組合せ最適化問題に対する解法

厳密解法 最悪の場合，求解までに時間がかかるかもしれないが厳密に最適解を求める

- 分枝限定法
- 分枝カット法
- 動的計画法

- ...

近似解法 最適解を求めることを諦めて，ある程度最適値に近い値をもつ実行可能解をなるべく早く求める

- 制度保証付き近似解法
- メタヒューリスティック（タブーサーチ，遺伝的アルゴリズム，アニーリング法...）
- 乱択アルゴリズム
- ...

0-1 ナップサック問題に対する動的計画法 1

$$
\begin{array}{ll}
\text{最大化} & c_1x_1 + c_2x_2 + \cdots + c_nx_n \\
\underset{\boldsymbol{x}}{\text{条件}} & a_1x_1 + a_2x_2 + \cdots + a_nx_n \leq b \\
& x_j \in \{0,1\}, \quad j = 1,2,\ldots,n
\end{array}
$$

仮定

$a_1, \ldots, a_n$ と b は全て整数

部分問題

最適値を [　　] **とする．**

(元問題の最適値は [　　] **)**

動的計画法 1 の漸化式

k, y をパラメータとして動かしたときの $V_k(y)$ の関係：

$$V_1(y) = \begin{cases} 0 & (a_1 > y) \\ c_1 & (a_1 \leq y) \end{cases}$$

$$V_k(y) = \begin{cases} V_{k-1}(y) & (a_k > y) \\ \max\{V_{k-1}(y), c_k + V_{k-1}(y - a_k)\} & (a_k \leq y) \end{cases}$$

※　上式の意味を説明せよ．

$\Longrightarrow$

動的計画法 1 のアルゴリズム

for $k = 1$ to n **do**
  for $y = 0$ to b **do**
    $V_k(y)$ **を計算**
  end for
end for

$y \leftarrow b$
for $k = n$ to 1 **do**
  if $V_k(y) \neq V_{k-1}(y)$ **then**
    $x_k^* \leftarrow 1;\ y \leftarrow y - a_k$
  else
    $x_k^* \leftarrow 0$
  end if
end for

($V_0(y) = 0,\ \forall y$ **を仮定**)

$$
\begin{array}{ll}
\textbf{最大化} & 6x_1 + 7x_2 + 8x_3 + x_4 + 4x_5 \\
\overset{x}{\textbf{条件}} & 3x_1 + 4x_2 + 6x_3 + x_4 + 5x_5 \leq 12 \\
 & x_j \in \{0,1\}, \qquad j = 1,2,3,4,5
\end{array}
$$

を動的計画法で解く．

$$
V_1(y) = \begin{cases} 0 & (a_1 > y) \\ c_1 & (a_1 \leq y) \end{cases}
$$

$$
V_k(y) = \begin{cases} V_{k-1}(y) & (a_k > y) \\ \max\{V_{k-1}(y), c_k + V_{k-1}(y - a_k)\} & (a_k \leq y) \end{cases}
$$

$k \backslash y$	0	1	2	3	4	5	6	7	8	9	10	11	12
1													
2													
3													
4													
5													

動的計画法 1 の特徴

- 計算の手間は計算過程の表の大きさに依存
  $\Rightarrow$ [　　]
- [　　] ときに有効

なぜ，係数が整数の仮定が必要だったのか？

仮定

$a_1, \ldots, a_n$ と b は全て整数

0-1 ナップサック問題に対する動的計画法 2

$$
\begin{array}{ll}
\textbf{最大化} & c_1x_1 + c_2x_2 + \cdots + c_nx_n \\
\underset{\boldsymbol{x}}{\textbf{条件}} & a_1x_1 + a_2x_2 + \cdots + a_nx_n \leq b \\
& x_j \in \{0,1\}, \quad j = 1,2,\ldots,n
\end{array}
$$

仮定

$c_1, \ldots, c_n$ **は全て整数**

部分問題

最適値を [ ] **とする．**

(元問題の最適値は [ ] **)**

動的計画法 2 の漸化式

k, v **をパラメータとして動かしたとき，** $Y_k(v)$ **の関係**:

$$Y_1(v) = \begin{cases} 0 & (v = 0) \\ a_1 & (v = c_1) \\ \infty & (\textbf{それ以外}) \end{cases}$$

$$Y_k(v) = \begin{cases} Y_{k-1}(v) & (c_k > v) \\ \min\{Y_{k-1}(v), a_k + Y_{k-1}(v - c_k)\} & (c_k \leq v) \end{cases}$$

※　**上式の意味を説明せよ．**

$\Longrightarrow$

v **は** 0 **から** $\square$ **まで調べる必要がある** $\Rightarrow$

動的計画法 2 のアルゴリズム

for $k = 1$ to n **do**
  for $v = 0$ to V **do**
    $Y_k(v)$ **を計算**
  end for
end for

$v \leftarrow \max\{v \mid Y_n(v) \leq b\}$
for $k = n$ to 1 **do**
  if $Y_k(v) \neq Y_{k-1}(v)$ **then**
    $x_k^* \leftarrow 1;\ v \leftarrow v - c_k$
  else
    $x_k^* \leftarrow 0$
  end if
end for

($Y_0(v) = 0\ (\forall v)$ **を仮定)**

$$
\begin{aligned}
&\textbf{最大化} \quad 6x_1 + 7x_2 + 8x_3 + x_4 + 4x_5 \\
&\overset{x}{\textbf{条件}} \quad 3x_1 + 4x_2 + 6x_3 + x_4 + 5x_5 \leq 12 \\
&\qquad\qquad x_j \in \{0,1\}, \qquad j = 1,2,3,4,5
\end{aligned}
$$

$V = 19$ **(連続緩和問題の最適値を切り下げ)**

$k \backslash v$	0	1	2	3	4	5	6	7	8	9	10	11
1												
2												
3												
4												
5												

12	13	14	15	16	17	18	19

$\max\{v \mid Y_n(v) \leq 12\}$

動的計画法 2 の特徴

- 計算の手間は計算過程の表の大きさに依存
  $\Rightarrow$ [　　]
- [　　] ときに有効

なぜ，係数が整数の仮定が必要だったのか？

仮定

$c_1, \ldots, c_n$ は全て整数

ビンパッキング問題に対する動的計画法

n 個のアイテムがあり，各アイテム j のサイズ s_j が分かっている．これらのアイテムを容量が S の容器に詰め込む．ただし，一つの容器に詰め込むアイテムのサイズの合計は S を超えてはいけない．容器の個数を最小にする詰め込み方を決定する

仮定

アイテムのサイズが p 通り $s_1, s_2, \ldots, s_p$ あったとし，サイズ s_k のアイテムが n_k 個ある

与えられるアイテムの情報は，それぞれのサイズのアイテム個数 $(n_1, n_2, \ldots, n_p)$ で表せる．

ビンパッキング問題に対する動的計画法の漸化式

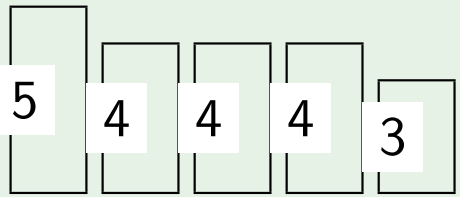
$B(j_1, j_2, \ldots, j_p)$: 各 $i = 1, \ldots, p$ に対して，サイズが s_i のアイテムが j_i 個ときの最小の容器数
(元問題の最適値は $B(n_1, n_2, \ldots, n_p)$)

アイテム数 $(j_1, j_2, \ldots, j_p)$ をパラメータとして動かしたとき，
$B(j_1, j_2, \ldots, j_p)$ の関係:

$$B(0, 0, \ldots, 0) = 0$$

$$B(j_1, j_2, \ldots, j_p) = \min_{(x_1, x_2, \ldots, x_p)} \{B(j_1 - x_1, j_2 - x_2, \ldots, j_p - x_p) + 1 \mid \sum_{k=1}^{p} x_k s_k \leq S\}$$

※　上式の意味を説明せよ．
$\Rightarrow$ 1 つの容器に詰め込めるアイテムの組み合わせ $(x_1, x_2, \ldots, x_p)$ の集合を保持し，任意の $j'_k < j_k$ $(k = 1, \ldots, p)$ を満たす $B(j'_1, j'_2, \ldots j'_p)$ が分かっていれば，この漸化式を用いて $B(j_1, j_2, \ldots, j_p)$ が求まる．



の $(1,3,1)$ を容量 10 の容器に詰める．

１つの容器に詰め込み可能な組合せ (j_1, j_2, j_3) は

$$\{(1,0,0),(0,1,0),(0,0,1),(1,1,0),(1,0,1),(0,1,1),(0,2,0)\}$$

$$\begin{aligned} B(1,1,1) &= \min\{B(0,1,1)+1, B(1,0,1)+1, B(1,1,0)+1, \\ &\qquad B(0,0,1)+1, B(0,1,0)+1, B(1,0,0)+1\} = 2 \end{aligned}$$

同様に， $B(1,2,0) = B(0,2,1) = B(0,3,0) = 2$

$$\begin{aligned} B(1,2,1) &= \min\{B(0,2,1)+1, B(1,1,1)+1, B(1,2,0)+1, \\ &\qquad B(0,1,1)+1, B(0,2,0)+1, B(1,1,0)+1, B(1,0,1)+1\} = 2 \end{aligned}$$

$$B(1,3,0) = \min\{B(0,3,0)+1, B(1,2,0)+1, B(0,2,0)+1, B(1,1,0)+1\} = 2$$

$$B(0,3,1) = \min\{B(0,2,1)+1, B(0,3,0)+1, B(0,2,0)+1, B(0,1,1)+1\} = 2$$

$$\begin{aligned} B(1,3,1) &= \min\{B(0,3,1)+1, B(1,2,1)+1, B(1,3,0)+1, \\ &\qquad B(0,2,1)+1, B(0,3,0)+1, B(1,2,0)+1, B(1,1,1)+1\} = 3 \end{aligned}$$

ビンパッキング問題に対する動的計画法の特徴

- アイテムのサイズの種類が少ないときに有効
- 最小完了時刻最小化スケジューリング問題への応用
  $\Rightarrow$ ビンパッキング問題に対する動的計画法を繰り返し利用する

> n 個のジョブを m 台の機械で処理する．各ジョブ j の処理時間 p_j と重み w_j が分かっており，$\max\{s_j + p_j \mid j = 1, \ldots, n\}$ を最小とする各ジョブ j の処理開始時刻 s_j と処理する機械を決定

  - スケジューリング問題で与えられるジョブの処理時間を，ビンパッキング問題でのアイテムのサイズ　とする
  - 容量 S の容器に詰め込む容器の最小個数が m 個以下ならば，最終完了時刻は S よりも早い
  - 最終完了時刻の候補となる時刻をビンの容量としてビンパッキング問題を解く

演習問題

1. **以下の 0-1 ナップサック問題を 2 通りの動的計画法で解け．**

$$
\left|
\begin{array}{lll}
\textbf{最大化} & 7x_1 + 12x_2 + 8x_3 + 5x_4 + 10x_5 & \\
\overset{\boldsymbol{x}}{\textbf{条}}\ \textbf{件} & 2x_1 + 3x_2 + 5x_3 + 2x_4 + 8x_5 \leq 11 & \\
& x_j \in \{0,1\}, \qquad j = 1,2,3,4,5 &
\end{array}
\right.
$$

2. **ビンパッキング問題に対する動的計画法は，アイテムのサイズの種類が少ないときに有効であるのはなぜか．**

3. **複数アイテムを選択できるナップサック問題：**

$$
\left|
\begin{array}{lll}
\textbf{最大化} & c_1x_1 + c_2x_2 + \cdots + c_nx_n & \\
\overset{\boldsymbol{x}}{\textbf{条}}\ \textbf{件} & a_1x_1 + a_2x_2 + \cdots + a_nx_n \leq b & \\
& \left.\begin{array}{r} 0 \leq x_j \leq u_j \\ x_j : \textbf{整数} \end{array}\right\} & j = 1,2,\ldots,n
\end{array}
\right.
$$

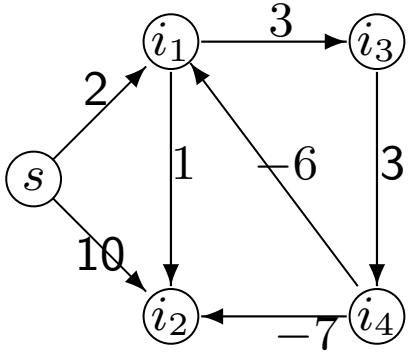
に対する動的計画法を設計せよ．

1. **有向グラフ $G=(V,E)$ と各枝 $(i,j)\in E$ の長さ c_{ij} によって定まるネットワーク上で特定の頂点 s から他のすべての頂点への最短路を求める最短路問題を考える．各頂点 i に対して，$d_k(i)$ は s から i への枝数 k 以下での最短路の長さを表すとする．d_k に関して，以下の関係が成り立つ．**

$$d_0(s)=0;\quad d_0(i)=+\infty\ (i\in V\setminus\{s\})$$

$$d_{k+1}(i)=\min\{d_k(i),\ \min_{j:(j,i)\in E}\{d_k(j)+c(j,i)\}\}\ (i\in V,\forall k\geq 0)$$

これをベルマン方程式とよぶ．(a) ベルマン方程式が正しいことを説明せよ．(b) n を頂点数とする．$d_n(i)$ を求めれば，s から i への最短路長が求められる．ベルマン方程式を用いて，図の問題例に対して，表に $d_k(i)$ の値を行ごとに埋めていき，s から各頂点への最短路長を求めよ．(c) ベルマン方程式とベルマン–フォード法の距離ラベル d との関係を述べよ．



k	s	i_1	i_2	i_3	i_4
0	0	∞	∞	∞	∞
1	0	2	10	∞	∞
2	0	2			
3					
4					
5					