PAGINATION

BY

BHARATH SUBRAMANYAM
INTRODUCTION

- Pagination is a process of breaking up a document into suitable pages for printing.
- For simple text like a novel without any pictures or stylistic considerations, pagination is trivial. Just break the page when it is full.
- However, when pictures, tables, and footnotes are involved and when you have stylistic considerations, pagination becomes a complex problem.
A BADLY PAGINATED BOOK
MOTIVATION

1
_________
_________
_________
_________
_________
(1)

2
_________
_________
_________
_________
_________
(2)

3
_________
_________
_________
_________
_________
(3)

4
_________
_________
_________
_________
_________
(3)

5
_________
_________
_________
_________
_________
(4)

(5)
GOALS OF PAGINATION

• Sequence – Pagination should maintain the order of various objects. The order of text, the order of figures, the order of tables etc. should be maintained. For example in the figure stream, Figure 1 can not come after Figure 2.

• Widows and Orphans – A widow is the last line of a paragraph on the first line of a page. An orphan is the first line of a paragraph on the last line of a page. Both widows and orphans are undesirable
GOALS OF PAGINATION

Orphan

Widow
GOALS OF PAGINATION

• Free space – A page should have as much content as possible. This not only looks good but also is economical.

• Floating objects – Floating objects should be placed as close as possible to their references.
ONE PASS GREEDY

• The one pass greedy algorithm tries to fit in as much content in a page as possible and then breaks the page.

• Once the page is broken it can not be modified by the algorithm.
ONE PASS GREEDY

- Orphans - This algorithm can easily avoid orphan lines by moving it to the next page and allowing the last line of the previous page to be blank.

- Widow - However, avoiding widows is complex. You have to determine one line before the last line of the paragraph if it would end up in the next page if you want the one pass algorithm to work. A multi-pass algorithm could be used in such a case where the widow is recognized in the first pass and the page is broken in the second pass.
ONE PASS GREEDY

• Floating Objects – If the algorithm encounters a reference to a floating object (figure, table or a footnote) it adds it to a queue and tries to empty the objects in the queue to the page as soon as possible.

• Footnotes – If a footnote has been referenced in a page, the greedy algorithm first reads enough content to end the current line. Then it tries to place the footnote at the bottom of that page. However, sometimes it may not be possible to fit the entire footnote. In that case, it tries to fit as much content of the footnote possible to that page and keeps the remaining in the queue to be added to the next page.
ONE PASS GREEDY

• Figures – When the algorithm encounters a reference to a figure, it first reads enough content to end the current line. Then it tries to fit the figure to the current page. If it is not possible due to lack of space, it adds the figure to a queue and tries to fit it to the next page.

• Tables – If a table is non splittable, then it is treated as a figure. If the table is non splittable, then the algorithm prints what it can on the current page and prints the remaining part of the table on the next page.
ONE PASS GREEDY - ADVANTAGES

• Requires less memory. Only the text of the current page needs to be stored in the main memory. Very useful during the days when the size of the main memory was small.

• The algorithm is stable since no change can affect the output of the text before the change.
ONE PASS GREEDY - DISADVANTAGES

• The position of the objects in the pages may not always be optimal.

• A floater could end up far away from the reference due to overbooking by other floaters because many references are near each other.
APPLICATIONS WHICH USE ONE PASS GREEDY

- Ditroff
- MS Word
- FrameMaker
HOW DOES TEX DO IT

- TEX breaks lists of lines into pages by computing badness ratings and penalties. But pages are made up one at a time and removed from TEX’s memory; there is no looking ahead to see how one page break will affect the next one. TEX uses a special method to find the optimum breakpoints for the lines in an entire paragraph, but it doesn’t attempt to find the optimum breakpoints for the pages in an entire document. The computer doesn’t have enough high-speed memory capacity to remember the contents of several pages, so TEX simply chooses each page break as best it can, by a process of “local” rather than “global” optimisation. TEX uses a two pass system where it gathers the information during the first pass and does the typesetting using dynamic programming during the second pass.
MULTIPASS ALGORITHM

- To overcome the disadvantage of placing floating objects in a non-optimal manner, all the content of the document could be read at once and dynamic programming could be used to find the optimal page breaks.
MEASURING QUALITY

- To use dynamic programming we would like to first measure the quality of the pagination and try to minimize it.

- Plass (student of Knuth) suggested to calculate the difference between the objects and all their references to calculate the badness function.
BADNESS FUNCTIONS

- $Quad(P) = \sum_{i=1}^{n} \sum_{j=1}^{r(i)} \left( P(f_i) - P(R_j(f_i)) \right)^2$

- $Linear(P) = \sum_{i=1}^{n} \sum_{j=1}^{r(i)} \left| P(f_i) - P(R_j(f_i)) \right|$

- $P(x)$ is the page number of object $x$

- $R(x)$ is the reference to object $x$

- $f$ is a figure or an object
BADNESS FUNCTIONS

• Plass proved that for finding the pagination of any arbitrary input which minimizes the quadratic badness function is NP Complete. The proof involves reducing the 2 satisfiability problem to the pagination problem.

• Therefore, the linear badness function preferred.

• If there are any dangling references (a reference without an object or an object without a reference), their count is added to the badness function.
DYNAMIC PROGRAMMING

\[ B_{jk} = \min_{0 \leq i \leq j} \{ B_{i, k-1} + \beta_{ijk} \} \]

- \( B_{jk} \) is the total badness when \( j \) lines have been paginated with \( k \) pages.
- \( \beta_{ijk} \) is the badness when lines from \( i+1 \) to \( j \) is put on the \( k^{th} \) page.
HIGH LEVEL CODE

(create a break node)
i = 0
while i < totalNumberOfTextLines
    for j=0 to totalNumberOfObjects
        (place object j after current line)
        (find the badness)
        (add a breakpoint at the optimal place)
        (deactivate unwanted breakpoints)
ADVANTAGES/DISADVANTAGES OF MULTIPASS

• Advantages
  • The floating objects are placed at a more optimal location than the greedy algorithm.

• Disadvantages
  • Objects can appear before references
  • Changes in the current page may cause a reorganization in the previous page.
THANK YOU