Improving Load Balance via Resource Exchange in Large-Scale Search Engines

Kaiyue Duan, Yusen Li, Trent Marbach, Gang Wang, Xiaoguang Liu

College of Computer Science
Nankai University

June 2020

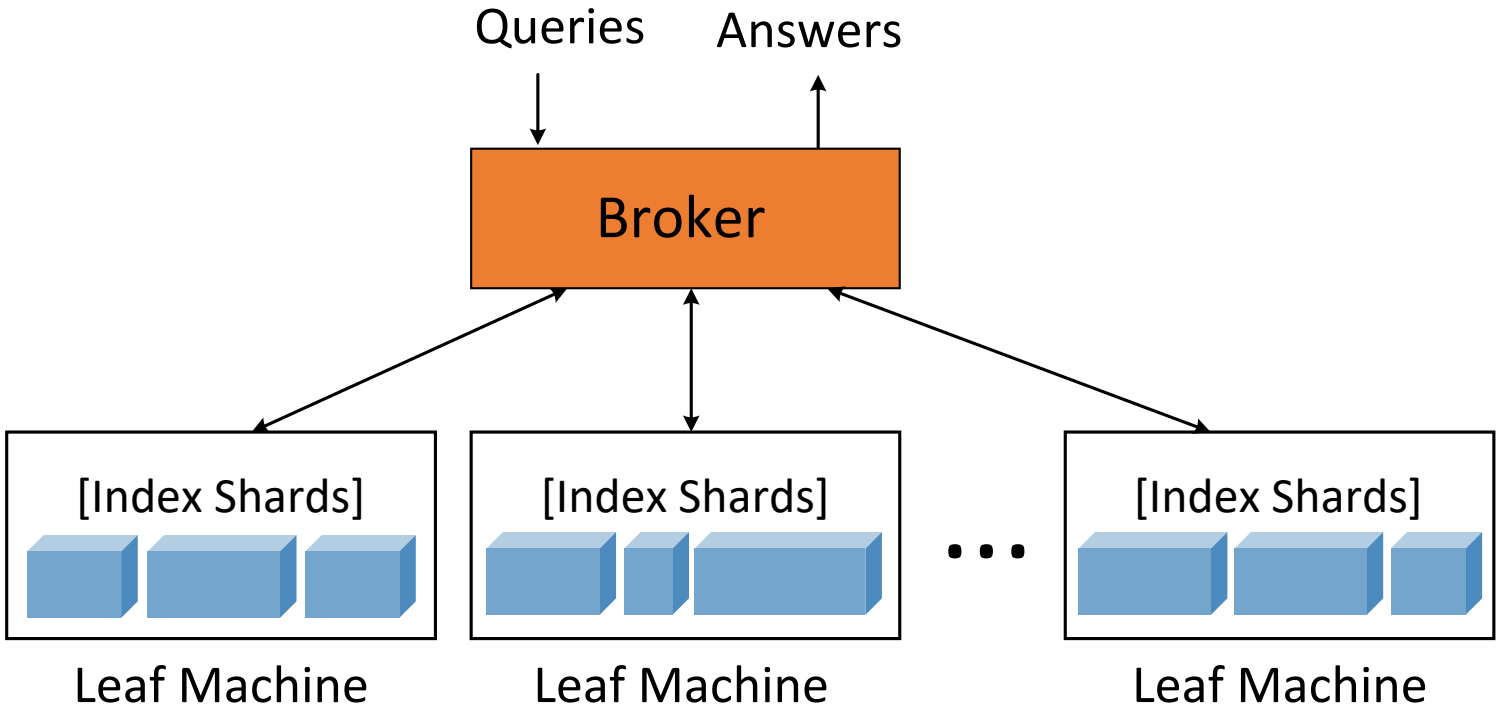
南開大學
Nankai University

Parallel and Distributed
Software Technology Lab

Nankai - Baidu
Joint Laboratory

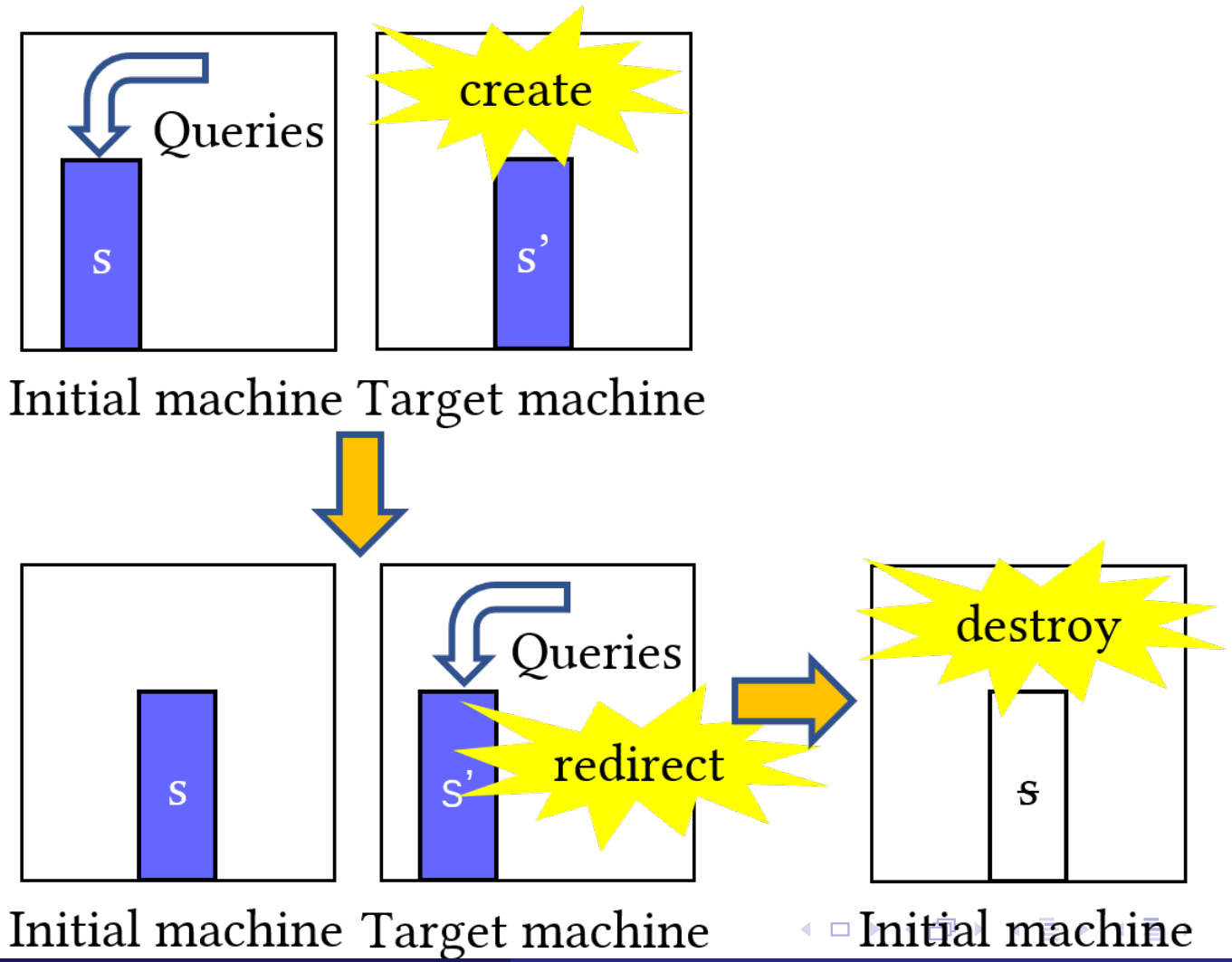
Background

- A simplified architecture of a search engine.



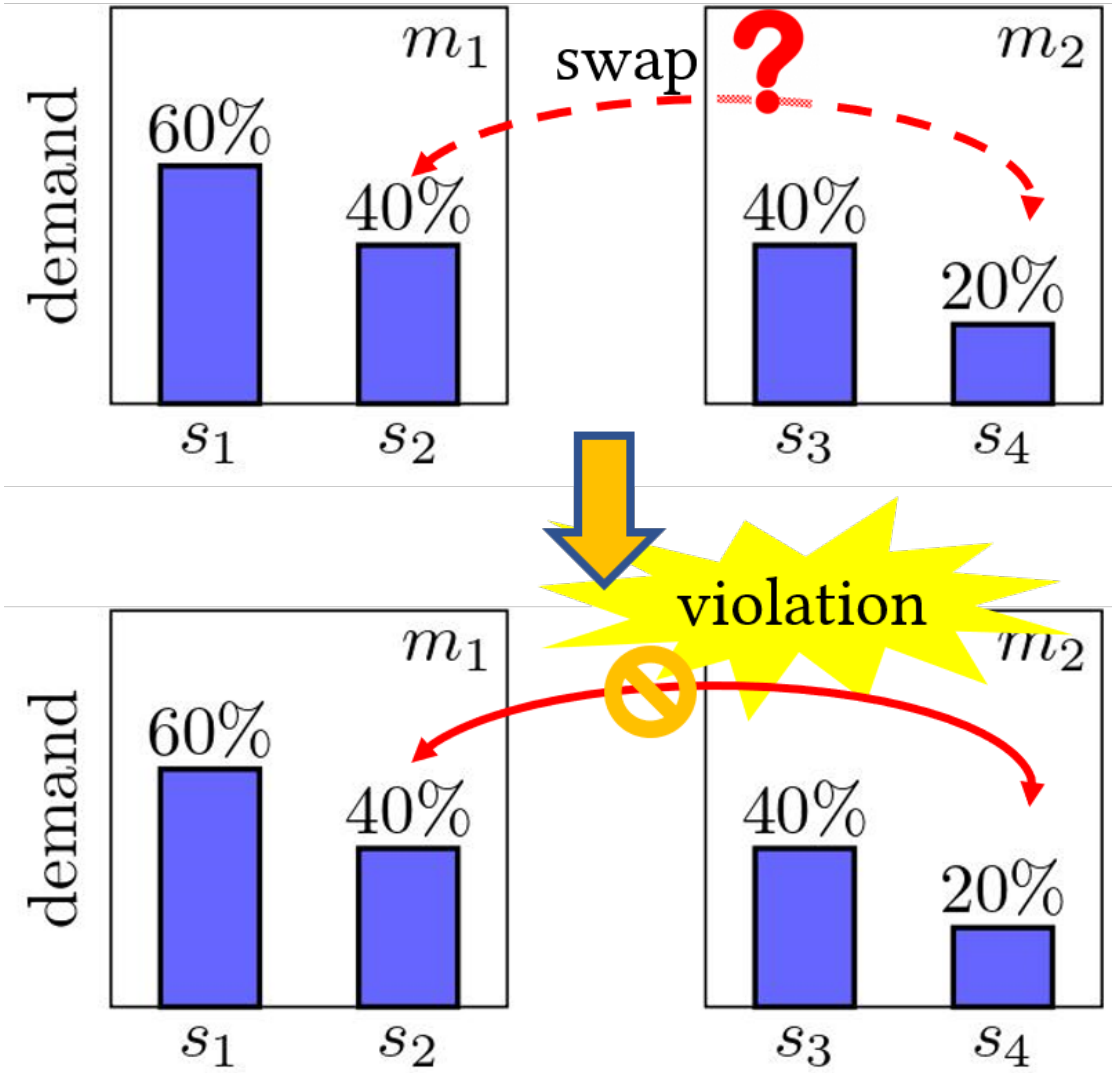
Background

- Load rebalancing is $\mathcal{NP}$-hard.
- E.g., Google machine reassignment problem.



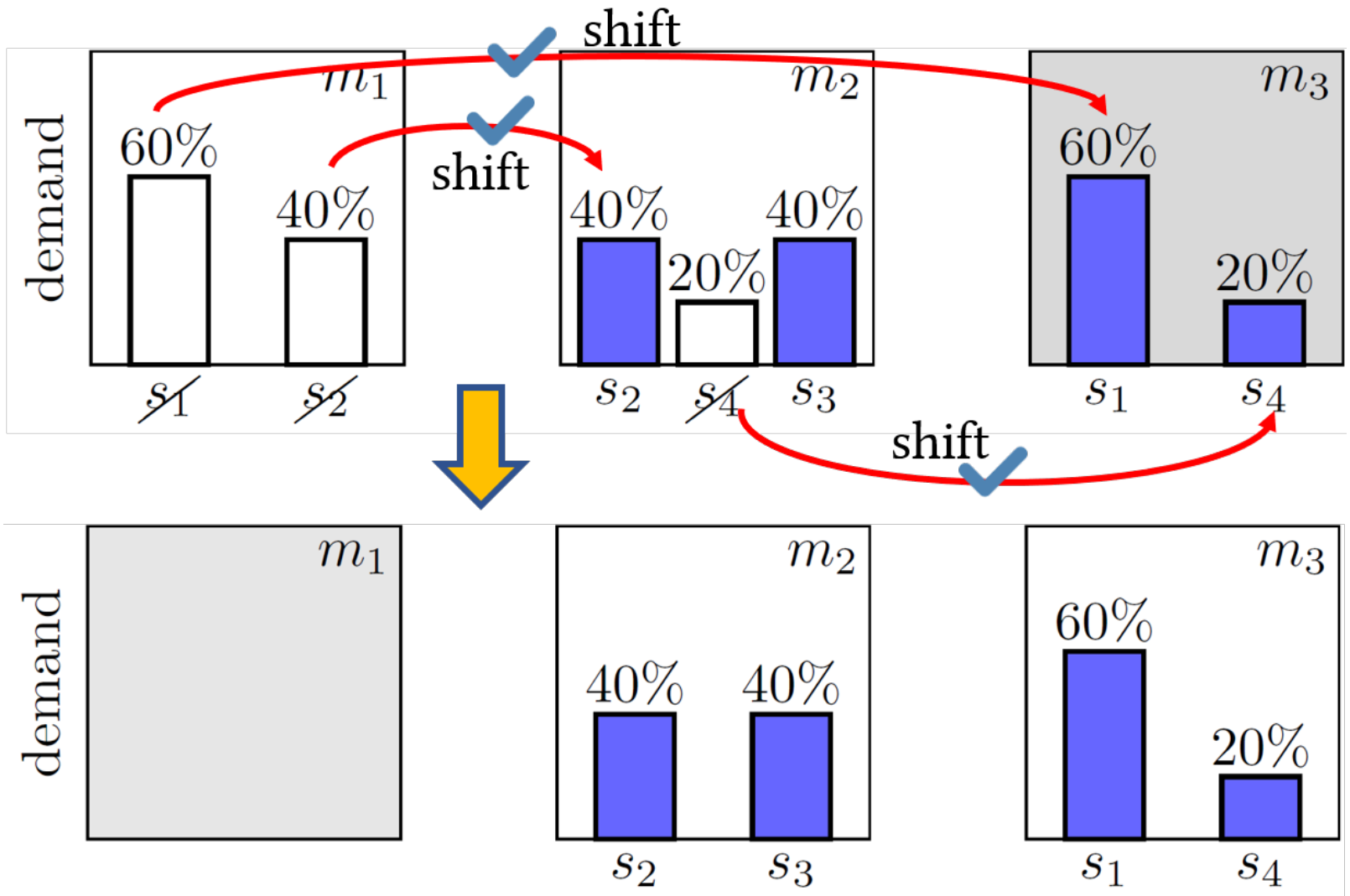
Motivation

- The transient resources affect shard reassignment!



Motivation

- We use the exchangeable machines to facilitate load balancing.



Shard Reassignment Problem

M_L	the set of leaf machines
M_E	the set of exchangeable machines
M	$M_L \cup M_E$
R	the set of resources
S	the set of shards
$c_m[r]$	the capacity of machine m for resource r
$u_m[r]$	the utilization of resource r by machine m
$utl[r]$	the expected utilization of resource r
$d_s[r]$	the demand for resource r by shard s
X	$X_{s,m} = 1$, shard s is assigned to machine m; $X_{s,m} = 0$, otherwise
V_m	if machine m will be returned

The objective function:

$$\min \sum_{r \in R} \sum_{m \in M} V_m \left| u_m[r] - utl[r] \right| .$$

,where the $u_m[r]$ and $utl[r]$ are defined as follows:

$$u_m[r] = \frac{\sum_{s \in S} X_{s,m} d_s[r]}{c_m[r]}.$$

$$utl[r] = \frac{\sum_{s \in S} d_s[r]}{\sum_{m \in M_L} c_m[r]}.$$

Shard Reassignment Problem

- Hard constraints:
  1. Single principle constraint
  2. Capacity constraint
  3. Transient constraint
  4. Conflict constraint
  5. Quota constraint

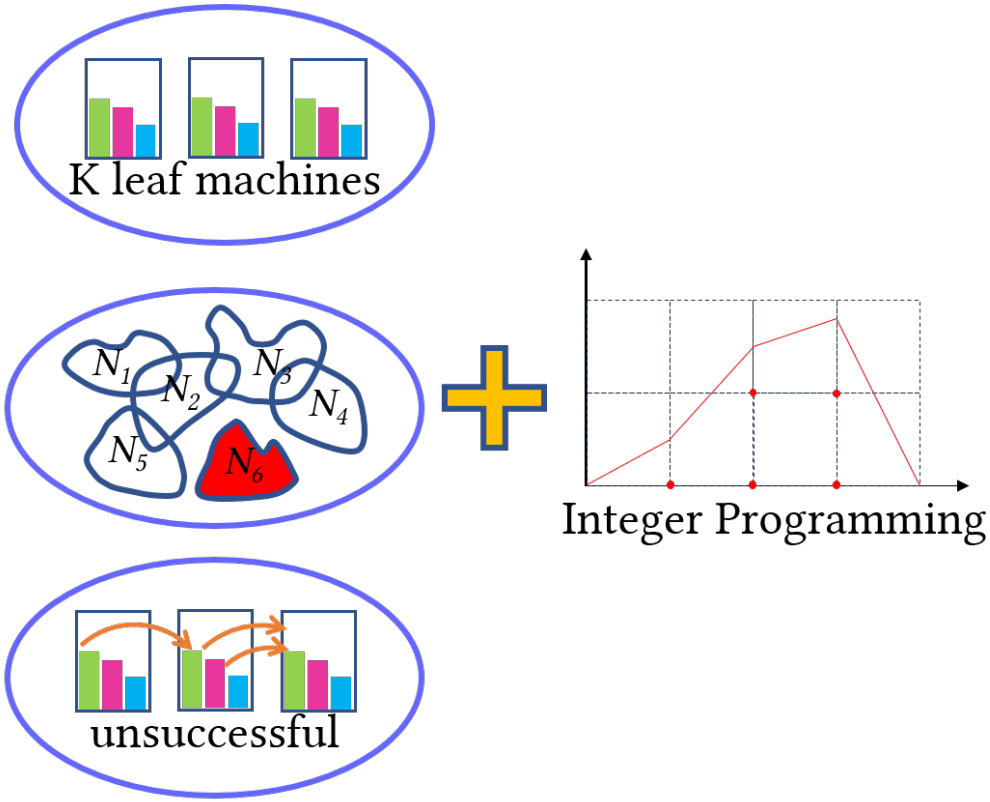
SRA overview

Shard Reassignment Algorithm (SRA)

1. Perform linear search on top K heaviest loaded machines

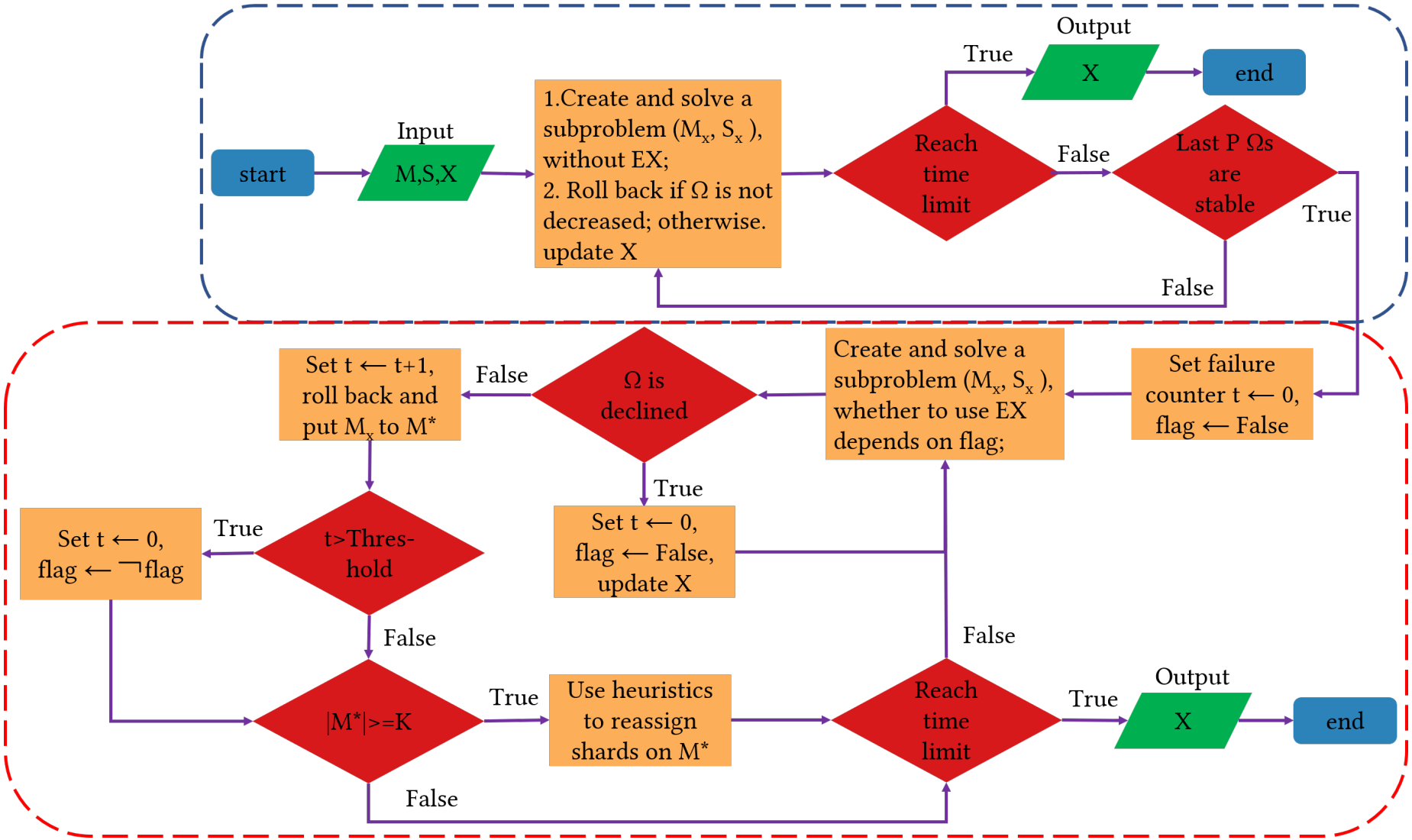
2. Use LNS to randomly create subproblems and solve them by IP and EX machines

   Apply linear search and local search to handle the unsuccessful subproblems



The core procedures of SRA in details

The flowchart:



Experiment

1. Input settings (synthetic and real) and baseline (noisy strategy based local search (NLS), SRA without EX machines)

Table: Summary of the synthetic input settings

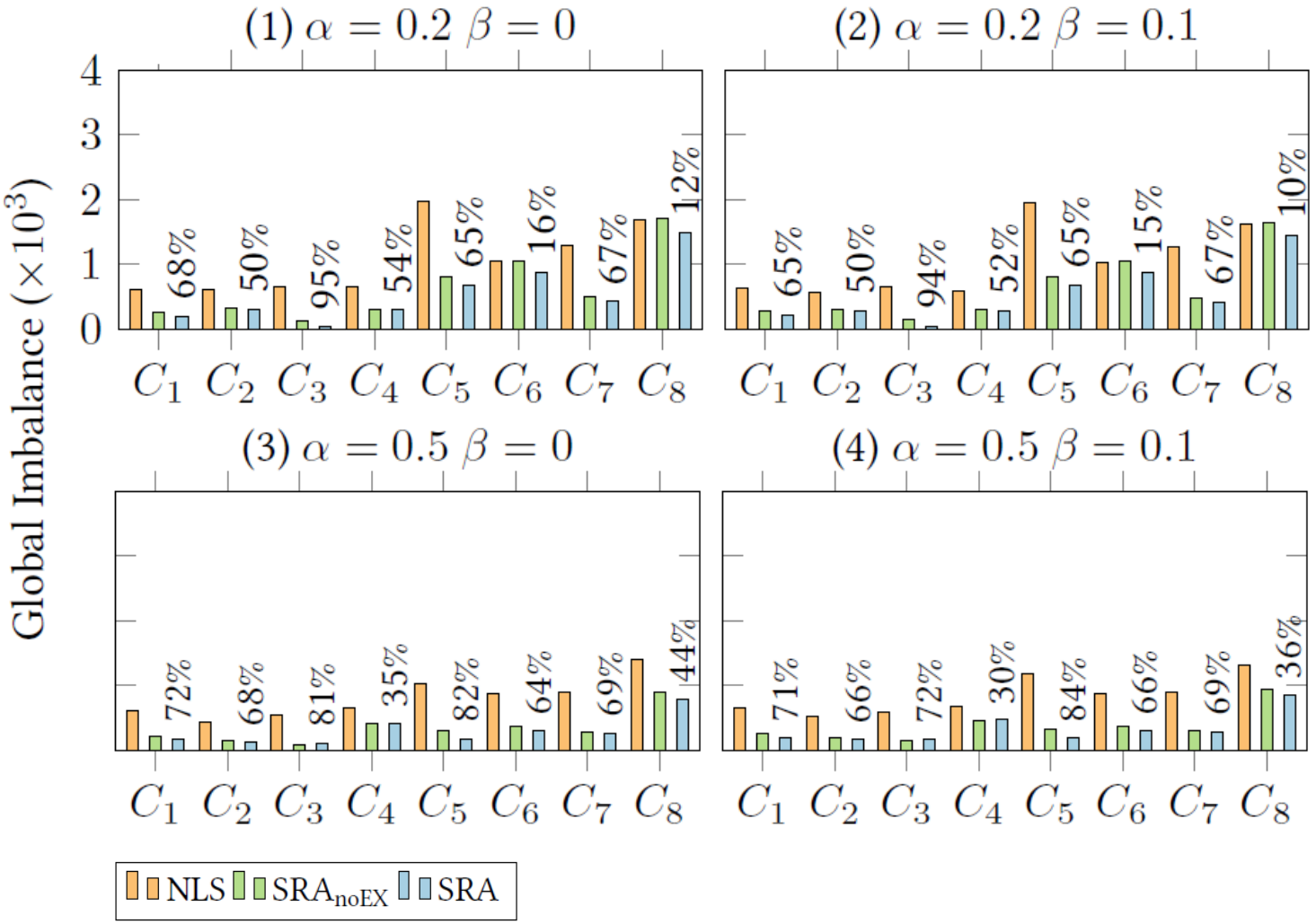
	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
r_1 range	S	S	L	L	S	S	L	L
r_2 range	S	L	S	L	S	L	S	L
machines	Homogeneous				Heterogeneous			
the No. shards ($\alpha = 0.2$)	14074	6281	6285	6121	9358	4572	4651	3882

Table: Summary of the real datasets DS_1 to DS_4

Instance ID	DS_1	DS_2	DS_3	DS_4
Machine type	Hetero	Hetero	Hetero	Hetero
Number of indexes	78	64	123	60
Number of shards	3072	2753	2936	3057
Number of machines	1816	1220	1987	1976
Machine mean Utl. (r_1)	0.50	0.74	0.66	0.42
Machine mean Utl. (r_2)	0.39	0.66	0.65	0.29

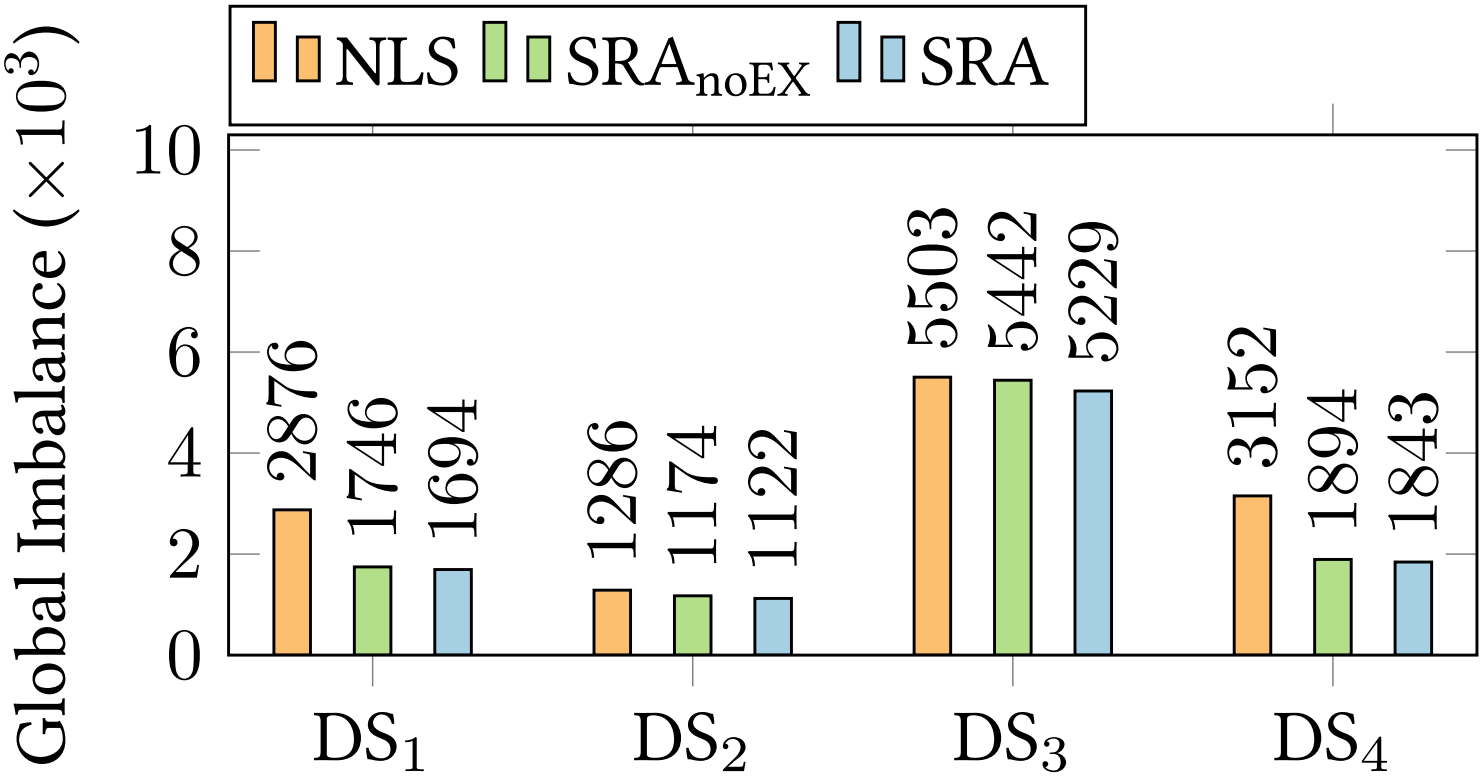
Experiment

2. Load balancing on synthetic datasets



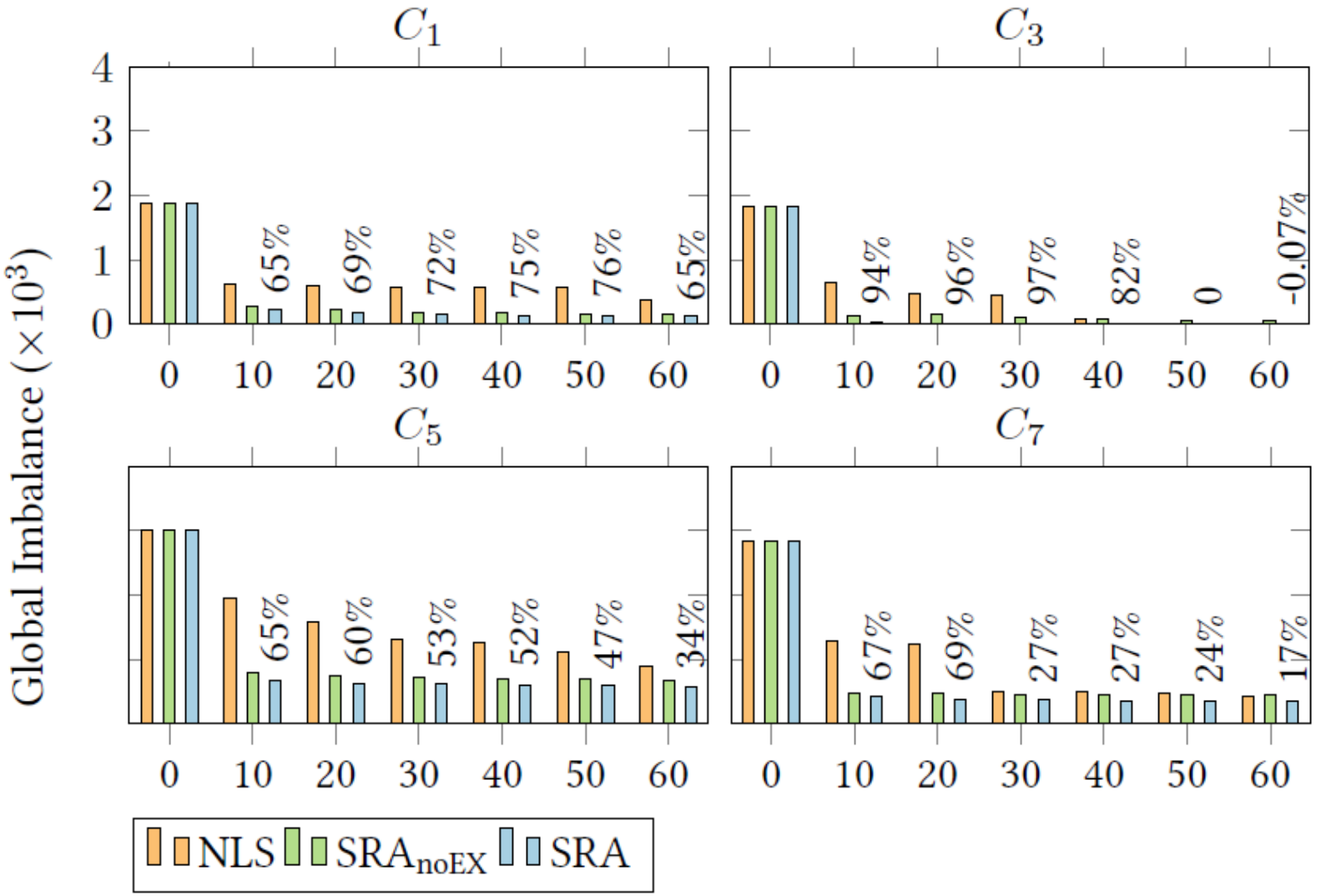
Experiment

3. Load balancing on real datasets



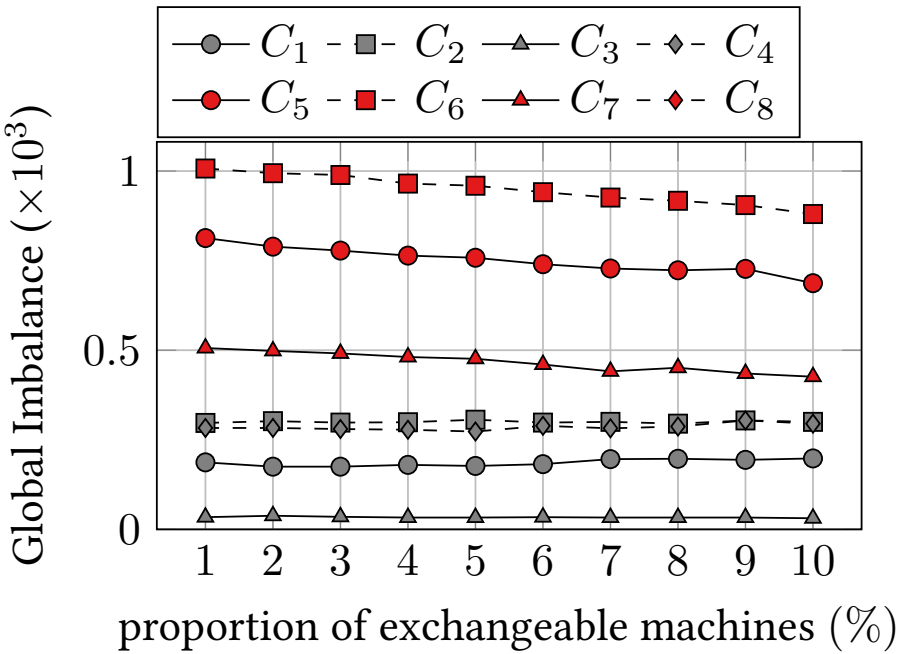
Experiment

4. Load balancing under different runtimes (unit: minutes)

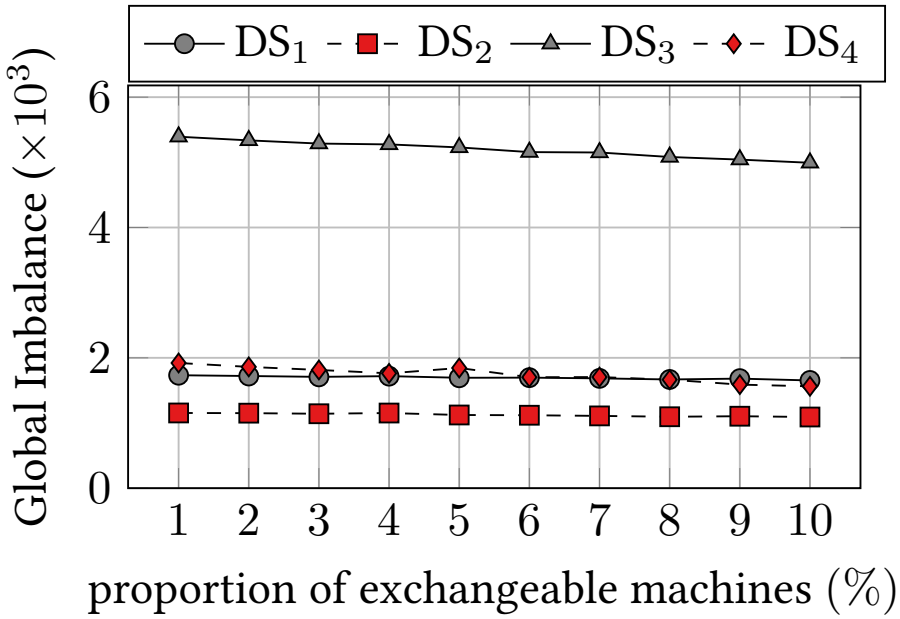


Experiment

5. Global imbalance and different No. EX. Machines on synthetic datasets

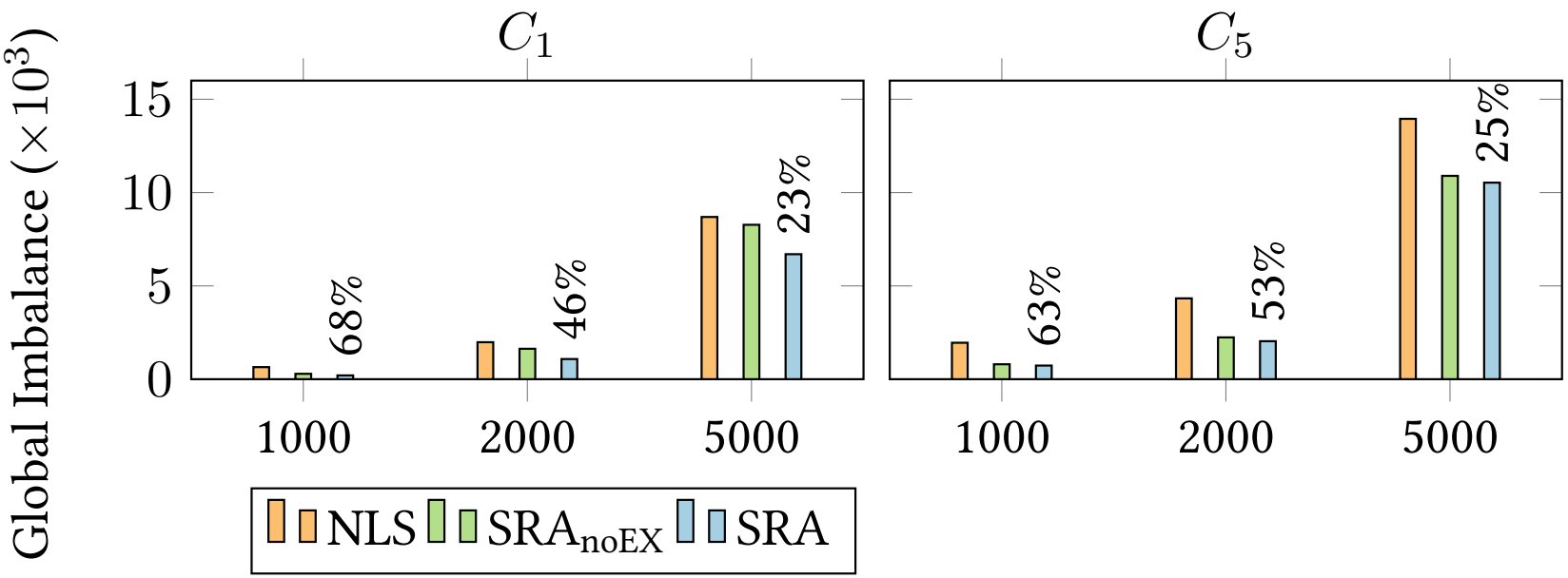


6. Global imbalance and different No. EX. Machines on real datasets



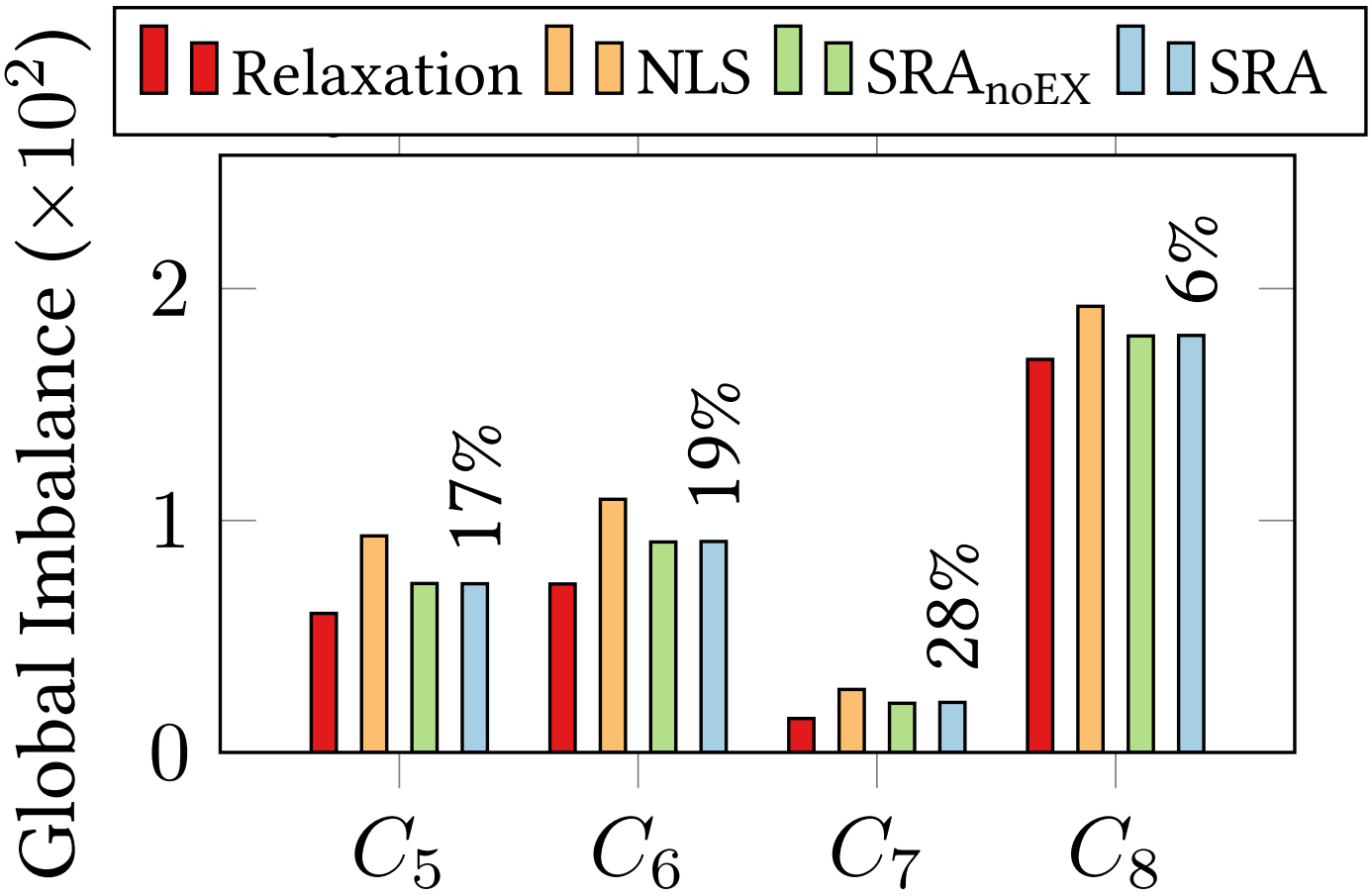
Experiment

7. Global imbalance under different No. leaf machines



Experiment

8. Comparing with the lower bound



Conclusion

- Use exchangeable machines and the SRA to handle the shard reassignment problem.
- There are two trade-offs.
- We have not deploy the SRA on a real search engine.

Thank You!