Transformation des Partitionsproblems in das Rucksackproblem

Sören Kersten

06. Juli 2015

Inhalt

Das Rucksackproblem

- Gegeben ist eine endliche Multimenge von n Zahlen $N = \{a_1, ..., a_n\}$ mit $a_i \in \mathbb{N}_{>0}$ sowie ein Zielgewicht b
- Gesucht ist eine Teilmenge $I \subseteq N$, sodass $\sum_{i \in I} a_i = b$
- Beispiel: $N = \{1, 2, 3, 4\}, b = 3$
  - Lösungen: $I = \{1, 2\}$, $I = \{3\}$
- NP-vollständig
- DNA-Algorithmus zur Lösung wurde in der Vorlesung präsentiert

Das Partitionsproblem

- Gegeben ist eine endliche Multimenge von n Zahlen $M = \{c_1, ..., c_n\}$ mit $c_i \in \mathbb{N}_{>0}$
- Gesucht ist eine Teilmenge $J \subset M$, sodass $\sum_{i \in J} c_i = \sum_{i \in M \setminus J} c_i$
- Beispiel: $M = \{1, 3, 5, 7\}$
  - Lösungen: $J = \{1, 7\}$ oder $J = \{3, 5\}$
- Es gibt immer eine gerade Anzahl möglicher Lösungen
- Transformation zu Rucksackproblem soll gefunden werden

Die Transformation

- Der Algorithmus zur Lösung des Rucksackproblems ist bekannt
  - $I = Knapsack(N, b)$
- Mit dieser Funktion als Basis soll das Partitionsproblem gelöst werden

Pseudocode

Algorithm 1 Pseudocode zur Berechnung einer Partition

Input: A multiset of interger values $M = \{c_1, ..., c_n\}$

Output: A set of integer values $I \subset M$

```
function Partition(M)
    sum = calculateSum(M)
    if sum % 2 == 1 then
        return null
    J = Knapsack(M, sum/2)
    return J
```

Formal

- Gegeben ist $M = \{c_1, ..., c_n\}$
- Setze $N := M$ und $b := \frac{1}{2}\sum_{i=1}^{n} c_i$
- Berechne eine Lösung I für die Instanz N und b des Rucksackproblems
- Setze $J := I$ als Lösung für das Partitionsproblem

Positivbeispiel

- Gegeben $M = \{1, 1, 2, 2, 4, 6\}$
- $N := M, b := \frac{1}{2}(1 + 1 + 2 + 2 + 4 + 6) = 8$
- Lösungen für das Rucksackproblem mit N und b
  - $I = \{2, 6\}$ oder $I = \{2, 2, 4\}$
  - $I = \{1, 1, 2, 4\}$ oder $I = \{1, 1, 6\}$
- Setze $J := I$

Negativbeispiel

- Gegeben $M = \{1, 2, 4, 9\}$
- $N := M, b := \frac{1}{2}(1 + 2 + 4 + 9) = 8$
- Lösungen für das Rucksackproblem mit N und b
  - Keine
- Dementsprechend gibt es keine Lösung für das Partitionsproblem

Beweis der Gültigkeit der Transformation

- Für ein beliebiges $M = \{c_1, ..., c_n\}$ gilt:
  - M ist eine Ja-Instanz des Partitionsproblems $\Leftrightarrow$ Die Transformation ist eine Ja-Instanz des Rucksackproblems
  - M lässt sich in zwei gleich große Mengen partitionieren $\Leftrightarrow$ $N := M$ enthält eine Teilmenge $I := J$ der Größe $b = \frac{1}{2}\sum_{i=1}^{n} c_i$

Beweis der Gültigkeit der Transformation - Hinrichtung

- M ist eine Ja-Instanz des Partitionsproblems
- M lässt sich in zwei gleich große Mengen partitionieren $\Rightarrow \exists\ J \subseteq \{c_1, \ldots, c_n\} \mid \sum_{i \in J} c_i = \sum_{i \in M \setminus J} c_i$
  - $\sum_{i \in J} c_i = \sum_{i \in M \setminus J} c_i$

  $\Rightarrow \sum_{i \in J} c_i = \sum_{i \in M} c_i - \sum_{i \in J} c_i$

  $\Rightarrow 2 \sum_{i \in J} c_i = \sum_{i \in M} c_i$

  $\Rightarrow \sum_{i \in J} c_i = \frac{1}{2} \sum_{i \in M} c_i = b$

  $\Rightarrow$ $I := J$ ist eine Lösung für die Transformation des Rucksackproblems

$\square$

Beweis der Gültigkeit der Transformation - Rückrichtung

- Die Transformation ist eine Ja-Instanz des Rucksackproblems
- N enthält eine Teilmenge I der Größe $b = \frac{1}{2} \sum\limits_{i \in N} a_i$
  - $b = \sum\limits_{i \in I} a_i$ und $b = \frac{1}{2} \sum\limits_{i \in N} a_i$

$\Rightarrow \frac{1}{2} \sum\limits_{i \in N} a_i = \sum\limits_{i \in I} a_i$

$\Rightarrow \sum\limits_{i \in N} a_i = 2 \sum\limits_{i \in I} a_i$

$\Rightarrow \sum\limits_{i \in N} a_i - \sum\limits_{i \in I} a_i = 2 \sum\limits_{i \in I} a_i - \sum\limits_{i \in I} a_i$

$\Rightarrow \sum\limits_{i \in N \setminus I} a_i = \sum\limits_{i \in I} a_i$

$\Rightarrow J := I$ ist eine Lösung für das Partitionsproblem

$\square$

Aufwand der Transformation

- Bei der Transformation muss $b = \frac{1}{2}\sum\limits_{i=1}^{n} c_i$ berechnet werden
  - Der zusätzliche Aufwand ist $(n-1)$ Additionen und eine Division

  $\Rightarrow$ Der zusätzliche Aufwand ist polynomiell beschränkt

Komplexität

- Bei gegebenem M ist die Potenzmenge $\mathcal{P}(M)$ die Menge der möglichen Lösungen für das Partitionsproblem und für das transformierte Rucksackproblem
  - $\Rightarrow$ Die Problemgröße ändert sich bei der Transformation nicht
- Um eine exakte Lösung zu finden, müssen alle Elemente von $\mathcal{P}(M)$ durchlaufen werden
- Im Best-Case ist das erste Element eine Lösung
- Im Worst-Case ist das letzte Element die Lösung
  - $\Rightarrow$ Die Laufzeit bei der Berechnung mit dem Computer beträgt $O(2^n)$
- Der DNA-Algorithmus verwendet keine Überprüfung. Dort sind in jedem Fall n Schritte und 2^n Speicher notwendig.

Vielen Dank für die Aufmerksamkeit