Bachelor-Seminarvortrag über das

Kapitel 9 Approximate Counting

aus dem Buch *Finite Markov Chains and Algorithmic Applications*

von *Olle Häggström*

Verfasser: Julius Witte

Studiengang: 1-Fach Bachelor im 6. Semester

Dozent: Prof. Dr. Matthias Löwe

Datum: 06.05.2012

Kapitel 9: Approximate Counting

Die Kombinatorik handelt von endlichen Objekten und Mengen, wobei wir uns hier mit Permutationen und Graphen beschäftigen. Die Hauptfragestellung ist dann immer die gleiche: *Wir haben eine Menge S. Welche Anzahl an Elementen besitzt diese?* Wir wollen nun folgendes Beispiel kennenlernen, welches uns durch das ganze Kapitel begleiten wird.

Beispiel 9.1: *Sei $q \in \mathbb{N}$ und G=(V,E) ein Graph. Wieviele q-Färbungen gibt es für G? $\rightarrow$ S entspricht der Menge der q-Färbungen auf G.*

Definition: *Ein* Graph G=(V,E) *besteht aus der Kantenmenge E und der Eckenmenge V. Falls eine Kante 2 Ecken miteinander verbindet, so heißen diese Nachbarn. 2 verschiedene Ecken können höchstens von einer Kante verbunden werden.*

Definition: *Eine* q-Färbung ($q \geq 2$) *auf einem Graph G=(V,E) ist eine Anordnung von den Farben $\{1, \ldots, q\}$, sodass kein Nachbar einer Ecke dieselbe Farbe annimmt wie die Ecke selbst. Die q-Färbungen sind gleichverteilt auf der Menge aller möglichen q-Färbungen auf G. Dann ist*

$$p_{G,q}(\xi) = \begin{cases} \frac{1}{Z} & \xi \text{ ist q-Färbung auf } G_{j-1}, \\ 0 & \text{sonst.} \end{cases}$$

die Gleichverteilung, wobei Z der Anzahl aller q-Färbungen auf G entspricht.

Wir möchten uns in diesem Kapitel mit Algorithmen beschäftigen, die das Zählproblem aus Beispiel 9.1 lösen. Durch die folgende Betrachtung des *naiven Algorithmus* wollen wir eine Motivation für einen geeigneten und guten Algorithmus erreichen:

Beispiel 9.2: Beim *naiven Algorithmus* geht man wie folgt vor:

- man betrachtet alle q^k möglichen Konfigurationen $\xi \in \{1, \ldots, q\}^V$ auf G
- falls ξ eine q-Färbung ist, macht man einen Strich
- am Ende zählt man alle Striche zusammen und erhält somit die Anzahl aller q-Färbungen auf G

Bemerkung: Der naive Algorithmus liefert stets die exakte Anzahl, die Laufzeit jedoch wächst exponentiell, weil man jedes Mal q^k Konfigurationen überprüfen muss. Also ist er nur für kleine k bzw. kleine Graphen G geeignet.

Dies motiviert uns, schnellere Algorithmen zu finden. Deshalb betrachten wir nun einen Algorithmus mit polynomieller Laufzeit, das bedeutet, dass ihre Laufzeit durch ein Polynom p(k) für alle bel. Größen k des Zählproblems begrenzt ist. Wenn mit großer Wahrscheinlichkeit dieser existieren sollen, müssen wir natürlich auch eine geringere Genauigkeit in Kauf nehmen:

- gehe über vom Zählen zum Approximieren
- erlaube dem Algorithmus, unter einer bestimmten W'keit (erheblich) falsche Werte zu liefern

Wir kommen deshalb zu folgender Definition:

Definition 9.3: *Ein Algorithmus mit folgenden Eigenschaften heißt* randomisiertes polynomielle-Zeit Approximations Schema*:*

1. *der Algorithmus liefert mit mindestens $\frac{2}{3}$ W'keit einen Wert zwischen $(1-\epsilon)N$ und $(1+\epsilon)N$, wobei N das wahre Ergebnis des Zählproblems ist*
2. *$\forall \epsilon > 0$ existiert ein Polynom $p_\epsilon(k)$, sodass für alle bel. Größen k des Zählproblems der Algorithmus nach maximal $p_\epsilon(k)$ Schritten abbricht*

Warum fordert man ausgerechnet $\frac{2}{3}$ W'keit? In Wahrheit ist nichts spezielles an $\frac{2}{3}$, man kann eigentlich jede beliebige Zahl aus $\left(\frac{1}{2}, 1\right)$ wählen. Falls wir aber eine W'keit sehr nah an 1 fordern, so muss unser Algorithmus erheblich länger laufen und damit dieser mit den Eigenschaften aus Definition 9.3 existiert, muss man auch einen größeren Fehler ϵ tolerieren.
Jetzt kommen wir zum Haupteil des Kapitels:

Theorem 9.4: *Man habe feste $q, d \in \mathbb{N}$, sodass $d \geq 2$ und $q > 2d^2$. Betrachte das Zählen von q-Färbungen auf einem Graph G, dessen Ecken jeweils höchstens d Nachbarn haben. Dann existiert ein randomisiertes polynomielle-Zeit Approximations Schema.*

Bemerkung:

- eine explizite Beschreibung des Algorithmus liefert der 1.Teil des Beweis. Im 2.Teil zeigt man daraufhin, dass genau dieser der Definition 9.3 genügt
- $d \geq 2$ ist keine ernste Einschränkung:

- 1.Fall: d = 0

  Die Ecken haben keine Nachbarn.
  → man hat q^k mögliche q-Färbungen
- 2.Fall: d = 1

  Jede bel. Ecke hat also höchstens einen Nachbarn.
  → Graph existiert nur aus:

  1. s isolierten Ecken, also solchen die keinen Nachbarn haben
  2. l Paaren von Ecken, die genau durch eine Kante verbunden sind

  → es gibt $q^s q^l (q-1)^l = q^{s+l}(q-1)^l$ mögliche q-Färbungen

1.Teil des Beweises von Theorem 9.5:
Sei G=(V,E) ein Graph mit der Kantenmenge $E = \{e_1, \ldots, e_{\tilde{k}}\}$. Da nach Annahme des Theorems von jeder Ecke höchstens d Kanten ausgehen können, folgt dann für insgesamt k Ecken, dass $\tilde{k} \geq dk$.
Definiere die Untergraphen G_j=$(V, \{e_1, \ldots, e_j\})$ $\forall j = 0, \ldots, \tilde{k}$. Insbesondere ist G = $G_{\tilde{k}}$.
Sei Z_j die Anzahl der q-Färbungen auf G_j → wir suchen also $Z_{\tilde{k}}$.
Man schreibe $Z_{\tilde{k}}$ durch das Teleskopprodukt um:

$$Z_{\tilde{k}} = \frac{Z_{\tilde{k}}}{Z_{\tilde{k}-1}} \cdot \frac{Z_{\tilde{k}-1}}{Z_{\tilde{k}-2}} \cdot \ldots \cdot \frac{Z_1}{Z_0} \cdot Z_0. \tag{1}$$

Warum schreiben wir nun $Z_{\tilde{k}}$ als Teleskopprodukt? Wir möchten einen Algoritmus finden der unser $Z_{\tilde{k}}$ schätzt. Wenn wir also jeden Faktor einigermaßen genau schätzen können, so haben wir auch mit einer bestimmten Genauigkeit einen Schätzer für unser $Z_{\tilde{k}}$.
G_0 hat keine Kanten → $Z_0 = q^k$.
Betrachte also alle Quotienten $\frac{Z_j}{Z_{j-1}}$ für $j = 1, \ldots, \tilde{k}$:
Seien x_j, y_j die Ecken, die die Kante e_j verbindet. Dann kommen wir jetzt zur wichtigsten Erkenntnis unseres Beweises:
Die q-Färbungen auf G_j sind genau die q-Färbungen ξ auf G_{j-1}, welche $\xi(x_j) \neq \xi(y_j)$ erfüllen. Dabei beschreibt $\xi(v)$ die Farbe der q-Färbung ξ an der Ecke v. Zur Erklärung der obigen Aussage:
Wenn wir eine q-Färbung auf G_{j-1} haben, bedeutet dies, dass die Bedingungen einer q-Färbung bzgl. der Kanten $e_1, \ldots, e_{j-1}$ erfüllt sind. Ecken, die noch durch keine Kante verbunden sind und somit auch keine Nachbarn haben, haben die Möglichkeit aus allen q Farben zu wählen. Fügen wir jetzt eine weitere Kante e_j bel. hinzu und erfüllen wir die Bedingung $\xi(x_j) \neq \xi(y_j)$, also dass die benachbarten Ecken x_j und y_j nicht die gleiche Farbe annehmen,

dann haben wir eine gültige q-Färbung bzgl. der Kanten $e_1, \ldots, e_j$ und somit eine auf G_j. Folglich beschreibt $\frac{Z_j}{Z_{j-1}}$ den Anteil der q-Färbungen auf G_{j-1}, die $\xi(x_j) \neq \xi(y_j)$ erfüllen.

Weiterhin ist $p_{G_{j-1},q}(\xi) = \begin{cases} \frac{1}{Z_{j-1}} & \xi \text{ ist q-Färbung auf } G_{j-1}, \\ 0 & \text{sonst.} \end{cases}$

Nach obiger Aussage ist $Z_j = |\{\text{X ist q-Färbung auf } G_{j-1} | X(x_j) \neq X(y_j)\}|$

$$\Rightarrow \frac{Z_j}{Z_{j-1}} = p_{G_{j-1},q}(X(x_j) \neq X(y_j)). \tag{2}$$

Diese Gleichung liefert uns einen Algorithmus für $\frac{Z_j}{Z_{j-1}}$: Wir haben bereits gesehen, dass wir unsere Gleichverteilung $p_{G_{j-1},q}$ mithilfe des *Gibbs sampler* schätzen können. Weil wir nun aber noch den Anteil der Konfigurationen X schätzen wollen, die $X(x_j) \neq X(y_j)$ erfüllen, also eine q-Färbung auf G_j sind, gehen wir wie folgt vor:

1. starte Gibbs sampler in bel., aber fester Konfiguration $\xi \in \{1, \ldots, q\}^V$ und lasse ihn n Schritte laufen
2. wiederhole diese Simulation, also das Durchführen von 1., endlich oft und betrachte den Anteil der Endkonfigurationen, die q-Färbungen sind. Die Endkonfiguration ist dabei die Konfiguration die man im n-ten Schritt des Gibbs sampler erhält.

Wenn wir diesen Algorithmus also für alle $j = 1, \ldots, \tilde{k}$ durchführen, erhalten wir einen für $Z_{\tilde{k}}$. □

Für den 2.Teil des Beweises benötigen wir zuerst 3 Hilfslemmata:

Lemma 9.5: $\epsilon \in [0,1]$, $k \in \mathbb{N}$ *und* $a_1, \ldots, a_k, b_1, \ldots, b_k > 0$, *sodass*

$$(1 - \frac{\epsilon}{2k}) \leq \frac{a_j}{b_j} \leq (1 + \frac{\epsilon}{2k}) \text{ für } j = 1, \ldots, k.$$

Dann gilt für $a = \prod_{i=1}^k a_i$ *und* $b = \prod_{i=1}^k b_i$, *dass*

$$(1 - \epsilon) \leq \frac{a}{b} \leq (1 + \epsilon).$$

Beweis: Man erkennt leicht, dass aus

$$\left(1 - \frac{\epsilon}{2k}\right)^2 \geq \left(1 - \frac{2\epsilon}{2k}\right)$$

$$\Rightarrow \quad \left(1-\frac{\epsilon}{2k}\right)^3 \geq \left(1-\frac{\epsilon}{2k}\right)\left(1-\frac{2\epsilon}{2k}\right) \geq \left(1-\frac{3\epsilon}{2k}\right).$$

Führt man dies weiter fort, so erhält man

$$\left(1-\frac{\epsilon}{2k}\right)^k \geq \left(1-\frac{k\epsilon}{2k}\right).$$

Deswegen ist

$$\begin{aligned}\frac{a}{b} &= \prod_{j=1}^{k}\frac{a_j}{b_j} \geq \prod_{j=1}^{k}\left(1-\frac{\epsilon}{2k}\right) = \left(1-\frac{\epsilon}{2k}\right)^k \\ &\geq 1-\frac{k\epsilon}{2k} = 1-\frac{\epsilon}{2} \geq 1-\epsilon.\end{aligned}$$

Für die zweite Ungleichung benutzen wir, dass $\exp(x) \geq x+1 \ \forall x \in \mathbb{R}$ und $\exp(\frac{x}{2}) \leq x+1 \ \forall x \in [0,1]$.

$$\begin{aligned}\Rightarrow \frac{a}{b} &= \prod_{j=1}^{k}\frac{a_j}{b_j} = \prod_{j=1}^{k}\left(1+\frac{\epsilon}{2k}\right) \leq \prod_{j=1}^{k}\exp\left(\frac{\epsilon}{2k}\right) \\ &= \exp\left(\frac{\epsilon}{2}\right) \leq 1+\epsilon.\end{aligned}$$

□

Lemma 9.6: *Sei $d, q \in \mathbb{N}$ mit $d \geq 2$ und $q > 2d^2$. Sei G=(V,E) Graph, dessen Ecken jeweils nicht mehr als d Nachbarn haben. Dann gilt für zufällige q-Färbungen X, dass*

$$p_{G,q}(X(x) \neq X(y)) \geq \frac{1}{2} \ \forall x, y \in V.$$

Beweis: Seien o.B.d.A. x und y keine Nachbarn, da Nachbarn nie dieselbe Farbe haben dürfen und somit immer $p_{G,q}(X(x) \neq X(y)) = 1$ ist. Betrachte zuerst die q-Färb. $X(V-\{x\})$ auf allen Ecken außer x und dann erst die Farbe bei x:
$p_{G,q}$ gleichverteilt $\rightarrow p_{G,q}(X(x) = .|X(V-\{x\})$ ist gleichverteilt auf allen Farben die kein Nachbar von x annimmt.
Wir können aus mind. $q-d$ Farben bei x wählen, da wir höchstens d Nachbarn haben.
$\rightarrow p_{G,q}(X(x) = .|X(V-\{x\}) \leq \frac{1}{q-d}$ für alle festen, aber bel. $X(V-\{x\})$.
Wir nehmen uns jetzt eine bel. Ecke y aus $X(V-\{x\})$, die kein Nachbar

von x ist. Wegen der Beliebigkeit von $X(V-\{x\})$ können wir dann jede bel. Farbe bei y betrachten und es gilt stets: $p_{G,q}(X(x)=X(y)) \leq \frac{1}{q-d}$.

$$\begin{aligned}\Rightarrow p_{G,q}(X(x)\neq X(y)) &= 1-p_{G,q}(X(x)=X(y))\\ &\geq 1-\frac{1}{q-d} \geq 1-\frac{1}{2d^2-d}\\ &= 1-\frac{1}{(2d-1)d} \geq \frac{1}{2}.\end{aligned}$$

□

Lemma 9.7: $n \in \mathbb{N}$, $p \in [0,1]$ *fest und* $H_j \sim B(n,p)$.

$$\Rightarrow \forall a > 0 \text{ } \textit{gilt } \mathbb{P}\left(|H_j - np|\right) \leq \frac{n}{4a^2}.$$

Beweis: Wir wissen, dass $\mathbb{E}[H] = np$ und $\mathbb{V}(H) = np(1-p)$ ist. Mit der Chebyshev-Ungleichung gilt dann:

$$\mathbb{P}\left(|H-np| \geq a\right) \leq \frac{np\,(1-p)}{a^2} \leq \frac{n}{4a^2},$$

da $p\,(1-p) \leq \frac{1}{4}$ ist $\forall p \in [0,1]$. □

2.Teil des Beweises von Theorem 9.4:
Sei Y_j Algorithmus von $\frac{Z_j}{Z_{j-1}}$ und $Y = \prod_{i=1}^{\tilde{k}} Y_i$. Dann ist $Y^* = YZ_0$ der gesuchte Algorithmus von $Z_{\tilde{k}}$.
Wir möchten nun, dass unser Algorithmus Y^* der Definition 9.3 genügt, aber nicht direkt mit diesem arbeiten, sondern mit den Faktoren Y_j. Deshalb müssen wir uns zum Beispiel fragen, welchen Fehler jeder Algorithmus Y_j für $j = 1,\ldots,\tilde{k}$ einhalten muss, damit unser Y^* den festgelegten Fehler ϵ einhält. Das zeigen wir nun durch folgende Annahme:

Annahme: Falls

$$\left(1-\frac{\epsilon}{2\tilde{k}}\right)\frac{Z_j}{Z_{j-1}} \leq Y_j \leq \left(1+\frac{\epsilon}{2\tilde{k}}\right)\frac{Z_j}{Z_{j-1}} \;\forall j = 1,\ldots,\tilde{k} \tag{3}$$

gilt, dann ist auch

$$(1-\epsilon)\,Z_{\tilde{k}} \leq Y^* \leq (1+\epsilon)\,Z_{\tilde{k}}.$$

Beweis:

$$\left(1-\frac{\epsilon}{2\tilde{k}}\right)\frac{Z_j}{Z_{j-1}} \leq Y_j \leq \left(1+\frac{\epsilon}{2\tilde{k}}\right)\frac{Z_j}{Z_{j-1}}$$
$$\Leftrightarrow \quad \left(1-\frac{\epsilon}{2\tilde{k}}\right) \leq \frac{Y_j}{Z_j/Z_{j-1}} \leq \left(1+\frac{\epsilon}{2\tilde{k}}\right)$$

Mit Lemma 9.5 folgt dann

$$\Leftrightarrow \quad (1-\epsilon) \leq \frac{Y}{\prod_{j=1}^{\tilde{k}} \frac{Z_j}{Z_{j-1}}} \leq (1+\epsilon)$$
$$\Leftrightarrow \quad (1-\epsilon) \leq \frac{Y Z_0}{Z_{\tilde{k}}} \leq (1+\epsilon)$$
$$\Leftrightarrow \quad (1-\epsilon)\, Z_{\tilde{k}} \leq Y^* \leq (1+\epsilon)\, Z_{\tilde{k}}.$$

Also haben wir jetzt eine Schranke an unser Y_j, welche wir noch etwas umschreiben und verschärfen wollen. Aus (3) ergibt sich

$$-\frac{\epsilon}{2\tilde{k}}\frac{Z_j}{Z_{j-1}} \leq Y_j - \frac{Z_j}{Z_{j-1}} \leq \frac{\epsilon}{2\tilde{k}}\frac{Z_j}{Z_{j-1}}.$$

Aus (2) und Lemma 9.6 folgt

$$\frac{Z_j}{Z_{j-1}} = p_{G_{j-1},q}(X(x_j) \neq X(y_j)) \geq \frac{1}{2}$$

$$\Rightarrow \quad \frac{\epsilon}{4\tilde{k}} \leq \frac{\epsilon}{2\tilde{k}}\frac{Z_j}{Z_{j-1}}.$$

Falls also

$$\left|Y_j - \frac{Z_j}{Z_{j-1}}\right| \leq \frac{\epsilon}{4\tilde{k}} \ \forall j = 1, \ldots, \tilde{k} \tag{4}$$

gilt, dann erfüllt Y_j für alle j auch die Schranke aus (3).

Betrachten wir jetzt nochmal den Vorgang unseres Algorithmus für Y_j, dann stoßen wir auf folgende 2 Fehlerquellen:

1. Wir lassen den Gibbs sampler endlich viele Schritte n laufen, sodass die Verteilung $\mu^{(n)}$, die die Färbung im n-ten Schritt erzeugt, sich von der stationären Verteilung $p_{G_{j-1},q}$ unterscheiden kann.

2. Wir führen nur endlich viele Simulationen durch, sodass sich der Anteil der q-Färbungen Y_j von dem zu erwartenden Wert $\mu^{(n)}(X(x_j) \neq X(y_j))$ unterscheiden kann. Dies lässt sich gut mit einem Münzwurf vergleichen: Wenn wir eine faire Münze 100-mal werfen, dann werden wir auch nicht immer 50-mal Kopf sehen.

Um (4) weiterhin einzuhalten, teilen wir den Fehler gleichmäßig auf:

$$\Rightarrow \quad \left|\mu^{(n)}(X(x_j) \neq X(y_j)) - p_{G_{j-1},q}(X(x_j) \neq X(y_j))\right| \leq \frac{\epsilon}{8\tilde{k}} \tag{5}$$

$$\left|Y_j - \mu^{(n)}(X(x_j) \neq X(y_j))\right| \leq \frac{\epsilon}{8\tilde{k}} \tag{6}$$

(I) Wieviele Schritte n benötigt der Gibbs sampler, damit (5) *erfüllt ist?*

Als erstes betrachten wir den Zusammenhang zur *Total-Variation*:

$$\begin{aligned} & \left|\mu^{(n)}(X(x_j) \neq X(y_j)) - p_{G_{j-1},q}(X(x_j) \neq X(y_j))\right| \\ \leq & \max_{A \subseteq \{1,\dots,q\}^V} \left|\mu^{(n)}(A) - p_{G_{j-1},q}(A)\right| \\ = & \; d_{TV}(\mu^{(n)}, p_{G_{j-1},q}). \end{aligned}$$

Es genügt also ein $n \in \mathbb{N}$ zu finden, sodass $\forall m \geq n$ gilt:

$$d_{TV}(\mu^{(m)}, p_{G_{j-1},q}) \leq \frac{\epsilon}{8\tilde{k}}.$$

Dieses n liefert uns das Theorem 8.1 aus dem letzten Vortrag:

$$n = k\left(\frac{\log k + \log \frac{8\tilde{k}}{\epsilon} - \log d}{\log \frac{q}{2d^2}} + 1\right),$$

mit $\tilde{k} \leq dk$ folgt

$$\begin{aligned} n &\leq k\left(\frac{\log k + \log \frac{8dk}{\epsilon} - \log d}{\log \frac{q}{2d^2}} + 1\right) \\ &= k\left(\frac{2\log k + \log \frac{1}{\epsilon} + \log 8}{\log \frac{q}{2d^2}} + 1\right). \end{aligned}$$

(II) Wieviele Simulationen m benötigen wir, um (6) *zu erfüllen?*

Denken wir nochmal an die Definition 9.3 zurück. Dort wird gefordert, dass der Algorithmus mit mindestens $\frac{2}{3}$ W'keit unseren festgelegten Fehler ϵ einhält. Über den erlaubten Fehler ϵ von Y^* sind wir schon an die Schranke (3)

für jedes Y_j gekommen. Wir benötigen nun noch die W'keit mit der jedes Y_j ihre Schranke einhält.
Damit unser Y^* mit höchstens $\frac{1}{3}$ W'keit außerhalb des multiplikativen Fehlerbereichs $(1-\epsilon, 1+\epsilon)$ liegt, darf jedes Y_j höchstens mit $\frac{1}{3\tilde{k}}$ W'keit ihre Schranke nicht einhalten, denn Y^* besteht, wie wir wissen, aus den $\tilde{k}$ Faktoren Y_j für $j = 1, \ldots, \tilde{k}$. Das werden wir gleich noch ausnutzen:
Weiterhin soll H_j die Anzahl der Endkonfigurationen angeben, die q-Färbungen sind $\rightarrow Y_j = \frac{H_j}{m}$.
Außerdem ist $p = \mu^{(n)}(X(x_j) \neq X(y_j))$ die zu erwartende W'keit, dass eine zufällige Konfiguration X eine q-Färbung ist. Wenn man sich die Analogie zum Münzwurf ins Gedächtnis ruft, dann sieht man, dass $H_j \sim B(m,p)$. Aus (6) folgt schließlich:

$$\begin{aligned} & \left| \frac{H_j}{m} - p \right| \leq \frac{\epsilon}{8\tilde{k}} \\ \Leftrightarrow \quad & |H_j - mp| \leq \frac{m\epsilon}{8\tilde{k}}. \end{aligned}$$

Wir betrachten jetzt die W'keit, dass diese Schranke nicht eingehalten wird, und benutzen unser Lemma 9.7:

$$\Rightarrow \quad \mathbb{P}\left(|H_j - mp| > \frac{m\epsilon}{8\tilde{k}} \right) \leq \frac{m}{4\left(\frac{m\epsilon}{8\tilde{k}}\right)^2} = \frac{16\tilde{k}^2}{\epsilon^2 m}.$$

Nun verwenden wir noch, was wir uns anfangs überlegt haben: Die W'keit, dass unser Y_j nicht die Schranke einhält, darf höchstens $\frac{1}{3\tilde{k}}$ betragen:

$$\Rightarrow \quad \frac{16\tilde{k}^2}{\epsilon^2 m} = \frac{1}{3\tilde{k}}$$

und Umstellen nach m und Einsetzen von $\tilde{k} \leq dk$ liefert

$$m = \frac{48\tilde{k}^3}{\epsilon^2} \leq \frac{48 d^3 k^3}{\epsilon^2}.$$

Zum Schluss müssen wir noch unsere Ergebnisse zusammentragen und schauen, ob der Algorithmus Y^*, sowie wir ihn im ersten Teil des Beweises beschrieben haben, auch wirklich ein *randomisiertes polynomielle-Zeit Approximations Schema* ist. Wir haben herausgefunden, dass unsere Simulationslaufzeit n und die Anzahl der Simulationen m von k und ϵ abhängen. Außerdem ist die Anzahl der Algorithmen Y_j nur von k abhängig, denn wir

wissen, dass $\tilde{k} \leq dk$. Die Gesamtanzahl der Schritte für den Algorithmus Y^* ist daher höchstens

$$dk \cdot \frac{48d^3k^3}{\epsilon^2} \cdot k \left(\frac{2\log k + \log \frac{1}{\epsilon} + \log 8}{\log \frac{q}{2d^2}} + 1 \right).$$

Dies ist von der Ordnung $Ck^5 \log k$ für $k \to \infty$ und eine von k unabhängige Konstante C. Da der $\log k \leq k$ ist, ist die Ordnung sogar geringer als Ck^6. Folglich haben wir ein Polynom $p_\epsilon(k)$ gefunden, das die Laufzeit des Algorithmus Y^* begrenzt. Also ist unser Algorithmus aus dem ersten Teil des Beweises ein *randomisiertes polynomielle-Zeit Approximations Schema*. □