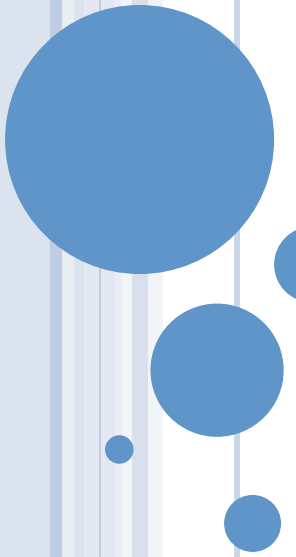


Final Exam Scheduler

4005-735-01 Parallel Computing I

G2. Team Kyz

Kevin Cheek, Yandong Wang, Ziyan Zhou

http://tinyurl.com/ritkyz

AGENDA

- Problem Description
- Solutions
- Approaches
  - Brute-Force
    - Sequential/Parallel Design, Performance Metrics, Investigation
  - Simulated Annealing (SA)
    - Paper Summary, Sequential/Parallel Design, Performance Metrics
  - Genetic Algorithm (GA)
    - Paper Summary, Sequential/Parallel Design, Performance Metrics
- Future Work
- Q&A

Final Exam Scheduling Problem

- Sections

0101-301-02	FINANCIAL ACCOUNTING	KEARNS,F		Open	39	38	TR	400P	55
0101-301-03	FINANCIAL ACCOUNTING	KEARNS,F		Close	40	40	TR	1200N	15
0101-301-71	FINANCIAL ACCOUNTING	EVANS,W	Open	40	38	T	600P	950P	12
0101-301-90	FINANCIAL ACCOUNTING	LEBOWITZ,P		Close	25	25	NA	ONLINE	CO
0101-302-01	MANAGEMENT ACCOUNTING	OLIVER,B		Open	40	39	TR	200P	35
0101-302-02	MANAGEMENT ACCOUNTING	DEY,R	*Open	40	38	MW	800A	950A	12
0101-345-71	ACCOUNTING INFO SYSTEMS	NEELY,M	*Open	28	17	W	600P	950P	12
0101-409-01	FINAN. REPT. & ANLYS. II	KEARNS,F			*Open	40	26	MW	12
0101-494-71	COST ACCTG TECH ORG	MORSE,W	*Open	15	9	W	600P	950P	12
0101-523-90	ADVANCED TAXATION	KLEIN,R	Open	25	19	NA	ONLINE	COURSE	NA
0101-540-71	ADVANCED ACCOUNTING	OLIVER,B		*Open	20	12	M	600P	95
0101-554-01	FORENSIC&FRAUD ACCOUNTNG	KLEIN,R	Open	12	0	W	400P	55
0101-703-71	ACCTG FOR DECISION MAKER	MORSE,W	*Open	35	31	T	600P	92
0101-703-90	ACCTG FOR DECISION MAKER	KLEIN,R	Open	25	23	NA	ONLINE	CO
0101-704-71	CORP FINANCIAL REPT I	KARIM,K	Open	35	15	W	600P	920P	12
0101-706-71	COST MANAGEMENT	KARIM,K	Open	34	16	T	600P	920P	12	11

- Slots
  - MTWRF 8:00-10:00 10:15-12:15p 12:30p-14:30p 14:45p-16:45p
- Students

SOLUTIONS

- Three different experimental approaches
  - Brute-force
  - Simulated Annealing (SA)
  - Genetic Algorithm (GA)

Ranking Criteria

- Variant of Exams distributed across exam slots
- Each student has a vote:
  - Exam Conflict: **POSITIVE_INFINITY**
  - Three or more exam per day: **5.0**
  - Consecutive Exams: **2.0**
  - Friday Exams: **0.5**

EXAMPLES

Schedule for Student 68997-3266 John Doe 1

	MON	TUE	WED	THU	FRI
8:00am	N/A	N/A	N/A	4003-233-70	4003-233-45
10:15am	N/A	N/A	N/A	4003-420-02	N/A
12:30pm	N/A	N/A	4003-232-01	N/A	4003-713-70
2:45pm	N/A	N/A	4003-334-40	N/A	N/A

Schedule for Student 54331-5905 John Doe 2

	MON	TUE	WED	THU	FRI
8:00am	4003-231-01	N/A	N/A	4003-334-02	N/A
10:15am	N/A	4003-232-55	N/A	N/A	N/A
12:30pm	N/A	N/A	4003-233-03	N/A	N/A
2:45pm	N/A	N/A	N/A	4003-410-01	N/A

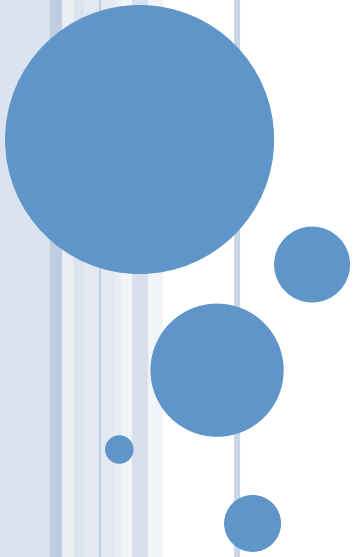
PACKAGE COMMON

- Objects
  - Section
  - Student
  - Schedule
- Utilities
  - Resources
    - Provide data loading, global access to resources
  - Random.generateSchedule()
    - Randomly generate a schedule
  - Ranker.rank(Schedule)
    - Rank the given schedule
  - GenerateData Program

Student	Section	Schedule
• getId() • getName()	• getId() • getTitle() • getProfessor() • …	• Map<SectionId, Slot> • getRank()

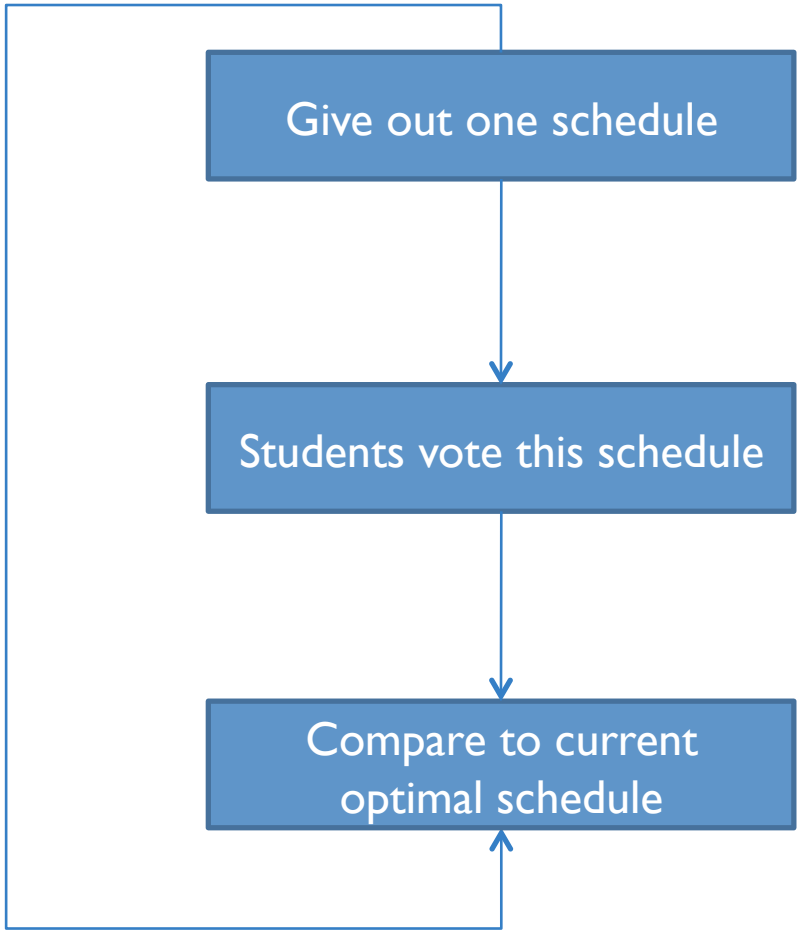
Test Data

N	Courses	Students	Student Load
5	5	2,00	4
10	10	4,00	4
20	20	8,00	4
50	50	2,000	4
100	100	4,000	4
200	200	8,000	4

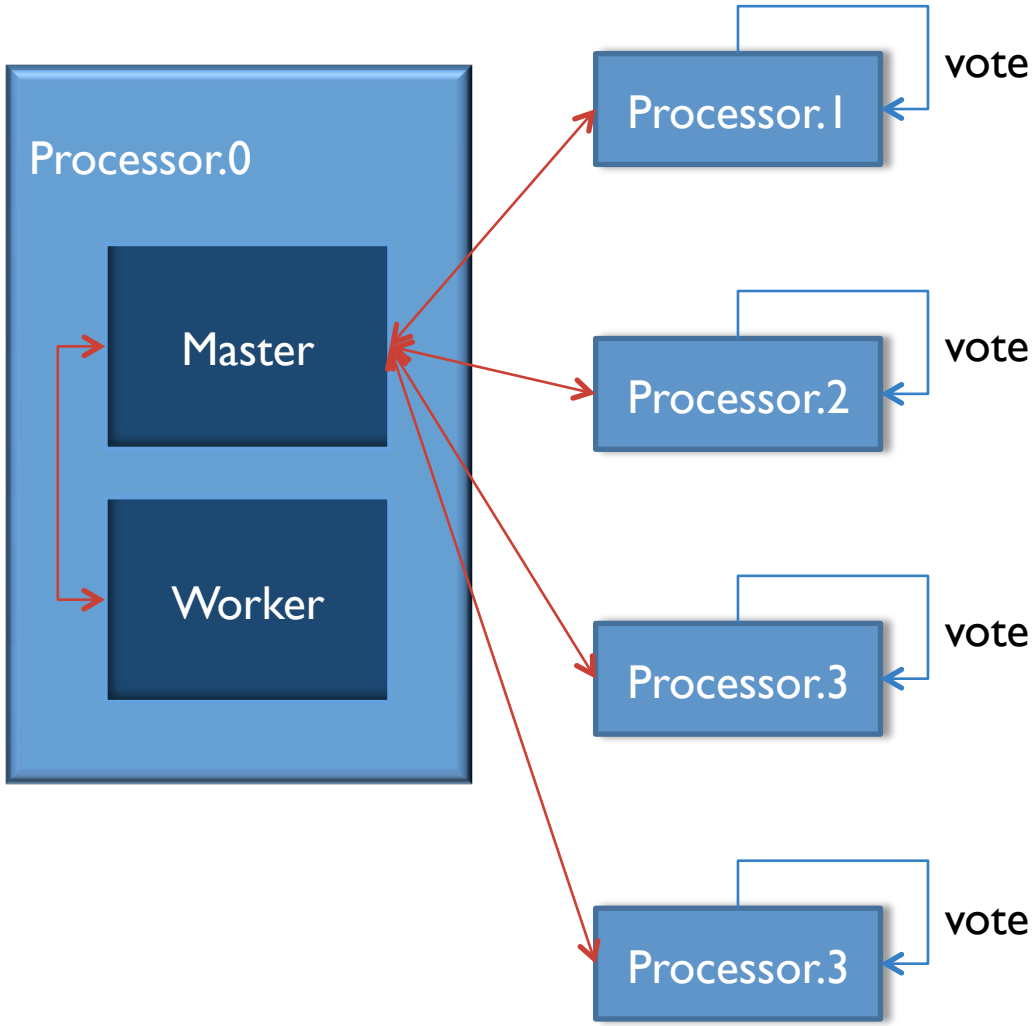


Brute-Force Approach

BRUTE-FORCE: SEQUENTIAL



BRUTE-FORCE: FIRST CLUSTER PARALLEL DESIGN

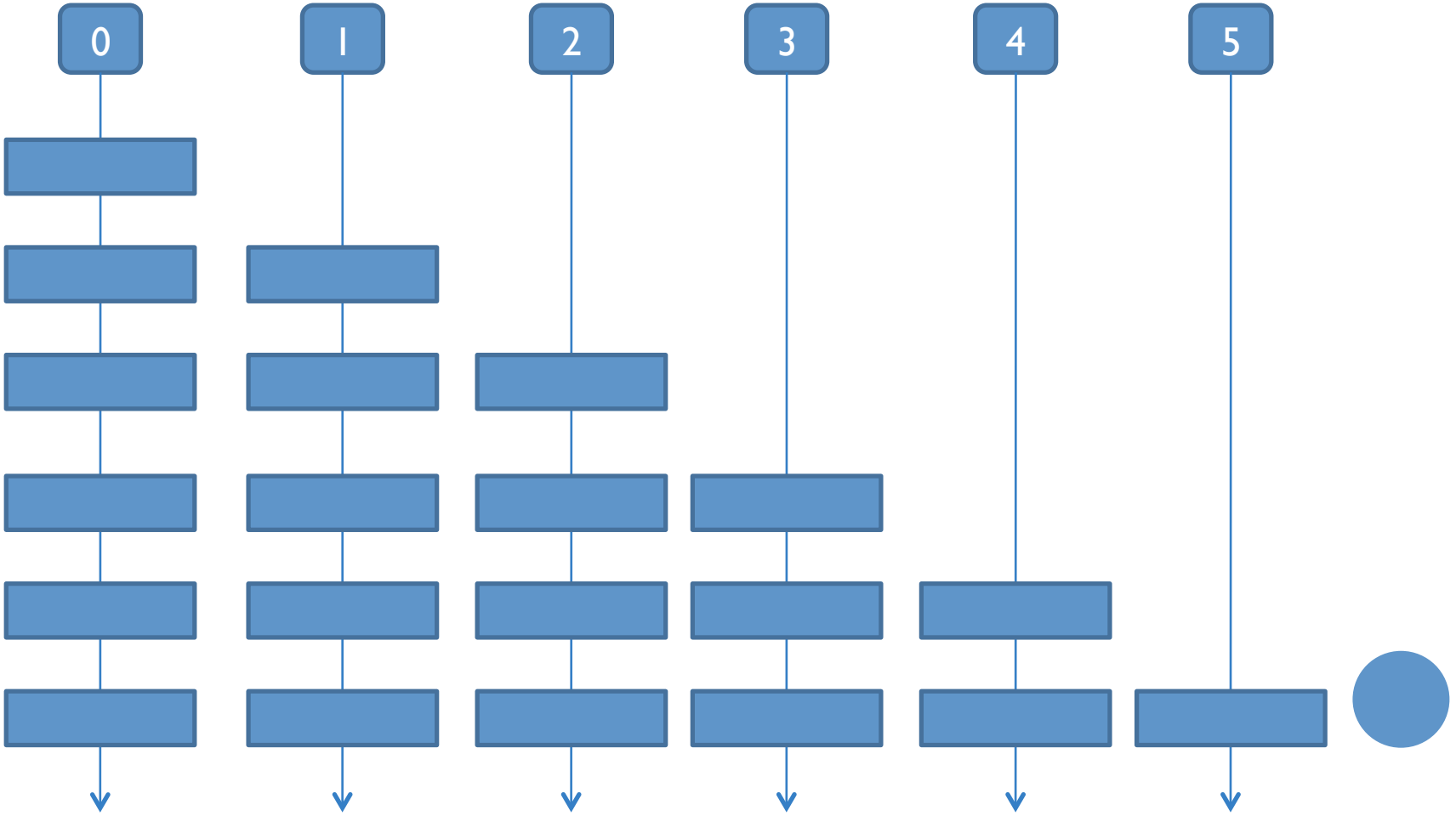


Master:
-Schedule []
-Schedule Generator
-All the information of courses
- ObjectBuf< Schedule > buf = ObjectBuf.*sliceBuffer(Schedule[], Range);*

Worker:
- Schedule[]
- Optimal Schedule
- All the information of students

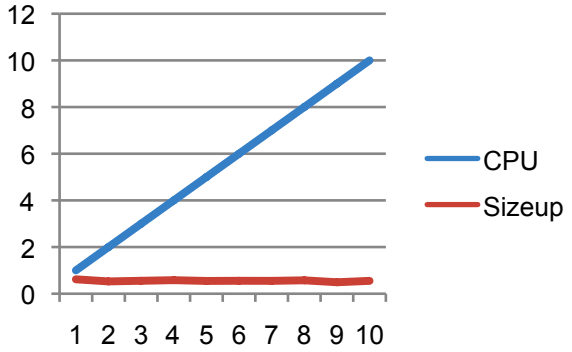
Communication Method:
-world.receive(pro, buf);
-world.send(pro,tag, buf);
- CommRequest request; *world.receive(0, tag, buf, request);*

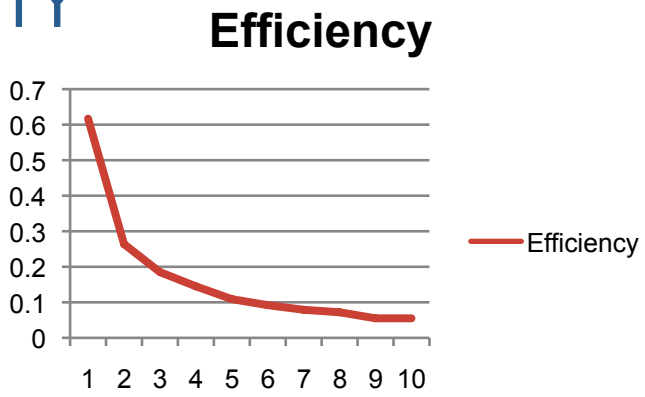
Brute-Force: First Design Expected Sizeup



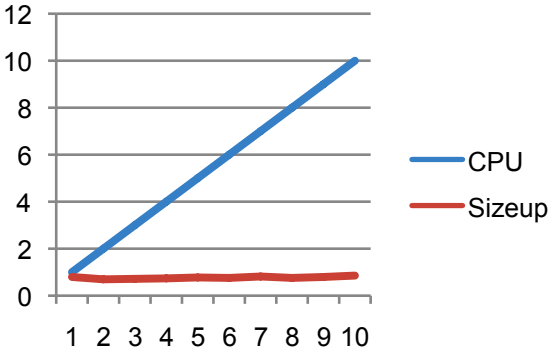
BRUTE-FORCE: RELENTLESS REALITY

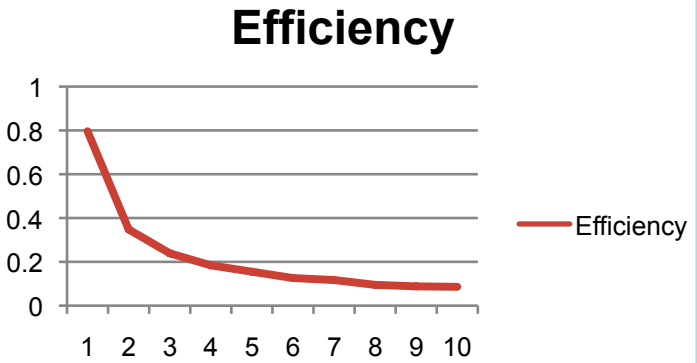
- (1): 10000 msc, pieces size 50



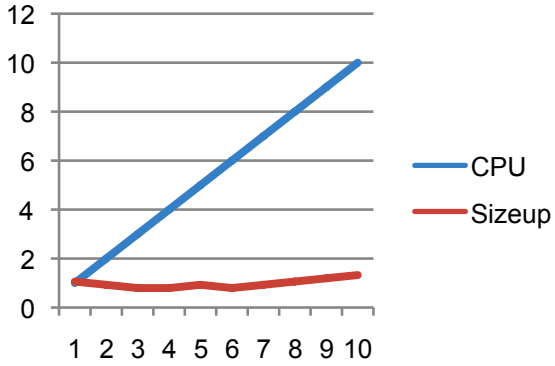


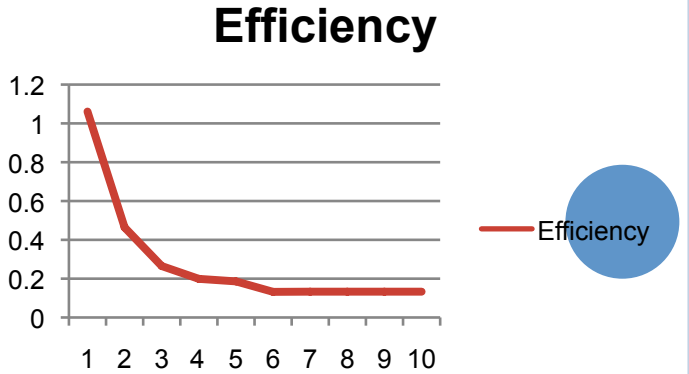
- (2): 10000 msc, Pieces size 150



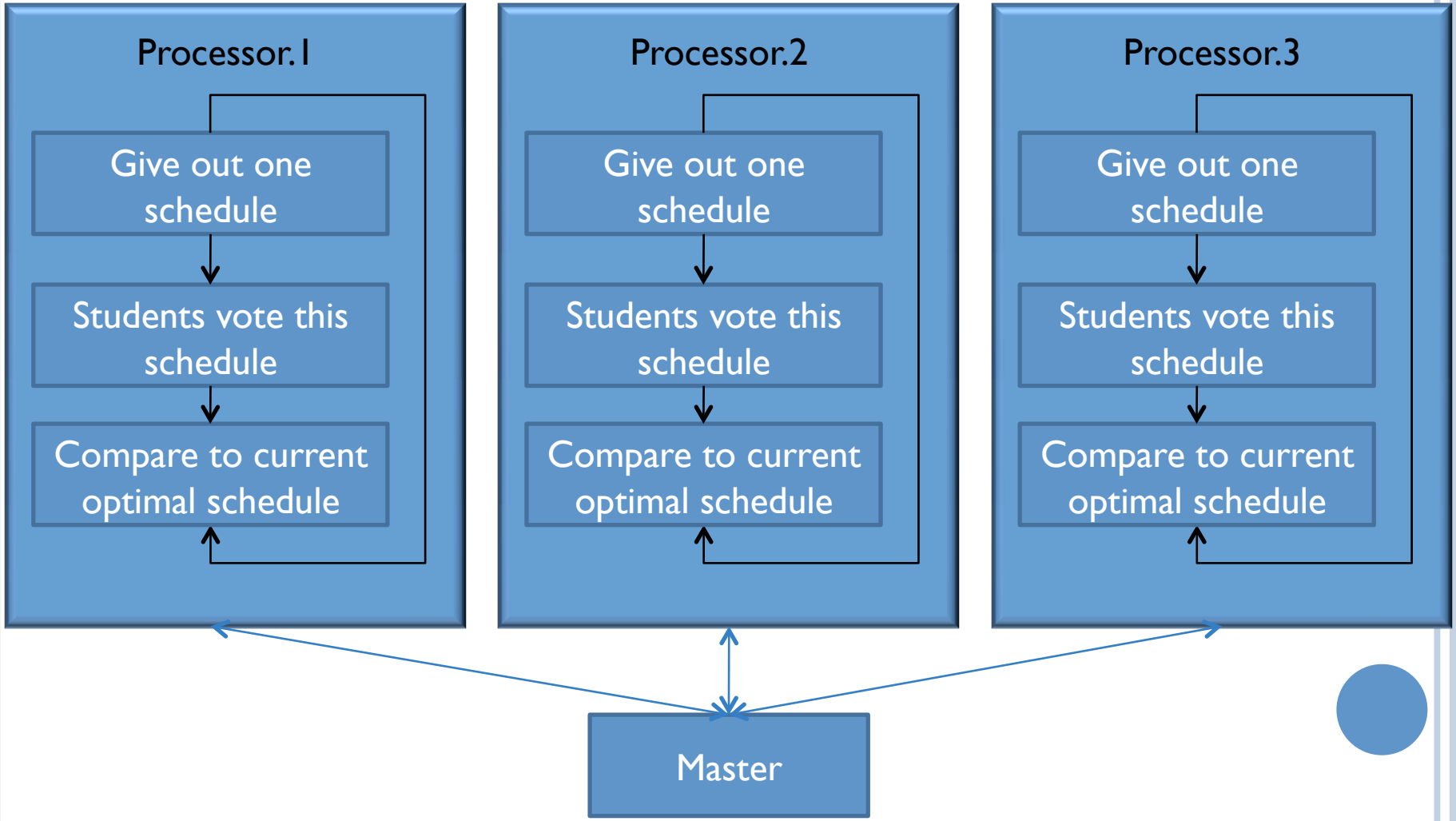


- (3): 10000 msc Pieces size 1000



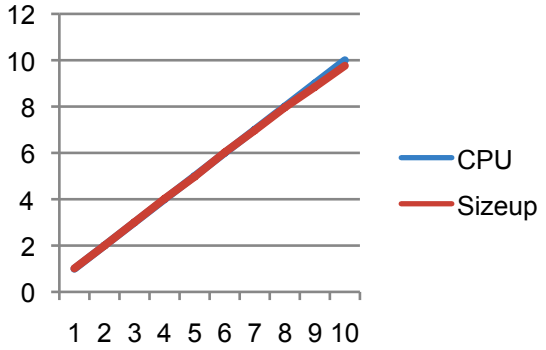


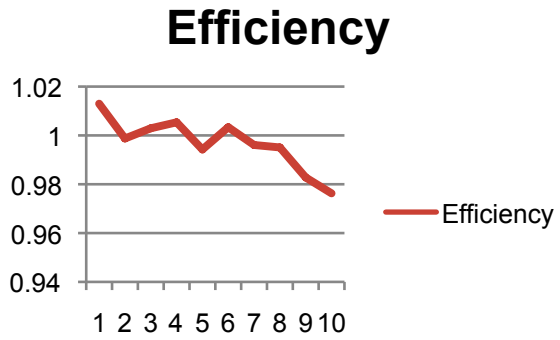
BRUTE-FORCE: REDESIGN



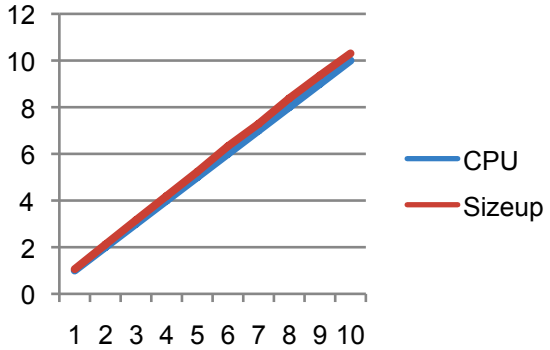
BRUTE-FORCE: SECOND DESIGN RESULT

- (1): 10000 msc

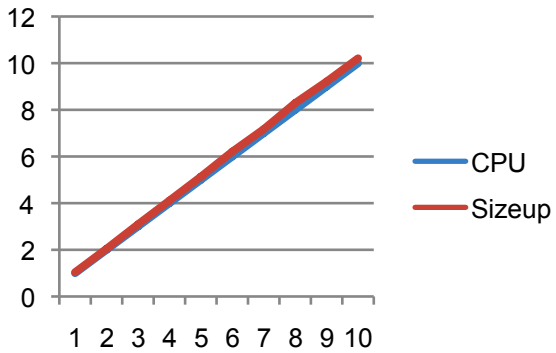


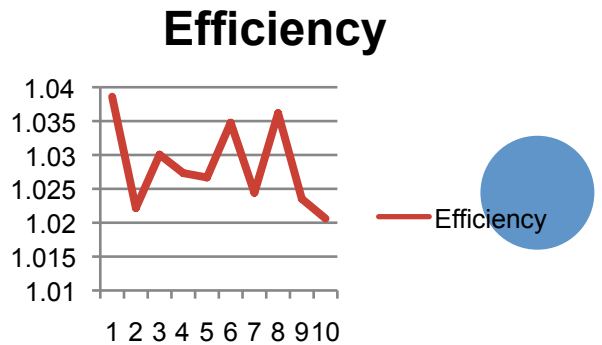


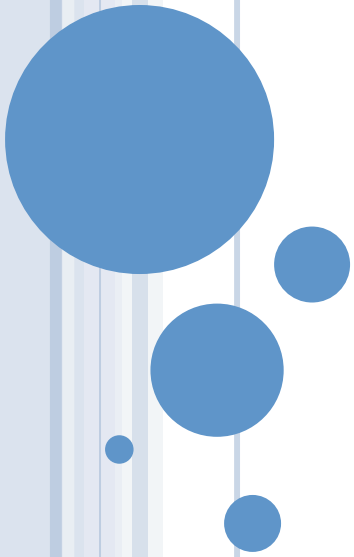
- (2): 20000 msc



- (3): 30000 msc







Simulated Annealing (SA)

Paper #1: Stochastic Search Algorithms for Exam Scheduling

By Mansour and Timany, published in International Journal of Computational Intelligence Research in 2007

- Exam scheduling as a modified weighted graph coloring problem
- Objective Functions
- Simulated Annealing Algorithm (SA)
- Results Reference

SA: SEQUENTIAL

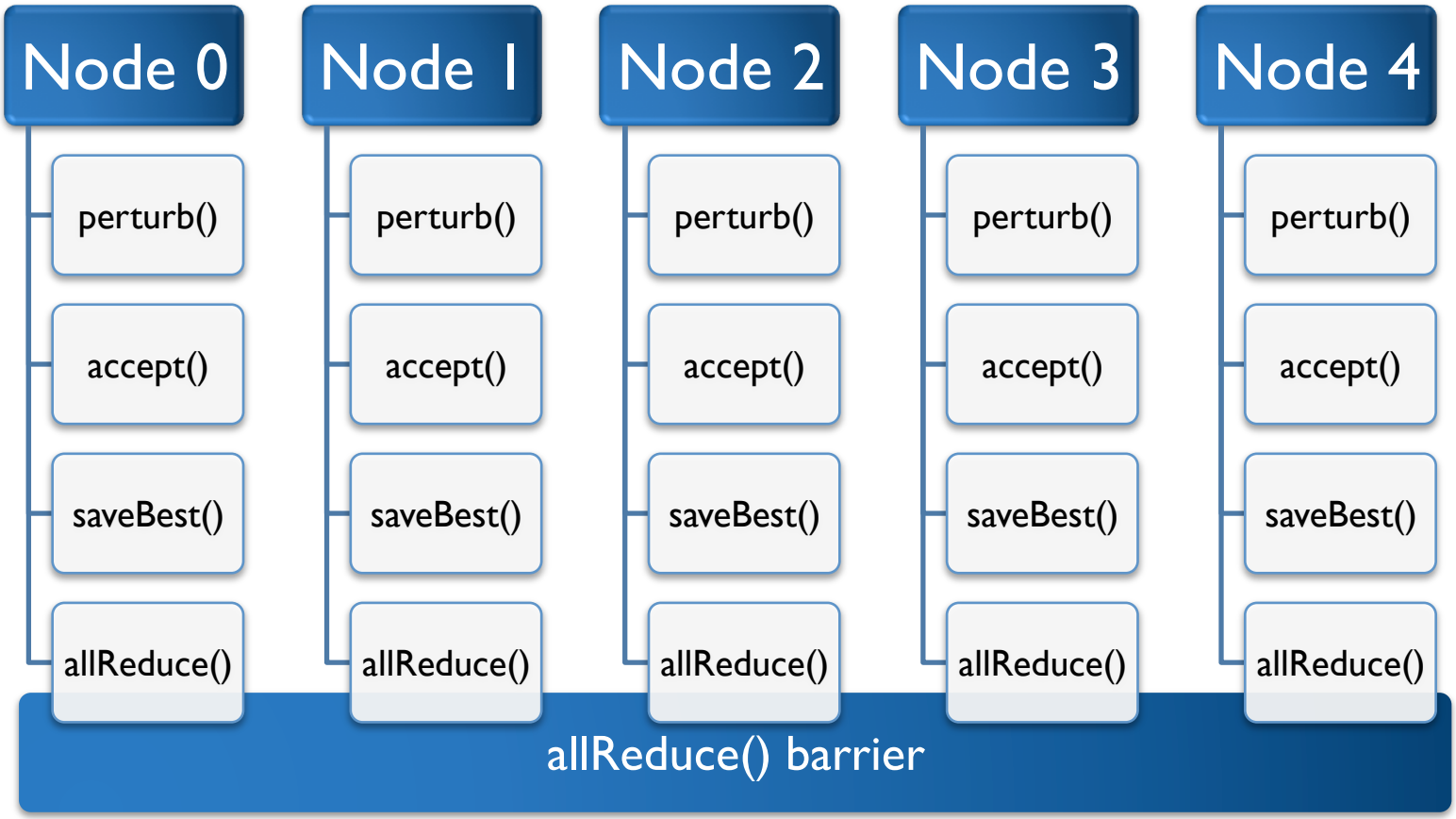
```
Initial configuration = random;
Initial temperature T(0) = 0.93; //high initial acceptance
Freezing temperature Tf = 2^-30; //uphill moves impossible

while(T(i) > Tf and not converged) {
  repeat (# of slots) * (# of exams) times
   generate_function();
  save_best_so_far(); //smallest OF1
  T(i) = phi * T(i); //phi = 0.95
}

function generate_function() {
  perturb(); //randomly change one exam's slot to another slot
  if (ΔOF1 <= 0)
   accept();
  else if (ramdon() < e^(-ΔOF1/T(i)))
   accept(); //accept with a probability
}
```

SA: Result Comparison

- Problem Size:
  - Exams: 336
  - Students 2456
  - Enrolments: 9550
  - Slots: 20
- Notations:
  - SE: Simultaneous Exam (conflict)
  - CE: Consecutive Exams
  - ME: Multiple Exam (3 or more on the same day)

	SE	CE	ME	Rank
Our	0	24	351	2594.66
Their	3	336	572	N/A

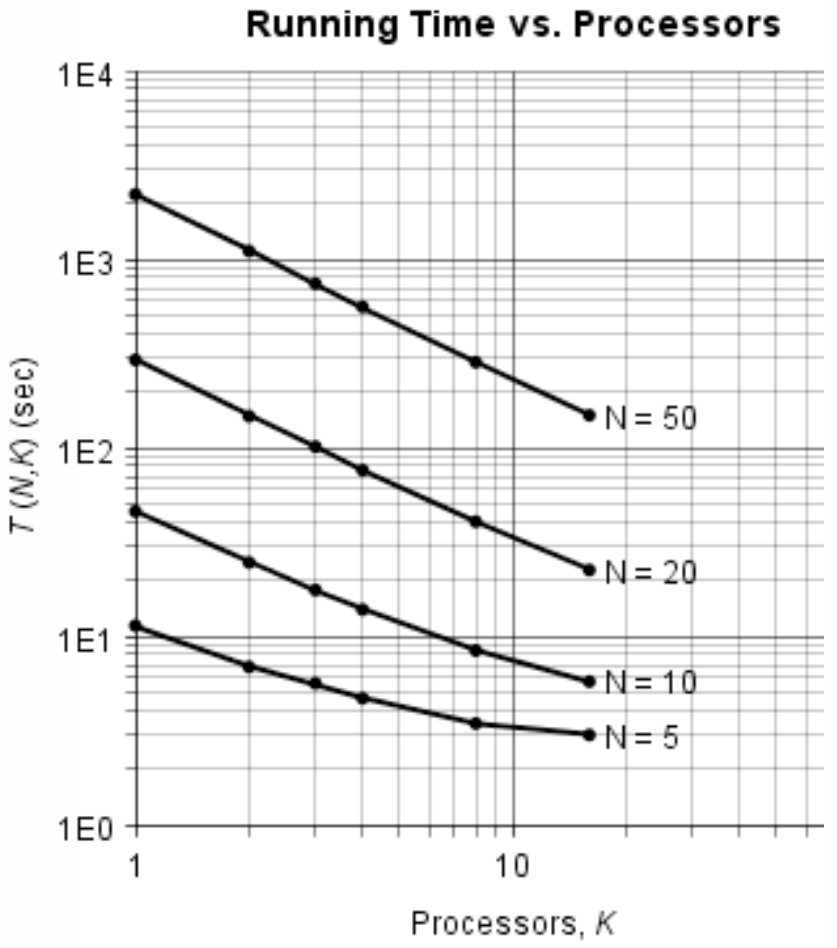
SA: PARALLEL

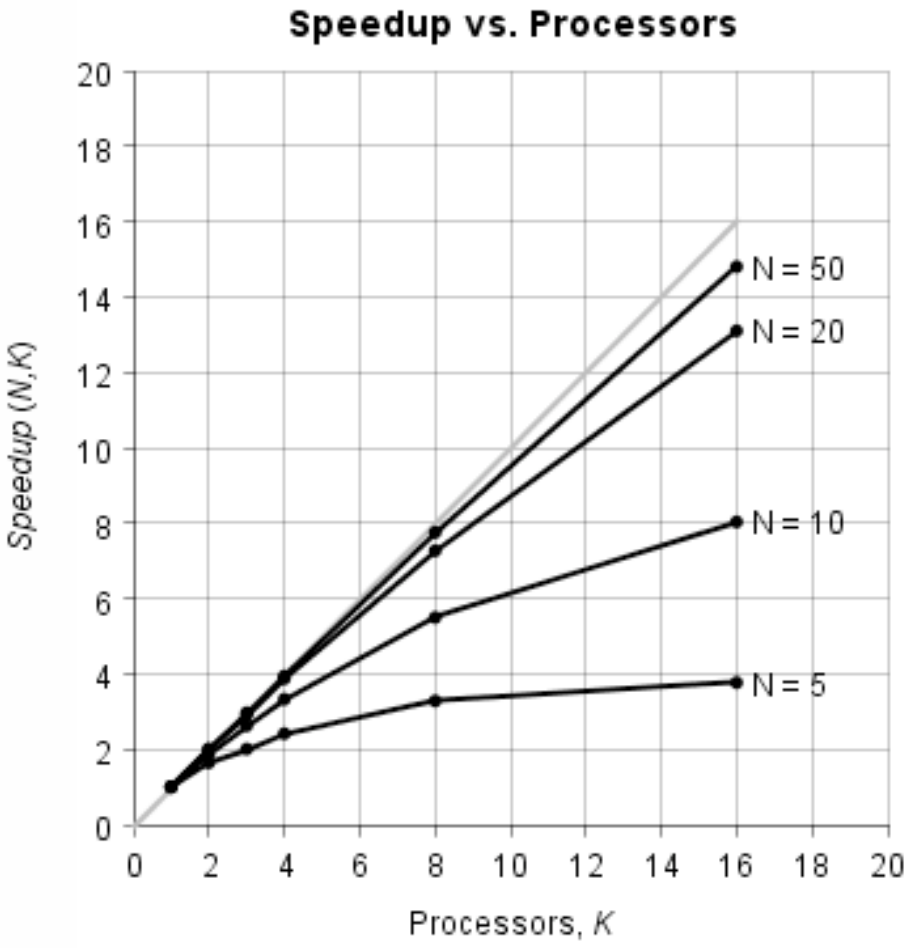

Node 0
Node 1
Node 2
Node 3
Node 4
perturb()
accept()
saveBest()
allReduce()
allReduce() barrier

SA: REDUCTION

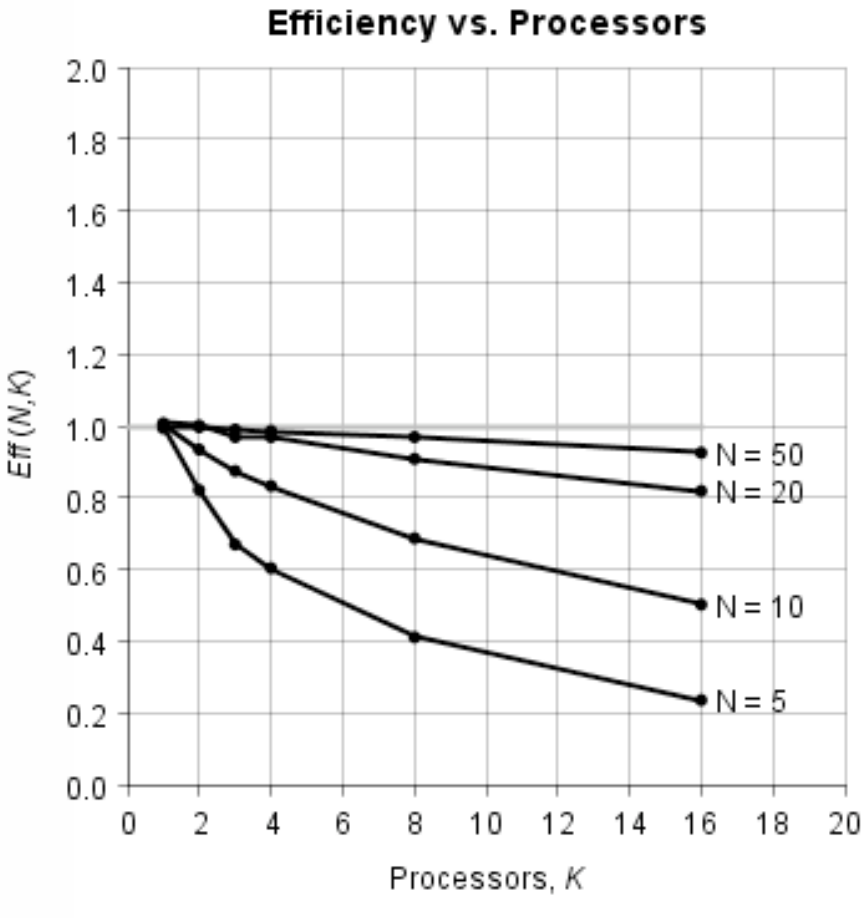
```
final ObjectItemBuf<Schedule> buf =
  ObjectItemBuf.buffer(sa.getBest());
Comm.world().allReduce(0, buf, new ObjectOp<Schedule>() {
  @Override
  public Schedule op(final Schedule x, final Schedule y) {
   if(x.getRank() < y.getRank())
        return x;
   return y;
  }
});
sa.setBest(buf.item);
```

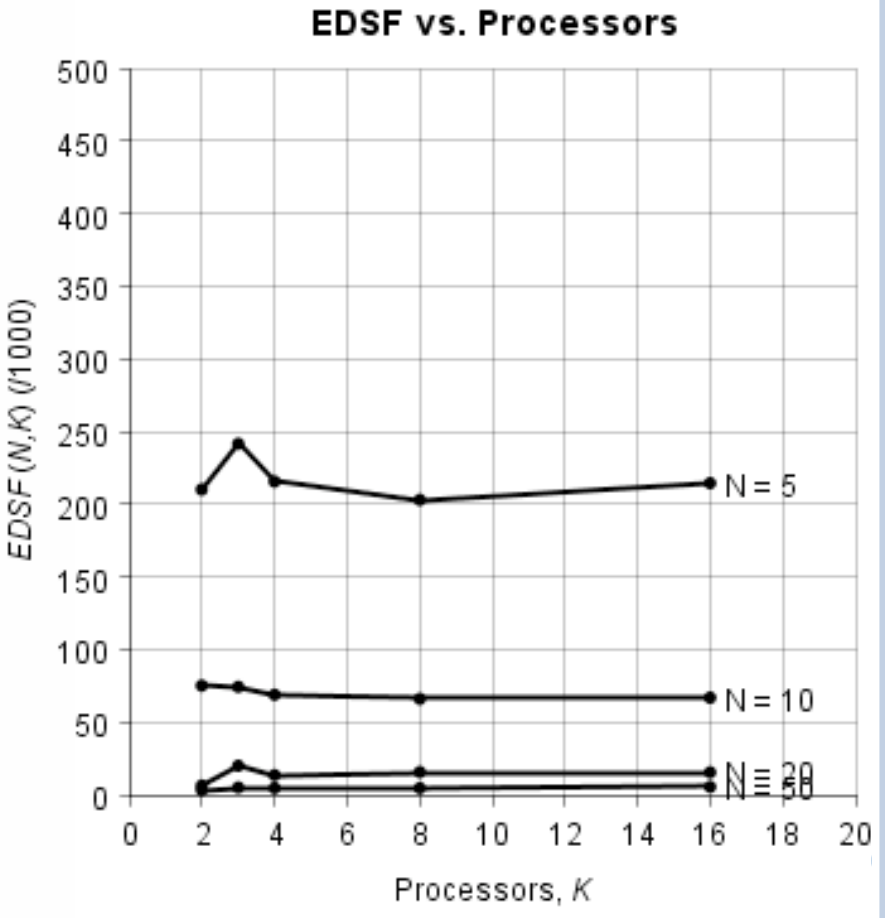
SA: Performance Metrics



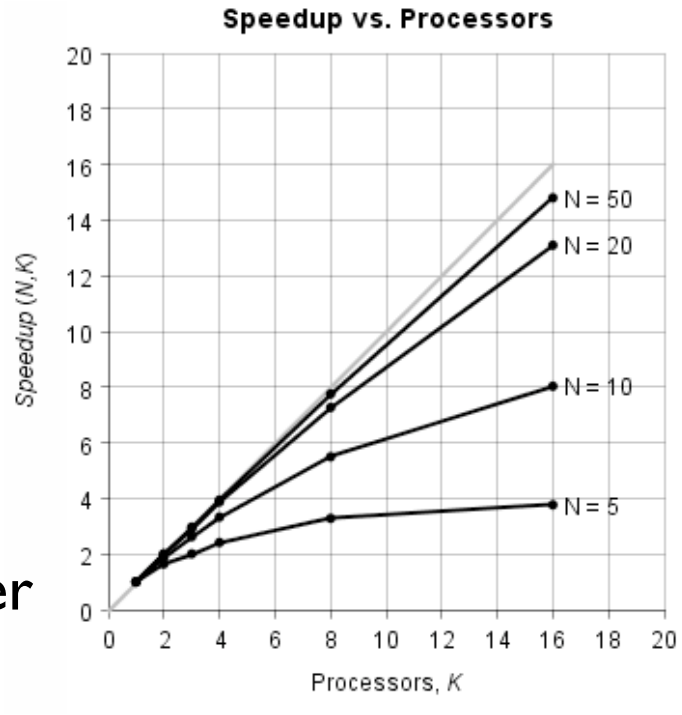


SA: Performance Metrics

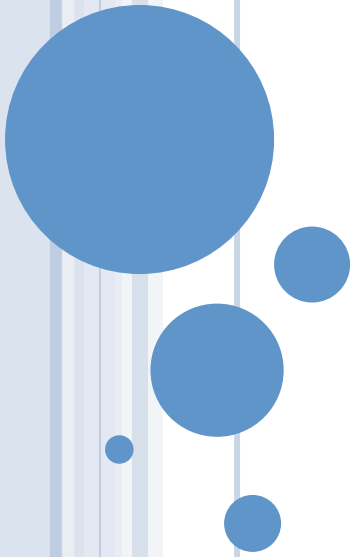




SA: INVESTIGATION



- Bottlenecks
  - Barrier makes it hard to load balance
  - Communication overhead
- Better efficiency as N becomes larger
  - # of temperature fixed
  - Amount of communication fixed
  - Perturb and ranking takes significantly longer, $O(N^2)$
- Limitation
  - When N=100, sequential program takes more than an hour to run on a cluster node, K=16 takes 613,783 ms
  - When N=200, K=16, it takes 2530,561 ms to finish



Genetic Algorithm

GA: GENETIC ALGORITHM

```
Randomly generate initial population size POP;
Evaluate fitness of individuals;

repeat {
  rank individuals and allocate reproduction trials;

  for i=1 to POP step 2 {
   randomly select two parents from list;
   apply crossover and mutation;
  }

  apply hill-climbing to offspring //hybridization

  evaluate fitness of offspring;
  save_best_so_far();

} until converge
```

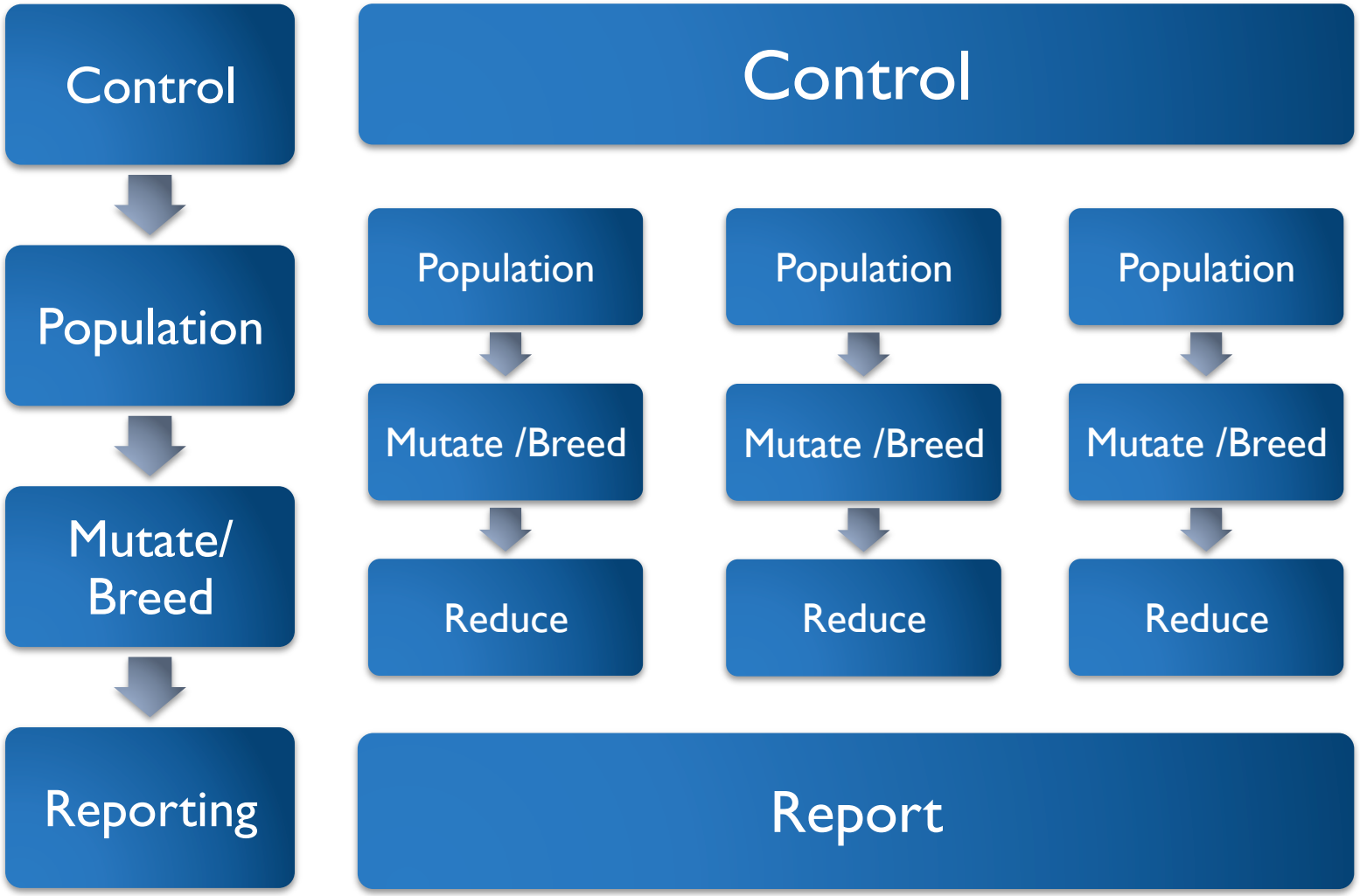
GA: PAPERS

- Used 3 Papers:
  - Genetic Algorithm Taxonomy by Mariusz Nowostawski and Riccardo Poli
  - Stochastic Search Algorithms for Exam Scheduling by Nashat Massour and Mazen Timany
  - Parallel Genetic Algorithms for the Hybrid Flow shop Scheduling Problem by K. Belkadi, M. Gourgand, and M. Benyettou

GA: Paper Findings

- Findings
  - Cross-over rate
  - Mutation rate
  - Number of mutations
  - Selection of individuals for reproduction
  - Replacement strategies
  - Convergence Detection
  - Population size
  - Migration Topologies

GA: Sequential and Parallel Design



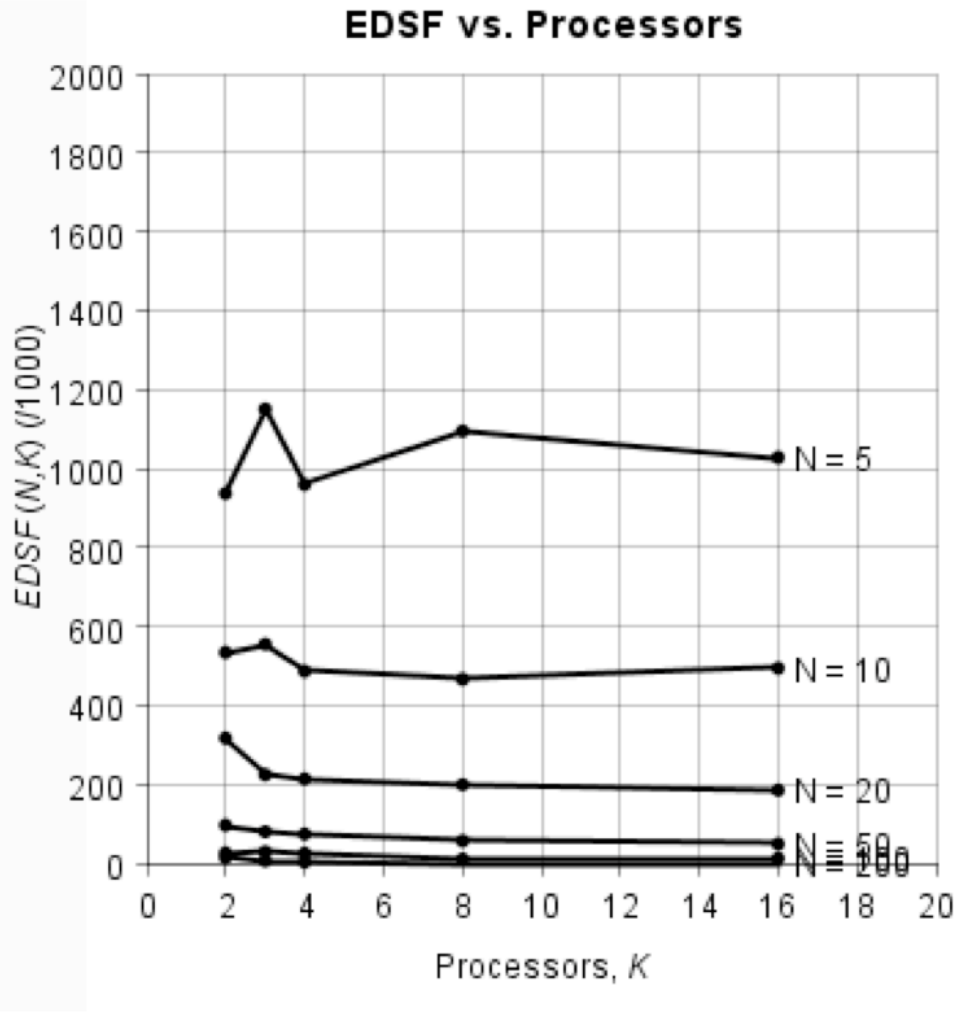
GA: Result Comparison

- Problem Size:
  - Exams: 336
  - Students 2456
  - Enrolments: 9550
  - Slots: 20

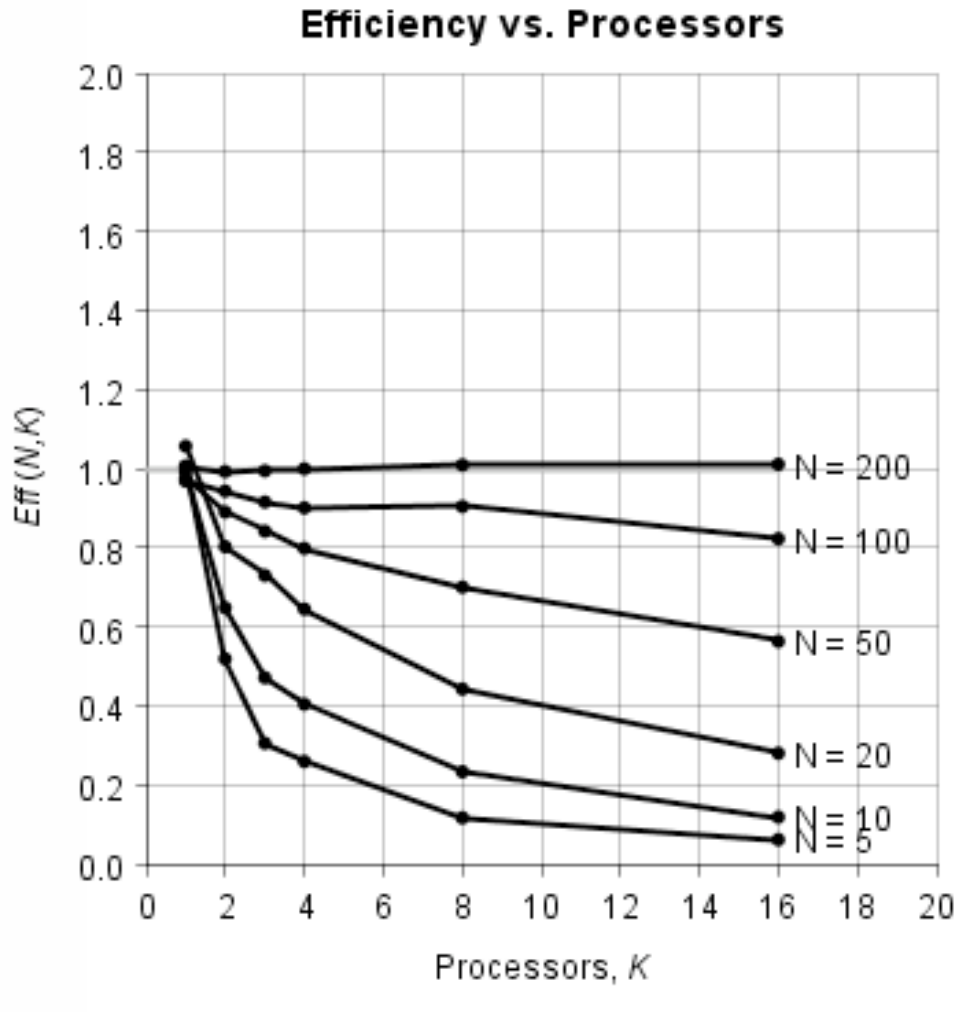
	SE	CE	ME	Rank
Our GA	0	19	275	2283.26
Their GA	4	308	537	N/A
Our SA	0	24	351	2594.66
Their SA	3	336	572	N/A

- Notations:
  - SE: Simultaneous Exam (conflict)
  - CE: Consecutive Exams
  - ME: Multiple Exam (3 or more on the same day)

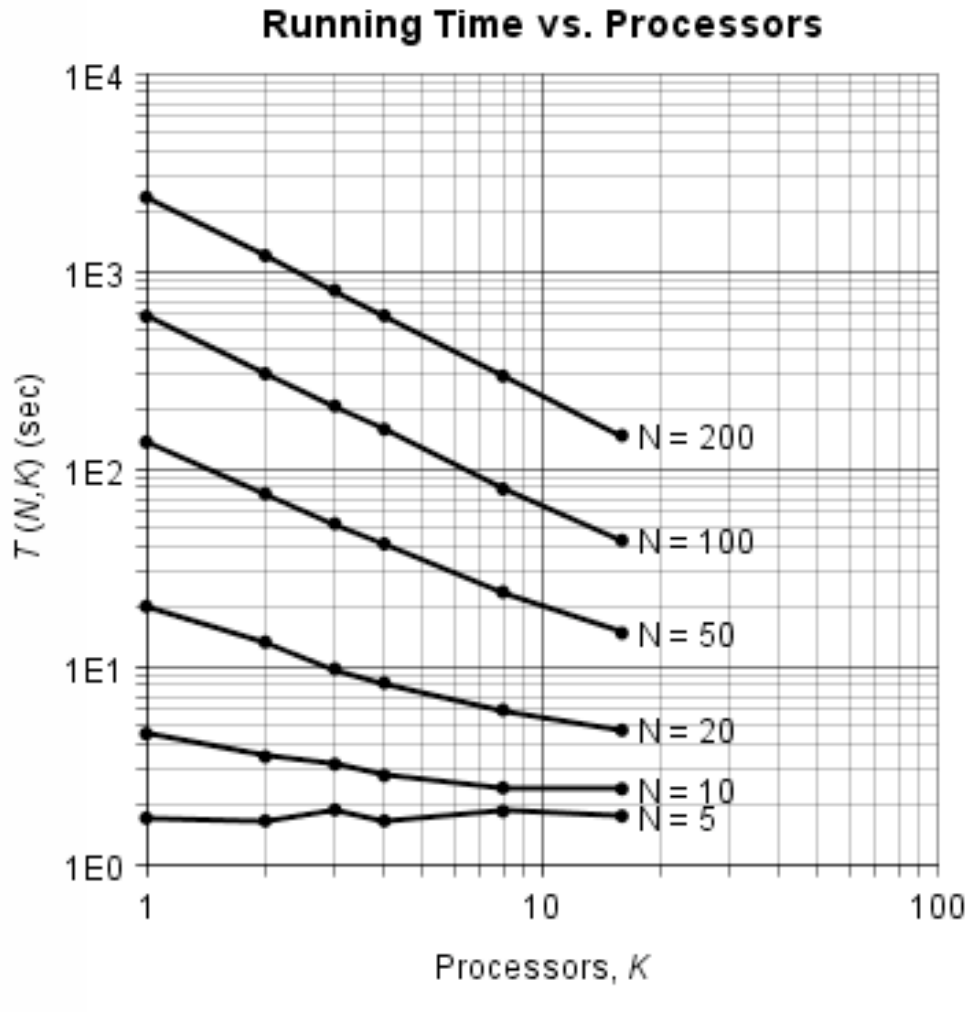
GA: PERFORMANCE



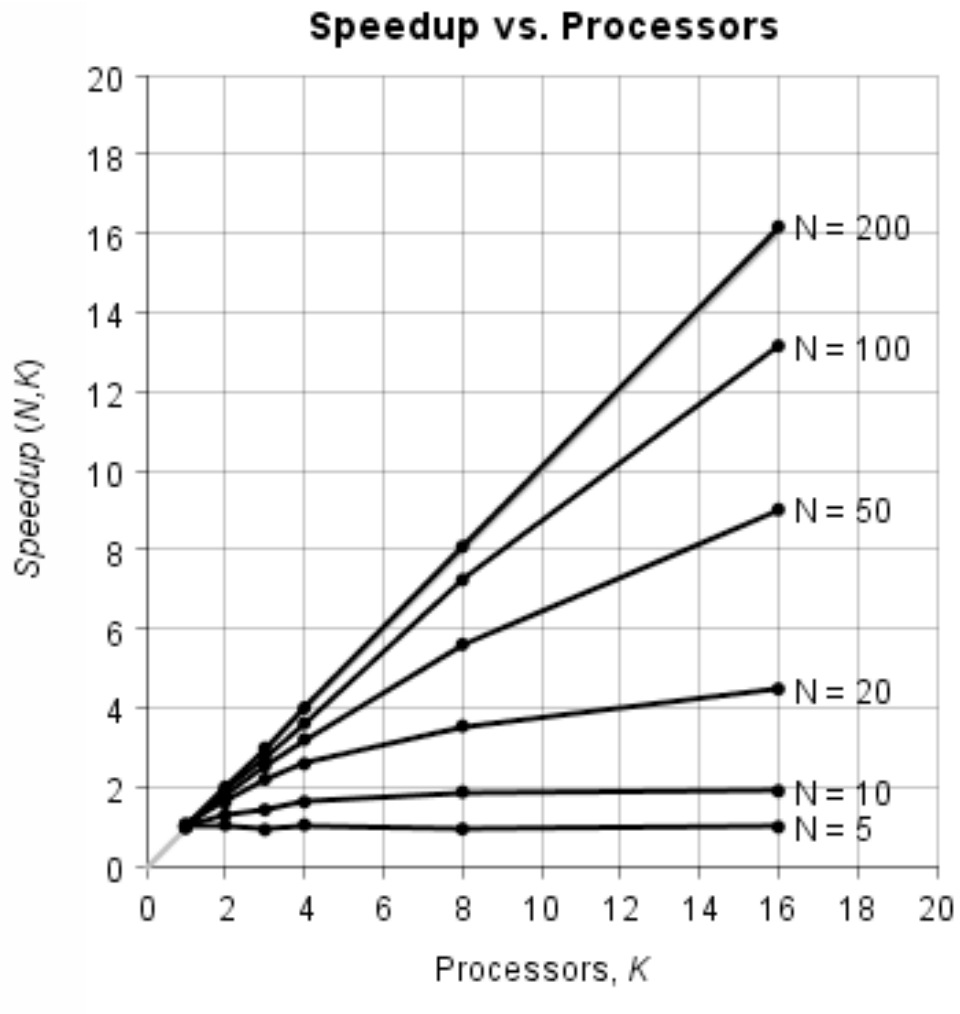
GA: PERFORMANCE



GA: PERFORMANCE



GA: PERFORMANCE



GA: Investigation

- Investigation Results and Problems:
  - It works!
  - Convergence Problems
  - Parameter Issues

Future Work

- Increased Problem Complexity
  - RIT: 2500 courses, 16,000 students
- More parameter for Ranker
- Minimize the schedule structure
  - Use only 32-bit integers instead of Section ID String
- Focus on Genetic Algorithm (GA)
  - True Migration
  - Better Convergence Detection
  - Solution Detection
  - Load Balancing with Convergence

Thank You for Listening

- G2. Team Kyz
  - Kevin Cheek
  - Yandong Wang
  - Ziyan Zhou
- Visit us at: http://tinyurl.com/ritkyz