科学通报
第41卷 第20期 CHINESE SCIENCE BULLETIN 1996年10月

多背包问题近似计算的复杂性*

张立昂 李路阳 黄 雄
(北京大学计算机科学技术系,北京 100871)

关键词 背包问题 组合优化的近似算法 计算复杂性

设 Π 是最大化问题,A 是关于 Π 的近似算法.对 Π 的每一个实例 I,记

$$R_A(I) = \mathrm{OPT}(I)/A(I),$$

其中 OPT(I)是 I 的最优值,$A(I)$是算法 A 求得的近似解的值.记

$$R_A = \inf\{r \geqslant 1: \text{对所有的实例 } I, R_A(I) \leqslant r\},$$

R_A 称作 A 的性能比.如果 $R_A < +\infty$,则称 A 具有常数比.关于 Π 的多项式时间近似方案(PTAS) A 是一个近似算法,它以 Π 的实例 I 和 $\varepsilon > 0$ 作为输入.对于每一个固定的 $\varepsilon > 0$,把它记作 A_ε.A_ε 是 Π 的多项式时间近似算法并且 $R_{A_\varepsilon} \leqslant 1+\varepsilon$.如果 A 是一个 PTAS 并且存在二元多项式 p 使得 A 的时间复杂度不超过 $p\left(|I|, \frac{1}{\varepsilon}\right)$,则称 A 是完全多项式时间近似方案(FPTAS),其中 $|I|$ 表示实例 I 的规模.

多背包问题定义如下:任给 n 件物品和 k 个背包,物品 j 的重量为 W_j,价值为 $V_j(1 \leqslant j \leqslant n)$,背包 i 的重量限制为 $B_i(1 \leqslant i \leqslant k)$,所有 $W_j, V_j, B_i > 0$.要求在满足重量限制的条件下,使装入背包的物品的总价值最大.即求 k 个不相交的子集 $J_i \subseteq \{1, 2, \cdots, n\}(1 \leqslant i \leqslant k)$使得

$$\sum_{i=1}^{k} \sum_{j \in J_i} V_j \text{ 最大}$$

且满足条件

$$\sum_{j \in J_i} W_j \leqslant B_i \quad 1 \leqslant i \leqslant k.$$

当 k 为固定的正整数时,把这个问题称作 k-背包问题.1-背包问题就是普通的 0/1 背包问题,它有 FPTAS[1].对于每一个固定的 $k \geqslant 2$,k-背包问题有 PTAS 和伪多项式时间算法,但不存在 FPTAS(除非 $P = NP$)[2,3].

本文讨论多背包问题的近似计算.根据下述定理,多背包问题不存在 PTAS.

定理 1 假设 $P \neq NP$,则多背包问题不存在多项式时间的近似算法 A 使得 $R_A \leqslant \frac{6}{5}$.即使所有背包的重量限制都相同,每一件物品的重量等于其价值,即 $B_i = B(1 \leqslant i \leqslant k)$且 $W_j = V_j(1 \leqslant j \leqslant n)$,这个结论仍然成立.

1995-08-29 收稿,1996-01-03 收修改稿
* 国家"八六三"计划资助项目

证 设 A 是问题的多项式时间近似算法,其性能比 $R_A=c$. 利用 A 构造装箱问题的近似算法 A' 如下:设装箱问题的实例 I 为 n 件物品的体积 $S_j(1\leqslant j\leqslant n)$ 和每只箱子的容积 B,这里 $S_j\leqslant B(1\leqslant j\leqslant n)$. 令 $S=\sum_{j=1}^{n}S_j$. 对 $k=1,2,\cdots,n$,构造 k-背包问题的实例 I_k: $W_j=V_j=S_j(1\leqslant j\leqslant n)$ 和 $B_j=B(1\leqslant i\leqslant k)$,把 A 运用于 I_k,直至找到 k_1 使得 $A(I_{k_1-1})<\frac{1}{c}S\leqslant A(I_{k_1})$,这里 $A(I_0)=0$. 然后用 FFD 法[1]把剩余物品全部装入箱子中,设又使用了 k_2 只箱子. A' 总共使用的箱子数为 $k=k_1+k_2$. 显然,A' 是多项式时间的.

设 $\mathrm{OPT}(I)=k^*$,则 $\mathrm{OPT}(I_{k^*})=S$, $A(I_{k^*})\geqslant\frac{1}{c}S$. 因此,$k^*\geqslant k_1$. 记 $k=k^*+k'$,则 $k'\leqslant k_2$. FFD 法使用的 k_2 只箱子里所装物品的总体积 $\leqslant\left(1-\frac{1}{c}\right)S$. 设装这些物品所需的最少箱子数为 l^*. 不妨设 $k_2>l^*$,则有[4]

(1) $k_2\leqslant\frac{4}{3}l^*+\frac{1}{3}$,从而 $l^*\geqslant\frac{3}{4}\left(k_2-\frac{1}{3}\right)$;

(2)装入号码大于 l^* 的箱子中的物品的体积都不超过 $\frac{B}{3}$,从而前 l^* 只箱子中每只箱子所装物品的体积大于 $\frac{2}{3}B$.

于是

$$\left(1-\frac{1}{c}\right)S>\frac{2}{3}Bl^*\geqslant\frac{1}{2}B\left(k_2-\frac{1}{3}\right),$$

$$k_2<\frac{2}{B}\left(1-\frac{1}{c}\right)S+\frac{1}{3}\leqslant2\left(1-\frac{1}{c}\right)k^*+\frac{1}{3},$$

$$\frac{k}{k^*}\leqslant1+\frac{k_2}{k^*}<1+2\left(1-\frac{1}{c}\right)+\frac{1}{3k^*}.$$

注意到当 $k^*=1$ 时 $k=k^*$,所以

$$\frac{k}{k^*}<\frac{7}{6}+2\left(1-\frac{1}{c}\right).$$

如果 $c\leqslant\frac{6}{5}$,则 $\frac{k}{k^*}<\frac{3}{2}$,即 A' 是装箱问题的多项式时间近似算法,其性能比 $R_{A'}<\frac{3}{2}$. 这是不可能的,除非 $P=NP$[1].

下面给出多背包问题的一个多项式时间近似算法,其性能比为 2. 算法的主要思想是按"价值密度"递减的顺序检查每一件物品. 对于每一个背包 i,先把能装进它的物品放在一起. 当这些物品的重量之和不小于 $\frac{1}{2}B_i$ 时把它们装入背包 i.

算法 L

1. 重新排列物品使得 $V_1/W_1\geqslant V_2/W_2\geqslant\cdots\geqslant V_n/W_n$. 重新排列背包使得 $B_1\leqslant B_2\leqslant\cdots\leqslant B_k$.
2. 令 $M=\{1,2,\cdots,k\}$, $J'_i=\varnothing$, $W'_i=0$ $(1\leqslant i\leqslant k)$.
3. 对于 $j=1,2,\cdots,n$,找到 M 中使 $W_j\leqslant B_i$ 的最小的 i(若无这样的 i,则舍去物品 j).

a. 如果 $W_j + W'_j < \frac{1}{2}B_i$，则令 $W'_i = W'_i + W_j$，$J'_i = J'_i \cup \{j\}$.

b. 如果$\frac{1}{2}B_i \leqslant W_j + W'_i \leqslant B_i$，则令 $J_i = J'_i \cup \{j\}$，$M = M - \{i\}$.

c. 如果 $W_j + W'_i > B_i$ 并且存在$t \in M$ 使$t > i$，设 t 是最小的一个，则令 $J_i = \{j\}$，$M = M - \{i\}$.

c.1 如果 $W'_i + W'_t < \frac{1}{2}B_t$，则令 $W'_t = W'_i + W'_t$，$J'_t = J'_i \cup J'_t$.

c.2 否则$\left(\frac{1}{2}B_t \leqslant W'_i + W'_t \leqslant B_t\right)$令$J_t = J'_i \cup J'_t$，$M = M - \{t\}$.

d. 否则如果 $V_j \leqslant \sum_{l \in J'_i} V_l$ 则令 $J_i = J'_i$，否则令 $J_i = \{j\}$，令 $M = M - \{i\}$（记 $r_i = \min\left\{V_j, \sum_{l \in J'_i} V_l\right\}$）.

4. 如果 $M \neq \varnothing$ 则对所有的 $i \in M$，令 $J_i = J'_i$.

设问题的最优解是$\{J_i^*\}$，算法 L 给出的解是$\{J_i\}$，那么不难看到：

1. 如果在步骤 4，$M \neq \varnothing$，令 $t = \max\{i : i \in M\}$，则所有 $W_j \leqslant B_t$ 的物品都被装进背包. 因此，存在$\{J_i^*\}$使得$J_i^* = J_i(1 \leqslant i \leqslant t)$. 此时，令 $r_i = 0(1 \leqslant i \leqslant t)$.

2. 如果 J_i 在步骤 3 的情况 b 或 c.2 得到，则 $\sum_{j \in J_i} W_j \geqslant \frac{B_i}{2}$. 设 l 是算法 L 下一个检查的物品的下标，令 $r_i = \frac{V_l}{W_l} \cdot \frac{B_i}{2}$，则有 $r_i \leqslant \sum_{j \in J_i} V_j$.

3. 如果 J_i 在步骤 3 的情况 d 得到，则也有 $r_i \leqslant \sum_{j \in J_i} V_j$.

从而

$$\mathrm{OPT}(I) = \sum_{i=1}^{k} \sum_{j \in J_i^*} V_j \leqslant \sum_{i=1}^{k} \left(\sum_{j \in J_i} V_j + r_i \right) \leqslant 2 \sum_{i=1}^{k} \sum_{j \in J_i} V_j = 2L(I).$$

于是，得到

定理 2 对于多背包问题的所有实例 I，有

$$\mathrm{OPT}(I) \leqslant 2L(I).$$

考虑实例 I：$W_j = V_j = 1/2 + \varepsilon(1 \leqslant j \leqslant k)$，$W_j = V_j = 1/2(k+1 \leqslant j \leqslant 3k)$，$B_i = 1(1 \leqslant i \leqslant k)$，其中 $0 < \varepsilon \leqslant 1/2$. 显然，$\mathrm{OPT}(I) = k$，而 $L(I) = (1/2 + \varepsilon)k$. ε 可以任意小，因此，算法 L 的性能比 $R_L = 2$.

参 考 文 献

1 Garey M R, Johnson D S. Computers and Intractability. San Francisco: W H Freeman and Company, 1979

2 张立昂，耿素云. 多背包问题的计算. 北京大学学报(自然科学版)，1987，1：65～76

3 Zhang Li'ang. The complexity of approximation for k-knapsack. 见：苏运霖主编. 国际离散数学与算法研讨会文集. 广州：暨南大学出版社，1994，177～180

4 Baase S. Computer Algorithms, Reading: Addison-Wesley Publishing Co, 1978. 268～270