

Vorlesung

Effiziente Algorithmen und Komplexitätstheorie

Sommersemester 2008

Ingo Wegener

Algorithmenklassen

- effizient, opt. Lösung
z. B. Goldberg/Tarjan für das Flussproblem
 - effiziente Approximation garantierter Mindestgüte
z. B. FPTAS für KP
 - optimale Lösung
z. B. 3-SAT in exp. Zeit
 - garantierte Güte
z. B. Branch&Bound
 - keinerlei Garantien
e. g. Lin/Kernighan-Heuristik für das TSP
- keine Laufzeitschranke
 - keine Gütegarantie
 - kein Erkennen einer Lösung
 - leicht& schnell implementierbar
 - häufig gute Resultate
 - rand. lokale Suche
 - Metropolisalgorithmus
 - Simulated Annealing
 - Tabu-Search
 - Ant Colony Optimization
 - Particle Swarm Optimization
 - evolutionäre Algorithmen

problemspezifische
Algorithmen

allgemeine
Suchheuristiken

Analyse allgemeiner randomisierter Suchheuristiken

„Benutze einen problemspezifischen Algo.
wenn irgend möglich.“

Und wann benutzt man allgemeine Suchheuristiken?

Falls

- kein problemspezifischer Algorithmus bekannt ist,
- keine Zeit, Expertise, Geld, ... zur Entwicklung vorhanden ist,
- Rechenkraft zur Verfügung steht,
- man dringend **irgendwelche** Resultate braucht.

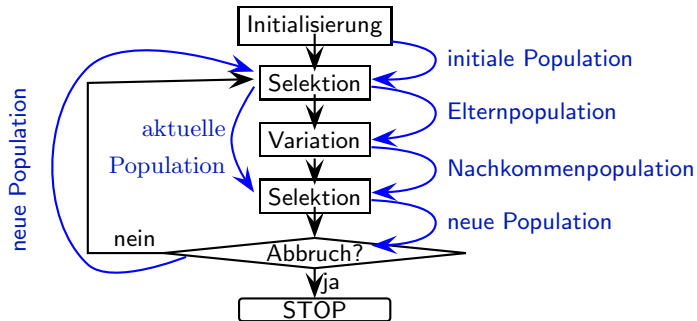
Und was analysieren wir da?

- allgemeine Positivresultate („globale Konvergenz“)
- allgemeine Negativresultate (Black-Box-Komplexität, NFL)
- paradigmatische Beispielfunktionen
- (interessante) Problemklassen
- kombinatorische Optimierungsprobleme

Analyseziel besseres Verständnis von Grenzen/Möglichkeiten

Evolutionäre Algorithmen

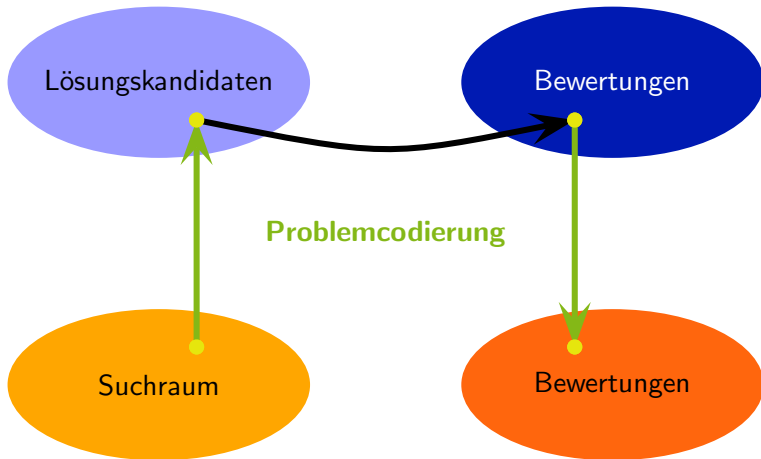
Ziel Finde ein $s \in S$, dass die **Fitness** $f: S \rightarrow R$ maximiert.



- **Initialisierung** häufig uniform zufällig
- **Selektion** fitness-basiert
- **Variation**
 - **Mutation** basierend auf einem Elter
 - **Crossover** basierend auf zwei Eltern
- **Abbruchkriterium** meist **ausgelassen** für Analyse

Optimierungsprobleme und allgemeine Suchheuristiken

Optimierungsproblem



allgemeine Suchheuristik

Allgemeine randomisierte Suchheuristiken

in der Vorlesung

- Einschränkung auf sehr einfache randomisierte Suchheuristiken
- immer Suchraum $\{0, 1\}^n$
- strukturell sehr ähnliche Suchheuristiken
- immer ohne echtes Stoppkriterium
- immer Analyse der (erwarteten) **Optimierzeit**
= #f-Aufrufe, bis zum Finden eines Maximums

Wir werden uns zunächst ansehen

- randomisierte lokale Suche
- Metropolis-Algorithmus
- Simulated Annealing
- (1+1)-EA

Simulated Annealing

Algorithmus 13.1

Sei $T: \mathbb{N} \rightarrow \mathbb{R}_0^+ \cup \{\infty\}$ eine Funktion, T heißt **Annealingschedule**.
Sei $N: \{0, 1\}^n \rightarrow \mathcal{P}(\{0, 1\}^n)$ eine **Nachbarschaft**. Die folgende Suchheuristik heißt **Simulated Annealing**.

1. $t := 1$
2. Wähle $x_t \in \{0, 1\}^n$ uniform zufällig.
3. Wähle $y \in N(x_t)$ gemäß einer festen Verteilung.
4. Mit Wahrscheinlichkeit $\min\{1, e^{-(f(y)-f(x_t))/T(t)}\}$,
setze $x_{t+1} := y$, sonst $x_{t+1} := x_t$.
5. $t := t + 1$
6. Weiter bei 3.

typische Nachbarschaft $N(x) = \{y \in \{0, 1\}^n \mid 0 < d(x, y) \leq d^*\}$
mit $d^* = 1$ oder $d^* = 2$

typische Verteilung Gleichverteilung

Metropolis-Algorithmus und RLS

Algorithmus 13.2 (Metropolis-Algorithmus)

Der **Metropolis-Algorithmus** ist Simulated Annealing mit fester **Temperatur** $T \in \mathbb{R}_0^+ \cup \{\infty\}$.

Algorithmus 13.3 (Randomisierte lokale Suche)

RLS ist der Metropolis-Algorithmus mit Temperatur $T = 0$, dabei sei $\min \{1, e^{-0/0}\} = 1$.

Beobachtungen

- RLS akzeptiert keine Verschlechterung im f -Wert.
- Metropolis-Algo. akzeptiert Verschlechterung im f -Wert mit W 'keit, die von T und f -Wert-Differenz abhängt.
- SA akzeptiert Verschlechterung im f -Wert mit W 'keit, die von T , f -Wert-Differenz und Zeitpunkt t abhängt.

Ein sehr einfacher evolutionärer Algorithmus

Definition 13.4 ((1+1)-EA)

Sei $p \in [0, 1]$, p heißt **Mutationswahrscheinlichkeit**. Die folgende Suchheuristik heißt **(1+1)-EA**.

1. $t := 1$
2. Wähle $x_t \in \{0, 1\}^n$ uniform zufällig.
3. Für $i \in \{1, 2, \dots, n\}$
 Mit W'keit p , setze $y[i] := 1 - x_t[i]$, sonst $y[i] := x_t[i]$.
4. If $f(y) \leq f(x_t)$,
 Then $x_{t+1} := y$ Else $x_{t+1} := x_t$.
5. $t := t + 1$
6. Weiter bei 3.

typische Mutationsw'keit $p = 1/n$

Minimale Spannbäume

Eingabe ungerichteter, gewichteter, zusammenhängender Graph $(G = (V, E), w)$ mit $w: E \rightarrow \mathbb{N}$

Ausgabe ungerichteter, gewichteter, zusammenhängender Graph $(G = (V, E'), w)$ mit $E' \subseteq E$ und $w(E') = \sum_{e \in E'} w(e)$

minimal unter allen solchen Teilgraphen

klar

- Problem bekannt und gut verstanden
- natürlich in P
- effiziente Algorithmen bekannt (**Kruskal**)
- in der Praxis **keine randomisierten Suchheuristiken** benutzen
- **interessantes** Beispielproblem **Testfall** für Effizienz

Problemcodierung

Notation $m = |E|$

Entscheidung Modellierung als Kantenauswahlproblem

Suchraum $\{0, 1\}^m$ mit $x[i] = 1 \Leftrightarrow e_i$ gewählt

Notation zu $x \in \{0, 1\}^m$ ist $E_x = \bigcup_{x[i]=1} e_i$

Definition einer passenden Zielfunktion $f: \{0, 1\}^m \rightarrow \mathbb{R}$

1. **Versuch** $f_1(x) := w(E_x)$

Kritik \emptyset ist optimal

2. **Versuch** $f_2(x) := \begin{cases} w(E_x) & \text{falls } (V, E') \text{ zusammenhängend} \\ \infty & \text{sonst} \end{cases}$

Kritik vielleicht **zu schwer**, überhaupt Spannbaum zu finden

Beobachtung Wahl einer **guten** Zielfunktion **nicht trivial**

Noch mehr Zielfunktionen

3. Versuch $f_3(x) := (c(x) - 1) \cdot w_b^2 + (|x| - (n - 1)) \cdot w_b + w(E_x)$
 mit $c(x) := \#$ Zusammenhangskomponenten in (V, E_x) ,
 $w_b := m \cdot \max\{w(e_1), \dots, w(e_m)\}$
 und $|x| = \sum_{i=1}^n x[i]$

Kritik vielleicht zu sehr in Richtung Lösung formuliert

4. Versuch $f_4(x) := (c(x) - 1) \cdot w_b + w(E_x)$

Kritik etwas künstliche Verschmelzung zweier Kriterien

5. Versuch $f_5(x) := (c(x), w(E_x))$
 zu minimieren in beiden Komponenten

Kritik nicht mehr $f_5: \{0, 1\}^m \rightarrow \mathbb{R}$
 \rightsquigarrow Algorithmenanpassung nötig

Randomisierte Suchheuristiken auf MST

Codierung und Fitnessfunktion

Weg frei

für Anwendung und Analyse

zu den Algorithmen

- **Nachbarschaft** $N(x) := \{y \in \{0, 1\}^m \mid 0 < d(x, y) \leq 2\}$
- **Verteilung** Mit W'keit $1/2$ wähle aus
 $\{y \in \{0, 1\}^m \mid d(x, y) = 1\}$ uniform zufällig, sonst wähle aus
 $\{y \in \{0, 1\}^m \mid d(x, y) = 2\}$ uniform zufällig

Beobachtung

- für jedes y mit $d(x, y) = 1$ ist W'keit, gewählt zu werden
 $1/(2n) = \Theta(1/n)$
- für jedes y mit $d(x, y) = 2$ ist W'keit, gewählt zu werden
 $1 / (2 \binom{n}{2}) = \Theta(1/n^2)$

jetzt Analyse für f_3, f_4, f_5 und verschiedene Suchheuristiken

MST modelliert mit f_3

$$f_3(x) = (c(x) - 1) \cdot w_b^2 + (|x| - (n - 1)) \cdot w_b + w(E_x) \rightarrow \min$$

Theorem

Die erwartete Zeit, bis RLS auf f_3 einen Spannbaum findet, ist $O(m \log m)$.

Beweis.

klar $f_3(x_t)$ fällt monoton in t

Idee Partitioniere $\{0, 1\}^m$ so in S_1, \dots, S_l , dass

- ① $\forall 1 \leq i < j \leq l: \forall x \in S_i, x' \in S_j: f_3(x) > f_3(x')$,
- ② $\forall x \in S_l: x$ erfüllt Zielforderung und
- ③ $\forall 1 \leq i < l: \forall x \in S_i: \exists j > i, y \in S_j: d(x, y) \leq 2$

Notation heißt f_3 -basierte Partition

f_3 -basierte Partition

$$S_i := \begin{cases} \{x \mid c(x) = n + 1 - i\} & \text{für } 1 \leq i < n \\ \{x \mid (c(x) = 1) \wedge |x| = m + n - i\} & \text{für } n \leq i \leq m + 1 \end{cases}$$

Ist $S_1, \dots, S_{m-(n-1)}$ wirklich f_3 -basierte Partition?

- ① f_3 -Werte fallen mit i : für $i < n$ klar, weil jede Zshg-Komponente weniger f_3 -Zuwachs um w_b^2 bedeutet und w_b^2 so groß ist, dass es andere Einflüsse überwiegt; für $i \geq n$ auch klar, weil dann immer $c(x) = 1$ und jede Kante weniger f_3 -Zuwachs um w_b bedeutet, was andere Einflüsse überwiegt
- ② $\forall x \in S_{m+1}: E_x$ ist Spannbaum: klar, weil $c(x) = 1$ und $|E_x| = n - 1$, also zusammenhängend, n Knoten und $n - 1$ Kanten \rightsquigarrow Spannbaum
- ③ von S_i aus ein S_j mit $j > i$ erreichbar: für $i < n$ klar, weil bei $c(x) = k$ immer $\geq k - 1$ Kanten existieren, deren Hinzunahme $c(x)$ verkleinert; für $i > n$ klar, weil bei $\geq n$ Kanten immer eine entfernbare ist, ohne Zusammenhang zu verlieren

Beweis obere Schranke „Spannbaum finden“

Sei $x \in S_i$.

1. Fall: $i < n$ Es sind $n - i$ Kanten wählbar, die zu S_{i+1} führen.

also in jedem Schritt: $\text{Prob}(\text{verlasse } S_i) \geq \frac{1}{2} \cdot \frac{n-i}{m}$

also $E(\text{Zeit in } S_i) \leq \frac{2m}{n-i}$

zusammen $E(\text{Zeit in } S_1, S_2, \dots, S_{n-1}) \leq \sum_{i=1}^{n-1} \frac{2m}{n-i} = 2m \sum_{i=1}^{n-1} \frac{1}{i}$

$< 2m(\ln(n-1) + 1) = O(m \log n)$

2. Fall: $i \geq n$ Es sind $m - i - 1$ Kanten zum Entfernen wählbar.

also in jedem Schritt: $\text{Prob}(\text{verlasse } S_i) \geq \frac{1}{2} \cdot \frac{m-i-1}{m}$

also $E(\text{Zeit in } S_i) \leq \frac{2m}{m-i-1}$

zusammen $E(\text{Zeit in } S_n, S_{n+1}, \dots, S_{m-(n-1)}) \leq \sum_{i=n}^{m-(n-1)} \frac{2m}{m-i-1} =$

$2m \sum_{i=1}^{m-(n-1)+1} \frac{1}{i} < 2m(\ln(m-n+2) + 1) = O(m \log m)$

also Gesamtzeit $O(m \log m)$



Spannbäume mit dem (1+1)-EA finden

Theorem

Die erwartete Zeit, bis der (1+1)-EA auf f_3 einen Spannbaum findet, ist $O(m \log m)$.

Beweis.

Betrachte gleiche f_3 -basierte Partition.

Beobachte:

- $f_3(x_t)$ monoton in t auch für (1+1)-EA
- S_i, \dots, S_l kann so durchlaufen werden, **Überspringen möglich**
- Wechsel zu Hamming-Nachbarn mit 1-Bit-Mutationen möglich
- Prob (erreiche in 1 Schritt einen von i Hamming-Nachbarn) = $\binom{i}{1} \left(\frac{1}{m}\right) \left(1 - \frac{1}{m}\right)^{m-1} \geq \frac{i}{em}$
- Wechsel von $1/2$ auf $1/e$ ändert asymptotisch nichts
- asymptotische obere Schranke bleibt unverändert



Formalismus und Idee

Einen Schritt zurücktreten ...

Mit f -basierten Partitionen finden wir obere Schranken, indem wir

- eine Folge der Länge $\leq l$ von f -verkleinernden Schritten beschreiben,
- für den i -ten Schritt eine untere Schranke p_i für seine Wahrscheinlichkeit finden,
- nachweisen, dass nach Durchlaufen dieser Folge das Ziel erreicht ist,
- die erwartete Zeit durch $\sum_{i=1}^l 1/p_i$ abschätzen.

Struktureinsichten zum MST

Sei $T = (V, E_T)$ ST, $T^* = (V, E_{T^*})$ MST, $k := |E_T \setminus E_{T^*}| \geq 1$.

\exists Bijektion $\alpha: E_{T^*} \setminus E_T \rightarrow E_T \setminus E_{T^*}$ mit

- $\alpha(e)$ liegt auf Kreis in $T \dot{\cup} e$ und
- $w(\alpha(e)) \geq w(e)$.

Erinnerung Korrektheitsbeweis von Kruskal:

$\alpha(e)$ ist die Kante, die gegen e getauscht werden kann, sodass das Gewicht von T nicht wächst.

Beobachtung T und $(T \dot{\cup} e) \setminus \alpha(e)$ haben Hammingabstand 2

Lemma 13.7

Sei T^* ein MST, T ein ST mit $w(T) > w(T^*)$.

$\exists k \in \{1, \dots, n-1\}$: $\exists k$ verschiedene T' mit $d(T', T) = 2$ und durchschnittlich $w(T) - w(T') \geq (w(T) - w(T^*)) / k$.

Beweis des Lemmas

Seien T und T^* definiert wie oben.

Betrachte α definiert wie oben.

Anwendung aller k disjunkten Austausche senkt Gewicht um $w(T) - w(T^*)$.

also je Austausch durchschnittlich $(w(T) - w(T^*))/k$ Abnahme.



Lemma 13.8

Sei T^* ein MST, T ein ST mit $w(T) > w(T^*)$.

$\exists n$ verschiedene T' mit $d(T', T) = 2$ und **durchschnittlich**
 $w(T) - w(T') \geq (w(T) - w(T^*))/n$.

Beweis.

Ergänze um $n - k$ nicht akzeptierte Kantenaustausche.



RLS für MST mit f_3

Theorem 13.9

Die erwartete Optimierzeit von RLS auf MST mit f_3 ist $O(m^2(\log m + \log w_{\max}))$ mit $w_{\max} = \max\{w(e)\}$.

Beweis.

erwartete Zeit bis zu einem Spannbaum $O(m \log m)$

Sei $w_{\text{opt}} = w(T^*)$ für MST T^* .

So lange x kein MST, gibt es $x'_1, \dots, x'_n \in N(x)$ mit durchschnittlichem Gewichtsverlust $\geq \frac{w(x) - w_{\text{opt}}}{n}$.

Prob (wähle eines dieser x'_i) = $\Theta(n/m^2)$

anfangs $D := w(T) - w_{\text{opt}} \leq n \cdot w_{\max}$

nach N solchen Schritten $E(w(T) - w_{\text{opt}}) \leq \left(1 - \frac{1}{n}\right)^N \cdot D$

nach $N := \lceil n \cdot \ln(2) \cdot (\log(D) + 1) \rceil$ ist $E(w(T) - w_{\text{opt}}) \leq \frac{1}{2}$ mit

$\forall e: w(e) \in \mathbb{N}$

Beweis von Theorem 13.9

nach $N := \lceil n \cdot \ln(2) \cdot (\log(D) + 1) \rceil$ ist $E(w(T) - w_{\text{opt}}) \leq \frac{1}{2}$ mit
 $\forall e: w(e) \in \mathbb{N}$

Beobachtung $w(T) - w_{\text{opt}} \geq 0$

Markov-Ungleichung

$$\begin{aligned} \text{Prob}(\text{nach } N := \lceil n \cdot \ln(2) \cdot (\log(D) + 1) \rceil \text{ ist } E(w(T) - w_{\text{opt}}) \geq 1) \\ \leq \frac{1}{2} \end{aligned}$$

also mit W'keit $\geq \frac{1}{2}$ fertig

wenn nicht fertig, nicht schlimmer als neu gestartet

$E(\text{Anzahl solcher Phasen}) \leq 2$

$$\begin{aligned} \text{also Gesamtzeit } O\left(N \frac{m^2}{n}\right) &= O(m^2 \log(D)) \\ &= O(m^2(\log(n) + \log(w_{\text{max}}))) \end{aligned}$$



(1+1)-EA für MST mit f_3

Theorem 13.10

Die erwartete Optimierzeit des (1+1)-EA auf MST mit f_3 ist $O(m^2(\log m + \log w_{\max}))$ mit $w_{\max} = \max\{w(e)\}$.

Beweis.

Alle Überlegungen bleiben gültig.

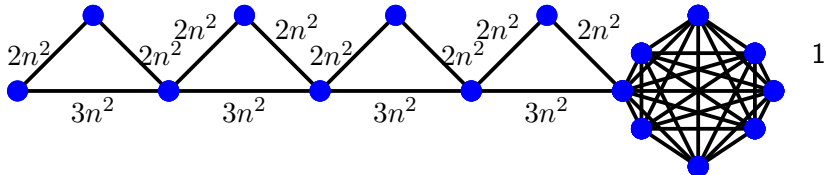
$$\text{Prob}(\text{mache einen von } k \text{ 2-Bit-Flips}) = \binom{k}{1} \left(\frac{1}{m}\right)^2 \left(1 - \frac{1}{m}\right)^{m-2} \geq \frac{k}{em^2} = \Theta\left(\frac{k}{m^2}\right)$$

also gelten asymptotisch die gleichen Ergebnisse □

$O(m^2(\log m + \log w_{\max}))$ ist **langsam**, aber **nicht katastrophal**

Sind RLS und (1+1)-EA wirklich so langsam?

Ein Beispielgraph G_n



allgemein

- $(n/2)$ -Clique mit Kantengewichten 1
- $n/4$ Dreiecke mit Kantengewichten $2 \times 2n^2$ und $1 \times 3n^2$
- n Knoten
- $\Theta(n^2)$ Kanten

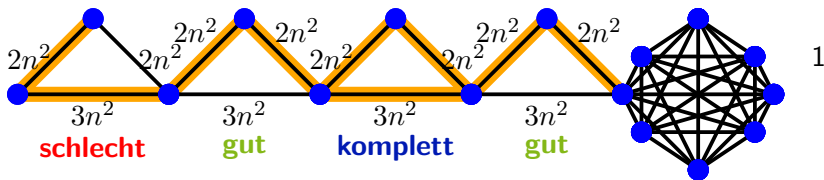
RLS auf G_n

Theorem 13.12

RLS hat auf G_n eine erwartete Optimierzeit von $\Theta(n^4 \log n)$.

Beweis. obere Schranke: wie gesehen ✓

für untere Schranke:



- $b(x) := \#\text{schlechter Dreiecke (bad)}$
- $g(x) := \#\text{guter Dreiecke (good)}$
- $ct(x) := \#\text{kompletter Dreiecke (complete triangles)}$
- $c_G(x) := \#\text{ZHK in } G_n$
- $c_D(x) := \#\text{ZHK in Dreiecken}$
- $c_C(x) := \#\text{ZHK in Clique}$

Beweisidee für untere Schranke

Satz von der totalen Wahrscheinlichkeit

$$E(A) = \sum_B \text{Prob}(B) \cdot E(A \mid B)$$

Anwendung für Optimierzeit T $E(T) \geq \text{Prob}(B) \cdot E(T \mid B)$

Wir zeigen

- für einige Ereignisse B_i jeweils $\text{Prob}(B_i) = 1 - o(1)$
- $\text{Prob}(\bigcap B_i) = 1 - o(1)$
- $E(T \mid \bigcap B_i) = \Omega(n^4 \log n)$

Ein Tool für „Tail Estimations“

Fakt 13.13 (Chernoff-Schranken)

Seien $X_1, \dots, X_n \in \{0, 1\}$ unabhängig ZV mit

$0 < \text{Prob}(X_i = 1) < 1$ für alle X_i , $X := \sum_{i=1}^n X_i$.

- $\forall \delta > 0: \text{Prob}(X > (1 + \delta)\mathbb{E}(X)) \leq \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^{\mathbb{E}(X)}$
- $\forall \delta \in]0; 1]: \text{Prob}(X > (1 + \delta)\mathbb{E}(X)) \leq e^{-\delta^2 \mathbb{E}(X)/3}$
- $\forall \delta \in]0; 1[: \text{Prob}(X < (1 - \delta)\mathbb{E}(X)) \leq e^{-\delta^2 \mathbb{E}(X)/2}$

Beweisidee:

$$\forall t \in \mathbb{R}^+: \text{Prob}(X > (1 + \delta)\mathbb{E}(X)) = \text{Prob}(e^{tX} > e^{t(1+\delta)\mathbb{E}(X)})$$

Markov-Ungleichung: $< \frac{\mathbb{E}(e^{tX})}{e^{t(1+\delta)\mathbb{E}(X)}}$

X_i unabhängig: $\mathbb{E}(e^{tX}) = \mathbb{E}\left(e^{t \sum_{i=1}^n X_i}\right) = \mathbb{E}\left(\prod_{i=1}^n e^{tX_i}\right) = \prod_{i=1}^n \mathbb{E}(e^{tX_i})$

Beweis der unteren Schranke

B_1 : nach Initialisierung $b(x) = \Theta(n)$ und $c_C(x) = 1$

Beobachtung $b(x)$ und $c_C(x)$ unabhängig

zu $b(x)$ $\text{Prob}(i\text{-tes Dreieck schlecht}) = \frac{2}{8} = \frac{1}{4}$
 $\text{Prob}(b(x) \in [(n/4)/8; (n/4)/2]) = 1 - 2 \cdot e^{-\Omega(n)}$ (Chernoff)

zu $c_C(x)$ Betrachte festen Knoten v in Clique.
 $\text{Prob}(\text{Grad von } v \text{ in } x \geq n/8) = 1 - e^{-\Omega(n)}$ (Chernoff)
 für andere Knoten v' :

$$E(\#\text{Nachbarn in Nachbarn von } v) = \frac{\#\text{Nachbarn von } v}{2}$$

$$\text{Prob}(v' \text{ ohne Nachbarn in Nachbarn von } v) = \left(\frac{1}{2}\right)^{\#\text{N. von } v}$$

$$\text{Prob}(\exists v' \text{ ohne Nachbarn in Nachbarn von } v)$$

$$\leq \frac{n}{2} \cdot \left(\frac{1}{2}\right)^{\#\text{N. von } v} = e^{-\Omega(n)}$$

also zusammen $\text{Prob}(B_1) = 1 - e^{-\Omega(n)}$

RLS auf G_n nach der Initialisierung

Begriff k -Schritt: k Dreieckskanten flippen

$k \in \{1, 2\}$:

$$\text{Prob}(k\text{-Schritt}) = \frac{1}{2} \cdot \binom{3(n/4)}{k} \cdot \binom{m}{k}^{-1} = \Theta\left(\left(\frac{n}{m}\right)^k\right) = \Theta\left(\frac{1}{n^k}\right)$$

$k \geq 3$:

$$\text{Prob}(k\text{-Schritt}) = 0$$

Betrachte Phase der Länge $n^{5/2}$.

$$\# \text{ 1-Schritte} = \Theta(n^{3/2}), \# \text{ 2-Schritte} = \Theta(n^{1/2}) \quad (\text{Chernoff})$$

Vorsicht in k -Schritt können in Clique weitere Kanten flippen

Die erste Phase

$\Theta(n^{1/2})$ 2-Schritte können

- $b(x)$ -Wert um $O(n^{1/2})$ senken
- $c_C(x)$ -Wert um $O(n^{1/2})$ erhöhen

1-Schritte können $c_C(x)$ -Wert **nicht** erhöhen,
weil neue Dreieckskante an Gewicht Cliquenkanten überwiegt

1-Schritte können schlechtes Dreieck entfernen, indem

- Kante entfernt wird $\rightsquigarrow c_D(x)$ -Wert steigt
- Kante eingefügt wird \rightsquigarrow Gewicht wächst um $2n^2$

also auf jeden Fall **nötig**: $c_C(x)$ -Wert muss gleichzeitig sinken

anfangs $c_C(x) = 1$ und $c_C(x)$ wächst nur $O(n^{1/2})$ Mal

also insgesamt $b(x)$ -Wert sinkt um $O(n^{1/2})$ und bleibt $\Theta(n)$

Ende 1. Phase: $c_G(x) = 1$ (wie früher) und $b(x) = \Theta(n)$

Die zweite Phase

eingangs $c_G(x) = 1$ und $b(x) = \Theta(n)$

$c_G(x) = 1$ bleibt erhalten

$b(x)$ -Wert kann nur in 2-Schritten sinken

Prob (senke $ct(x)$ -Wert) $\geq \frac{1}{2} \cdot \frac{3ct(x)}{m}$

$ct(x)$ -Wert immer $\leq n/4$

im Erwartungswert nach $O(m \log m)$ Schritten $ct(x) = 0$

analog in dieser Zeit wird Clique zum Baum

Ende 2. Phase: $c_G(x) = 1$, $b(x) = \Theta(n)$, x ist Baum

Nach der zweiten Phase

$c_G(x) = 1$, $b(x) = \Theta(n)$, x ist Baum

1-Schritte **nicht mehr akzeptiert**

2-Schritte **nur akzeptiert**,

wenn schlechtes Dreieck \rightsquigarrow gutes Dreieck

Prob ($b(x)$ -Wert sinkender 2-Schritt) = $\Theta\left(\frac{b(x)}{m^2}\right)$

Summe der erwarteten Wartezeiten

$$\sum_{b=1}^{\Theta(n)} \frac{m^2}{b} = n^4 \sum_{b=1}^{\Theta(n)} \frac{1}{b} = \Theta(n^4 \log n)$$

□

Anmerkung

Experimente auf zufälligen Graphen

\rightsquigarrow RLS langsamer als auf G_n

(1+1)-EA auf G_n

Theorem 13.14

(1+1)-EA hat auf G_n eine erwartete Optimierzeit von $\Theta(n^4 \log n)$.

Beweis.

obere Schranke: wie gesehen ✓

für untere Schranke:

Wir gehen den Beweis für RLS noch einmal durch und wollen möglichst viel **noch einmal verwenden**.

klar Initialisierung unverändert

also nach Initialisierung $b(x) = \Theta(n)$ und $c_C(x) = 1$

(1+1)-EA auf G_n nach der Initialisierung

Begriff k -Schritt: k Dreiecksseiten flippen

$k \in \{1, 2\}$:

$$\text{Prob}(k\text{-Schritt}) = \binom{3(n/4)}{k} \cdot \left(\frac{1}{m}\right)^k \cdot \left(1 - \frac{1}{m}\right)^{3(n/4)-k} = \Theta\left(\frac{1}{n^k}\right)$$

$k \geq 3$:

$$\text{Prob}(k\text{-Schritt}) = \Theta\left(\frac{1}{n^k}\right) \text{ für konstante } k$$

Betrachte Phase der Länge $n^{5/2}$.

$$\# \text{ 1-Schritte} = \Theta(n^{3/2}), \# \text{ 2-Schritte} = \Theta(n^{1/2}) \quad (\text{Chernoff})$$

Vorsicht in k -Schritt können in Clique weitere Kanten flippen

Die erste Phase

$\Theta(n^{1/2})$ 2-Schritte können

- $b(x)$ -Wert um $O(n^{1/2})$ senken
- $c_C(x)$ -Wert um $O(n^{1/2})$ erhöhen

1-Schritte können $c_C(x)$ -Wert **nicht** erhöhen,
weil neue Dreieckskante an Gewicht Cliquenkanten überwiegt

1-Schritte können schlechtes Dreieck entfernen, indem

- Kante entfernt wird $\rightsquigarrow c_D(x)$ -Wert steigt
- Kante eingefügt wird \rightsquigarrow Gewicht wächst um $2n^2$

also auf jeden Fall **nötig**: $c_C(x)$ -Wert muss gleichzeitig sinken

anfangs $c_C(x) = 1$ und $c_C(x)$ wächst nur $O(n^{1/2})$ Mal

also insgesamt $b(x)$ -Wert sinkt um $O(n^{1/2})$ und bleibt $\Theta(n)$

Ende 1. Phase: $c_G(x) = 1$ (wie früher) und $b(x) = \Theta(n)$

Die zweite Phase

eingangs $c_G(x) = 1$ und $b(x) = \Theta(n)$

$c_G(x) = 1$ bleibt erhalten

$b(x)$ -Wert kann nur in 2-Schritten sinken

Prob (senke $ct(x)$ -Wert) $\geq \binom{3ct(x)}{1} \cdot \frac{1}{m} \left(1 - \frac{1}{m}\right)^{m-1}$

$ct(x)$ -Wert immer $\leq n/4$

im Erwartungswert nach $O(m \log m)$ Schritten $ct(x) = 0$

analog in dieser Zeit wird Clique zum Baum

Ende 2. Phase: $c_G(x) = 1$, $b(x) = \Theta(n)$, x ist Baum

Nach der zweiten Phase

$c_G(x) = 1$, $b(x) = \Theta(n)$, x ist Baum

1-Schritte **nicht mehr akzeptiert**

2-Schritte **nur akzeptiert**,

wenn schlechtes Dreieck \rightsquigarrow gutes Dreieck

Prob ($b(x)$ -Wert sinkender 2-Schritt) = $\Theta\left(\frac{b(x)}{m^2}\right)$

Summe der erwarteten Wartezeiten

~~$$\sum_{b=1}^{\Theta(n)} \frac{m^2}{b} = n^4 \sum_{b=1}^{\Theta(n)} \frac{1}{b} = \Theta(n^4 \log n)$$~~

Neu: k -Schritte mit $k > 2$ können wichtig werden

Der (1+1)-EA nach der zweiten Phase

$c_G(x) = 1$, $b(x) = \Theta(n)$, x ist Baum

1-Schritte **nicht mehr akzeptiert**

2-Schritte **nur akzeptiert**, wenn schlechtes $\Delta \rightsquigarrow$ gutes Δ

Betrachte nur $\Theta(n^4 \log n)$ Schritte, $\text{Prob}(k\text{-Schritt}) = O(n^{-k})$

also keine k -Schritte mit $k \geq 5$

3-Schritte **nicht mehr akzeptiert**

manche 4-Schritte ändern $b(x)$ -Wert nicht \rightsquigarrow **egal**

manche 4-Schritte senken $b(x)$ -Wert um 2

$\text{Prob}(\text{dafür}) = \Theta\left(\frac{b(x)^2}{m^4}\right) = \Theta\left(\frac{b(x)^2}{n^8}\right) = O\left(\frac{1}{n^6}\right)$

nicht in $O(n^4 \log n)$ Schritten

also Summe der Wartezeiten asymptotisch auch $\Omega(n^4 \log n)$



RLS und (1+1)-EA auf f_4

$$f_4(x) = (c(x) - 1) \cdot w_b + w(E_x) \rightarrow \min$$

leicht zu sehen im Vergleich zu f_3

untere Schranke auf G_n **unverändert**

Beweis für Finden von zsh. Graph

ganz unverändert

Finden irgendeines Spannbaums

funktioniert so nicht mehr (nicht schlimm)

Obere Schranke für RLS und (1+1)-EA auf f_4

immer noch

$$n \text{ gute 2-Bit-Flips mit Durchschnittsgewinn} \geq \frac{w(E_x) - w_{\text{opt}}}{n}$$

neu

$$m - (n - 1) \text{ gute 1-Bit Flips mit Durchschn.gewinn} \\ \geq \frac{w(E_x) - w_{\text{opt}}}{m}$$

analoge Rechnung führt zu $O(m^2(\log(n) + \log(w_{\text{max}})))$

Theorem

Die erwartete Optimierzeit von RLS und dem (1+1)EA auf MST mit f_4 ist $O(m^2(\log m + \log w_{\text{max}}))$ und $\Omega(m^2 \log m)$.

