Avoiding Global Synchronization for High-Performance Evolutionary Computation

Yan Liu[1,2] and Wendy K. Tam Cho[1,3,4,5,6]

[1] National Center for Supercomputing Applications (NCSA)

[2] Department of Geography and Geographic Information Science

[3] Department of Political Science

[4] Department of Statistics

[5] College of Law

[6] Department of Mathematics

University of Illinois at Urbana-Champaign

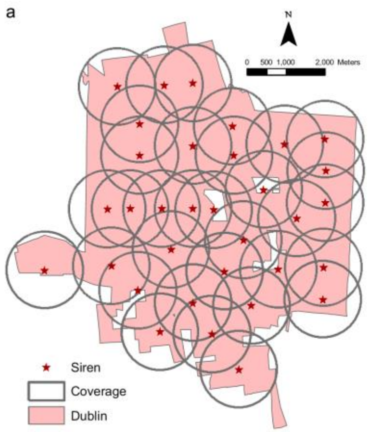
Spatial Optimization (SO)

- Combinatorial optimization

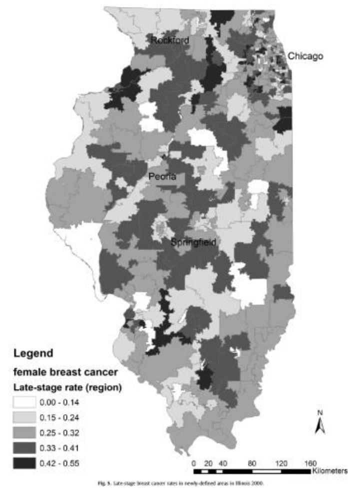
$$\text{minimize} \quad f(x) = cx$$
subject to:
$$Ax <= b$$
x is integer vector

- Spatial elements (Tong & Murray 2012)
  - In problem formulation:
    - Variables, coefficients, functions, constraints
  - Distance, adjacency, contiguity, containment, intersection, shape, districts, pattern, …
- Large optimization in practice
  - Political redistricting at census block level (11,155,486 blocks)
  - Maximum coverage with a large number of facility sites
  - Grid-based optimization on high-resolution rasters
  - …

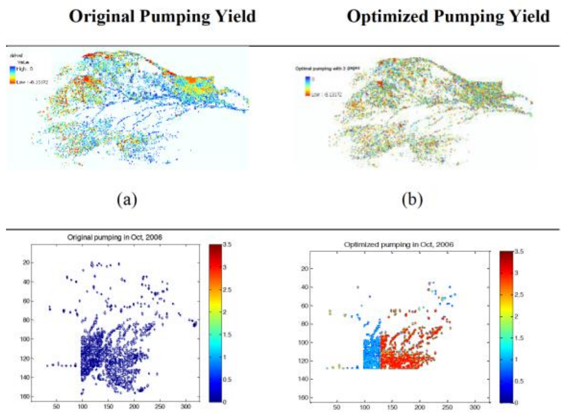
Spatial Optimization Problems



(Maximum Coverage, Wei & Murray 2015)
Location Modeling



(Breast Cancer Risk, Wang *et al*. 2012)
Public Health

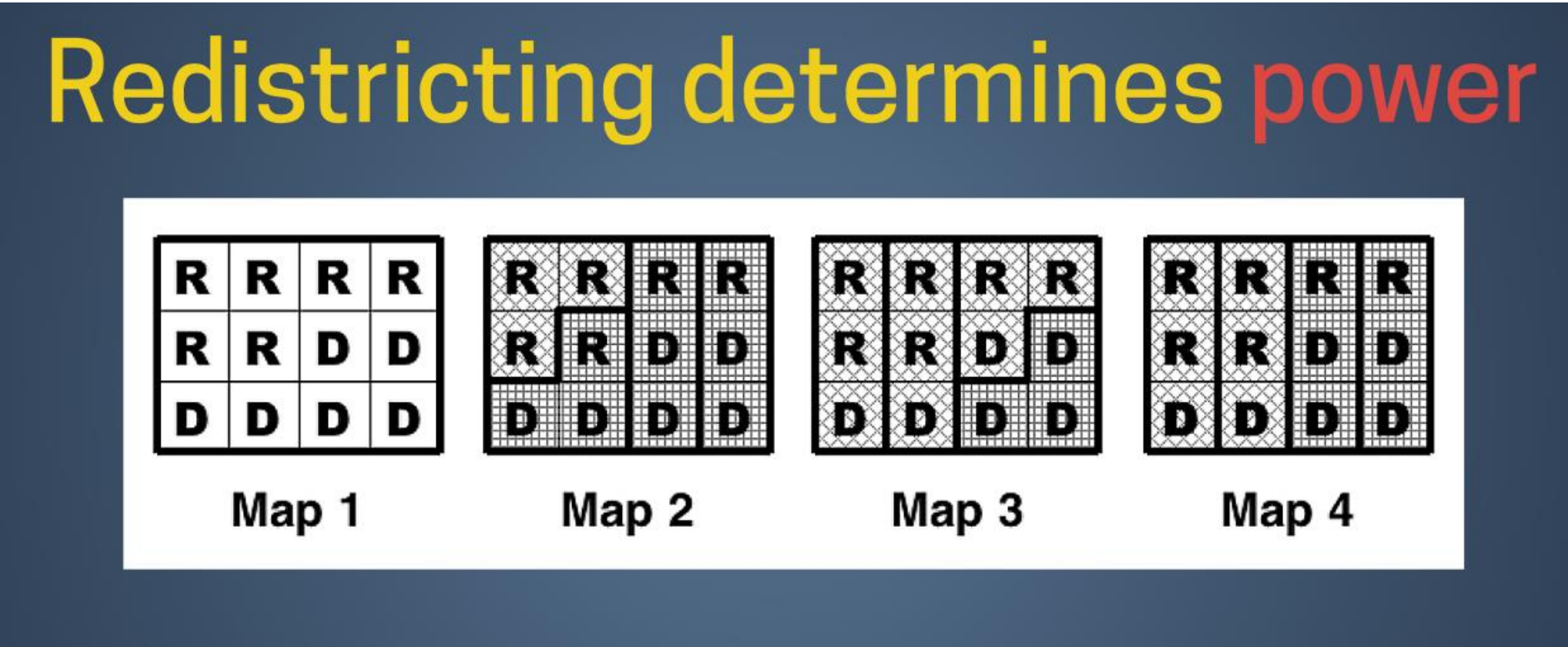


(Optimized Pumping, Wang 2013)
Water Resource Management

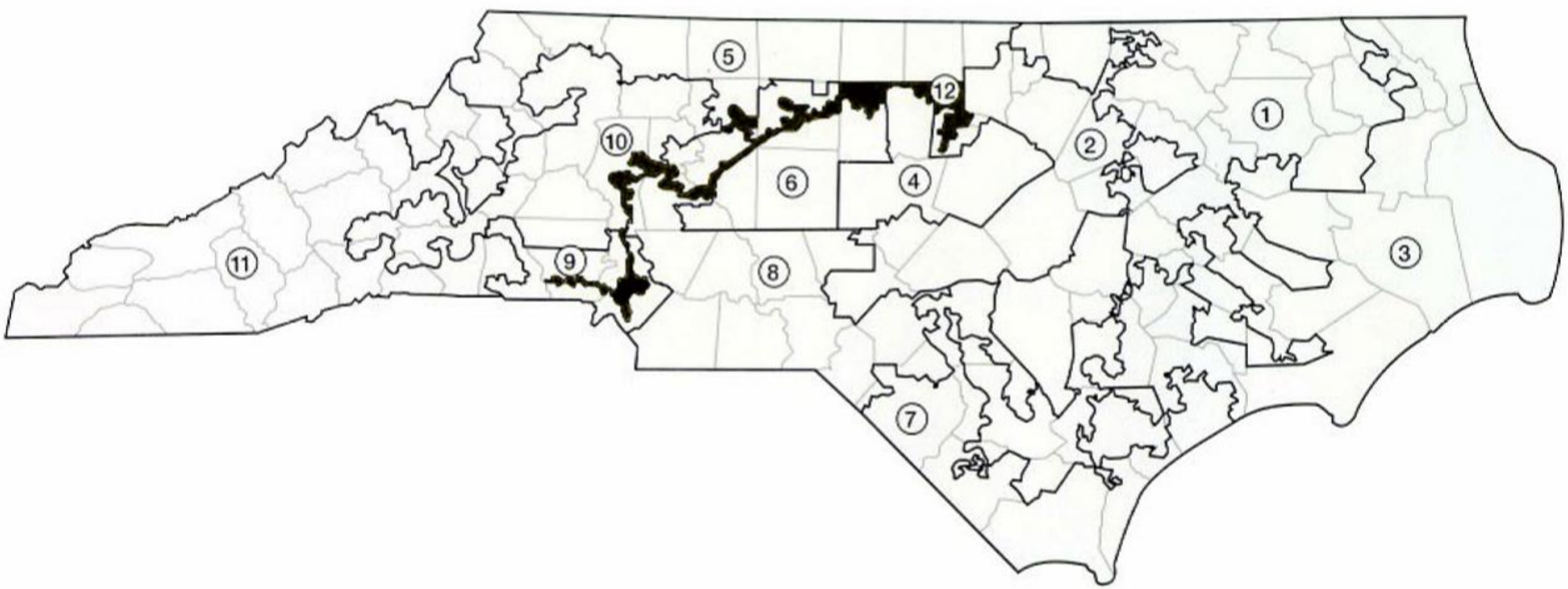
([Truck routing, 2016 Informs Wagner Prize]())
Transportation

Political Redistricting

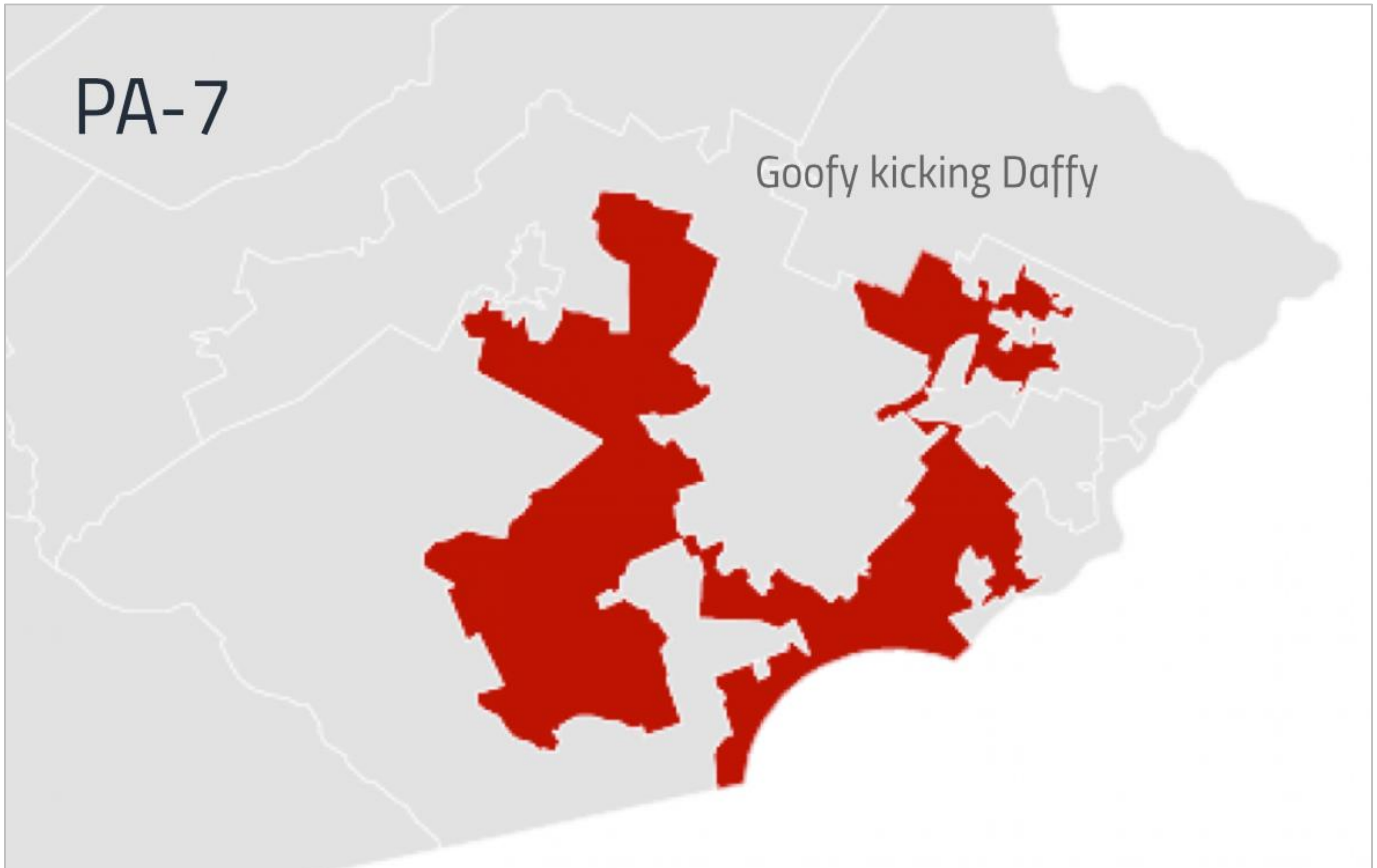
- Partitioning a set of discrete political units into a few districts for the election of legislative seats



NC-12

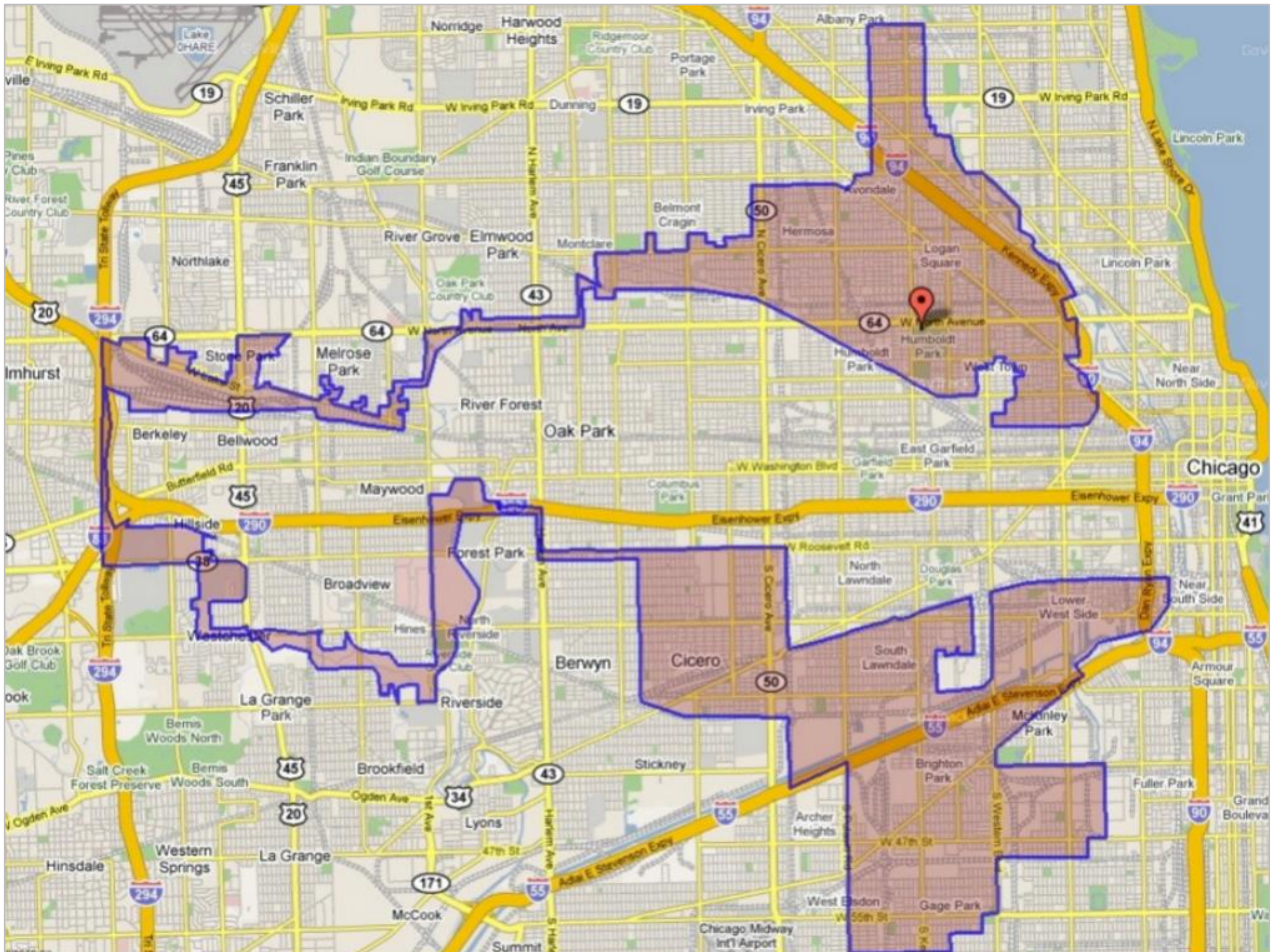


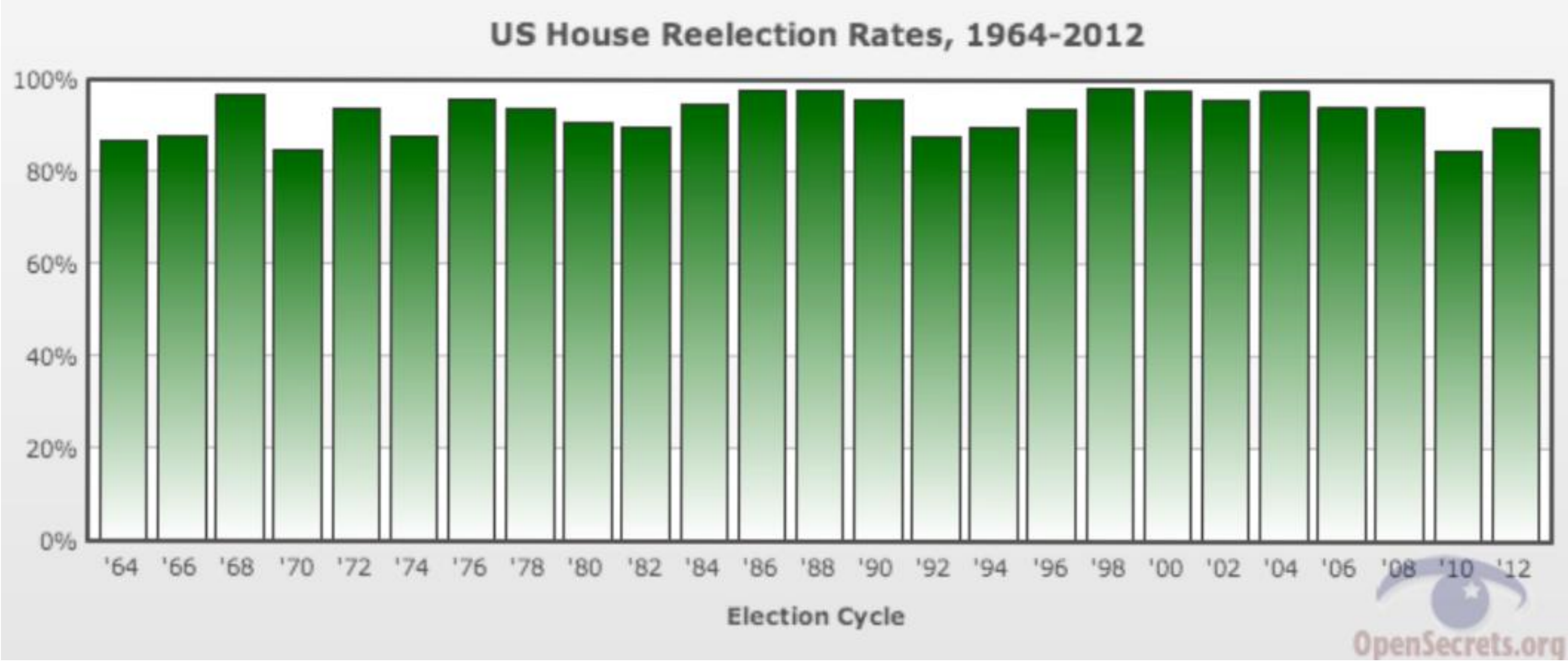
PA-7

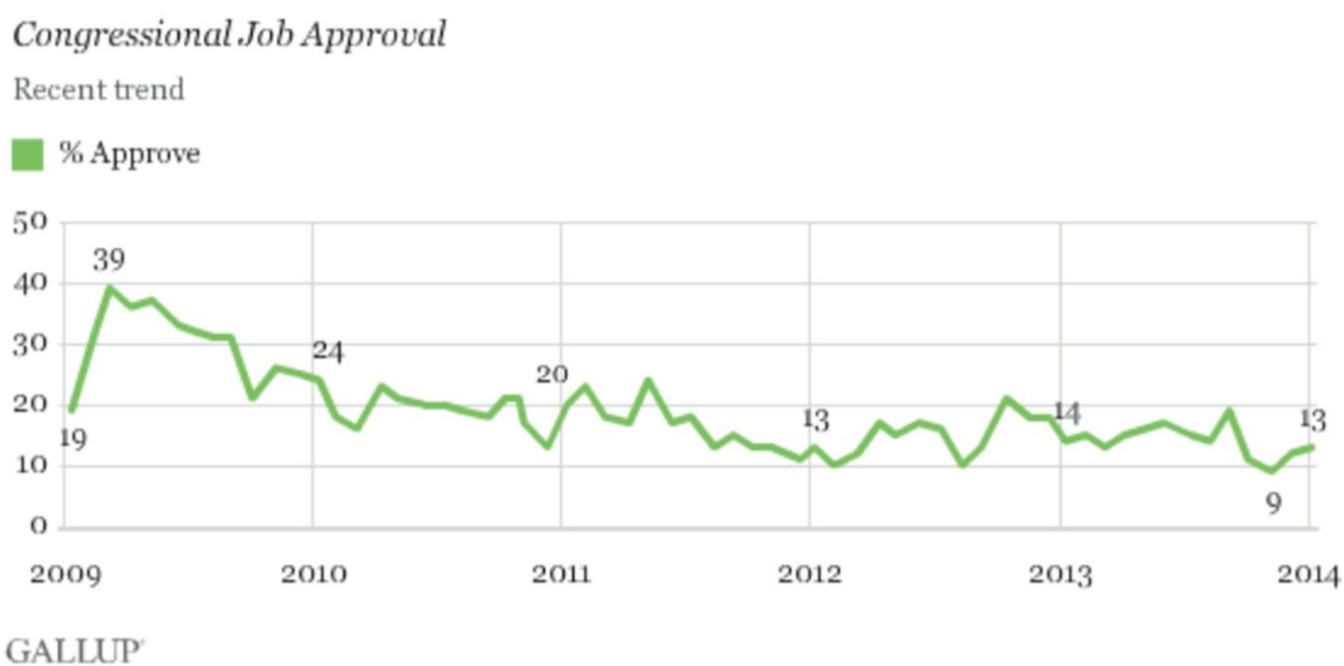


TX-29 (1992)

IL-4



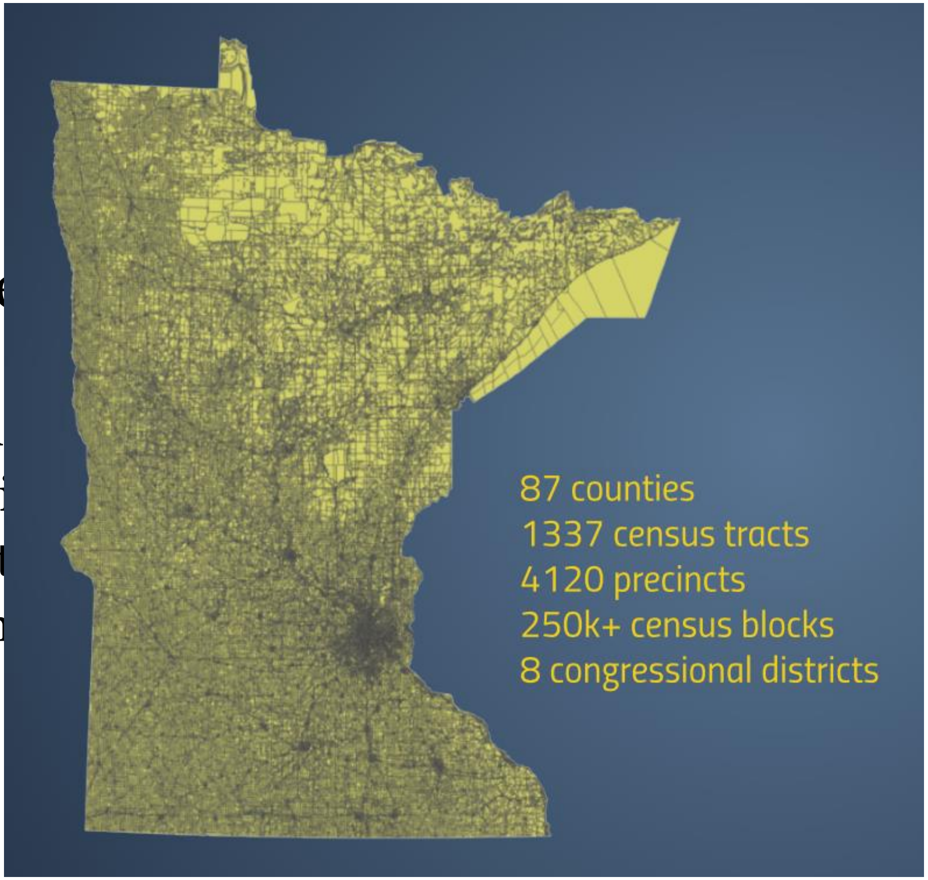

US House Reelection Rates, 1964-2012
100%
80%
60%
40%
20%
0%
'64 '66 '68 '70 '72 '74 '76 '78 '80 '82 '84 '86 '88 '90 '92 '94 '96 '98 '00 '02 '04 '06 '08 '10 '12
Election Cycle
OpenSecrets.org


Congressional Job Approval
Recent trend
% Approve
50
40
30
20
10
0
39
19
24
20
13
14
13
9
2009 2010 2011 2012 2013 2014
GALLUP

Evolutionary Computation for Spatial Optimization

Redistricting: Problem Formulation

- Variables: political units
  - E.g., precincts, voting districts,
- Legal requirements/prefere
  objectives
  - Contiguity/adjacency: inter- an
  - Containment: hole-free districti
  - Shape: compactness, city/count
  - Political interests: competitiven
    districts, incumbency

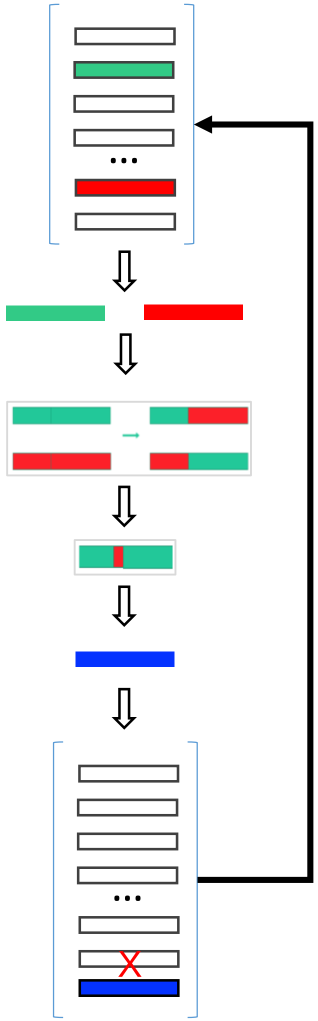


- Computing complexity
  - *NP*-hard.
  - Decision space size: at the same magnitude as the Stirling number of the second kind

$$\left\{ {n \atop k} \right\} = \frac{1}{k!} \sum_{j=0}^{k} (-1)^{k-j} \binom{k}{j} j^n$$

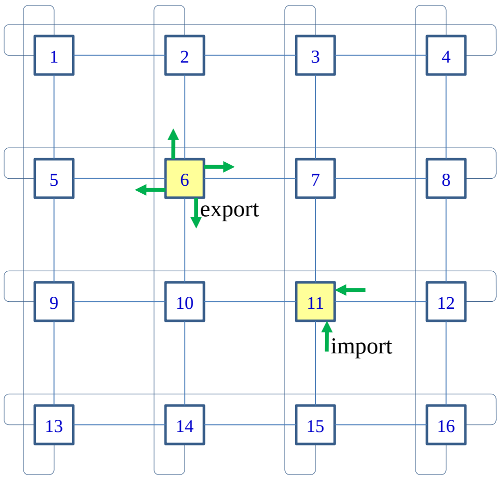
Classic Evolutionary Algorithm (EA)

Population initialization;
`do`
    Select two solutions;
    Crossover;
    Mutation;
    Fitness evaluation;
    Replacement;
`while` stopping criteria are not met;
Output the best solution found;



Scalable Evolutionary Computation

PEA Parallelisms



Population initialization;
do
    Select two solutions
    Crossover;
    Mutation;
    Fitness evaluation;
    Replacement;
while stopping criteria are not met;
Output the best solution found;

Fine-grained PEA

Global parallelization

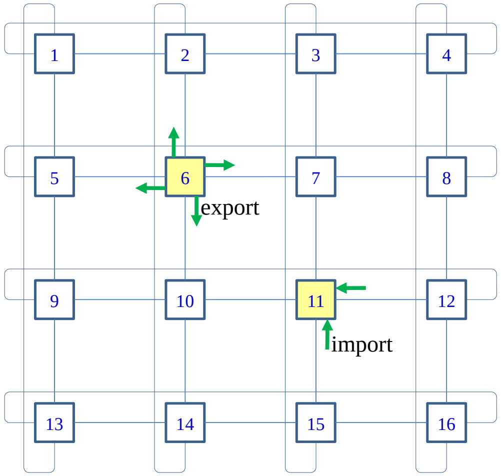
Coarse-grained PEA

“With Extreme Computing, the Rules Have Changed”

- Communication is a bottleneck for scaling
  - Communication avoidance
  - Non-blocking and asynchronous communication
- Randomized algorithms have great potential at extreme scale

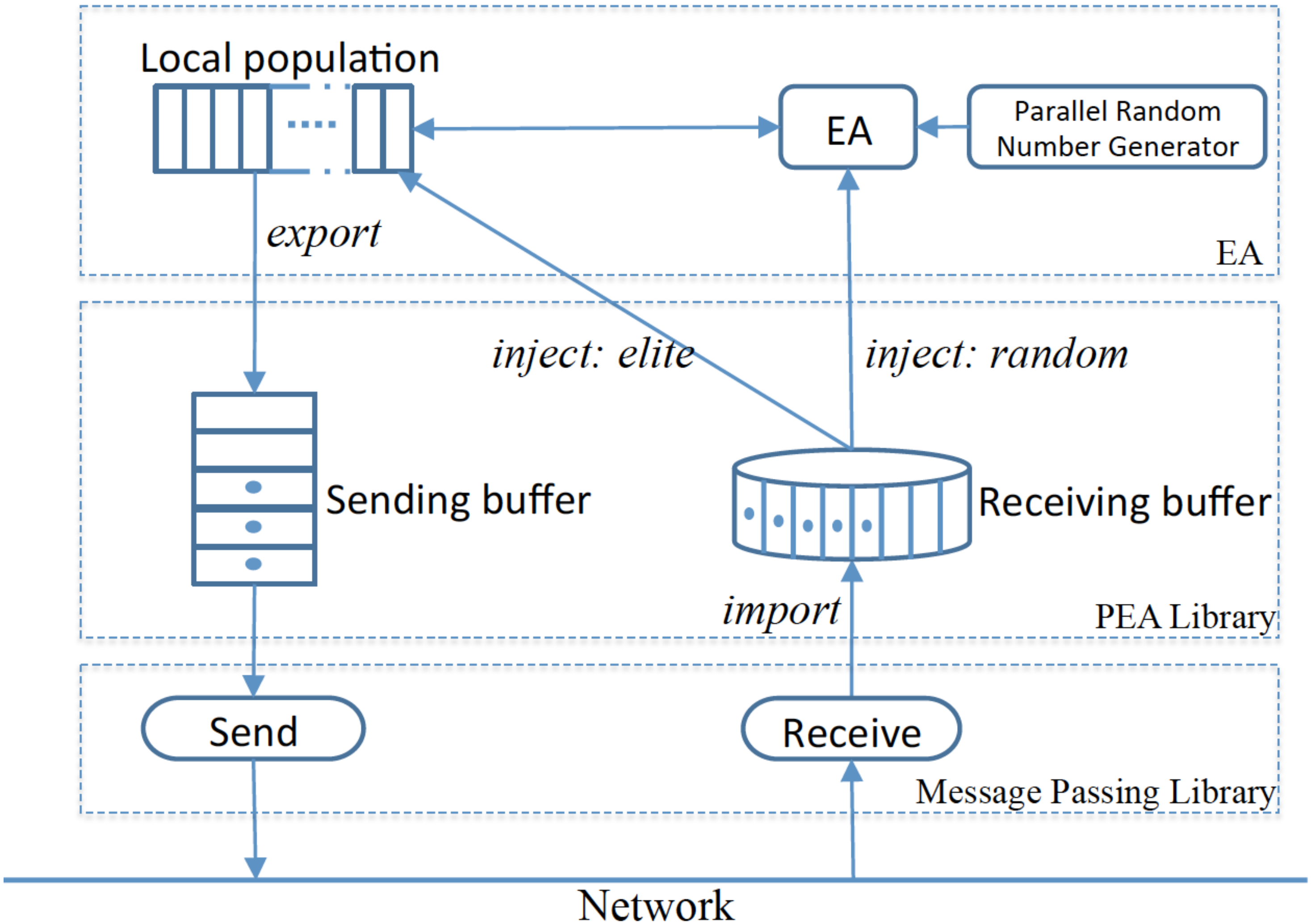
With Extreme Computing, the Rules Have Changed, Jack Dongarra, Stanimire Tomov, Piotr Luszczek, Jakub Kurzak, Mark Gates, Ichitaro Yamazaki, Hartwig Anzt, Azzam Haidar, and Ahmad Abdelfattah, IEEE CISE, April 2017, DOI:10.1109/MCSE.2017.48

Research Challenges for Coarse-grained PEA



- Scalability issue of existing PEAs
  - Solution migration
    - Migration interval, rate
    - Implementation: global synchronization
  - Communication cost
    - Performance drops more than 50% on supercomputers when more than 1,024 cores were used
- Synchronous *vs*. Asynchronous migration
  - Benefit of asynchronous migration
    - Increase the overlapping of computing and communication
  - Cost of breaking the global barrier
    - Asynchronous migration
    - Application-level message handling

PEAR: A Scalable PEA for Redistricting



Buffer Overflow Analysis

- Goals
  - Theoretical analysis on buffer overflow issues
  - Avoid buffer overflow caused by improper PEA parameter configuration

M_{impt}: import interval

M_{expt}: export interval

r : migration rate

d : degree of connectivity

$K_{sendbuf}$: size of sending buffer

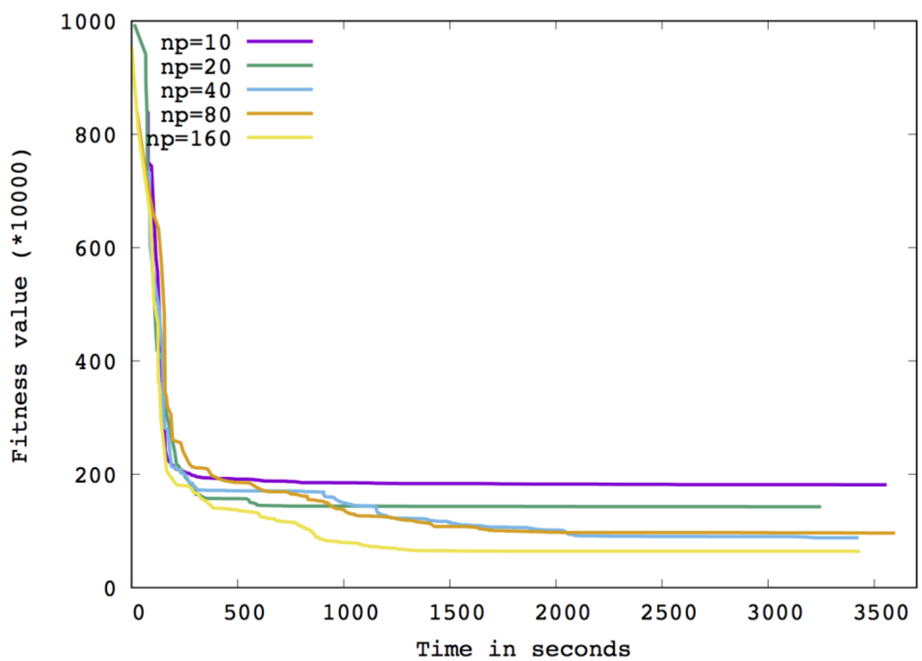
K_{impt}: size of receiving buffer

(1): $M_{impt} / M_{expt} <= K_{sendbuf} / r$

(2): ($d * r / M_{expt} <= 1$) and ($K_{impt} >= K_{sendbuf} * d$)
→ no overflow

PRCRX+ECMUT : Weak Scaling

Number of processors	Fitness thresholds							
	1000	750	500	250	200	150	100	80
10	77.05	93.38	158.19	158.19	278.02			
20	13.26	78.78	119.90	204.28	251.64	563.06		
40	76.46	76.46	125.61	188.10	253.12	995.91	2043.65	
80	25.99	67.45	154.85	242.60	377.14	921.40	1769.73	
160	0.66	88.95	124.24	165.59	200.18	368.39	846.78	998.98



Implementation

- Asynchronous migration strategy
  - Sending
    - MPI_Ibsend() and MPI_Waitall(); or
    - *(new)* MPI_Isend() and MPI_Testall()
  - Receiving
    - MPI_Iprobe() and MPI_Recv()
- Deployment on XSEDE
  - Stampede
  - Lonestar
  - Trestles
  - Gordon
  - Ranger

Table 1: PGAP default configuration

Parameters	Settings
Population size per deme	100
Initial population generation	Random with feasibility improvement or constraint-based improvement
Selection	Binary tournament
Crossover	1-point. Probability: 0.8
Mutation	1-item mutation. Probability: 0.2
Replacement	Replacing the unfittest or worst
Elitism	Yes
Stopping rules	No solution improvement, bounded solution quality reached, or fixed number of iterations
Connectivity d	4
Migration rate r	2
Export interval M_{expt}	100
Import interval M_{impt}	50
Probability of holding	1/20 (the probability to export when no better solution found during a previous export interval)
Sending buffer size $K_{sendbuf}$	20 solutions. Actual memory requirement is $(20 \times n \times 4 + buffer\_overhead)$ bytes
Import pool size K_{impt}	80 solutions. Actual memory requirement is $(80 \times n \times 4)$

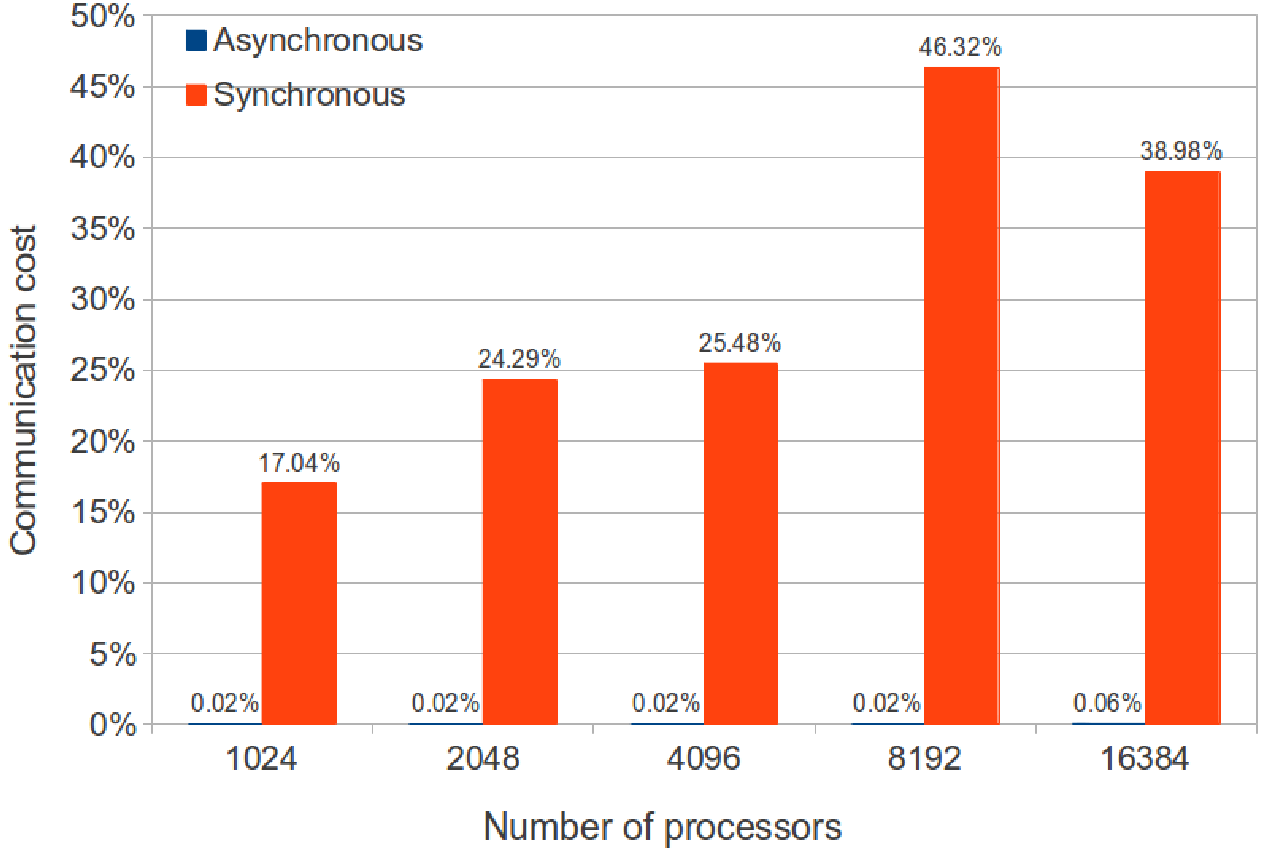
Computational Performance

- Performance evaluation on Stampede
- Scalability
  - Strong scaling
  - Weak scaling
- Communication cost
- Solution quality improvement

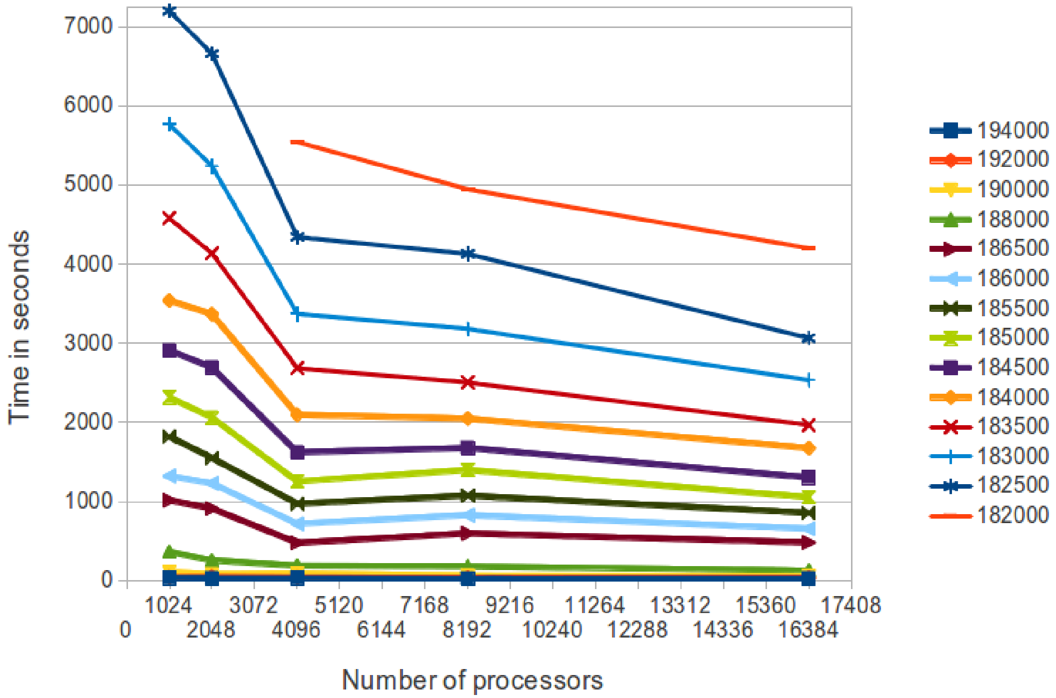
Scalability: Weak Scaling

- Weak scaling evaluation in PGA
  - Computational intensity is problem-specific
    - Small problems may be harder to solve
  - Our approach
    - Local population size is kept same
    - By increasing the number of processors, the total amount of computational work increases as well (referred to as “new era” weak scaling by Sarkar *et.al*)
- Weak scaling experiment
  - Local population size: 100
  - Base case: 512 processor cores
  - Execution time is measured at different fitness thresholds
  - Number of processors: 512, 1024, 2048, 4096, 8192, 16384

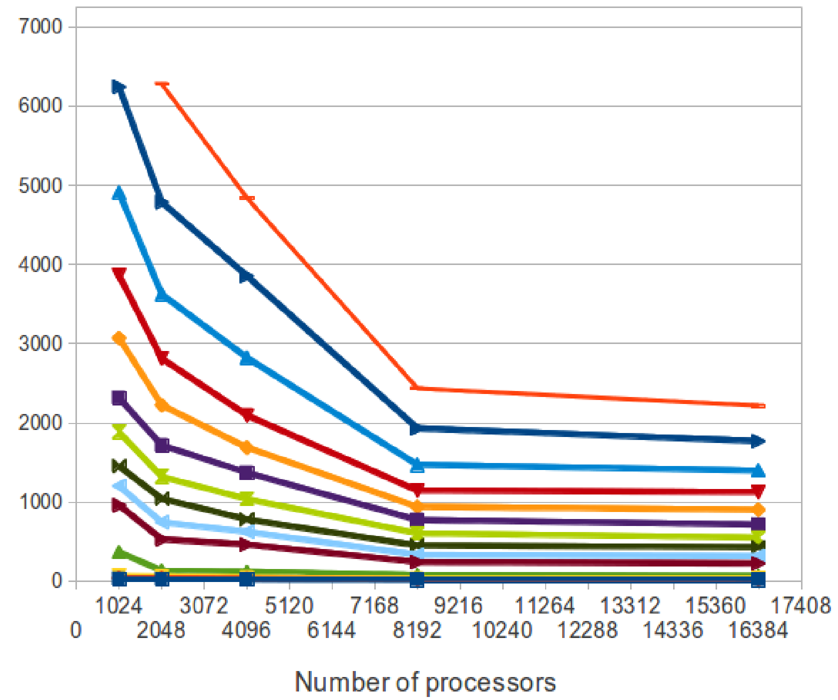
Communication Cost



Weak Scaling Results



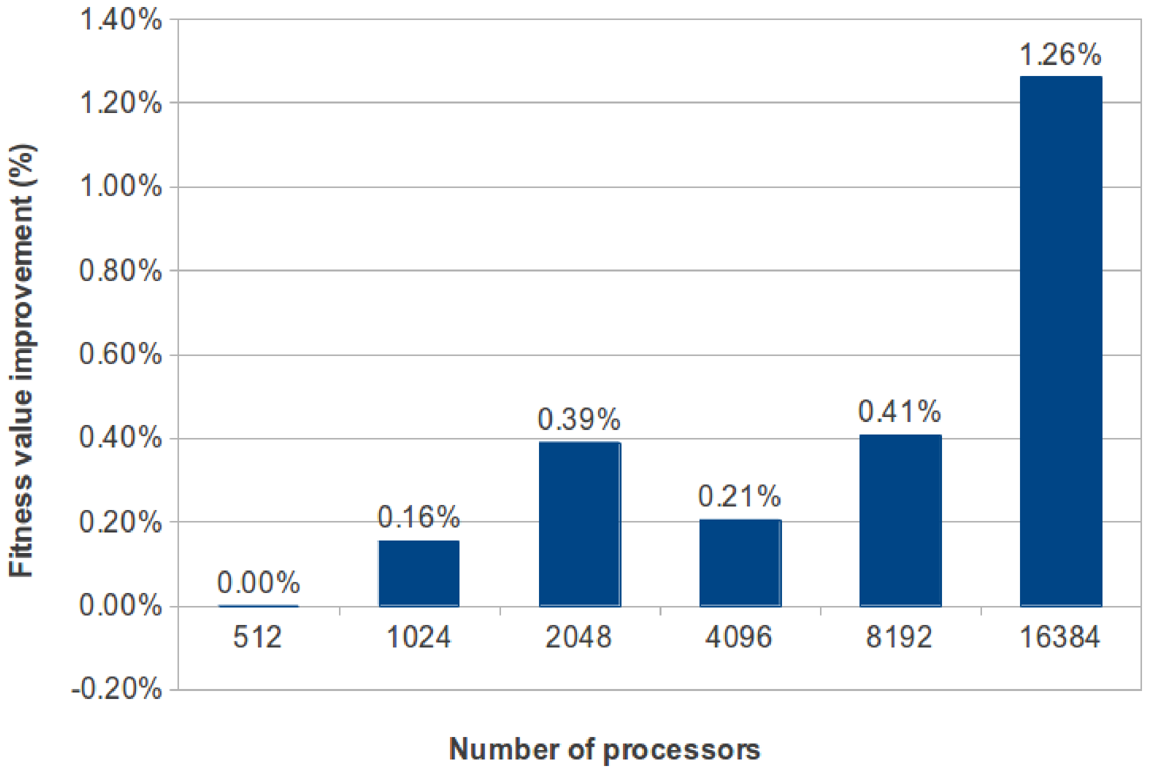
Synchronous



Asynchronous

- Doubling the number of processors significantly reduces the time to achieve a solution quality threshold
- Using larger number of processors is likely to achieve tighter solution quality thresholds
- Asynchronous migration strategy outperformed synchronous migration dramatically

Solution Quality Improvement



- Solution quality improvement is measured as the improvement of fitness value over synchronous migration
- 1.26% improvement using 16,384 processors is significant as better solution is harder to find

Symmetric Computing

- Hybrid architecture on Stampede
  - CPU: E5 Sandy Bridge processor (16 cores/node)
  - MIC: Intel Xeon Phi co-processor (61 cores/card)
- Symmetric *vs*. offloading
- Exploration of symmetric computing in PGAP
  - Impact on GA computing
    - Some fast MPI processes on CPU *vs*. a large number of slow processes on MIC
    - Will solutions migrated from fast processes overwhelm local populations on MIC processes?
    - Will asynchronous migration code break given the high heterogeneity of computing and networking between CPU processes and MIC processes?
  - Impact on communication and the effect on numerical performance of PGAP
    - Inter-node and intra-node
    - CPU-MIC communications
  - Impact on the configuration of asynchronous migration strategy
    - Should we set different PGA parameters on CPU processes and MIC processes?

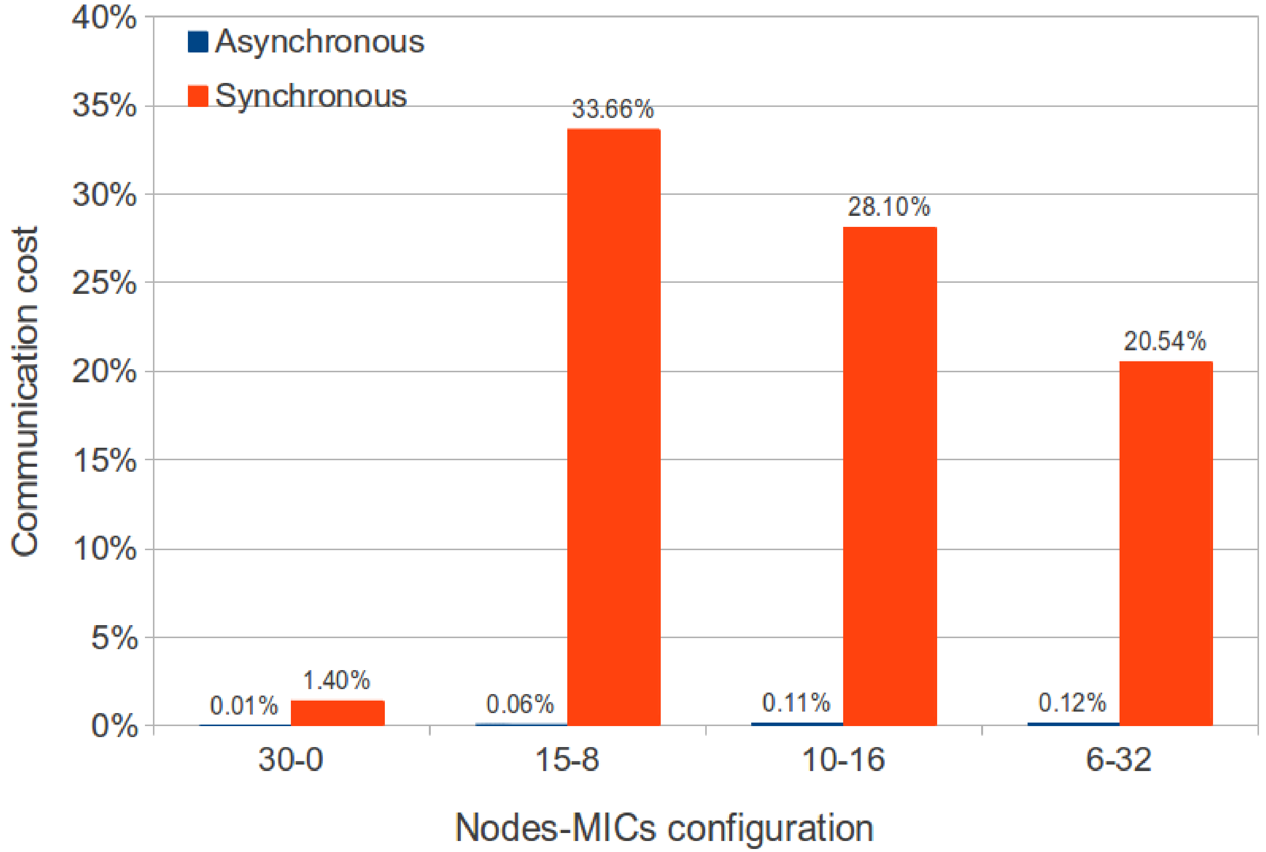
Status

- Implementation
  - Export operator
    - Sending buffer is controlled by PGAP, not MPI
      - Runtime errors were experienced when using user-provided MPI sending buffer
    - Strategy to handle the difference in computing between CPU and MIC processes
      - A PGAP process will skip an export operation if sending buffer is full, instead of waiting for the completion of all pending sending operations
      - MPI_Isend() + MPI_Testall()
- Scalability tests
  - Tests using 256 or less cores were successful without any modification of the code
  - Tests using more than 256 cores failed due to system-level configuration issues
    - Up to 1,024 cores (512 CPU cores, 512 MIC cores) were used
    - Runtime communication issue at system-level
    - TACC staff are working on resolving the issue

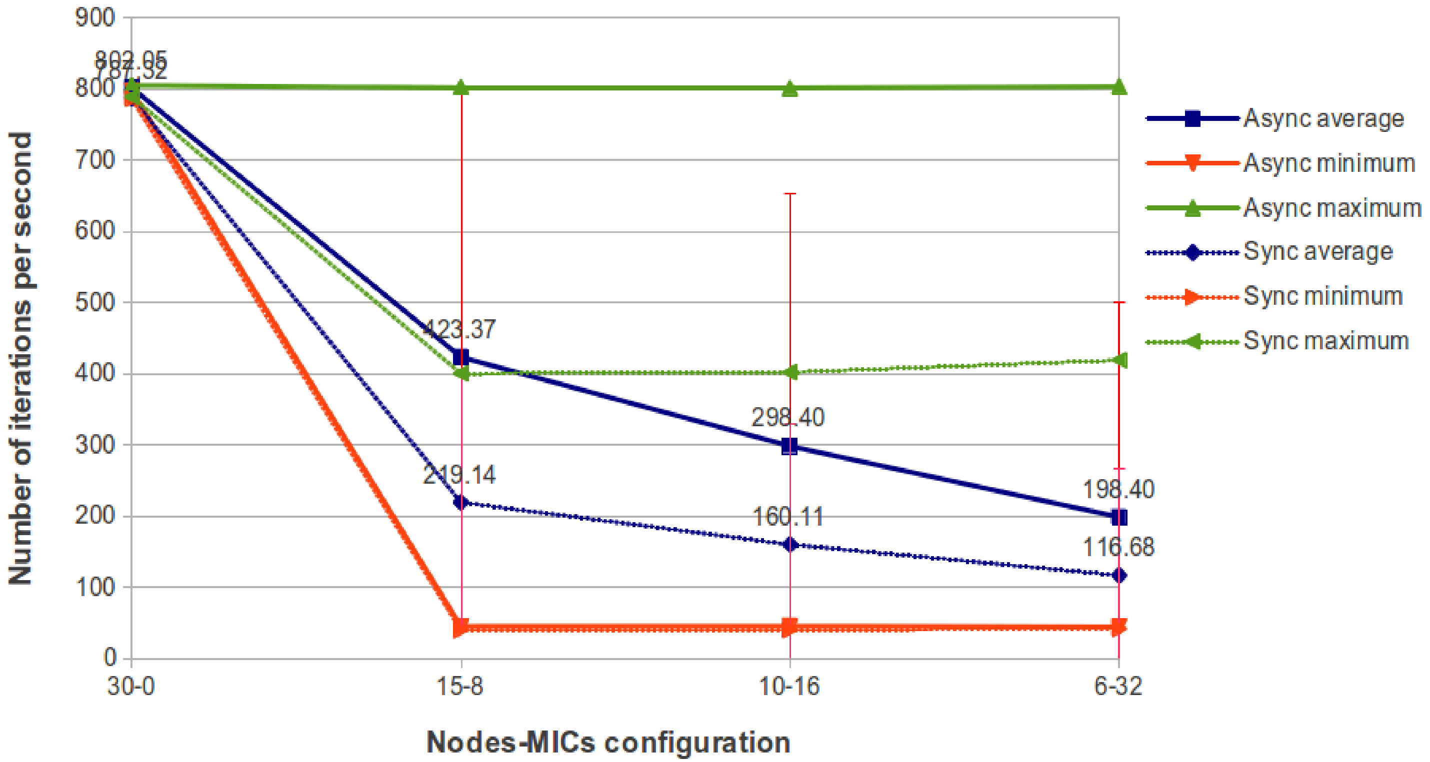
Experiment Design

- CPU-MIC configuration
  - 240 cores in total
  - Four configurations
    - 30 nodes: 8 CPU processes, 0 MIC processes on each node
    - 15 nodes: 8 CPU processes, 8 MIC processes on each node
    - 10 nodes: 8 CPU processes, 16 MIC processes on each node
    - 6 nodes: 8 CPU processes, 32 MIC processes on each node
- Experiments
  - Communication cost
  - Iteration rate: number of iterations per second
    - Measure the difference between CPU processes and MIC processes
  - Weak scaling
  - Numerical performance: number of local solutions generated at each process
    - Measure the capability of generating local solutions given high heterogeneity among neighbors
    - Local solution is counted if it is the current best in the local population

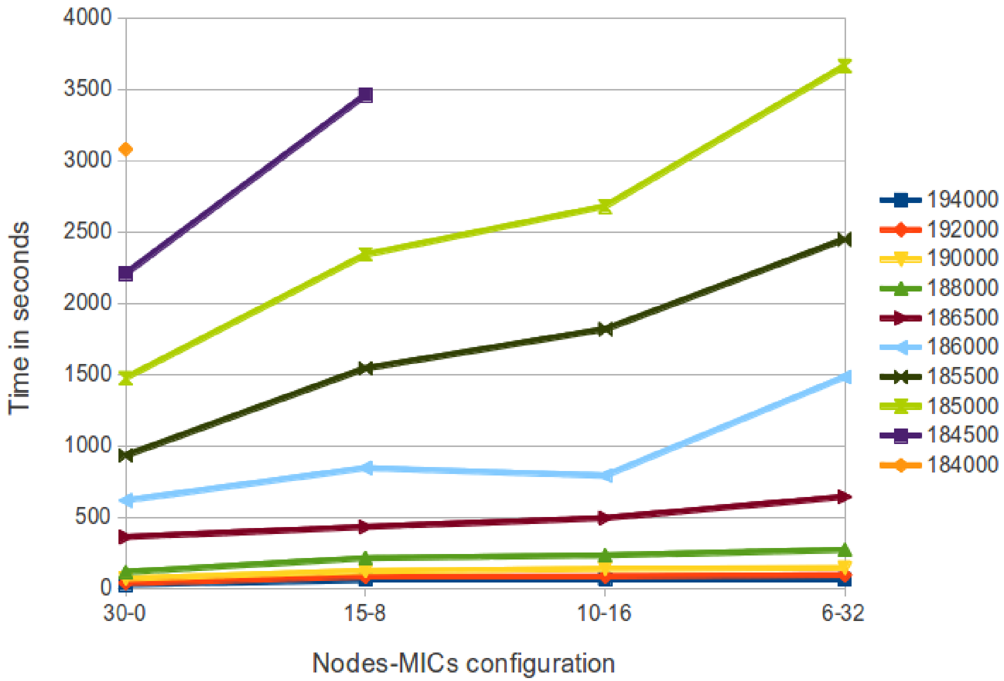
Communication Cost



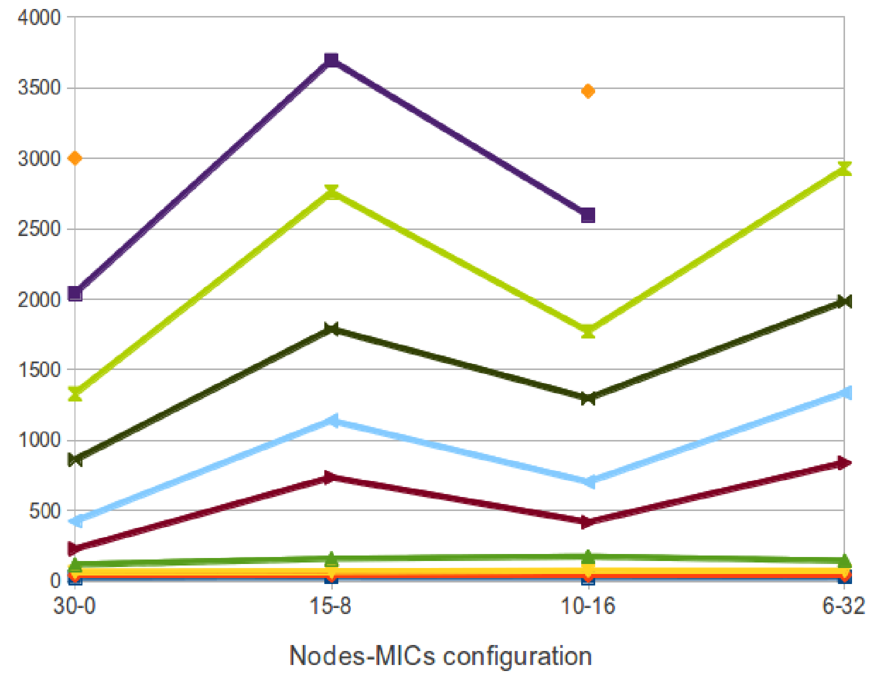
Iteration Rate



Weak Scaling



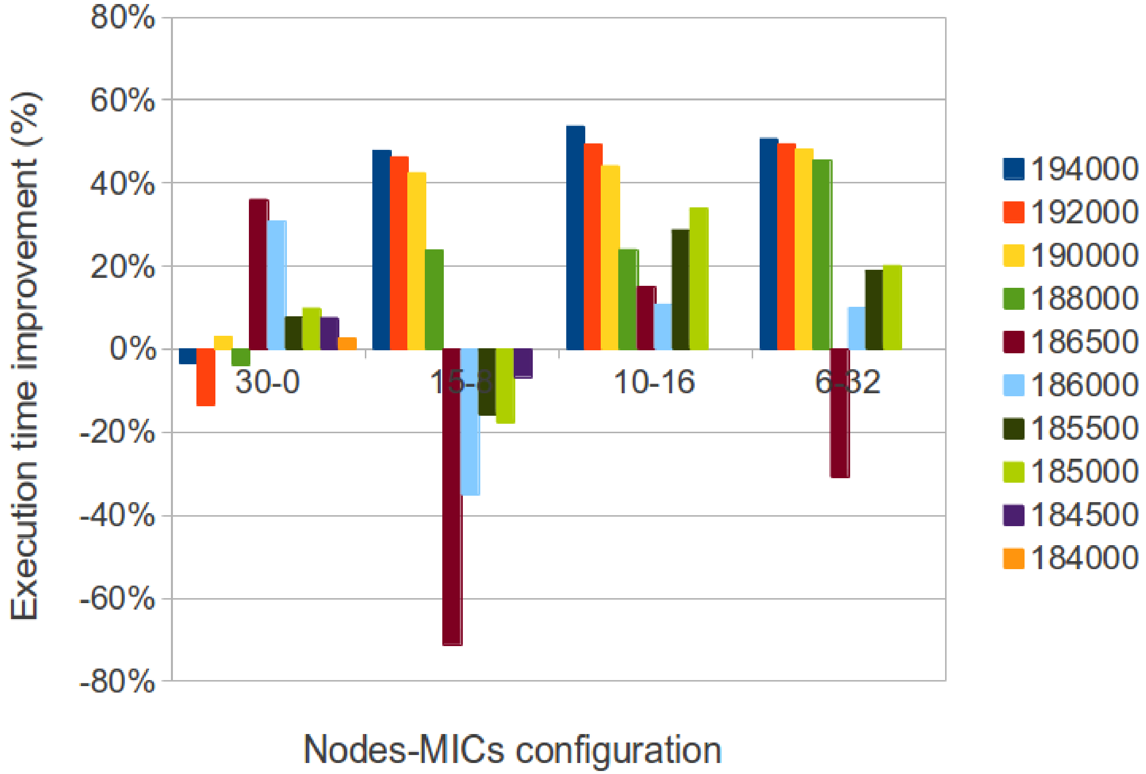
Synchronous



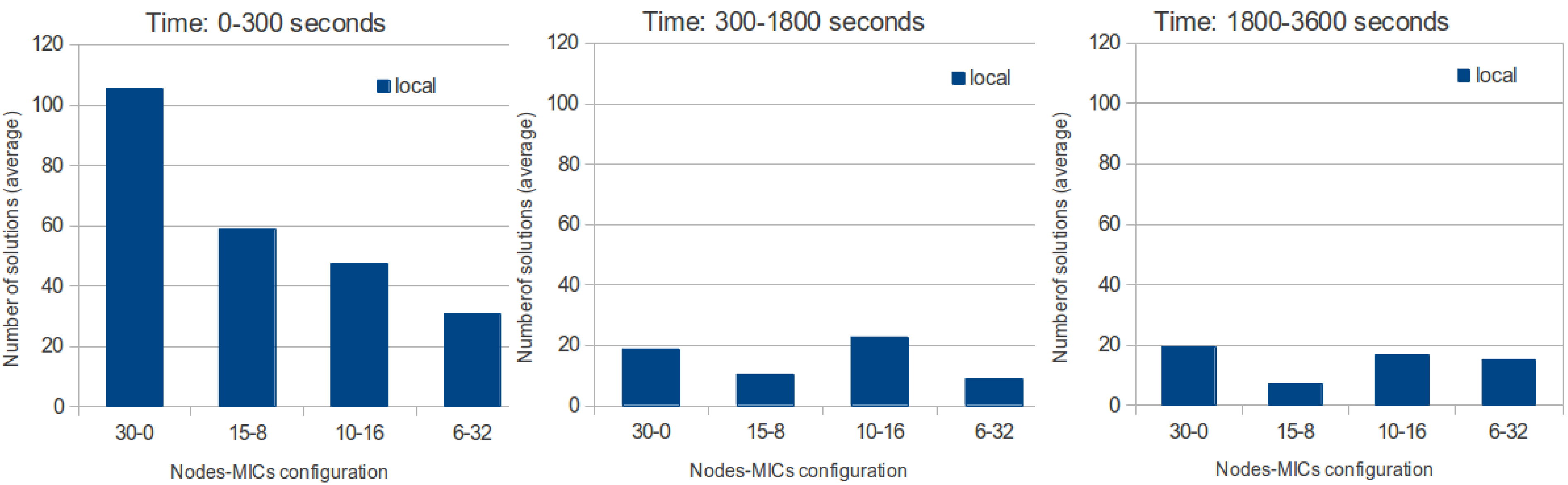
Asynchronous

- Synchronous migration: execution time increases as more MIC processor cores are used
- Asynchronous migration: variation observed
  - Anomaly 1: configuration 15-8 took longer than expected
  - Anomaly 2: configuration 10-16 took less time than expected

Weak Scaling: Execution Time Improvement

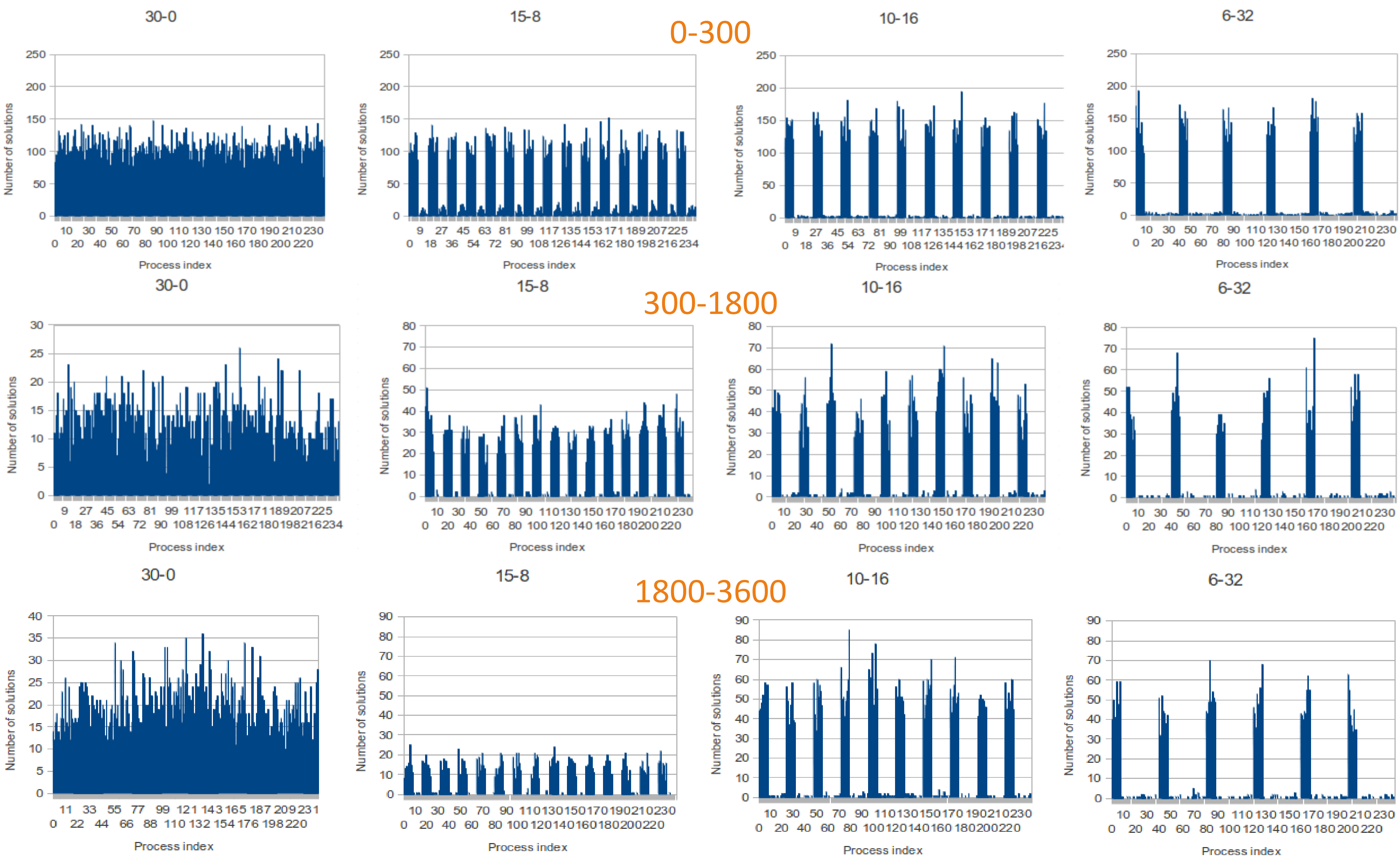


Number of Local Solutions

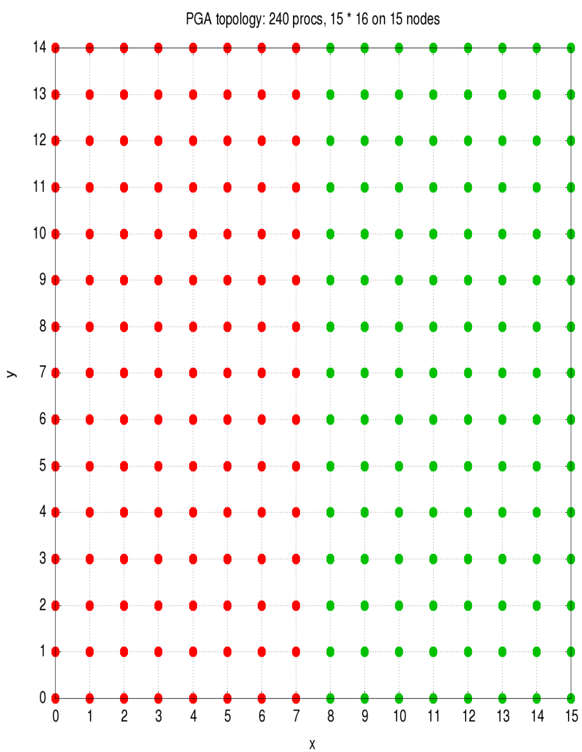


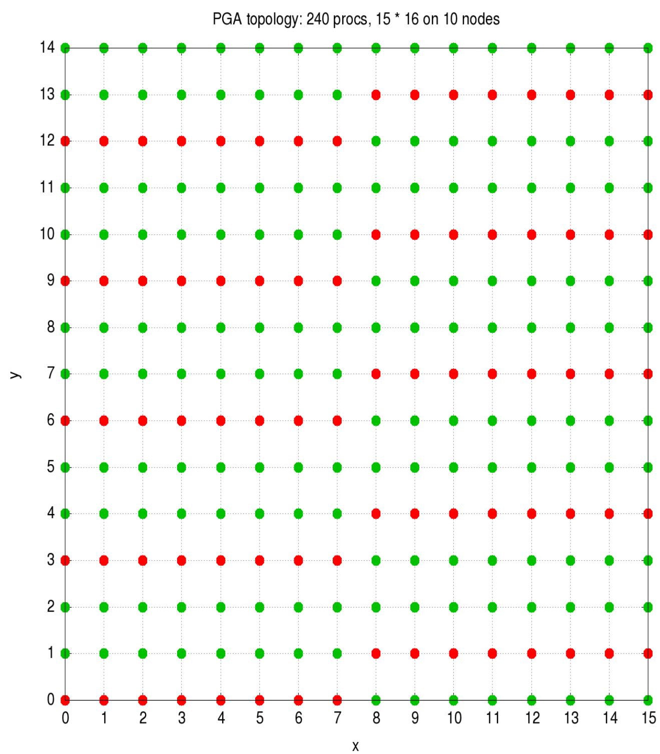
- 0-300: a lot of new solutions were generated
- 300-1800: configuration 10-16 outperformed 15-8 and 6-32
- 1800-3600: similar to 300-1800

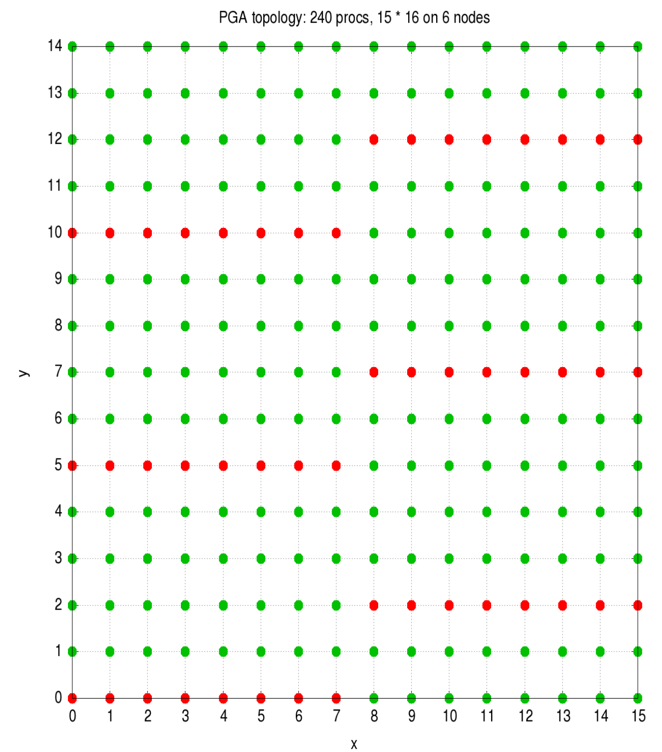
Number of Local Solutions: Snapshots



Topology Analysis





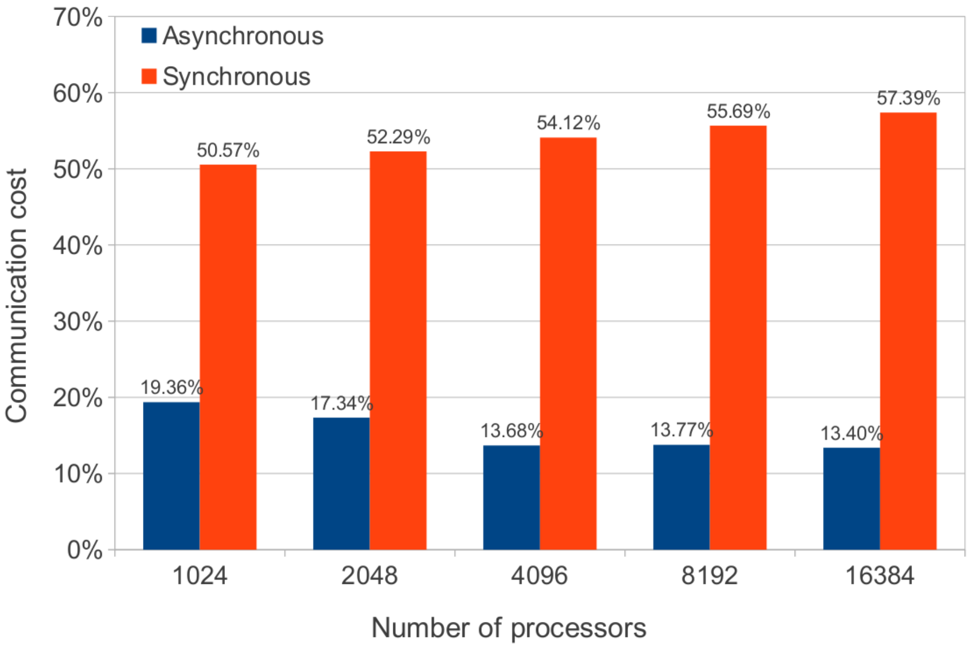


- 15-8: Left half are CPUs; right half are MICs
- 10-16: Every CPU process has at least one MIC process; vice versa
- 6-32: There are MIC processes without CPU process neighbors

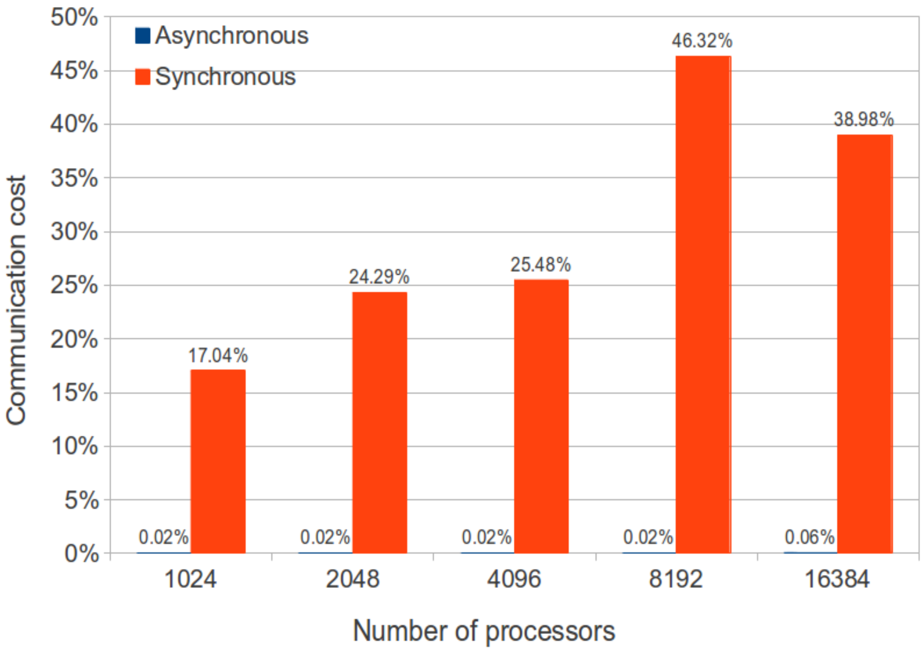
Impact of Topology

- MIC processes generate much fewer solutions than CPU processes
- 15-8:
    - Contributions from MIC processes were suppressed
    - CPU processes may have generated a lot of solutions, but may not be competitive as to be counted as the current best
- 10-16:
    - CPU processes were able to conduct longer local evolutionary computation with fewer external solutions injected
    - More likely to be counted as local best
- 6-32:
    - Similar topology, but the number of CPU processes is less than 10-16
- Hint: Interleaving CPU-MIC processes on topology may perform better than having more CPU processes

Communication Cost Comparison



Ranger



Stampede

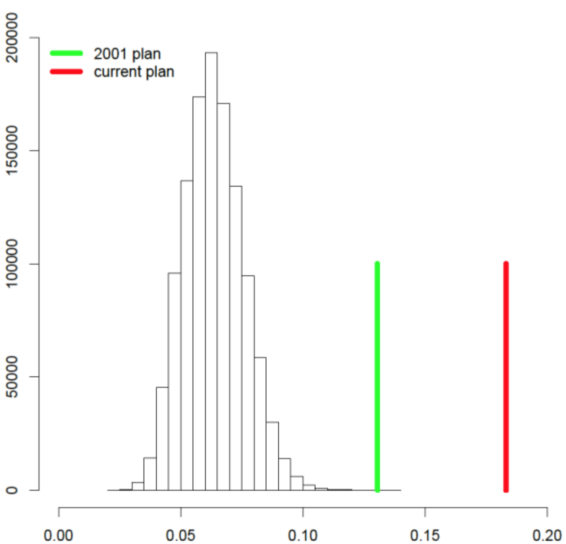
Concluding Discussions

- Parallel genetic algorithm
  - Asynchronous migration strategy can significantly reduce the communication cost among massive computing resources; increase computation/communication ratio; and improve the numerical performance of PGA
  - Desirable scalability was achieved on both Ranger and Stampede by using up to 16384 processor cores
- Symmetric computing on hybrid computing architecture
  - Asynchronous migration strategy is adaptive to the heterogeneity of computing power
  - Topology of PGAP processes has impact on numerical performance when heterogeneous processors are used

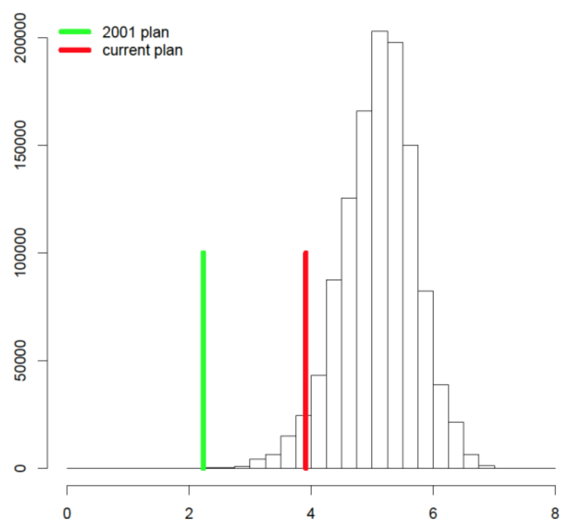
Redistricting Application

- Legal definition of partisan gerrymandering: "the use of partisan information is excessive"
- How to know if a plan uses excessive partisan information?
- Need references and baseline maps

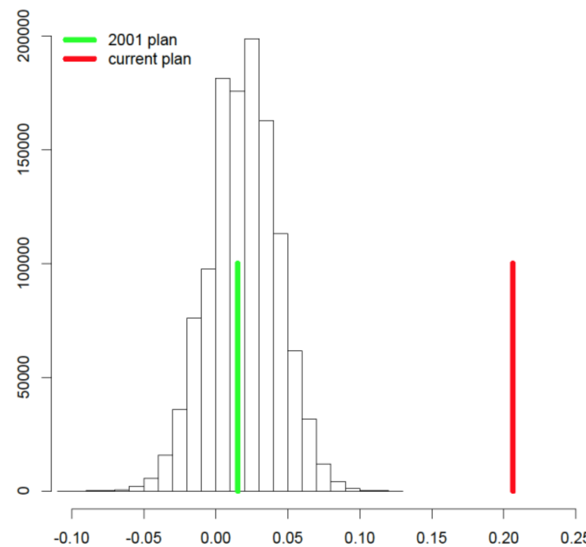
Statistical Analysis on Partisan Gerrymandering: North Carolina (1,174,702 feasible plans)



Competitiveness



Responsiveness



Biasedness