Page breaking

Victor Eijkhout

Notes for CS 594 – Fall 2004

Page breaking

- Break vertical list into page-size chunks
- provide headers, footers
- complication: floating material

Theory and practice

- Can be solved with global optimization (dynamic programming?)
- TEX uses 'best fit' like strategy

TEX's page breaking algorithm

Typographical constraints

- No page breaks after section heading
- no break after first / before last line of paragraph
- no break between paragraph and display math
- First and last line of page in same location on each page

No breaks after heading

- Header is box
- follow with `\penalty10000`
- appropriate skip for white space and baseline distance

Input:

```
\section{Head}
Text
```

Output:

```
...\hbox(9.99998+0.0)x345.0, glue set 283.76672fil
    % header
...\write1{\@writefile{toc}{\protect \contentsline % ET CET
...\penalty 10000
...\glue 9.90276 plus 0.86108
...\glue(\parskip) 0.0 plus 1.0
...\glue(\baselineskip) 5.16669
...\hbox(6.83331+0.0)x345.0, glue set 324.99997fil
    % the paragraph
```

```
...\hbox(9.99998+0.0)x345.0, glue set 283.76672fil
    % section Head
...\write1{\@writefile{toc}{\protect % ET CETERA
...\penalty 10000
...\glue 9.90276 plus 0.86108
...\glue(\parskip) 0.0 plus 1.0
...\glue(\baselineskip) 5.66669
...\hbox(8.33331+0.0)x345.0, glue set 286.52779fil
    % subsection head
...\write1{\@writefile{toc}{\protect % ET CETERA
...\penalty 10000
...\penalty 10000
...\glue 6.45831 plus 0.86108
...\glue(\parskip) 0.0 plus 1.0
...\glue(\baselineskip) 5.16669
...\hbox(6.83331+0.0)x345.0, glue set 324.99997fil
    % text
```

Breaks around formula

- No break before display formula:
- `\penalty10000` followed by skip
- Break possible after:
- `\penalty0` followed by skip

Input:

```
paragraph
\[ display \]
more text
```

Output:

```
...\hbox(6.94444+1.94444)x345.0, glue set 285.49988fil
    % the first paragraph
...\penalty 10000
...\glue(\abovedisplayshortskip) 0.0 plus 3.0
...\glue(\baselineskip) 3.11111
...\hbox(6.94444+1.94444)x32.0968, shifted 156.4516
    % the display formula
...\penalty 0
...\glue(\belowdisplayshortskip) 6.0 plus 3.0 minus 3.0
...\glue(\baselineskip) 3.90477
...\hbox(6.15079+0.0)x345.0, glue set 302.47214fil
    % the second paragraph
```

Top line of page

- Insert `\topskip` glue to put baseline in fixed space: glue is `\topskip` parameter minus height of first line
- topskip never negative
- $\Rightarrow$ very big first lines will drop

Input:

```
\begin{document}
Text
\end{document}
```

Output:

```
\vbox(633.0+0.0)x407.0
 % page contents:
.\vbox(617.0+0.0)x345.0, shifted 62.0
..\vbox(12.0+0.0)x345.0, glue set 12.0fil
   % header
..\glue 25.0
..\glue(\lineskip) 0.0
  % the text box :
..\vbox(550.0+0.0)x345.0, glue set 539.94232fil
...\glue(\topskip) 3.16669
...\hbox(6.83331+0.0)x345.0, glue set 309.99997fil
    % one line of text
...\glue 0.0 plus 1.0fil
..\glue(\baselineskip) 23.55556
..\hbox(6.44444+0.0)x345.0, glue set 170.0fil
   % footer: page number
```

No break first/last line of paragraph

- Insertion of appropriate penalties:
- `\clubpenalty`, `\widowpenalty`
- (entirely automatically)

```
..\vbox(550.0+0.0)x345.0, glue set 503.94617fil
...\write-{}
...\glue(\topskip) 3.05556
...\hbox(6.94444+1.94444)x345.0, glue set 0.74356
....\hbox(0.0+0.0)x15.0
....\OT1/cmr/m/n/10 t
    % rest of first line
....\glue(\rightskip) 0.0
...\penalty 150
...\glue(\baselineskip) 3.11111
...\hbox(6.94444+1.94444)x345.0, glue set - 0.18942
    % second line
...\glue(\baselineskip) 3.11111
...\hbox(6.94444+0.0)x345.0, glue set 0.6437
    % third line
...\penalty 150
...\glue(\baselineskip) 5.84921
...\hbox(6.15079+0.0)x345.0, glue set 139.99959fil
    % last line
....\penalty 10000
....\glue(\parfillskip) 0.0 plus 1.0fil
```

Output routine

Best fit strategy

- Material goes onto the vertical list
- Mark when break badness $< 10\,000$ (shrink/stretch or penalties)
- Mark when badness becomes $> 10\,000$ again
- Find best point in between

Output routine

- Vertical material is packed into `\box255`
- Code in `\output` is executed
- Final command: `\shipout`

Really simple output routines

```
\output={\shipout\box255}
```

or

```
\output={
  \setbox255=\vbox{ <header>
                    \box255
                    <footer>
                  }
  \shipout\box255
  }
```

Prevent widow lines

Idea: if page ends with widow line, make bigger by one line

Setup:

```
\newif\ifEnlargedPage \widowpenalty=147
\newdimen\oldvsize \oldvsize=\vsize
```

Output routine: test for widowpenalty,
if so: enlarge and try again

```
\output={
    \ifEnlargedPage <output the page, reset parms>
    \else \ifnum \outputpenalty=\widowpenalty
              \global\EnlargedPagetrue
              \global\advance\vsize\baselineskip
              \unvbox255 \penalty\outputpenalty
          \else  \shipout\box255
    \fi   \fi}
```

Missing bit

```
\ifEnlargedPage \shipout\box255
      \global\LargePagefalse
      \global\vsize=\oldvsize
```

Inserts

- Floating material is called 'insert'
- Different classes:

  `\insert<class number>{ <material> }`
- example: top material, footnotes

Inserts in the output routine

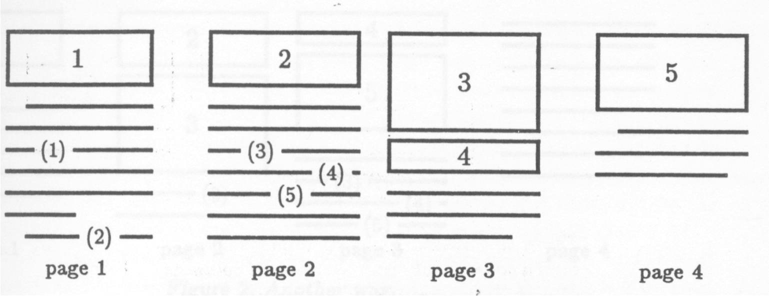
- During output, material available in `\box<class number>`
- `\dimen<class number>` limits amount of material

Page breaking theory

Page breaking vs line breaking

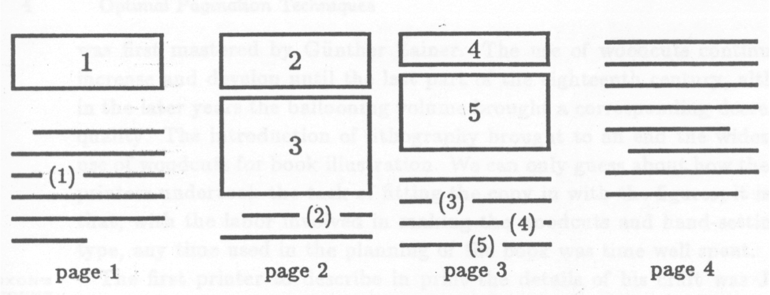
- Similarities:
  break list into chunks
  visual badness criteria
- Difference: floating material
- $\Rightarrow$ minimize distance reference and referent

Closeness of figure and reference



4 out of 5 references not on same page

Closeness of figure and reference



only one off

Strategies

- Moving figures back and forth
- breaking page early or stretching / shrinking

Model problem MQ

Model pagination problem

- Model case: text blocks and figures, each a whole page

$$T = \{t_1, \ldots, t_N\}, \qquad F = \{f_1, \ldots, f_N\}$$

- Reference count function $W : T \times F$

$$W(t_i, f_j) \qquad \text{is number of reference to } f_j \text{ in block } t_i$$

- Wanted: pagination function

$$P : (T \cup F) \rightarrow \{1, 2, \ldots, 2N\}$$

such that

$$\begin{array}{l} P(t_i) < P(t_j) \\ P(f_i) < P(f_j) \end{array} \qquad \forall_{1 \leq i < j \leq N}$$

Multiply reference, quadratic badness

- Goal:
$$S = \sum_{i,j} W(t_i, f_j)(P(t_i) - P(f_j))^2 \leq B$$
minimize quadratic badness
- Assumption: polynomial bound $W(\cdot, \cdot) < q(N)$

NP-completeness

- Polynomial transformation from 'maximum 2-satisfiability' (MAX 2-SAT):
- n binary variables $x_1, \ldots, x_n$
- m clauses $\{u_1 \vee v_1, \ldots, u_m \vee v_m\}$ where $u_i = x_j$ or $u_i = \neg x_j$ for some j
- Given bound K, is there a way of setting the x_i variables such that at least K of the clauses are satisfied? This problem is known to be NP-complete.
- partial bounds: $B = B_1 + B_2 + B_3$

Force adjacency

- Recall $W(t_i, f_j)(P(t_i) - P(f_j))^2$
- $W(t_i, f_i) = b_1$ sufficiently large:
  enforce $|P(t_i) - P(f_i)| = 1$
- to allow this, partial bound $B_1 = Nb_1$
- Define

  $$b_1 = \lceil (B_2 + B_3)/3 \rceil + 1$$

  then, if adjacency violated

  $$S \geq (N - 1)b_1 + 2^2 b_1 = (N + 3)b_1 > B$$

- blocks adjacent: only question which first

Encode truth values

- Number of variables n, clauses m
- Make $N = 2n + 2m$ pages:
  first $4n$ pages encode values of x_i
  4 pages per variable

$$\begin{array}{lllll} 4i-3 & 4i-2 & 4i-1 & 4i & \\ t_{2i-1} & f_{2i-1} & f_{2i} & t_{2i} & \text{if } x_i \text{ is true} \\ f_{2i-1} & t_{2i-1} & t_{2i} & f_{2i} & \text{if } x_i \text{ is false} \end{array}$$

prohibit other configurations

- Set $W(t_{2i-1}, f_{2i}) = W(f_{2i-1}, t_{2i}) = b_2$ large enough.
- recall

$$\begin{array}{lllll} 4i-3 & 4i-2 & 4i-1 & 4i & \\ t_{2i-1} & f_{2i-1} & f_{2i} & t_{2i} & \text{if } x_i \text{ is true} \\ f_{2i-1} & t_{2i-1} & t_{2i} & f_{2i} & \text{if } x_i \text{ is false} \end{array}$$

- These patterns contribute $2 \cdot 2^2 b_2 = 8b_2$,
  illegal possibilities ($t\ f\ t\ f$ and $f\ t\ f\ t$): $(1^2 + 3^2)b_2 = 10b_2$.
- Allow $B_2 = 4b_2 \sum (i-j)^2$ where i, j range over $W(t_i, f_j) = b_2$
- $b2 = \ldots$

Encode clauses

- $4m$ remaining pages:

$4n+4j-3$	$4n+4j-2$	$4n+4j-1$	$4n+4j$	
$t_{2n+2j-1}$	$f_{2n+2j-1}$			if u_j is true
$f_{2n+2j-1}$	$t_{2n+2j-1}$			if u_j is false
		t_{2n+2j}	f_{2n+2j}	if v_j is true
		f_{2n+2j}	t_{2n+2j}	if v_j is false

- Ensure correspondence x_i and u_j:

$$W(t_{2n+2j-1}, f_{2i}) = W(t_{2i} = f_{2n+2j-1}) = b_3 \quad \text{if } u_j = x_i$$

- Let d twice distance between subscripts; in this case $d = (2n + 2j - 2i)$
- correct terms contribute $2d^2 b_2$
- mismatch: t and f reversed, so contribution $\left((d-1)^2 + (d+1)^2\right) = 2(d^2+1)b_2$.

truth values

- Recall

$$\begin{array}{lllll} 4n+4j-3 & 4n+4j-2 & 4n+4j-1 & 4n+4j & \\ t_{2n+2j-1} & f_{2n+2j-1} & & & \text{if } u_j \text{ is true} \\ f_{2n+2j-1} & t_{2n+2j-1} & & & \text{if } u_j \text{ is false} \\ & & t_{2n+2j} & f_{2n+2j} & \text{if } v_j \text{ is true} \\ & & f_{2n+2j} & t_{2n+2j} & \text{if } v_j \text{ is false} \end{array}$$

- Only false clause:

$$f_{2n+2j-1} \quad t_{2n+2j-1} \quad f_{2n+2j} \quad t_{2n+2j}$$

truth values

- Recall

$$\begin{array}{lllll} 4n+4j-3 & 4n+4j-2 & 4n+4j-1 & 4n+4j & \\ t_{2n+2j-1} & f_{2n+2j-1} & & & \text{if } u_j \text{ is true} \\ f_{2n+2j-1} & t_{2n+2j-1} & & & \text{if } u_j \text{ is false} \\ & & t_{2n+2j} & f_{2n+2j} & \text{if } v_j \text{ is true} \\ & & f_{2n+2j} & t_{2n+2j} & \text{if } v_j \text{ is false} \end{array}$$

- Only false clause:

$$f_{2n+2j-1} \quad t_{2n+2j-1} \quad f_{2n+2j} \quad t_{2n+2j}$$

- Here f_{2n+2j_1} and t_{2n+2j} furthest apart: define

$$W(t_{2n+2j}, f_{2n+2j_1}) = 5, \qquad W(t_{2n+2j-1}, f_{2n+2j}) = 3.$$

- Badness 32 for true cases, and 48 for false. Allow $B_3 = 48(m-K) + 32K$.

NP-completeness

- Polynomial transformation
- K Clauses satisfied iff pagination within bound B

Model problem ML

Linear badness

- Text blocks and figures as before

$$T = \{t_1, \ldots, t_N\}, \qquad F = \{f_1, \ldots, f_M\}$$

- Reference counter $W : T \times F$
- Asked: page mapping $P : (T \cup F) \to \{1, 2, \ldots, M + N\}$

$$\begin{array}{c} P(t_i) < P(t_j) \\ P(f_i) < P(f_j) \end{array} \qquad \forall_{1 \leq i \leq N, 1 \leq j \leq M}$$

and so that

$$S = \sum_{i,j} W(t_i, f_j)\, |P(t_i) - P(f_j)| \leq B$$

- Note linear badness

Dynamic programming solution

- Subproblem: B_{ij} is badness of placing i text blocks and j figures on $i+j$ pages
- Problem: 'dangling' references (t_r, f_s) with $r > i, s \leq j$ or $r \leq i, s > j$
- Define $R_{i,j}$ as the number of dangling references

Badness of a reference

- No problem for references within the $i+j$ block
- Badness of dangling reference: only part to the boundary:

$$B_{ij} = B_{ij}^{(1)} + B_{ij}^{(2)}$$

where

$$B_{ij}^{(1)} = \sum_{\substack{r\leq i\\ s\leq j}} W(t_r, f_s)\,|P(t_r) - P(f_s)|$$

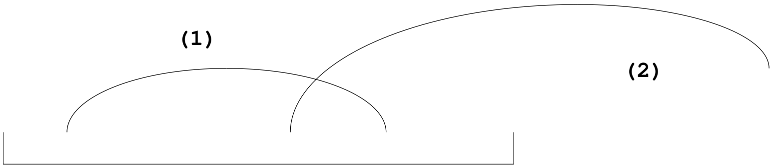
and

$$B_{ij}^{(2)} = \sum_{\substack{r>i\\ s\leq j}} W(t_r, f_s)\,\ell(i,j;r,s) + \sum_{\substack{r\leq i\\ s>j}} W(t_r, f_s)\,\ell(i,j;r,s)$$

where

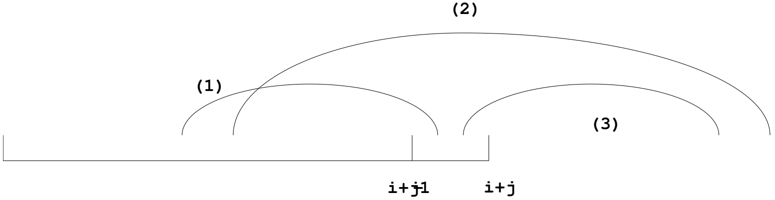
$$\ell(i,j;r,s) = \begin{cases} i+j-P(f_s) & \text{if } r>i \\ i+j-P(t_r) & \text{if } s>j \end{cases}$$

$i + j - 1$ blocks placed



Resolved and dangling references of a block of pages

Add block t_i



(1) Unresolved references from t_i:
contribution from $B^{(2)}$ to $B^{(1)}$

(2) Stay unresolved: $B^{(2)}$ increased by
$\left(\sum_{r\leq i-1,s>j}+\sum_{r>i,s>j}\right)W(t_r,f_s)$.

(3) New unresolved: contribution $\sum_{r=i,s>j} W(t_r,f_s)$ to $B^{(2)}$

Conclusion: $B_{ij}=B_{i-1,j}+R_{ij}$

B_{ij} recurrence

- Shown for t_i, (same for figure f_j)

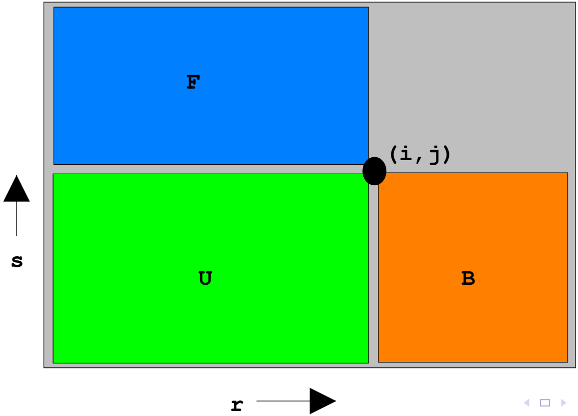
$$B_{ij} = \min(B_{i-1,j}, B_{i,j-1}) + R_{ij}.$$

Dangling references from $i + j$

Forward reference: $F_{ij} = \sum_{\substack{1 \leq r \leq i \\ j < s \leq M}} W(t_r, f_s)$

Backward reference: $B_{ij} = \sum_{\substack{i < r \leq N \\ 1 \leq s \leq j}} W(t_r, f_s)$

which makes $R_{ij} = F_{ij} + B_{ij}$.



Calculation

- Tabulate
$$U_{ij} = \sum_{\substack{1\leq r\leq i\\ 1\leq s\leq j}} W(t_r, f_s)$$
in time $O(NM)$
- Then
$$R_{ij} = U_{iM} + U_{Nj} - 2U_{ij}$$

Conclusion

- Problems MQ and ML are structurally the same
- Only difference badness function
- $\Rightarrow$ difference between NP-complete and polynomial