Lösungsstrategien für NP-schwere Probleme der Kombinatorischen Optimierung

— Übungsblatt 6 —

Walter Stieben (4stieben@inf)
Tim Reipschläger (4reipsch@inf)
Louis Kobras (4kobras@inf)
Hauke Stieler (4stieler@inf)

Abgabe am: 30. Mai 2016

Aufgabe 6.1

a)

Zu zeigen ist, dass der angegebene Algorithmus kein 2-Approximationsalgorithmus ist. Zeigen kann man das mit einem Gegenbeispiel:

Sei $A = \{1, 2, 8\}$ und $B = 10$. Der Algorithmus findet nun folgende Mengen:

Index i	Gefundene Menge S
1	$\{1\}$
2	$\{1, 2\}$
3	$\{1, 2\}$

Der Algorithmus nimmt keine Zahlen mehr ab dem Index auf, da dann die Bedingung $\sum\limits_{a_i \in S} a_i \leq B$ nicht mehr gelten würde, da $1 + 2 + 8 = 11 > 10$ gilt.

Das Ergebnis erfüllt somit nicht die Bedingung eines ρ-Approqimationsalgorithmus für Maximierungsprobleme $L^*/L_A \leq \rho$. Stattdessen gilt für das Ergebnis $L_A = 3$, die totale Summe $L^* = B = 10$ und $\rho = 2$ die Gleichung $L^*/L_A = 10/3 = \overline{3,3} \not\leq \rho$.

Damit ist der angegebene Algorithmus kein 2-Approximationsalgorithmus.

□

b)

Algorithm 1 FindTotalSum

```
procedure FindTotalSum(A, B, ρ)
    A ← ConvertToList(A)
    A ← MergeSort(A)
    T := 0
    S := ∅
    for i ∈ {n, ..., 1} do
        if T + a_i ≤ B then
            T ← T + a_i
            S ← S ∪ {a_i}
        end if
```

```
    end for
end procedure
```

Aufgabe 6.2