Performance Tuning of MapReduce Jobs Using Surrogate-Based Modeling

Travis Johnston[1], Mohammad Alsulmi[1], Pietro Cicotti[2], Michela Taufer[1]

[1]University of Delaware
[2]San Diego Supercomputer Center

International Conference on Computational Science
June 1-3, 2015

Question:

How many parameters are there to configure before running a MapReduce job?

Question:

How many parameters are there to configure before running a MapReduce job?

- ☐ 10
- ☐ 20
- ☐ 50
- ☐ 100

Question:

How many parameters are there to configure before running a MapReduce job?

- ☐ 10
- ☐ 20
- ☐ 50
- ☐ 100
- ■ **Hundreds**

Hadoop's configuration file has more than 220 parameters.

A Sampling of Hadoop Configuration Parameters

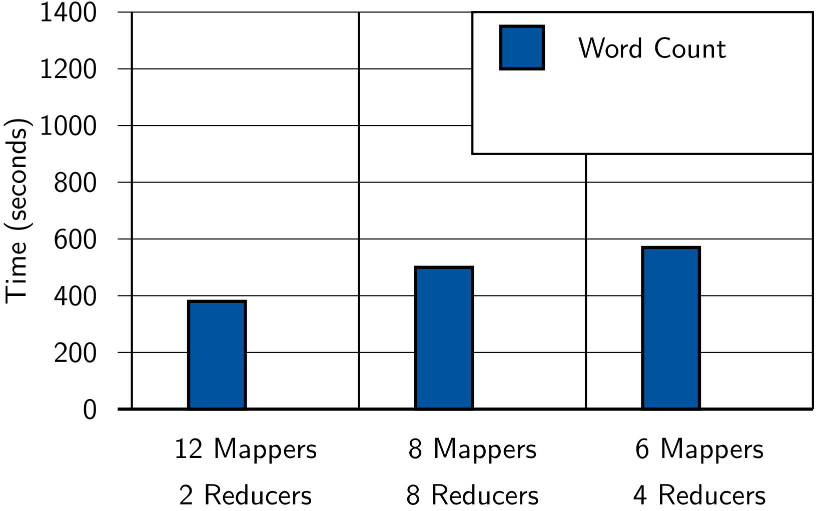
mapreduce.job.maps
mapreduce.job.reduces
mapreduce.map.memory.mb
mapreduce.map.cpu.vcores
mapreduce.reduce.memory.mb
mapreduce.reduce.cpu.vcores
mapreduce.job.userhistorylocation
mapreduce.task.io.sort.factor
mapreduce.task.io.sort.mb
mapreduce.map.sort.spill.percent
mapreduce.jobtracker.address
mapreduce.jobtracker.http.address
mapreduce.jobtracker.handler.count
mapreduce.tasktracker.report.address
mapreduce.cluster.local.dir
mapreduce.jobtracker.system.dir
mapreduce.jobtracker.staging.root.dir
mapreduce.cluster.temp.dir
mapreduce.tasktracker.instrumentation
mapreduce.jobtracker.restart.recover
mapreduce.jobtracker.taskscheduler
mapreduce.job.running.map.limit
mapreduce.job.running.reduce.limit
mapreduce.job.max.split.locations
mapreduce.job.split.metainfo.maxsize
mapreduce.map.maxattempts
mapreduce.reduce.maxattempts
mapreduce.reduce.shuffle.read.timeout
mapreduce.task.timeout
mapreduce.jobtracker.instrumentation
mapred.child.java.opts
mapreduce.map.java.opts

mapreduce.admin.user.env
mapreduce.map.log.level
mapreduce.reduce.log.level
mapreduce.reduce.merge.inmem.threshold
mapreduce.reduce.shuffle.merge.percent
mapreduce.reduce.input.buffer.percent
mapreduce.shuffle.ssl.enabled
mapreduce.shuffle.ssl.file.buffer.size
mapreduce.shuffle.max.connections
mapreduce.shuffle.max.threads
mapreduce.shuffle.transferTo.allowed
mapreduce.shuffle.transfer.buffer.size
mapreduce.map.speculative
mapreduce.reduce.speculative
mapreduce.job.jvm.numtasks
mapreduce.job.ubertask.enable
mapreduce.job.ubertask.maxmaps
mapreduce.job.ubertask.maxreduces
mapreduce.job.ubertask.maxbytes
mapreduce.job.emit-timeline-data
mapreduce.jobtracker.maxtasks.perjob
mapreduce.tasktracker.dns.interface
mapreduce.tasktracker.dns.nameserver
mapreduce.tasktracker.http.threads
mapreduce.tasktracker.http.address
mapreduce.map.output.compress
mapreduce.map.output.compress.codec
map.sort.class
mapreduce.task.userlog.limit.kb
yarn.app.mapreduce.shuffle.log.backups
mapreduce.job.userlog.retain.hours
mapreduce.jobtracker.hosts.filename

mapreduce.task.profile.params
mapreduce.task.profile.map.params
mapreduce.task.profile.reduce.params
mapreduce.task.skip.start.attempts
mapreduce.map.skip.proc.count.autoincr
mapreduce.job.skip.outdir
mapreduce.map.skip.maxrecords
mapreduce.reduce.skip.maxgroups
mapreduce.ifile.readahead
mapreduce.ifile.readahead.bytes
mapreduce.jobtracker.taskcache.levels
mapreduce.job.queuename
mapreduce.job.tags
mapreduce.cluster.acls.enabled
mapreduce.job.acl-modify-job
mapreduce.job.acl-view-job
mapreduce.tasktracker.indexcache.mb
mapreduce.job.token.tracking.ids
mapreduce.task.merge.progress.records
mapreduce.tasktracker.taskcontroller
mapreduce.tasktracker.group
mapreduce.shuffle.port
mapreduce.job.counters.limit
mapreduce.framework.name
yarn.app.mapreduce.am.staging-dir
mapreduce.am.max-attempts
mapreduce.job.end-notification.url
mapreduce.job.log4j-properties-file
yarn.app.mapreduce.am.env
yarn.app.mapreduce.am.admin.user.env
yarn.app.mapreduce.am.command-opts
yarn.app.mapreduce.client.max-retries

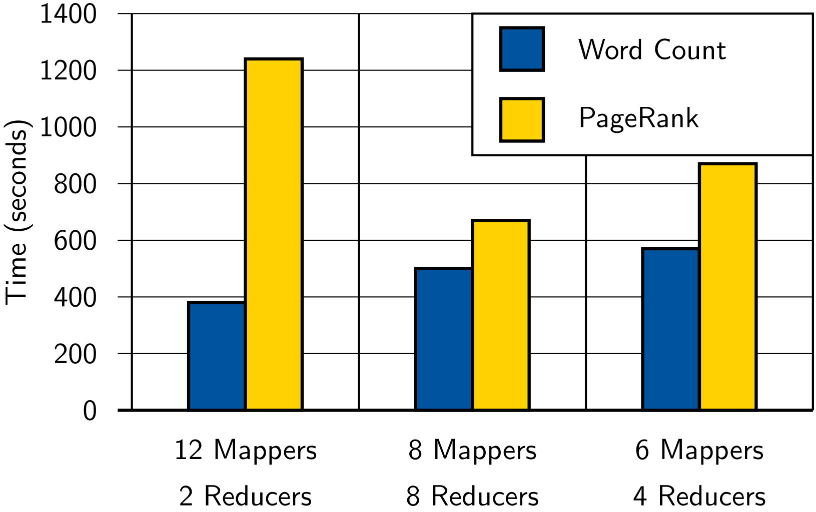
How Much Does the Configuration Matter?

Two benchmarks: Word Count and PageRank



How Much Does the Configuration Matter?

Two benchmarks: Word Count and PageRank



Question:

How do we select the best configuration?

Question:

How do we select the best configuration?

Challenges:

- Every framework has 100's of parameters that affect performance.

Question:

How do we select the best configuration?

Challenges:

- Every framework has 100's of parameters that affect performance.
- Sampling sets of parameters quickly becomes expensive.

Question:

How do we select the best configuration?

Challenges:

- Every framework has 100's of parameters that affect performance.
- Sampling sets of parameters quickly becomes expensive.

Opportunities:

- We can learn from frequently run jobs at production level.
  - PageRank, (Google) run frequently to keep website ranking up-to-date.
  - Word Count, (Facebook/Twitter) run frequently to see *what's trending*.

Question:

How do we select the best configuration?

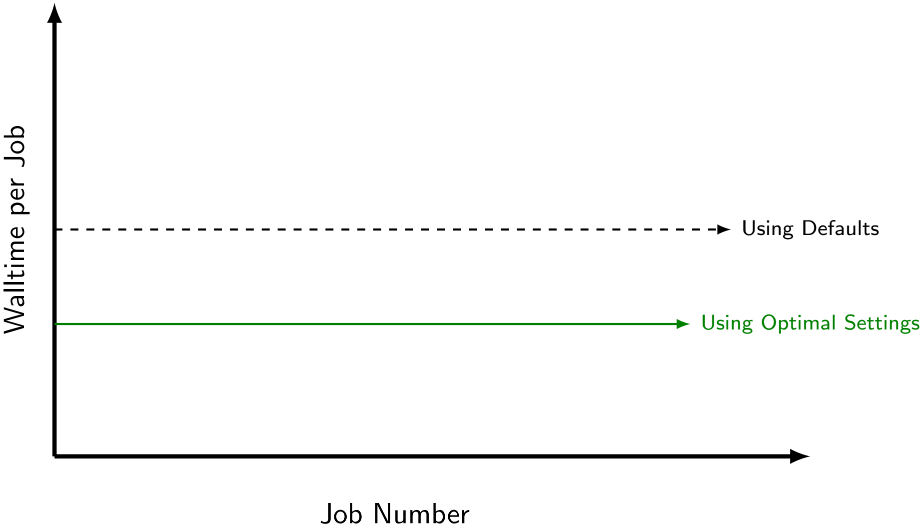
Challenges:

- Every framework has 100's of parameters that affect performance.
- Sampling sets of parameters quickly becomes expensive.

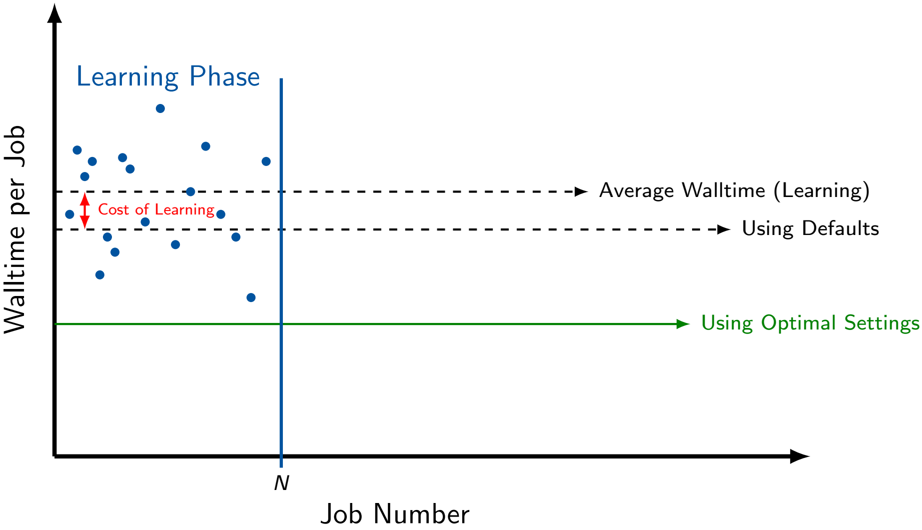
Opportunities:

- We can learn from frequently run jobs at production level.
  - PageRank, (Google) run frequently to keep website ranking up-to-date.
  - Word Count, (Facebook/Twitter) run frequently to see *what's trending*.
- Amortize the cost of sampling with future gains in performance.

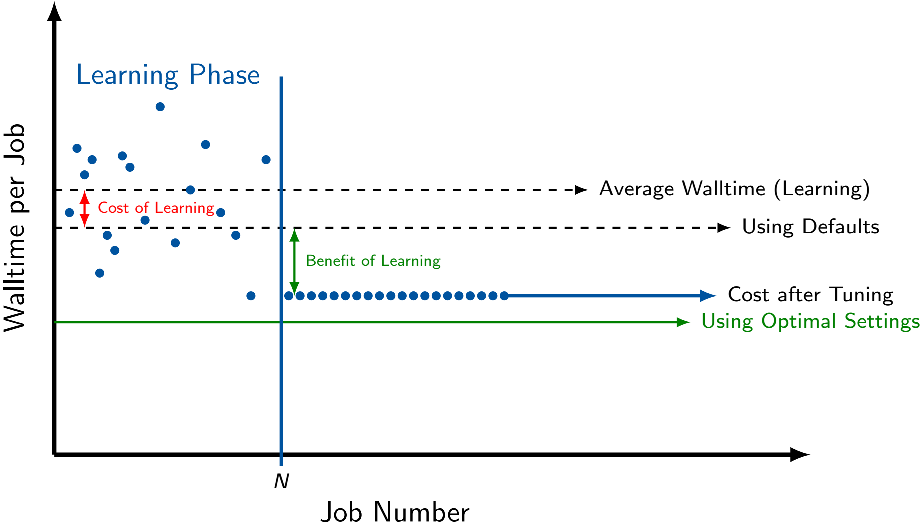
Amortizing the Cost of Sampling



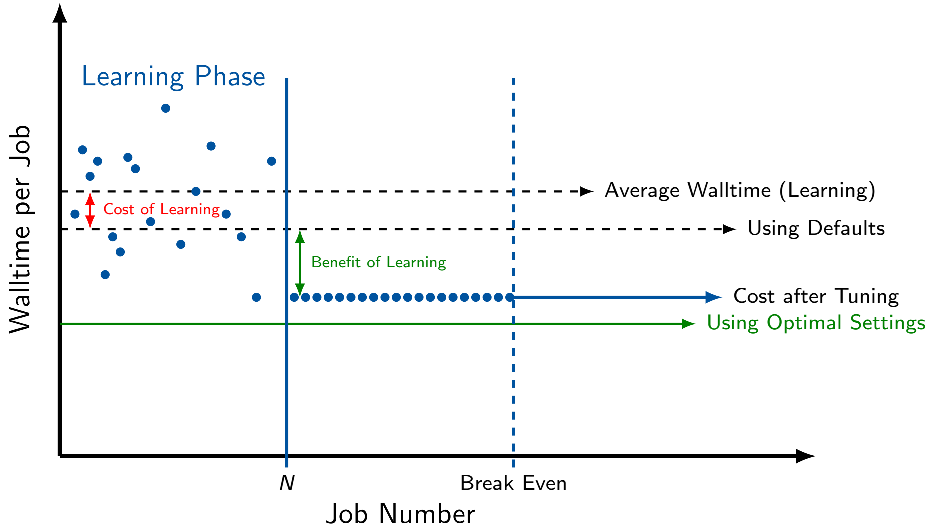
Amortizing the Cost of Sampling



Amortizing the Cost of Sampling



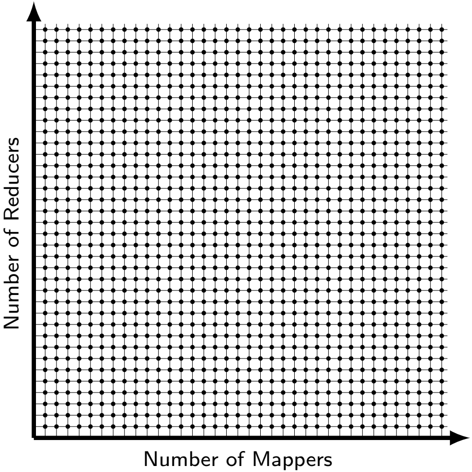
Amortizing the Cost of Sampling



Question:

How do we focus the learning phase to **maximize** the return on our investment?

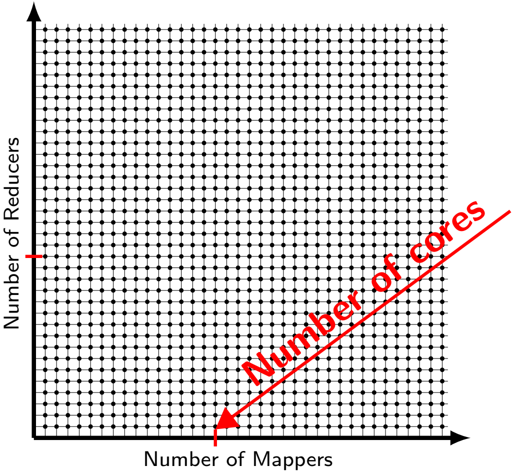
Exhaustive Searching [1]



Step 1:

Reduce to finite search space

[1] P. K. Lakkimsetti, *A Framework for Automatic Optimization of MapReduce Programs Based on Job Parameter Configurations*, M. Thesis, Kansas State University, 2011.

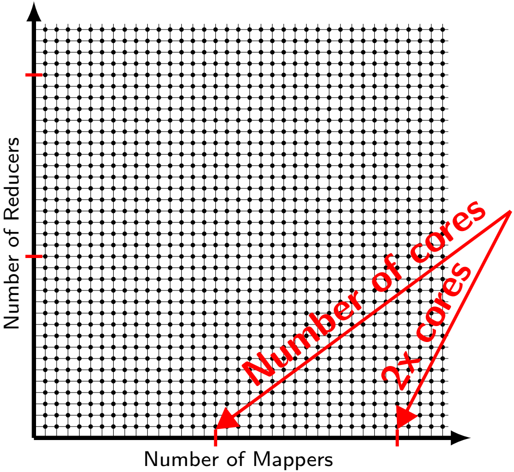


Step 1:

Reduce to finite search space

[1] P. K. Lakkimsetti, *A Framework for Automatic Optimization of MapReduce Programs Based on Job Parameter Configurations*, M. Thesis, Kansas State University, 2011.

Exhaustive Searching [1]

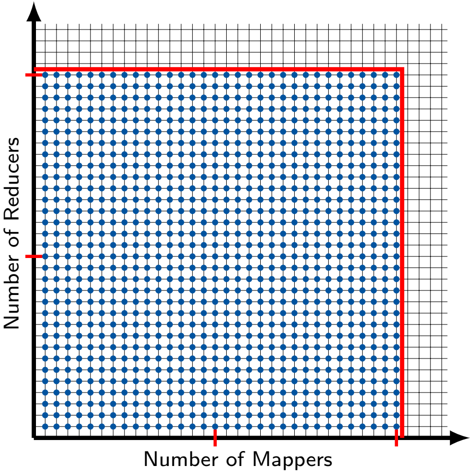


Step 1:

Reduce to finite search space

[1] P. K. Lakkimsetti, *A Framework for Automatic Optimization of MapReduce Programs Based on Job Parameter Configurations*, M. Thesis, Kansas State University, 2011.

Exhaustive Searching [1]

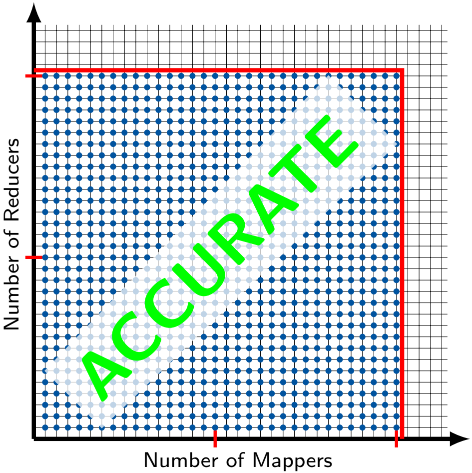


Step 2:

Sample all these points

[1] P. K. Lakkimsetti, *A Framework for Automatic Optimization of MapReduce Programs Based on Job Parameter Configurations*, M. Thesis, Kansas State University, 2011.

Exhaustive Searching [1]

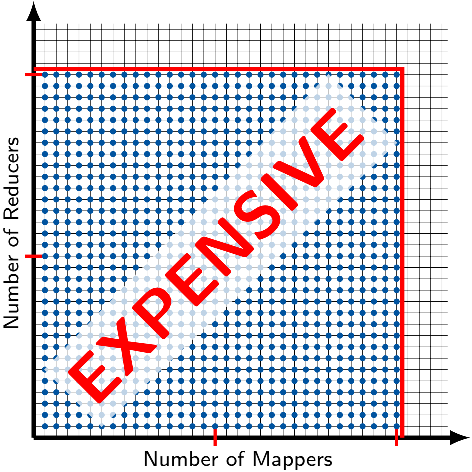


Step 2:

Sample all these points

[1] P. K. Lakkimsetti, *A Framework for Automatic Optimization of MapReduce Programs Based on Job Parameter Configurations*, M. Thesis, Kansas State University, 2011.

Exhaustive Searching [1]



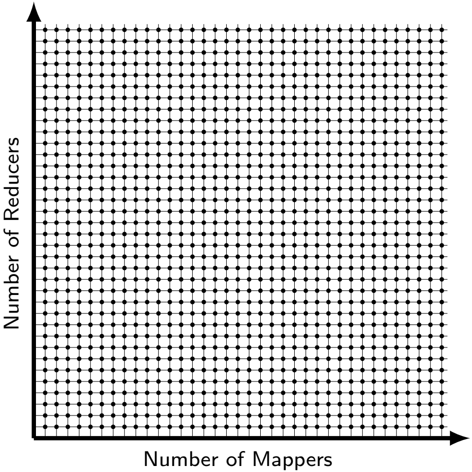
Step 2:

Sample all these points

[1] P. K. Lakkimsetti, *A Framework for Automatic Optimization of MapReduce Programs Based on Job Parameter Configurations*, M. Thesis, Kansas State University, 2011.

Local Searching Algorithms

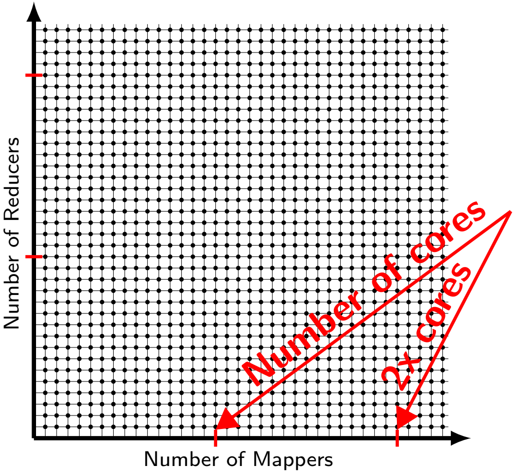
Grid Hill [2] and Simulated Annealing [3]



Step 1:

Reduce to finite search space

[2] K. Wang, X. Lin, W. Tang, *Predator-An Experience Guided Configuration Optimizer...*, IEEE CloudCom 2012.
[3] D. Wu, *A Profiling and Performance Analysis Based Self-tuning System for Optimization...*, M. Thesis, Vanderbilt, 2013.



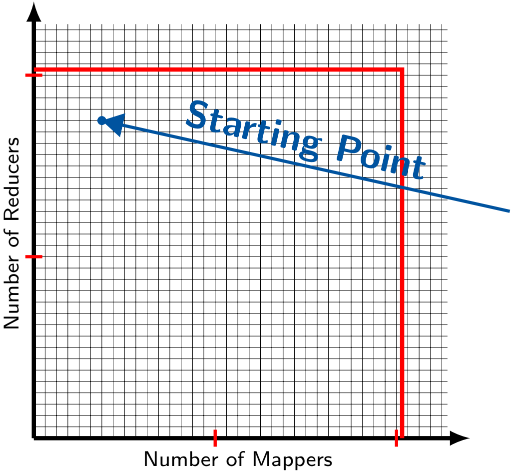
Step 1:

Reduce to finite search space

[2] K. Wang, X. Lin, W. Tang, *Predator-An Experience Guided Configuration Optimizer...*, IEEE CloudCom 2012.
[3] D. Wu, *A Profiling and Performance Analysis Based Self-tuning System for Optimization...*, M. Thesis, Vanderbilt, 2013.

Local Searching Algorithms

Grid Hill [2] and Simulated Annealing [3]



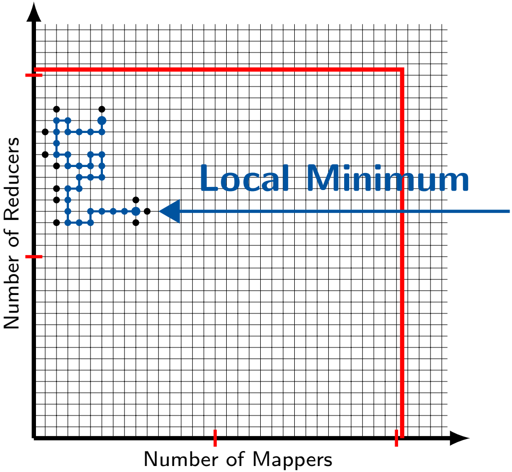
Step 2:

Randomly choose starting point and sample neighbors

[2] K. Wang, X. Lin, W. Tang, *Predator-An Experience Guided Configuration Optimizer...*, IEEE CloudCom 2012.
[3] D. Wu, *A Profiling and Performance Analysis Based Self-tuning System for Optimization...*, M. Thesis, Vanderbilt, 2013.

Local Searching Algorithms

Grid Hill [2] and Simulated Annealing [3]



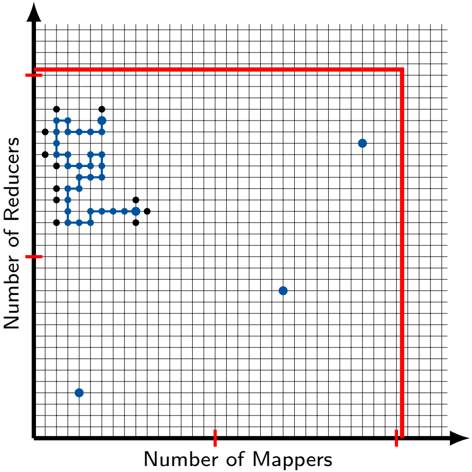
Step 2:

Randomly choose starting point and sample neighbors

[2] K. Wang, X. Lin, W. Tang, *Predator-An Experience Guided Configuration Optimizer...*, IEEE CloudCom 2012.
[3] D. Wu, *A Profiling and Performance Analysis Based Self-tuning System for Optimization...*, M. Thesis, Vanderbilt, 2013.

Local Searching Algorithms

Grid Hill [2] and Simulated Annealing [3]



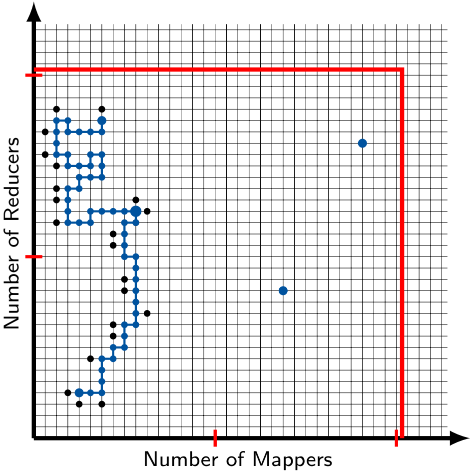
Step 3:

Repeat **Step 2** with new starting point(s)

[2] K. Wang, X. Lin, W. Tang, *Predator-An Experience Guided Configuration Optimizer...*, IEEE CloudCom 2012.
[3] D. Wu, *A Profiling and Performance Analysis Based Self-tuning System for Optimization...*, M. Thesis, Vanderbilt, 2013.

Local Searching Algorithms

Grid Hill [2] and Simulated Annealing [3]



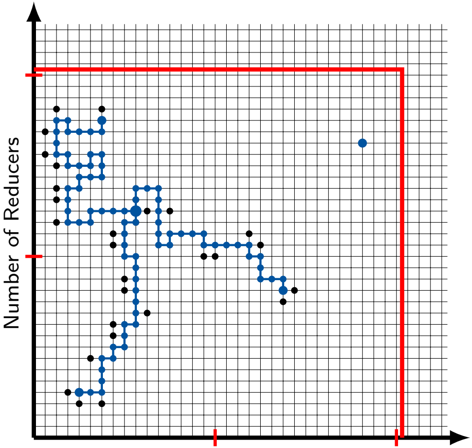
Step 3:

Repeat **Step 2**
with new starting point(s)

[2] K. Wang, X. Lin, W. Tang, *Predator-An Experience Guided Configuration Optimizer...*, IEEE CloudCom 2012.
[3] D. Wu, *A Profiling and Performance Analysis Based Self-tuning System for Optimization...*, M. Thesis, Vanderbilt, 2013.

Local Searching Algorithms

Grid Hill [2] and Simulated Annealing [3]



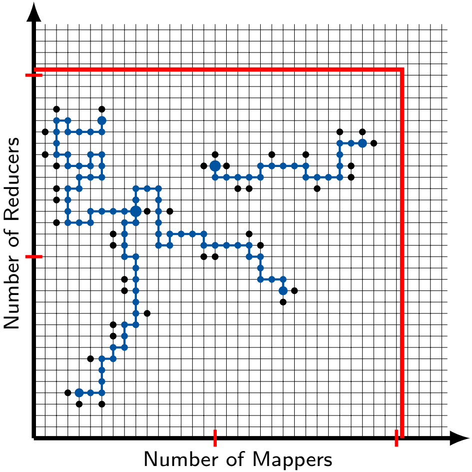
Step 3:

Repeat **Step 2**
with new starting point(s)

[2] K. Wang, X. Lin, W. Tang, *Predator-An Experience Guided Configuration Optimizer...*, IEEE CloudCom 2012.
[3] D. Wu, *A Profiling and Performance Analysis Based Self-tuning System for Optimization...*, M. Thesis, Vanderbilt, 2013.

Local Searching Algorithms

Grid Hill [2] and Simulated Annealing [3]



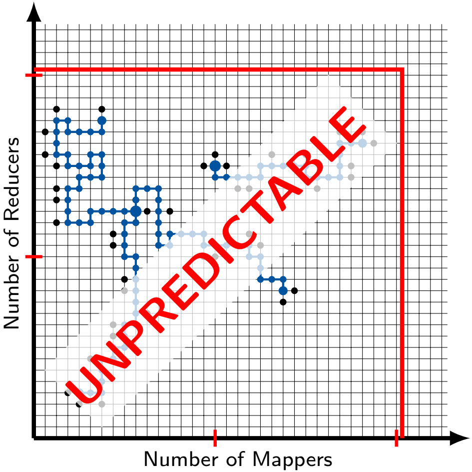
Step 3:

Repeat **Step 2**
with new starting point(s)

[2] K. Wang, X. Lin, W. Tang, *Predator-An Experience Guided Configuration Optimizer...*, IEEE CloudCom 2012.
[3] D. Wu, *A Profiling and Performance Analysis Based Self-tuning System for Optimization...*, M. Thesis, Vanderbilt, 2013.

Local Searching Algorithms

Grid Hill [2] and Simulated Annealing [3]



Step 3:

Repeat **Step 2**
with new starting point(s)

[2] K. Wang, X. Lin, W. Tang, *Predator-An Experience Guided Configuration Optimizer...*, IEEE CloudCom 2012.
[3] D. Wu, *A Profiling and Performance Analysis Based Self-tuning System for Optimization...*, M. Thesis, Vanderbilt, 2013.

Overcoming Pitfalls Using Surrogate-Based Modeling

- Exhaustive sampling is too expensive.
- Local search algorithms (LSAs) sample fewer points but are unpredictable.

Overcoming Pitfalls Using Surrogate-Based Modeling

- Exhaustive sampling is too expensive.
- Local search algorithms (LSAs) sample fewer points but are unpredictable.
- Surrogate-based modeling can achieve near optimal results sampling 67% fewer points than LSAs!

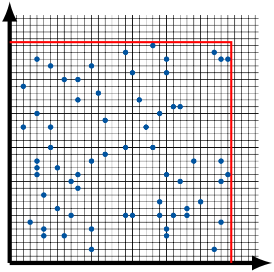
Overcoming Pitfalls Using Surrogate-Based Modeling

- Exhaustive sampling is too expensive.
- Local search algorithms (LSAs) sample fewer points but are unpredictable.
- Surrogate-based modeling can achieve near optimal results sampling 67% fewer points than LSAs!
- We can explicitly determine the number of points required to build a surrogate model!

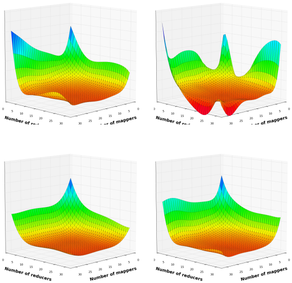
Overcoming Pitfalls Using Surrogate-Based Modeling

- Exhaustive sampling is too expensive.
- Local search algorithms (LSAs) sample fewer points but are unpredictable.
- Surrogate-based modeling can achieve near optimal results sampling 67% fewer points than LSAs!
- We can explicitly determine the number of points required to build a surrogate model!
- The surrogate model may predict optimal configurations that were **never sampled** in the learning phase!

Overview of Surrogate-Based Modeling

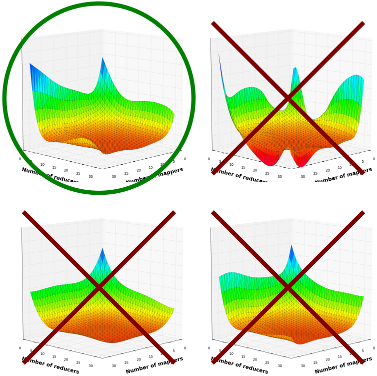


Step 1: Sample parameter space

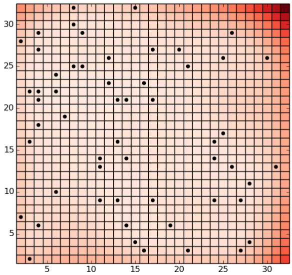


Step 2: Build candidate surfaces

Overview of Surrogate-Based Modeling



Step 3: Select the best surface



Step 4: Apply confidence interval

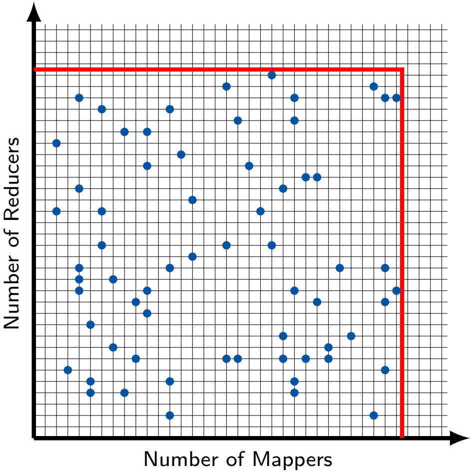
Case Study: Optimizing PageRank

Experimental Setup

- Use Hadoop and PageRank from the HiBench benchmarking suite
- Select two parameters to tune:
  - the number of mappers, x, and
  - the number of reducers, y.
- Input file is 1GB in size consisting of 5M inter-linked web pages.
- Computations are done on a single large memory node of Stampede with
  - 32 CPU cores, and
  - 1 TB memory.
- To enable validation of our method, we initially sample all points $2 \leq x, y \leq 32$: 961 total points.

Surrogate-Based Modeling

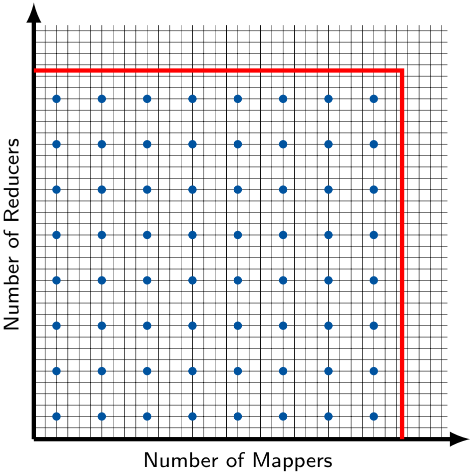
Step 1: Sampling the parameter space ($N = 64$ samples)



Random Sampling

Surrogate-Based Modeling

Step 1: Sampling the parameter space ($N = 64$ samples)



Grid-based Sampling

Surrogate-Based Modeling

Step 2: Building candidate surfaces

What kind of surfaces do we build?

We represent our surface by a multivariate polynomial.

The candidate surfaces become:

- $z_1(x, y) = \beta_1 + \beta_2 x + \beta_3 y$ Degree 1
- $z_2(x, y) = \beta_1 + \beta_2 x + \beta_3 y + \beta_4 x^2 + \beta_5 xy + \beta_6 y^2$ Degree 2
- $z_3(x, y) = z_2(x, y) + \beta_7 x^3 + \beta_8 x^2 y + \beta_9 xy^2 + \beta_{10} y^3$ Degree 3
- And so forth ...

Surrogate-Based Modeling

Step 2: Building candidate surfaces

What kind of surfaces do we build?

We represent our surface by a multivariate polynomial.

The candidate surfaces become:

- $z_1(x, y) = \beta_1 + \beta_2 x + \beta_3 y$ Degree 1
- $z_2(x, y) = \beta_1 + \beta_2 x + \beta_3 y + \beta_4 x^2 + \beta_5 xy + \beta_6 y^2$ Degree 2
- $z_3(x, y) = z_2(x, y) + \beta_7 x^3 + \beta_8 x^2 y + \beta_9 xy^2 + \beta_{10} y^3$ Degree 3
- And so forth ...

Why build a polynomial surface?

Surrogate-Based Modeling

Step 2: Building candidate surfaces

What kind of surfaces do we build?

We represent our surface by a multivariate polynomial.

The candidate surfaces become:

- $z_1(x, y) = \beta_1 + \beta_2 x + \beta_3 y$ Degree 1
- $z_2(x, y) = \beta_1 + \beta_2 x + \beta_3 y + \beta_4 x^2 + \beta_5 xy + \beta_6 y^2$ Degree 2
- $z_3(x, y) = z_2(x, y) + \beta_7 x^3 + \beta_8 x^2 y + \beta_9 xy^2 + \beta_{10} y^3$ Degree 3
- And so forth ...

Why build a polynomial surface?

- Polynomials are easy to describe and represent in memory.
- Although the description is simple, polynomials can generate quite complex surfaces.
- Polynomials easily generalize to any number of variables.

How many points do we need to sample?

How many points do we need to sample?

Theoretical Minimum Number of Points

- To build the surface, we solve the matrix equation:

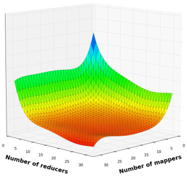
$$X^T X\beta = X^T Z$$

  to determine the coefficients β.
- If $X^T X$ is not invertible, then there is not a unique solution for β.
- If the number of samples taken is smaller than the number of terms in our polynomial, then $X^T X$ is not invertible.
- To build a surface of degree d with v variables we need at least $\binom{d+v}{v}$ samples.

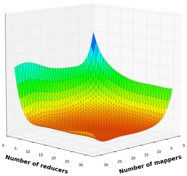
Surrogate-Based Modeling

Step 2: Building candidate surfaces

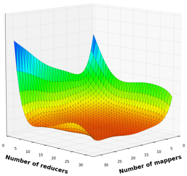
Degree 5



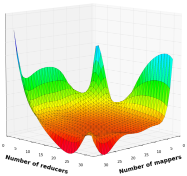
Degree 6

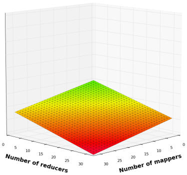


Degree 7

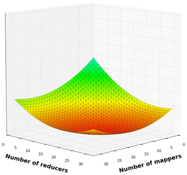


Degree 8

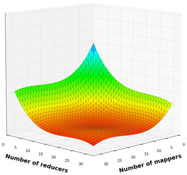




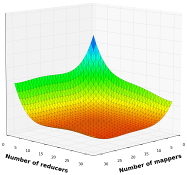
Degree 1



Degree 2



Degree 3

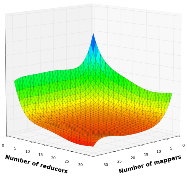


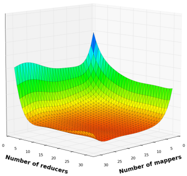
Degree 4

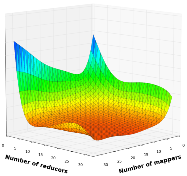
Surrogate-Based Modeling

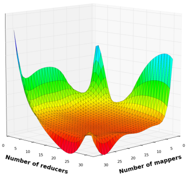
Step 2: Building candidate surfaces

Degree 5 | Degree 6 | Degree 7 | Degree 8

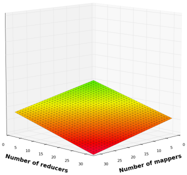


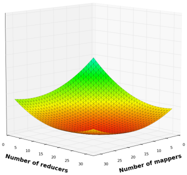


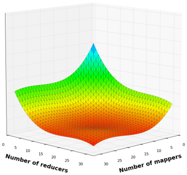


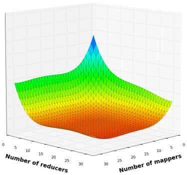


Which Model is Best?







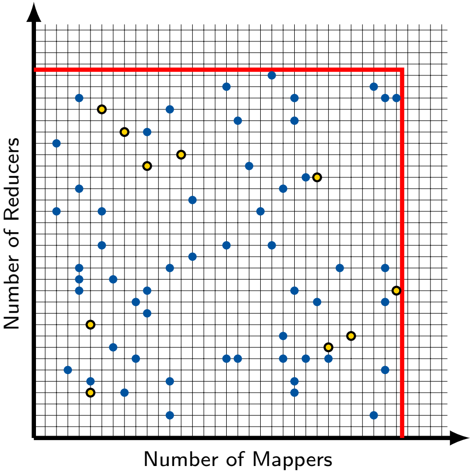


Degree 1 | Degree 2 | Degree 3 | Degree 4

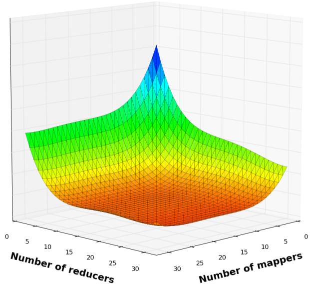
Step 3: Selecting the best surface using k-fold cross validation

Partition the samples into k sets of (nearly) equal size.

One set is reserved for testing; $k-1$ sets are used for learning.



- **Learning Sets**
- **Testing Set**



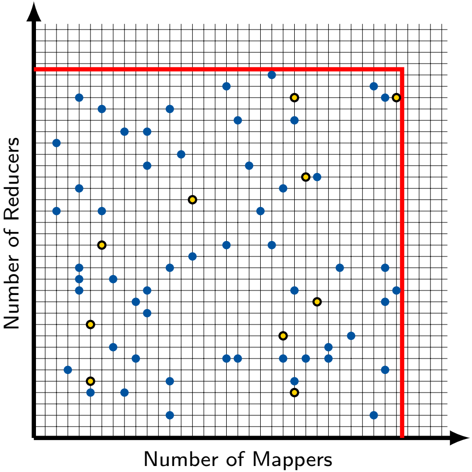
Degree 4 Surface

Surrogate-Based Modeling

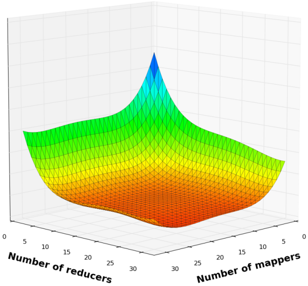
Step 3: Selecting the best surface using k-fold cross validation

Partition the samples into k sets of (nearly) equal size.

One set is reserved for testing; $k-1$ sets are used for learning.



- **Learning Sets**
- **Testing Set**



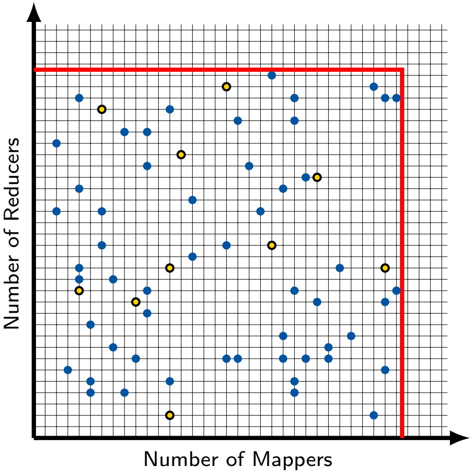
Degree 4 Surface

Surrogate-Based Modeling

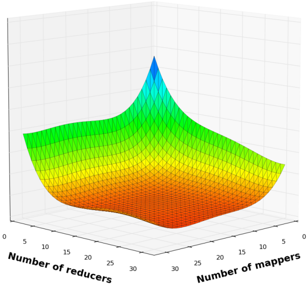
Step 3: Selecting the best surface using k-fold cross validation

Partition the samples into k sets of (nearly) equal size.

One set is reserved for testing; $k-1$ sets are used for learning.



- **Learning Sets**
- **Testing Set**



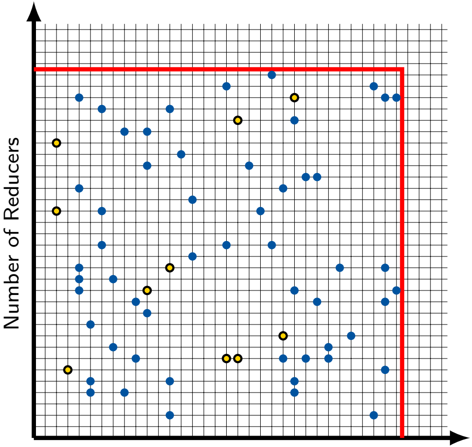
Degree 4 Surface

Surrogate-Based Modeling

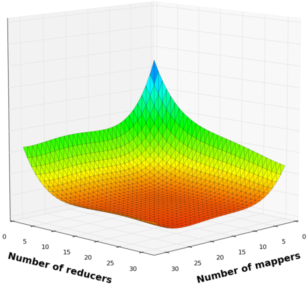
Step 3: Selecting the best surface using k-fold cross validation

Partition the samples into k sets of (nearly) equal size.

One set is reserved for testing; $k-1$ sets are used for learning.



- **Learning Sets**
- **Testing Set**



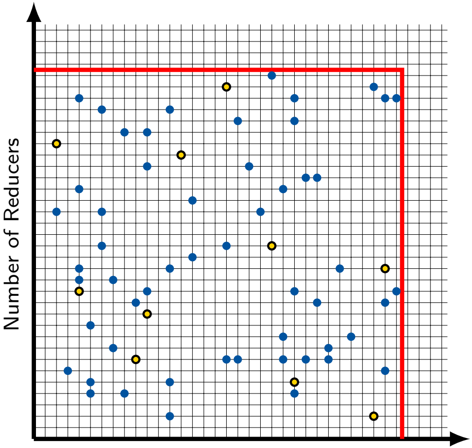
Degree 4 Surface

Surrogate-Based Modeling

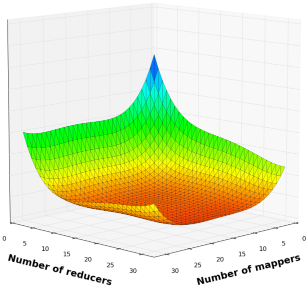
Step 3: Selecting the best surface using k-fold cross validation

Partition the samples into k sets of (nearly) equal size.

One set is reserved for testing; $k-1$ sets are used for learning.



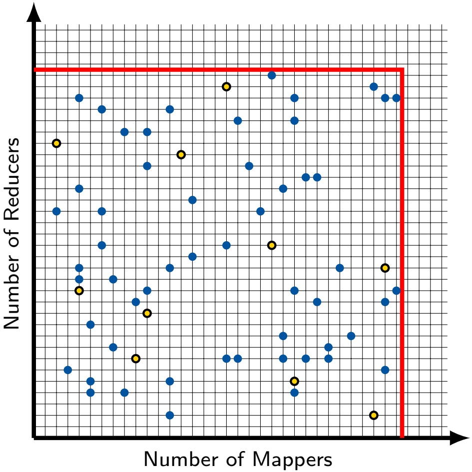
- **Learning Sets**
- **Testing Set**



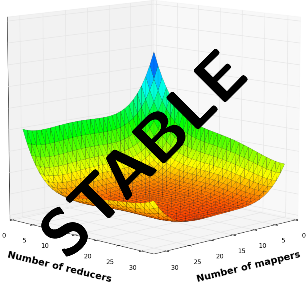
Degree 4 Surface

Partition the samples into k sets of (nearly) equal size.

One set is reserved for testing; $k - 1$ sets are used for learning.



- **Learning Sets**
- **Testing Set**



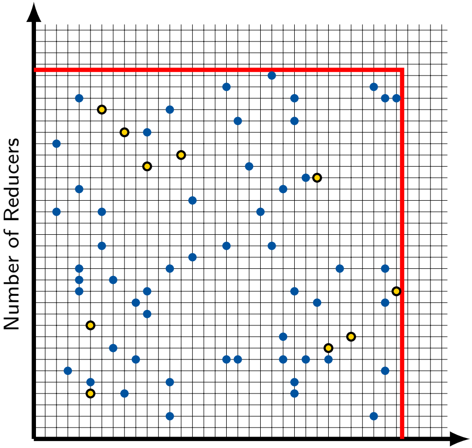
Degree 4 Surface

Surrogate-Based Modeling

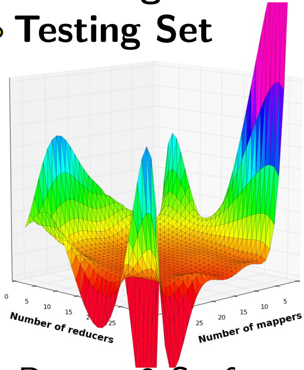
Step 3: Selecting the best surface using k-fold cross validation

Partition the samples into k sets of (nearly) equal size.

One set is reserved for testing; $k-1$ sets are used for learning.



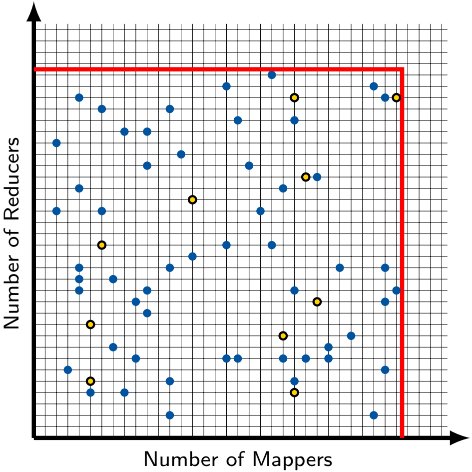
- **Learning Sets**
- **Testing Set**



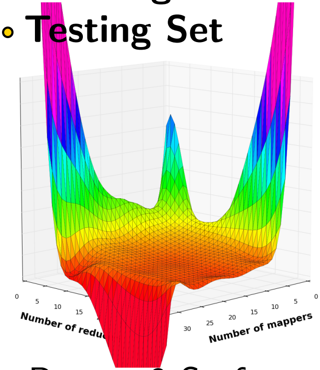
Degree 8 Surface

Partition the samples into k sets of (nearly) equal size.

One set is reserved for testing; $k-1$ sets are used for learning.



- **Learning Sets**
- **Testing Set**



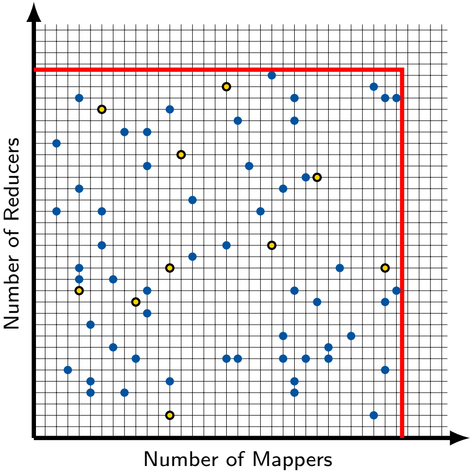
Degree 8 Surface

Surrogate-Based Modeling

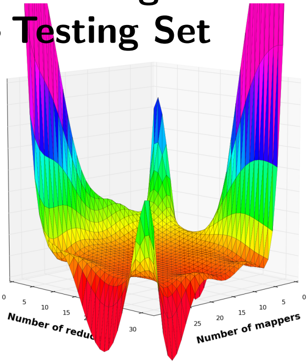
Step 3: Selecting the best surface using k-fold cross validation

Partition the samples into k sets of (nearly) equal size.

One set is reserved for testing; $k - 1$ sets are used for learning.



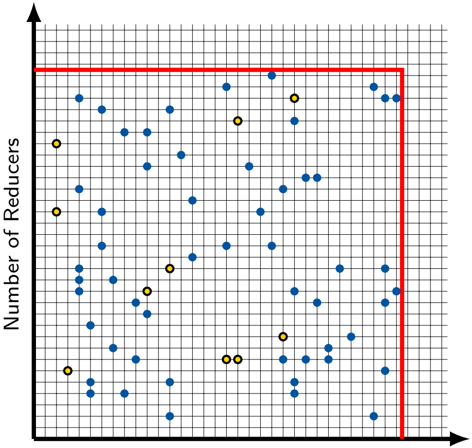
- **Learning Sets**
- **Testing Set**



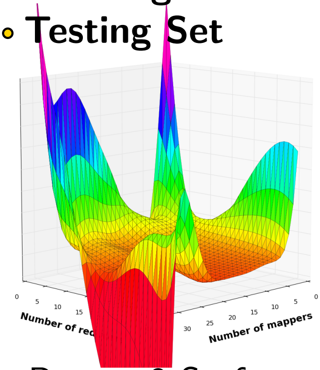
Degree 8 Surface

Partition the samples into k sets of (nearly) equal size.

One set is reserved for testing; $k-1$ sets are used for learning.



- **Learning Sets**
- **Testing Set**



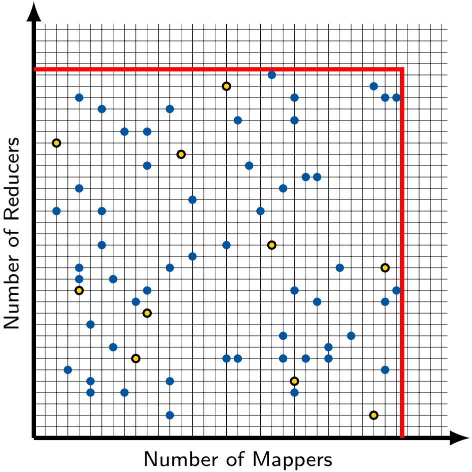
Degree 8 Surface

Surrogate-Based Modeling

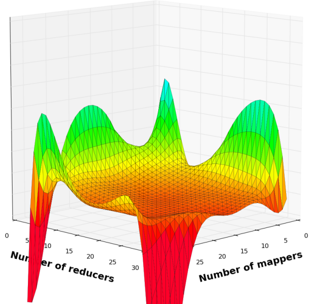
Step 3: Selecting the best surface using *k*-fold cross validation

Partition the samples into k sets of (nearly) equal size.

One set is reserved for testing; $k-1$ sets are used for learning.



- **Learning Sets**
- **Testing Set**



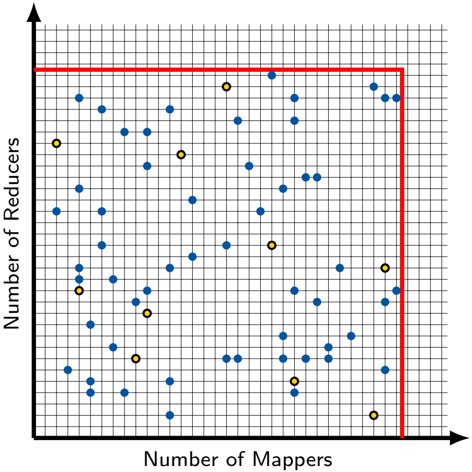
Degree 8 Surface

Surrogate-Based Modeling

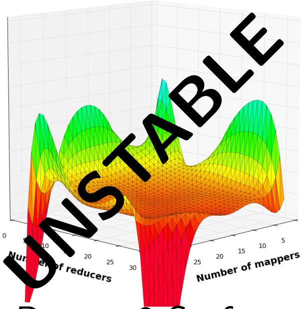
Step 3: Selecting the best surface using *k*-fold cross validation

Partition the samples into k sets of (nearly) equal size.

One set is reserved for testing; $k-1$ sets are used for learning.

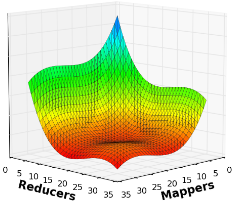


- **Learning Sets**
- **Testing Set**

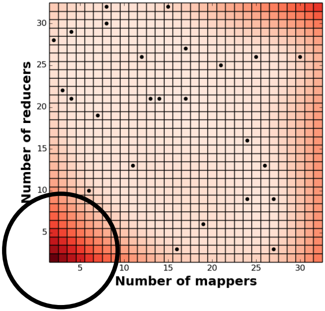


Degree 8 Surface

The Model



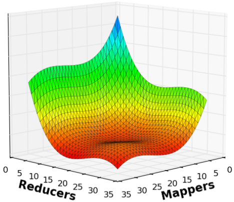
Width of Confidence Interval



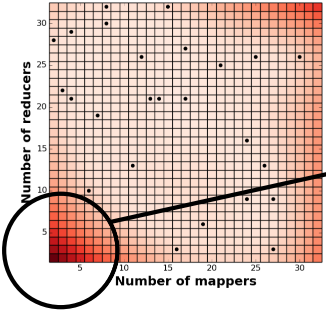
Surrogate-Based Modeling

Step 4: Applying a confidence interval

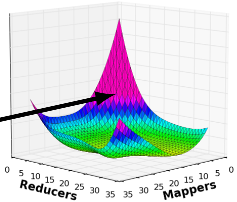
The Model



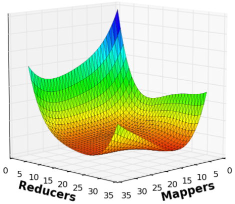
Width of Confidence Interval



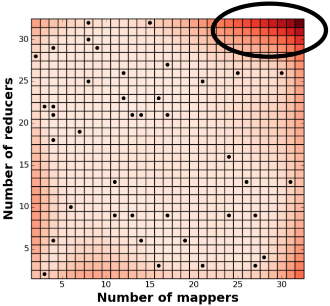
Model + Confidence Interval



The Model



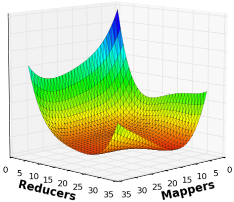
Width of Confidence Interval



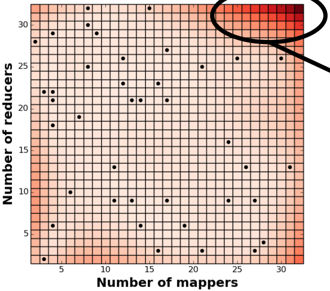
Surrogate-Based Modeling

Step 4: Applying a confidence interval

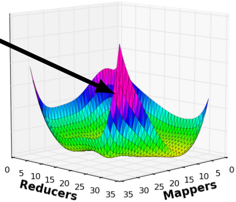
The Model



Width of Confidence Interval



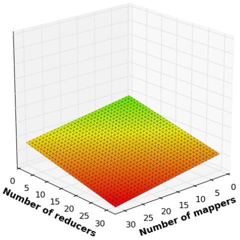
Model + Confidence Interval



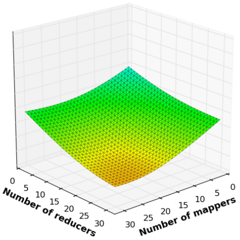
Case Study: Optimizing PageRank

Random sampling: 10 points, degree 1 surface

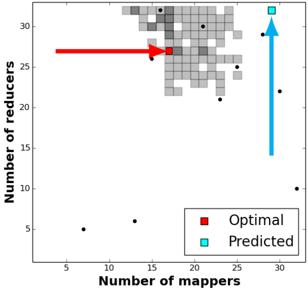
Constructed Model



Model + Conf. Interval



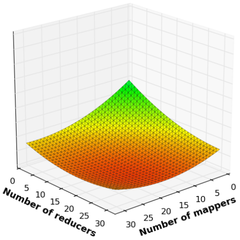
Sampling and Prediction



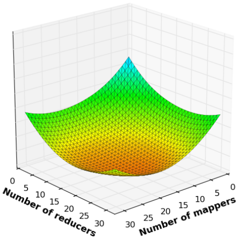
Case Study: Optimizing PageRank

Random sampling: 20 points, degree 2 surface

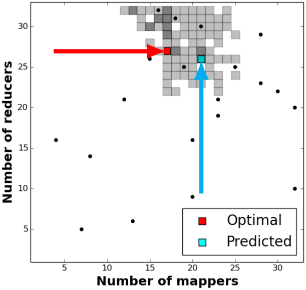
Constructed Model



Model
+
Conf. Interval



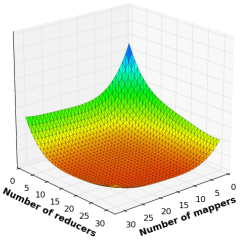
Sampling and Prediction



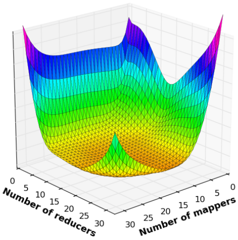
Case Study: Optimizing PageRank

Random sampling: 30 points, degree 4 surface

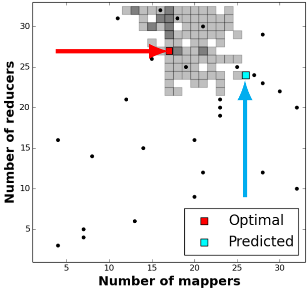
Constructed Model



Model + Conf. Interval

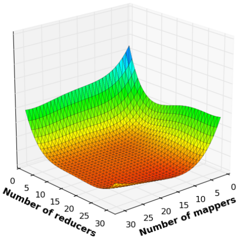


Sampling and Prediction

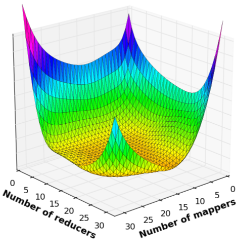


Random sampling: 40 points, degree 4 surface

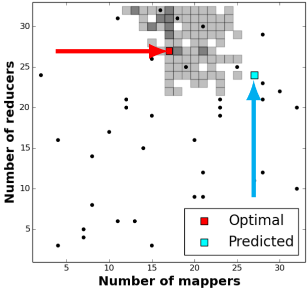
Constructed Model



Model + Conf. Interval



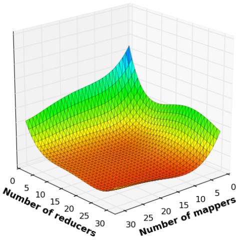
Sampling and Prediction



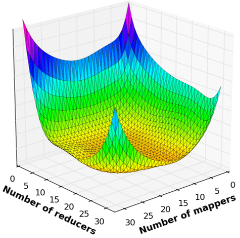
Case Study: Optimizing PageRank

Random sampling: 50 points, degree 4 surface

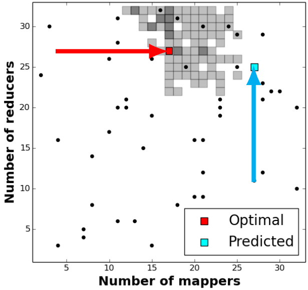
Constructed Model



Model + Conf. Interval



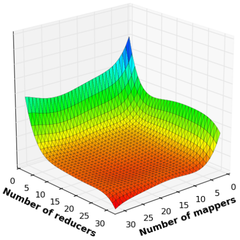
Sampling and Prediction



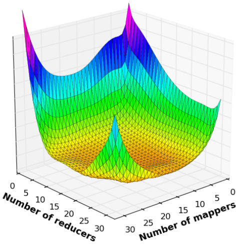
Case Study: Optimizing PageRank

Random sampling: 60 points, degree 5 surface

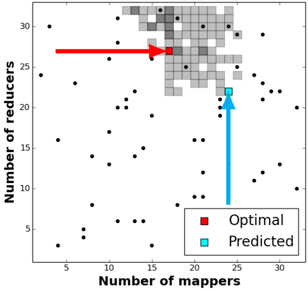
Constructed Model



Model + Conf. Interval



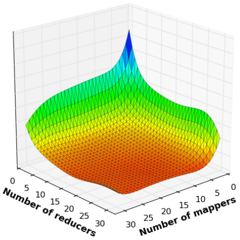
Sampling and Prediction



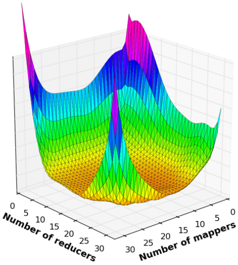
Case Study: Optimizing PageRank

Random sampling: 70 points, degree 6 surface

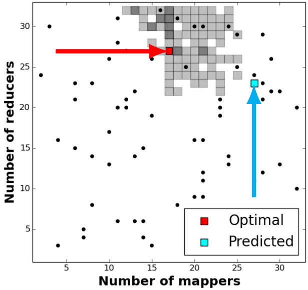
Constructed Model



Model + Conf. Interval



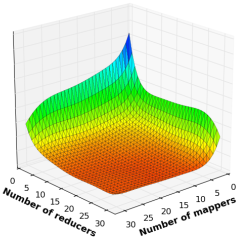
Sampling and Prediction



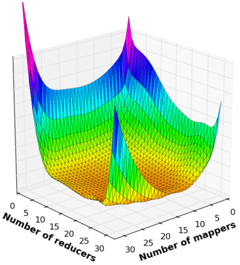
Case Study: Optimizing PageRank

Random sampling: 80 points, degree 6 surface

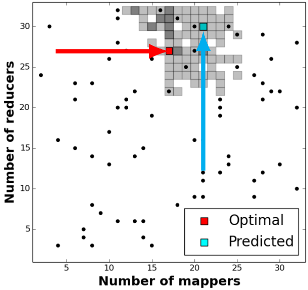
Constructed Model



Model
+
Conf. Interval



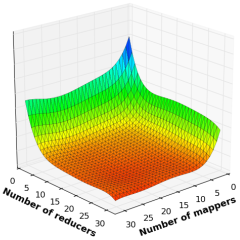
Sampling and Prediction



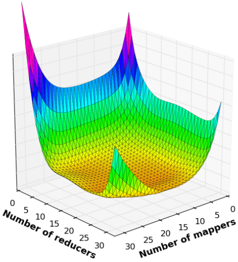
Case Study: Optimizing PageRank

Random sampling: 90 points, degree 5 surface

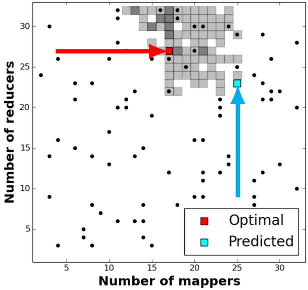
Constructed Model



Model
+
Conf. Interval



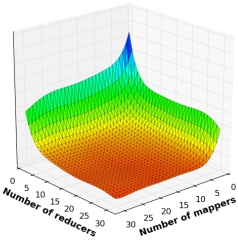
Sampling
and
Prediction



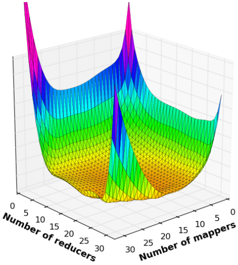
Case Study: Optimizing PageRank

Random sampling: 100 points, degree 6 surface

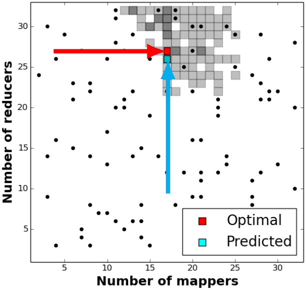
Constructed Model



Model + Conf. Interval

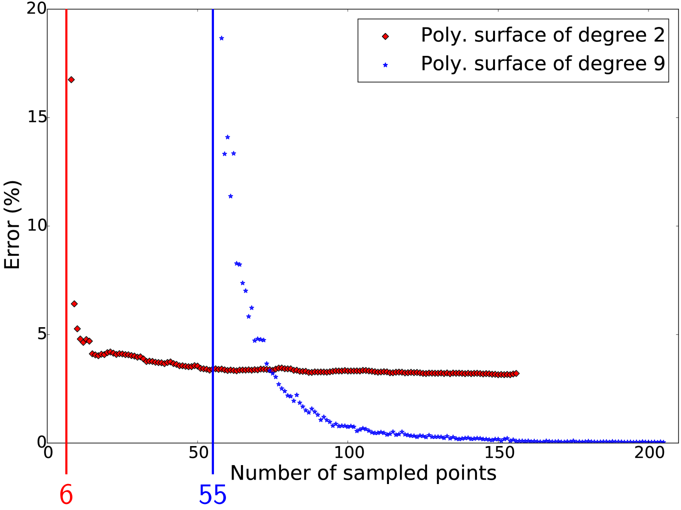


Sampling and Prediction



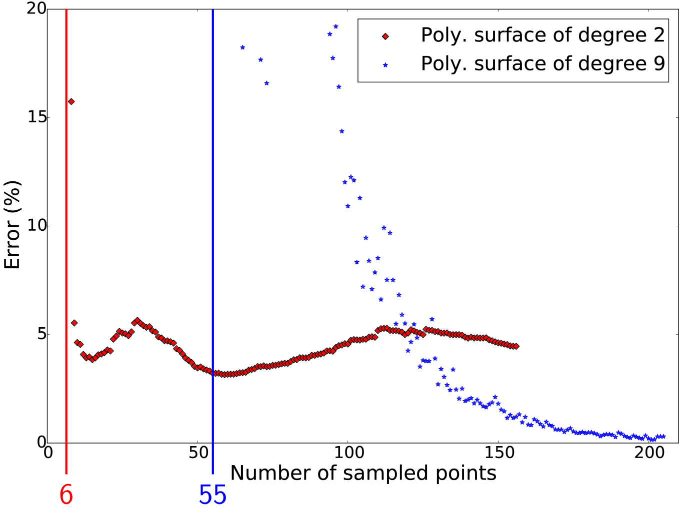
Case Study: Optimizing PageRank

Accuracy by random sampling



Case Study: Optimizing PageRank

Accuracy by grid sampling



Conclusion

- The surrogate model predicted configurations within 1% of optimal sampling only 90 configurations.
  - 90% reduction in sampling from exhuastive search
  - 67% reduction in (expected) sampling compared to grid hill
- The reduction in sampling makes it feasible to
  - tune more parameters
  - explore a wider range of possible parameter values

Thank You

This work is supported by NSF grant 1318445.