GoSeed: Generating an Optimal Seeding Plan for Deduplicated Storage

Aviv Nachman, Gala Yadgar, Sarai Sheinvald

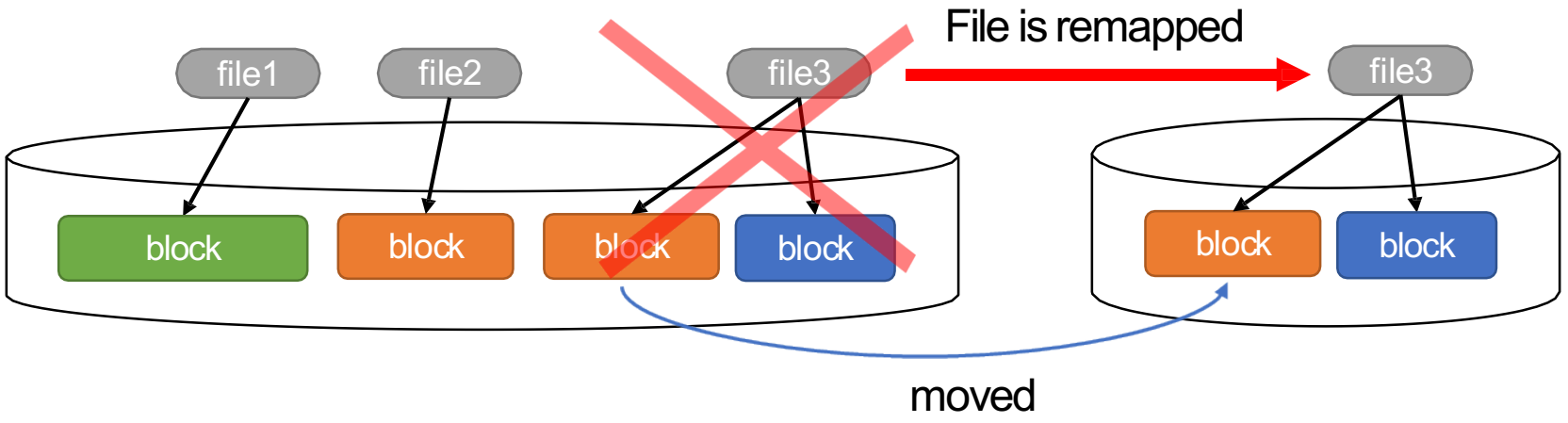
Speaker： Liang, Jiacheng

Background

➢Data Migration:

- Moving data between volume
- Typically performed for load balancing and resizing

➢Seeding : Data migration with empty destination



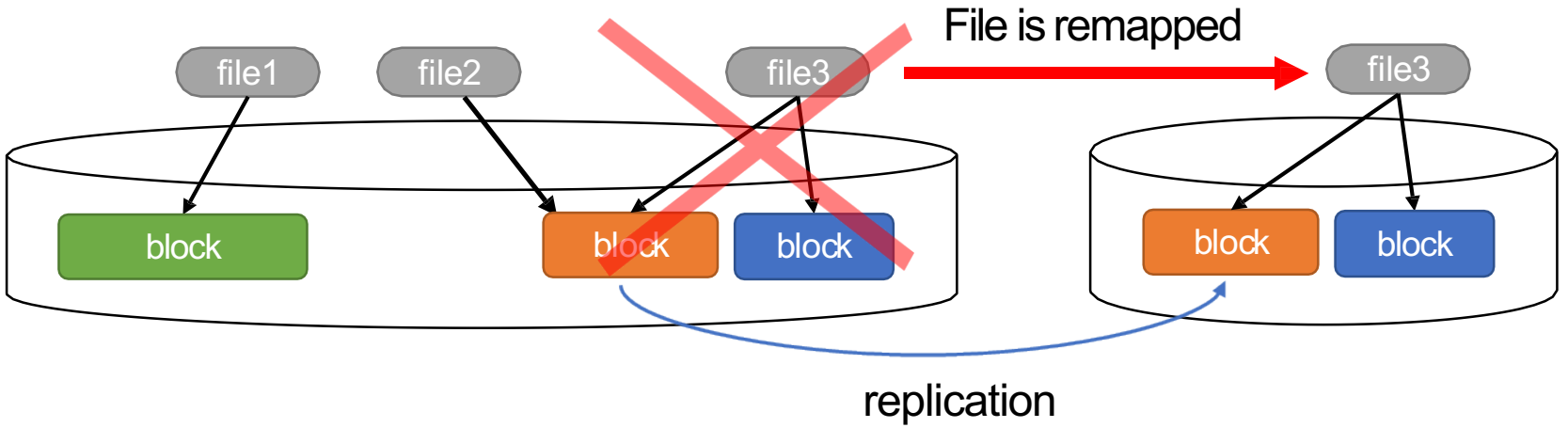
Motivation

➢Deduplication complicates the migration:

- File remapped will cause replication.

➢Current status:

- Greedy algorithms for determining the set of migrated files, but the efficiency has never been systematically compared.



Background :What is ILP

➢Satisfies

$a_0x_0 + a_1x_1 + a_2x_3 + \cdots + a_{n-1}x_{n-1} \leqslant c$, where $a_0, \ldots, a_{n-1}, c \in Z$

➢Maximizes

$Tx = a_0x_0 + a_1x_1 + a_2x_3 + a_0x_0 + \cdots + a_{n-1}x_{n-1}$

➢How to combine data migration

c as the the migration amount, make $a_0x_0 + a_1x_1 + a_2x_3 + a_0x_0 + \cdots a_{n-1}x_{n-1}$ closest to c .

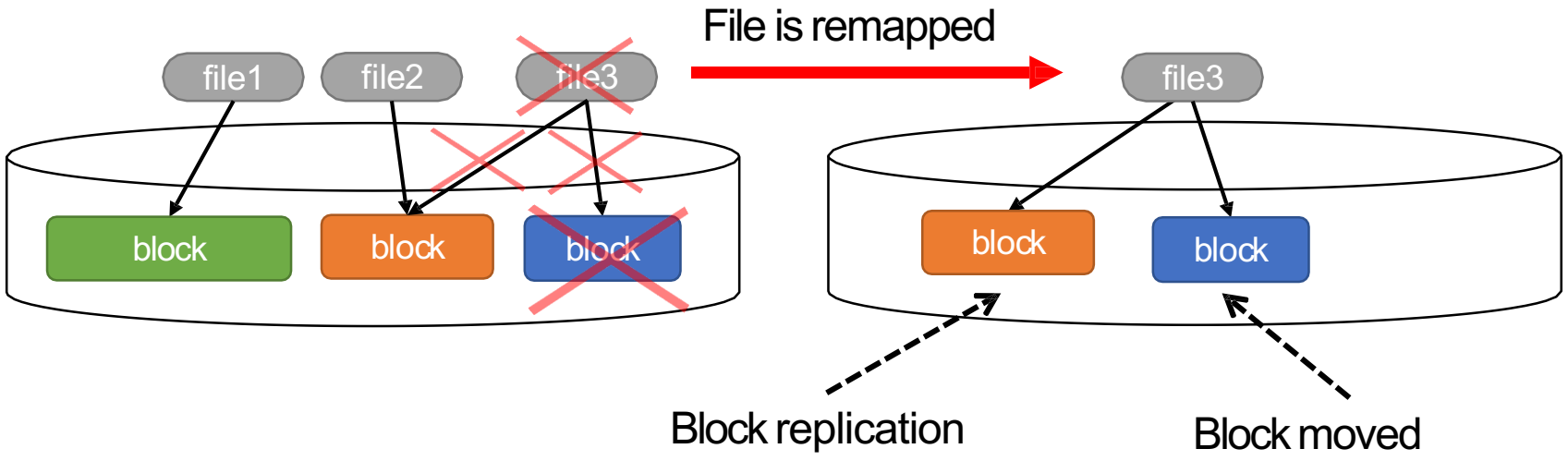
Design：ILP combine data migration

- Migrate M% of physical occupancy
- Actually $M \pm \epsilon$
- R as the cost of the migration
- m_i : block i migrated (0/1)
- r_i : block i replicated (0/1)
- x_i : file i remmaped (0/1)
- s : the size of the block
- M : $\sum b_i \in B_{V_1} s(b_i) \cdot m_i = M$, The total size of migrated blocks
- R : $\sum b_i \in B_{V_1} s(b_i) \cdot r_i$

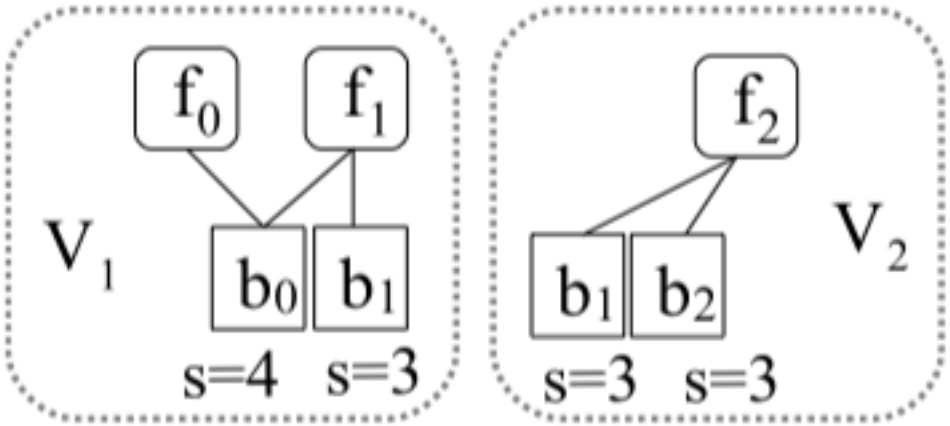
target : minimize R ,closest M

Constraint : blocks & file are always bound

- Blocks are copied or moved with their files
- Blocks cannot move without their files



Design：ILP formulation



- m_i : block i migrated (0/1)
- r_i : block i replicated (0/1)
- x_i : file i remmaped (0/1)
- s : the size of the block
- M : $\sum b_i \in B_{V_1} s(b_i) \cdot m_i = M$, The total size of migrated blocks
- R : $\sum b_i \in B_{V_1} s(b_i) \cdot r_i$

target : minimize R ,closest M

- Block b_2 is migrated: $m_2 = 1$
- File f_2 is remapped: $x_2 = 1$
- Block b_1 is replicated: $r_1 = 1$
- The remaining files and blocks are untouched: $x_0 = x_1 = m_0 = m_1 = r_0 = r_2 = 0$
- The total cost is $R = 3 \cdot r_1 = 3$

Theory vs. Practice

- Dedup： ~10 logical TB ⇒ ~billion variables and constraints
- ILP Solvers： Solvers are efficient with several 100K

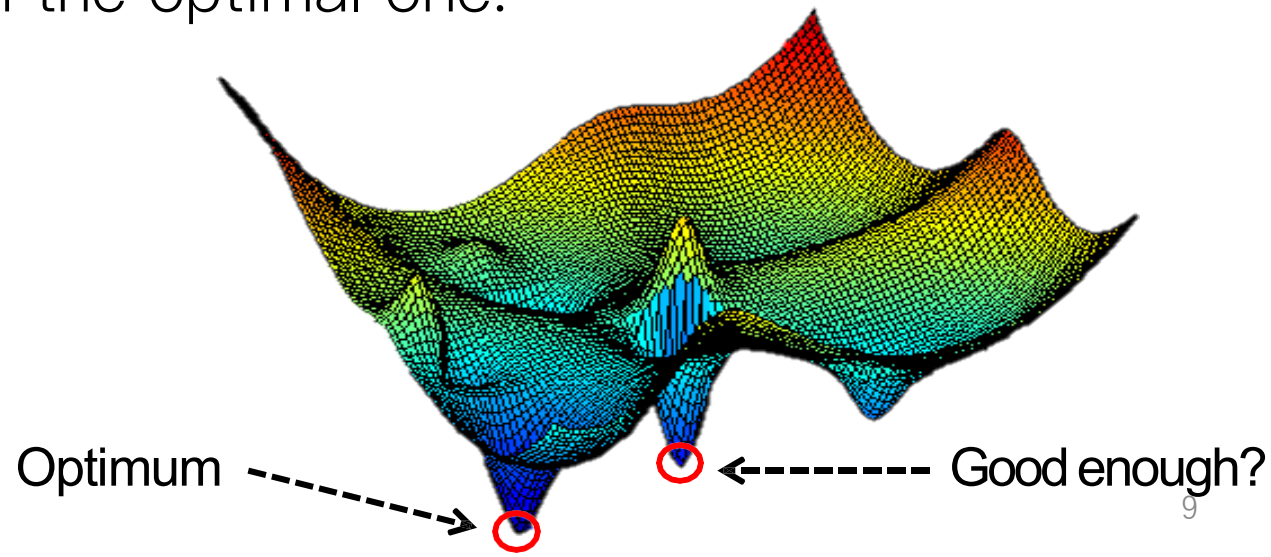
Method I: Solver timeout

Limited the runtime of ILP solver

- Specifying a timeout value
- Return the best feasible solution found thus far

Problem:

- When the solver is timed out, we cannot necessarily tell how far the suboptimal solution is from the optimal one.



Method II: fingerprint sampling

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

Problem:

- Smaller sample

⇒ a larger ILP instance

⇒ more accurately represents the sampled system

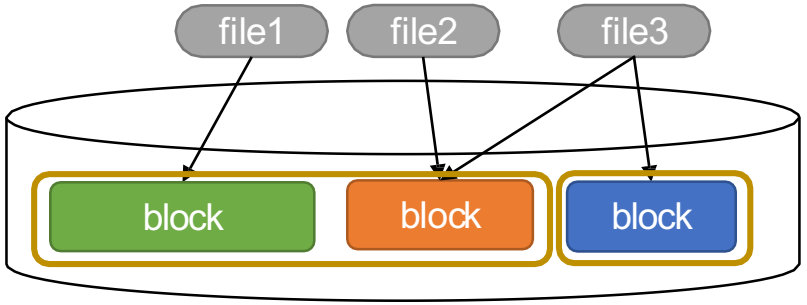
⇒ more likely to time out

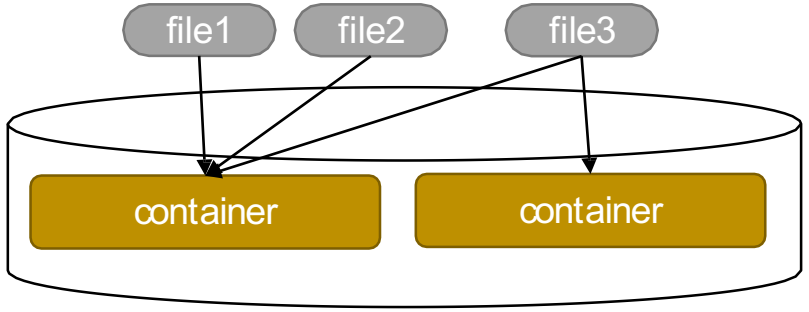
Method III: container–based aggregation

- Deduplication system stores blocks in containers
- We treat each container as a block
- use only the container IDs for generating the variables and constraints.

+smaller ILP problem
+no need to decompress/unpack containers
–false positives

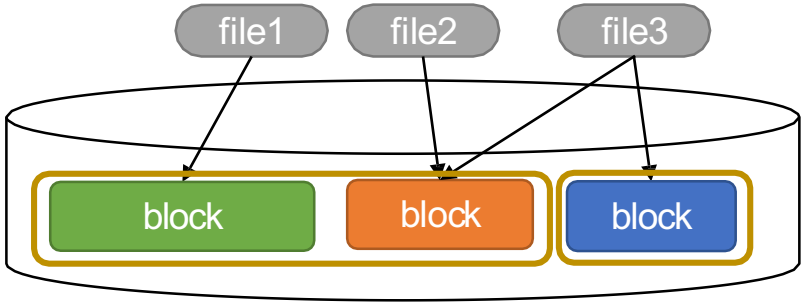
Original representation

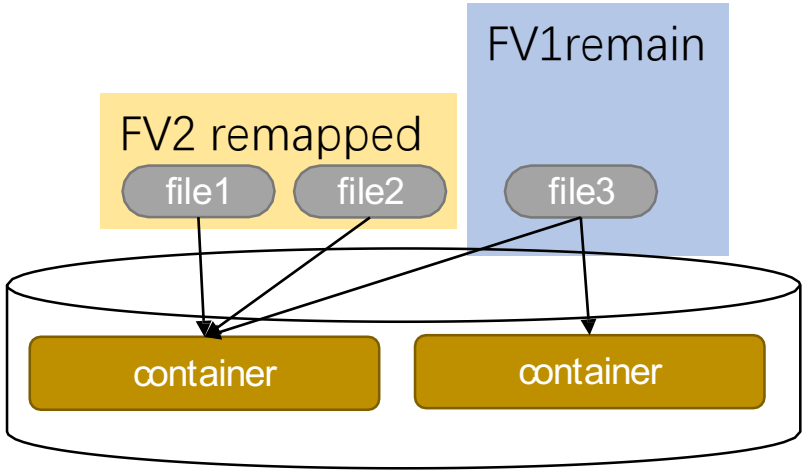


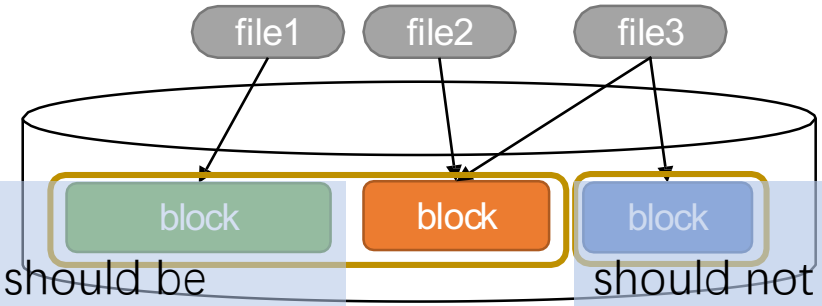


Container representation

–false positives

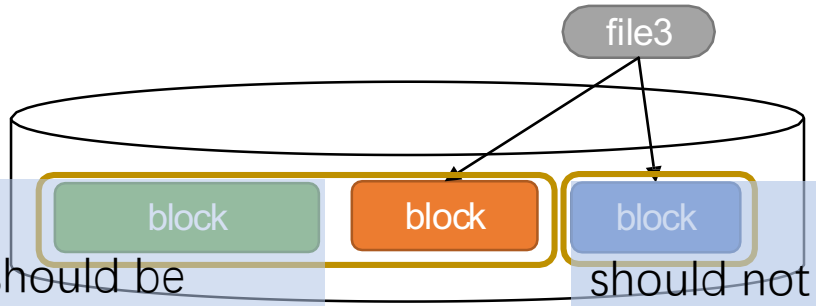






should be migrated rather than replicated

should not be part of the migration



should be migrated rather than replicated

should not be part of the migration

Experimental setup

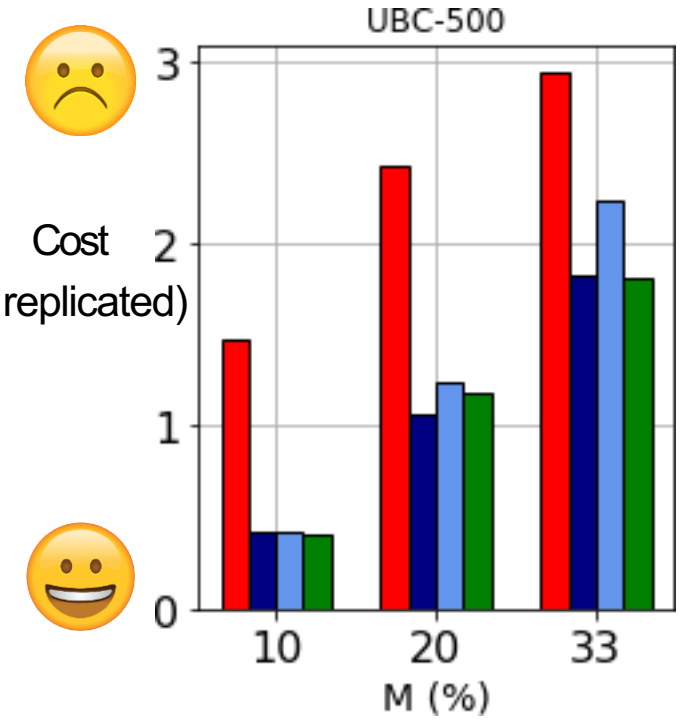
- We compare:
- GoSeed: our ILP–based approach
- Rangoli [Nagesh & Kathpal Systor'13]:
  - Sort & divide the blocks into bins
  - Migrate the "best" bin
- SGreedy [Harnik et al. FAST19]:
  - Iterative method
  - Each iteration remap the "best " file

	UBC 500	Homes	MacOS –day
Logical size	19.5TB	8.9TB	43TB
Dedup ratio	0.31	0.13	0.01

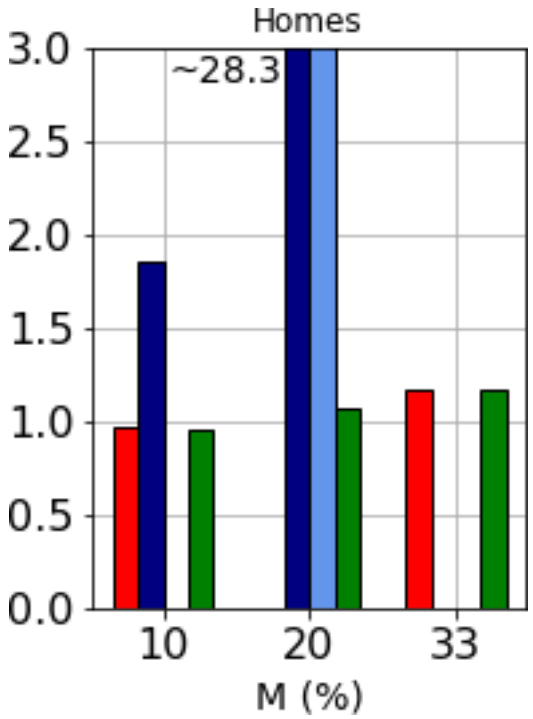
Cost

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

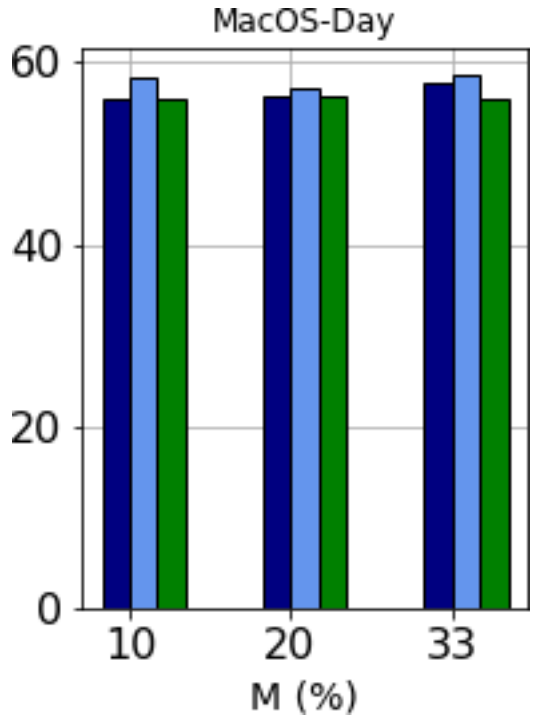
Low dedup



High dedup

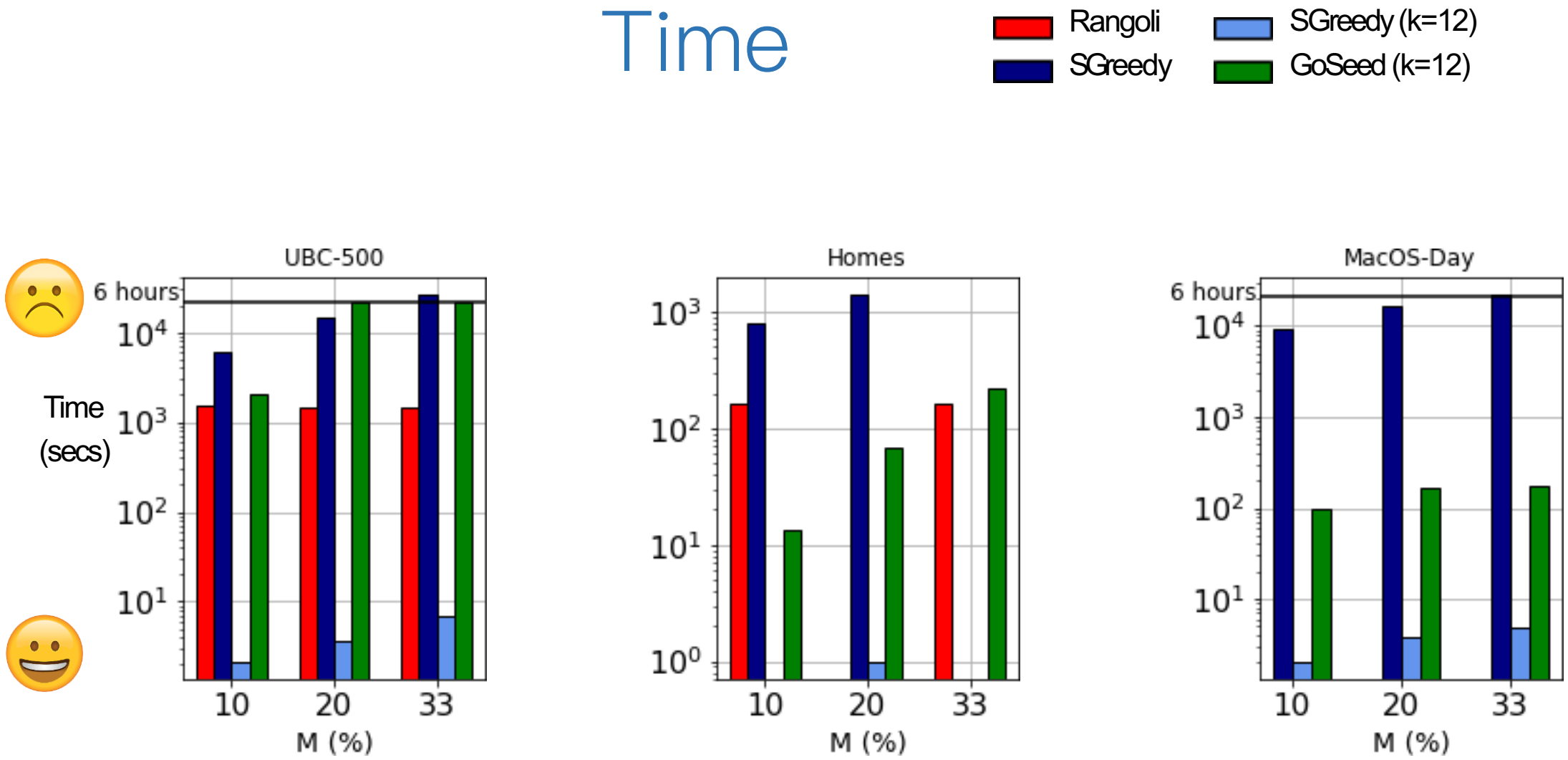


Very high dedup



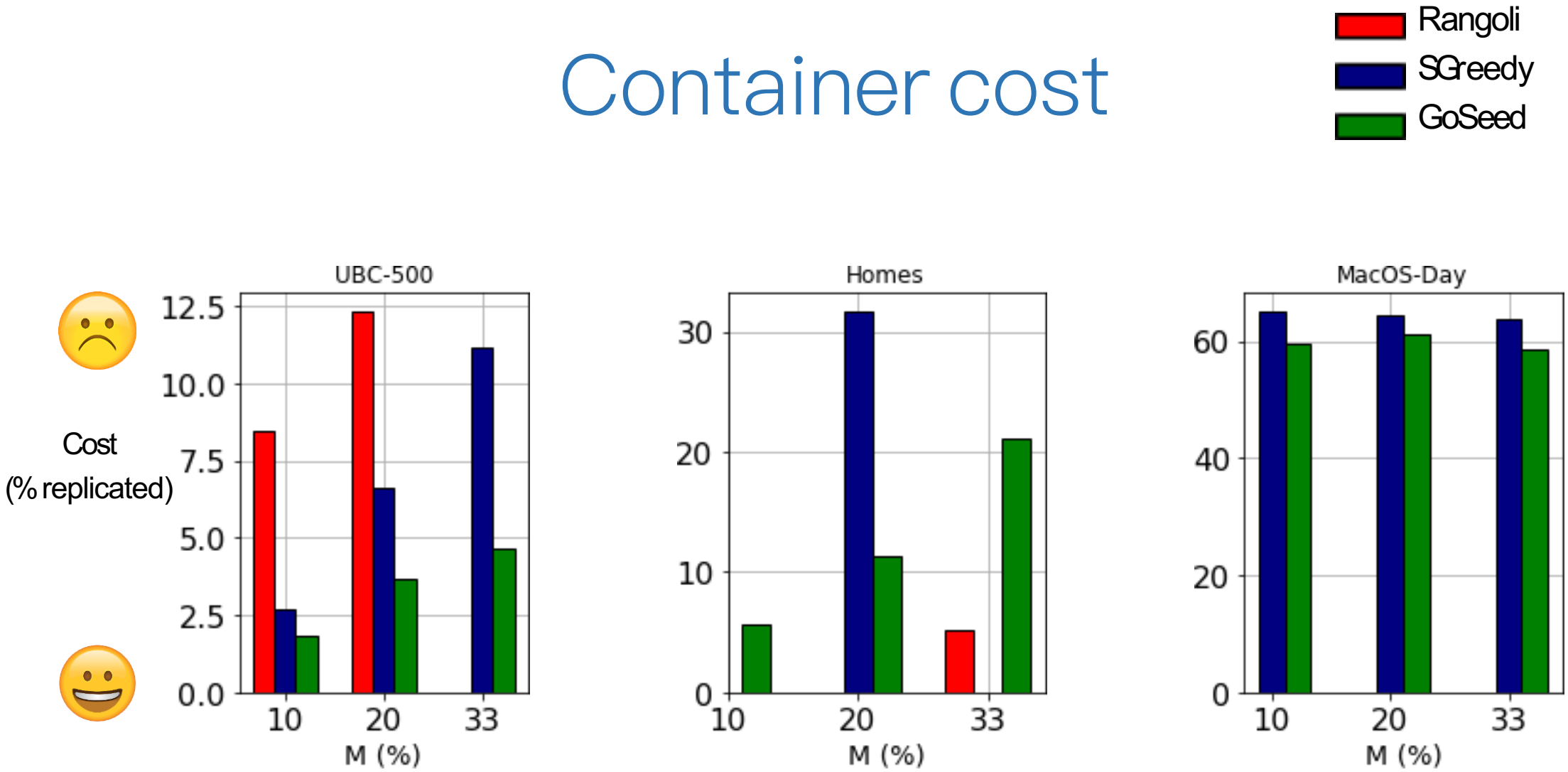
- GoSeed always finds a solution, it's often the best
- Greedy algorithms sometimes fail to find a solution

Time



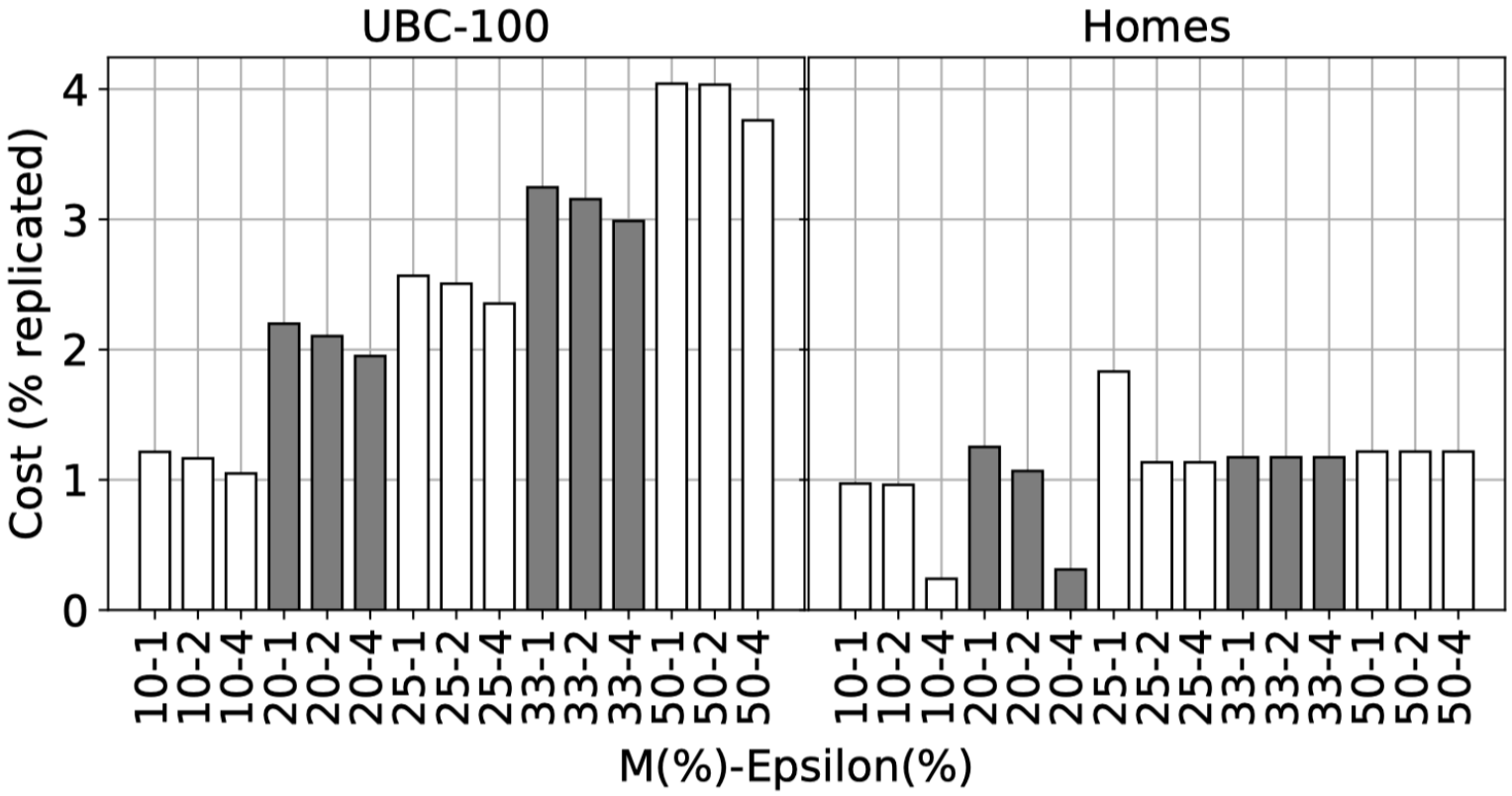
- Runtimes and resources are acceptable.

Container cost



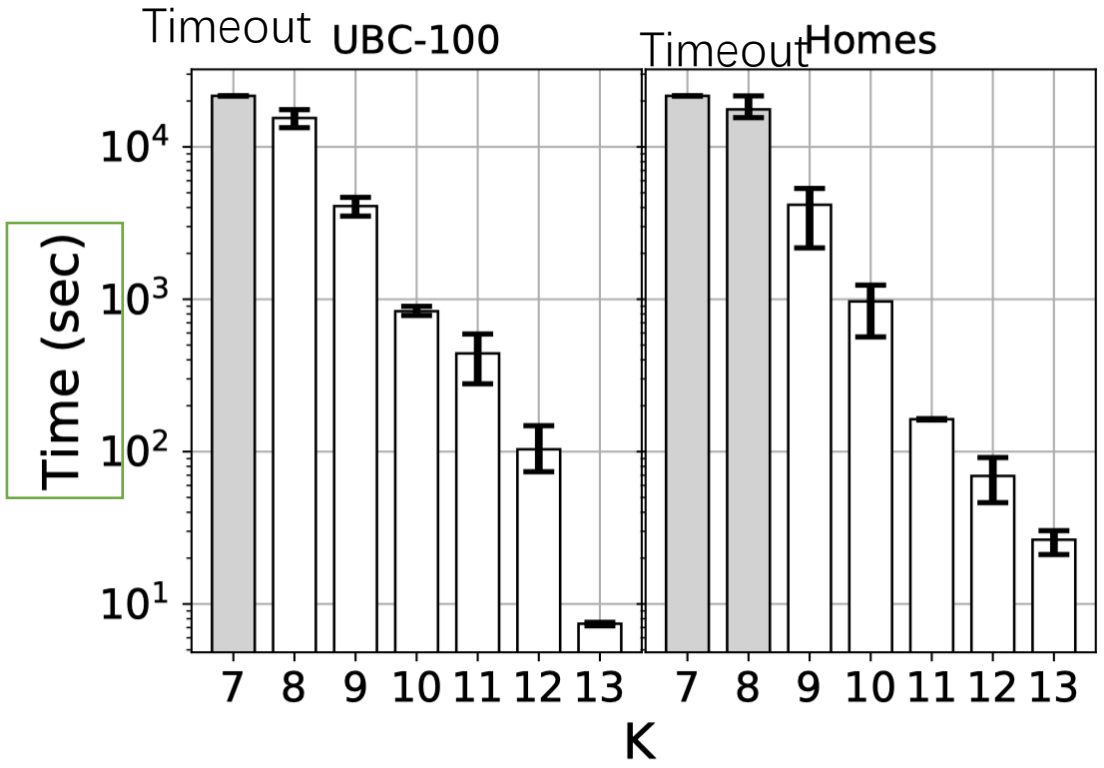
- GoSeed always gives a solution, almost always better

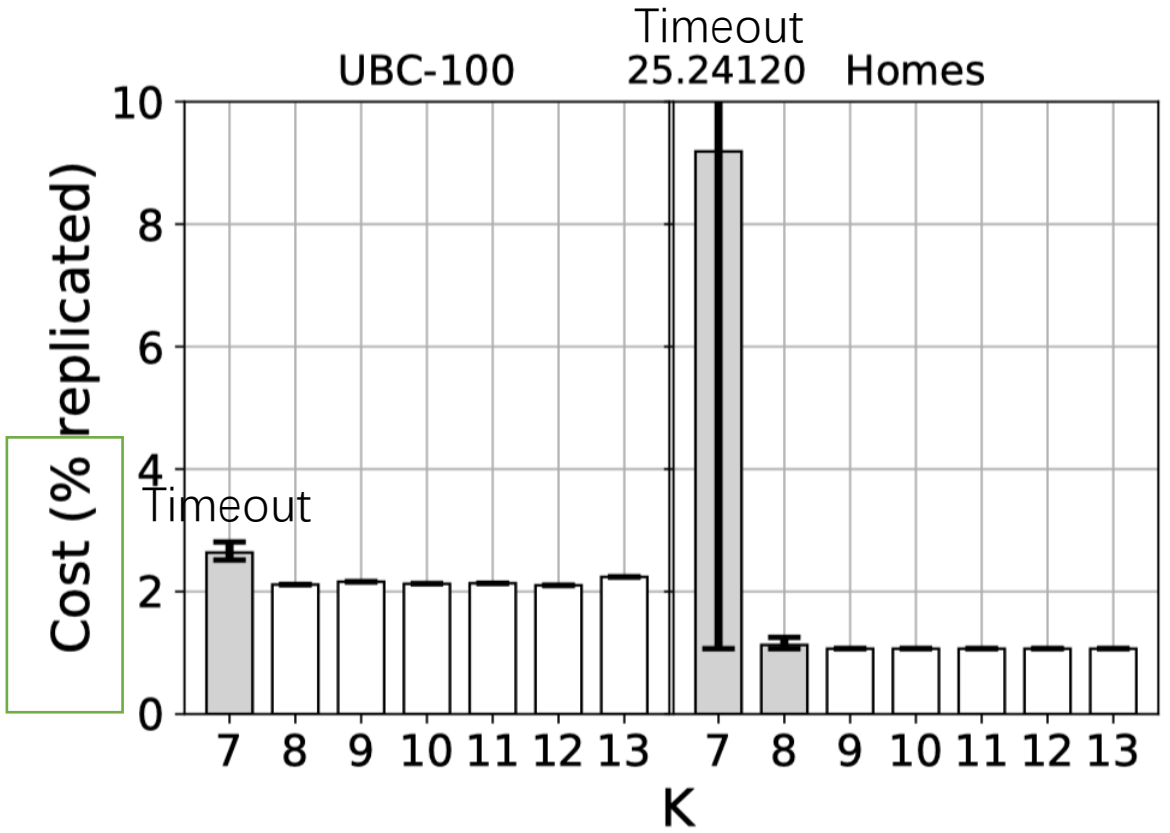
ILP parameter M–Epsilon Evalution



- The increase of M leads to higher cost, and the increase of ε leads to the decrease of cost

Effect of fingerprint sampling

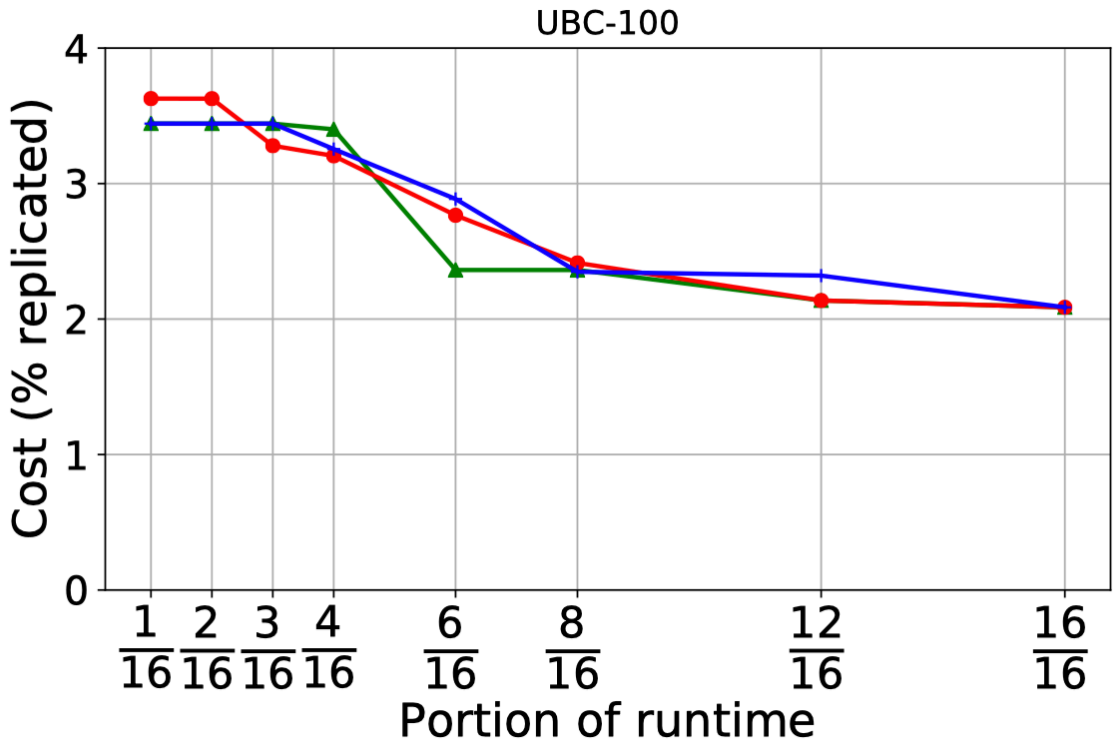




- Solving time increases exponentially with instance size
- Cost is hardly affected by the sampling degree, unless the instance becomes too large.

Suboptimal solution

—Solve Timeout



- Migration cost decreases when timeout increases

Conclusions

- Formulate of data migration as an ILP problem and solve it with ILP solvers
- Use three acceleration methods and they are essential and effective
- GoSeed often outperforms greedy approaches, and always find a feasible solution to the problem.
- The container-based approach is separate from the sample-based approach, applicable to different deduplication systems