Block Rearranging Elements within Matrix Columns to Minimize the Variability of the Row Sums

Kris Boudt

Vrije Universiteit Brussel, Amsterdam

(with: E. Jakobsons, S. Vanduffel)

- Problem: Partition a set of numbers into two subsets S_1 and S_2 such that the difference between the sum of elements in S_1 and the sum of elements in S_2 is minimized.

S = {8,7,6,5,4} ➡ S_1 = { ? ? } S_2 = {? ?? }

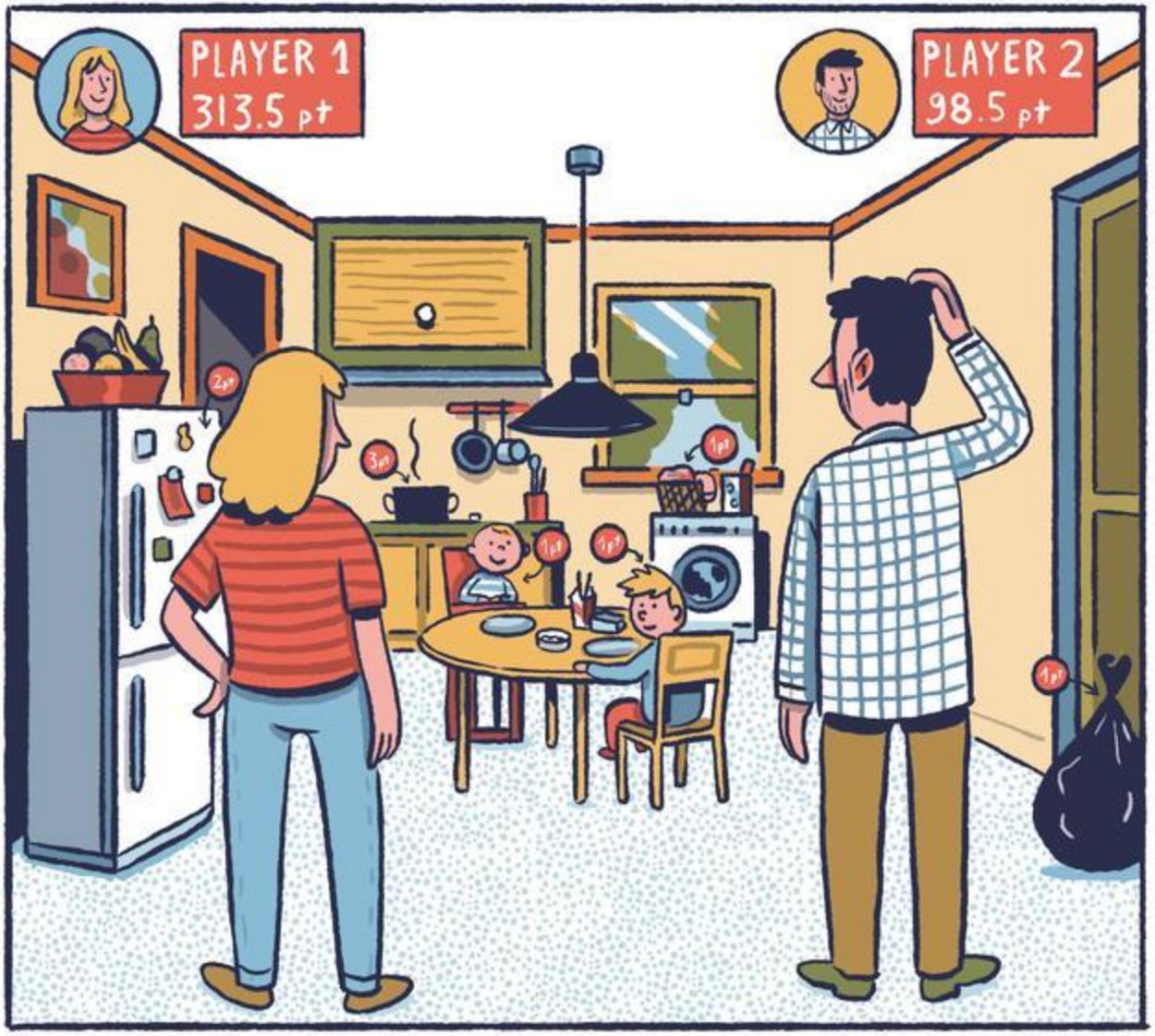
- Solution:

$S = \{8,7,6,5,4\}$ $\xrightarrow{\text{Optimum}}$ $S_1 = \{8,7\}$ $S_2 = \{6,5,4\}$ **Delta= 0**

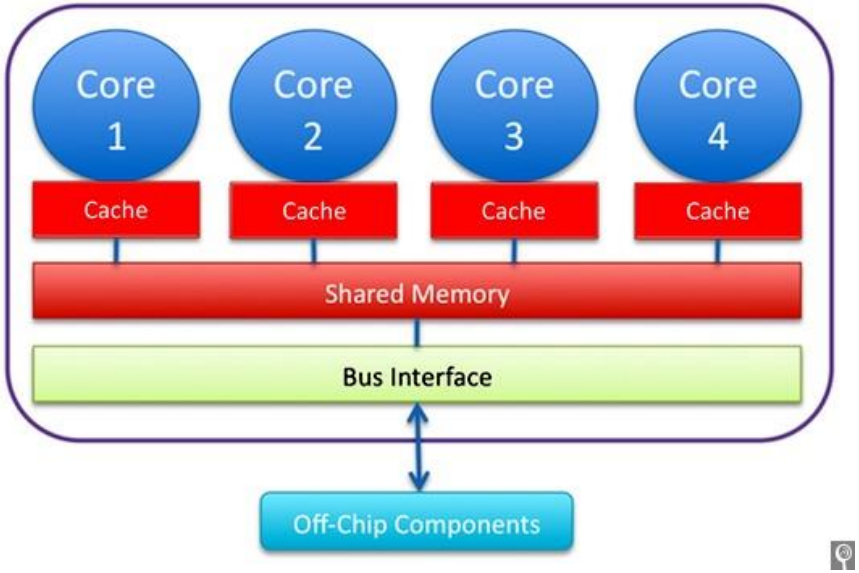
- This so-called partitioning problem is widespread in daily life

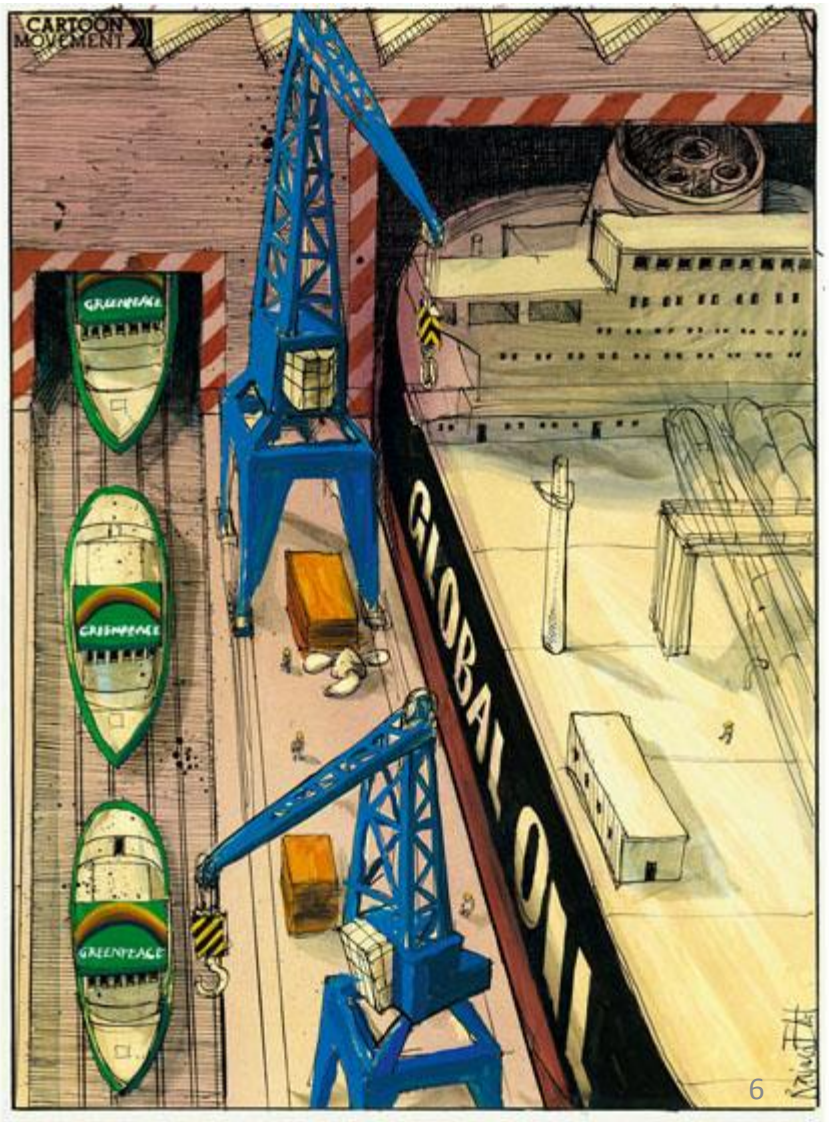
On the playground – teams need to be competitive

At home: “fair” division of labour



At work: efficient distribution of tasks among the available ressources





Heuristic to solve this problem

- **Greedy Algorithm:** iterates through the numbers in descending order, assigning each of them to whichever subset has the smallest sum.

$S = \{8,7,6,5,4\}$ $\xrightarrow{\text{Greedy}}$ $S_1 = \{8,5,4\}$ $S_2 = \{7,6\}$ **Delta= 4**

- Greedy algorithm:
  - Can be relatively far off from the optimum
  - Needs adaptation to work with negative numbers and in case of many groups.

Our solution - treat it as a matrix re-arrangement problem

- If you need to partition d numbers in 2 groups, make a 2xd matrix, with:
  - Row 1 containing the d numbers
  - Rows 2: zeros

  In our example with 2 groups:

$$\begin{pmatrix} 8 & 7 & 6 & 5 & 4 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

- Then rearrange the elements per column such that the rowsums are as close as possible (minimum variability condition)

How to do the rearrangement?

– Under the greedy algorithm: sort the elements of subsequent columns in reverse order of the row sums taken across *all previous* columns

$$\begin{pmatrix} 8 & 7 & 6 & 5 & 4 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \Rightarrow \begin{pmatrix} 8 & 0 & 0 & \mathbf{5} & \mathbf{4} \\ 0 & \mathbf{7} & \mathbf{6} & 0 & 0 \end{pmatrix} \begin{pmatrix} 17 \\ 13 \end{pmatrix}$$

– Under our proposed block rearrangement: Split matrix in two blocks and rearrange the rows of the smallest block such that row sums per block are in reverse order

$$\begin{pmatrix} 8 & 7 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 6 & 5 & 4 \\ 0 & 0 & 0 \end{pmatrix} \Rightarrow \begin{pmatrix} 0 & 0 \\ 8 & 7 \end{pmatrix} \begin{pmatrix} 6 & 5 & 4 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 15 \\ 15 \end{pmatrix}$$

Delta= 0

How to determine the blocks?

- Possibility 1: random block choice
- Our proposal: choose blocks with equal variance
  - Motivation: var(X+Y) = 0 if var(X)=var(Y) and corr(X,Y)= -1

Approximately achieved by the sorting in reverse order (spearman correlation of -1)

  - We find those blocks by using the result that var(X)=var(Y) $\Leftrightarrow$ cov(X,X+Y)=cov(Y,X+Y)

Linear in the decision vector to allocate it to block 1 or block 2

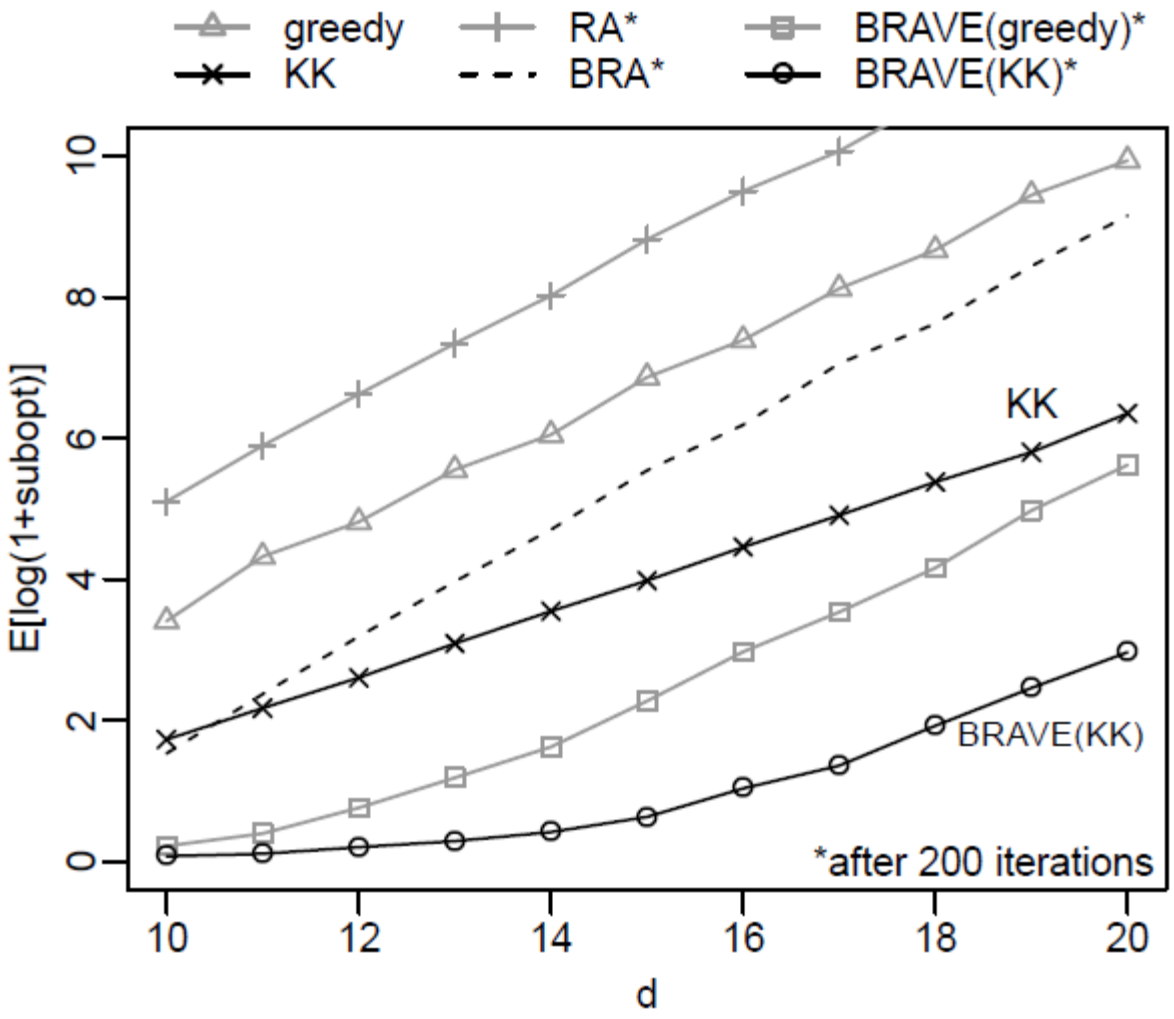
Does it work?

- Simulation study in which we generate N=1000 random vectors of d integers drawn from:

  $$\{1, 2, \ldots, 2^d\}$$

  for d=10,…,20.
- Split in 2 blocks and compute the difference in row sums between the blocks:
  - Best possible solution: Δ^{opt}
  - Heuristic solution: Δ^{heur}
- Average loss function in terms of optimality

$$\frac{1}{N}\sum_{i=1}^{N}\log(1+\Delta_i^{Heur}-\Delta_i^{Opt})$$



Greedy algorithm

Block rearrangements

Want more?

- See our paper on ssrn, at: http://ssrn.com/abstract=2634669
  - Includes:
    - Some theory on the type of variability objective functions that are guaranteed to always decrease when rearranging by sorting one (block of) column(s) in reverse order to the row sums of the remaining columns;
    - Applications :
      - To minimizing bottleneck problems in assembly line scheduling;
      - And assessing model risk in value-at-risk:
        - » Confidence in the marginal distribution;
        - » Distrust the dependence structure → worst case value at risk estimates.
- Preliminary version of the code on R-forge, at: https://r-forge.r-project.org/R/?group_id=2041