《算法设计与分析》

第四章 贪心方法

马丙鹏

2023年10月15日

中国科学院大学
University of Chinese Academy of Sciences

第四章 贪心方法

- 4.1 一般方法
- 4.2 背包问题
- 4.3 带有限期的作业排序
- 4.4 最优归并模式
- 4.5 最小生成树
- 4.6 单源点最短路径

4.1 一般方法

- 1. 优化问题
  - 社会生产和生活中经常碰到的问题
  - 宗旨: 少花钱多办事
    - 企业经营管理中 → 优化生产计划
    - 新产品设计中 → 优化性能成本比
    - 日常生活中 → 花尽可能少的钱办尽可能多的事
    - 行程中 → 走最短的路程到达目的地
    - 等等
  - 期望以事半功倍之术，以求或提效、或增收、或节约等等之目标。
  - 所有类似的这种课题统称为最优化问题。

4.1 一般方法

■ 1. 优化问题

- 从数学的角度看，最优化问题可以概括为这样一种数学模型：
  - 给定一个“函数”F(X)，以及“自变量”X，X应满足的一定条件。
  - 求X为怎样的值时，F(X)取得其最大值或最小值。用数学符号简洁地表示成：

    Min F(X) 或 Max F(X)
  - F(X)称为“目标函数”，
  - X应满足的条件为“约束条件”。约束条件一般用一个集合D表示为：X∈D。

4.1 一般方法

- 1. 优化问题
  - 更通俗地讲就是：
    - 求问题的解，但有一定的条件限制，
    - 满足条件的解可能有多个，不同的解性质上有差异，
    - 如何找出其中最好的解。
  - 枚举法：解决大部分最优化问题的最直接的办法：
    - 罗列可能的解，比较解的“效益值”，确定问题的最优解。
    - ——最“笨”的方法！
  - 有没有更好的方法解决该类问题？

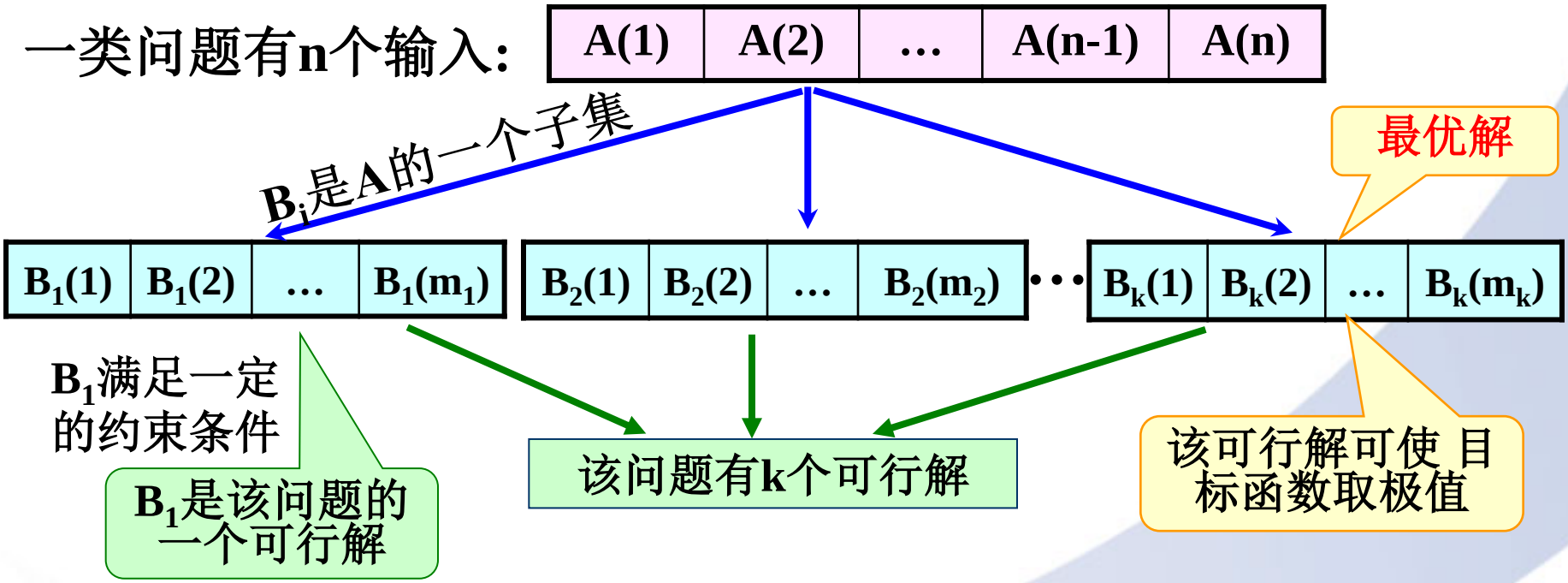
4.1 一般方法

■ 2. 问题的一般特征

- 问题有n个输入，问题的解是由这n个输入的某个子集组成，这个子集必须满足某些事先给定的条件。
  - 约束条件: 子集必须满足的条件；
  - 可行解: 满足约束条件的子集；可行解可能不唯一；
  - 目标函数: 用来衡量可行解优劣的标准，一般以函数的形式给出；
  - 最优解: 能够使目标函数取极值(极大或极小)的可行解。

4.1 一般方法

■ 2. 问题的一般特征



4.1 一般方法

■ 2. 问题的一般特征

□例[找零钱]

- 一个小孩买了价值少于1元的糖，并将1元的钱交给售货员。售货员希望用数目最少的硬币找给小孩。假设提供数目不限的面值为50分、10分、5分及1分的硬币。售货员分步骤组成要找的零钱数，每次加入一个硬币。
- 选择硬币时所采用的贪心算法如下：
  - ✓每一次选择应使零钱数尽量增大。
  - ✓为确保解法的可行性(即：所给的零钱等于要找的零钱数)，所选择的硬币不应使零钱总数超过最终所需的数目。

4.1 一般方法

■ 2. 问题的一般特征

□例[找零钱]

- ➢假设需要找给小孩67分，
  - ✓首先入选的是一枚50分的硬币，
  - ✓第二枚入选的不能是50分的硬币，否则将不可行(零钱总数超过67分)，
  - ✓第二枚应选择10分的硬币，
  - ✓然后是一个5分的硬币，
  - ✓最后加入两个1分的硬币。
- ➢贪心算法有种直觉的倾向，在找零钱时，直觉告诉我们应使找出的硬币数目最少(至少是接近最少的数目)。

4.1 一般方法

■ 3. 贪心方法的一般策略

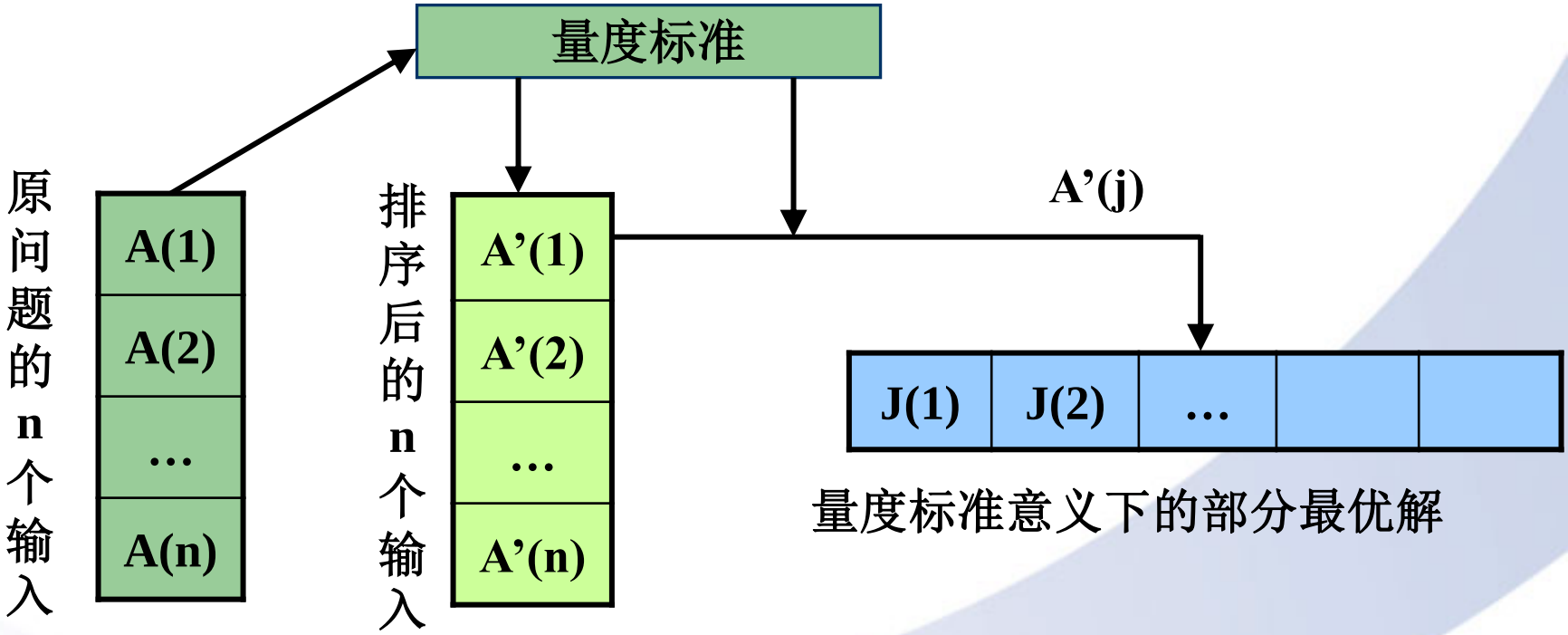
□一般方法

- 初始化：已知问题有n个输入，置问题的解集合J为空；
- 选量度标准：根据题意，选取一种度量标准。然后按照这种度量标准对n个输入排序。
- 考察输入：按序一次输入一个量，看该量能否和J中已选出来的元素（称为该量度意义下的部分最优解）加在一起构成新的可行解：
  如果可以，则把该量并入J集合，从而得到一个新的部分解集合；
  如果不可以，则丢弃该量，J集合保持不变。
  之后，继续上述过程，考察下一输入量，直到所有的输入都考察完毕。
- 获得贪心解：所有的n个输入都被考虑完毕，被记入到集合J中的输入量构成了这种量度意义下的问题的最优解。

4.1 一般方法

■ 3. 贪心方法的一般策略

□一般方法



4.1 一般方法

■ 3. 贪心方法的一般策略

□贪心方法

- ➢这种能够得到某种度量意义下的最优解的分级处理方法称为贪心方法
- ➢注：
  - ✓贪心解 $\stackrel{?}{=}$ 最优解
  - ✓直接将目标函数作为度量标准也不一定能够得到问题的最优解

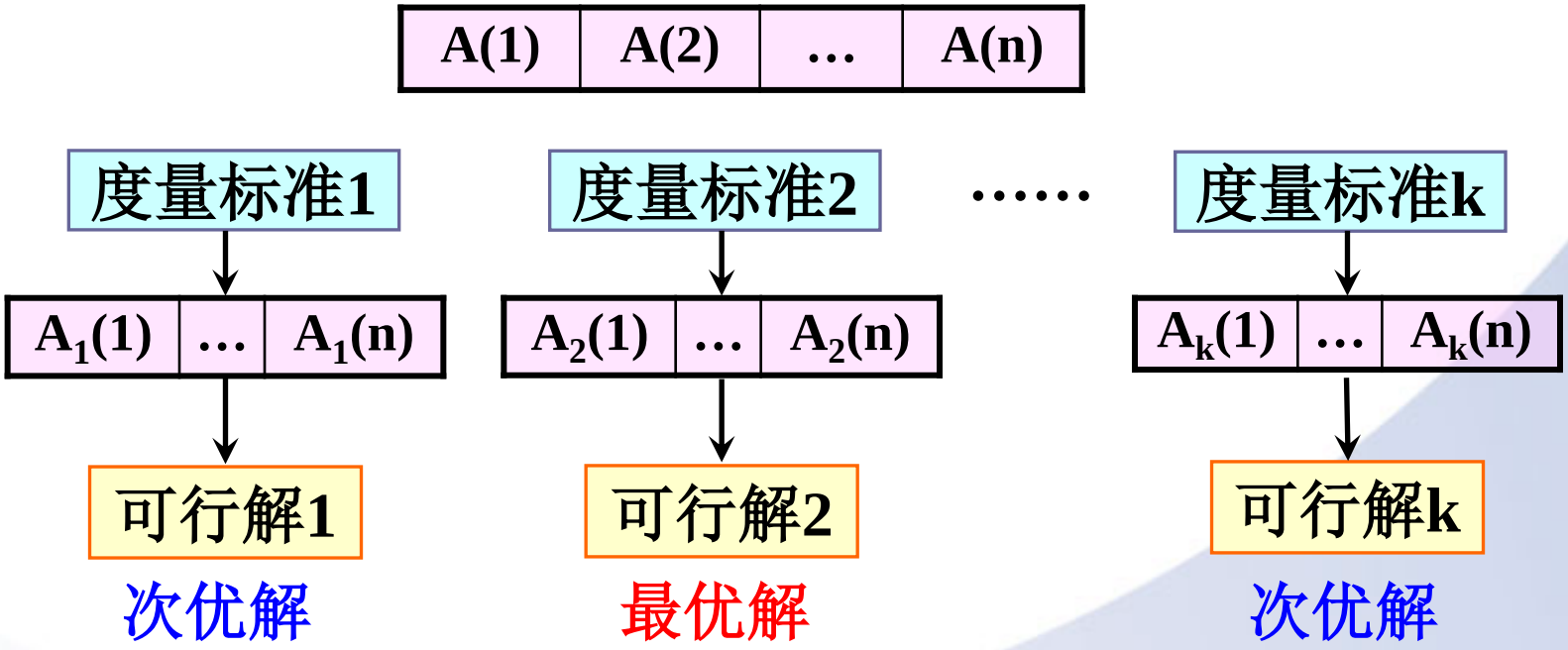
□使用贪心策略求解的关键

- ➢选取能够得到问题最优解的度量标准

4.1 一般方法

■ 3. 贪心方法的一般策略

□贪心方法



4.1 一般方法

■ 4. 贪心方法的抽象化控制描述

```
procedure GREEDY(A, n)
    //A(1:n)包含n个输入//
    solution←Φ     //将解向量solution初始化为空//
    for i←1 to n do
      x←SELECT(A)
      if FEASIBLE(solution, x)  then
          solution←UNION(solution, x)
      endif
    repeat
    return (solution)
end GREEDY
```

按照度量标准，从A中选择一个输入，其值赋予x，并将之从A中删除

判断x是否可以包含在解向量中

将x和当前的解向量合并成新的解向量，并修改目标函数

第四章 贪心方法

- 4.1 一般方法
- 4.2 背包问题
- 4.3 带有限期的作业排序
- 4.4 最优归并模式
- 4.5 最小生成树
- 4.6 单源点最短路径

4.2 背包问题

- 1. 问题的描述
  - 已知n种物品具有重量($w_1, w_2, \ldots, w_n$)和效益值($p_1, p_2, \ldots, p_n$)，
  - 一个可容纳M重量的背包；
  - 设当物品i全部或一部分x_i放入背包将得到$p_i x_i$的效益，
  - 这里，$0 \le x_i \le 1, p_i > 0$。
  - 问题：采用怎样的装包方法才能使装入背包的物品的总效益最大？

4.2 背包问题

■ 1. 问题的描述

□分析：

① 装入背包的总重量不能超过M

② 如果所有物品的总重量不超过M，即$\sum_{1\le i\le n} w_i x_i \le M$，则把所有的物品都装入背包中将获得最大可能的效益值

③ 如果物品的总重量超过了M，则将有物品不能(全部)装入背包中。由于$0\le x_i\le 1$，所以可以把物品的一部分装入背包，所以最终背包中可刚好装入重量为M的若干物品(整个或一部分)

4.2 背包问题

■ 1. 问题的描述

□ 目标：

- ➢ 使装入背包的物品的总效益达到最大。

□ 问题的形式描述

- ➢ 目标函数：$\sum_{1\le i\le n} p_i x_i$

- ➢ 约束条件：$\sum_{1\le i\le n} w_i x_i \le M$

$$0 \le x_i \le 1, p_i > 0, w_i > 0, 1 \le i \le n$$

4.2 背包问题

■ 1. 问题的描述

□ 问题的形式描述

- ➢ 可行解：
  - ✓ 满足上述约束条件的任一集合$(x_1, x_2, \ldots, x_n)$ 都是问题的一个可行解——可行解可能为多个。$(x_1, x_2, \ldots, x_n)$称为问题的一个解向量。
- ➢ 最优解：
  - ✓ 能够使目标函数取最大值的可行解是问题的最优解
  - ✓ ——最优解也可能为多个。

4.2 背包问题

■ 1. 问题的描述

□例4.1 背包问题的实例

➢设，n=3，M=20，$(p_1, p_2, p_3) = (25, 24, 15)$，$(w_1, w_2, w_3) = (18, 15, 10)$。可能的可行解如下：

	(x_1, x_2, x_3)	$\sum w_i x_i$	$\sum p_i x_i$
①	(1, 2/15, 0)	20	28.2
②	(1, 0, 1/5)	20	28
③	(0, 2/3, 1)	20	31
④	(0, 1, 1/2)	20	31.5

4.2 背包问题

■ 2. 贪心策略求解

□度量标准的选择：三种不同的选择

(1) 以目标函数作为度量标准

➢每装入一件物品，就使背包获得最大可能的效益增量。

该度量标准下的处理规则：

① 按效益值的非增次序将物品一件件地放入到背包；

② 如果正在考虑的物品放不进去，则只取其一部分装满背包：

如果该物品的一部分不满足获得最大效益增量的度量标准，则在剩下的物品种选择可以获得最大效益增量的其它物品，将它或其一部分装入背包。

4.2 背包问题

■ 2. 贪心策略求解

(1) 以目标函数作为度量标准

➢如：若ΔM=2,背包外还剩两件物品i，j，且有(p_i=4，w_i=4) 和(p_j=3，w_j=2)，则下一步应选择j而非i放入背包：$p_i/2 = 2 < p_j = 3$

➢实例分析(例4.1)

$(p_1, p_2, p_3) = (25, 24, 15)$，$(w_1, w_2, w_3) = (18, 15, 10)$

∵ $p_1 > p_2 > p_3$

∴ 首先将物品1放入背包，此时$x_1=1$，背包获得$p_1=25$的效益增量，同时背包容量减少$w_1=18$个单位，剩余空间ΔM=2。

其次考虑物品2和3。就ΔM=2而言有，只能选择物品2或3的一部分装入背包。

4.2 背包问题

■ 2. 贪心策略求解

(1) 以目标函数作为度量标准

➢实例分析(例4.1)

$(p_1, p_2, p_3) = (25, 24, 15)$，　$(w_1, w_2, w_3) = (18, 15, 10)$

为使背包的效益有最大的增量，应选择物品2的2/15装包，即

$x_2=2/15$

最后，背包装满，$\Delta M=0$，故物品3不能装入背包，$x_3=0$ 。

背包最终可以获得效益值$= x_1 p_1 + x_2 p_2 + x_3 p_3$

$=28.2$ (次优解，非问题的最优解)

这是一个次优解，原因是背包容量消耗过快

4.2 背包问题

■ 2. 贪心策略求解

(2) 以容量作为度量标准

- 以目标函数作为度量标准所存在的问题：尽管背包的效益值每次得到了最大的增加，但背包容量也过快地被消耗掉了，从而不能装入更多的物品。
- 改进：让背包容量尽可能慢地消耗，从而可以尽量装入更多的物品。
  即，新的标准是：以容量作为度量标准
  该度量标准下的处理规则：

① 按物品重量的非降次序将物品装入到背包；

② 如果正在考虑的物品放不进去，则只取其一部分装满背包；

4.2 背包问题

■ 2. 贪心策略求解

(2) 以容量作为度量标准

➢实例分析(例4.1)

$(p_1, p_2, p_3) = (25, 24, 15)$，　$(w_1, w_2, w_3) = (18, 15, 10)$

∵ $w_3 < w_2 < w_1$

∴ 首先将物品3放入背包，此时$x_3=1$，背包容量减少$w_3=10$个单位，还剩余空间$\Delta M=10$。同时，背包获得$p_3=15$的效益增量。

其次考虑物品1和2。就$\Delta M=10$而言有，也只能选择物品1或2的一部分装入背包。为使背包的按照“统一”的规则，下一步将放入物品2的10/15装包，即

$x_2=10/15=2/3$

4.2 背包问题

■ 2. 贪心策略求解

(2) 以容量作为度量标准

➢实例分析(例4.1)

$(p_1, p_2, p_3) = (25, 24, 15)$， $(w_1, w_2, w_3) = (18, 15, 10)$

最后，背包装满$\Delta M=0$，故物品1将不能装入背包，$x_1=0$ 。

背包最终可以获得效益值$= x_1 p_1 + x_2 p_2 + x_3 p_3$

$=31$ (次优解,非问题的最优解)

存在的问题：效益值没有得到“最大”的增加

这仍是一个次优解, 原因是重量在慢慢消耗的过程中, 效益值却没有迅速的增加。

4.2 背包问题

■ 2. 贪心策略求解

(3) 最优的度量标准

➢影响背包效益值得因素：

① 背包的容量M

② 放入背包中的物品的重量及其可能带来的效益值

➢可能的策略是：

✓在背包效益值的增长速率和背包容量消耗速率之间取得平衡，

✓即每次装入的物品应使它所占用的每一单位容量能获得当前最大的单位效益。

4.2 背包问题

■ 2. 贪心策略求解

(3) 最优的度量标准

- ➢在这种策略下的度量是：
  - ✓已装入的物品的累计效益值与所用容量之比。
- ➢故，新的度量标准是：
  - ✓每次装入要使累计效益值与所用容量的比值有最多的增加(首次装入)和最小的减小(其后的装入)。
- ➢此时，将按照物品的单位效益值：p_i/w_i 比值(密度)的非增次序考虑。

4.2 背包问题

24/18= 1.3
24/15= 1.6
15/10= 1.5

■ 2. 贪心策略求解

(3) 最优的度量标准

➢实例分析(例4.1)

$(p_1, p_2, p_3) = (25, 24, 15)$， $(w_1, w_2, w_3) = (18, 15, 10)$

∵ $p_1/w_1 < p_3/w_3 < p_2/w_2$

∴ 首先将物品2放入背包，此时$x_2=1$，背包容量减少$w_2=15$个单位，还剩余空间$\Delta M=5$。同时，背包获得$p_2=24$的效益增量。

其次考虑物品1和3。此时，应选择物品3,且就$\Delta M=5$而言有，也只能放入物品3的一部分到背包中 。即

$x_3=5/10=1/2$

4.2 背包问题

■ 2. 贪心策略求解

(3) 最优的度量标准

➢实例分析(例4.1)

$(p_1, p_2, p_3) = (25, 24, 15)$， $(w_1, w_2, w_3) = (18, 15, 10)$

最后，背包装满$\Delta M=0$，故物品1将不能装入背包，$x_1=0$ 。

背包最终可以获得效益值$=x_1 p_1 + x_2 p_2 + x_3 p_3$

$=31.5$ (最优解)

4.2 背包问题

■ 3. 背包问题的贪心求解算法

算法4.2 背包问题的贪心算法

```
procedure GREEDY－KNAPSACK(P, W, M, X, n)
  //p(1:n)和w(1:n)分别含有按P(i)/W(i)≥P(i＋1)/W(i＋1)排序的n件物品的效益值和重量。M是背包的容量大小，而x(1:n)是解向量//
  real P(1:n), W(1:n), X(1:n), M, cu;
  integer i, n
  X←0 //将解向量初始化为空//
  cu←M //cu是背包的剩余容量//
  for i←1 to n do
     if W(i) > cu then exit endif
     X(i) ←1
     cu ←cu-W(i)
  repeat
  if i≤n then X(i) ←cu/W(i)  endif
end GREEDY-KNAPSACK
```

若物品i的重量大于背包的剩余重量,则退出循环

若物品i的重量小于等于背包的剩余容量,则物品i可放入背包内

放入物品i 的一部分

4.2 背包问题

- 4. 最优解的证明
  - 即证明：
    - 由第三种策略所得到的贪心解是问题的最优解。
  - 最优解的含义：
    - 在满足约束条件的情况下，可使目标函数取极(大或小)值的可行解。
    - 贪心解是可行解，故只需证明：贪心解可使目标函数取得极值。

4.2 背包问题

■ 4. 最优解的证明

□证明的基本思想：

I. 设出问题最优解的一般形式。

II. 将贪心解与(假设中的)任一最优解相比较。

III. 如果这两个解相同，则显然贪心解就是最优解。

IV. 如果这两个解不同，

① 则肯定有不同的地方，即至少在一个分量x_i上存在不同。

② 设法找出第一个不同的分量位置i；

③ 设法用贪心解的x_i替换最优解对应的分量 ——削去不同的地方；

④ 作某些调整，以使得替换后的“新”最优解还是可行解，即不违反任何约束条件。

⑤ 证明经过上述处理得到的“新”最优解在效益值上与原最优解是相同的，至少不会降低。

4.2 背包问题

■ 4. 最优解的证明

□证明的基本思想：

V. 继续比较新最优解和贪心解，若还存在不同，则重复步骤(IV)，反复进行代换，直到代换后产生的“最优解”与贪心解完全一样。

VI. 结论，在上述代换中，最优解的效益值没有任何损失，那么贪心解的效益值与代换前、后最优解的效益值都是相同的。
所以贪心解如同最优解一样，可使得目标函数取得极值。
从而得证：该贪心解也即问题的最优解。

4.2 背包问题

■ 4. 最优解的证明

定理4.1 如果$p_1/w_1 \geq p_2/w_2 \geq \ldots \geq p_n/w_n$,则算法GREEDY-KNAPSACK对于给定的背包问题实例生成一个最优解。

证明：

设$X=(x_1, x_2, \ldots, x_n)$是GRDDDY-KNAPSACK所生成的贪心解。

① 如果所有的x_i都等于1，则显然X就是问题的最优解。否则，

② 设j是使$x_i \neq 1$的最小下标。由算法可知，

- ✓ $x_i=1 \quad 1 \leq i < j$,
- ✓ $0 \leq x_j < 1$
- ✓ $x_i=0 \quad j < i \leq n$

i	0	1	2	…	j	…	n
x[i]	-	1	1	1	x_j	0	0

4.2 背包问题

■ 4. 最优解的证明

若X不是问题的最优解，则必定存在一个可行解 $Y=(y_1, y_2, \cdots, y_n)$, 使得：

$$\sum p_i y_i > \sum p_i x_i$$

且应有：

$$\sum w_i y_i = M$$

设k是使得$y_k \neq x_k$的最小下标，则有$y_k < x_k$：

a) 若$k<j$，则$x_k=1$。因为$y_k \neq x_k$，从而有$y_k < x_k$

b) 若$k=j$，由于$\sum w_i x_i = M$，且对$1 \leq i < j$，有$y_i=x_i=1$,而对$j<i \leq n$,有$x_i=0$；故此时若$y_k > x_k$，则将有$\sum w_i y_i > M$，与Y是可行解相矛盾。而$y_k \neq x_k$,所以$y_k < x_k$

c) 若$k>j$，则 $\sum w_i y_i > M$，不能成立

4.2 背包问题

■ 4. 最优解的证明

在Y中作以下调整：将y_k增加到x_k，因为$y_k \le x_k$,为保持解的可行性，必须从$(y_{k+1},\ldots,y_n)$中减去同样多的量。设调整后的解为$Z=(z_1, z_2, \ldots, z_n)$，其中$z_i=x_i$，$1\le i\le k$，且有：

则对于Z有：

$$w_k(z_k - y_k) = \sum_{k<i\le n} w_i(y_i - z_i)$$

$$\sum_{1\le i\le n} p_i z_i = \sum_{1\le i\le n} p_i y_i + (z_k - y_k) w_k p_k / w_k - \sum_{k<i\le n} (y_i - z_i) w_i p_i / w_i$$

$$\ge \sum_{1\le i\le n} p_i y_i + (z_k - y_k) w_k p_k / w_k - \sum_{k<i\le n} (y_i - z_i) w_i p_k / w_k$$

$$= \sum_{1\le i\le n} p_i y_i + \left[(z_k - y_k) w_k - \sum_{k<i\le n} (y_i - z_i) w_i\right] p_k / w_k$$

$$= \sum_{1\le i\le n} p_i y_i$$

4.2 背包问题

■ 4. 最优解的证明

由以上分析得，

- 若 $\sum p_i z_i > \sum p_i y_i$ ，则Y将不是最优解；
- 若 $\sum p_i z_i = \sum p_i y_i$ ，
  - ✓ Z是最优解
  - ✓ 则或者Z=X，则X就是最优解；
  - ✓ 或者Z≠X，则重复以上替代过程，或者证明Y不是最优解，或者把Y转换成X，从而证明X是最优解。

作业-课后练习10

- 求以下情况下的背包问题的最优解
  - 利用贪心策略求解下列背包问题设，n=4，M=54，
    $(p_1, p_2, p_3, p_4) = (20, 16, 10, 18)$，
    $(w_1, w_2, w_3, w_4) = (16, 12, 15, 24)$
    求解向量X
    计算$\sum p_i x_i$

作业-课后练习11

- 求以下情况下的背包问题的最优解
  - n=7, M=15, $(p_1, p_2, p_3, p_4, p_5, p_6, p_7)$= (10, 5, 15, 7, 6, 18, 3)和$(w_1, w_2, w_3, w_4, w_5, w_6, w_7)$=(2, 3, 5, 7, 1, 4, 1)
  - 将以上数据情况的背包问题记为I。
    设FG(I)是物品按照p_i的非增次序输入时由GREEDY-KNAPSACK所生成的解，FO(I)是一个最优解。
    问FO(I)/FG(I)是多少？
  - 当物品按照w_i的非降次序输入时，重复上述讨论。

End