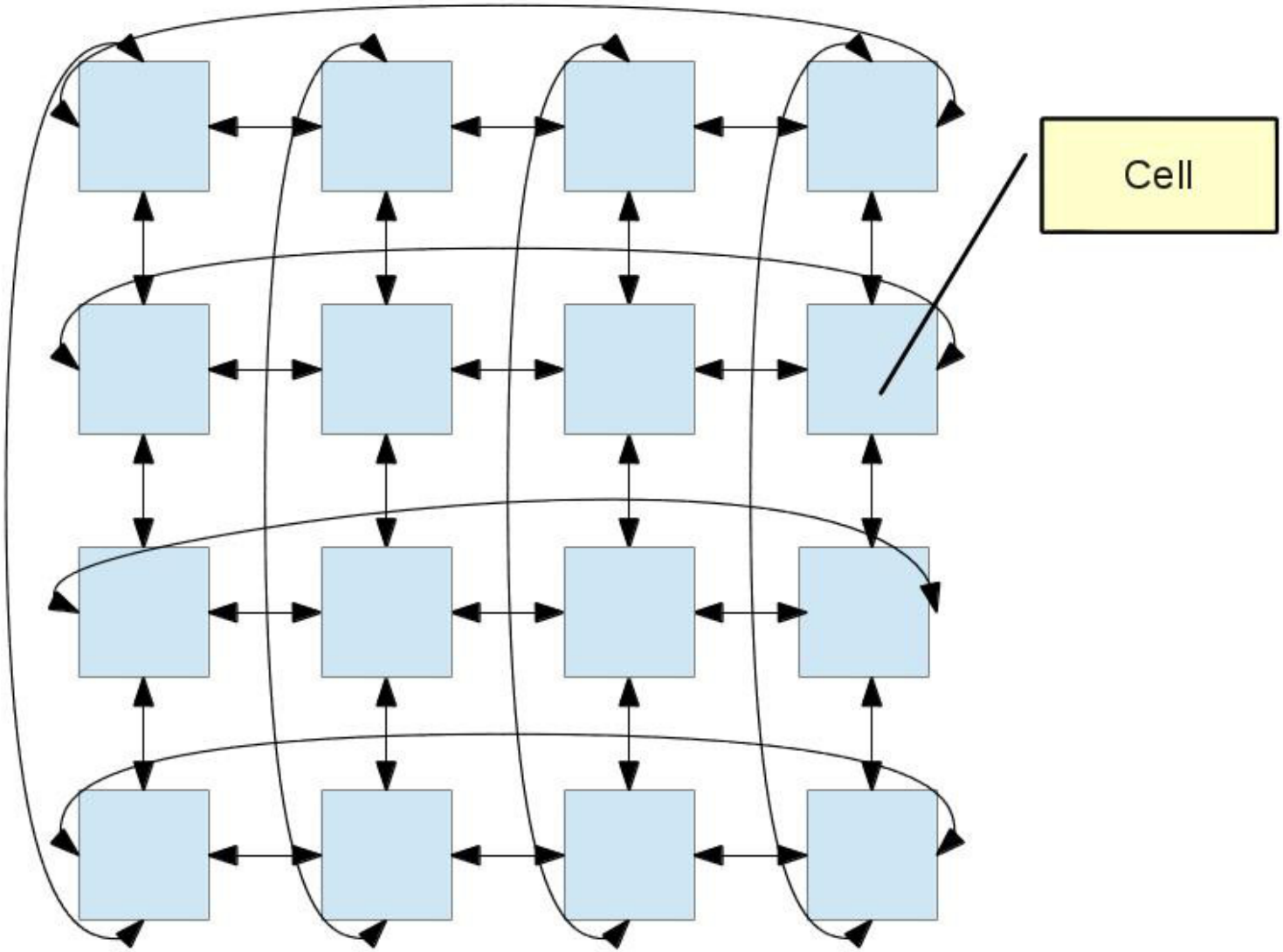
Benchmarking Cellular Genetic Algorithms on the BBOB Noiseless Testbed

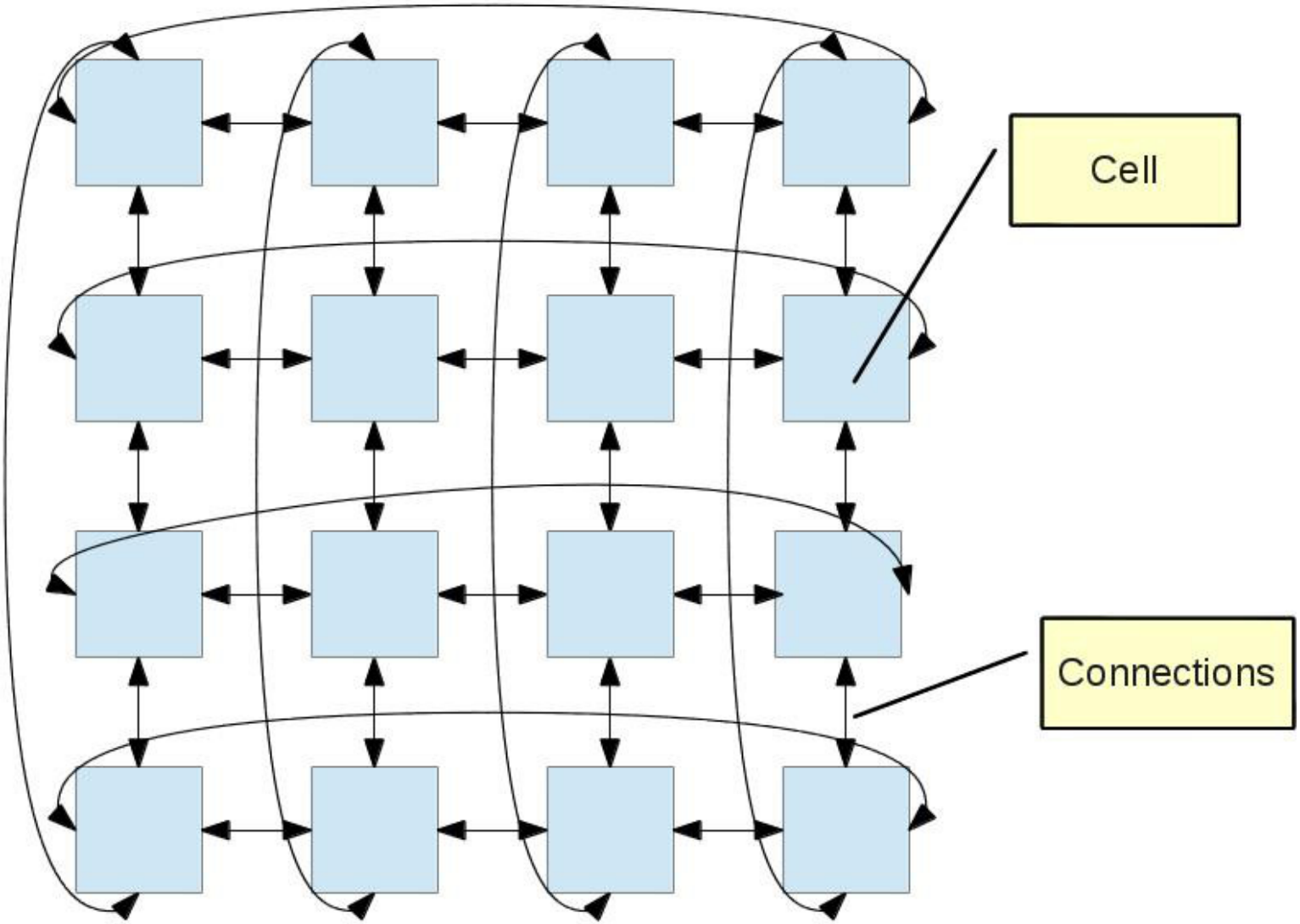
Neal Holtschulte

July 6, 2013

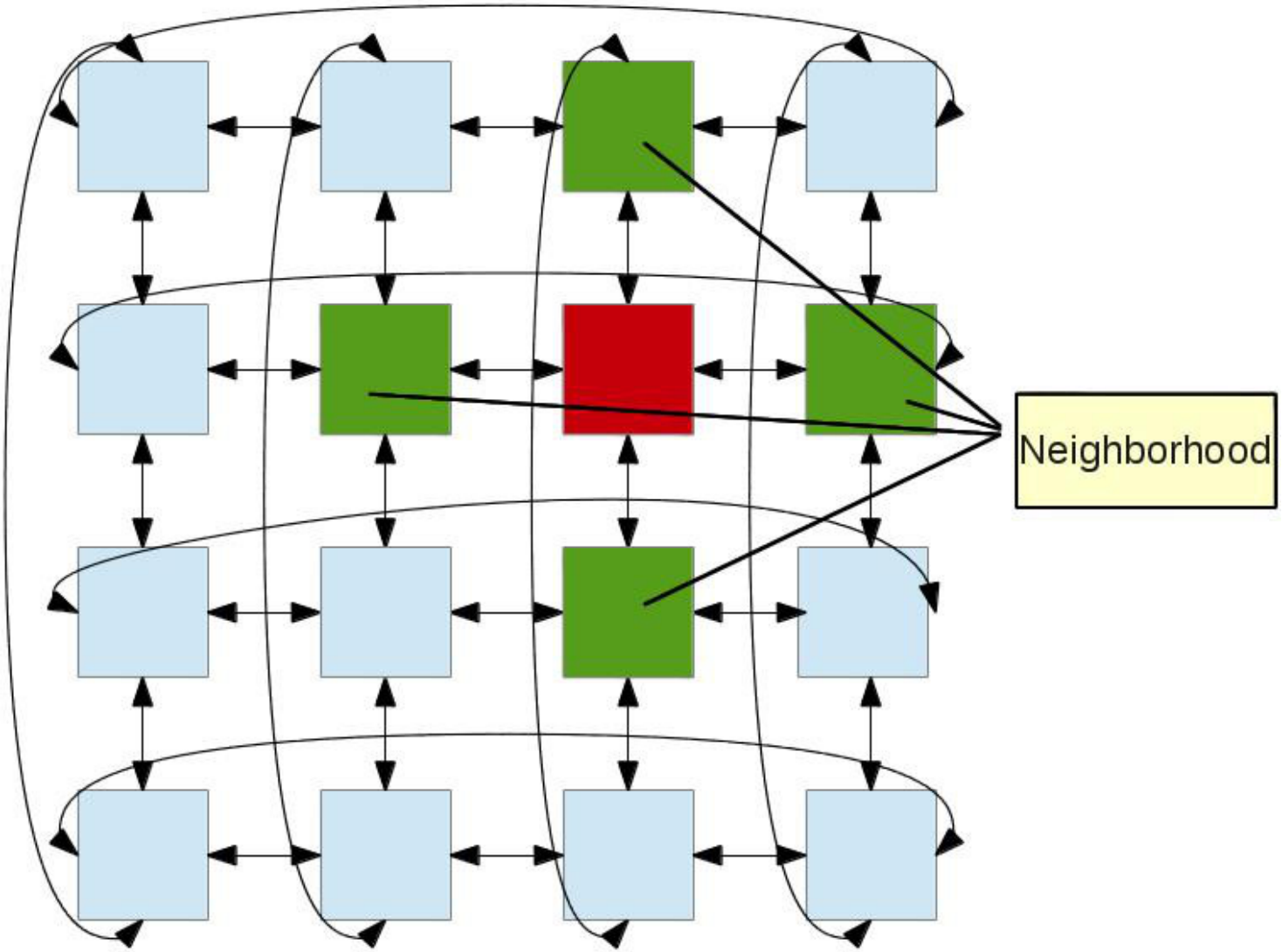
Cellular Genetic Algorithm



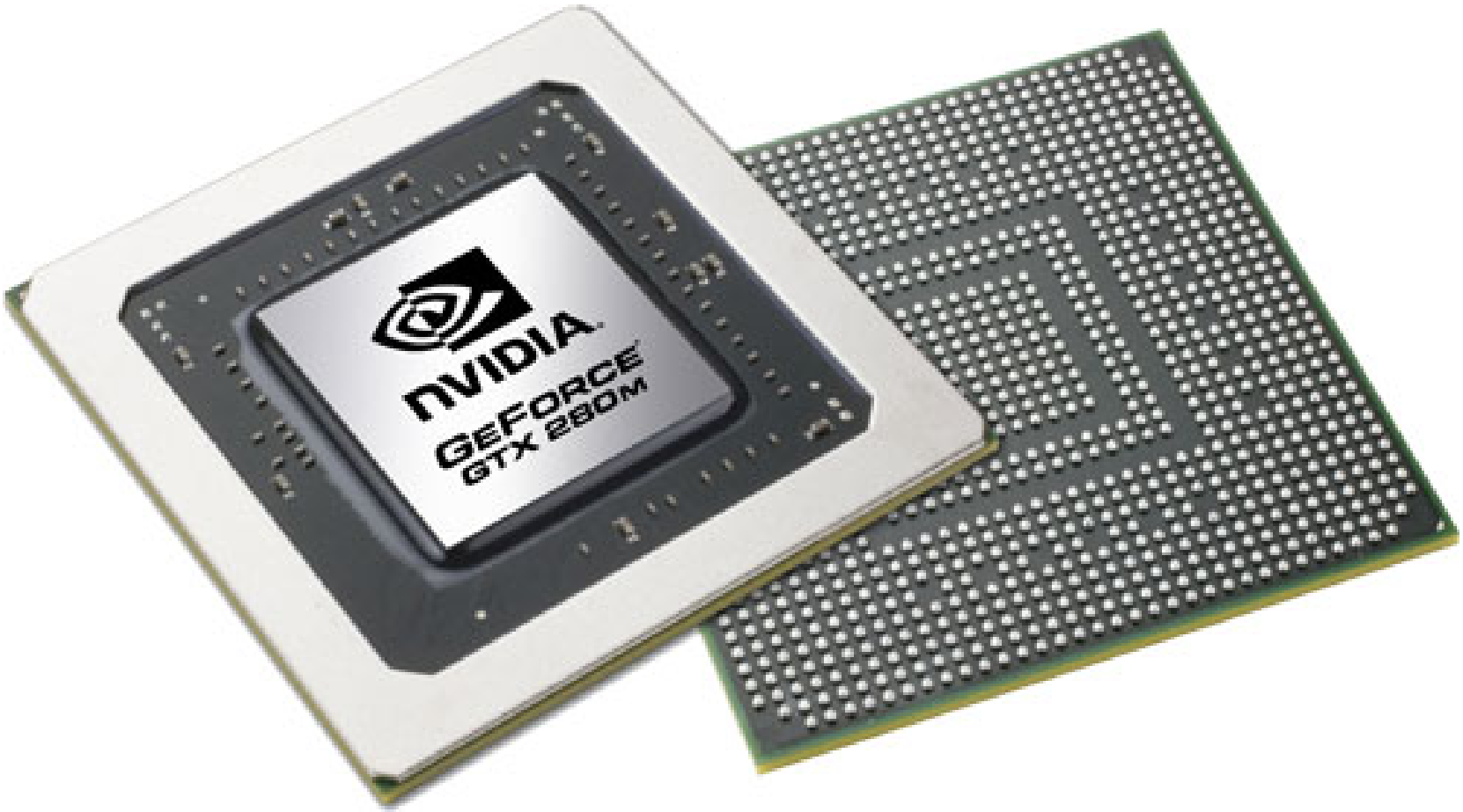
Cellular Genetic Algorithm



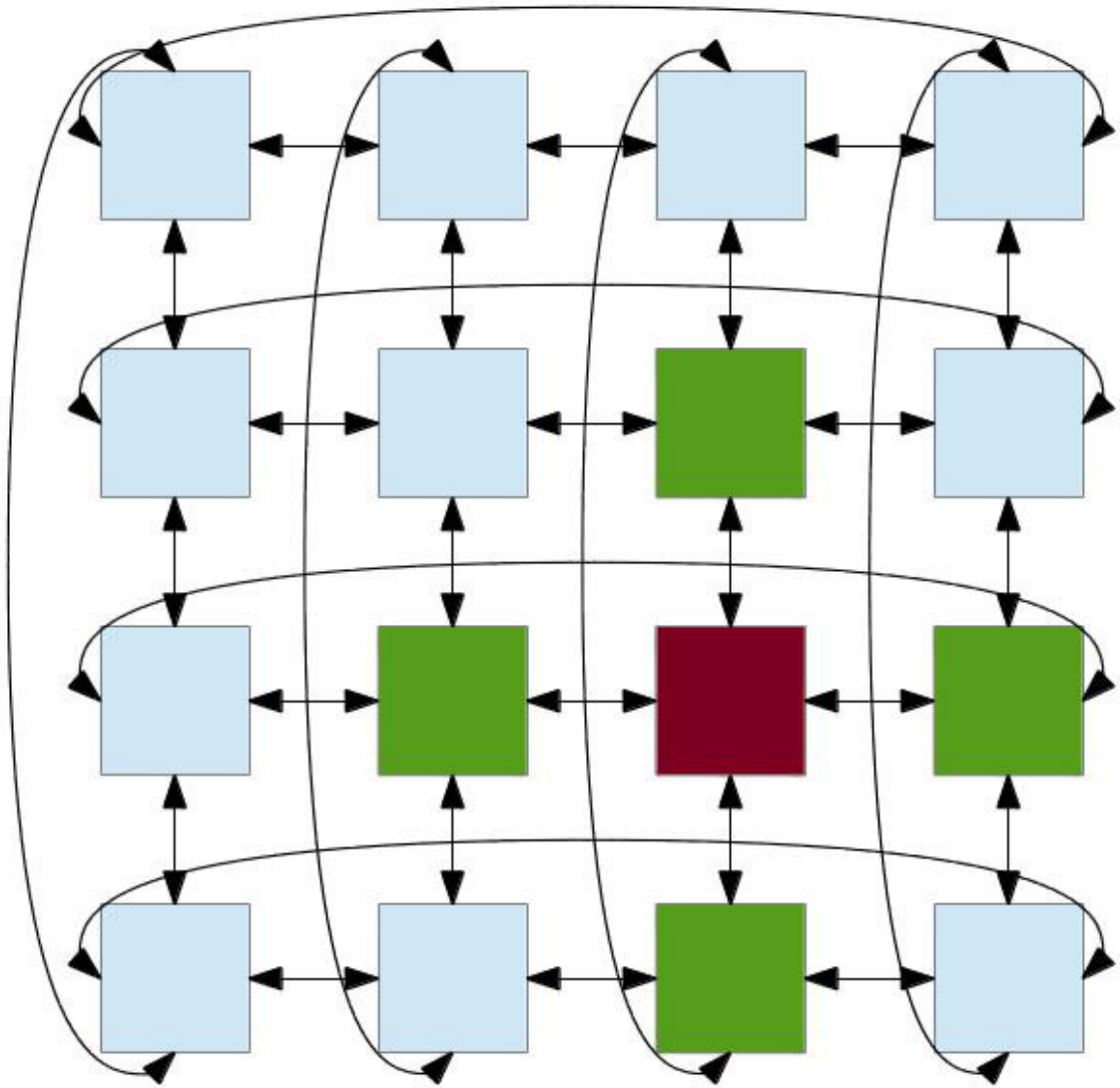
Cellular Genetic Algorithm



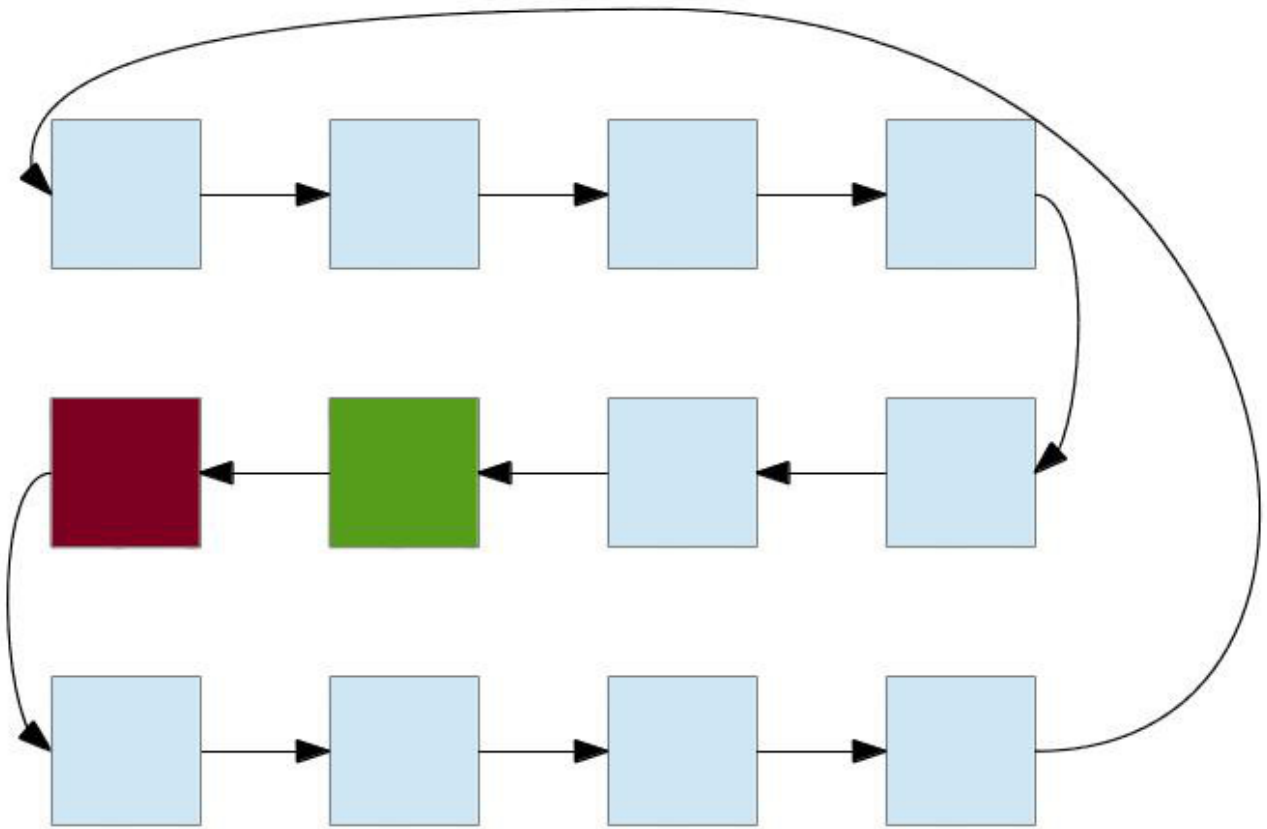
Hardware-motivated topology



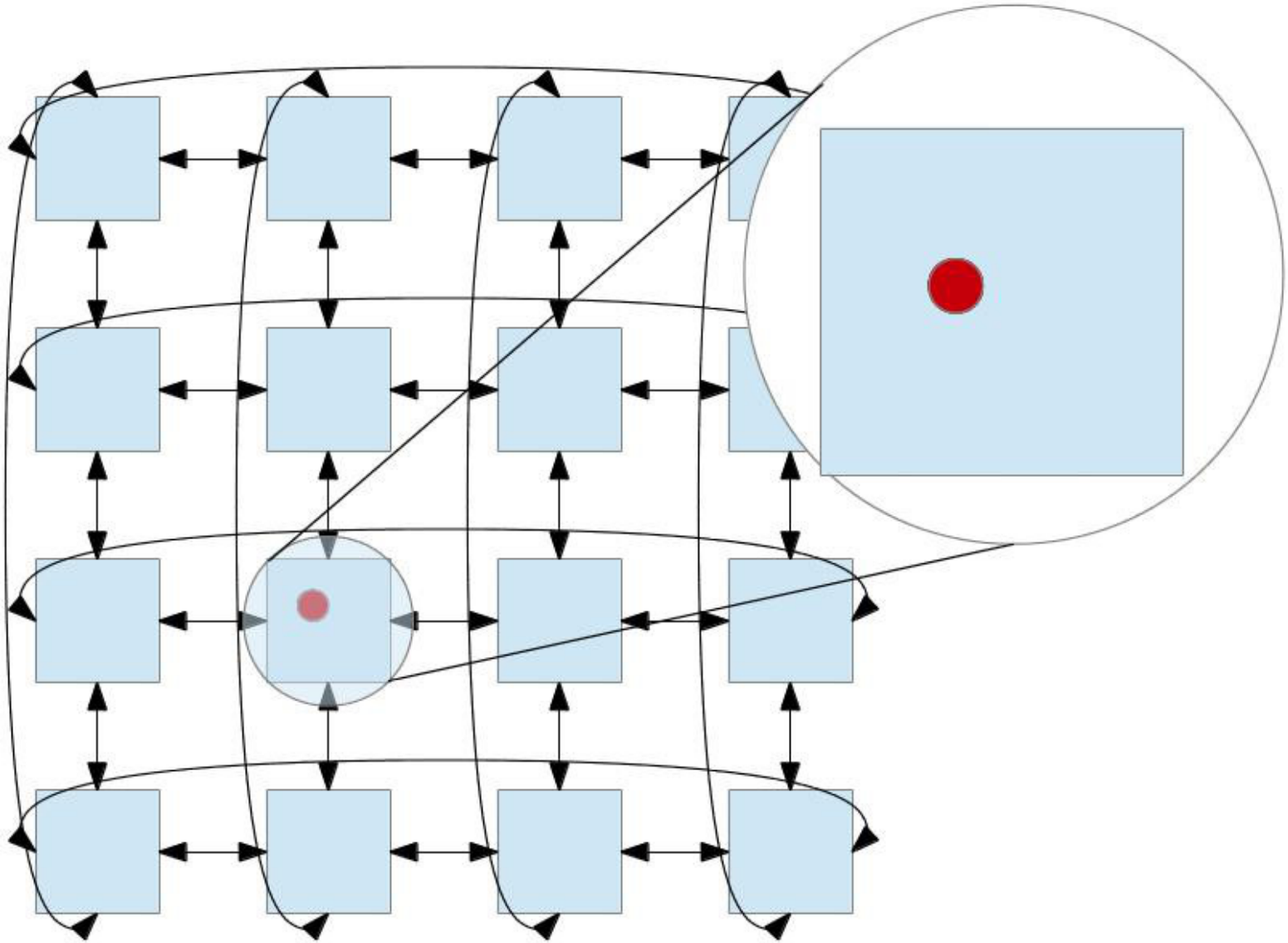
Neighborhoods



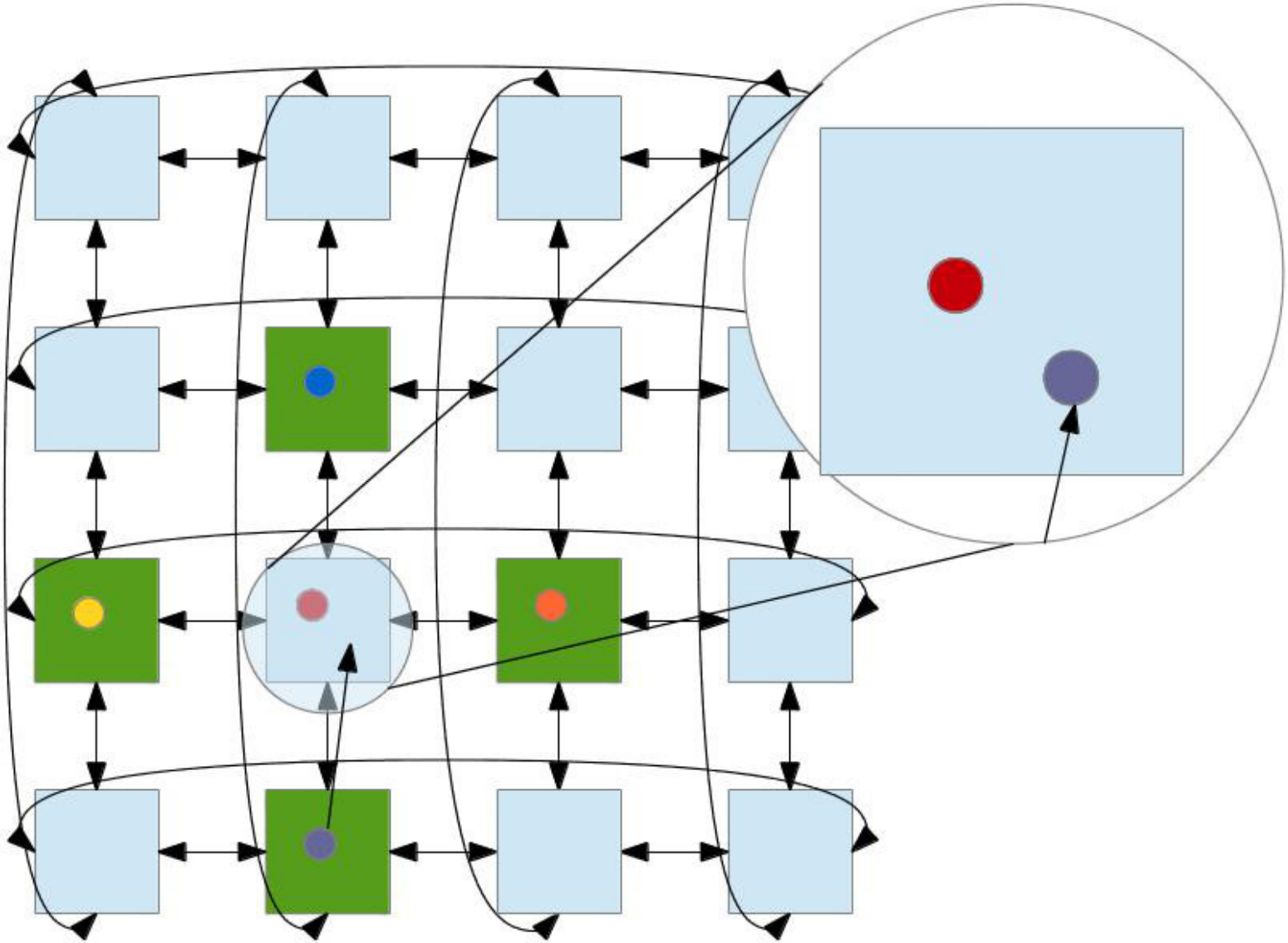
Neighborhoods



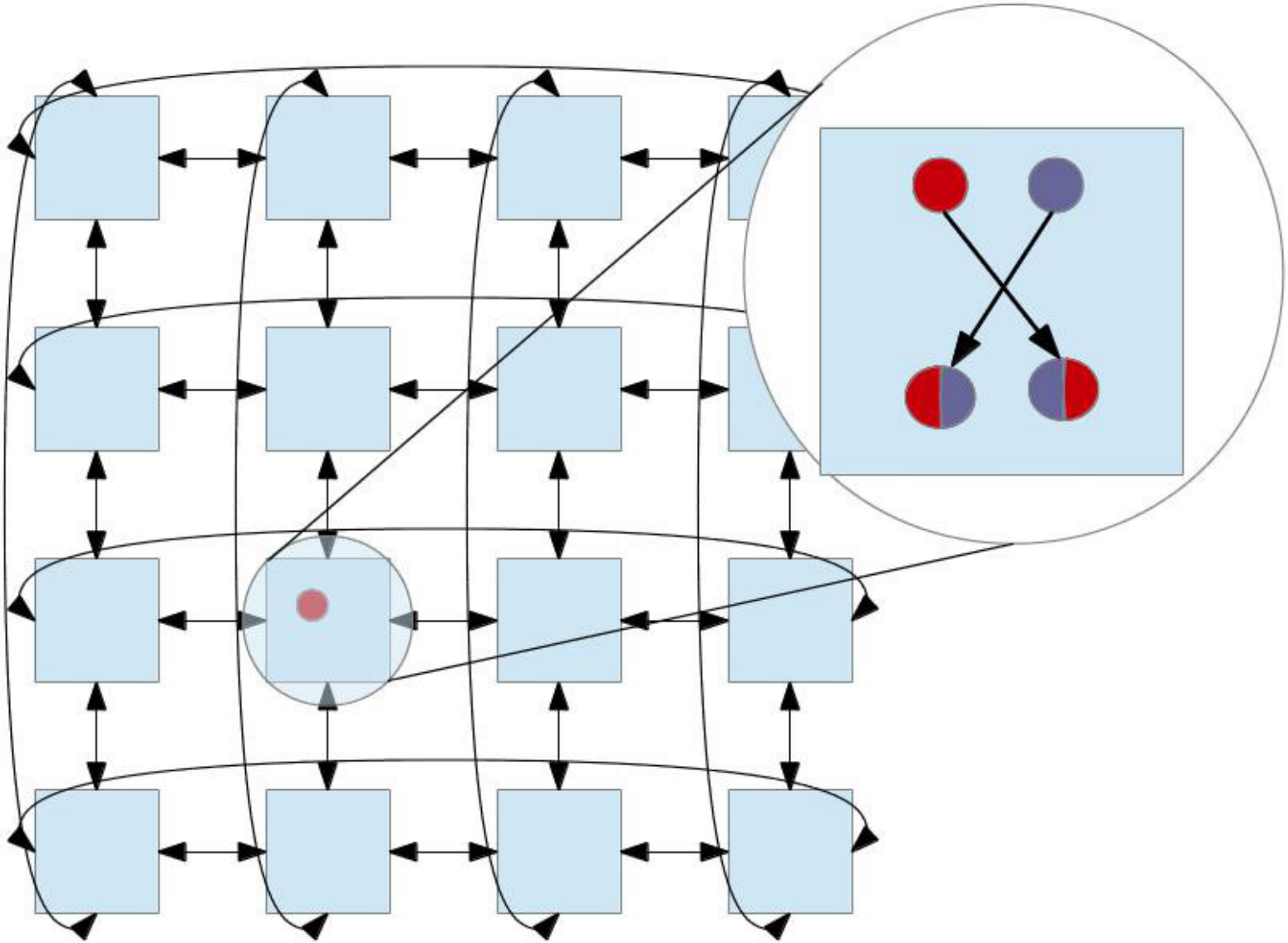
One generation at one node



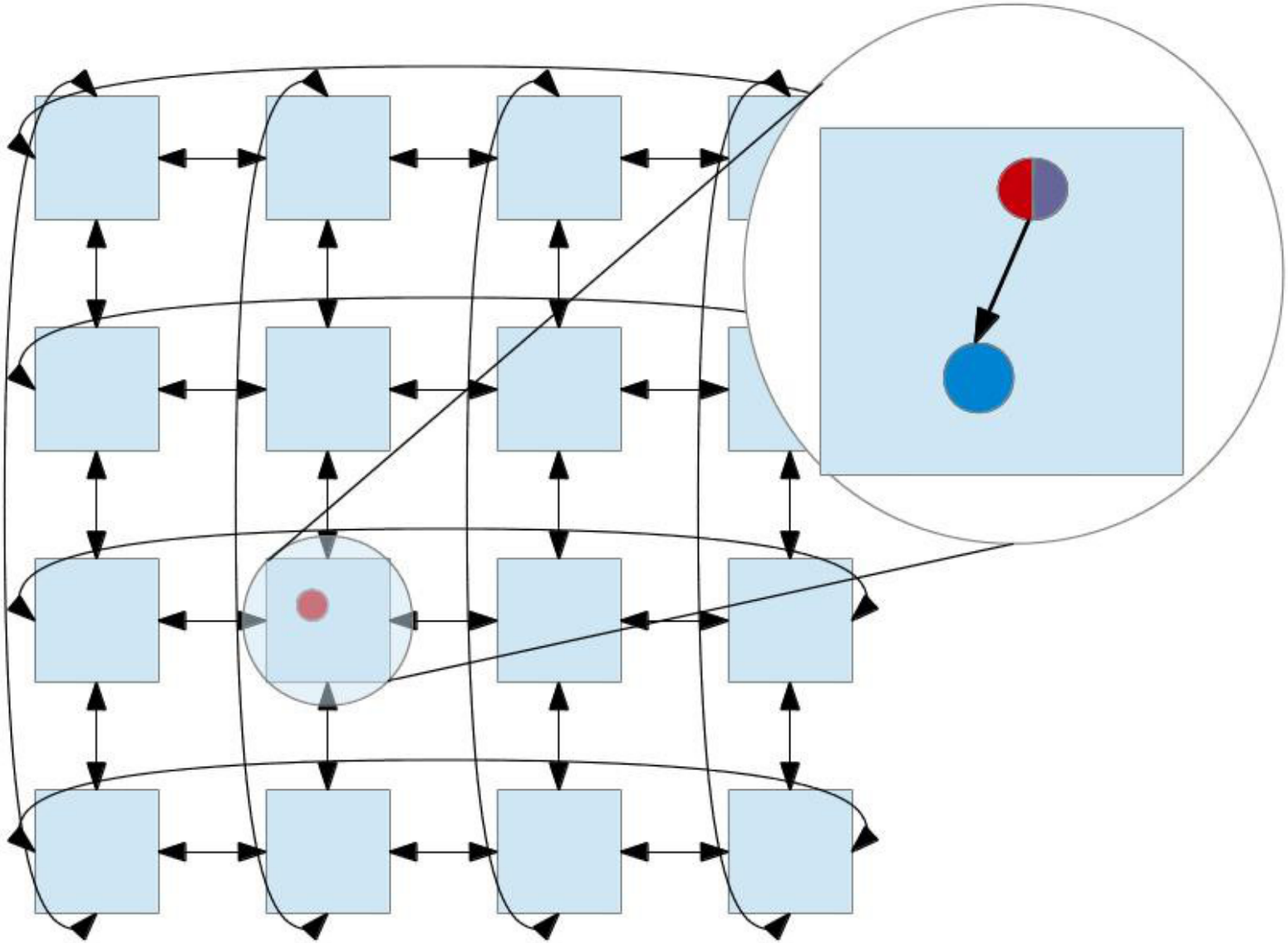
Migration



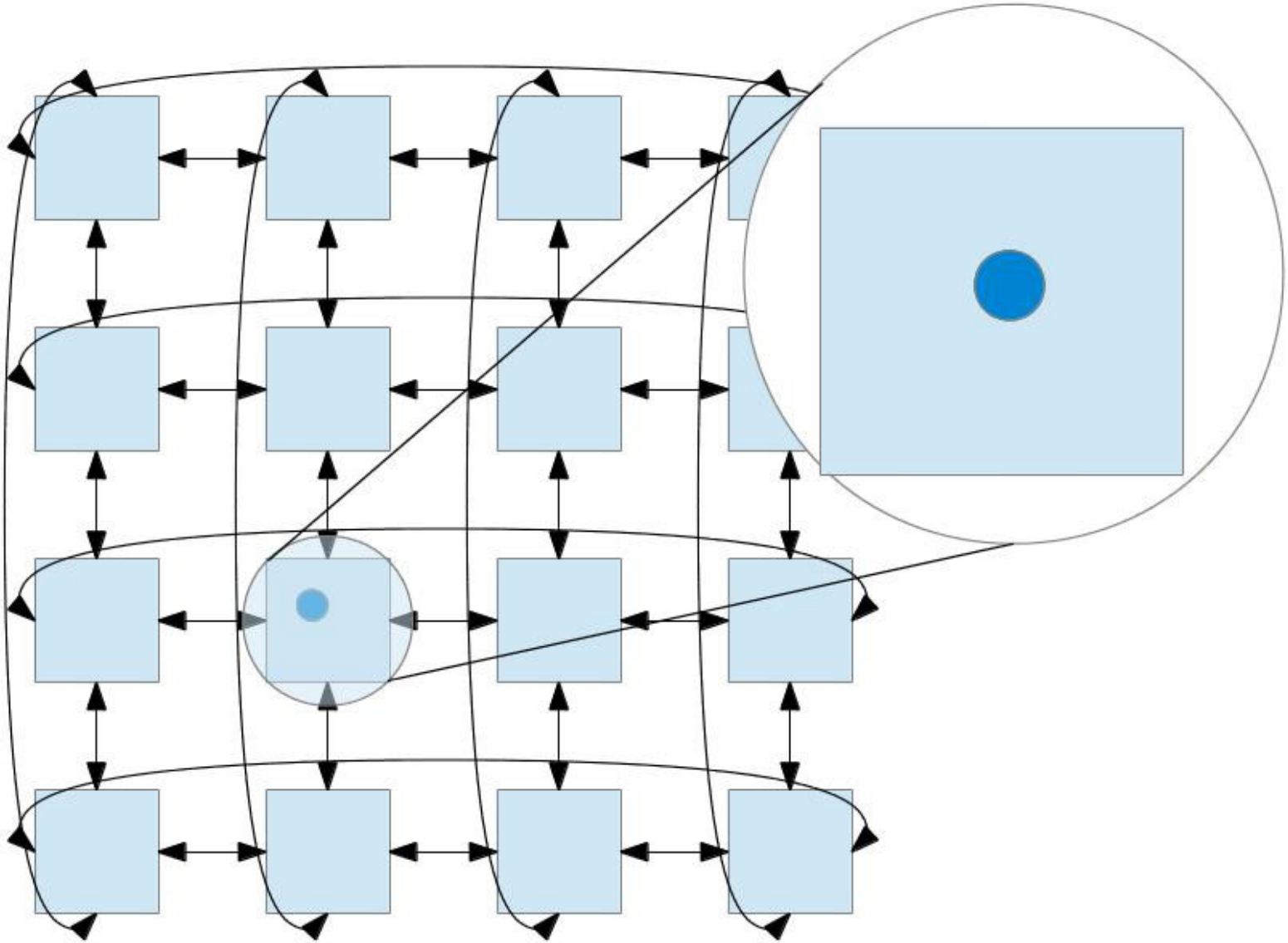
Crossover



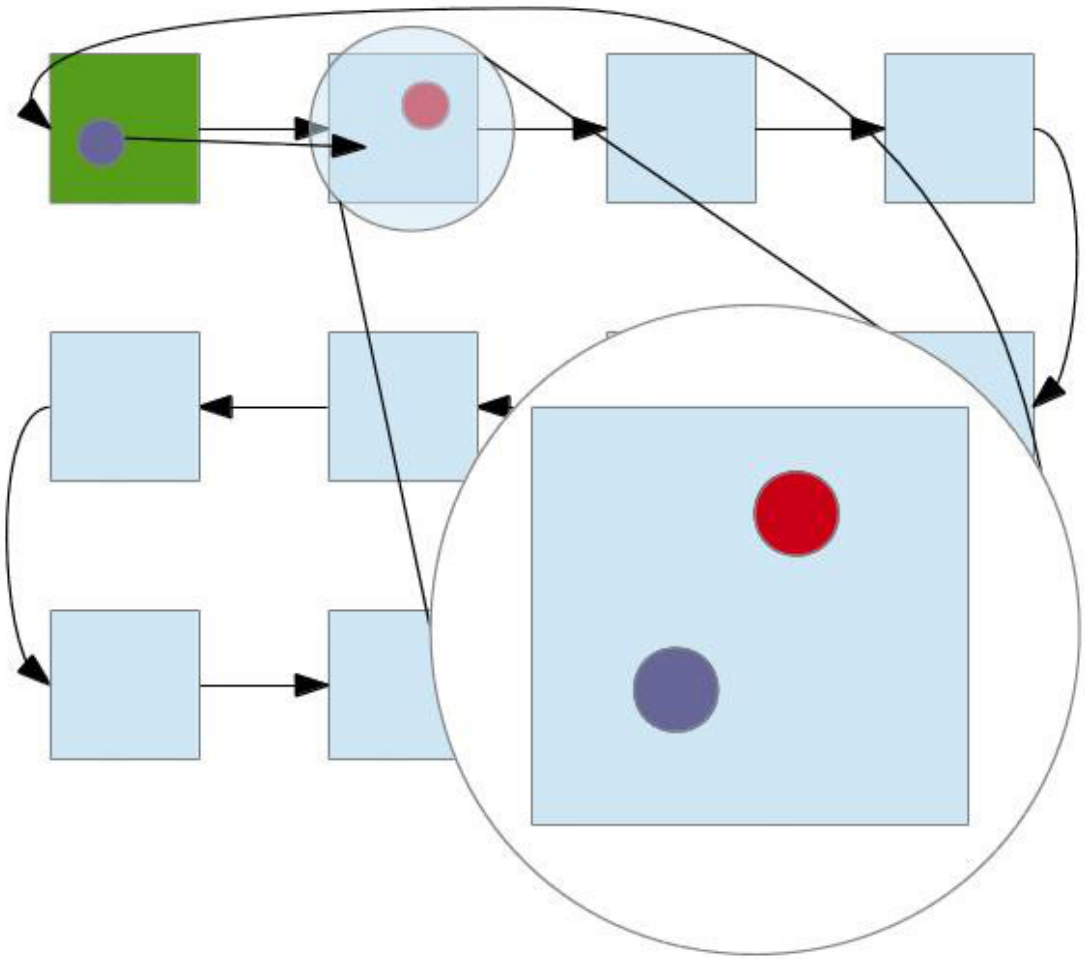
Mutation



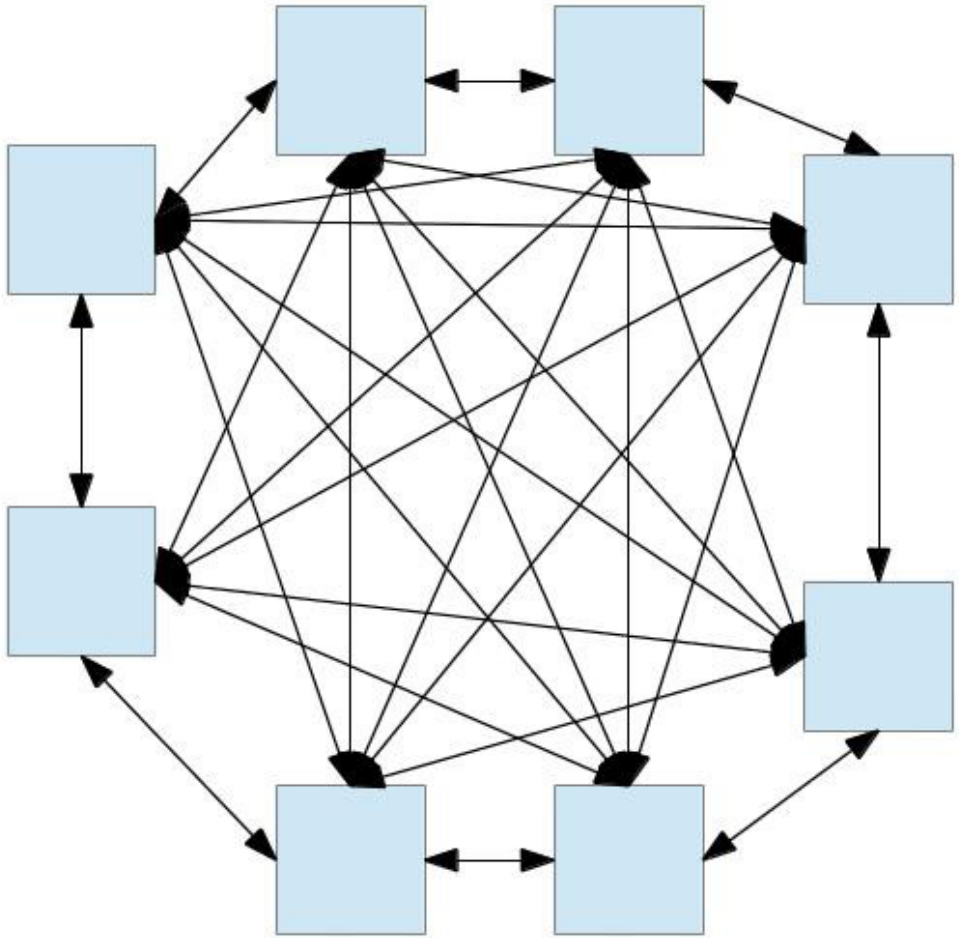
Update



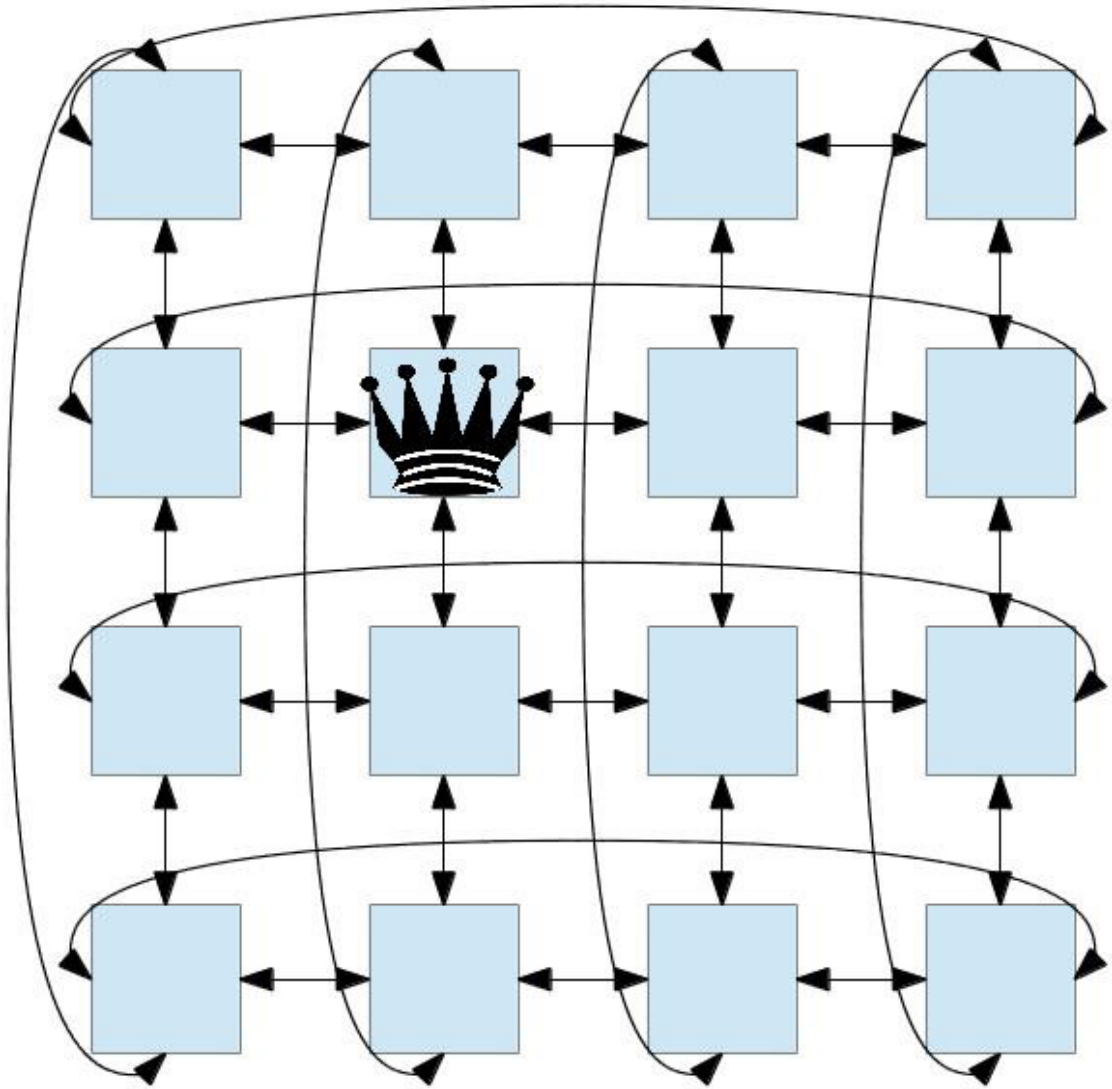
Migration in a Ring CGA



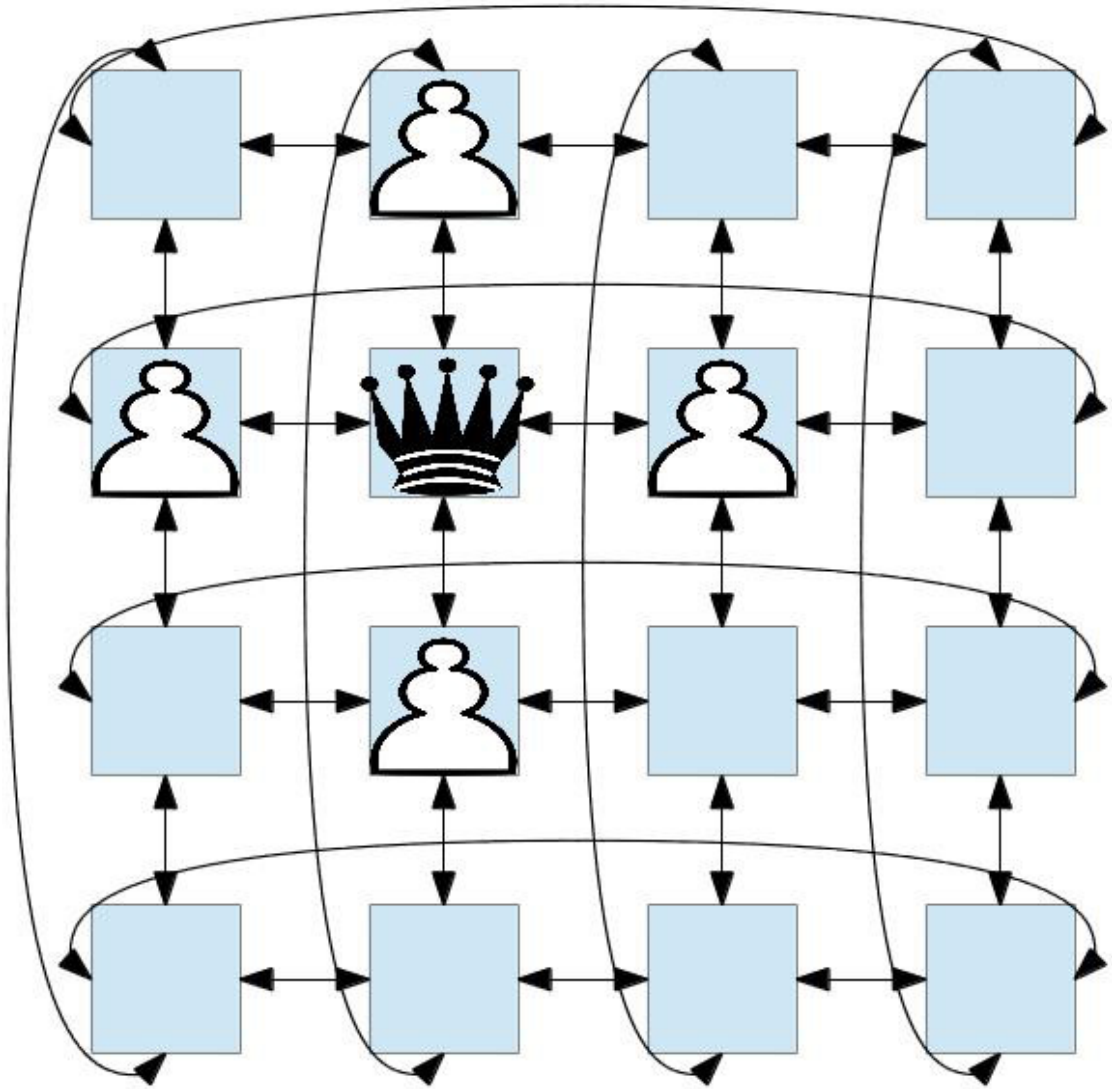
Conventional GA
Population



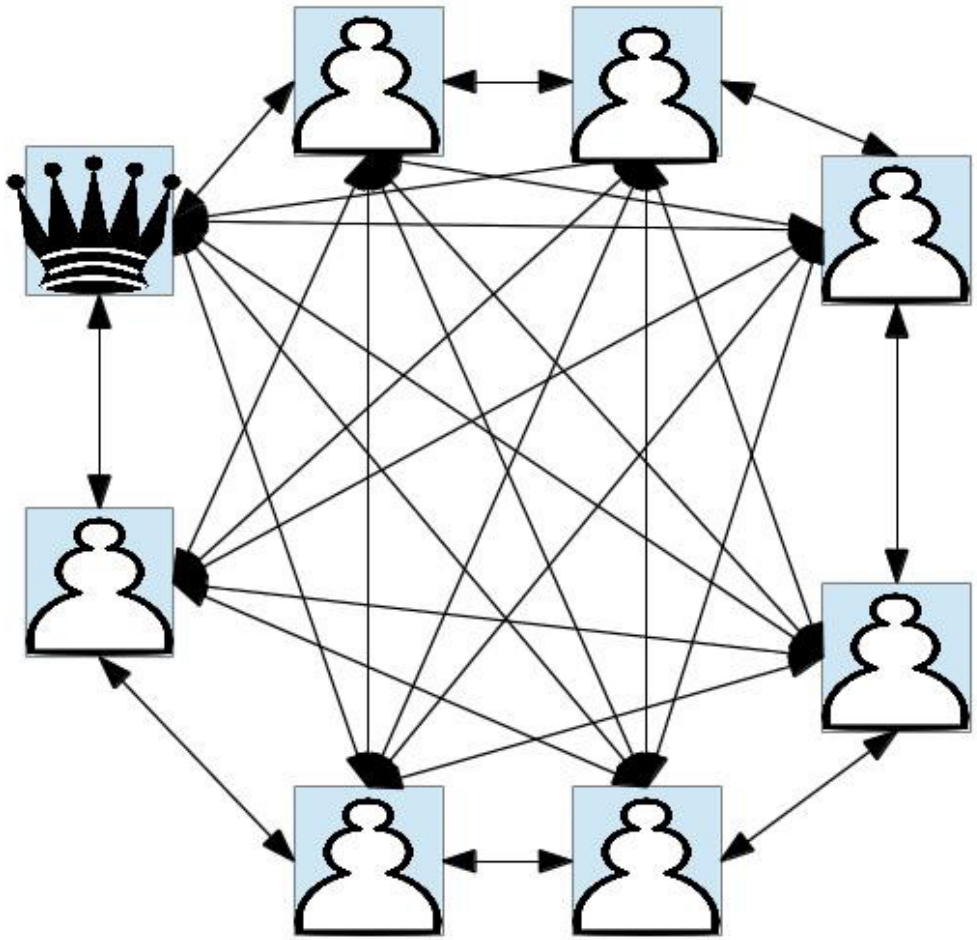
Explore vs. Exploit



Explore vs. Exploit



Explore vs. Exploit



So you want to use a GA on your distributed hardware.

Which GA should you use?

Questions

- Which CGA topology is better? Grid or ring?

Questions

- Which CGA topology is better? Grid or ring?
- How does CGA compare to classic GA or random search?

Questions

- Which CGA topology is better? Grid or ring?
- How does CGA compare to classic GA or random search?
- How does CGA compare to a state-of-the-art GA? (Tran and Jin)

Questions

- Which CGA topology is better? Grid or ring?
- How does CGA compare to classic GA or random search?
- How does CGA compare to a state-of-the-art GA? (Tran and Jin)
- How much does operator choice affect performance?

Experimental Design

- Algorithms: CGA, GA, random search
- Population Sizes: 16, 49, and 100
- Mutation rate: 1/dimensionality
- Mutation Op: Gaussian with 20% variance
- Crossover rate: 90%
- Crossover Op: Two point crossover
- Evaluation limit: 50,000*dimensionality

Two-point Crossover

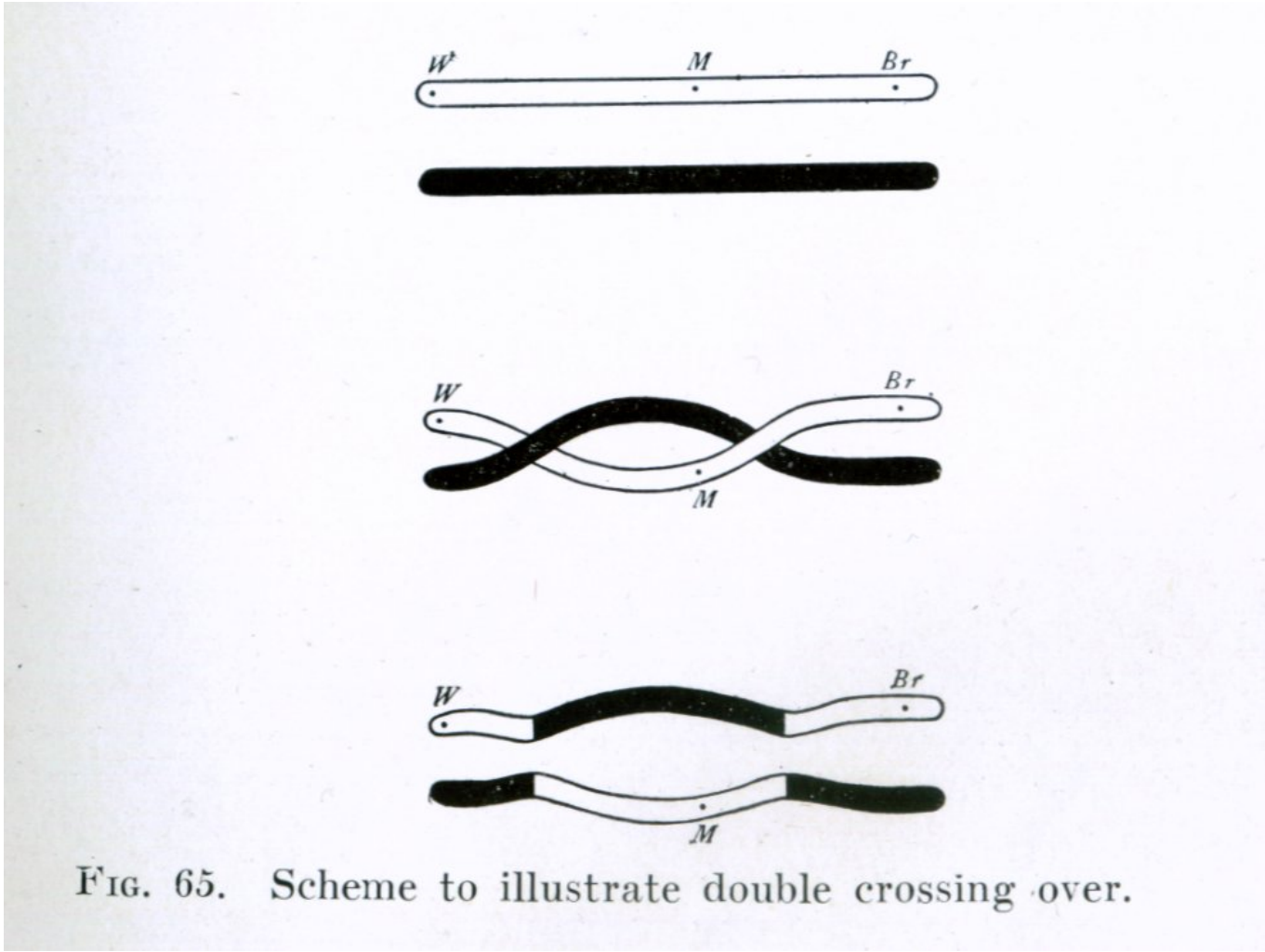
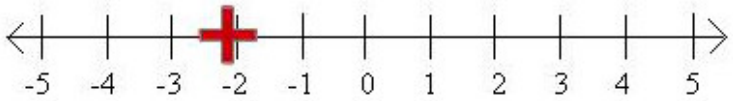


FIG. 65. Scheme to illustrate double crossing over.

http://fga.sourceforge.net/doc/tutorial.html

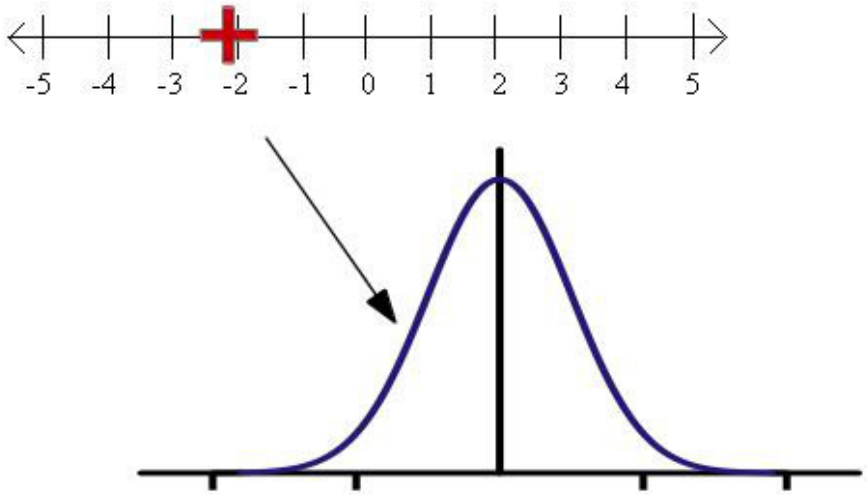
Gaussian Mutation

Value to be mutated



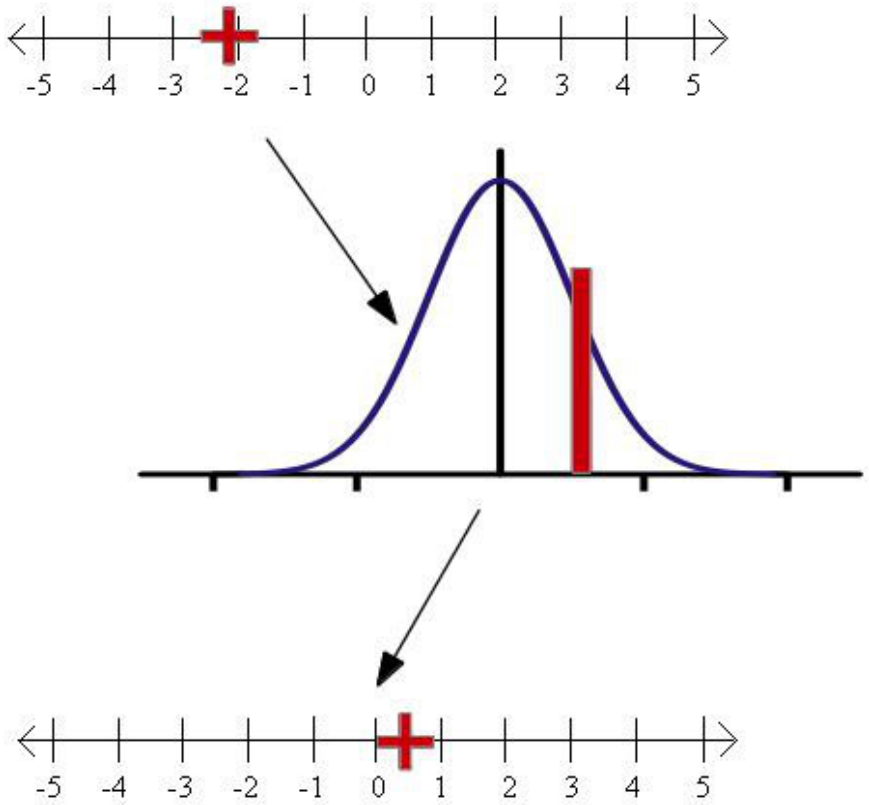
Gaussian Mutation

Sample from a normal distribution



Gaussian Mutation

Add the sampled value to the value to be mutated



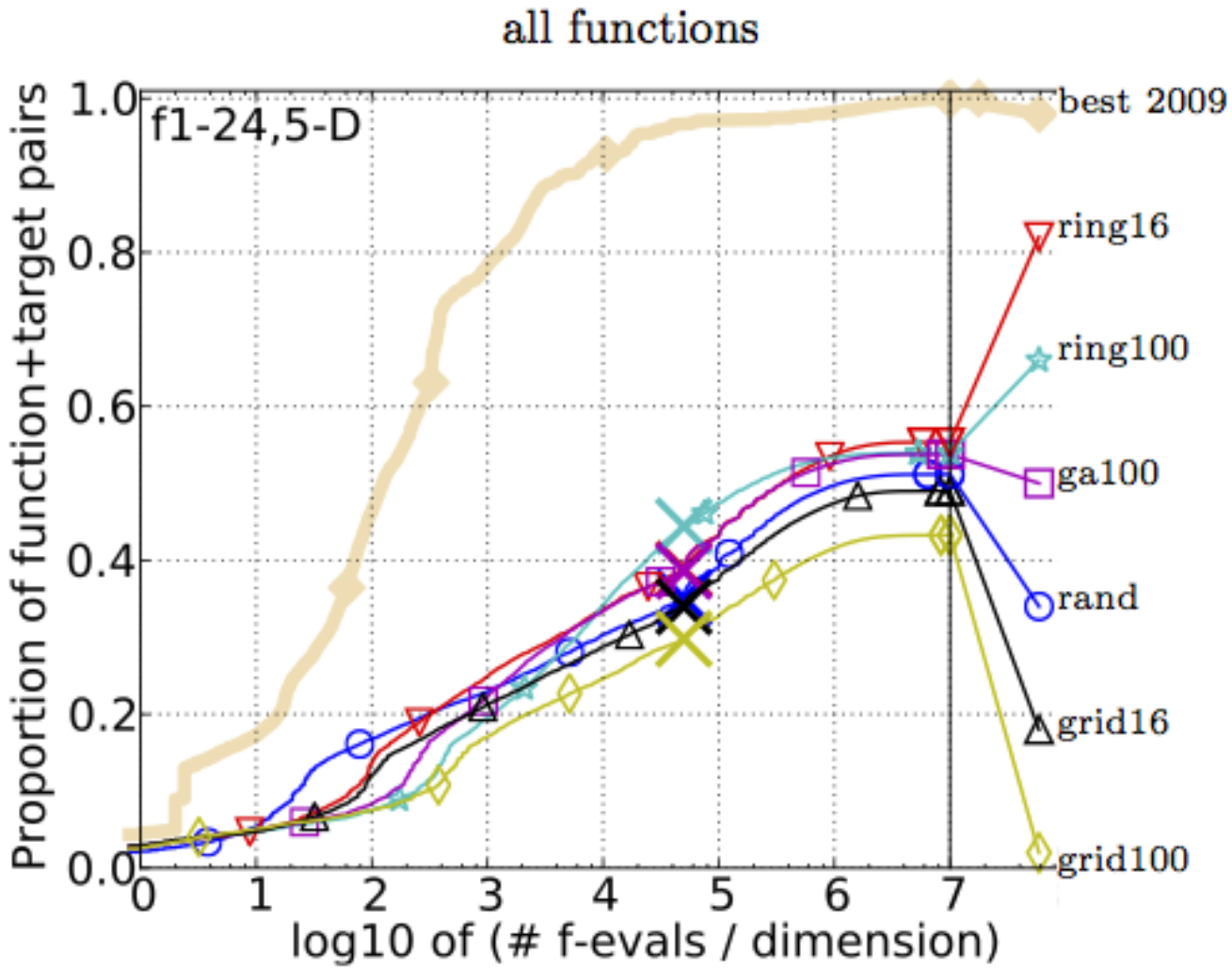
Evaluated algorithms

- Cellular Genetic Algorithm
  - Grid
  - Ring
- Standard Genetic Algorithm
- Random search

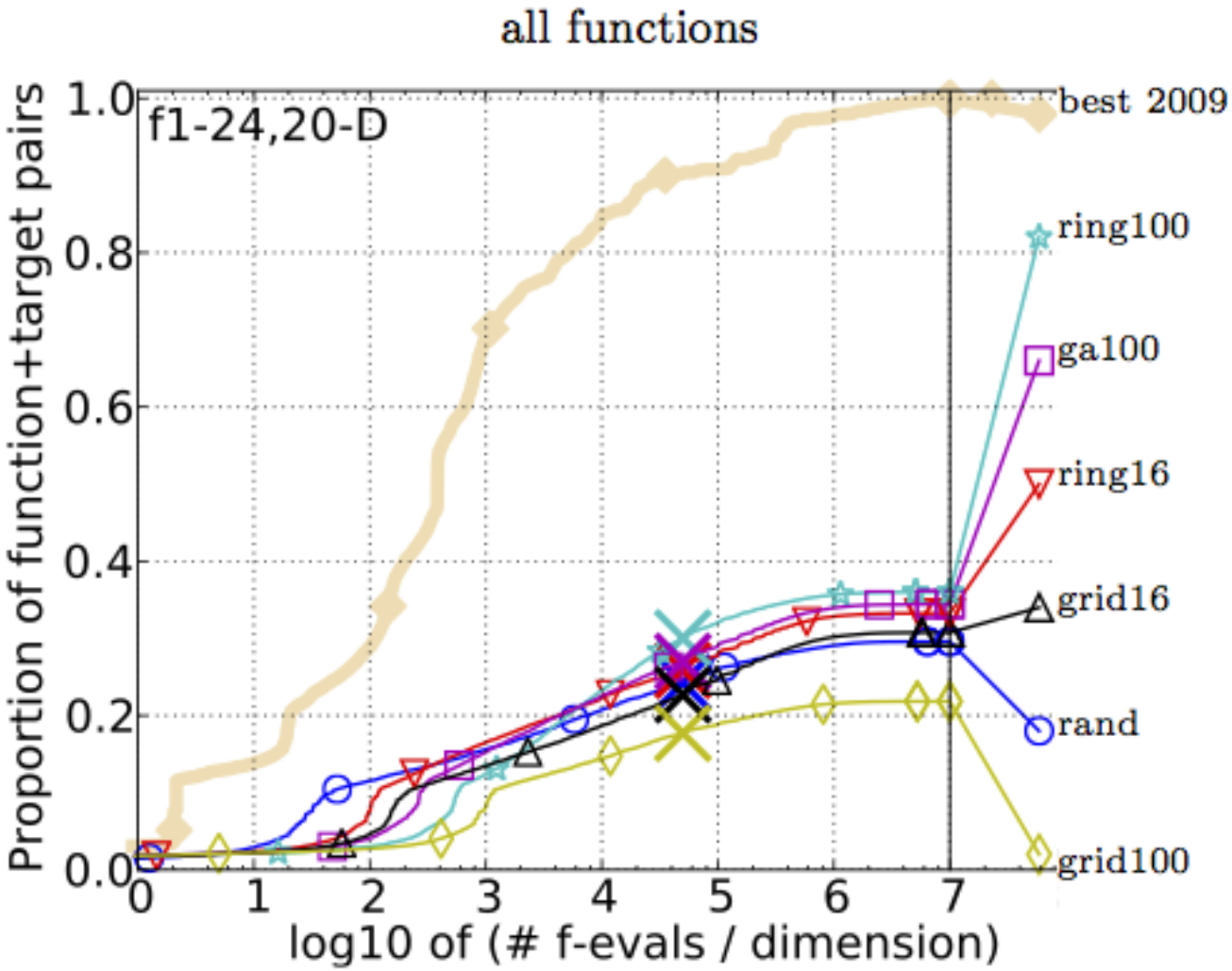
Which CGA topology to use?

Use Ring

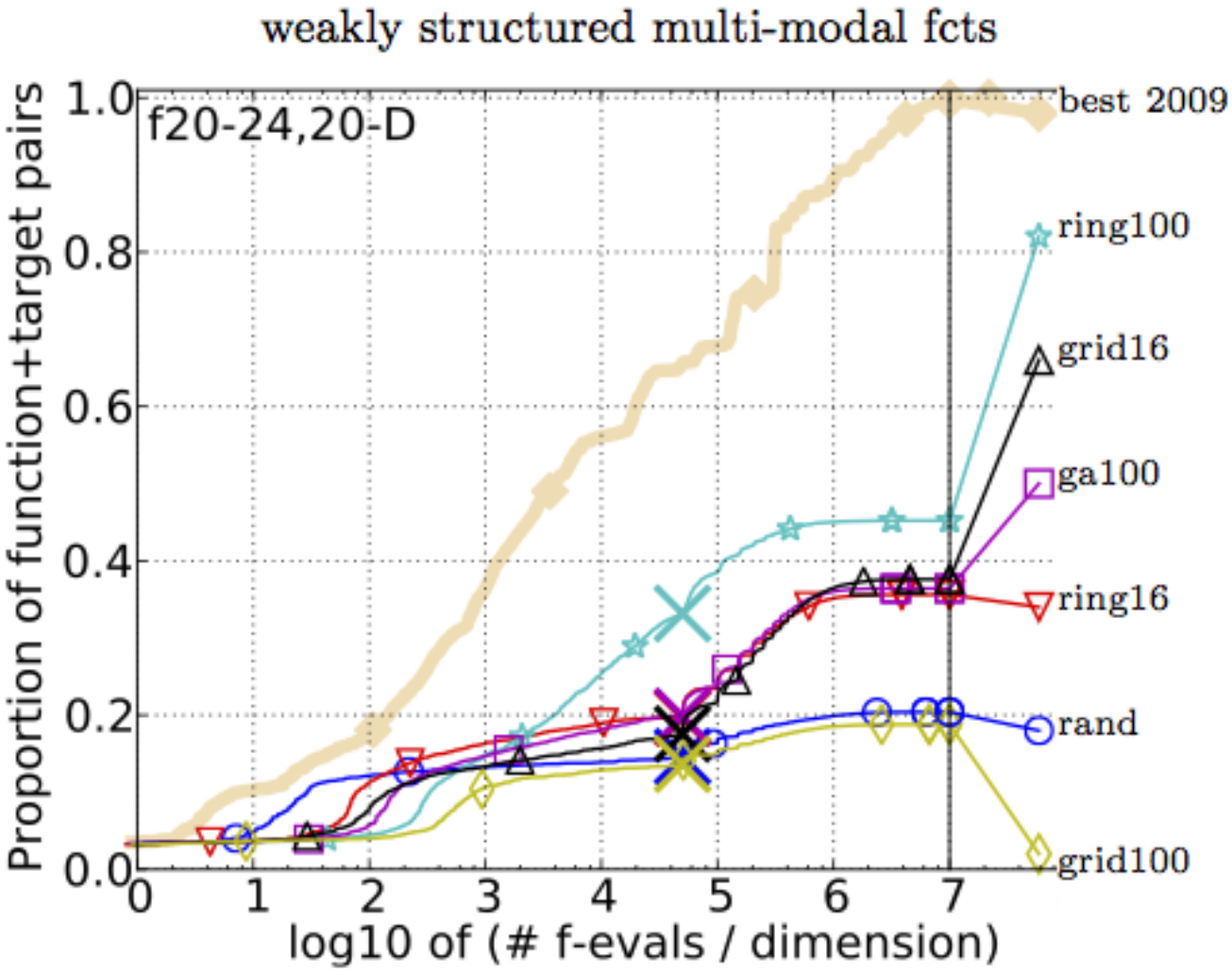
5-D All Functions



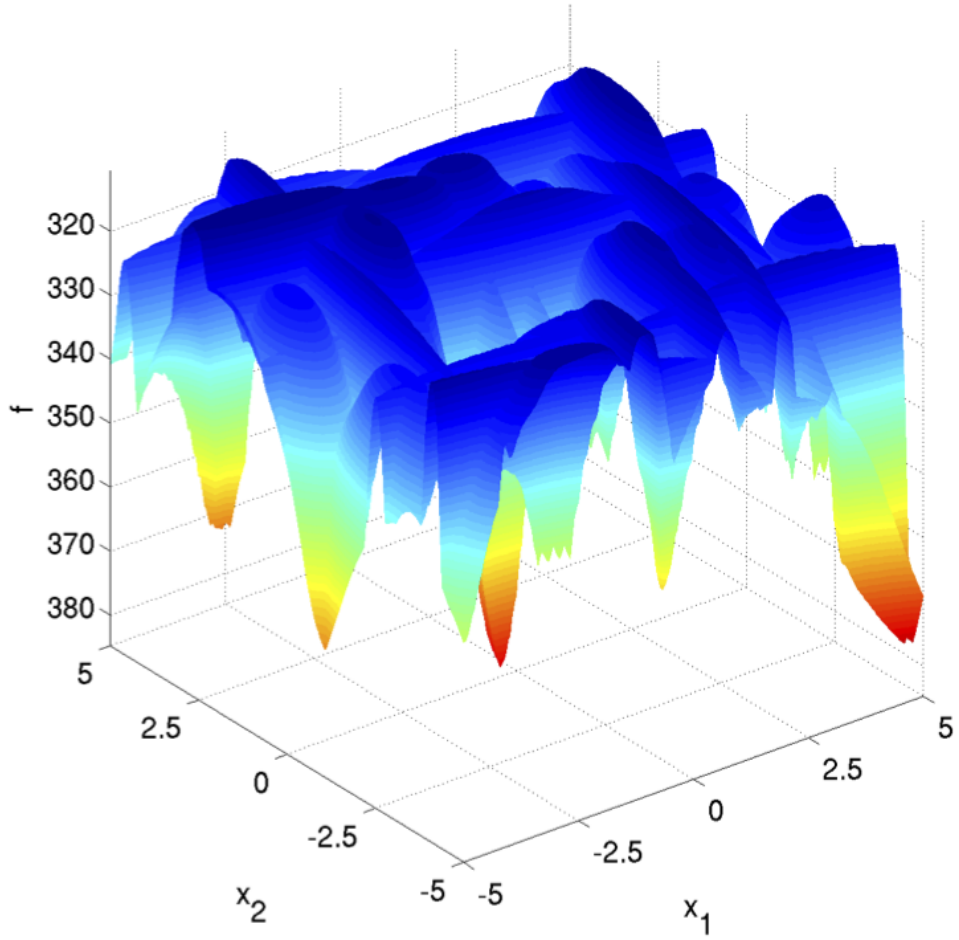
20-D All Functions



20-D Weakly Structured

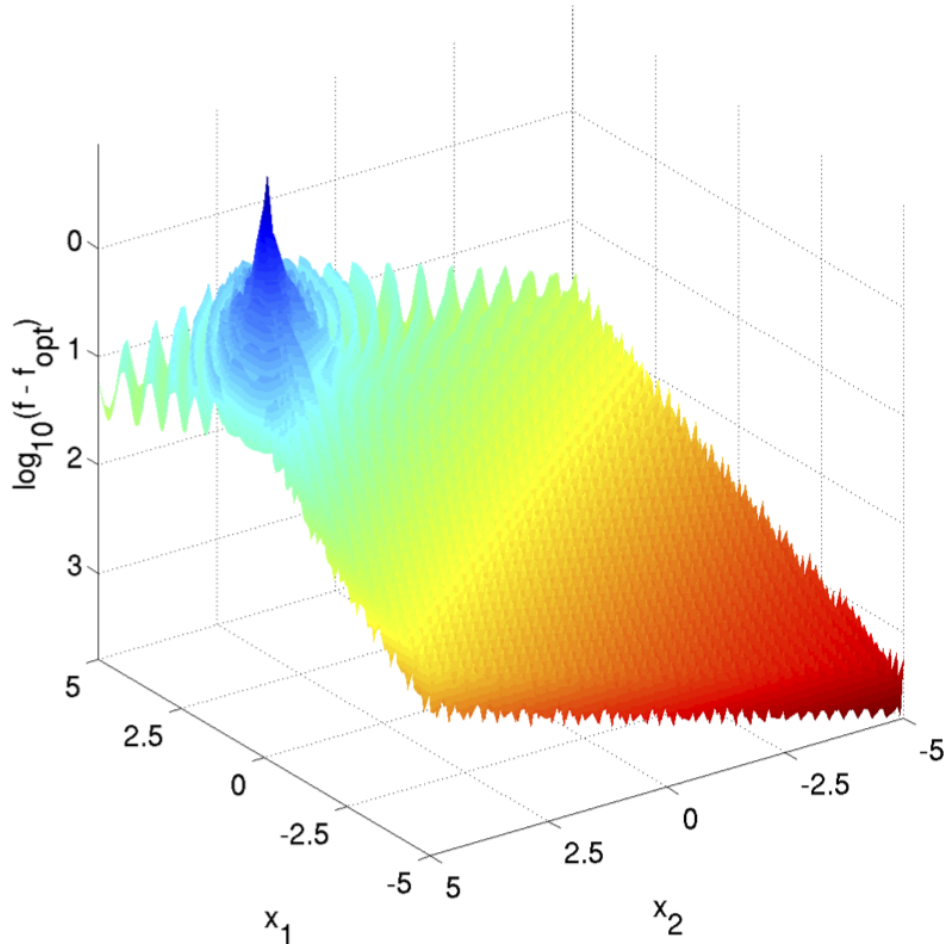


Brief aside: What is Weakly Structured?



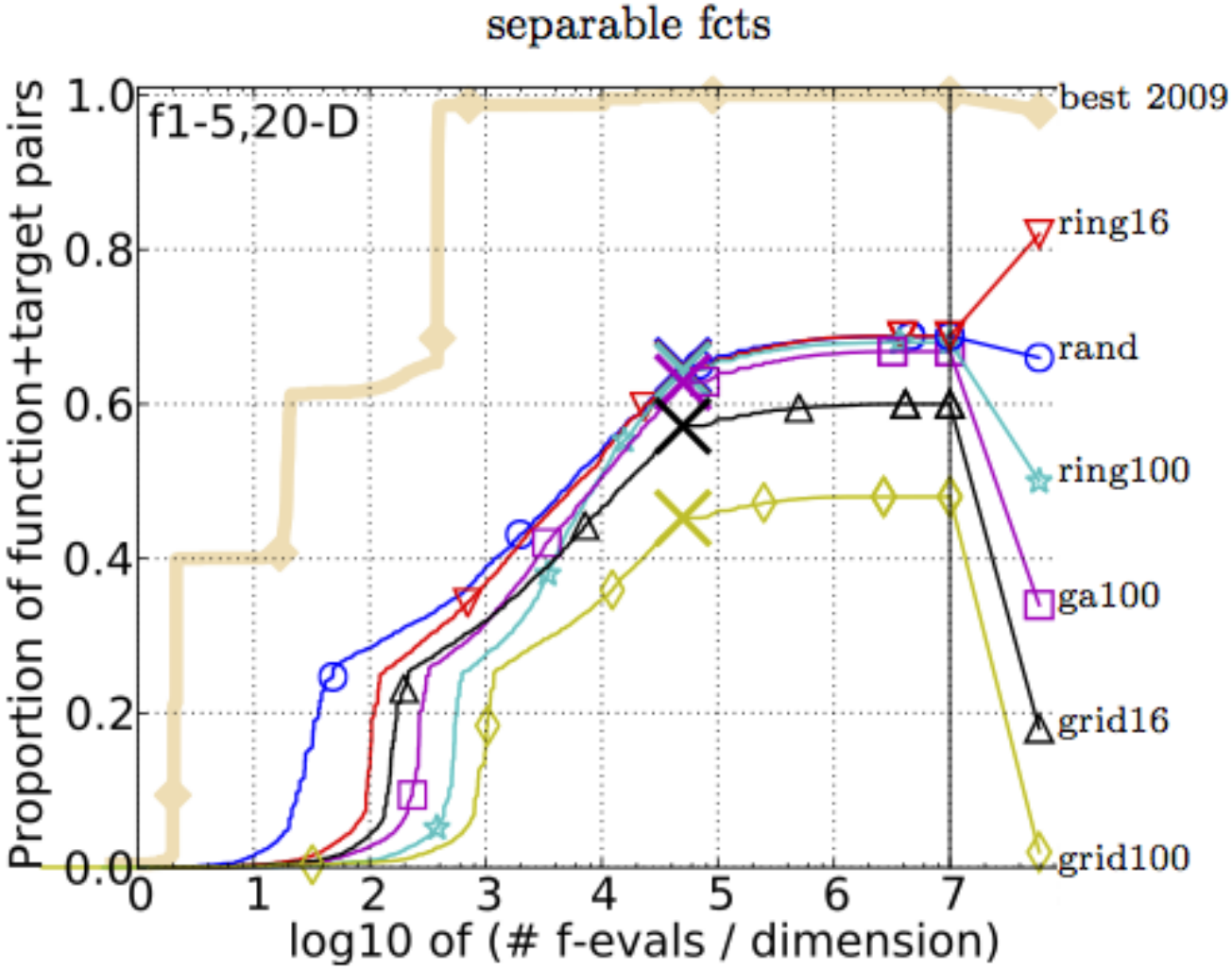
Gallagher's Gaussian 101-me Peaks Function

Brief aside: What is Weakly Structured?



Schaffers F7 Function

20-D Separable



Apples to Apples

Real-coded genetic algorithm benchmarked on noiseless black-box optimization testbed

by Thanh-Do Tran and Gang-Gyoo Jin

GECCO 2010

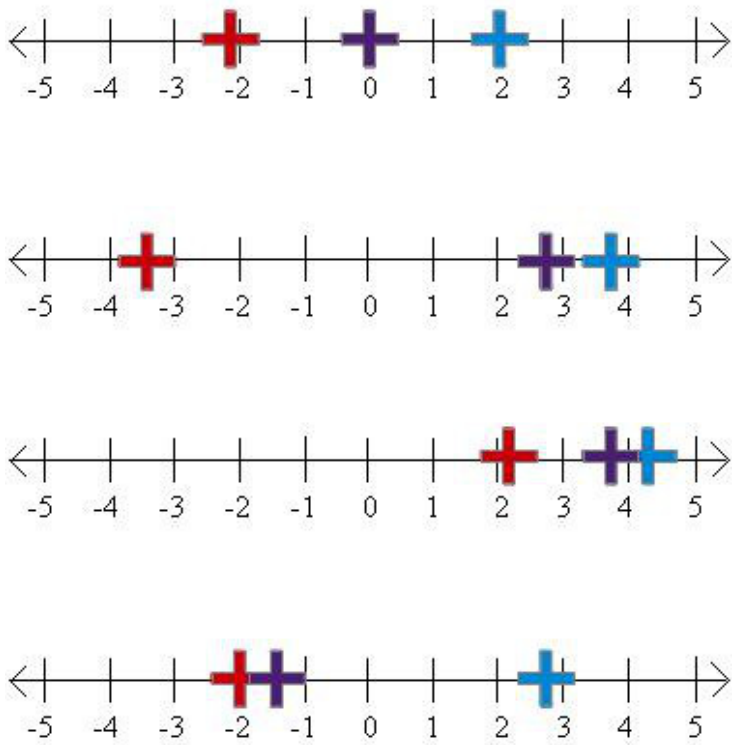
Tran & Jin's RCGA

Population Size	100
Crossover rate	0.95
Tournament size	2
Mutation rate	0.05, 0.1, 0.2
Restart trigger A	After a fixed number of iterations
Restart trigger B	Based on lack of improvement in population fitness

Restarts

Mutation rate	0.05, 0.1, 0.2
Restart trigger A	floor$(100 + 3800D\sqrt{D})$ iterations
Restart trigger B	best population fitness varying less than 10^{-12} in $(50 + 25D)$ generations

Arithmetic crossover

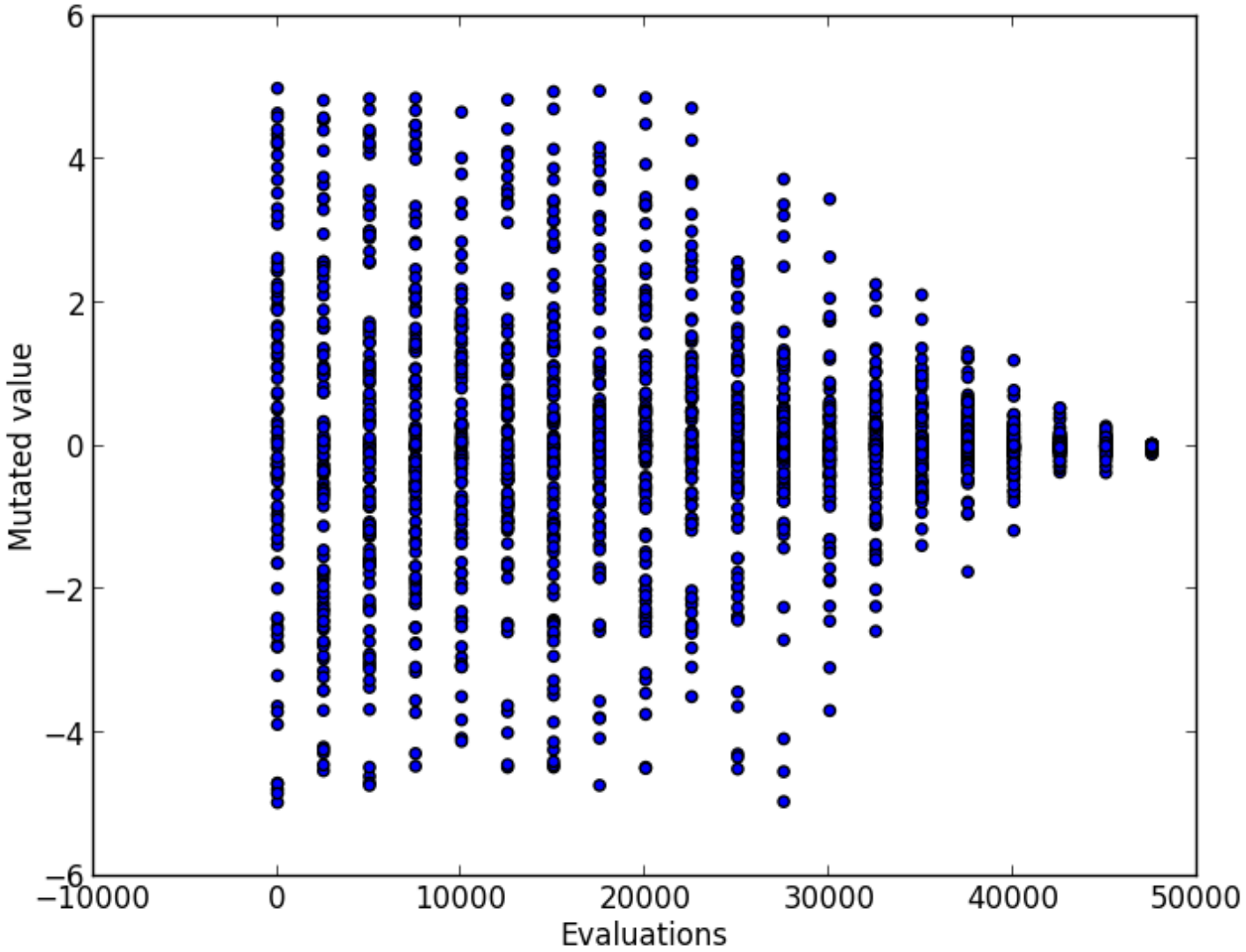


Adaptive range mutation

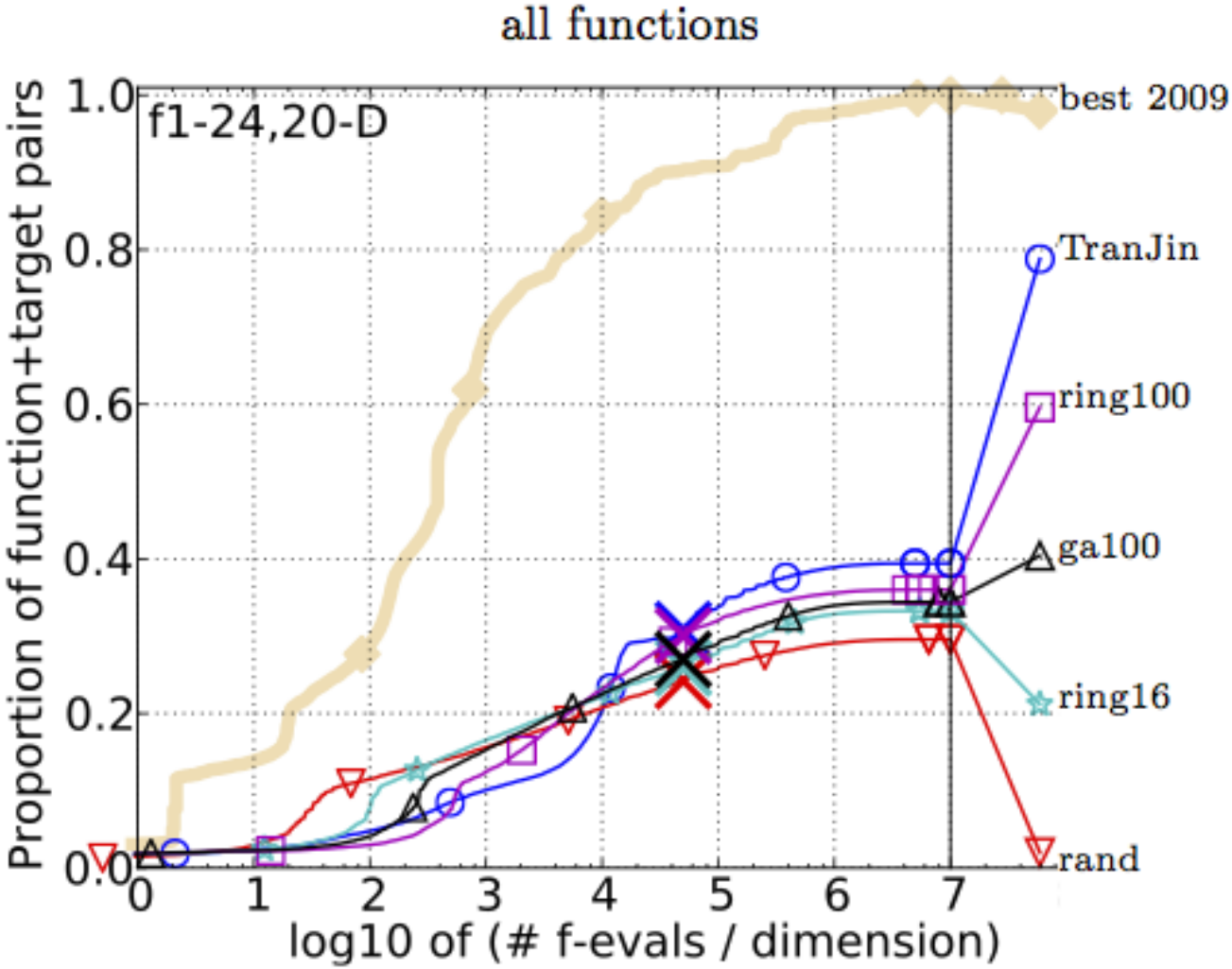
This operator is a modification of Michalewicz non-uniform mutation from:

An adaptive range mutation operator for real-coded genetic algorithms
by Kevin Austin and Peter Jacobs
2001

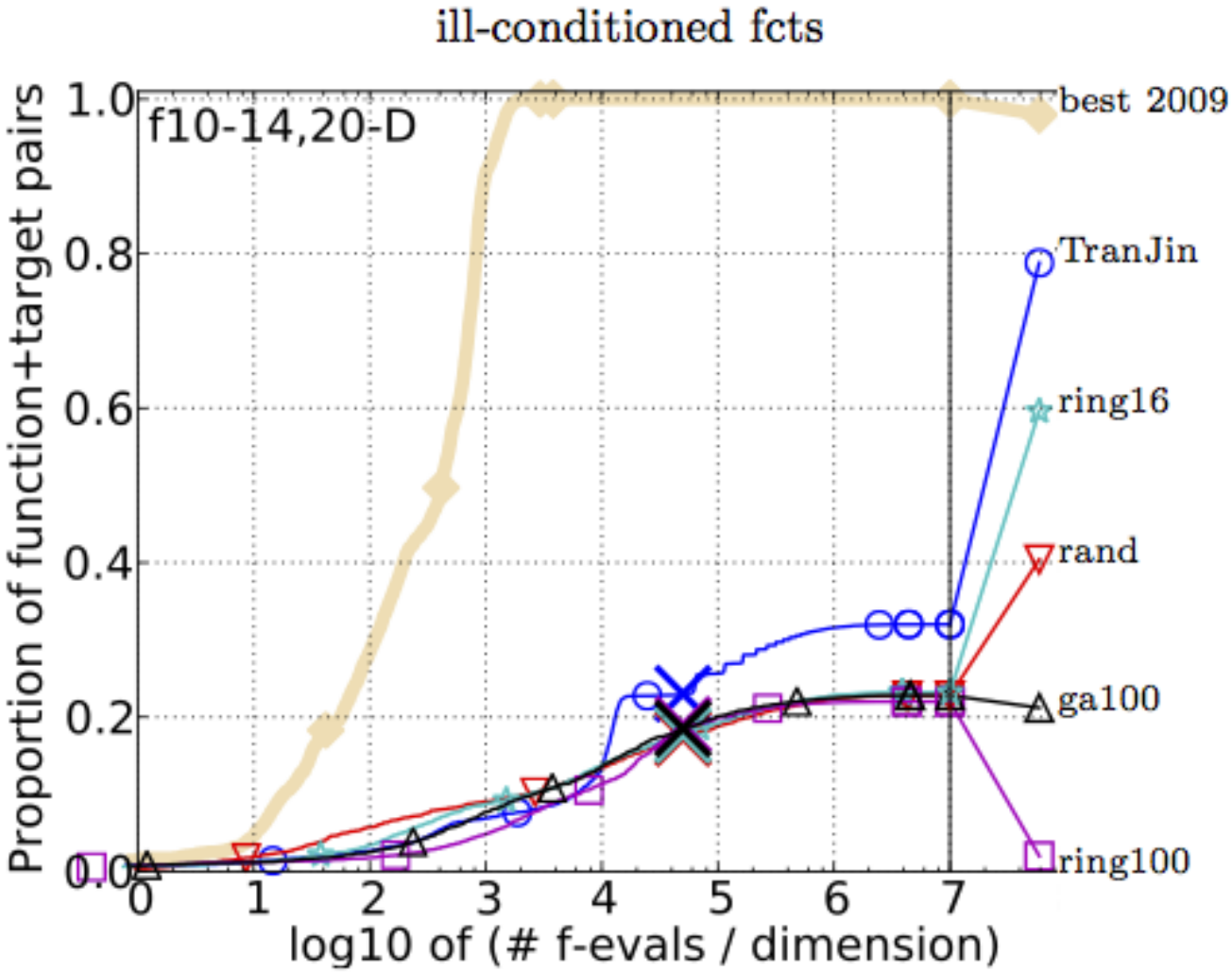
Adaptive range mutation



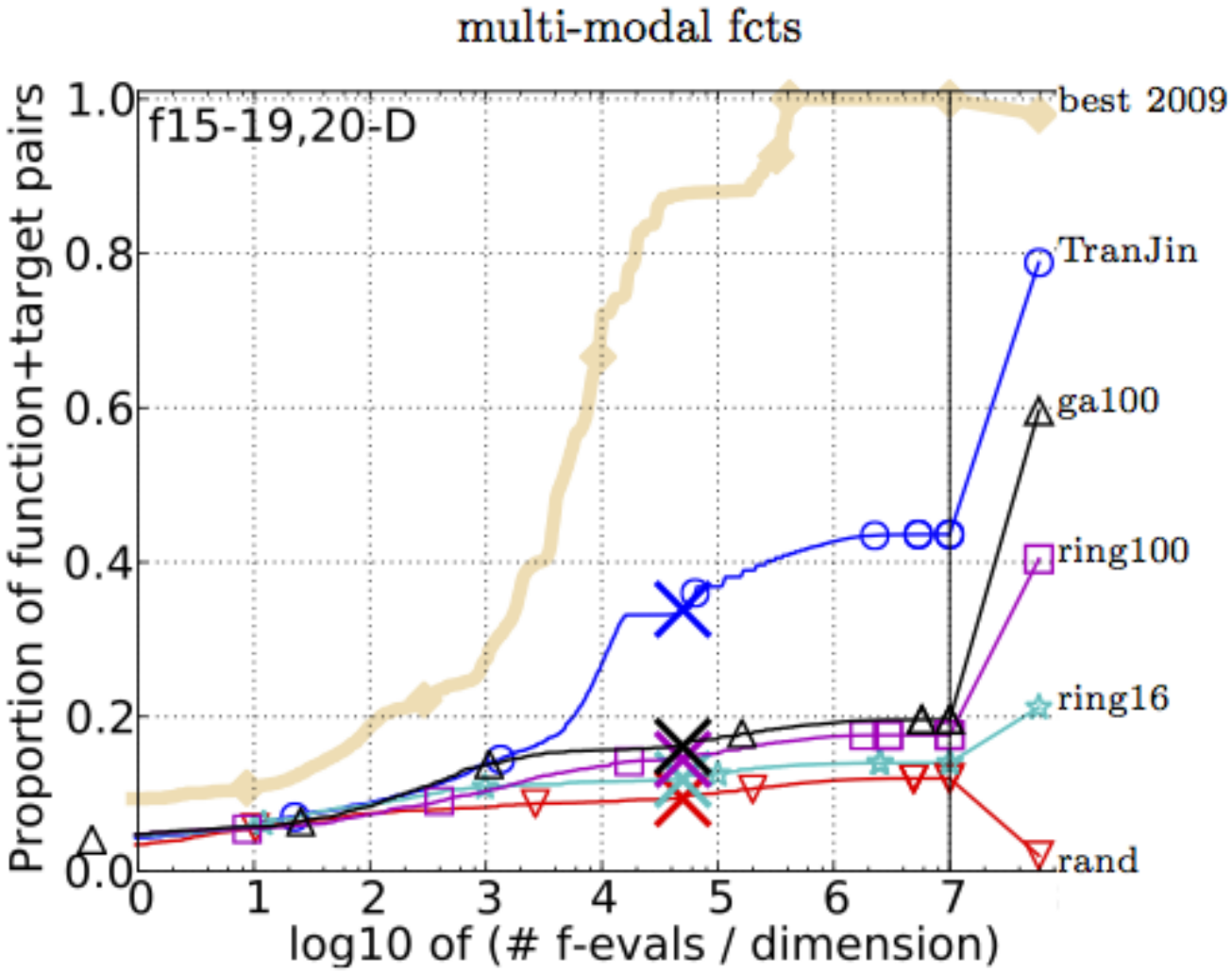
CGA compared unfavorably overall (20-D)



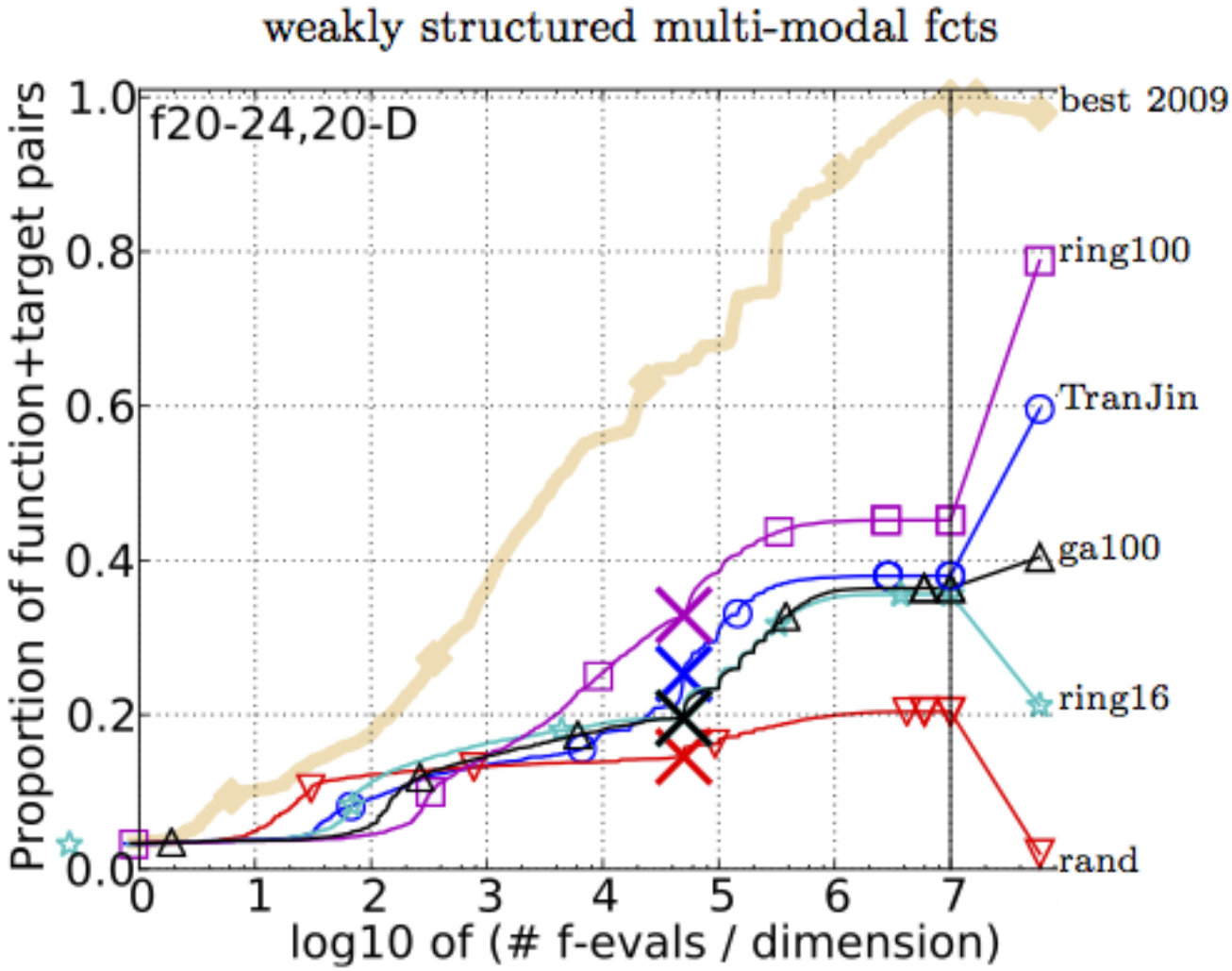
...particularly on ill-conditioned (20-D)



...and multi-modal functions (20-D)



CGA outperformed RCGA on weakly structured functions (20-D)



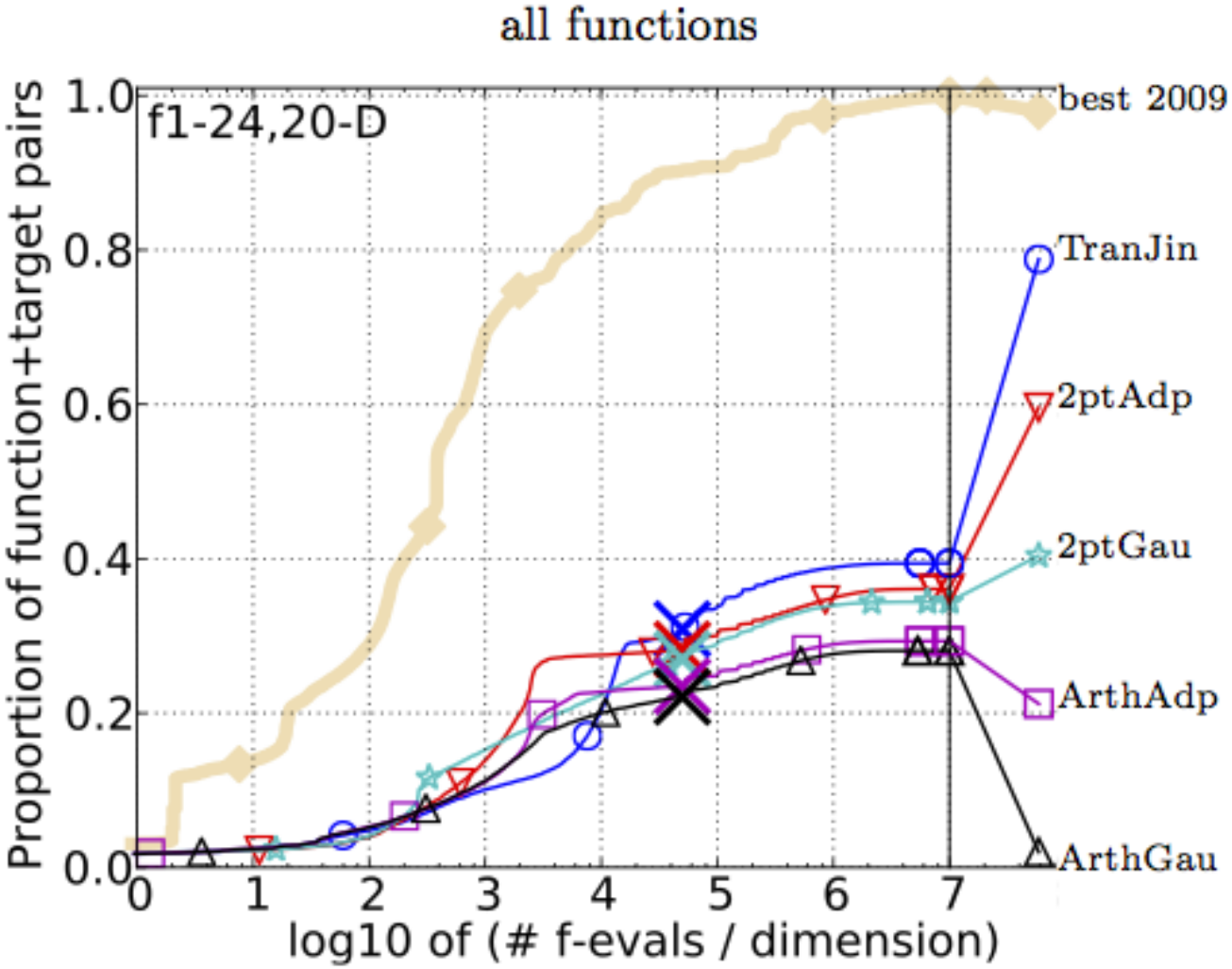
Interesting operator

Operator comparison

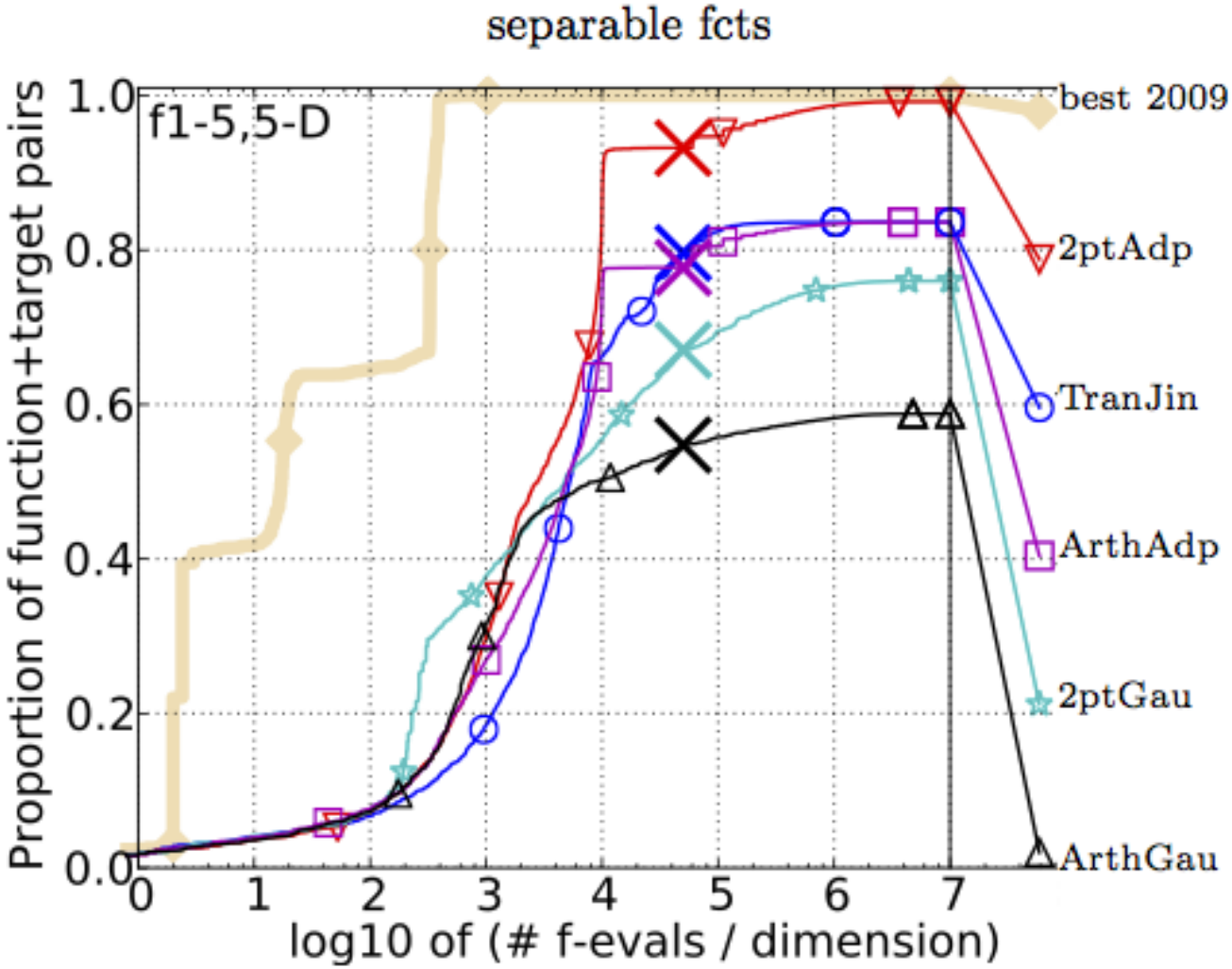
Algorithm name	Crossover Operator	Mutation Operator	Restarts
2ptAdp	2-point	Adaptive	no
2ptGau	2-point	Gaussian	no
ArthAdp	Arithmetic	Adaptive	no
ArthGau	Arithmetic	Gaussian	no
TranJin	Arithmetic	Adaptive	yes

Population size 100

20-D All Functions

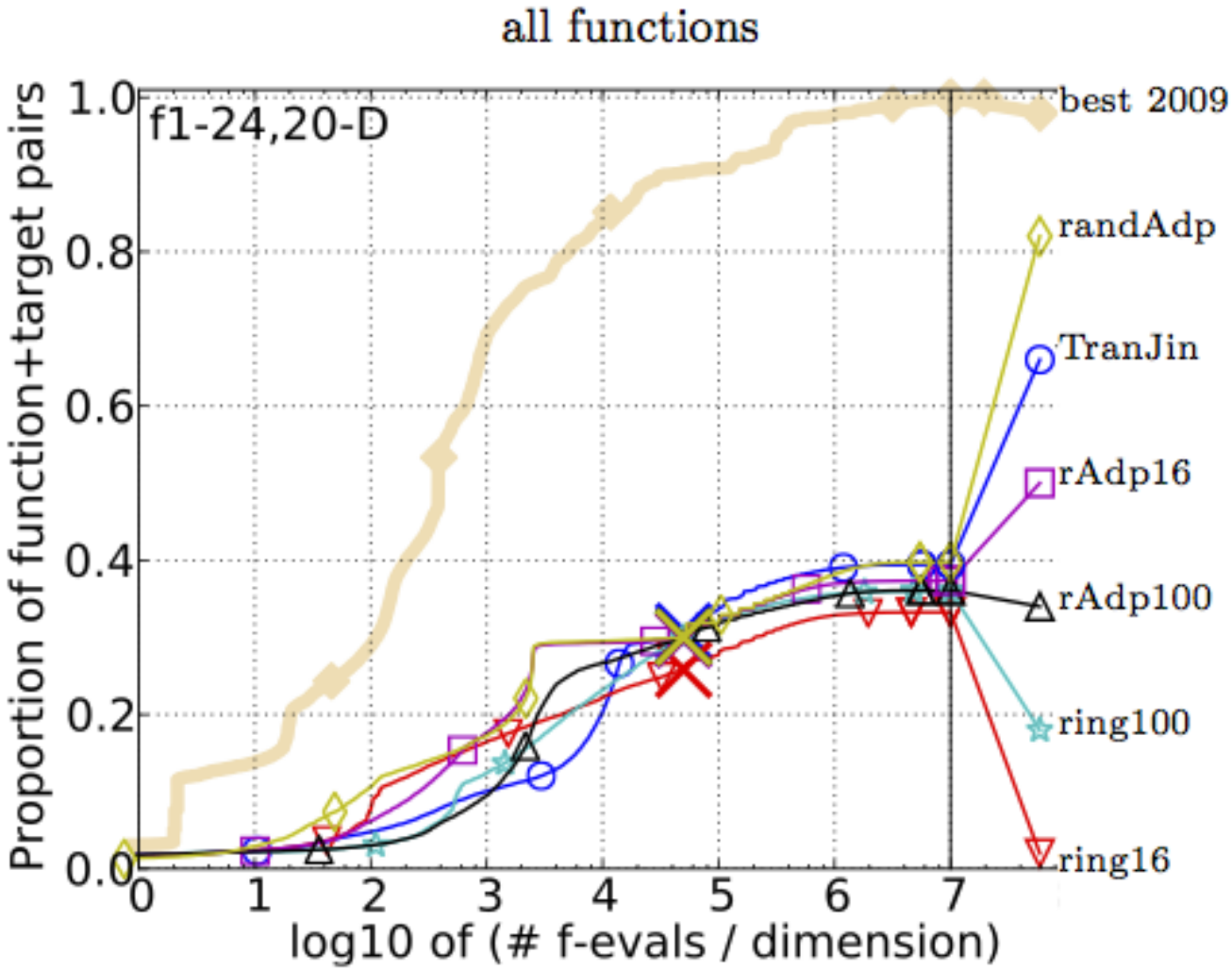


5-D Separable

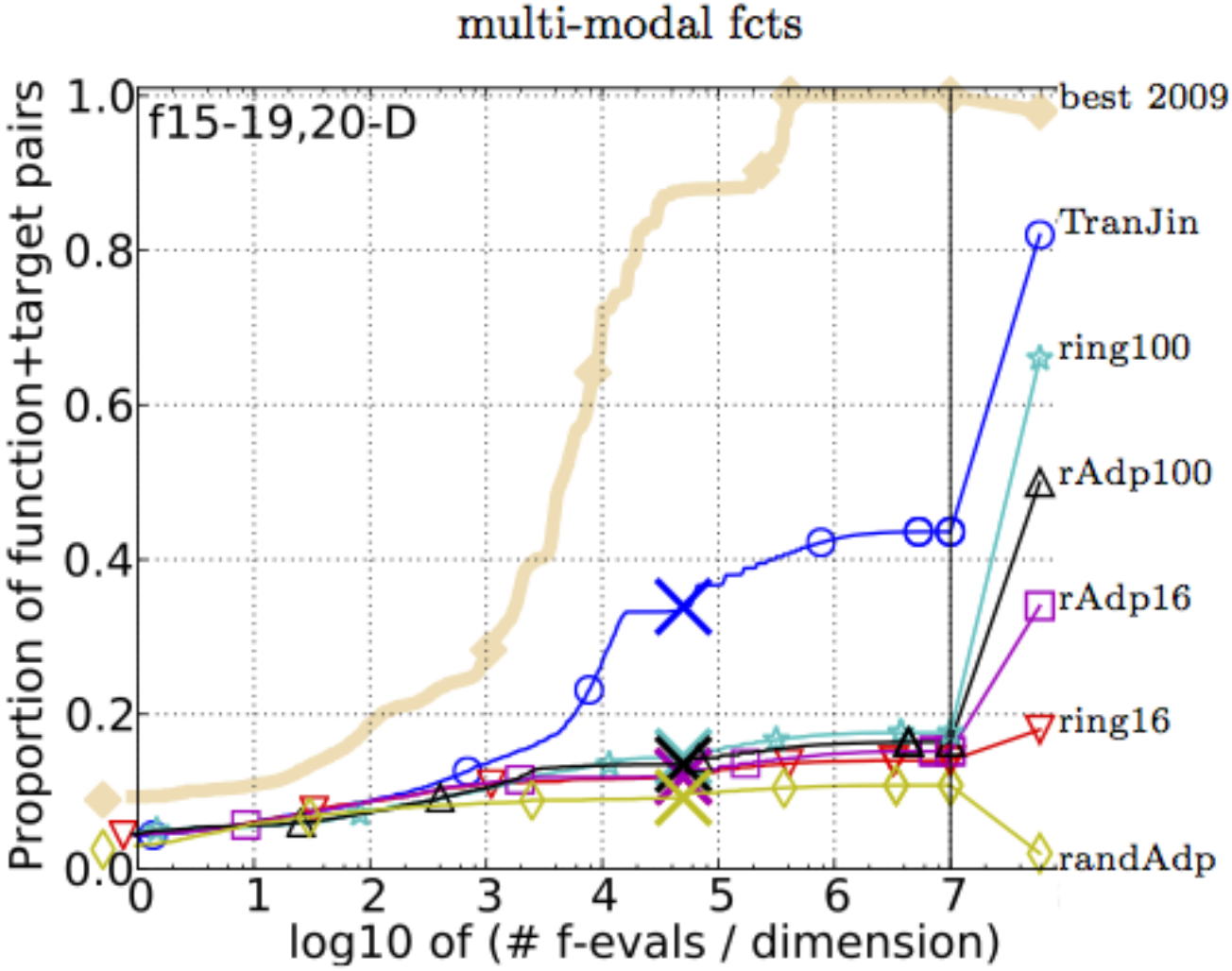


So what happens if we bolster ring CGA with adaptive range mutation?

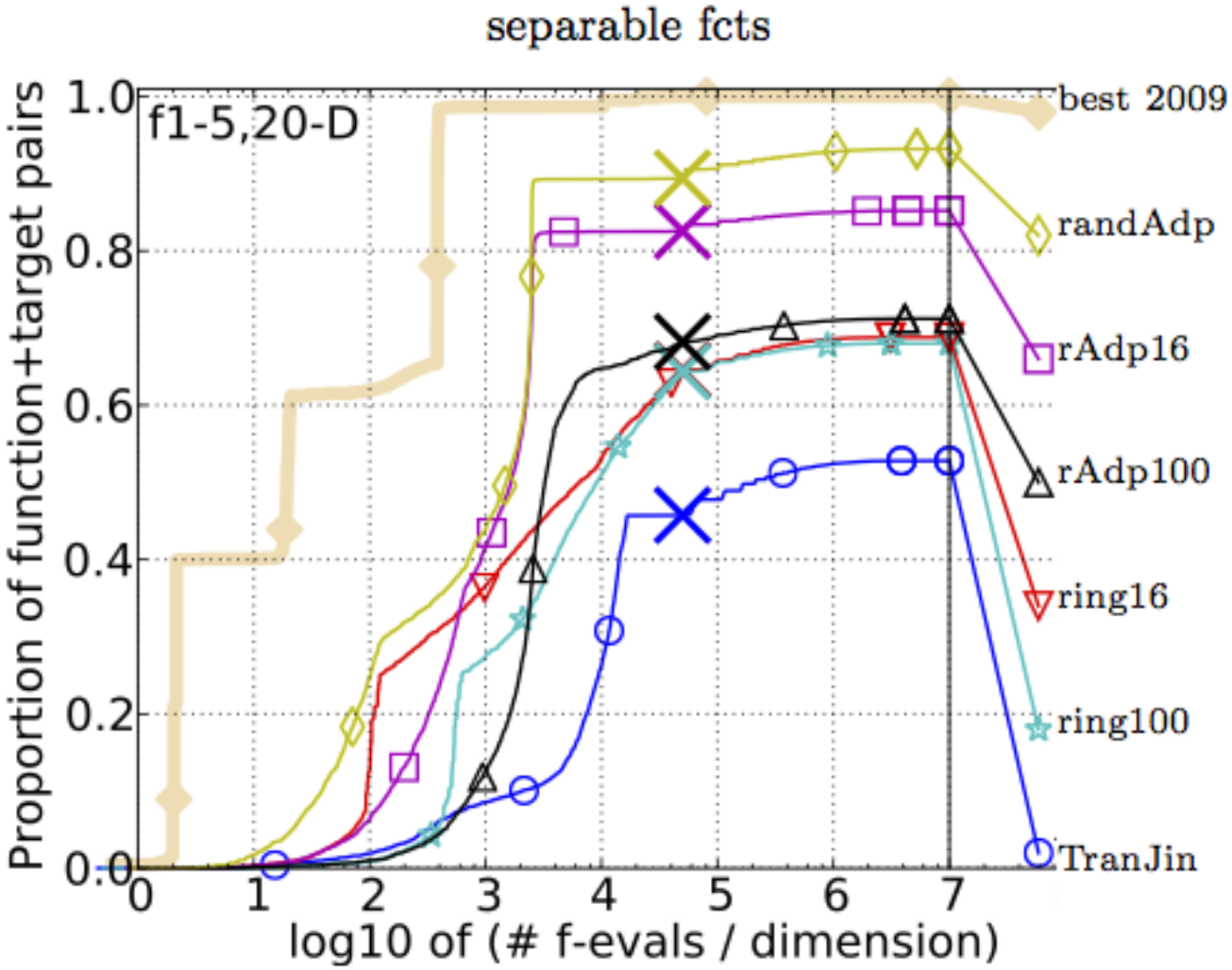
20-D All functions



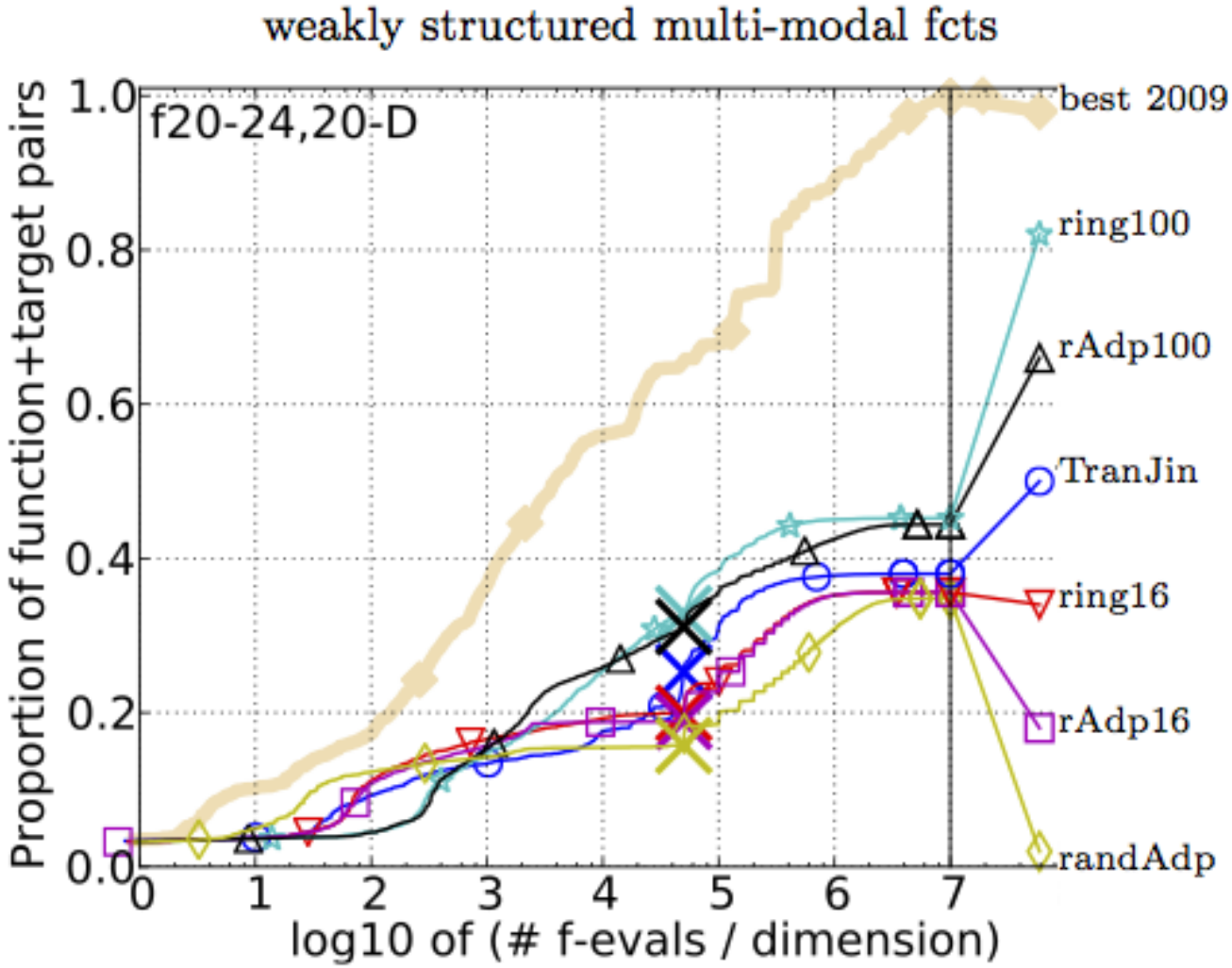
20-D Multi-modal



20-D Separable



20-D Weakly structured



Conclusion

Which algorithm is best?

Conclusion

Which algorithm is best?

It depends on the function to be optimized.

Conclusion

Random search can be highly effective on separable functions.

Conclusion

Single-population, panmictic GAs are competitive with grid CGAs run on a single core.

Conclusion

Adaptive range mutation is an effective mutation operator for closing the distance to the optimal solution.

Arithmetic crossover is not particularly effective on these benchmarks.

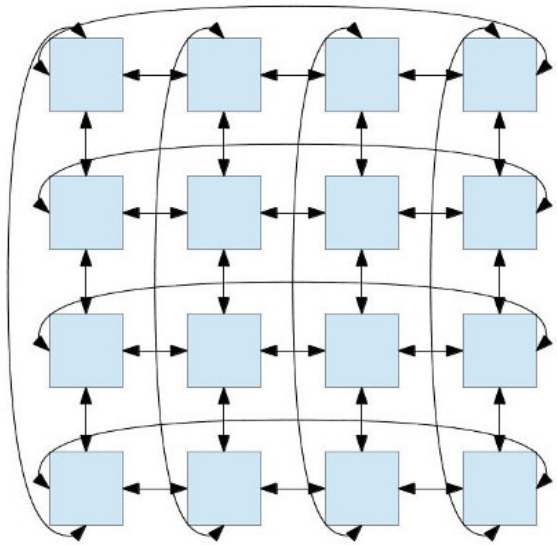
Conclusion

Ring CGA outperforms single-population GAs on weakly-structured, multi-modal functions most likely due to the Ring CGA's greater emphasis on exploration over exploitation.

Conclusion

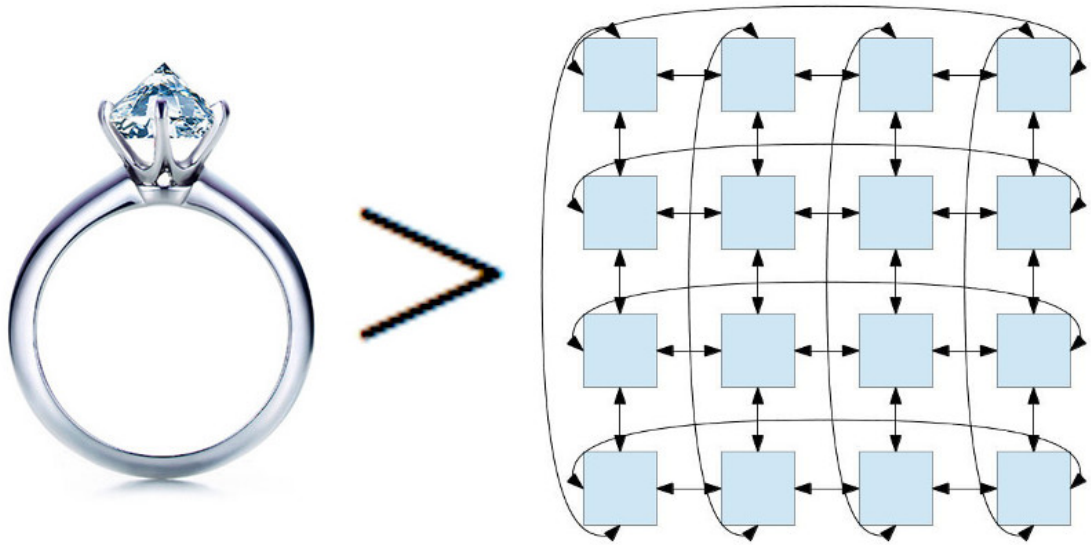
Ring CGAs outperform Grid CGAs.

$>$



Conclusion

Ring CGAs outperform Grid CGAs.



Bonus! Lower communication overhead

Thank you for your time.

Questions?