

组合优化理论

第5章 背包问题

主讲教师：陈安龙

第5章 背包问题

§ 1 背包问题的描述

§ 2 分支定界法解背包问题

§ 3 动态规划法解背包问题

§ 4 近似算法解背包问题

§ 5 背包问题的一些相关问题

背包问题 (*Knapsack Problem*) 是一个有着广泛应用的组合优化问题，它不仅在投资决策、装载、库存等方面有应用，而且常以子问题形式出现在大规模优化问题中，它的理论与算法具有一定的代表性。

§ 1 背包问题的描述

背包问题的一般描述为：设有物品集 $U = \{u_1, u_2, \dots, u_n\}$ 是一个准备放入容量为 $C \in \mathbb{Z}^+$ 的背包中的 n 项物品的集合。

如何选择 U 中的一些物品装入背包，使这些物品的总重量不超过 C ，且使总价值达到最大？

§ 1 背包问题的描述

背包问题的数学模型:

$$\max \quad z = \sum_{j=1}^n p_j x_j$$

(KP)

$$s.t. \quad \sum_{j=1}^n w_j x_j \leq C$$

$$x_j = 0 \text{ or } 1 \quad j \in N = \{1, 2, \dots, n\}$$

因为决策变量 $x_j = 0 \text{ or } 1$, 所以也称 0-1 背包问题.

一般背包问题: $x_j \in \mathbb{Z}^+ \cup \{0\} \quad j \in N = \{1, 2, \dots, n\}$

(*General Knapsack Problem*)

$w_j \in \mathbb{Z}$ —— 重量

$p_j \in \mathbb{Z}$ —— 价值

$C \in \mathbb{Z}^+$ —— 容量

为讨论方便，总可假定（相当于标准化）：

1、 $w_j > 0, p_j > 0 \quad j = 1 \sim n;$

2、 $w_j \leq C \quad j = 1 \sim n;$

3、 $\sum_{j=1}^n w_j > C$

4、 $\frac{p_1}{w_1} \geq \frac{p_2}{w_2} \geq \dots \geq \frac{p_n}{w_n}$ 即按价值密度从大到小排列.

§ 1 背包问题的描述

对 4 只需 $O(n \ln n)$ 次运算即可;

对 3 若 $\sum_{j=1}^n w_j \leq C$, 最优解为 $x_1 = x_2 = \cdots = x_n = 1$;

对 2 若 $w_j > C$, 则最优解中 $x_j = 0$, \therefore 事先可去掉;

对 1 分三种情况讨论:

(1) 若 $p_j \leq 0$ 且 $w_j \geq 0$ 令 $N^0 = \{j \in N \mid p_j \leq 0, w_j \geq 0\}$

此时, 最优解中 $x_j = 0$ 所以, 该物品事先可去掉;

(2) 若 $p_j \geq 0$ 且 $w_j < 0$ 令 $N^1 = \{j \in N \mid p_j \geq 0, w_j < 0\}$

此时, 最优解中 $x_j = 1$ 所以, 该物品事先可去掉;

(3) 若 $p_j < 0$ 且 $w_j < 0$ 令 $N^- = \{j \in N \mid p_j < 0, w_j < 0\}$

只需在模型中, 令 $x_j = 1 - y_j$, 则系数即为大于零了.

综上 (1) (2) (3) 的假定, 可作如下处理, 使之满足:

对 $\forall j \in N^-$ 令 $x_j = 1 - y_j$ $\bar{p}_j = -p_j$ $\bar{w}_j = -w_j$

对 $\forall j \in N^+ = N \setminus (N^0 \cup N^-)$ 令

$x_j = y_j$ $\bar{p}_j = p_j$ $\bar{w}_j = w_j$

则原问题化为:

$$\begin{aligned} \max \quad & z = \sum_{j \in N^- \cup N^+} \bar{p}_j y_j + \sum_{j \in N^- \cup N^1} p_j \\ \text{s.t.} \quad & \sum_{j \in N^- \cup N^+} \bar{w}_j y_j \leq C - \sum_{j \in N^- \cup N^1} w_j \\ & y_j = 0 \text{ or } 1, \quad j \in N^- \cup N^+ \end{aligned}$$



§ 1 背包问题的描述

如果在 (KP) 中, 令 $x_j = 1 - y_j$

$$\max \quad z = \sum_{j=1}^n p_j - \sum_{j=1}^n p_j y_j$$

$$s.t. \quad \sum_{j=1}^n w_j - \sum_{j=1}^n w_j y_j \leq C$$

$$y_j = 0 \text{ or } 1 \quad j \in N = \{1, 2, \dots, n\}$$

$$\text{令} \quad q = \sum_{j=1}^n w_j - C$$

至少有 q 的物品不能放进去

该问题的实际意义是求不
放在包中的物品的价值和最小

$$\begin{aligned} (KP) \quad & \max \quad z = \sum_{j=1}^n p_j x_j \\ & s.t. \quad \sum_{j=1}^n w_j x_j \leq C \\ & \quad x_j = 0 \text{ or } 1 \\ & \quad j \in N = \{1, 2, \dots, n\} \end{aligned}$$

$$\min \quad u = \sum_{j=1}^n p_j y_j$$

$$s.t. \quad \sum_{j=1}^n w_j y_j \geq q$$

$$y_j = 0 \text{ or } 1 \quad j \in N = \{1, 2, \dots, n\}$$

§ 2 背包问题的分支定界法

分支定界法 (*Branch and Bound Method*) 的基本思想在课程中已介绍，它的重要在于它提出了一类新的思路（隐枚举法），使得许多原来不好解决的问题有了解决的可能性。（具有普适性）

确定问题（子问题）的最优值的上（下）界

通常是通过求解松弛问题，用松弛问题的解作为界

Note：松弛问题选择的原则

- 1、松弛问题要与原问题的最优值尽量接近；
- 2、松弛问题要尽量容易解。

这两个原则不易统一，所以可选择不同的松弛问题

考虑 KP 的松弛问题： $x_j \in \{0,1\} \Rightarrow x_j \in [0,1]$

$$\max \quad z = \sum_{j=1}^n p_j x_j$$

$$C(KP) \quad s.t. \quad \sum_{j=1}^n w_j x_j \leq C$$

$$0 \leq x_j \leq 1 \quad j \in N = \{1, 2, \dots, n\}$$

思路：将物品按价值密度从大到小的顺序放入包内，.....

记 $s = \min \left\{ j \mid \sum_{i=1}^j w_i > C \right\}$ —— 关键项

第一个放不下的
物品的序号

§ 2 背包问题的分支定界法

Theorem 6.1 $C(KP)$ 最优解为

$$\bar{x}_j = 1 \quad j = 1 \sim s-1 \quad \bar{x}_j = 0 \quad j = s+1 \sim n$$

$$\bar{x}_s = \frac{\bar{C}}{w_s} \quad \text{其中} \quad \bar{C} = C - \sum_{j=1}^{s-1} w_j$$

最优解值为: $z_{opt}(C(KP)) = \sum_{j=1}^{s-1} p_j + \bar{C} \frac{p_s}{w_s}$

显然, $z_{opt}(KP) \leq z_{opt}(C(KP))$, 由于 p_j 的整数性,

得到 $z(KP)$ 的一个上界:

$$U_1 = \left\lfloor z_{opt}(C(KP)) \right\rfloor = \sum_{j=1}^{s-1} p_j + \left\lfloor \bar{C} \frac{p_s}{w_s} \right\rfloor$$

$\lfloor \alpha \rfloor$ 表示不超过 α 的最大整数.

证明： 显然 $C(KP)$ 的最优解必满足 $\sum_{j=1}^n w_j x_j = C$

设 x^* 是其最优解，要证 $x^* = \bar{x}$ 若存在 $k < s$ 使 $x_k^* < 1$

则至少存在 $q \geq s$ 使 $x_q^* > \bar{x}_q$. 取充分小的 $\varepsilon > 0$
(满足: 1、 $x_k^* + \varepsilon \leq 1$ 2、 $x_q^* - \varepsilon \frac{w_k}{w_q} \geq 0$)

$\lambda = (1, \dots, 1, \frac{C}{w_s}, 0, \dots, 0)$
 $\lambda_1 \lambda_2 \dots \lambda_{s-1} \lambda_s \lambda_{s+1} \dots \lambda_n$
 $\lambda^* = (1, \dots, 1, \lambda_s^*, \dots, \lambda_n)$
 $\lambda_s^* < 1$ $\lambda_s > \bar{\lambda}_s$ $\lambda_j^* > \bar{\lambda}_j$ $\lambda_j < \bar{\lambda}_j$

$x_k^* + \varepsilon$ 则重量增加 εw_k , $x_q^* - \varepsilon \frac{w_k}{w_q}$ 则重量减少 εw_k

将 x_k^* 增加 ε , x_q^* 减少 $\varepsilon \frac{w_k}{w_q}$, 此时, 仍是一个可行解,

且目标函数值增加 $\varepsilon(p_k - p_q \frac{w_k}{w_q}) > 0$, 矛盾. $\frac{p_k}{w_k} > \frac{p_q}{w_q} \Rightarrow p_k - p_q \frac{w_k}{w_q} > 0$

\therefore 对 $\forall k < s, x_k^* = \bar{x}_k = 1$ 同理可证 对 $\forall k > s, x_k^* = 0$

又由极大性知: $x_s^* = \frac{C}{w_s} = \bar{x}_s$ 因此, \bar{x} 是最优解.

易证 $z_{opt}(C(KP))$ 为最优解值

§ 2 背包问题的分支定界法

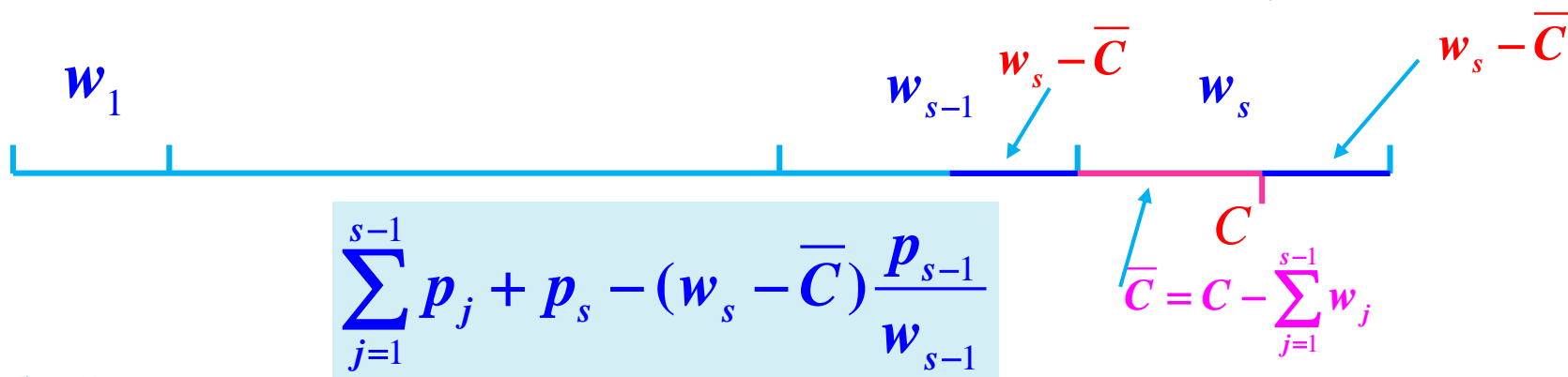
Theorem 6.2 设 $U^0 = \sum_{j=1}^{s-1} p_j + \left\lfloor \bar{C} \frac{p_{s+1}}{w_{s+1}} \right\rfloor$

$$U^1 = \sum_{j=1}^{s-1} p_j + \left\lfloor p_s - (w_s - \bar{C}) \frac{p_{s-1}}{w_{s-1}} \right\rfloor \quad U_1 = \left\lfloor z_{opt}(C(KP)) \right\rfloor = \sum_{j=1}^{s-1} p_j + \left\lfloor \bar{C} \frac{p_s}{w_s} \right\rfloor$$

其中 s 与 \bar{C} 定义同前。 则

1、 $z(KP)$ 的一个上界为 $U_2 = \max(U^0, U^1)$;

2、对背包问题任一实例, $U_2 \leq U_1$ 作为上界 U_2 比 U_1 更好



Proof: 1、因为 KP 中 x_s 不能取分数，所以， KP 的最优解一定是 $\bar{x}_s = 0$ or $\bar{x}_s = 1$ 情形之一。

当 $\bar{x}_s = 0$ ，由 Th 6.1 可知， U^0 是此情形的上界；

当 $\bar{x}_s = 1$ ，这时，若 $\bar{x}_1 = \bar{x}_2 = \cdots = \bar{x}_{s-1} = 1$

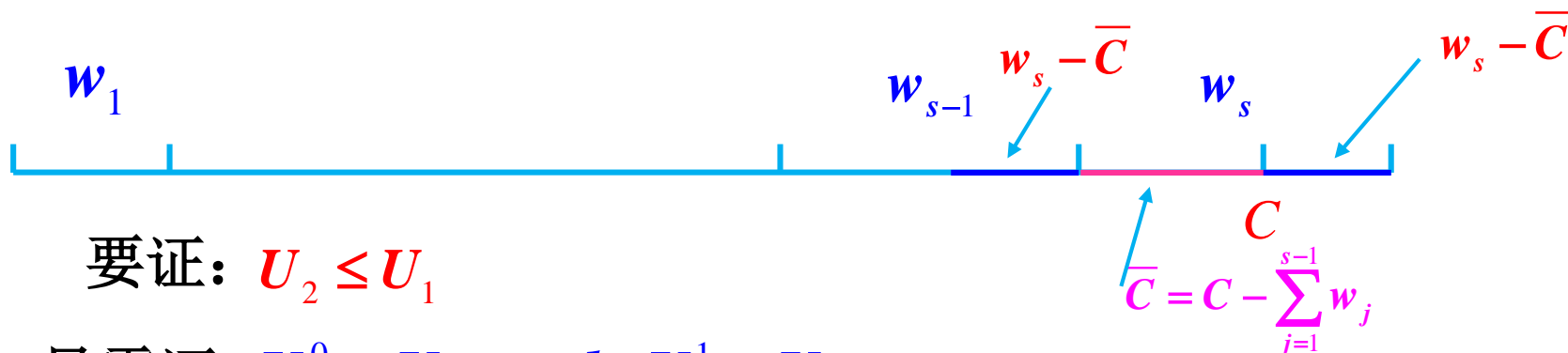
重量超出 $w_s - \bar{C}$ ，而此时价值密度值最小的是 $\frac{p_{s-1}}{w_{s-1}}$ 。

$\therefore U^1$ 是此情形的上界。

从而 $U_2 = \max(U^0, U^1)$ 是 $z(KP)$ 的上界。



§ 2 背包问题的分支定界法



要证: $U_2 \leq U_1$

只需证: $U^0 \leq U_1$ and $U^1 \leq U_1$

$$U_1 = \sum_{j=1}^{s-1} p_j + \left\lfloor \bar{C} \frac{p_s}{w_s} \right\rfloor$$

$$U^0 = \sum_{j=1}^{s-1} p_j + \left\lfloor \bar{C} \frac{p_{s+1}}{w_{s+1}} \right\rfloor$$

$$\therefore \bar{C} \frac{p_{s+1}}{w_{s+1}} \leq \bar{C} \frac{p_s}{w_s}$$

§ 2 背包问题的分支定界法

2、 $U^0 \leq U_1$ 是显然的；

$$C(KP) \text{ 的最优解值 } z_{opt}(C(KP)) = \sum_{j=1}^{s-1} p_j + \bar{C} \frac{p_s}{w_s}$$

$$= \sum_{j=1}^s p_j - (w_s - \bar{C}) \frac{p_s}{w_s} \quad (*)$$

$C(KP)$ 当 $\bar{x}_s = 1$ 的最优解值

$$\sum_{j=1}^s p_j - (w_s - \bar{C}) \frac{p_{s-1}}{w_{s-1}} \quad (**)$$

$$\because w_s > \bar{C}, \frac{p_{s-1}}{w_{s-1}} \geq \frac{p_s}{w_s} \therefore \sum_{j=1}^s p_j - (w_s - \bar{C}) \frac{p_s}{w_s} \geq \sum_{j=1}^s p_j - (w_s - \bar{C}) \frac{p_{s-1}}{w_{s-1}}$$

$$\therefore U^1 \leq U_1 \quad \text{从而} \quad U_2 \leq U_1.$$

一般给出的上界越小，计算量越大，但越容易被剪枝。

用分支定界法求如下 KP :

$$n = 7 \quad (p_j) = (70, 20, 39, 37, 7, 5, 10)$$

$$(w_j) = (31, 10, 20, 19, 4, 3, 6) \quad C = 50$$

$$u = 70 + 20 + \left\lfloor \frac{39 \times 9}{20} \right\rfloor = 107$$

Solution : 可验证

$$\frac{70}{31} \geq \frac{20}{10} \geq \frac{39}{20} \geq \dots \geq \frac{10}{6}$$

$$x_1 = (1, 1, 0, \frac{9}{19}, 0, 0, 0)$$

$$u = 107$$

$$\bar{x}_3 = (1, 1, 0, 0, 1, 1, \frac{1}{3})$$

$$u = 105 *$$

$$\therefore x_{opt} = (1, 0, 0, 1, 0, 0, 0) \quad z_{\max} = 107$$

$$\tilde{K}_0 \quad \bar{x}_0 = (1, 1, \frac{9}{20}, 0, 0, 0, 0)$$

$$u = 107$$

$$x_3 = 0$$

$$x_3 = 1$$

$$\tilde{K}_2$$

$$\bar{x}_2 = (30/31, 0, 1, 0, 0, 0, 0)$$

$$u = 106 *$$

$$\tilde{K}_4$$

$$\bar{x}_4 = (1, 0, 0, 1, 0, 0, 0)$$

$$u = 107$$

*

动态规划法求解背包问题

有一个徒步旅行者，其可携带物品重量的限度为 a 公斤，设有 n 种物品可供他选择装入包中。已知每种物品的重量及使用价值（作用），问此人应如何选择携带的物品（各几件），使所起作用（使用价值）最大？

物品	1	2	\dots	j	\dots	n
重量（公斤/件）	a_1	a_2	\dots	a_j	\dots	a_n
每件使用价值	c_1	c_2	\dots	c_j	\dots	c_n

这就是背包问题。类似的还有工厂里的下料问题、运输中的货物装载问题、人造卫星内的物品装载问题等。

设 x_j 为第 j 种物品的装件数（非负整数）则问题的数学模型如下：

$$\max Z = \sum_{j=1}^n c_j x_j$$

$$\begin{cases} \sum_{j=1}^n a_j x_j \leq a \\ x_j \geq 0 \text{ 且为整数 } (j = 1, 2, \dots, n) \end{cases}$$

用动态规划方法求解，令

$f_k(y)$ = 在选择装前 k 种物品后，当前还可装重量不超过 y 公斤的物品的最大使用价值，其中 $y \geq 0, k = 1, 2, \dots, n$ 。

所以问题的本质就是求 $f_n(a)$

其递推关系式为：

$$f_k(y) = \max_{0 \leq x_k \leq \frac{y}{a_k}} \{ \mathbf{c}_k x_k + f_{k-1}(y - a_k x_k) \}$$

其中 $2 \leq k \leq n$

当 $k=1$ 时，有：

$$f_1(y) = c_1 \left(\frac{y}{a_1} \right), \quad \left[x_1 = \left(\frac{y}{a_1} \right) \right]$$

其中 $\left(\frac{y}{a_1} \right)$ 表示不超过 $\frac{y}{a_1}$ 的最大整数

例：求下面背包问题的最优解($a=5$)

$$\max Z = 8x_1 + 5x_2 + 12x_3$$

$$\begin{cases} 3x_1 + 2x_2 + 5x_3 \leq 5 \\ x_1, x_2, x_3 \geq 0 \text{ 且为整数} \end{cases}$$

物品	1	2	3
重量 (公斤)	3	2	5
使用价值	8	5	12

解： $a=5$ ，问题是求 $f_3(5)$

$$f_3(5) = \max_{\substack{0 \leq x_3 \leq \frac{5}{a_3} \\ x_3 \text{ 整数}}} \{ 12x_3 + f_2(5 - 5x_3) \}$$

$$f_3(5) = \max_{\substack{0 \leq x_3 \leq \frac{5}{a_3} \\ x_3 \text{ 整数}}} \{ 12x_3 + f_2(5 - 5x_3) \}$$

$$= \max_{\substack{0 \leq x_3 \leq \frac{5}{5} \\ x_3 \text{ 整数}}} \{ 12x_3 + f_2(5 - 5x_3) \}$$

$$= \max_{x_3=0,1} \{ 12x_3 + f_2(5 - 5x_3) \}$$

$$= \max \left\{ \begin{array}{ll} 0 + f_2(5), & (x_3=0) \\ 12 + f_2(0) & (x_3=1) \end{array} \right\}$$

$$f_2(5) = \max_{\substack{0 \leq x_2 \leq \frac{5}{a_2} \\ x_2 \text{ 整数}}} \{ 5x_2 + f_1(5 - 2x_2) \}$$

$$= \max_{\substack{0 \leq x_2 \leq \frac{5}{2} \\ x_2 \text{ 整数}}} \{ 5x_2 + f_1(5 - 2x_2) \}$$

$$= \max_{x_2=0,1,2} \{ 5x_2 + f_1(5 - 2x_2) \}$$

$$= \max \left\{ \begin{array}{ccc} 0 + f_1(5), & 5 + f_1(3), & 10 + f_1(1) \\ (x_2=0) & (x_2=1) & (x_2=2) \end{array} \right\}$$

$$\begin{aligned} f_2(\mathbf{0}) &= \max_{\substack{0 \leq x_2 \leq \frac{0}{a_2} \\ x_2 \text{ 整数}}} \{ 5x_2 + f_1(\mathbf{0} - 2x_2) \} \\ &= \max_{\substack{0 \leq x_2 \leq \frac{0}{2} \\ x_2 \text{ 整数}}} \{ 5x_2 + f_1(\mathbf{0} - 2x_2) \} \\ &= \max_{x_2=0} \{ 5x_2 + f_1(\mathbf{0} - 2x_2) \} \\ &= \max_{(x_2=0)} \left\{ \mathbf{0} + f_1(\mathbf{0}) \right\} = f_1(\mathbf{0}) \end{aligned}$$

$$f_1(5) = c_1 x_1 = 8 \times \left\lfloor \frac{5}{3} \right\rfloor = 8 \quad (x_1 = 1)$$

$$f_1(3) = c_1 x_1 = 8 \times \left\lfloor \frac{3}{3} \right\rfloor = 8 \quad (x_1 = 1)$$

$$f_1(1) = c_1 x_1 = 8 \times \left\lfloor \frac{1}{3} \right\rfloor = 0 \quad (x_1 = 0)$$

$$f_1(0) = c_1 x_1 = 8 \times \left\lfloor \frac{0}{3} \right\rfloor = 0 \quad (x_1 = 0)$$

$$\begin{aligned} \therefore f_2(5) &= \max \left\{ \underset{(x_2=0)}{0 + f_1(5)}, \quad \underset{(x_2=1)}{5 + f_1(3)}, \quad \underset{(x_2=2)}{10 + f_1(1)} \right\} \\ &= \max \{ 8, \quad 5 + 8, \quad 10 \} = 13 \quad (x_1 = 1, x_2 = 1) \end{aligned}$$

$$f_2(0) = \max_{(x_2=0)} \left\{ 0 + f_1(0) \right\} = f_1(0) = 0 \quad (x_1 = 0, x_2 = 0)$$

$$\begin{aligned} \therefore f_3(5) &= \max_{(x_3=0)} \left\{ 0 + f_2(5), \quad 12 + f_2(0) \right\} \\ &= \max \left\{ 0 + 13, \quad 12 + 0 \right\} \\ &= 13 \quad (x_1 = 1, x_2 = 1, x_3 = 0) \end{aligned}$$

所以，最优解为 $X = (1, 1, 0)$ ，最优值为 $Z = 13$ 。

§ 3 背包问题的近似算法

有改进的吗?

通过前面介绍的 $C(KP)$, 自然想到如下贪婪算法

(Greedy Algorithm): $x_1 = \cdots = x_{s-1} = 1, x_s = \cdots = x_n = 0$

其目标函数值为 $\sum_{j=1}^{s-1} p_j$.

GA_0 是近似
算法吗?

GA_0 step 1 $x_0 = w_0 = 0 \quad k = 1$

step 2 若 $\sum_{j=0}^{k-1} w_j x_j + w_k \leq C$, 则 $x_k = 1$,

否则 $x_k = 0$;

step 3 若 $k = n$, 则结束;

否则 $k := k + 1$, 转 step 2 .

构造例子 I : $n = 2$, $p_1 = 1 + \varepsilon$, $w_1 = 1$, $p_2 = k$, $w_2 = k$, $C = k$.

ε 为充分小的正数.

按上述算法, 得: $x_1 = 1$, $x_2 = 0$, $z_{GA_0}(I) = 1 + \varepsilon$;

而显然最优解为: $x_1 = 0$, $x_2 = 1$, $z_{opt}(I) = k$.

$$\frac{z_{GA_0}(I)}{z_{opt}(I)} = \frac{1 + \varepsilon}{k} \rightarrow 0 \quad (k \rightarrow +\infty)$$

说明 GA_0 的绝对性能比不会大于任意给定的正数,
所以, 它不能作为近似解. 但稍加改进, 就可得到一个绝对性能比为常数的较好的近似算法.

§ 3 背包问题的近似算法

GA_1 *step 1* 求解 $C(KP)$, 得关键项记为 s ;

step 2 取 $\max \left\{ \sum_{j=1}^{s-1} p_j, p_s \right\}$ 作为近似解值.

即若 $\sum_{j=1}^{s-1} p_j > p_s$, 则 $x_1 = \cdots = x_{s-1} = 1, x_s = \cdots = x_n = 0$ (即前 $s-1$ 件物品的总价值还不如第 s 件物品的价值大, 则即使第 s 件物品容量小, 也可不装入)

否则 $x_1 = \cdots = x_{s-1} = x_{s+1} = \cdots = x_n = 0, x_s = 1$

Example 2 用 GA_1 求如下 KP :

$$n = 4 \quad (p_j) = (3, 7, 17, 20), (w_j) = (1, 3, 8, 10) \quad C = 11$$

Solution: 易验证 $\frac{3}{1} \geq \frac{7}{3} \geq \frac{17}{8} \geq \frac{20}{10}$ 有改进的吗?

显然, 物品3 为关键项 (即 $s = 3$). $\because p_1 + p_2 = 3 + 7 = 10,$

而 $p_3 = 17$ 近似解为 $x_1 = x_2 = x_4 = 0, x_3 = 1; z_{GA_1} = 17$

GA_2 *step 1* 求解 $C(KP)$, 得关键项记为 s ;

step 2 令 $p_i = \max_j \{p_j\}$ (即先找集合中最大的价值, 然后用前述算法来比较求最优)

若 $\sum_{j=1}^{s-1} p_j > p_i$ 则 $x_1 = \cdots = x_{s-1} = 1, x_s = \cdots = x_n = 0$

否则 $x_1 = \cdots = x_{i-1} = x_{i+1} = \cdots = x_n = 0, x_i = 1$

Example 3 用 GA_2 求如下 KP :

$$n = 4 \quad (p_j) = (3, 7, 17, 20), (w_j) = (1, 3, 8, 10) \quad C = 11$$

Solution: $\because p_1 + p_2 = 3 + 7 = 10,$

而 $\max_j \{p_j\} = p_4 = 20$ 近似解为

$$x_1 = x_2 = x_3 = 0, x_4 = 1; \quad z_{GA_2} = 20$$

§ 3 背包问题的近似算法

Theorem 6.3 $R_{GA_1} = \frac{1}{2}$ $R_{GA_1} = \sup_I \left\{ r \leq 1 \mid \frac{z_{GA_1}(I)}{z_{opt}(I)} \geq r \right\}$

Proof: 由 s 的定义知: 对于任意实例 I

$$z_{opt}(I) \leq \sum_{j=1}^{s-1} p_j + p_s \quad \text{又} \quad z_{GA_1}(I) = \max \left\{ \sum_{j=1}^{s-1} p_j, p_s \right\} \geq \frac{1}{2} \left(\sum_{j=1}^{s-1} p_j + p_s \right)$$

(因为 $\sum_{j=1}^{s-1} p_j$ 和 p_s 都大于 0)

因此, $\frac{z_{GA_1}(I)}{z_{opt}(I)} \geq \frac{1}{2}$ 从而 $R_{GA_1} \geq \frac{1}{2}$

取实例 I : $n=3, p_1=1+\varepsilon, w_1=1, \varepsilon$ 为充分小的正数.

$$p_2 = p_3 = w_2 = w_3 = k, C = 2k$$

$$z_{GA_1}(I) = 1 + \varepsilon + k, x_1 = x_2 = 1, x_3 = 0$$

$$z_{opt}(I) = 2k, x_1 = 0, x_2 = x_3 = 1$$

则 $R_{GA_1} \leq \frac{1}{2}$ 从而 $R_{GA_1} = \frac{1}{2}$

$$\frac{z_{GA_1}(I)}{z_{opt}(I)} = \frac{1 + \varepsilon + k}{2k} \rightarrow \frac{1}{2} \quad (k \rightarrow \infty)$$

$$= \frac{1 + \varepsilon}{2k} + \frac{1}{2}$$

§ 3 背包问题的近似算法

Theorem 6.4 $R_{GA_2} = \frac{1}{2}$. 证明与 *Th 6.3* 的类似.

对 *Ex . 3*

$$n = 4 \quad (p_j) = (3, 7, 17, 20), (w_j) = (1, 3, 8, 10) \quad C = 11$$

考虑对 GA_2 进行修改, 进一步可取

$$x_1 = x_4 = 1, \quad x_2 = x_3 = 0 \quad \text{则 } z = 23$$

这在实际计算中是会有好处的. 但绝对性能比不会改进.

§ 3 背包问题的近似算法

已知 GA_0 对解 0-1 背包问题效果很差，但在一般背包问题中却是可以的。

设

$$\begin{aligned} \max \quad & z = \sum_{j=1}^n p_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^n w_j x_j \leq C \\ & x_j \in \mathbb{Z}^+ \cup \{0\}, j = 1 \sim n \end{aligned}$$

显然, $x_1 = \left\lfloor \frac{C}{w_1} \right\rfloor, x_2 = \dots = x_n = 0$

是一个可行解. $\therefore z_{GA_0} \geq p_1 \left\lfloor \frac{C}{w_1} \right\rfloor$

松弛问题的最优解值

通过求解 $C(KP)$ 得 $z_{opt}(I) \leq p_1 \frac{C}{w_1}$

$$\therefore \frac{z_{GA_0}(I)}{z_{opt}(I)} \geq \frac{\left\lfloor \frac{C}{w_1} \right\rfloor}{\frac{C}{w_1}} \geq \frac{\left\lfloor \frac{C}{w_1} \right\rfloor}{1 + \left\lfloor \frac{C}{w_1} \right\rfloor} \geq \frac{1}{2}$$

$$\therefore R_{GA_0} \geq \frac{1}{2}$$

$I : p_1 = k + 3\varepsilon, p_2 = k$
 $w_1 = \varepsilon + k/2, w_2 = k/2$
 $\varepsilon > 0, C = k$

进一步可证 $R_{GA_0} = \frac{1}{2}$

1975年 *Sahni* 给出一个多项式时间近似算法 .

算法 S_k :

step 1 对任意满足 $M \subseteq N = \{1, 2, \dots, n\}, |M| \leq k$

且 $\sum_{i \in M} w_i \leq C$ 的子集 M , 先将 M 中的物品放入包内,

然后用算法 GA_1 或 GA_2 求解一个如下定义的 KP :

物品集为 $N \setminus M$, 包容量为 $C - \sum_{i \in M} w_i$, 将得到的解与

M 的并作为原问题的近似解 .

step 2 取上述所有不同解中最好一个作为输出 .

Theorem 6.5 $R_{S_k} = 1 + \frac{1}{k}$

§ 4 0-1背包问题的一些相关问题

一、有界背包问题 (*Bounded Knapsack Problem*)

0-1 背包问题: $x_j = 0 \text{ or } 1$;

一般背包问题: $x_j \in \mathbb{Z}^+ \cup \{0\}$; (无界背包问题)

有界背包问题: $0 \leq x_j \leq b_j$ b_j 为给定的正整数 .

显然, GKP 可化为 BKP , 只需令 $b_j = \left\lfloor \frac{C}{w_j} \right\rfloor$

BKP 可化为等价的 0-1 KP

思想: 任一整数可用 0, 1 变量来表示, 如 $x < 16$ 非负整数

$x = x_1 + 2x_2 + 2^2 x_3 + 2^3 x_4$ 如 13 $x_1 = 1, x_2 = 0, x_3 = 1, x_4 = 1$

有界背包问题的求解

与前面讨论一样，总可假定 p_j, w_j, b_j, C 都是正整数；

$$\sum_{j=1}^n b_j w_j > C, \quad b_j w_j \leq C, \quad \frac{p_1}{w_1} \geq \frac{p_2}{w_2} \geq \dots \geq \frac{p_n}{w_n}.$$

Theorem 6.7 记 $s = \min \left\{ j \mid \sum_{i=1}^j b_i w_i > C \right\}$, $\bar{C} = C - \sum_{j=1}^{s-1} b_j w_j$,

则 (1) $U_1 = \sum_{j=1}^{s-1} b_j p_j + \left\lfloor \bar{C} \frac{p_s}{w_s} \right\rfloor$ 是 BKP 的一个上界；

(2) 取 $\bar{x}_j = b_j (j = 1, 2, \dots, s-1)$ $\bar{x}_j = 0 (j = s, \dots, n)$
得到一个可行解，将此解的值与 $b_s p_s$ 的值比较，
取大者为输出。

该算法的绝对性能比 $R = \frac{1}{2}$ ，计算复杂性为 $O(n \log n)$ 。

二、子集和问题

在组合优化问题中，很多问题是相通的，参数稍作修正，可能就是另一个问题。

在 0-1 背包问题中，令 $p_j = w_j$ (这时 $\frac{p_j}{w_j} = 1, j = 1 \sim n$)

即 $\max z = \sum_{j=1}^n w_j x_j$ 称为子集和问题

(SSP) s.t. $\sum_{j=1}^n w_j x_j \leq C$ Subset Sum Problem

$x_j = 0 \text{ or } 1 \quad j \in N = \{1, 2, \dots, n\}$

SSP 是 0-1KP 的特殊情形，所以原方法皆可用。但因其特殊，它应有更简单的方法。

子集和问题的求解

1、贪婪算法(GA):

按任意顺序将物品逐个放入包内，直到放不下为止，然后将其结果与最大重量的物品比较，取优者为输出。

可证 $R_{GA} = \frac{1}{2}$ 计算复杂性为 $O(n)$ 。

2、近似算法 MTGS :

将上述的贪婪算法分别用于物品集 $\{1, 2, \dots, n\}$, $\{2, 3, \dots, n\}$, $\{3, \dots, n\}$, \dots , $\{n-1, n\}$, $\{n\}$, 取最好者为输出。

这里假定 $w_1 \geq w_2 \geq \dots \geq w_n$ 。

可证 $R_{MTGS} = \frac{3}{4}$ 计算复杂性为 $O(n \log n + n^2) = O(n^2)$ 。

本章结束