第5章 动态规划

5.1, 5.2 多阶段决策和最优化原理

动态规划研究的问题

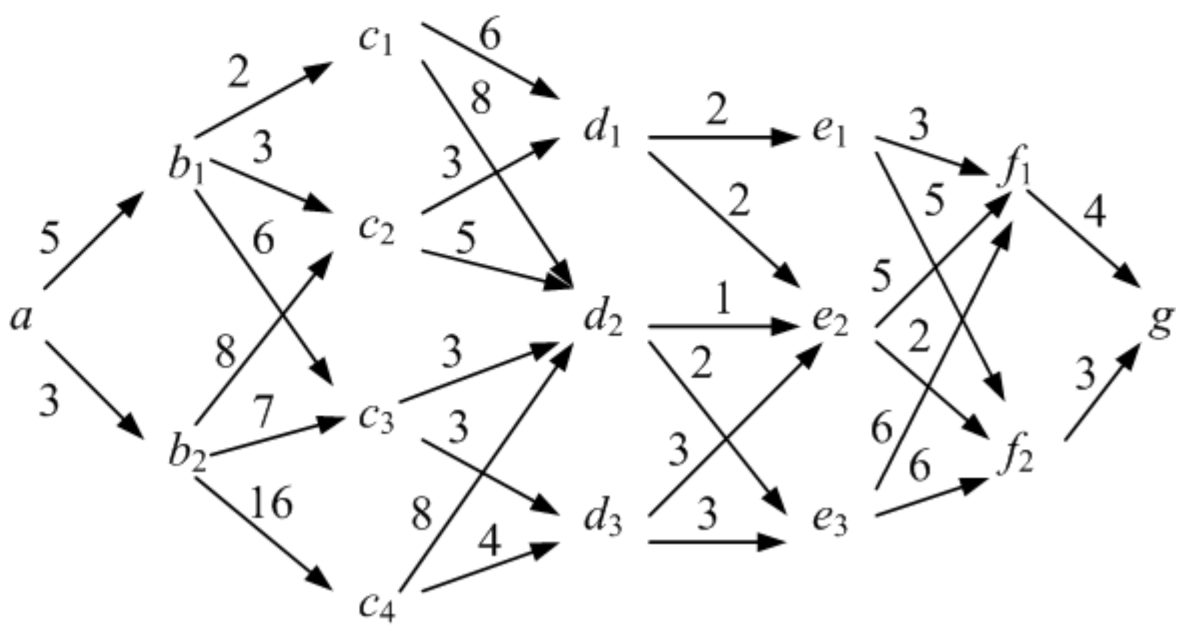
- 动态规划起源于1950年代，创始人为R. Bellman。
- 动态规划所研究的对象是多阶段决策问题。
  - 这样的问题可以转化为一系列相互联系的单阶段优化问题。在每个阶段都需要作出决策。
  - 每个阶段的决策确定以后，就得到一个决策序列，称为策略。多阶段决策问题就是求一个策略，使各阶段的总体目标达到最优（如最小化费用，或最大化收益）。
- 相对于线性规划一次性地对一个问题求出整体最优解，多阶段决策问题的这种解决办法称为动态规划（Dynamic Programming）。而原来的线性规划方法被称为静态规划。

内容

- 多阶段决策问题和最优化原理
- 定期多阶段决策问题
- 不定期多阶段决策问题

问题举例一：最短路问题

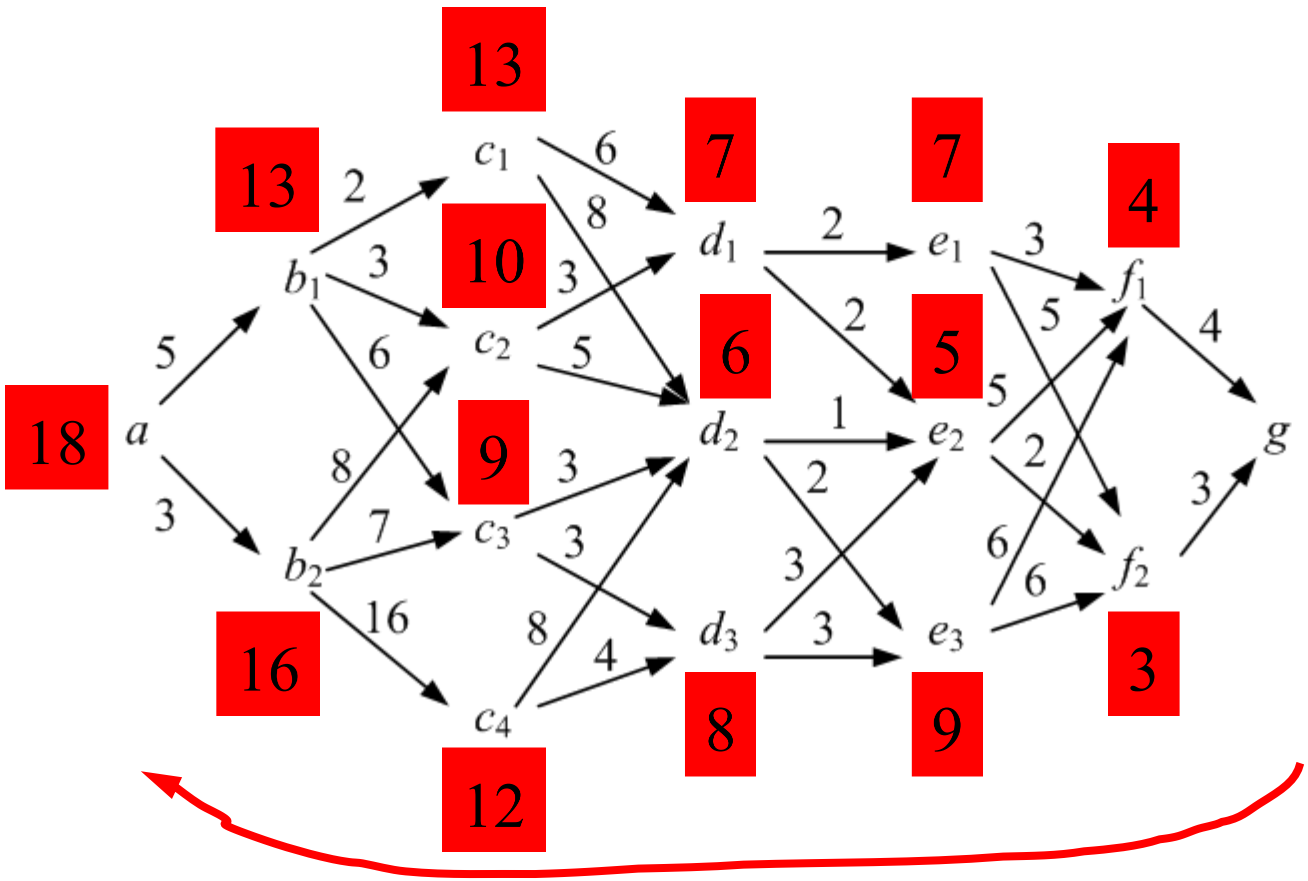
- 如图，求从a到g的最短路。
- 由于这是一个多阶段图（分层图），该图上的最短路问题是一个多阶段决策（优化）问题。



如何求解？（后向优化）

- 由于这是一个多阶段图，从 a 到 g 的任何一条路径的边数都是 6。
- 从 a 到 g，必然要经过 a 的下一个阶段中的顶点 b_1 或 b_2。因此，从 a 到 g 的最短路就是从 a 到 b_1，然后再从 b_1 走到 g，以及从 a 到 b_2，再从 b_2 走到 g，两种走法中最短的一个。
- 于是，定义 $f_k(u, g)$为从当前顶点 u 开始经过 k 条边到达 g 的最短路长度。则有：$\begin{cases} f_k(u,g) = \min\limits_{v \in N^+(u)} \{l(u,v) + f_{k-1}(v,g)\}, & k \geq 2 \\ f_1(u,g) = d(u,g), & k = 1 \end{cases}$。
- 原问题即是求 $f_n(a, g)$（$n = 6$）。

逆向求解递推方程（标号法）



动态规划表格

- 行标 i：从目标顶点 g 开始，倒数第 i 层（即，计算 $f_i(u)$）。
- p_1, p_2, p_3, p_4：倒数第 i 层的自上而下的各个顶点（最多 4 个）。

	p_1	p_2	p_3	p_4
1	4	3	×	×
2	7	5	9	×
3	7	6	8	×
4	13	10	9	12
5	13	16	×	×
6	18	×	×	×

	p_1	p_2	p_3	p_4
1	1	1	×	×
2	1	2	2	×
3	2	2	2	×
4	1	1	2	3
5	2	3	×	×
6	1	×	×	×

- 最短路为：$a \rightarrow b_1 \rightarrow c_2 \rightarrow d_1 \rightarrow e_2 \rightarrow f_2 \rightarrow g$。

递归如何？循环如何？

- 递归程序

$f(i, u, g)$

if $i = 1$ **then return** $d(u, g)$，

else return min $_{v \in N(u)} \{d(u, v) + f(i-1, v, g)\}$。

- 循环程序

$f(u, g)$

$p \leftarrow (a, b_1, c_1, d_1, e_1, d_1, f_1, g)$。

for $b \leftarrow b_1$ 到 b_2 **do**

    for $c \leftarrow c_1$ 到 c_4 **do**

        for $d \leftarrow d_1$ 到 d_3 **do**

            for $e \leftarrow e_1$ 到 e_3 **do**

循环？

```
            for f ← f1 到 f2 do
                q ← (a, b, c, d, e, f, g)。
                if q的长度 < p的长度 then p ← q。
            endfor  /* f */
        endfor  /* e */
    endfor  /* d */
  endfor  /* c */
endfor  /* b */
return p。
```

问题举例二：资源分配问题

- 设有数量为x的某种资源，将它投入两种生产方式A和B中（或称为投给部门A和部门B）。
- 若投给部门A的数量为z，则可获收益$g(z)$，回收az，其中a（$0 \le a \le 1$）为部门A的回收率。类似地，若投给部门B的数量为z，则可获收益$h(z)$，回收bz，其中b（$0 \le b \le 1$）为部门B的回收率。
- 连续投放n个阶段，问每个阶段如何分配资源才能使总收入最大?

再描述一遍

- 设第k个阶段的资源总数为x_k，投给部门A的资源数量为y_k。则投给部门B的数量为$x_k - y_k$。于是可得到收入$g(y_k) + h(x_k - y_k)$，回收$ax_k + b \cdot (x_k - y_k)$。
- 因此，问题就成为：求$y_1, y_2, \cdots, y_n$，最大化 $\sum_{1 \le k \le n} g(y_k) + h(x_k - y_k)$，且满足条件

$$x_1 = x$$

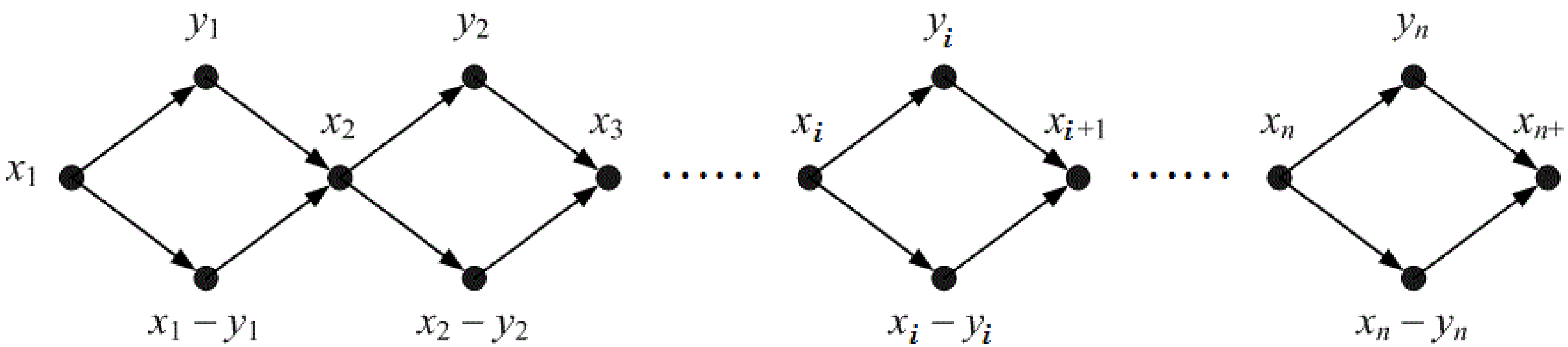
$$x_2 = ay_1 + b(x_1 - y_1)$$

$$\cdots\cdots$$

$$x_n = ay_{n-1} + b(x_{n-1} - y_{n-1})$$

$$y_k \ge 0，\quad x_k \ge 0，\quad k = 1..n-1$$

如何求解？（后向优化）



- 令 $f_k(x)$表示当前资源数量为 x，再经过 k 个阶段投放完成系统目标，所得到的最大总收入。
- 则有：$\begin{cases} f_k(x)=\max\limits_{0\le y\le x}\{g(y)+h(x-y)+f_{k-1}(ay+b(x-y))\}, & k\ge 2 \\ f_1(x)=\max\limits_{0\le y\le x}\{g(y)+h(x-y)\}, & k=1 \end{cases}$。
- 原问题即是求 $f_n(x)$。

例5.1.2

例5.1.2：离散变量的资源分配问题。

- 今有 1000 台机床（$x = 1000$），投放到 A、B 两个部门。
- 若给部门 A 投放 z 台机床，则产生效益 $g(z) = z^2$，回收 $0.8z$ 台机床（$a = 0.8$）。
- 若给部门 B 投放 z 台机床，则产生效益 $h(z) = 2z^2$，回收 $0.4z$台机床（$b = 0.4$）。
- 问连续投放 5 年（$n = 5$），每年如何投放，可使 5 年的总收益最大?

如何求解？

- 行标 k：到达目标，还需要多少个阶段，即，计算 $f_k(x)$。
- 列标 x：可能的资源数。

$k \backslash x$	0	1	2	3	…			1000
1	$f_1(0)$	$f_1(1)$	$f_1(2)$	$f_1(3)$	…			$f_1(1000)$
2	$f_2(0)$	$f_2(1)$	$f_2(2)$		…			$f_2(1000)$
3	$f_3(0)$				…			$f_3(1000)$
4	$f_4(0)$				…			$f_4(1000)$
5	$f_5(0)$				…			$f_5(1000)$

- 问题的目标就是计算 $f_5(1000)$。
- 从左上到右下，计算整个表，可求得问题的解。

计算一个单元格

$f(k, x)$

$v \leftarrow -\infty$。

for $y \leftarrow 0$ 到 x do

  $t \leftarrow g(y) + h(x-y) + \text{table}(k-1, \lfloor ay + b(x-y) \rfloor)$。

  /* 令 $\text{table}(0, x) = 0$ 。 */

  if $t > v$ then $v \leftarrow t$。

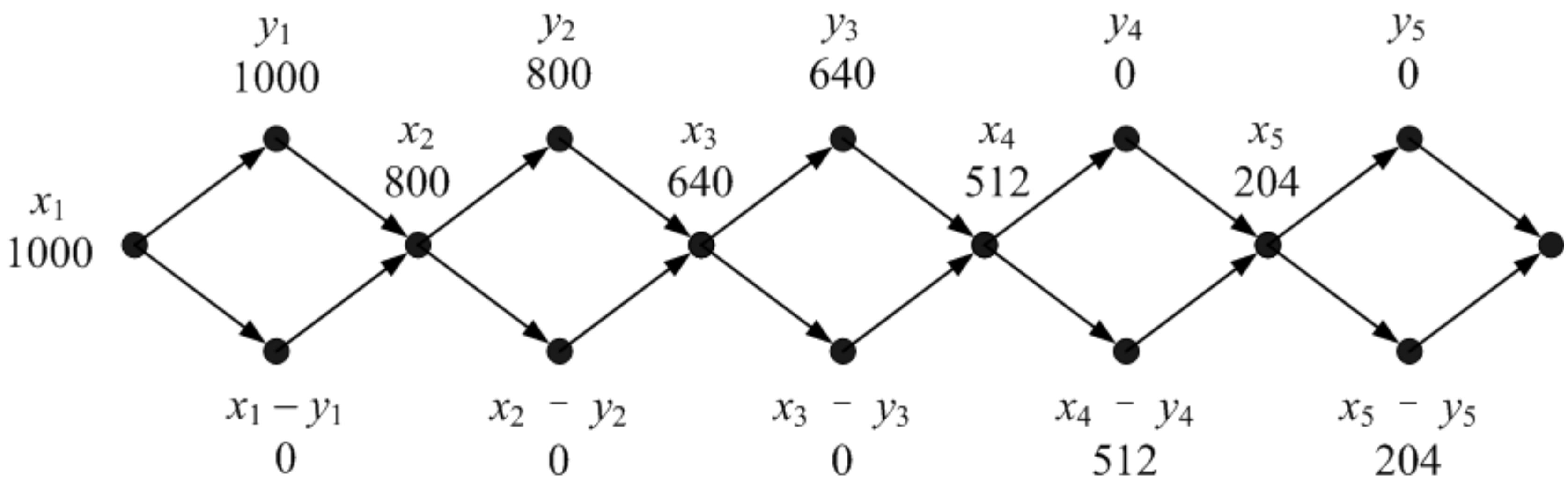
endfor

return v。

说明

- 每计算一个单元格的 $f_k(x)$，都需要计算一个 $\max_{0 \le y \le x} \{\ldots\}$ 函数。因此，尽管使用表格暂存了计算结果，为计算出最后的 $f_n(x)$ 仍需要大量的计算。
- 小技巧：不用每行都从 0 计算到 1000。每年无论如何投放，回收的机床最多是 $0.8x$ 台（$\max\{a, b\} = 0.8$）。例如表格第 5 行表示最后一个阶段，其前面有 4 个阶段。因此对于第5行，只需要从 0 计算到 $0.8^4 \times 1000 \cong 4096$。
- 但动态规划法已经比直接用递归的方法解递推方程减少了大量的计算。

计算结果



$f_5(1000) = 2657120$ $f_3(640) = 950272$ $f_1(204) = 83232$

$f_4(800) = 1657120$ $f_2(512) = 607520$

递归的方法

$f(k, x)$

$v \leftarrow 0$。
if $k = 1$ then
    for $y \leftarrow 0$ 到 x do
        $t \leftarrow g(y) + h(x - y)$。
        if $t > v$ then $v \leftarrow t$。
    endfor
else
    for $y \leftarrow 0$ 到 x do
        $t \leftarrow g(y) + h(x - y) + f(k - 1, ay + b(x - y))$。
        if $t > v$ then $v \leftarrow t$。

递归的方法

```
    endfor
endif
return v。
```

多阶段决策问题

- 有一个系统，可以分成若干个阶段。
- 任意一个阶段 k，系统的状态可以用 x_k 表示（可以是数量、向量、集合等）。
- 每一状态 x_k 都有一个决策集合 $Q_k(x_k)$，在 $Q_k(x_k)$ 中选定一个决策 $q_k \in Q_k(x_k)$，状态 x_k 就转移到新的状态$x_{k+1} = T_k(x_k, q_k)$，并且得到效益（或费用）$R_k(x_k, q_k)$。
- 系统的目标就是在每一个阶段都在它的决策集合中选择一个决策，使所有阶段的总效益达到最大（或总费用达到最小）。
- 这样的多阶段决策问题通常使用动态规划方法来求解。

动态规划的最优子结构性质

动态规划的最优化原理：需要问题的最优解具有如下所述的最优子结构性质：

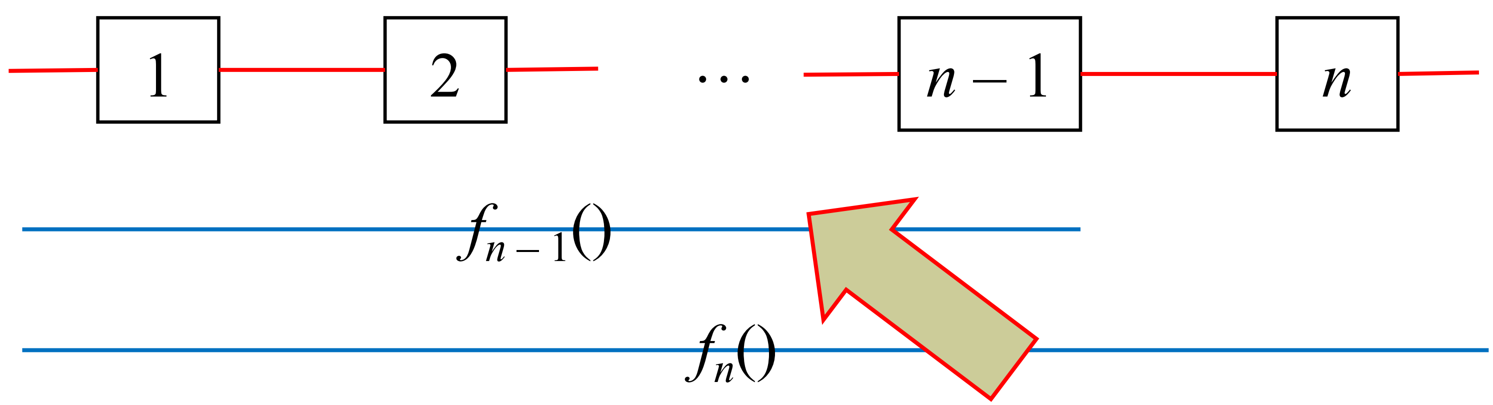
- 一个多阶段决策问题，假设其最优策略的第一阶段的决策为 q_1，系统转移到的新状态为 x_2。则该最优策略以后诸决策对以 x_2 为初始状态的子问题而言，必须构成其最优策略。
- 该子问题与原问题是同一类问题，只是问题规模下降了。
- 当观察到问题解的最优子结构性质时，就意味着问题可能用动态规划法求解。

动态规划的子问题重叠性质

- 从算法角度而言，（离散变量的）动态规划所依赖的另一个要素是子问题重叠性质：一个问题的求解可以划分成若干子问题的求解，而处理这些子问题的计算是部分重叠的。
- 动态规划法利用问题的子问题重叠性质设计算法，能够节省大量的计算。对一些看起来不太可能快速求解的问题，往往能设计出多项式时间算法。
- 在算法理论中，多项式时间算法通常被认为是“有效的算法”（**efficient algorithm**）。

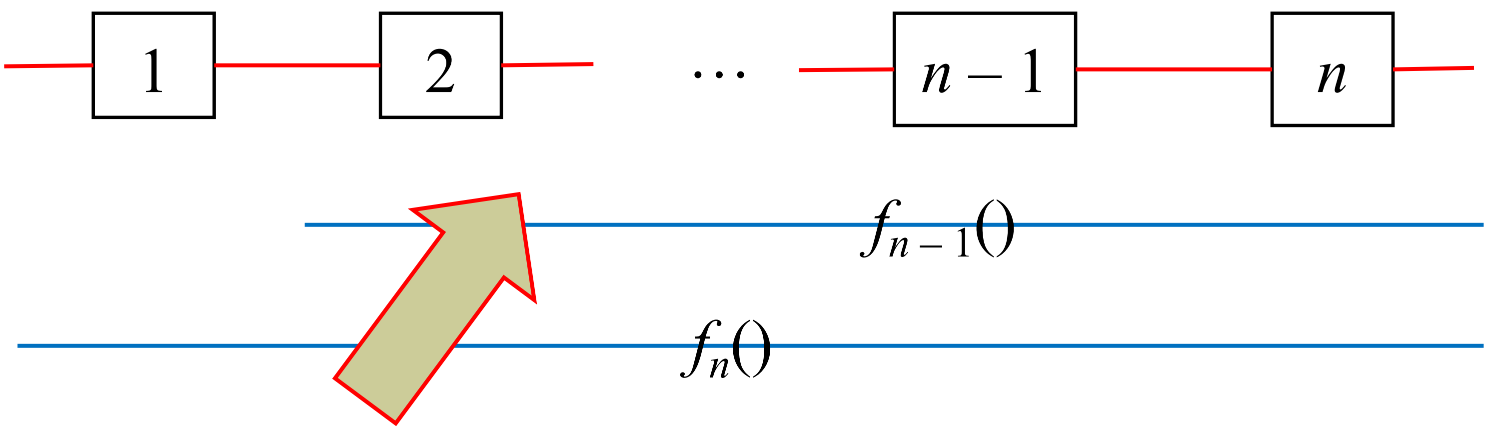
前向优化

- 写动态规划递推方程时，一般有两种写法：前向优化和后向优化。假设问题有n个阶段。
- 定义$f_k()$ 为问题的前k个阶段（从第1阶段到第k阶段）的最优解值，然后将$f_k()$递推至$f_{k-1}()$。
- 最后写出递推的终止条件$f_1()$的表达式。
- 原问题就是计算$f_n()$。这称为前向优化。



后向优化

- 假设问题有n个阶段。
- 定义$f_k()$ 为问题的后k个阶段（从第$n-k+1$阶段到第n阶段）的最优解值，然后将$f_k()$递推至$f_{k-1}()$。
- 最后写出递推的终止条件$f_1()$的表达式。
- 原问题就是计算$f_n()$。这称为后向优化。



说明

- 前面给出的两个例子，最短路问题和资源分配问题，都是采用后向优化技术解决的。
- 原则上，多阶段决策问题既可以使用前向优化技术解决，也可以使用后向优化技术解决。
- 依据问题不同，前向优化和后向优化其中的一种或二者是“自然”的解法。

最短路问题，前向优化

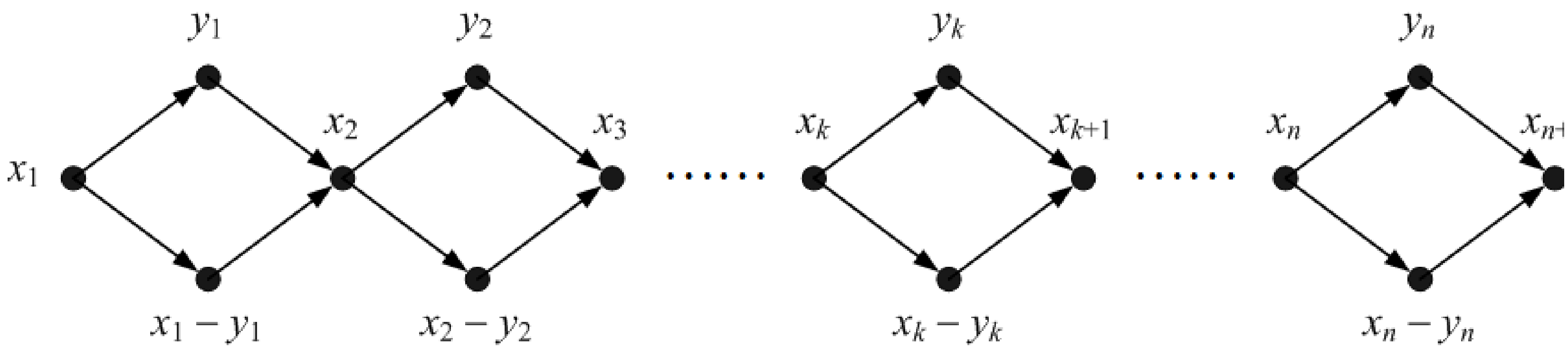
- 定义 $f_k(a, u)$为从顶点 a 经过 k 条边到达当前顶点 u 的最短路长度。则有：

$$\begin{cases} f_k(a,u) = \min\limits_{w \in N^-(u)} \{f_{k-1}(a,w) + l(w,u)\}, & k \ge 2 \\ f_1(a,u) = l(a,u), & k = 1 \end{cases}$$

。

- 原问题即是求 $f_n(a, g)$（$n = 6$）。

资源分配问题，前向优化



- 令 $f_k(x_{k+1})$表示从第 1 阶段连续生产到第 k 阶段，还余下恰好 x_{k+1} 份资源，所产生的最大效益。
- 则有如下递推方程：当 $k = 2..n$ 时，

$$f_k(x_{k+1}) = \begin{cases} \max\limits_{0 \le y_k \le \frac{1}{a}x_{k+1}} \{f_{k-1}(x_k) + g(y_k) + h(x_k - y_k)\}, & \min\{a^{k-1}x_1, b^{k-1}x_1\} \le x_k \le \max\{a^{k-1}x_1, b^{k-1}x_1\} \\ -\infty, & \text{o.w.} \end{cases},$$

给定 x_{k+1}、y_k，由于 x_k、x_{k+1}、y_k 需满足 $ay^k + b(x_k - y_k) = x_{k+1}$，

可直接计算出 $x_k = \frac{1}{b}\left(x_{k+1} + (b-a)y_k\right)$。

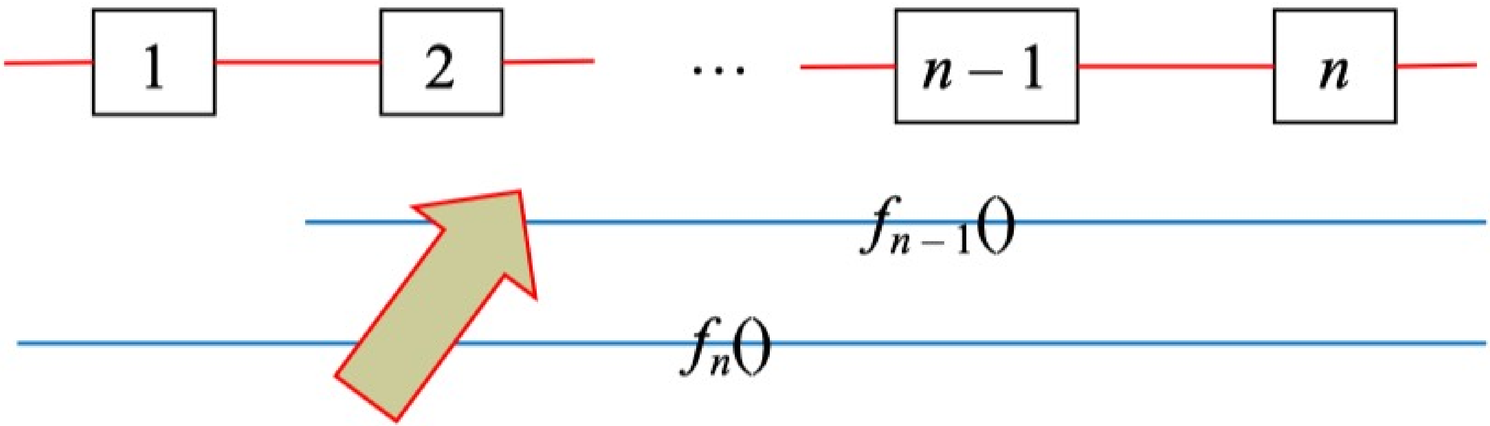
资源分配问题，前向优化

- 当 $k=1$ 时，$f_1(x_2)=\begin{cases} g(y_1)+h(x_1-y_1), & 0\le y_1=\dfrac{bx_1-x_2}{b-a}\le x_1 \\ -\infty, & \text{o.w.} \end{cases}$。

  由已知 x_1、x_2，因为 x_1、x_2、y_1 需要满足 $ay_1+b(x_1-y_1)=x_2$，可直接计算出 $y_1=\dfrac{bx_1-x_2}{b-a}$。
- 最后，对 $\min\{a^n x_1, b^n x_1\}\le x_{n+1}\le \max\{a^n x_1, b^n x_1\}$ 计算所有可能的 $f_n(x_{n+1})$，然后选取一个最大值，即为原问题的最优解。

动态规划所研究的问题

- 阶段数固定还是不固定？
  - 阶段数固定（也称为“定期”），指阶段数有限且固定。



但不固

划方法

求解。

谢谢大家