BWT Tunnel Planning is hard but manageable

Uwe Baier, Kadir Dede

Institute of Theoretical Computer Science
Ulm University

Ulm
March 15, 2019

In this talk

From BWT to graphs to tunneling to TBWT

Tunnel planning is hard...

... but manageable

Experimental results

Conclusion

BWT [Burrows and Wheeler, 1994]

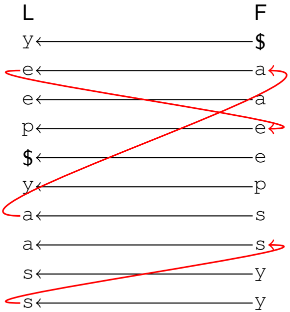
"The BWT L *is a string generated by concatenating all cyclic preceding characters of the lexicographically sorted suffixes of a string S."*

BWT generation of $S = \texttt{easypeasy\$}$

prec. char.	suffixes		L	sorted suffixes
y	$		y	$
s	y$		e	asy$
a	sy$		e	asypeasy$
e	asy$	sort →	p	easy$
p	easy$		$	easypeasy$
y	peasy$		y	peasy$
s	ypeasy$		a	sy$
a	sypeasy$		a	sypeasy$
e	asypeasy$		s	y$
$	easypeasy$		s	ypeasy$

BWT - backward step

- F - column: obtained by sorting characters in L
- *k*-th occurence of character *c* in L corresponds to *k*-th occurence of character *c* in F (LF-mapping)

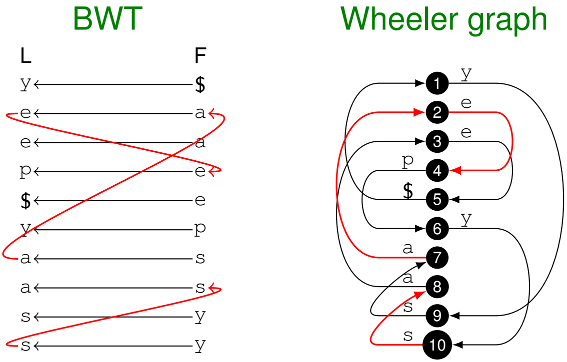


- following LF-mapping and collecting characters of L during walk yields reverse of original string

Wheeler Graphs [Gagie et al., 2017]

For a (simple) BWT, definition simplifies as follows:

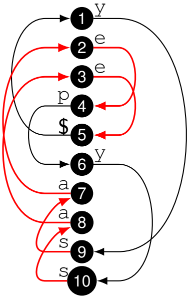
- nodes are integers from 1 to n
- edges are arrows from node i to node LF[i] with label L[i]



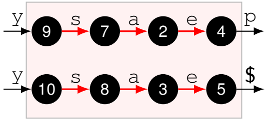
Tunneling [Baier, 2018]

- parallel equally labeled paths (called a Block) can be contracted to a “tunnel”
- results in another Wheeler graph [Alanko et al., 2019]

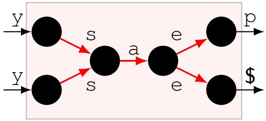
Wheeler graph



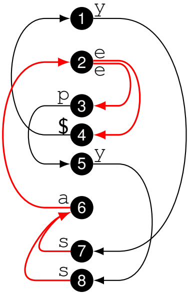
Block



Tunnel



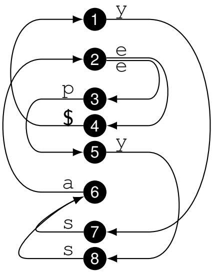
Tunneled graph



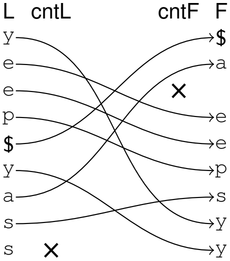
Tunneled BWT

- mark start and end of a tunnel in 2 bitvectors cntL and cntF except for uppermost entries
- Tunneled BWT: L and bitvectors cntL and cntF
- F-column: sort unmarked characters in L to “free places” in F
- *k*-th occurence of unmarked character *c* in L corresponds to *k*-th occurence of unmarked character *c* in F
- use uppermost row of a tunnel for all rows of original block

Tunneled graph

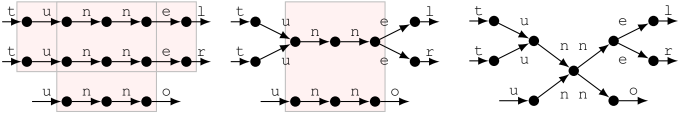


Tunneled BWT



On block choice strategies

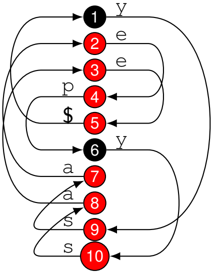
- every tunneled block achieves a benefit but causes costs
- tunneled blocks are allowed to overlap each other



- What complexity is needed to do a good block choice?

WHEELER GRAPH BLOCK COVER PROBLEM

Given a Wheeler graph G and a positive integer k, is there a collection of k or fewer blocks such that each node belonging to any block in G also belongs to a block in the collection?

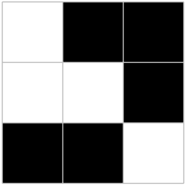


Block choice complexity

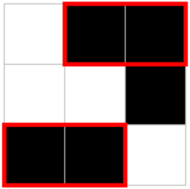
Rectilinear Picture Rectangle Cover Problem [Garey and Johnson, 1990]

Given a $n \times n$ matrix M of 0's and 1's and a positive integer k, is there a collection of k or fewer rectangles that covers precisely the 1's in M?

problem instance



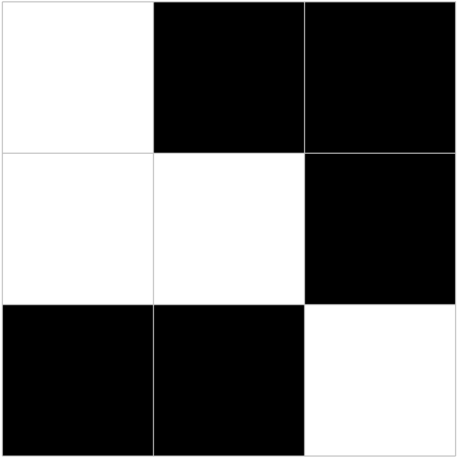
unsolvable for $k = 2$



- Problem is NP-complete if rectangles are allowed to overlap [Masek, 1978]
- Problem can be reduced to Wheeler graph Block Cover

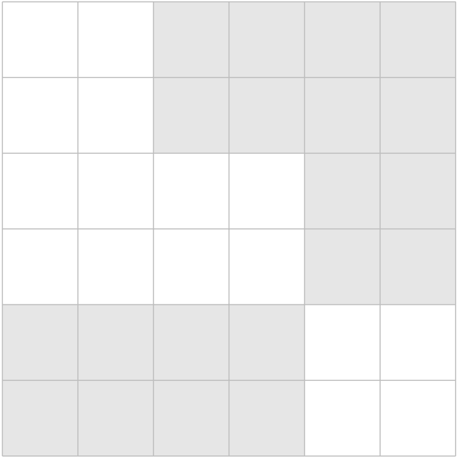
Reduction

Starting point: problem instance (binary picture)



Reduction

1. Split each pixel in 2×2 - minipixels



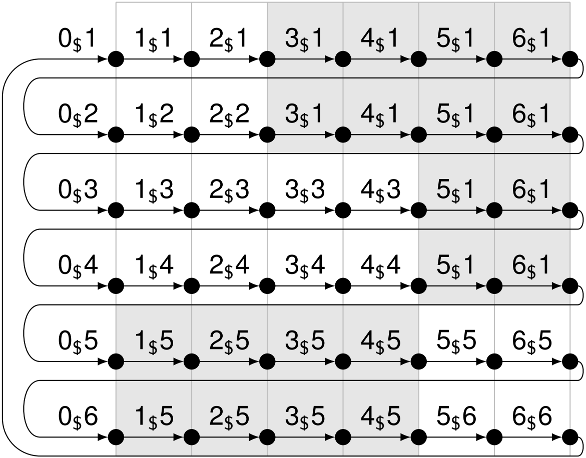
Reduction

2. Label each pixel with its coordinate;
   if overlying minipixel is black, copy coordinate from overlying minipixel

$0_{\$}1$	$1_{\$}1$	$2_{\$}1$	$3_{\$}1$	$4_{\$}1$	$5_{\$}1$	$6_{\$}1$
$0_{\$}2$	$1_{\$}2$	$2_{\$}2$	$3_{\$}1$	$4_{\$}1$	$5_{\$}1$	$6_{\$}1$
$0_{\$}3$	$1_{\$}3$	$2_{\$}3$	$3_{\$}3$	$4_{\$}3$	$5_{\$}1$	$6_{\$}1$
$0_{\$}4$	$1_{\$}4$	$2_{\$}4$	$3_{\$}4$	$4_{\$}4$	$5_{\$}1$	$6_{\$}1$
$0_{\$}5$	$1_{\$}5$	$2_{\$}5$	$3_{\$}5$	$4_{\$}5$	$5_{\$}5$	$6_{\$}5$
$0_{\$}6$	$1_{\$}5$	$2_{\$}5$	$3_{\$}5$	$4_{\$}5$	$5_{\$}6$	$6_{\$}6$

Reduction

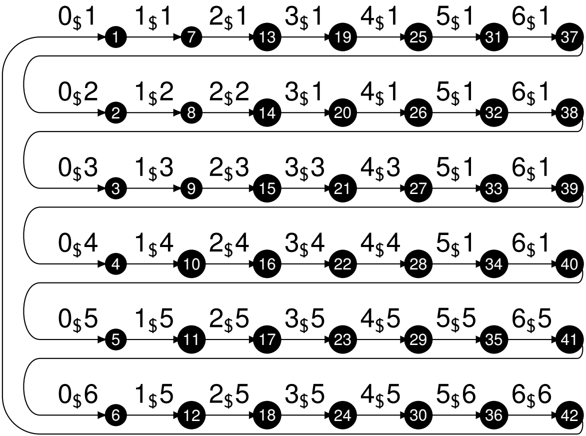
3. Place nodes between minipixels; write edges between each node and its right neighbor node; write edges between rightmost nodes and leftmost nodes of cyclically next row



Reduction

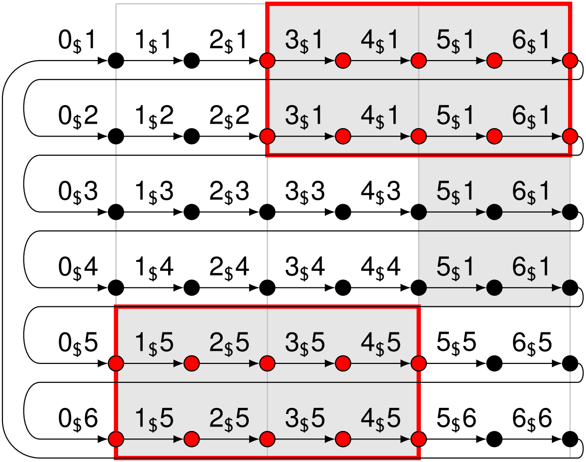
Graph is a Wheeler graph if alphabet order is as follows:

$$i_{\$}j \prec k_{\$}l \quad \Leftrightarrow \quad i < k \text{ or } i = k \text{ and } j < l$$



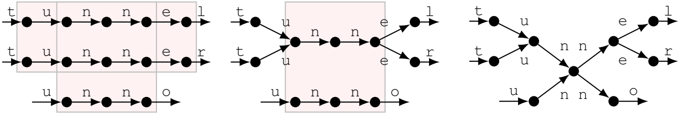
Reduction

k rectangles cover precisely all black pixels ⇔
k blocks cover precisely all nodes of tunnelable blocks



Remarks on complexity

- in practice not every Block is worth tunneling
  ⇒ cover at least *n* nodes instead of all
  ⇒ WHEELER GRAPH BLOCK COVERAGE PROBLEM is NP-hard
- RECTILINEAR PICTURE RECTANGLE COVER is MaxSNP - hard [Berman and DasGupta, 1997]
  ⇒ No PTAS for WHEELER GRAPH BLOCK COVER exists
- Reduction also works the other way; RECTILINEAR PICTURE RECTANGLE COVER is in P if rectangles are not allowed to overlap [Ohtsuki, 1982]
  ⇒ non-overlapping WHEELER GRAPH BLOCK COVER is in P
- open problem: Is “cross-overlay” WHEELER GRAPH BLOCK COVERAGE also NP-hard?



Block Restrictions

Consider only length-maximal blocks with same height as the run they start and end in $\rightarrow$ any block collection can be tunneled

Greedy block choice strategy [Baier, 2018]

- chooses blocks in a greedy fashion, depending on their benefit
- considers block collisions and updates benefits of not-yet chosen blocks
- final choice: blocks whose benefit overcomes their costs

Pro

- restricted block set is a matroid
- optimal without run-length encoding of BWT

Con

- run-length encoding of BWT is crucial for compression
- complicated to implement (requires collision graph)
- resource-intensive

A simple Block choice heuristic

Idea: ignore collisions and tunnel blocks whose benefit overcomes the tunnel costs

Tunneling cost model [Baier, 2018]

n length of run-length encoded BWT

r #runs in BWT

$r_{h>1}$ #runs with height greater 1

tc_B #characters removed from rle-encoded BWT by tunneling B

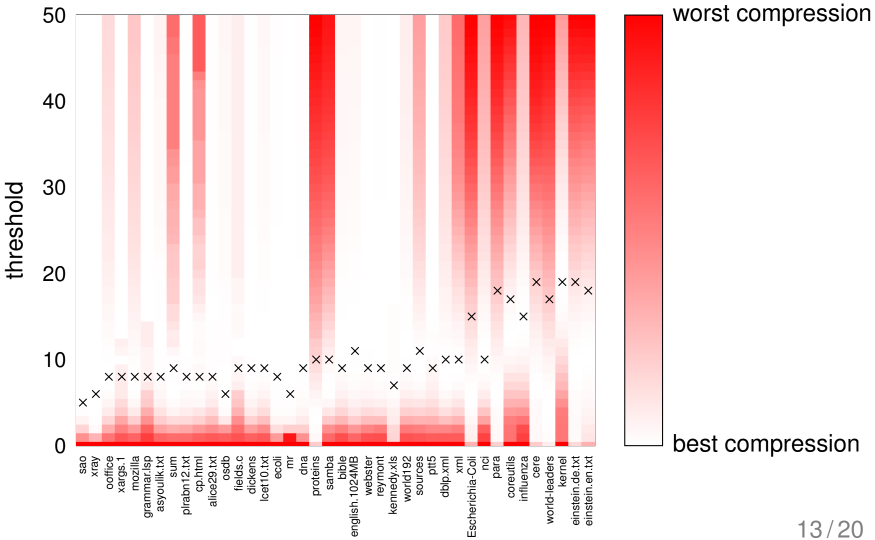
- $benefit_B \approx tc_B \cdot \left(1 + \log_2\left(\frac{n}{n-r}\right)\right)$ bits
- $cost_B \approx 6 + 4 \cdot \log_2\left(\log_2\left(\frac{r_{h>1}}{2}\right)\right)$ bits

Approach: tunnel a block B if $benefit_B \geq cost_B$, or equivalently

$$tc_B \geq threshold \text{ with } threshold := \left\lceil \frac{6 + 4 \cdot \log_2\left(\log_2\left(\frac{r_{h>1}}{2}\right)\right)}{1 + \log_2\left(\frac{n}{n-r}\right)} \right\rceil .$$

Estimator quality

Heuristic with thresholds 0 – 50 on tunneling-enhanced `bzip2`, threshold estimator is indicated with black crosses



Experiments: Overview

BWT compressors enhanced with tunneling

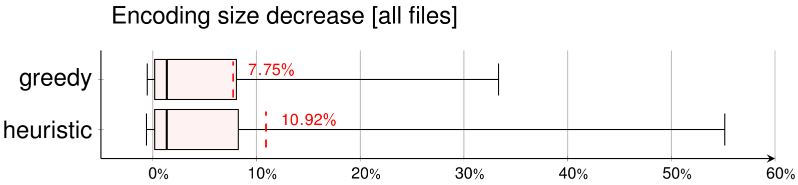
- `bwz`: original scheme by Burrows & Wheeler (≈ `bzip2`)
- `bcm`: one of the best open-source BWT compressors
- `wt`: wavelet tree using hybrid bitvectors

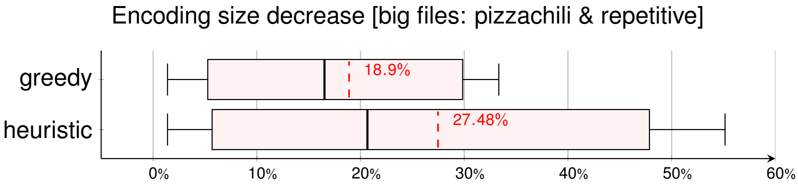
Test Data

Corpus	#Files	Filesizes (MB)
Canterbury	11	0.003 - 1
Large Canterbury	3	2 - 5
Silesia	12	6 - 49
Pizza & Chili	6	54 - 1130
Repetitive	9	45 - 446

Tunneling compression improvements

- Comparison with normal BWT compression
- BWT backend encoder: `bcm`
  (similar for `bwz` and even better for `wt`)





Conclusion

Tunnel planning is hard...

Overlapping block cover and coverage is

- hard to solve (NP-hardness)
- hard to approximate (MaxSNP - hardness)

...but manageable

greedy vs. heuristic strategy on restricted block set

- heuristic achieves better compression
- heuristic performs better
  - 16.5% encoding time speedup
  - 20.8% encoding memory peak decrease

Open problems:

- hardness of “cross-overlay” block coverage
- study of other block set restrictions

Questions

?

References I

Jarno Alanko, Travis Gagie, Gonzalo Navarro, and Louisa Seelbach Benkner.
Tunneling on Wheeler Graphs.
In Proceedings of the 2019 Data Compression Conference, DCC '19, 2019.

Uwe Baier.
On Undetected Redundancy in the Burrows-Wheeler Transform.
In Annual Symposium on Combinatorial Pattern Matching (CPM 2018), CPM '18, pages 3:1–3:15, 2018.

Piotr Berman and Bhaskar DasGupta.
Complexities of Efficient Solutions of Rectilinear Polygon Cover Problems.
Algorithmica, 17(4):331–356, 1997.

References II

Michael Burrows and David J. Wheeler.
A block-sorting lossless data compression algorithm.
Technical Report 124, Digital Equipment Corporation, 1994.

Luca Foschini, Roberto Grossi, Ankur Gupta, and Jeffrey Scott Vitter.
When Indexing Equals Compression: Experiments with Compressing Suffix Arrays and Applications.
ACM Transactions on Algorithms, 2(4):611–639, 2006.

Travis Gagie, Giovanni Manzini, and Jouni Sirén.
Wheeler graphs: A framework for BWT-based data structures.
Theoretical Computer Science, 698:67–78, 2017.

References III

Michael R. Garey and David S. Johnson.
Computers and Intractability; A Guide to the Theory of NP-Completeness.
W. H. Freeman & Co., 1990.

Juha Kärkkainen, Dominik Kempa, and Simon J. Puglisi.
Hybrid Compression of Bitvectors for the FM-Index.
In Proceedings of the 2014 Data Compression Conference, DCC '14, pages 302–311, 2014.

William J. Masek.
Some NP-complete set covering problems, 1978.
unpublished manuscript.

References IV

Ilya Muravyov.
`bcm` File Compressor.
https://github.com/encode84/bcm.
last visited January 2018.

Tatsuo Ohtsuki.
Minimum dissection of rectilinear regions.
In Proceedings 1982 IEEE Symposium on Circuits and Systems, pages 1210–1213. IEEE, 1982.

Julian Seward.
`bzip2` File Compressor.
http://bzip.org/.
last visited January 2018.