A Constant Approximation Algorithm for Sorting Buffers

Jens Svalgaard Kohrt

svalle@imada.sdu.dk

University of Southern Denmark, Denmark

Kirk Pruhs

kirk@cs.pitt.edu

University of Pittsburgh, PA, USA

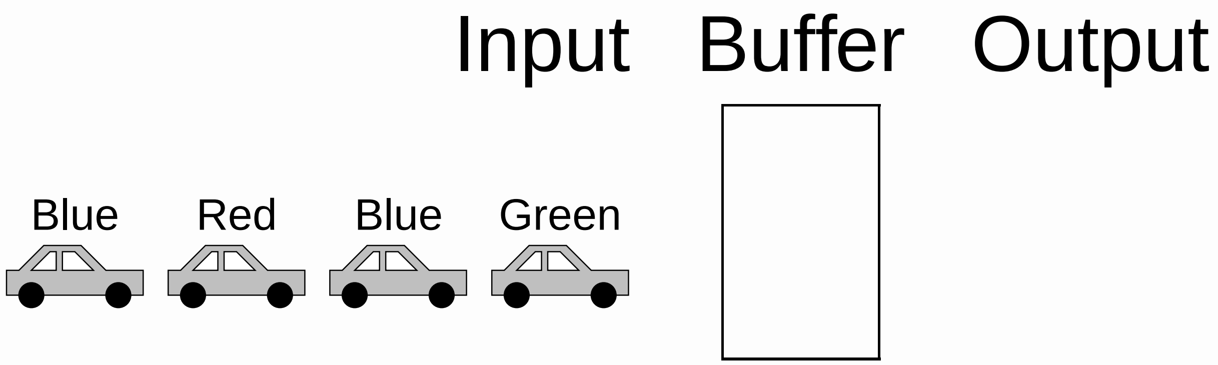
The Sorting Buffers Problem

[Räcke, Sohler, Westerman: ESA 2002]

The Sorting Buffers Problem

Re-order the cars using a small buffer of size k:

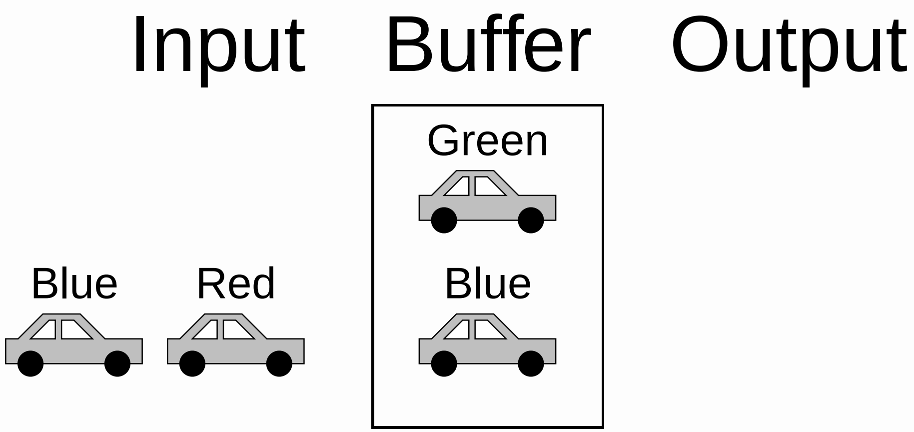
Example with $k = 2$:



The Sorting Buffers Problem

Re-order the cars using a small buffer of size k:

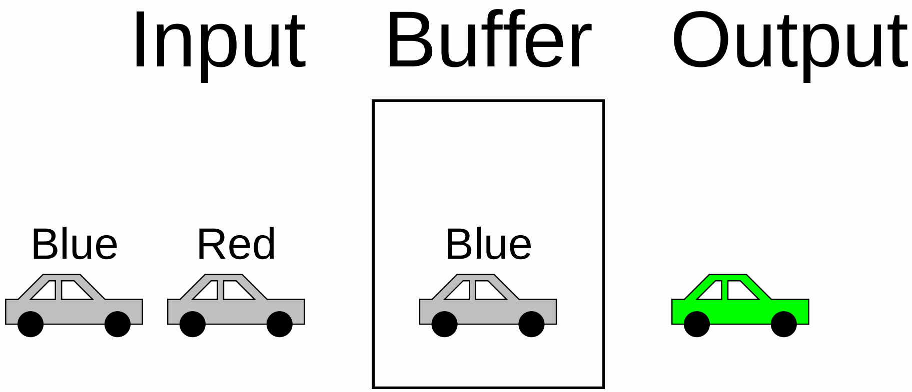
Example with $k = 2$:



The Sorting Buffers Problem

Re-order the cars using a small buffer of size k:

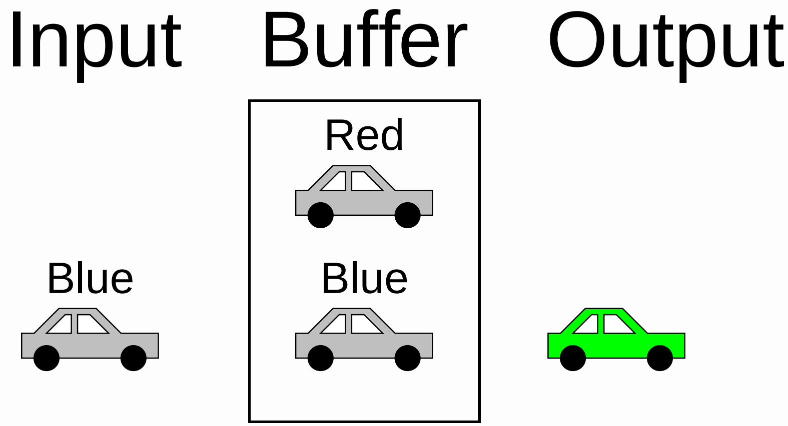
Example with $k = 2$:



The Sorting Buffers Problem

Re-order the cars using a small buffer of size k:

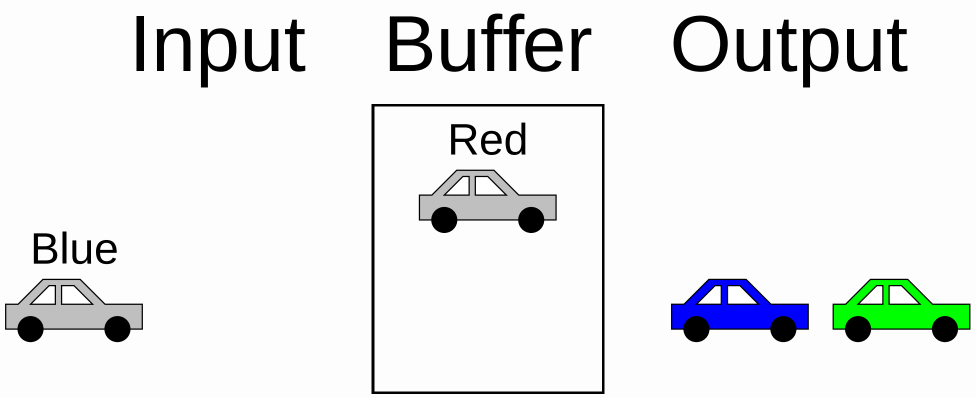
Example with $k = 2$:



The Sorting Buffers Problem

Re-order the cars using a small buffer of size k:

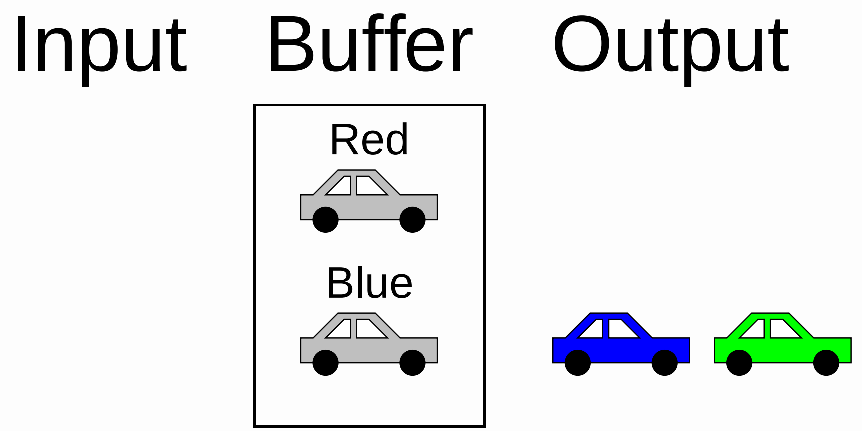
Example with $k = 2$:



The Sorting Buffers Problem

Re-order the cars using a small buffer of size k:

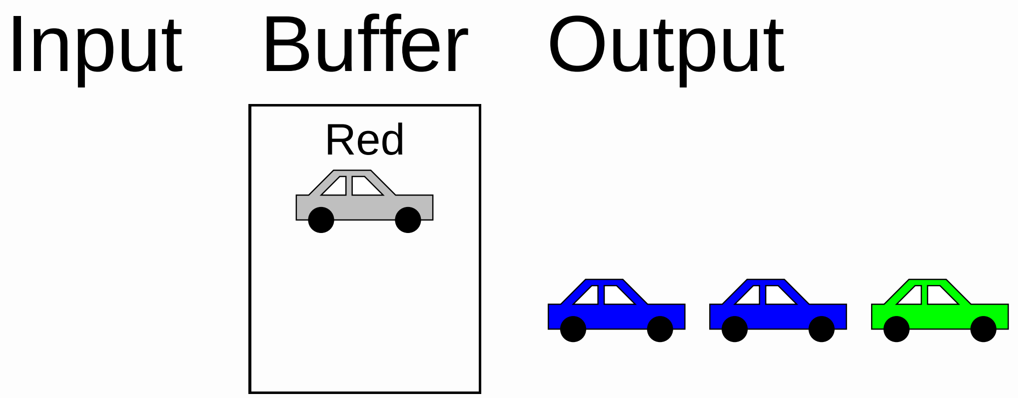
Example with $k = 2$:



The Sorting Buffers Problem

Re-order the cars using a small buffer of size k:

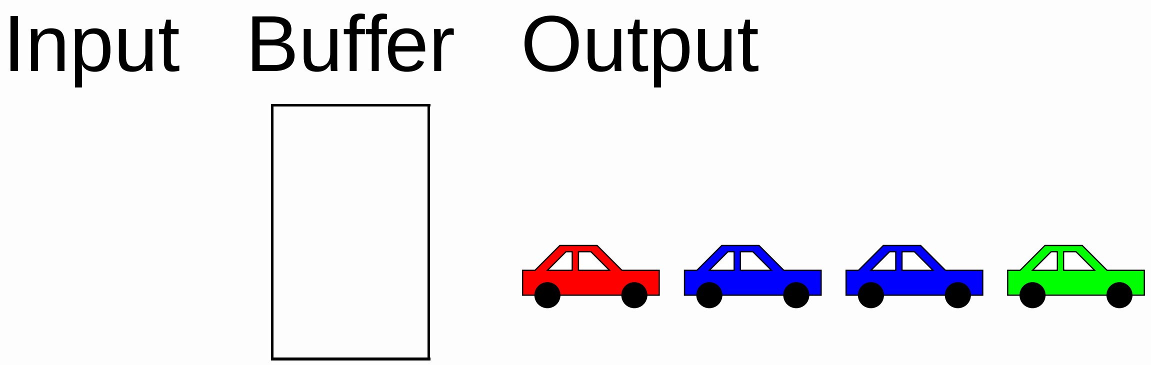
Example with $k = 2$:



The Sorting Buffers Problem

Re-order the cars using a small buffer of size k:

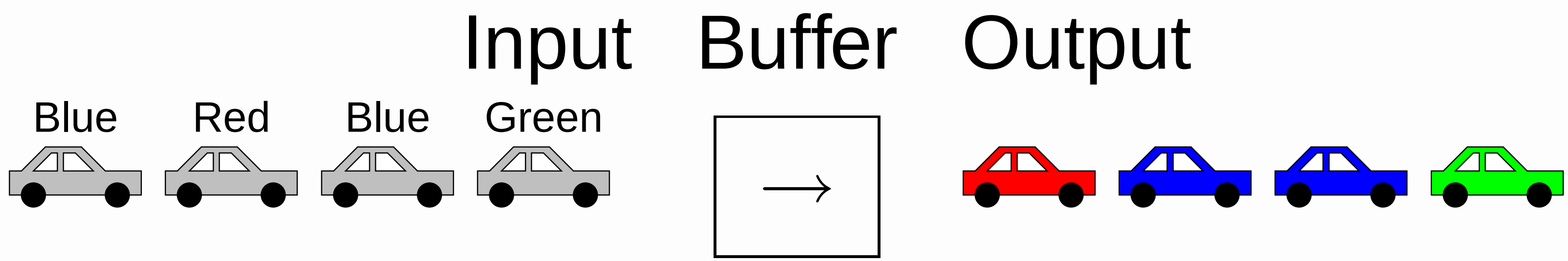
Example with $k = 2$:



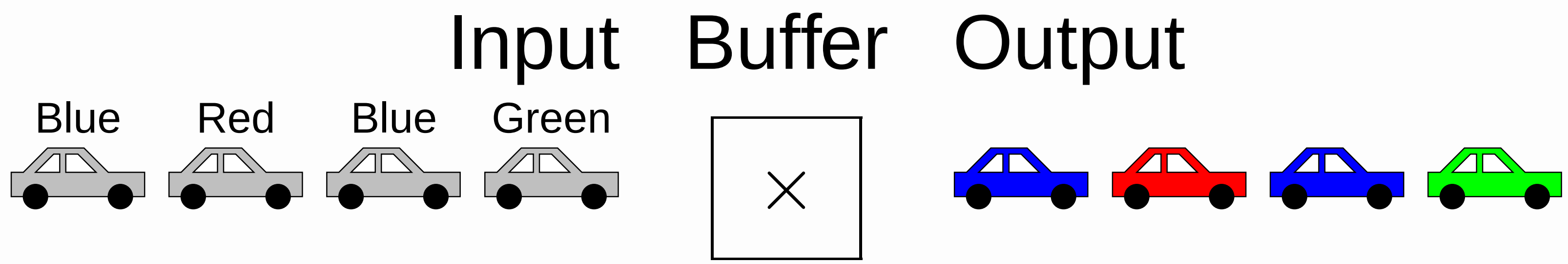
The Sorting Buffers Problem

Re-order the cars using a small buffer of size k:

Example with $k = 2$:



Without buffer:



Previous Work

As an *on-line* minimization problem:

[Räcke, Sohler, Westermann: ESA 2002]

$\mathcal{R}\left(\textit{LCF}\right) \in \Omega(k)$,

$\mathcal{R}\left(\textit{LRU}\right), \mathcal{R}\left(\textit{FIFO}\right) \in \Omega(\sqrt{k})$

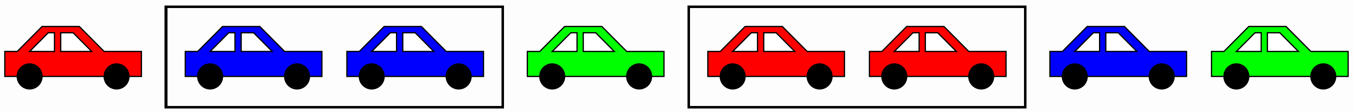
$\mathcal{R}\left(\textit{Bounded-Waste}\right) \in O(\log^2 k)$.

[Sokol: MIT PhD.-thesis 1999]

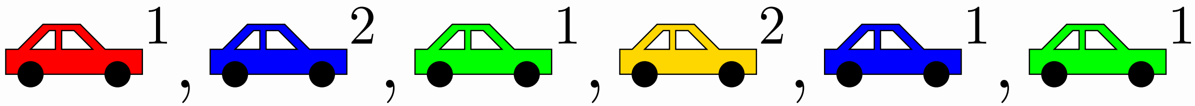
Similar problem: Paint Blocking with two buffers: O(1)-approximation algorithms.

Formal definition

- This can be seen from two sides:
  - maximize number of color savings using buffer
  - minimize number of color changes in output.
- In the input sequence, we can group cars of the same color together:

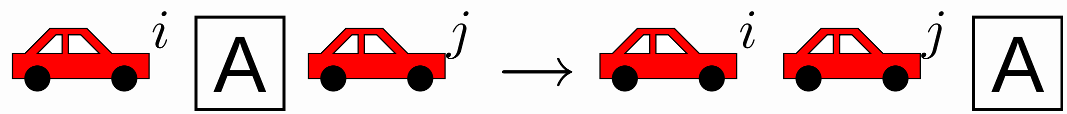


Formal definition

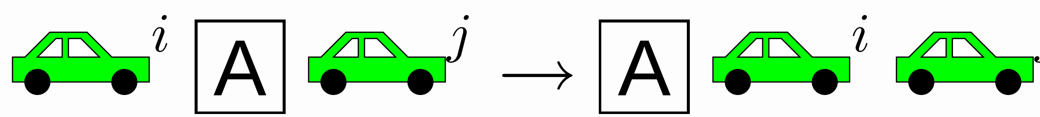
- Input sequence
  $\sigma = \langle$  $\rangle$
- Output sequence is a permutation of σ.
- Buffer of size k.
- *Maximization version:* A savings is gained each time we put two groups of the same color together in the output sequence.

Three classes of savings

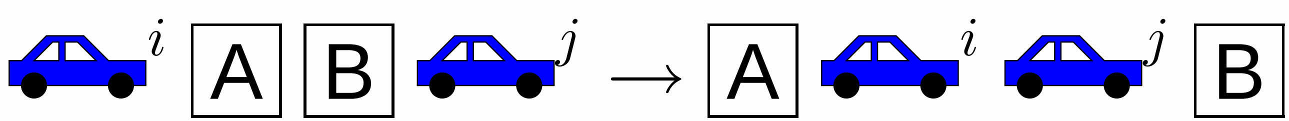
- Move out-saving (MOS)



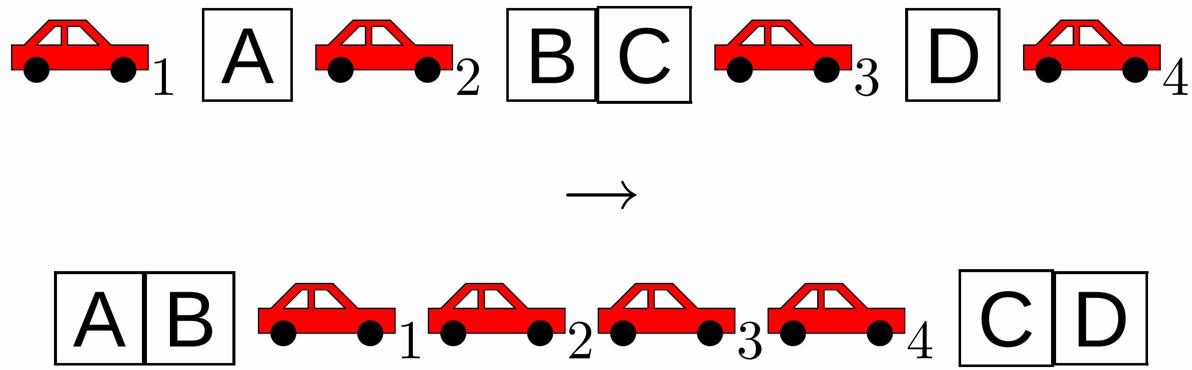
- Move backward-saving (MBS)

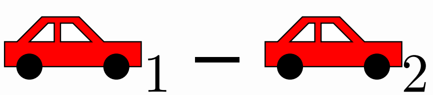
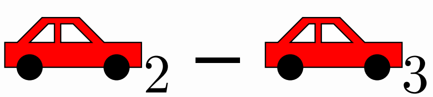
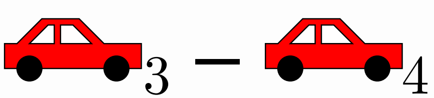


- Move backward and out-saving (MBOS)



Three classes of savings – Example



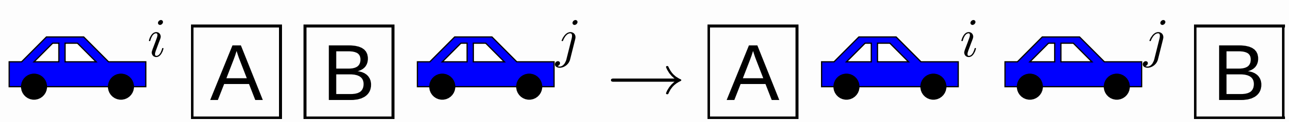
- MBS: 
- MBOS: 
- MOS: 

SBMP-R – a reduced problem

The Reduced Sorting Buffers Maximization Problem SBMP-R.

Theorem 1: We lose an approximation factor of **four** by

- only considering savings involving two groups
- only considering MBS and MOS, i.e., no MBOS:



Extension of the unified algorithm

[Bar-Noy, Bar-Yehuda, Freund, Naor, Schieber: JACM 2001]

A general framework for scheduling problems.

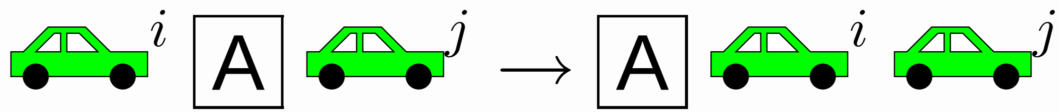
Key element: **Instances** $I = ([b_I, e_I), w_I, p_I, A_I)$.

A_I does not have to be transitive.

Theorem 2: For any $\alpha > 0$, the approximation ratio of the UA is at least

$$\frac{\min\{1, \alpha \max\{w_{\min}, 1 - w_{\max}\}\}}{1 + \alpha}$$

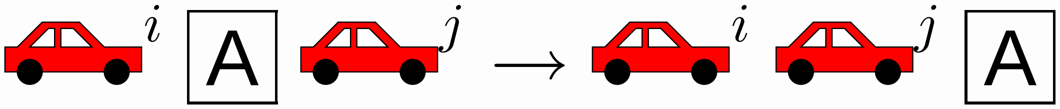
A SBMP-R-MBS approx. algorithm



Theorem 3: There is a $\frac{1}{4}$-approximation algorithm for SBMP-R-MBS running in polynomial time.

This is done using the unified algorithm.

A SBMP-R-MOS algorithm



Theorem 4: There is a polynomial time algorithm solving SBMP-R-MOS.

Solved by transforming the problem into finding a maximum independent set in an interval graph.

Putting the pieces together

- SBMP-R is a $\frac{1}{4}$-approximation of SBMP.
- SBMP-R-MBS has a $\frac{1}{4}$-approximation.
- SBMP-R-MOS can be solved exactly.

Theorem 5: There is an $\frac{1}{20}$-approximation algorithm for SBMP.

Future work

- Improve approximation algorithm
- On-line version
  - Different buffer sizes
- Two-buffer case