

## 身近な適用例: 文書整形問題

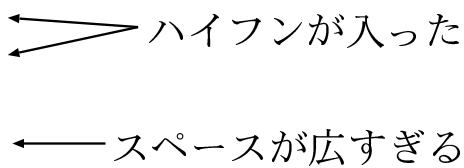
与えられた文章の1つの段落を,

- 右端をそろえて,
- 単語間 (あるいは文字間) のスペースをなるべく理想的な幅にし,
- 単語をハイフンで分割することによる改行をできるだけ少なくなるように

整形する問題を考える (図 2.5) .

応用例: 英字新聞,  $\text{T}_{\text{E}}\text{X}$ ,  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  など.

LAFAYETTE — The Boulder  
County Health Department's La-  
fayette office is reducing serv-  
ices and hours because of recent  
state budget cuts to public  
health services.



ハイフンが入った

スペースが広すぎる

図 2.5 英文レイアウトの例 (Daily Camera, 2002/9/18 より)

以下, 簡単のため, 単語の分割は行わないものとし, 単語間のスペースのみを考える. (以下に紹介するアルゴリズムは, 単語内部での文字間のスペースや単語の分割によるペナルティを考慮する場合にも容易に拡張できる.)

入力: 各単語  $w_i$  の幅  $l_i$  ( $i = 1, 2, \dots, n$ ), 理想的なスペース幅  $s$ , およびテキストの幅  $L$ .

単語  $w_i$  から  $w_j$  ( $1 \leq i \leq j \leq n$ ) を1行にレイアウトするときのスペース幅

$$s(i, j) = \frac{L - \sum_{k=i}^j l_k}{\max\{j - i, 1\}}.$$

そのときのコスト

$$c(i, j) = \begin{cases} +\infty & s(i, j) \leq 0 \text{ のとき} \\ (s - s(i, j))^2 & s(i, j) > 0 \text{ かつ } j < n \text{ のとき} \\ \max\{s - s(i, j), 0\}^2 & s(i, j) > 0 \text{ かつ } j = n \text{ のとき.} \end{cases}$$

出力: 総コスト最小のレイアウト.

## DPの構成

$f^*(i)$ :  $w_i, w_{i+1}, \dots, w_n$  をレイアウトしたときの最小コスト.

$f^*(i)$  を計算する漸化式:

$$f^*(i) = \begin{cases} 0, & i = n + 1 \\ \min_{i \leq j \leq n} \{c(i, j) + f^*(j + 1)\}, & i \leq n. \end{cases}$$

## DPの計算時間:

- ・コスト  $c(i, j)$  を毎回1から計算  $\implies$  全体で  $O(n^3)$  時間.
- ・  $\sum_{k=i}^j l_k = \sum_{k=i}^{j-1} l_k + l_j$  を利用してコスト  $c(i, j)$  を  $O(1)$  時間で計算  $\implies$  全体で  $O(n^2)$  時間.

コメント: コスト  $c(i, j)$  には2乗誤差を用いたが, 簡単に計算できる関数ならば, 同様のDPが構成できる.