

# Approximate Counting

## Ein FPRASC für #RUCKSACK

Florian Forster

<http://verplant.org/>

Friedrich Alexander Universität Erlangen-Nürnberg

Sommerschule Randomisierte Algorithmen, 2007-09-18

# Gliederung

## Einleitung

Motivation

Zählprobleme und #P-Vollständigkeit

Die Monte-Carlo-Methode

## Eine deterministische Approximation von #RUCKSACK

Ein pseudopolynomieller DP-Algorithmus für #RUCKSACK

Expansion

Zwischenergebnis

## Ein FPRASC für #RUCKSACK

Ein uniformer Generator

Laufzeit

Zusammenfassung

# Warum Zählprobleme?

- ▶ Physikalische Eigenschaften können durch Graph-Probleme modelliert werden
- ▶ Die Eigenschaften eines Stoffes hängen dabei von der **Anzahl** der möglichen Lösungen ab

# Zählprobleme

- ▶  $\#\Pi$  ist kombinatorisches Zählproblem zu dem Optimierungsproblem  $\Pi$
- ▶  $S(I)$  ist die Menge der Lösungen für  $\Pi$
- ▶ Anzahl  $\#(I) = |S(I)|$  ist Lösung von  $\#\Pi$

# Die Komplexitätsklasse #P

- ▶ In #P sind die Zählprobleme mit polynomzeit beschränktem Algorithmus, der die Zertifikate überprüfen kann
- ▶  $\Pi \in NP \Rightarrow \#\Pi \in \#P$
- ▶ Der Begriff der „Vollständigkeit“ kann in Polynomzeit reduziert werden

# Einige #P-Vollständige Probleme

#DNF	Anzahl der erfüllenden Lösungen für eine Boolesche-Formel in DNF
#COL <sub>k</sub>	Anzahl der Graphfärbungen mit $k$ Farben
#RUCKSACK	Anzahl der gültigen Rucksack-Füllungen → Dazu gleich mehr

# Die Monte-Carlo-Methode

- ▶  $S(I)$  wird zu einem Universum  $U$  erweitert, dessen Größe  $|U|$  bekannt ist
- ▶ Der Faktor der Erweiterung  $\xi = \frac{|U|}{\#(I)}$  heißt Expansion
- ▶ Ein gleichverteilter Generator zieht Elemente aus  $U$
- ▶ Aus dem Anteil der gezogenen Elemente die in  $S(I)$  liegen und  $|U|$  kann  $\#(I)$  abgeschätzt werden

## Definition von #RUCKSACK

- ▶ Eine Instanz  $I$  von #RUCKSACK ist charakterisiert durch die Rucksackgröße  $B$ , die Elemente  $W = \{w_1, w_2, \dots, w_{\max}\}$  und die Volumenfunktion  $\text{vol} : W \rightarrow \mathbb{N}$ . Der Wert der Elemente ist in der Zählvariante des Problems unerheblich
- ▶ Es gilt weiterhin:

$$B \geq \max \{ \text{vol}(w) \mid w \in W \}$$
$$B < \sum_{w \in W} \text{vol}(w)$$

# Ein exakter, pseudopolynomieller Algorithmus für #RUCKSACK

- Die Anzahl der gültigen Lösungen für einen Rucksack der Größe  $\beta$  mit den Elementen  $w_1 \dots w_j$ ,  $F_I(j, \beta)$ , kann rekursiv berechnet werden:

$$F_I(j, \beta) = \begin{cases} 0 & \beta < 0 \\ 1 & \beta \geq 0, j = 0 \\ F_I(j-1, \beta) + F_I(j-1, \beta - \text{vol}(w_j)) & \beta \geq 0, j \geq 1 \end{cases}$$

- ▶ Für einen Rucksack mit der Größe  $B$  und  $|W|$  Elementen hat der Algorithmus eine Laufzeit von  $O(B \cdot |W|)$
- ▶ Sei  $v_{\max} = \max \{\text{vol}(w) \mid w \in W\}$
- ▶ Es gilt:

$$\begin{aligned} |W| &\leq v_{\max} \\ B &< |W| \cdot v_{\max} \leq v_{\max}^2 \end{aligned}$$

- ▶ Die Laufzeit kann also auch als  $O(v_{\max}^3)$  angegeben werden
- ▶ Beachte jedoch, dass  $v_{\max} = 2^{O(n)}$

# Expansion

- ▶ Zu einer gegebenen Instanz  $I$  generieren wir nun eine „kleine“ Instanz  $I'$ :

$$\begin{aligned}
 W' &= W = \{w_1, w_2, \dots, w_{\max}\} \\
 \text{vol}'(w) &= \left\lfloor \frac{|W|^2}{B} \cdot \text{vol}(w) \right\rfloor \\
 B' &= |W|^2
 \end{aligned}$$

- ▶ Die Größe des größten Elements ist jetzt beschränkt durch:

$$v'_{\max} \leq B' = |W|^2$$

- ▶ Das heißt  $\#(I') = F_{I'}(|W|, |W|^2)$  kann in Zeit  $O(|W|^3)$  berechnet werden
- ▶ Das ist echt besser als  $O(v_{\max}^3)$ , da  $|W| = O(n)$

- Sei  $A$  eine Lösung für  $I$ , also  $A \in S(I)$ . Dann gilt:

$$\begin{aligned} \text{vol}'(A) &= \sum_{w \in A} \left[ \frac{|W|^2}{B} \cdot \text{vol}(w) \right] \\ &\leq \frac{|W|^2}{B} \cdot \sum_{w \in A} \text{vol}(w) \\ &\leq \frac{|W|^2}{B} \cdot B = |W|^2 \end{aligned}$$

$\Rightarrow A$  ist also auch eine Lösung für  $I'$

$\Rightarrow S(I) \subseteq S(I')$

# Größe der Expansion

- ▶ Sei  $A \in S(I') \setminus S(I)$
- ▶  $A$  muss ein Element  $w^*$  mit  $\text{vol}(w^*) \geq \frac{B}{|W|}$  enthalten
- ▶ Ohne dieses Element gilt:

$$\begin{aligned}
 \text{vol}(A \setminus w^*) &\leq \frac{B}{|W|^2} \cdot (\text{vol}'(A \setminus w^*) + |A \setminus w^*|) \\
 &= \frac{B}{|W|^2} \cdot (\text{vol}'(A) - \text{vol}'(w^*) + |A| - 1) \\
 &\leq \frac{B}{|W|^2} \cdot (|W|^2 - |W| + |W| - 1) \\
 &\leq B
 \end{aligned}$$

- Da zu jeder Lösung  $A \in S(I)$  „nur“ maximal  $|W|$  Lösungen  $A' \in S(I') \setminus S(I)$  angegeben werden können, gilt:

$$\begin{aligned} |S(I') \setminus S(I)| &\leq |W| \cdot |S(I)| \\ |S(I')| &\leq |W| \cdot |S(I)| + |S(I)| \\ |S(I')| &\leq |S(I)| \cdot (|W| + 1) \\ \Rightarrow \xi &\leq |W| + 1 \end{aligned}$$

## Zwischenergebnis

- ▶ Da  $\xi \leq |W| + 1$  haben wir also schon einen deterministischen Approximationsalgorithmus mit der relativen Güte:

$$A(I) = \frac{F_I(|W|, |W|^2)}{\sqrt{|W| + 1}}$$

## „Zutaten“ für die Monte-Carlo-Methode

- ▶ Wir haben  $S(I)$  zu einem Universum  $S(I')$  erweitert, das maximal um den Faktor  $\xi$  größer ist als  $S(I)$  selbst
- ▶ Um die Monte-Carlo-Methode anwenden zu können brauchen wir jetzt noch einen Generator, der gleichverteilt Elemente aus  $S(I')$  zieht

# Ein uniformer Generator

- Ein Generator kann wie folgt angegeben werden:  
**for**  $j = |W|$  **downto** 1

$$\left\{ \begin{array}{l} \frac{F_V(j-1, \beta - \text{vol}(w_j))}{F_V(j, \beta)} \\ \frac{F_V(j-1, \beta)}{F_V(j, \beta)} \end{array} \right. \begin{array}{l} : \\ : \\ : \end{array} \begin{array}{l} A_{i+1} = A_i \cup \{w_j\} \\ \beta_{i+1} = \beta_i - \text{vol}(w_j) \\ A_{i+1} = A_i \\ \beta_{i+1} = \beta_i \end{array}$$

- ▶ Sei  $A \in \mathcal{S}(I')$  beliebig aber fest
- ▶ Sei  $p_i = \frac{z_i}{n_i}$  die Wahrscheinlichkeit, die zum tragen kommt wenn die Entscheidung für bzw. gegen das  $i$ -te Element gefällt wird

$$\begin{aligned}
 z_1 &= F_{I'}(0, \beta') = 1 \\
 n_{|W|} &= F_{I'}(|W|, |W|^2) \\
 z_i &= n_{i-1} \\
 \Rightarrow P[A] &= \frac{z_{|W|}}{n_{|W|}} \cdot \frac{z_{|W|-1}}{n_{|W|-1}} \cdot \dots \cdot \frac{z_1}{n_1} \\
 &= \frac{z_1}{n_{|W|}} = \frac{1}{F_{I'}(|W|, |W|^2)} = \frac{1}{\#(I')}
 \end{aligned}$$

# Laufzeit

- ▶ Um mit Wahrscheinlichkeit  $p \geq \frac{3}{4}$  einen Fehler kleiner als  $\epsilon$  zu machen, sind

$$T_{\xi}(\epsilon) = \left\lceil \frac{4}{\epsilon^2} \cdot (\xi - 1) \right\rceil$$

Stichproben nötig (Estimator Theorem)

- ▶ Mit  $\xi \leq |W| + 1$  und  $|W| = O(n)$  ist die Gesamtlaufzeit

$$O\left(n^3 + \frac{n^2}{\epsilon^2}\right)$$

# Zusammenfassung

- ▶ Generiere **Universum**  $U = S(I') \supseteq S(I)$  durch ganzzahlige Multiplikation mit  $\frac{|W|^2}{B}$
- ▶ **Expansion**  $\xi = \frac{|S(I')|}{|S(I)|} \leq |W| + 1$
- ▶ Berechne  $|S(I')| = F_{\nu}(|W|^2, |W|)$  in  $O(n^3)$
- ▶ Ziehe  $T_{\xi}(\epsilon) = \left\lceil \frac{4|W|}{\epsilon^2} \right\rceil$  Stichproben in Zeit  $O\left(\frac{n^2}{\epsilon^2}\right)$
- ▶ Laufzeit ist in  $O\left(n^3 + \frac{n^2}{\epsilon^2}\right)$

# Literatur

- ▶ Rolf Wanka, „Approximationsalgorithmen“, Seiten 151-182, 2006.
- ▶ Martin Dyer, „Approximate Counting by Dynamic Counting“, 2003.