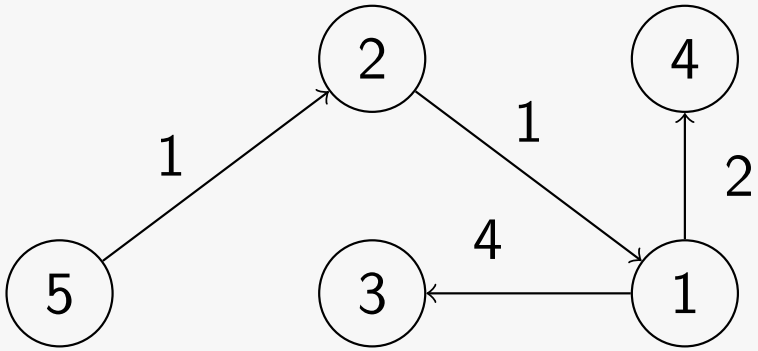
2	4	1	3

$\Rightarrow$

1	2	3	4

(1+1) EA für Fitnessfunktionen INV, HAM, EXC, LAS, RUN

Analyse von EA für Kürzeste-Wege-Problem:



(1+1) multi-objective EA

Agenda

1. Wrap Up GAD: Sortieren und SSSP
   - Traditionelles Sortieren
   - Algorithmus von Dijkstra
2. Evolutionäre Algorithmen
   - Konzept eines EA
   - EA als Sortierverfahren
3. Analyse: Evolutionäre Algorithmen für Sortierprobleme
4. Analyse: Evolutionäre Algorithmen für Kürzeste-Wege-Problem
5. Ausblick

Und wie geht's weiter ..?

Präzisere Analyse für Fitnessfunktionen INV, HAM und RUN

Und wie geht's weiter ..?

Präzisere Analyse für Fitnessfunktionen INV, HAM und RUN

Weitere Fitnessfunktionen wie zum Beispiel

- $\mathrm{MAX}(X) = \max_{1 \leq i \leq n} \mid i - \pi(i) \mid$
- $\mathrm{Block}(X) = \#\{i \mid 1 \leq i < n \ \wedge \ \pi(i) + 1 \neq \pi(i+1)\} + 1$

Und wie geht's weiter ..?

Präzisere Analyse für Fitnessfunktionen INV, HAM und RUN

Weitere Fitnessfunktionen wie zum Beispiel

- $\mathrm{MAX}(X) = \max_{1 \leq i \leq n} \mid i - \pi(i) \mid$
- $\mathrm{Block}(X) = \#\{i \mid 1 \leq i < n \ \wedge \ \pi(i) + 1 \neq \pi(i+1)\} + 1$

Analyse des (1+1) EA für weitere kombinatorische Probleme (MST)

$\rightarrow$ **Martin Bullinger**

Danke!