Primfaktorzerlegung mit Simulated Annealing

Bachelor-Präsentation

Fabian Köhler

TU Dortmund

19.09.2014

Agenda

Definition: Primfaktorzerlegung

Definition

Die eindeutige Darstellung

$$N = \prod_{i=1}^{M} p_i^{m_i}.$$

einer Zahl $N \in \mathbb{N}$ mit Primzahlen $p_1, \ldots, p_M$ mit $p_i \neq p_j$ für $i \neq j$ und Exponenten $m_1, \ldots, m_M \in \mathbb{N}$ ist die *Primfaktorzerlegung* der Zahl N.

Definition: Primfaktorzerlegung

Definition

Die eindeutige Darstellung

$$N = \prod_{i=1}^{M} p_i^{m_i}.$$

einer Zahl $N \in \mathbb{N}$ mit Primzahlen $p_1, \ldots, p_M$ mit $p_i \neq p_j$ für $i \neq j$ und Exponenten $m_1, \ldots, m_M \in \mathbb{N}$ ist die *Primfaktorzerlegung* der Zahl N.

Zerlegung kann rekursiv aufgebaut werden:

$$N = A \cdot B$$

$\Rightarrow A, B$ weiterzerlegen

Bedeutung der Primfaktorzerlegung

Primfaktorzerlegung ist ein Problem der Komplexitätsklasse NP

- "Harte Probleme" generell interessant
  - Zerlegung in Faktoren nicht effizient möglich
  - Prüfung der Lösung schon
- Primfaktorzerlegung in der Kryptografie (z.B. RSA [1])

Lösungsansätze

- Zahlkörpersieb $\mathcal{O}\left(\exp\left(c \cdot (\log n)^{\frac{2}{3}} (\log\log n)^{\frac{1}{3}}\right)\right)$ [2]
- Shor-Algorithmus auf Quantencomputern
  $\mathcal{O}\left(n^2 \log n \log\log n\right)$ [3]
- Adiabatic Quantum Computing [4, 5]
- Simulated Annealing nach E.L. Altschuler und T.J. Williams [6]

Grundidee

- Zerlegung $N = A \cdot B$ mit $B \leq A$

Grundidee

- Zerlegung $N = A \cdot B$ mit $B \leq A$
- A, B und N werden als binäre Zahlen dargestellt

Grundidee

- Zerlegung $N = A \cdot B$ mit $B \leq A$
- A, B und N werden als binäre Zahlen dargestellt
- a, b und n sind die Längen der Zahlen ($\lceil \log_2 A \rceil$ usw.)

Grundidee

- Zerlegung $N = A \cdot B$ mit $B \leq A$
- A, B und N werden als binäre Zahlen dargestellt
- a, b und n sind die Längen der Zahlen ($\lceil \log_2 A \rceil$ usw.)
- a_1 und b_1 sind die Zahlen der 1en

$$\begin{aligned} 2 \leq a \leq n & \qquad 1 \leq a_1 \leq a \\ 2 \leq b \leq a & \qquad 1 \leq b_1 \leq b \end{aligned}$$

Grundidee

- Zerlegung $N = A \cdot B$ mit $B \leq A$
- A, B und N werden als binäre Zahlen dargestellt
- a, b und n sind die Längen der Zahlen ($\lceil \log_2 A \rceil$ usw.)
- a_1 und b_1 sind die Zahlen der 1en

$$\begin{aligned} 2 \leq a \leq n \qquad & 1 \leq a_1 \leq a \\ 2 \leq b \leq a \qquad & 1 \leq b_1 \leq b \end{aligned}$$

- Der Suchbereich kann eingeschränkt werden (ca. 16%)

$$a_{\min} = \begin{cases} 2 & \text{falls } \left\lfloor \frac{n}{2} \right\rfloor = 1 \\ \left\lfloor \frac{n}{2} \right\rfloor & \text{sonst} \end{cases}$$

$$b_{\min} = \begin{cases} 2 & \text{falls } n - a = 1 \\ n - a & \text{sonst} \end{cases}$$

Vorgehen

1. Initialisierung des Systems (A, B, E)

Vorgehen

1. Initialisierung des Systems (A,B,E)
2. Operationen zur Modifikation des Zustandes

Vorgehen

1. Initialisierung des Systems (A,B,E)
2. Operationen zur Modifikation des Zustandes
3. Algorithmus zur Akzeptanz oder Zurückweisung des neuen Zustandes (Metropolis)

Vorgehen

1. Initialisierung des Systems (A,B,E)
2. Operationen zur Modifikation des Zustandes
3. Algorithmus zur Akzeptanz oder Zurückweisung des neuen Zustandes (Metropolis)
   $\Rightarrow$ Einführung einer Energiedefinition

Vorgehen

1. Initialisierung des Systems (A,B,E)
2. Operationen zur Modifikation des Zustandes
3. Algorithmus zur Akzeptanz oder Zurückweisung des neuen Zustandes (Metropolis)
   $\Rightarrow$ Einführung einer Energiedefinition
4. Prüfung ob der Zustand eine Lösung ist

Energiedefinition

Es wird eine Energiedefinition eingeführt:

$$E\left(A,B,N\right)=\sum_{i=1}^{N}\begin{cases}f\left(i\right) & \text{falls}\{A\cdot B\}_i=\{N\}_i\\ 0 & \text{sonst}\end{cases}$$

Energiedefinition

Es wird eine Energiedefinition eingeführt:

$$E\left(A,B,N\right)=\sum_{i=1}^{N}\begin{cases}f\left(i\right) & \text{falls}\{A\cdot B\}_i=\{N\}_i\\ 0 & \text{sonst}\end{cases}$$

$f\left(i\right)$ ist eine monoton steigende Funktion.
$\Rightarrow$ Übereinstimmung von $A\cdot B$ mit N erhöht die Energie
$\Rightarrow$ Energie maximieren

Energiedefinition

Es wird eine Energiedefinition eingeführt:

$$E(A,B,N) = \sum_{i=1}^{N} \begin{cases} f(i) & \text{falls}\{A \cdot B\}_i = \{N\}_i \\ 0 & \text{sonst} \end{cases}$$

$f(i)$ ist eine monoton steigende Funktion.
$\Rightarrow$ Übereinstimmung von $A \cdot B$ mit N erhöht die Energie
$\Rightarrow$ Energie maximieren

$$f(i) = i \quad \Rightarrow \quad E_{\max}(n) = \frac{n(n+1)}{2}$$

$$f(i) = i^2 \quad \Rightarrow \quad E_{\max}(n) = \frac{n(n+1)(2n+1)}{6}$$ [7]

Operationen

- **Swap:** Tausche zwei zufällige Bits mit unterschiedlichem Wert

1	0	1	1	0	1	0	0

Operationen

- **Swap:** Tausche zwei zufällige Bits mit unterschiedlichem Wert

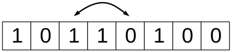
1	0	1	1	0	1	0	0

- **Reverse:** Zufällige Bitsequenz umkehren

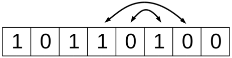
1	0	1	1	0	1	0	0

Operationen

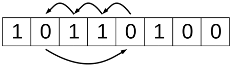
- **Swap:** Tausche zwei zufällige Bits mit unterschiedlichem Wert



- **Reverse:** Zufällige Bitsequenz umkehren

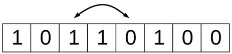


- **Slide:** Es wird eine zufällige Bitsequenz nach rechts geschoben

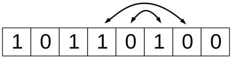


Operationen

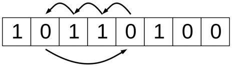
- **Swap:** Tausche zwei zufällige Bits mit unterschiedlichem Wert



- **Reverse:** Zufällige Bitsequenz umkehren



- **Slide:** Es wird eine zufällige Bitsequenz nach rechts geschoben



- **Shuffle:** Bits zufällig auswählen und permutieren

Operationen haben Laufzeit $\mathcal{O}\left(n\right)$

Metropolis-Algorithmus

procedure $\textsc{Metropolis}(A, B, E, N)$
  if $\text{randomInt}\,(0,1) = 0$ **then**
    $A' \leftarrow \text{randomOperation}\,(A)$
  else
    $B' \leftarrow \text{randomOperation}\,(B)$
  if $A \cdot B = N$ **then**
    Exit ▷ Faktoren A, B wurden gefunden
  $E' = E\,(A', B', N)$
  Acceptance probability: $p = \begin{cases} 1 & \text{falls } E' > E \\ \exp\left(\frac{E'-E}{k_\mathrm{B} T}\right) & \text{sonst} \end{cases}$

Annealing-Algorithmus

Parameter:

- N_a: Anzahl der Abkühlungsschritte
- N_c: Anzahl der Konfigurationen pro Abkühlungsschritt
- F_c: Abkühlungsfaktor

procedure $\textsc{Anneal}(A, B, E, N)$
    $T \leftarrow 1.0$
    for $i \leftarrow 1$ to N_a **do**
        for $j \leftarrow 1$ to N_c **do**
            $\text{Metropolis}\,(A, B, E, N)$
        $T \leftarrow T \cdot F_c$

Zerlegungsschritt

procedure FACTORIZE(N)
    $a_{\min}$ berechnen
    for $a \in [a_{\min}, n]$ **do**
        for $a_1 \in [1, a]$ **do**
            $A \leftarrow \text{randomBitset}\,(a, a_1)$
            $b_{\min}$ berechnen
            for $b \in [b_{\min}, a]$ **do**
                for $b_1 \in [1, b]$ **do**
                    $B \leftarrow \text{randomBitset}\,(b, b_1)$
                    $E \leftarrow E\,(A, B, E, N)$
                    if $A \cdot B = N$ **then**
                        Exit   ▷ Faktoren A, B wurden gefunden
                    $\text{anneal}\,(A, B, E, N)$

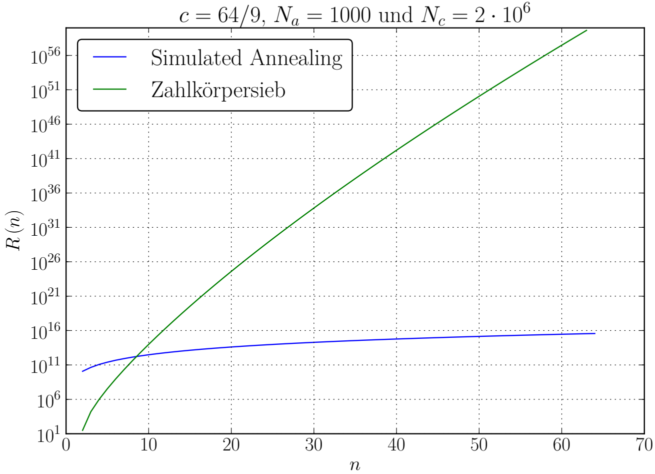
Abschätzung der Laufzeit I

Abschätzung der Worst-Case-Laufzeit:

- Annealing-Algorithmus wird $\mathcal{O}\left(n^4\right)$-mal ausgeführt (Wertebereiche von a, b, a_1 und b_1 skalieren grob mit $\mathcal{O}\left(n\right)$)
- dabei wird der Metropolis-Algorithmus $\mathcal{O}\left(N_a \cdot N_c\right)$-mal ausgeführt
- dort jeweils eine der 4 Operationen mit Laufzeit $\mathcal{O}\left(n\right)$

$\Rightarrow$ Laufzeit eines Zerlegungsschrittes $\mathcal{O}\left(n^5 \cdot N_a \cdot N_c\right)$

Abschätzung der Laufzeit II



Untersuchungen

- Abschätzung von k_{B}
- Einzelner Zerlegungsschritt
- Komplette Faktorisierung
- Parallelisierbarkeit

Abschätzung von k_{B}

- zwei Primzahlen $A = 104723$ und $B = 66889$ gewählt
- Semiprimzahl $N = A \cdot B = 7004816747$ mit $n = 33$

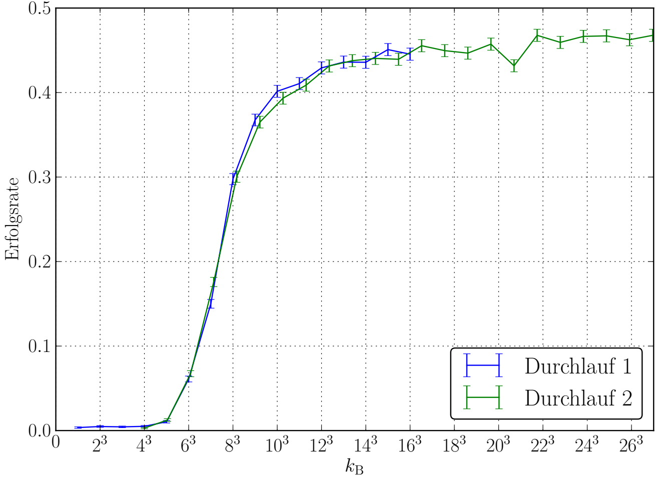
Abschätzung von k_B

- zwei Primzahlen $A = 104723$ und $B = 66889$ gewählt
- Semiprimzahl $N = A \cdot B = 7004816747$ mit $n = 33$
- verschiedene Werte $1^3 \leq k_\mathrm{B} \leq 16^3$
- für jeden Wert 4800 Wiederhohlungen

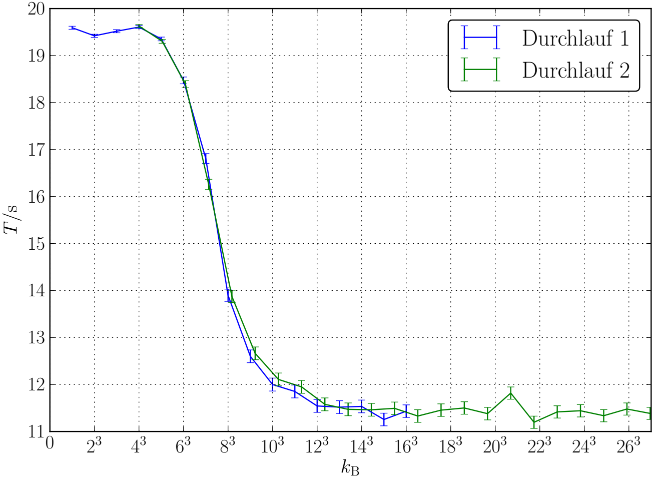
Abschätzung von k_B

- zwei Primzahlen $A = 104723$ und $B = 66889$ gewählt
- Semiprimzahl $N = A \cdot B = 7004816747$ mit $n = 33$
- verschiedene Werte $1^3 \leq k_\mathrm{B} \leq 16^3$
- für jeden Wert 4800 Wiederholungen
- $N_a = 1000$, $N_c = 80000$, $T_0 = 1$, $F_c = 0.997$
- Messung von Laufzeit und Erfolgsrate

Erfolgsrate



Laufzeit


Durchlauf 1
Durchlauf 2
T/s
k_B
20
19
18
17
16
15
14
13
12
11
0
2^3
4^3
6^3
8^3
10^3
12^3
14^3
16^3
18^3
20^3
22^3
24^3
26^3

Einzelner Zerlegungsschritt

- zufällige Semiprimzahlen/allgemeine Zahlen

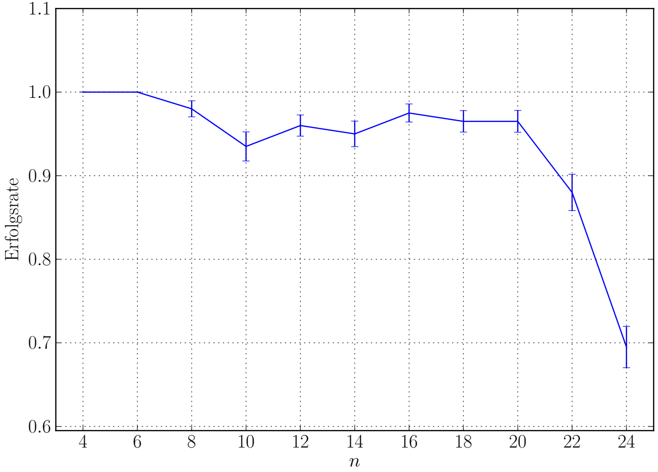
Einzelner Zerlegungsschritt

- zufällige Semiprimzahlen/allgemeine Zahlen
- $N_a = 500$, $N_c = 1000$, $T_0 = 1$, $F_c = 0.997$
- k_B automatisch abgeschätzt

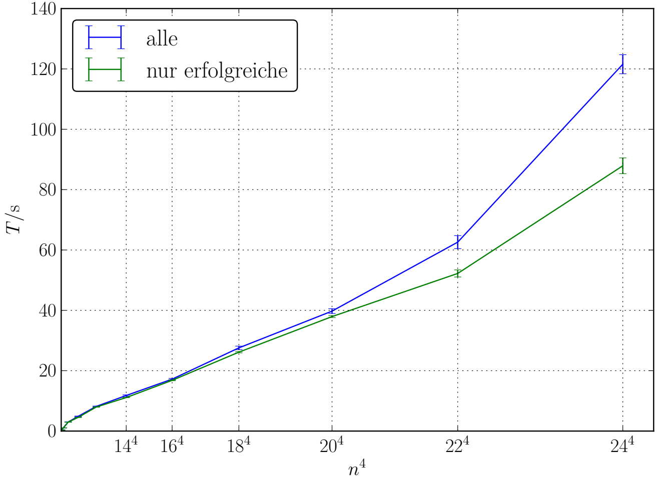
Einzelner Zerlegungsschritt

- zufällige Semiprimzahlen/allgemeine Zahlen
- $N_a = 500$, $N_c = 1000$, $T_0 = 1$, $F_c = 0.997$
- k_B automatisch abgeschätzt
- jede Zahl mehrfach zerlegen

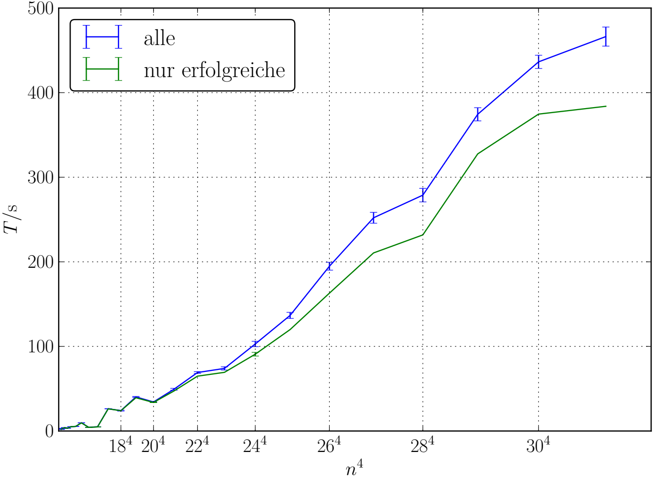
Semiprimzahlen: Erfolgsrate



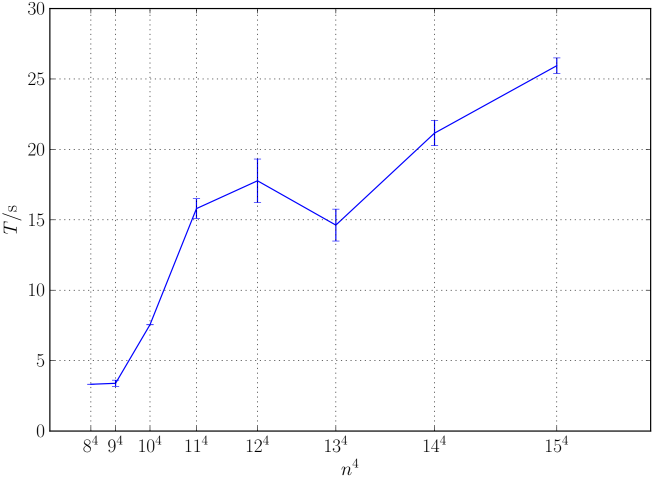
Semiprimzahlen: Laufzeit



Allgemeine Zahlen

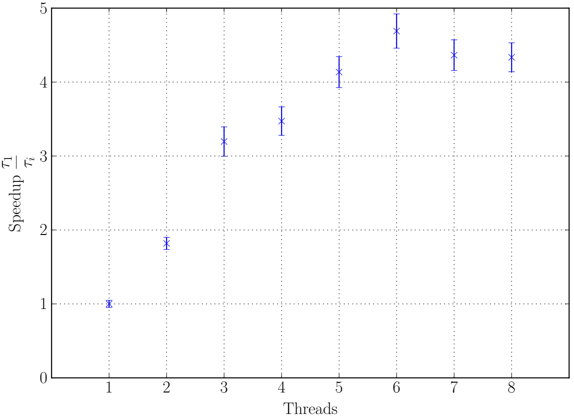


Komplette Faktorisierung



Parallelisierbarkeit

- Verteile Tupel (a, a_1, b, b_1) auf Threads
- $N = 783061$ 200-mal für verschiede Threadzahl zerlegen



Fazit & Ausblick I

Ergebnisse:

- Prinzipielle Funktionsfähigkeit des Verfahrens
- Laufzeit $\mathcal{O}\left(n^5 \cdot N_a \cdot N_c\right)$
- Optimaler Wert für k_B
- Gute Parallelisierbarkeit

Fazit & Ausblick II

Ausblick:

- Test weiterer Energiedefinitionen
- Optimierung der Operationen
- Andere Anfangstmeperaturen oder anderes F_c
- Einfluss von N_a und N_c
- Skalierungsverhalten auf großen Rechnersystemen (Cluster)

Arbeit und Code:

`https://github.com/f-koehler/bachelor-thesis`
`https://github.com/f-koehler/primefac`

Literatur I

[1] R. L. Rivest, A. Shamir und L. Adleman. „A method for obtaining digital signatures and public-key cryptosystems“. In: *Communications of the ACM* 21 (1978), S. 120–126. DOI: 10.1145/359340.359342.

[2] C. Pomerance. „A Tale of Two Sieves“. In: *Notices of the AMS* 43 (1996), S. 1473–1485. URL: http://www.ams.org/notices/199612/pomerance.pdf (besucht am 13. 06. 2014).

[3] P. W. Shor. „Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer“. In: *SIAM Journal on Computing* 26 (1996), S. 1484–1509. DOI: 10.1137/S0097539795293172.

Literatur II

[4] X. Peng, Z. Liao, N. Xu, G. Qin, X. Zhou, D. Suter und J. Du. „Quantum Adiabatic Algorithm for Factorization and Its Experimental Implementation“. In: *Physical Review Letters* 101 (2008). DOI: 10.1103/PhysRevLett.101.220405.

[5] N. Xu, J. Zhu, D. Lu, X. Zhou, X. Peng und J. Du. „Quantum Factorization of 143 on a Dipolar-Coupling NMR system“. In: *Physical Review Letters* 108 (2012). DOI: 10.1103/PhysRevLett.108.130501.

[6] E. L. Altschuler und T. J. Williams. *Using Simulated Annealing to Factor Numbers*. 17. Feb. 2014. arXiv: 1402.1201v2.

Literatur III

[7] N. J. A. Sloane. *Online Encyclopedia of Integer Sequences (OEIS): Square pyramidal numbers*. 11. März 2010. URL: http://oeis.org/A000330 (besucht am 13. 06. 2014).

Danke für ihre Aufmerksamkeit!

Haben sie Fragen?