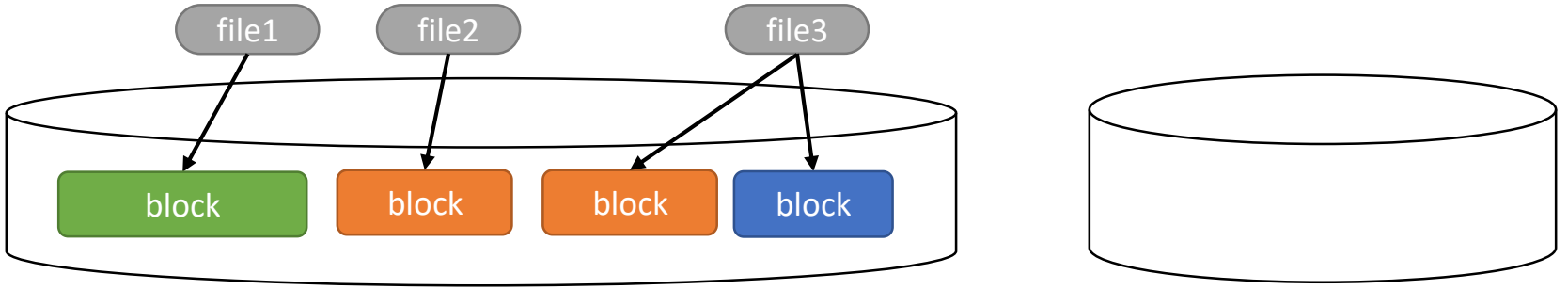
GoSeed: Generating an Optimal Seeding Plan for Deduplicated Storage

Aviv Nachman, Gala Yadgar, Sarai Sheinvald

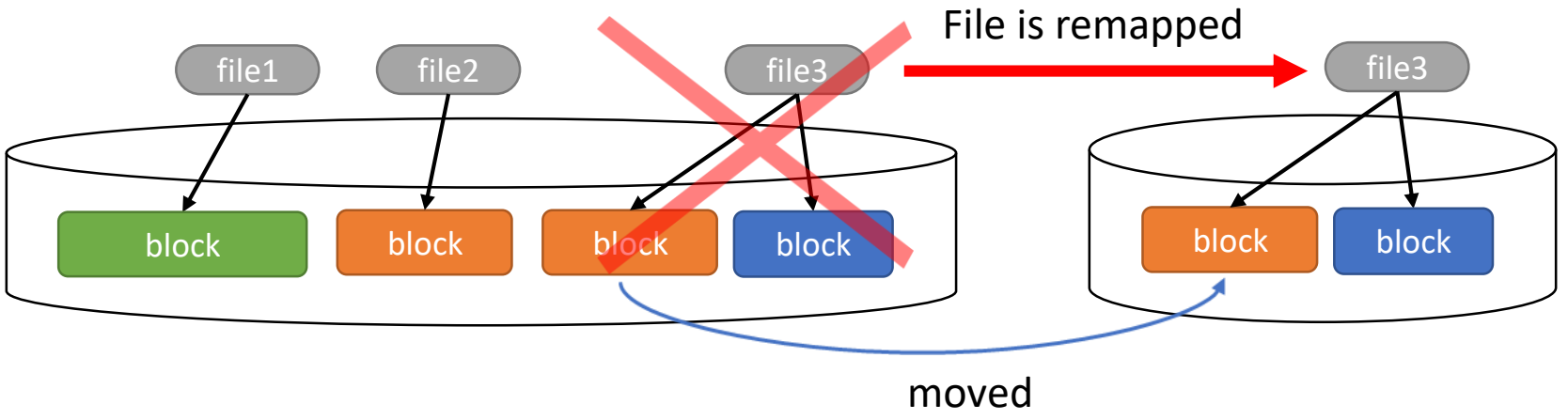
Seeding

- Data migration: transfer files and their blocks between volumes
- Seeding is data migration with empty destination



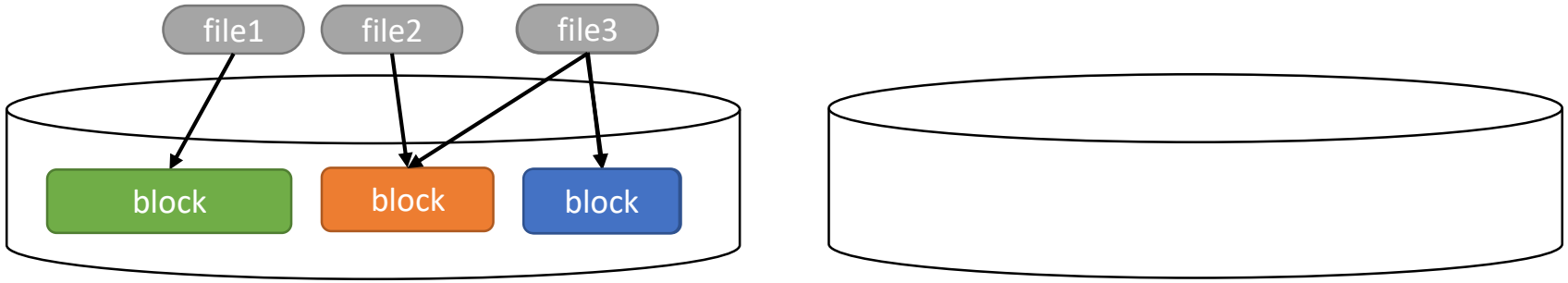
Seeding

- Data migration: transfer files and their blocks between volumes
- Seeding is data migration with empty destination



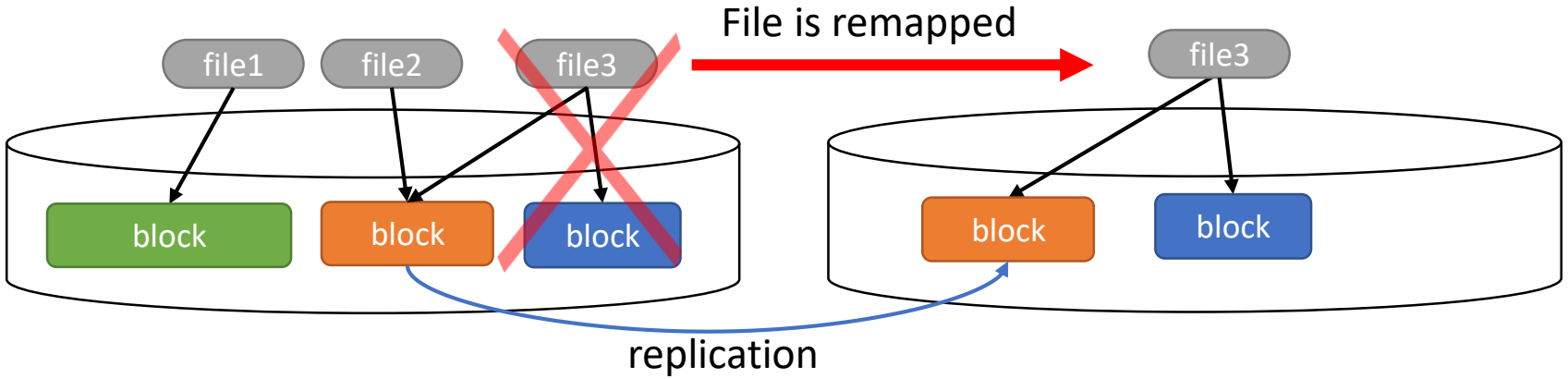
Seeding in deduplicated systems

- System contains only unique physical blocks
- File remap can cause replication

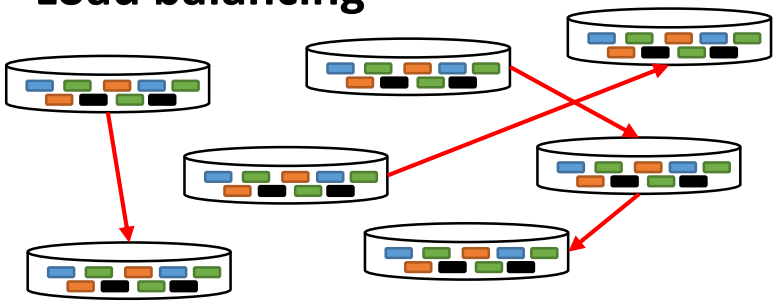


Seeding in deduplicated systems

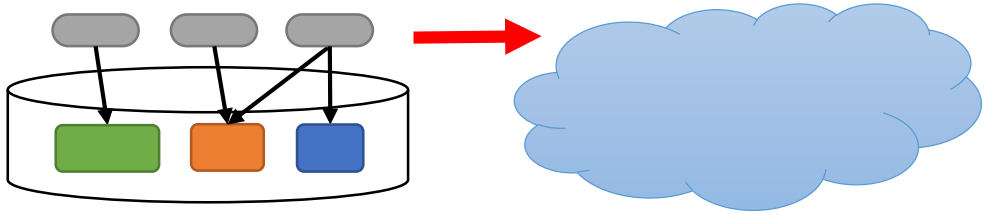
- System contains only unique physical blocks
- File remap can cause replication



Load balancing

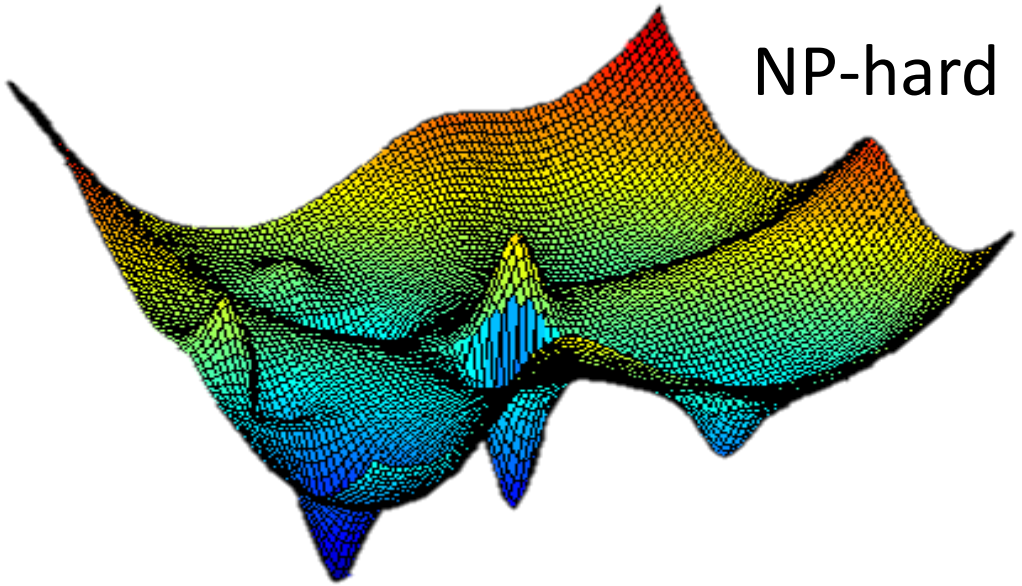


Data domain cloud tier [Duggal et al. ATC'19]



The seeding optimization problem

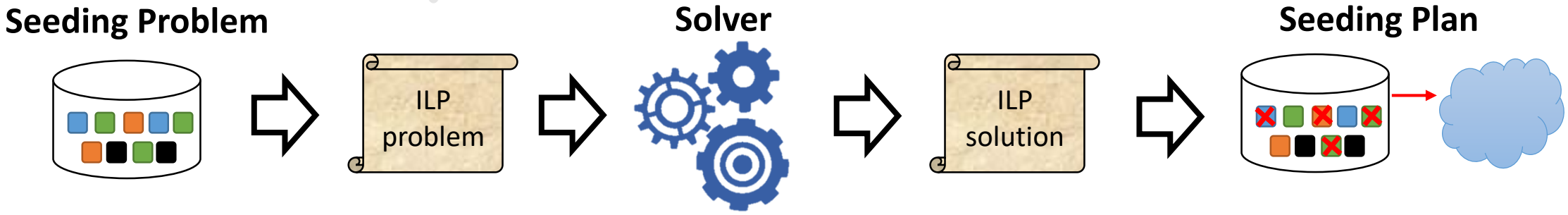
- Migrate $M\%$ of physical occupancy to an empty destination
- **Minimize block replication**
- Each file is kept/remapped
- Each block is kept/moved/replicated



Our approach: formulate seeding as ILP

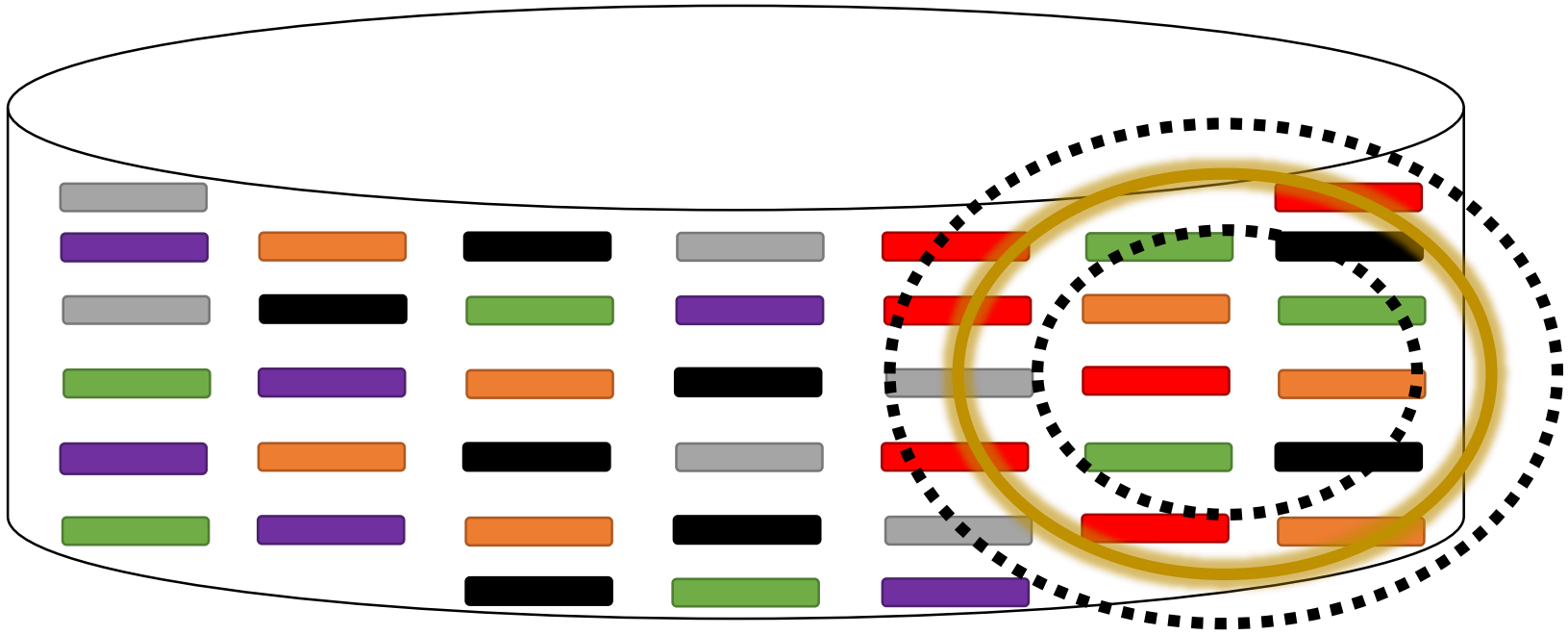
- ILP [Integer Linear Programming] optimization with linear constraints
- NP-hard
- Industrial and open source ILP solvers

Use ILP for data migration



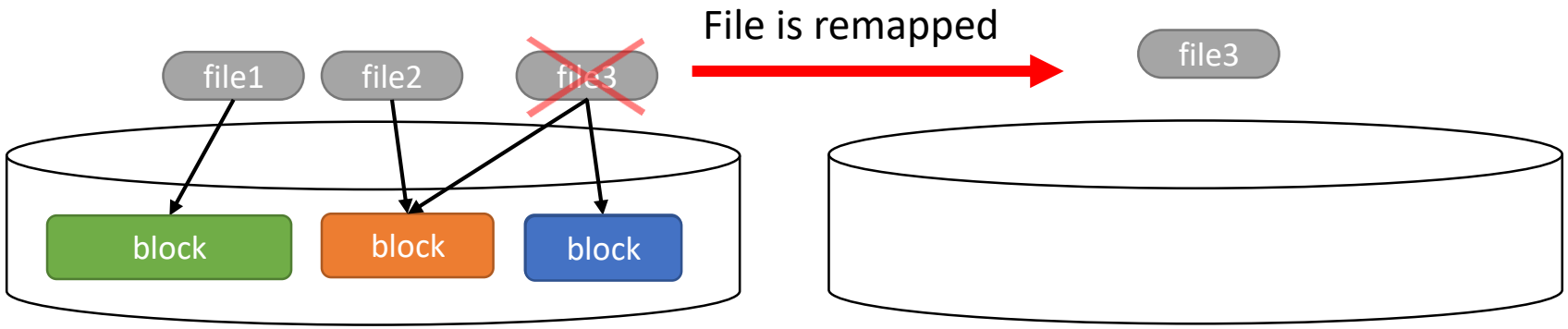
Constraint I: migration goal

- Migrate M% of physical occupancy
- Actually $M \pm \epsilon$



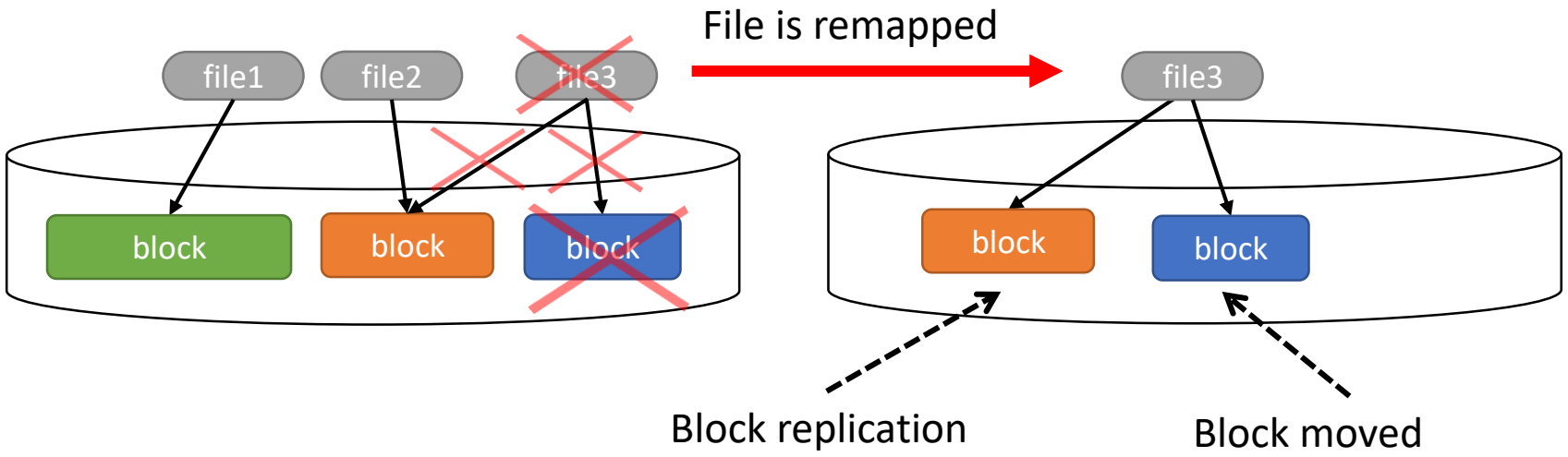
Constraint II: blocks follow their file

- Blocks are copied or moved with their files



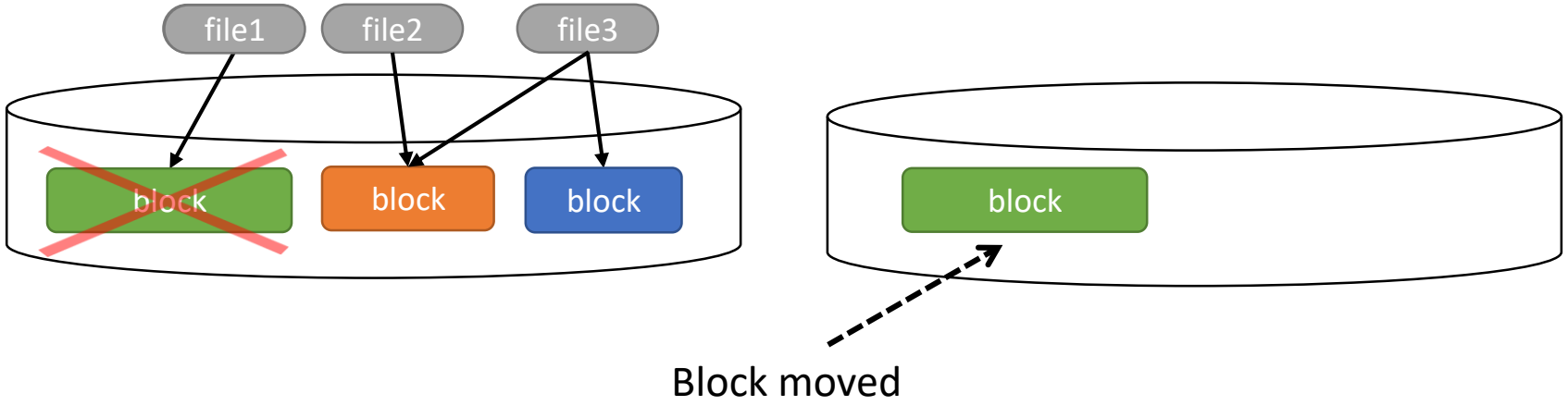
Constraint II: blocks follow their file

- Blocks are copied or moved with their files



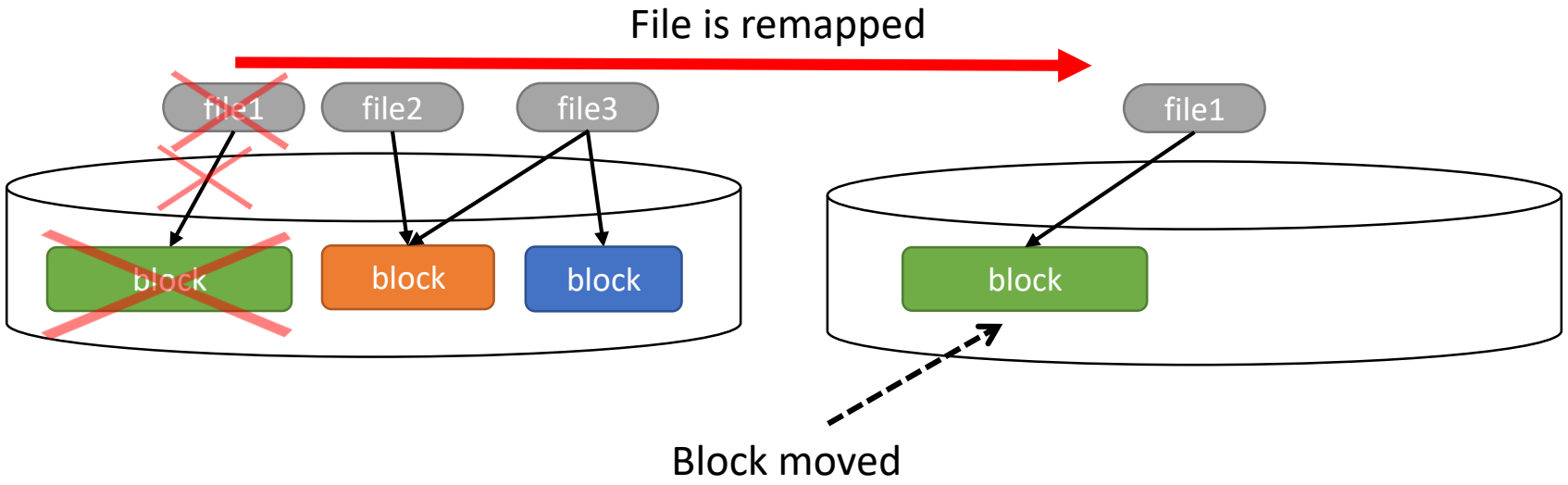
Constraint III: file follows removed blocks

- Blocks cannot move without their files



Constraint III: file follows removed blocks

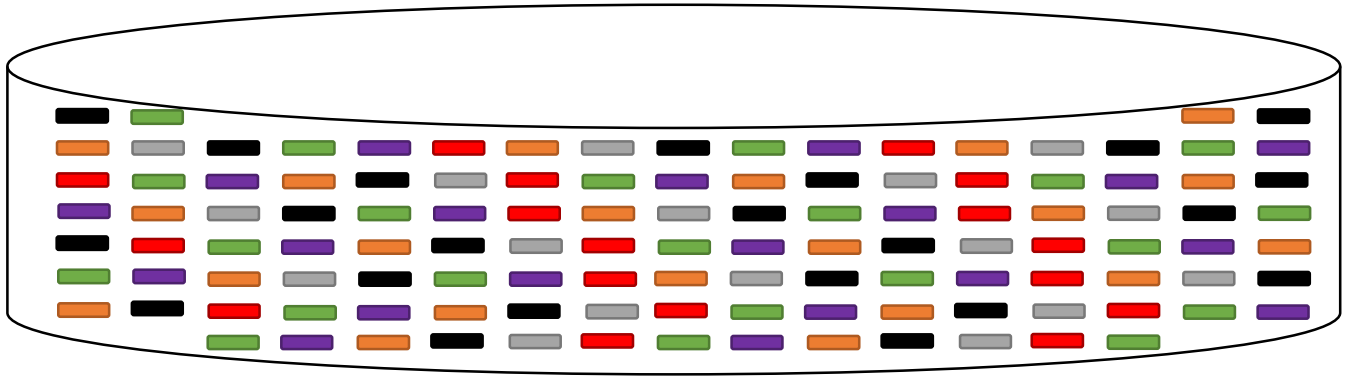
- Blocks cannot move without their files



Theory vs. Practice

- ~10 logical TB ⇒ ~billion variables and constraints
- Solvers are efficient with several 100K
- Acceleration (approximation) methods:

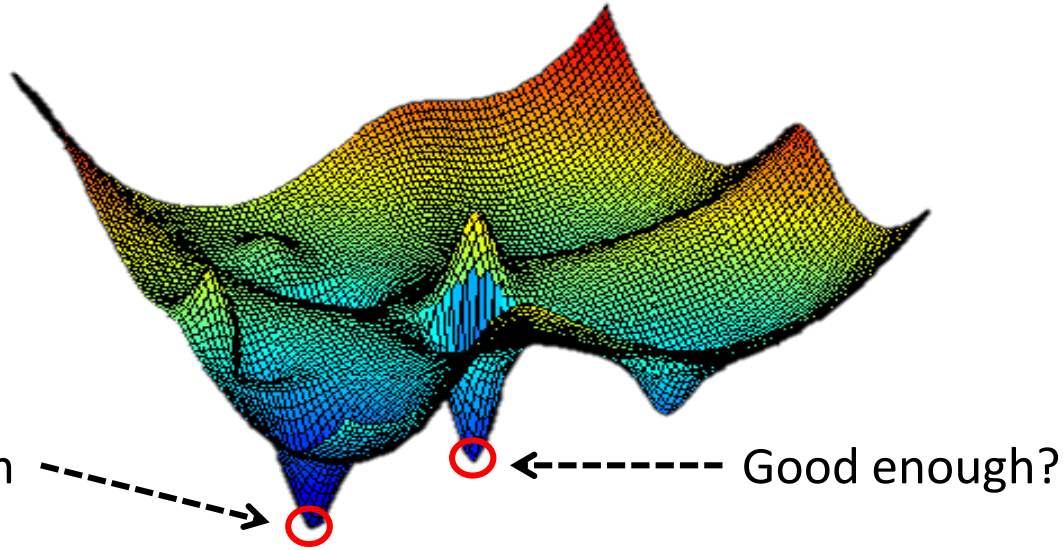
  Shorter running time ⇒ Further from optimum



Method I: solver timeout

- Solvers can output best solution so far
- Most progress is made early

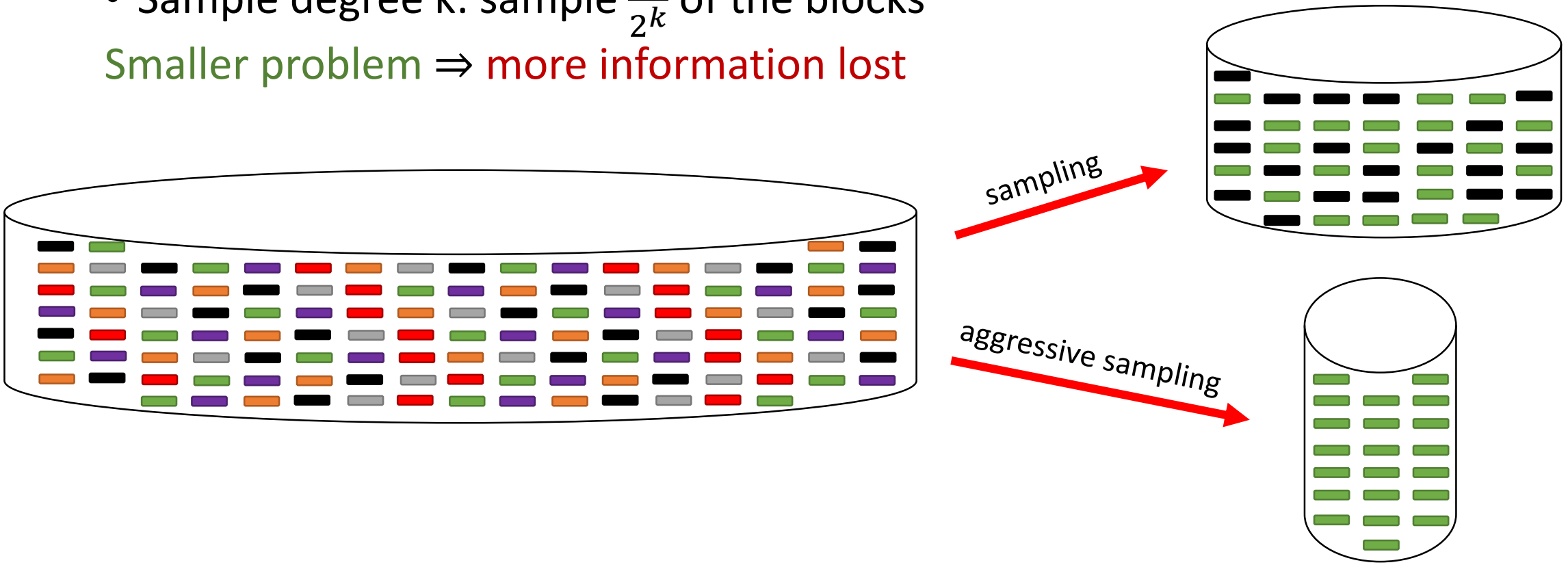
Shorter guaranteed running time ⇒ Further from optimum



Method II: fingerprint sampling

- Sample a subset of the fingerprints [Harnik et al. FAST19]
- Sample degree k: sample $\frac{1}{2^k}$ of the blocks

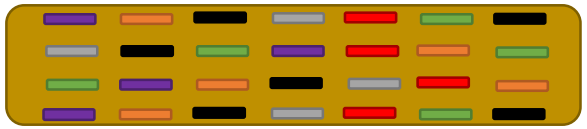
Smaller problem ⇒ more information lost

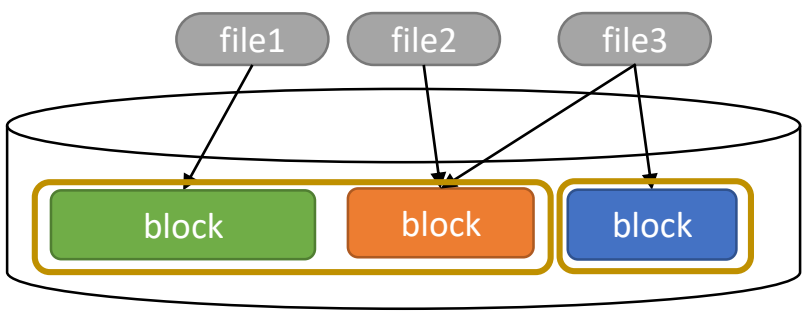


Method III: container-based aggregation

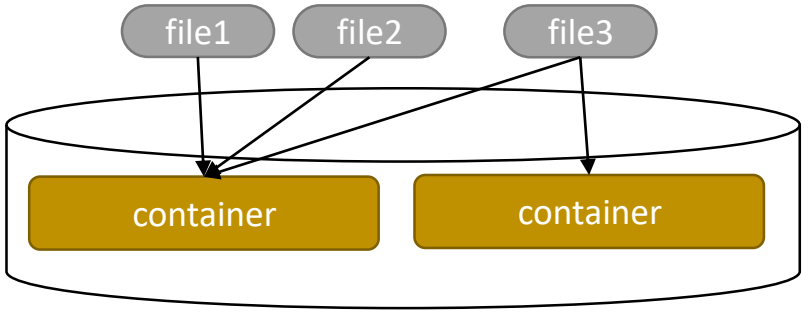
- Deduplication system stores blocks in containers
- We treat each container as a block

+smaller problem
+no need to decompress/unpack containers
- Creates false dependencies





Original representation



Container representation

Experimental setup

We compare:

- **GoSeed: our ILP-based approach**
- **Rangoli [Nagesh & Kathpal Systor'13]:**
  - Sort & divide the blocks into bins
  - Migrate the "best" bin

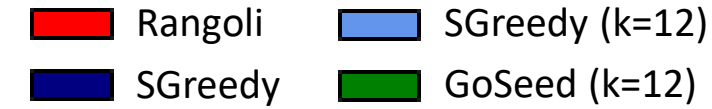
- **SGreedy [Harnik et al. FAST19]:**
  - Iterative method
  - Each iteration remap the "best" file

	UBC-500	Homes	MacOS-day
Logical size	19.5TB	8.9TB	43TB
Dedup ratio	0.31	0.13	0.01
# files	500	81	200

More workloads in the paper

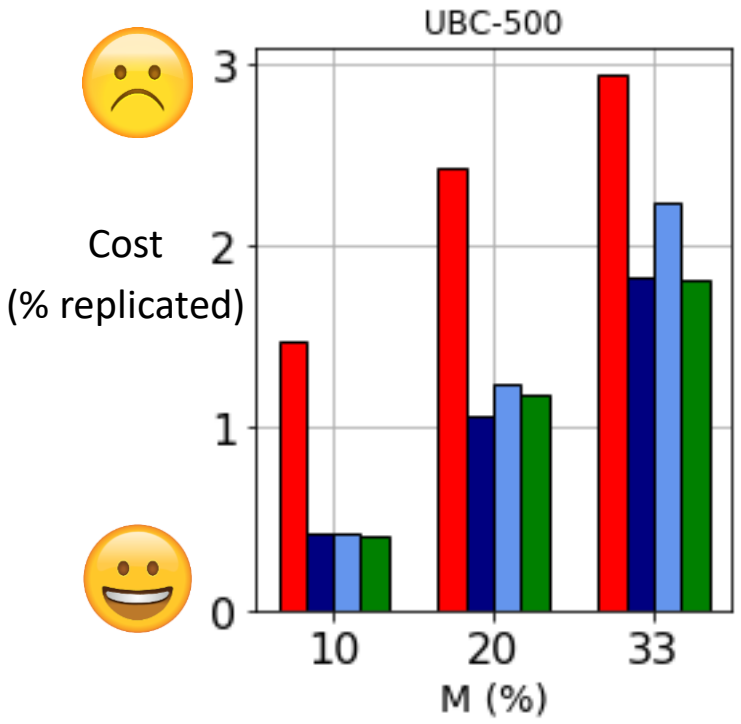
UBC traces: http://iotta.snia.org/traces/
FSL traces: http://tracer.filesystems.org/

Cost

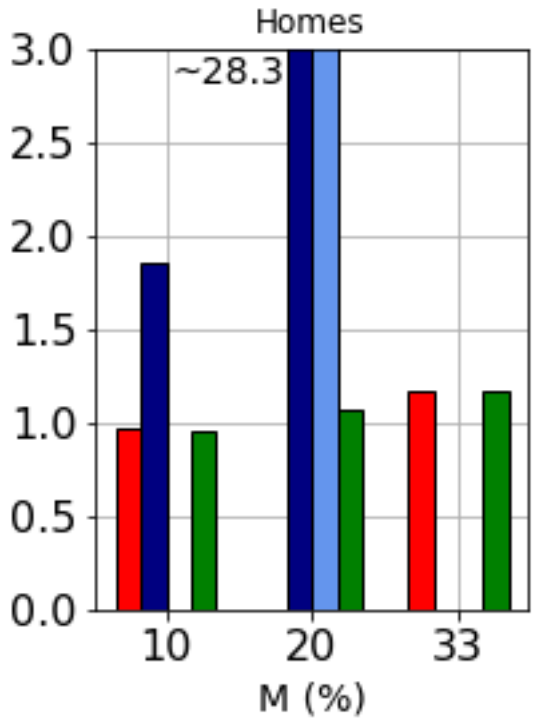


- UBC-500 is considered “easy”
- Homes is harder
- MacOS-Day consists daily backups

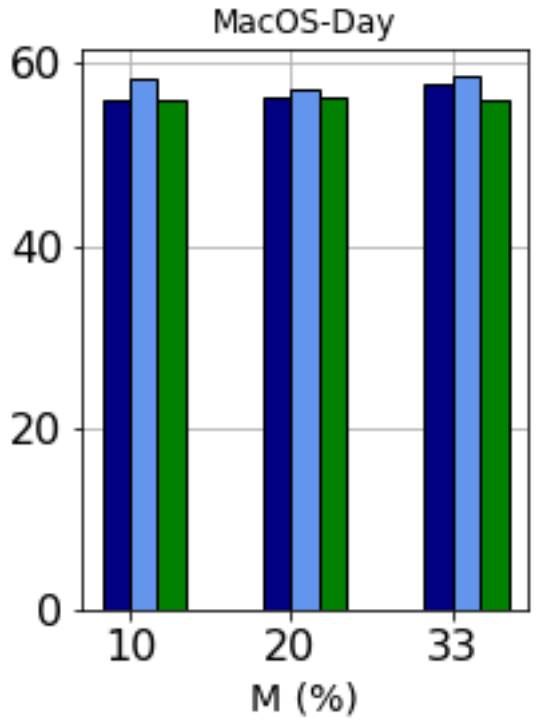
Low dedup



High dedup



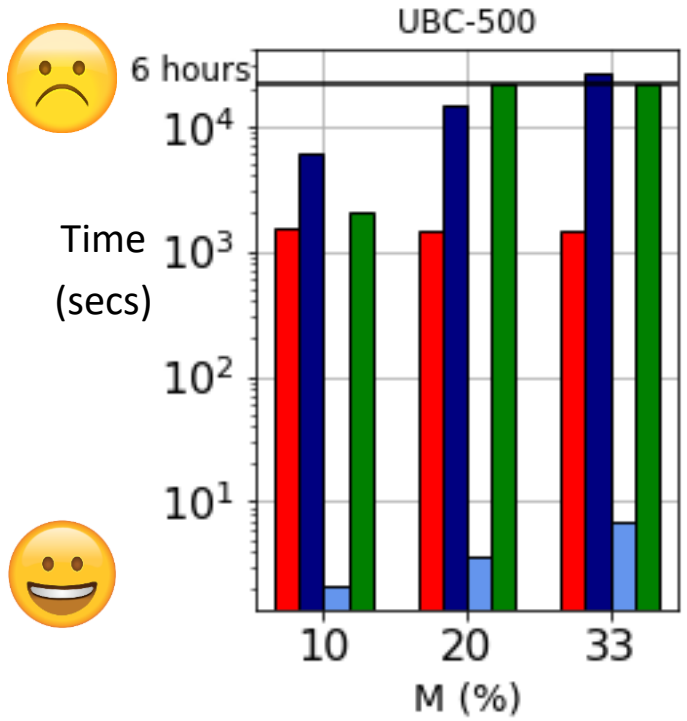
Very high dedup



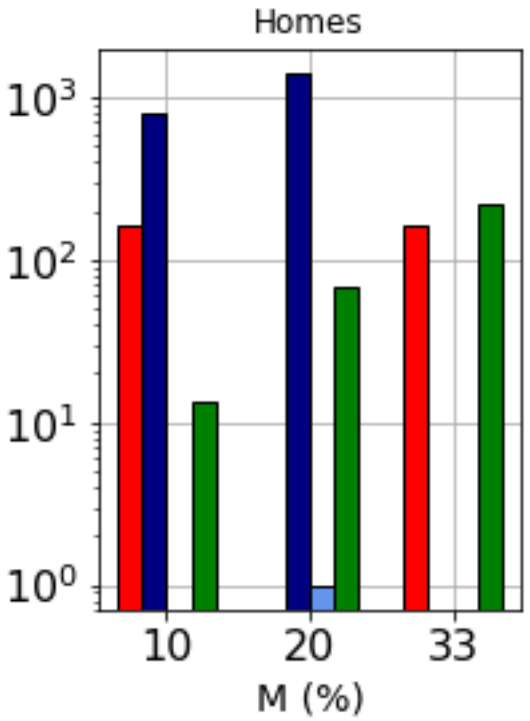
GoSeed always finds a solution, it’s often the best

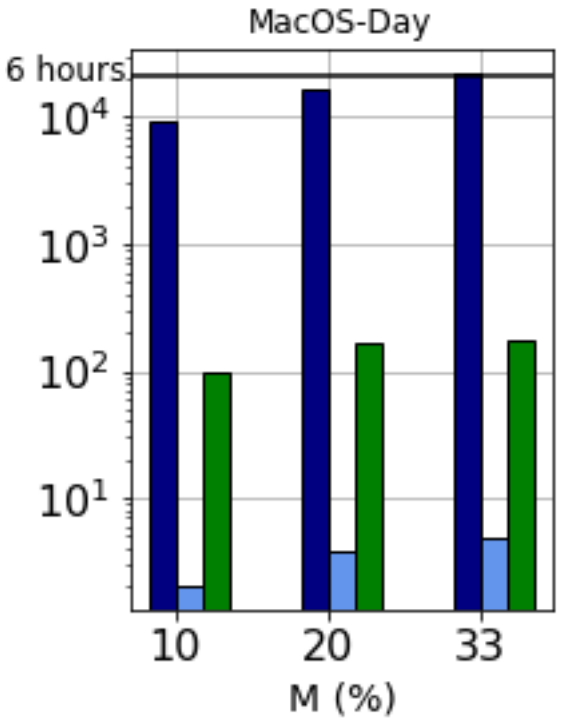
Time

- Sampling reduces solving time



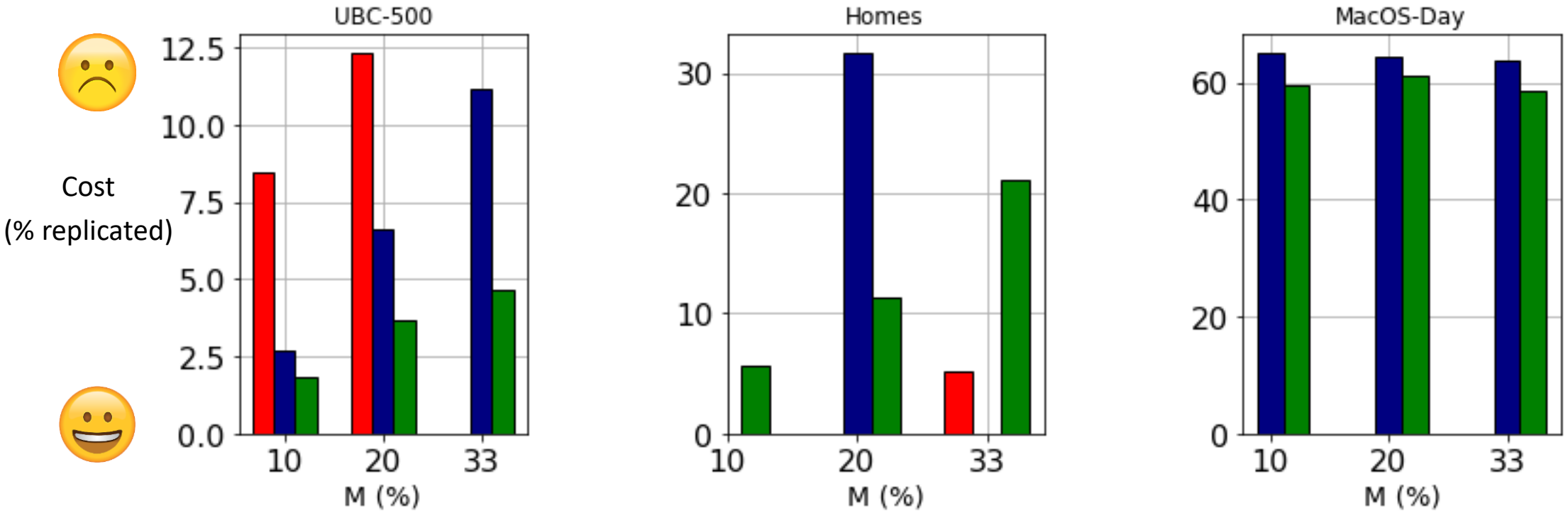
- Complexity costs time (tradeoff)





Container cost

- GoSeed always gives a solution, almost always better



Additional evaluation in the paper: effects of M, ϵ, k, timeout

Conclusions

- We solve the seeding problem with ILP solvers
- Acceleration methods are essential and effective
- GoSeed often outperforms greedy approaches
- Future steps: general data migration

More details in the paper