Combining Genetic Algorithms & Simulation to Search for Failure Scenarios in System Models

Project Team:
Kevin Mills, computer scientist – kmills@nist.gov
Chris Dabrowski, computer scientist – cdabrowski@nist.gov
Jim Filliben, statistician – jfilliben@nist.gov
Sandy Ressler, information visualization specialist – sressler@nist.gov

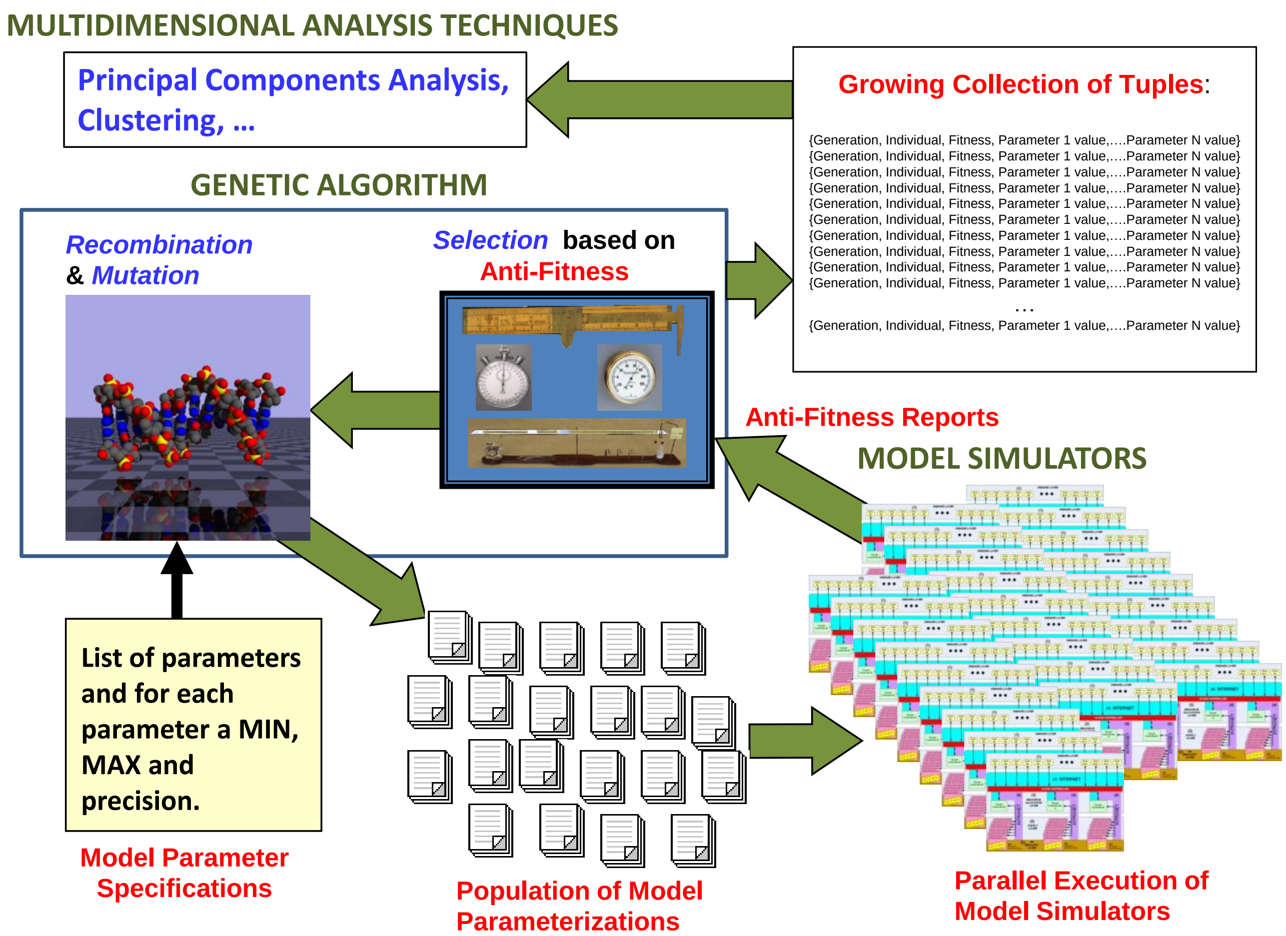
The 5th International Conference on Advances in System Simulation

Oct. 27-Nov. 1, 2013

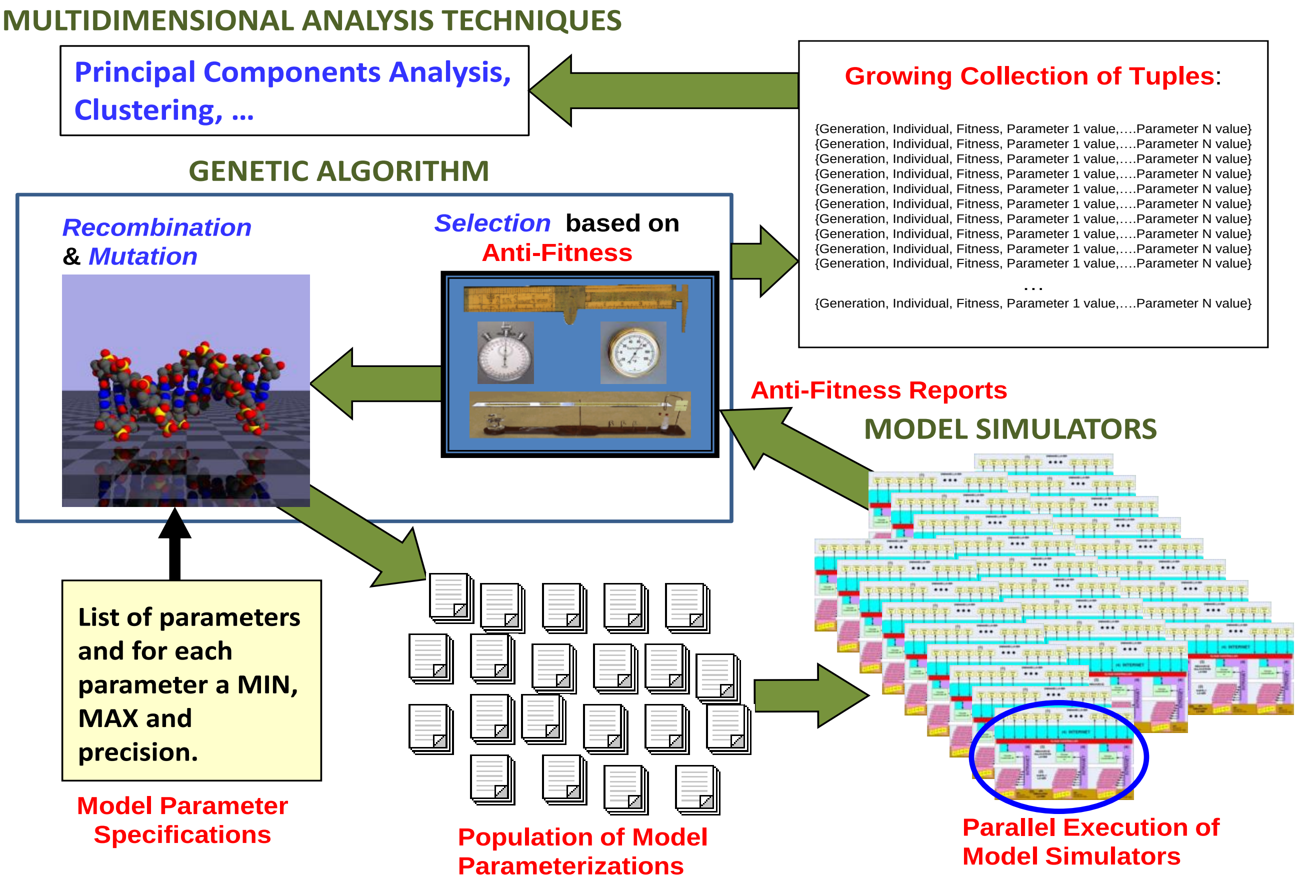
Outline

- Method
- Cloud Computing Case Study in 7 Acts
  1. **Simulator**
  2. **Parameters & Representation as Chromosomes**
  3. **Genetic Algorithm**
  4. **Population of Simulators**
  5. **Dynamics of GA Search**
  6. **Analysis Method**
  7. **Results from Three GA Searches**
- Conclusions

Method: Genetic Algorithm (GA) steers a population of simulators to search for parameter combinations that lead to system failure



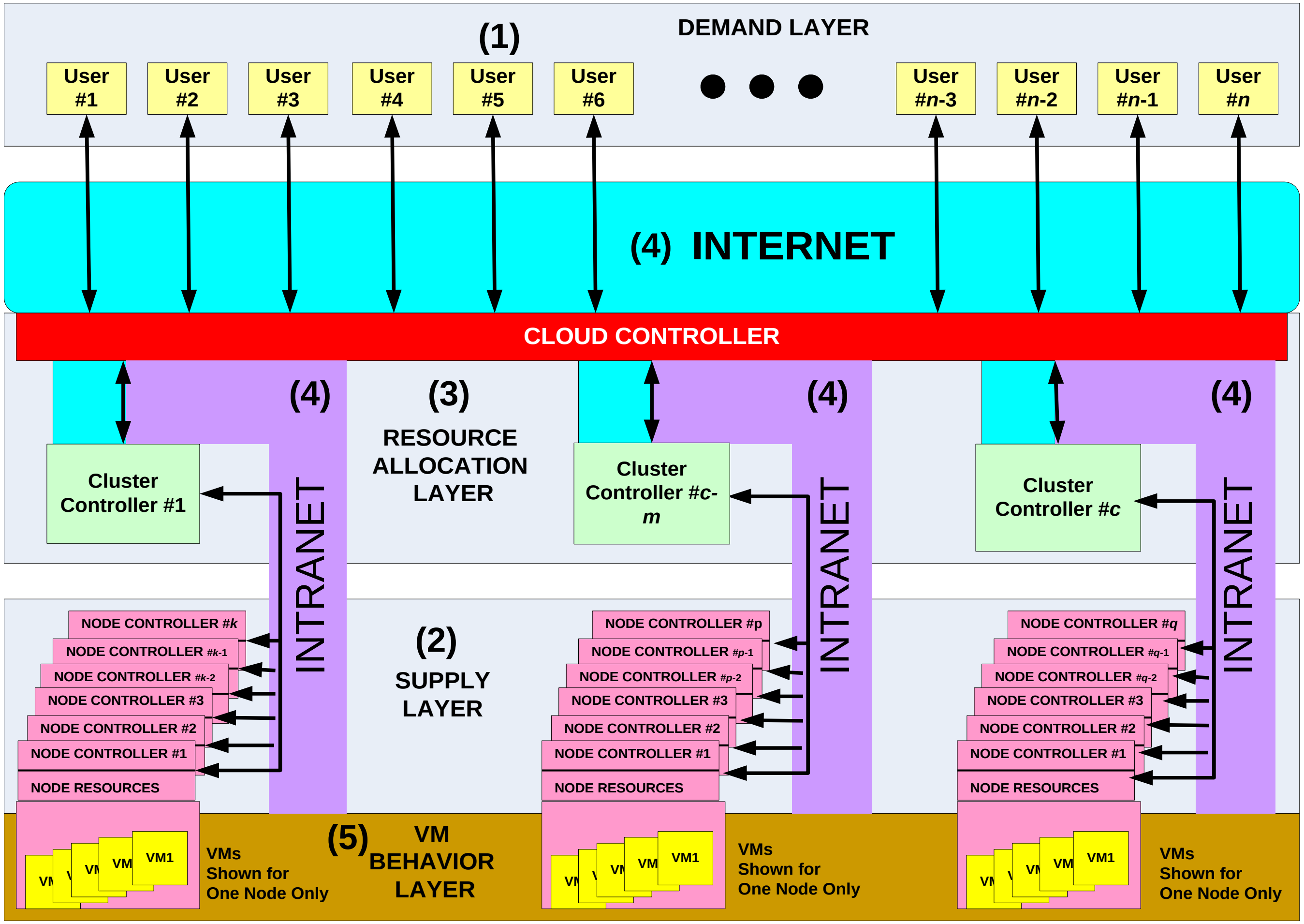
Act 1: *Koala* Simulator



In our case study we defined anti-fitness as the proportion of arriving users not served

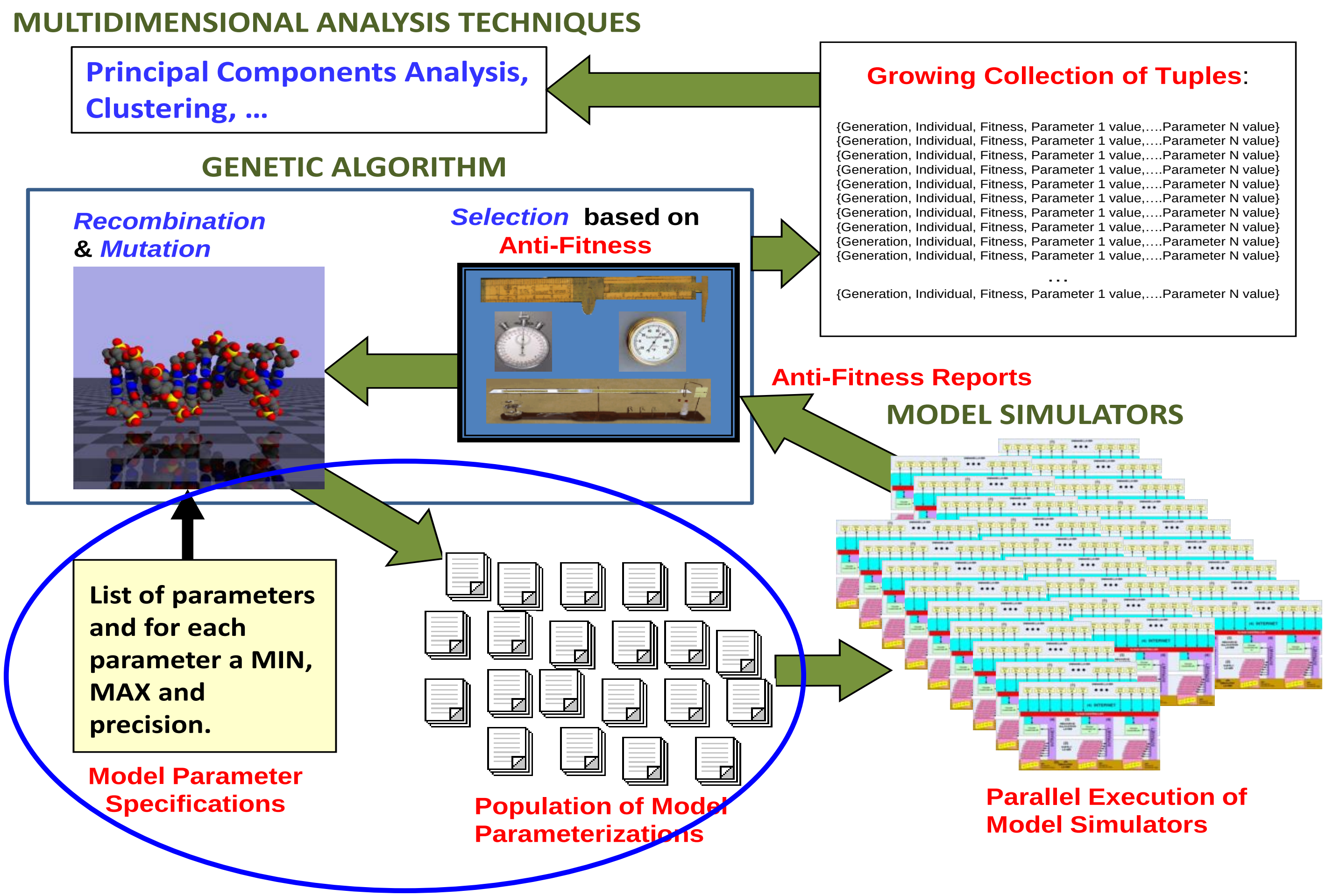
Schematic of *Koala* IaaS Cloud Simulator

Demand and supply layers modeled after Amazon EC2
Internal structure modeled after Eucalyptus v1.6



Mention of specific commercial products does not imply endorsement by NIST

Act 2: *Koala* Parameters & GA Representation as Chromosomes (i.e., bit strings)



Summary of *Koala* Parameters to Search Over

Test Case – Can GA find VM Leakage due to message loss and lack of orphan control?

Failure scenario found manually by accident and described in C. Dabrowski and K. Mills, "VM Leakage and Orphan Control in Open-Source Clouds", *Proceedings of IEEE CloudCom 2011*, Nov. 29-Dec. 1, Athens, Greece, pp. 554-559.

Model Element	Parameter Category				
	Behavior	Structure	Asymmetry	Failure	Total
User	28	2	4	0	34
Cloud Controller	21	4	5	0	30
Cluster Controllers	11	5	3	0	19
Nodes	6	0	0	14	20
Intra-Net/Inter-Net	4	11	2	9	26
Totals	70	22	14	23	129

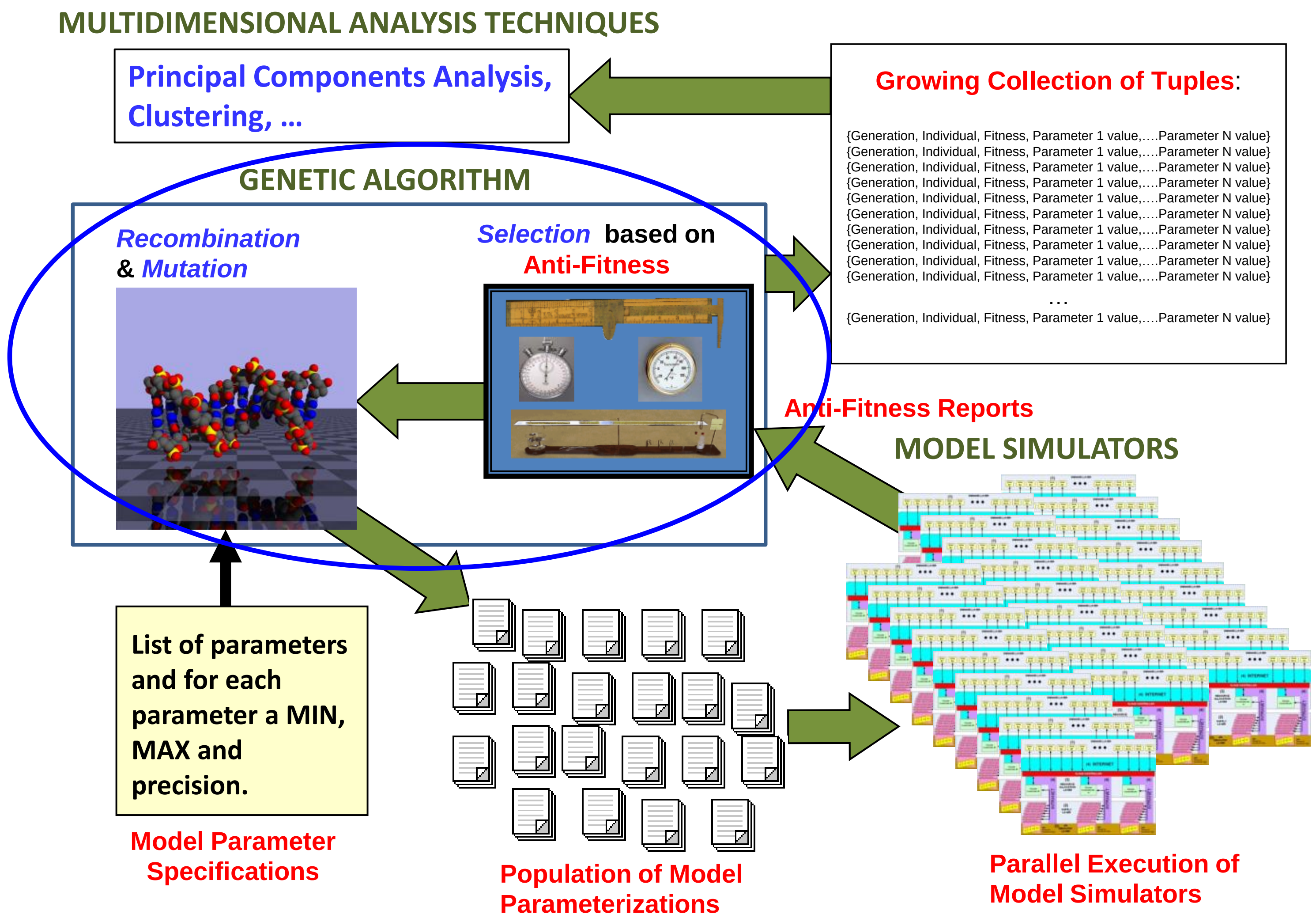
Average # values per parameter is about 6, so search space is ≈ 6^{129}
i.e., ≈ 10^{100} scenarios are possible

- **adapted 125-parameter Koala IaaS simulator to be GA controllable**
- **added 4 *Koala* parameters to turn on/off logic to control (a) creation orphans, (b) termination orphans, (c) relocation orphans and (d) administrator actions**

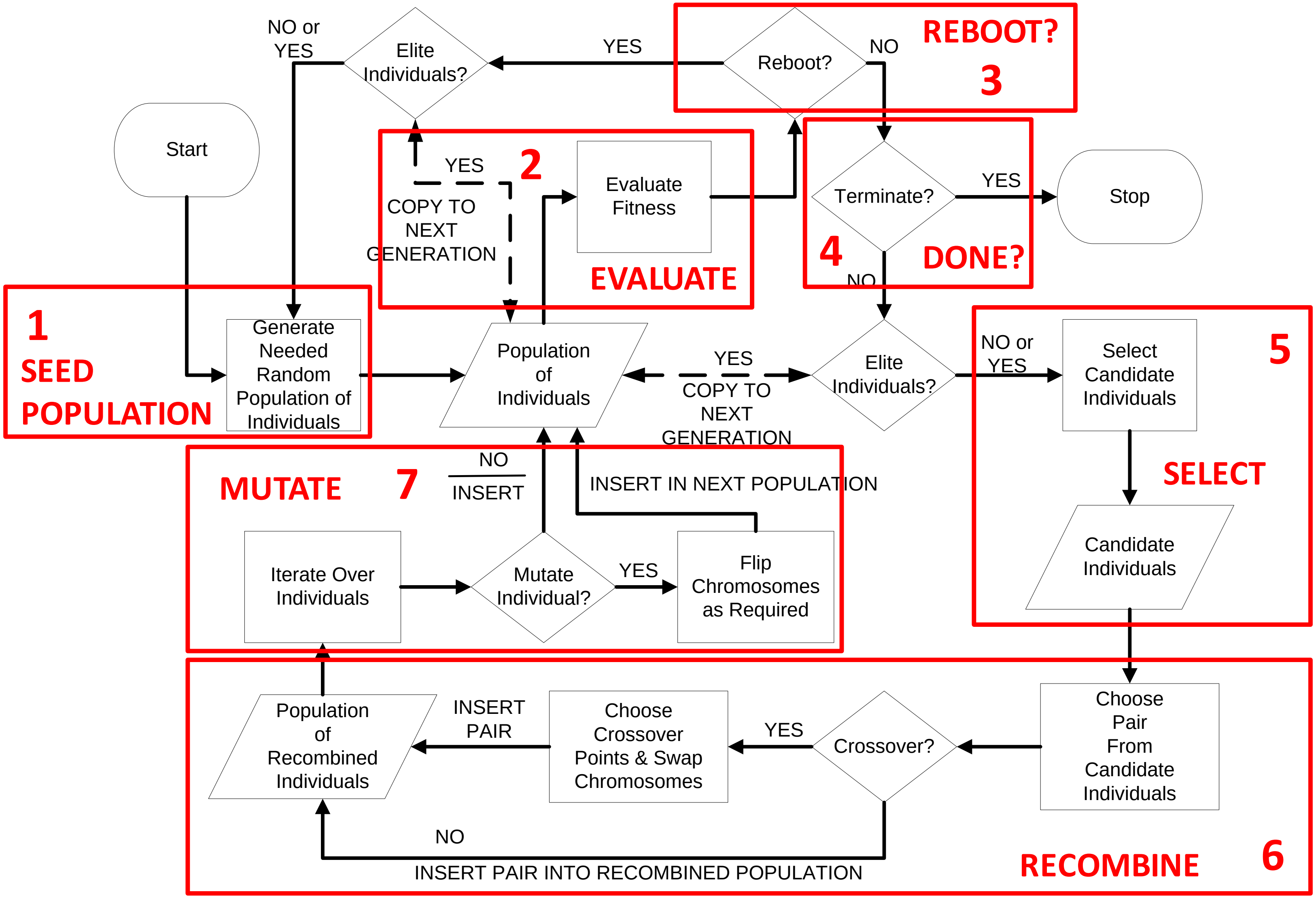
Sample Chromosome Specification

	Koala Parameter Space (Size = 10^{100})			Genetic Algorithm Computed Chromosome Map (Size = 2^{334})			
PARAMETER	MIN	MAX	PRECISION	#VALUES	LOW_BIT	HIGH_BIT	#BITS
P_CreateOrphanControlOn	0	1	1	2	36	36	1
P_TerminationOrphanControlOn	0	1	1	2	58	58	1
P_RelocationOrphanControlOn	0	1	1	2	11	11	1
P_AdministratorActive	0	1	1	2	330	330	1
P_clusterAllocationAlgorithm	0	5	1	6	31	33	3
P_describeResourcesInterval	600	3600	600	6	81	83	3
P_nodeResponseTimeout	30	90	30	3	210	211	2
P_TerminatedInstancesBackOffThreshold	3	6	1	4	56	57	2
P_TerminationBackOffInterval	180	360	60	4	88	89	2
P_TerminationRetryPeriod	600	1200	300	3	316	317	2
P_StaleShadowAllocationPurgeInterval	600	3600	600	6	242	244	3
P_cloudAllocationCriteria	0	3	1	4	321	322	2
P_clusterShadowPurgeLimit	1	21	5	5	290	292	3
P_instancePurgeDelay	180	600	60	8	98	100	3
P_clusterEvaluationResponseTimeout	60	120	30	3	14	15	2
P_MaxPendingRequests	1	10	1	10	72	75	4
P_CloudTerminatedInstancesBackOffThreshold	3	6	1	4	169	170	2
P_CloudTerminationBackOffInterval	180	360	60	4	40	41	2
P_CloudTerminationRetryPeriod	3600	10800	1800	5	297	299	3
P_ClusterShutdownGracePeriod	86400	2.59E+05	43200	5	147	149	3
●	●	●	●	●	●	●	●
P_RequestEvaluatorTimeoutWaitProportion	0.1	0.4	0.1	4	145	146	2
P_RequestEvaluatorClusterMinimumResponse	0.6	0.9	0.1	3	269	270	2
P_MaxRelocationDuratonProportion	0.65	0.95	0.1	4	90	91	2
P_MaximumRelocateDescribeRetries	4	16	2	7	254	256	3
P_AverageCloudAdministratorAttentionLatency	28800	86400	14400	5	308	310	3
P_AverageCloudAdministratorShutdownDelay	300	900	300	3	45	46	2
P_avgTimeToClusterCommunicationCut	2.88E+06	2.88E+07	2.88E+06	10	217	220	4

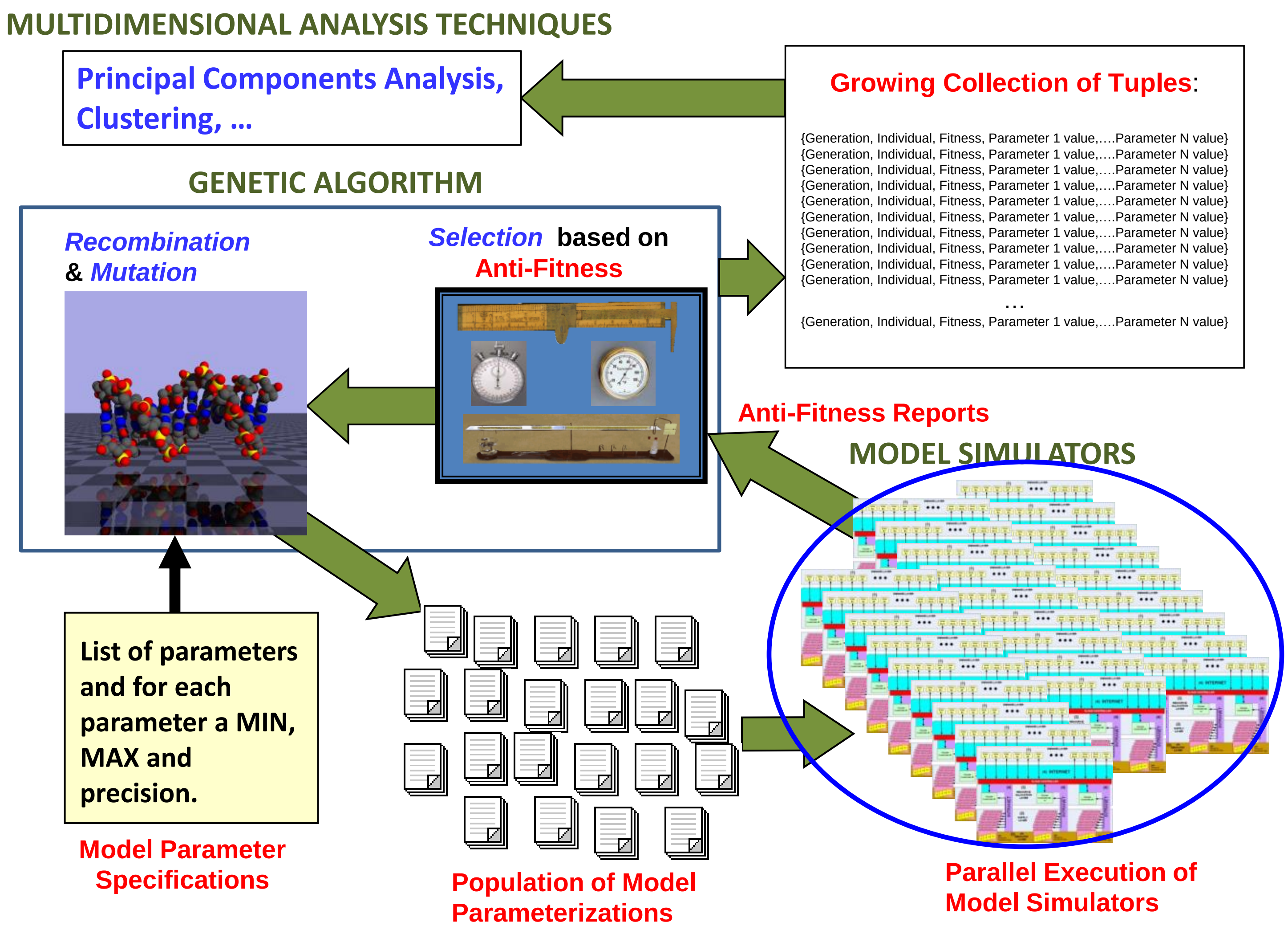
Act 3: Genetic Algorithm



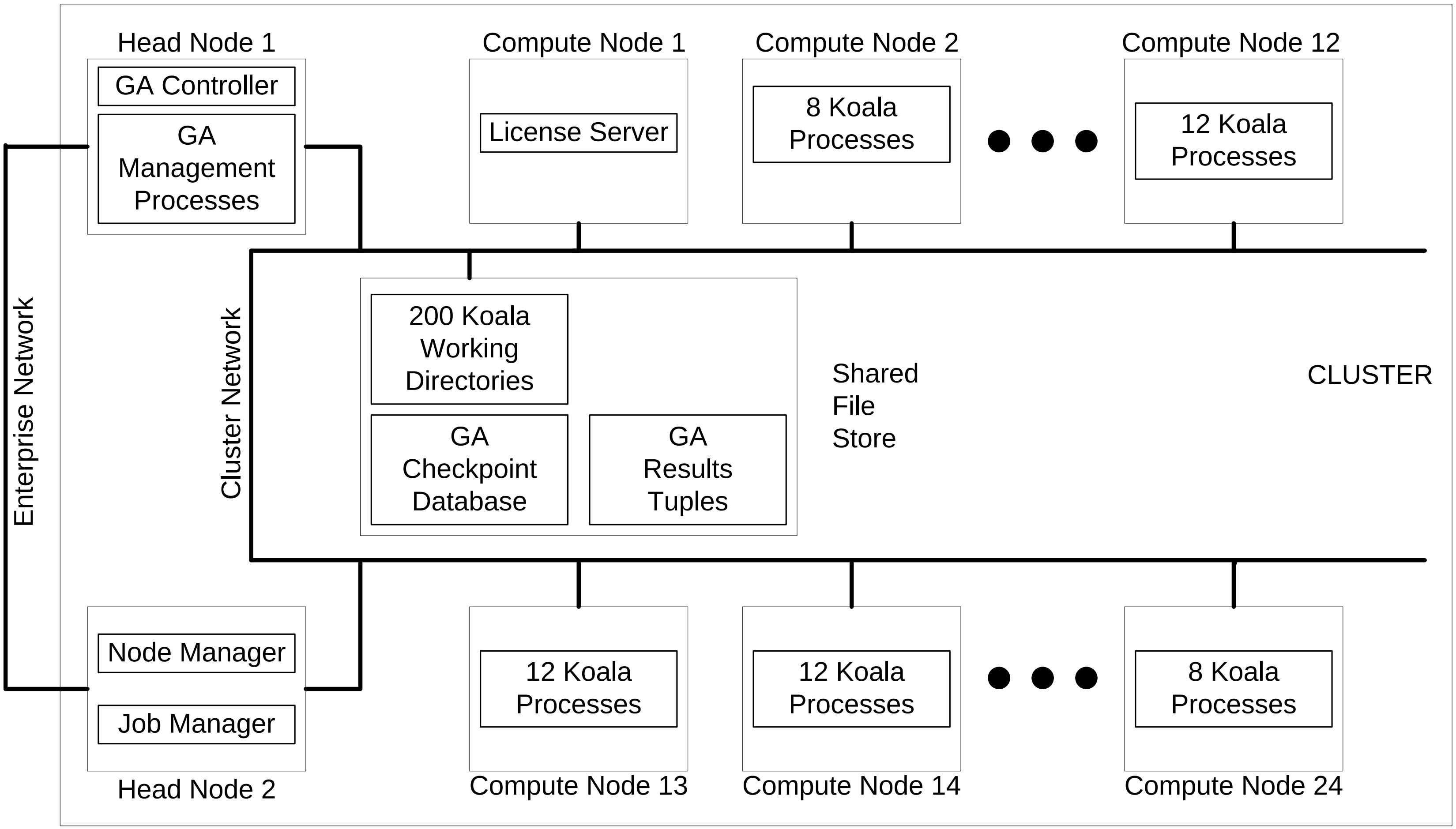
Genetic Algorithm Flow Chart



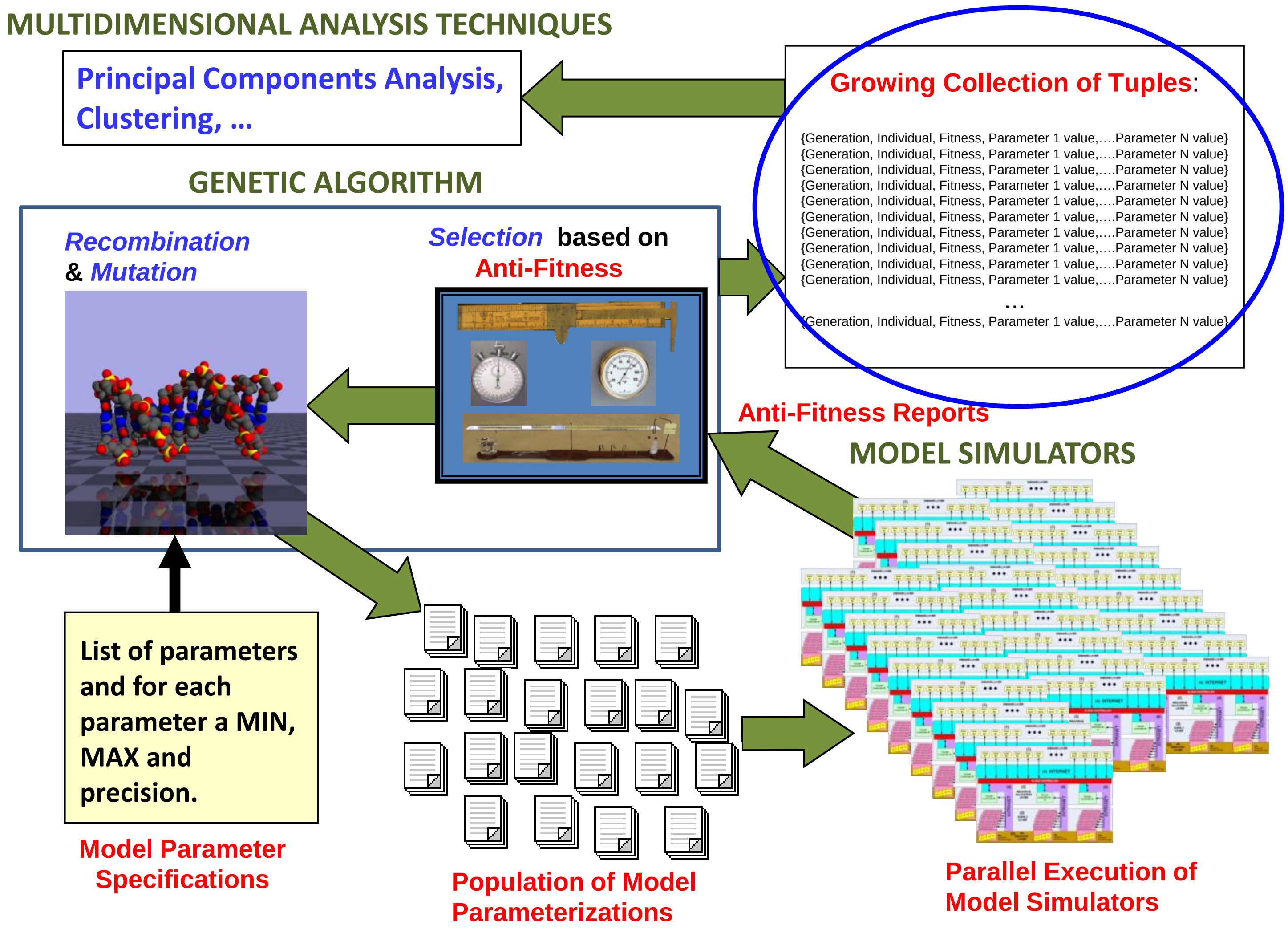
Act 4: Population of *Koala* Simulators



Population of *Koala* Simulators Deployed on a High Performance Computing Cluster



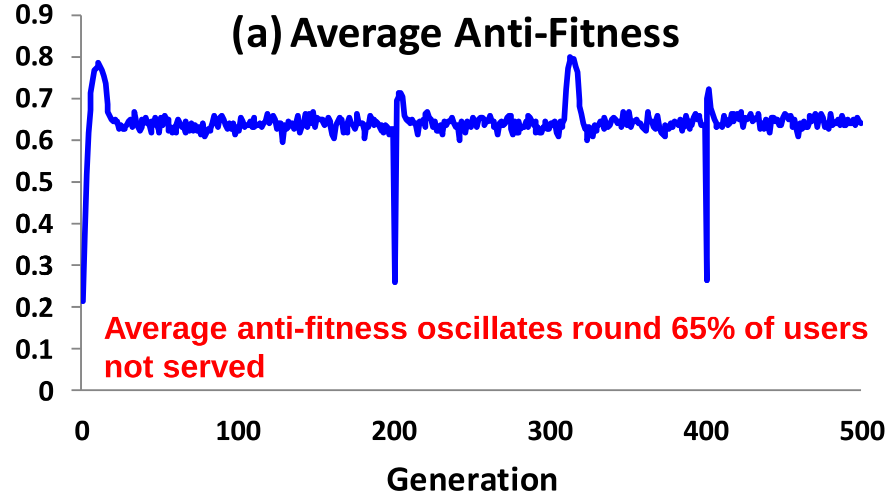
Act 5: Dynamics of GA Search

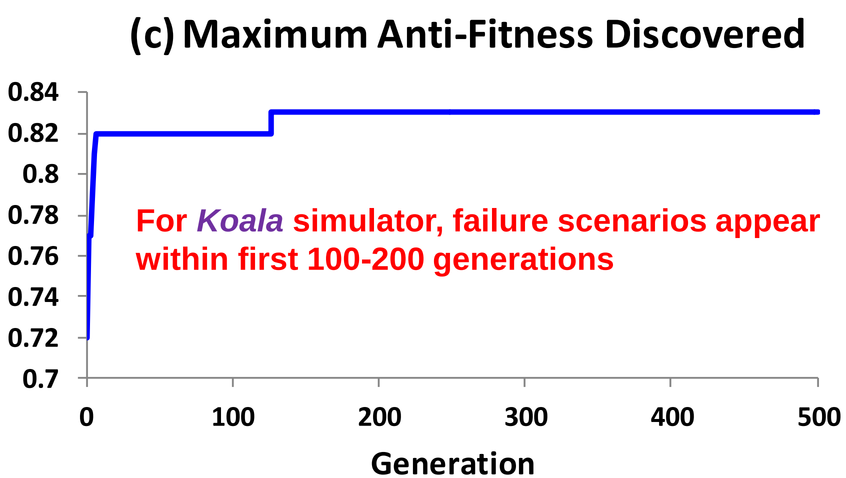


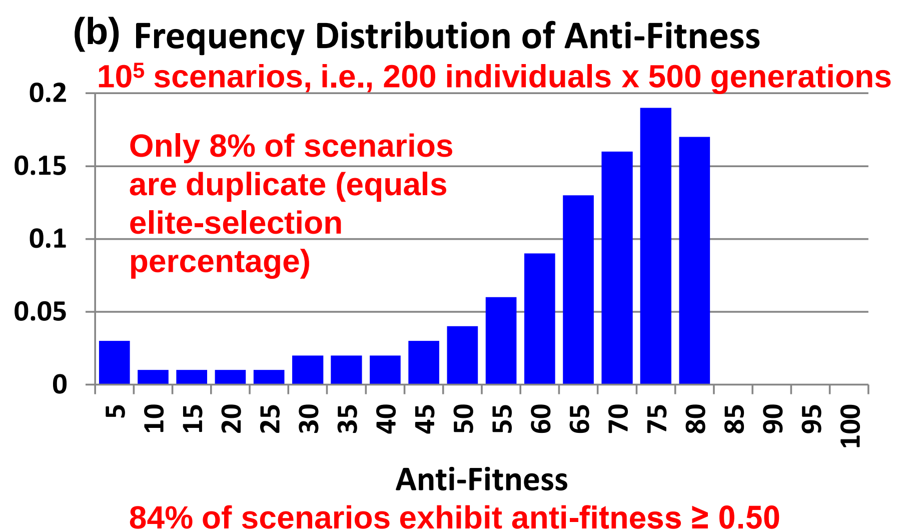
One GA Search over 500 Generations

GENETIC ALGORITHM CONTROL PARAMETERS

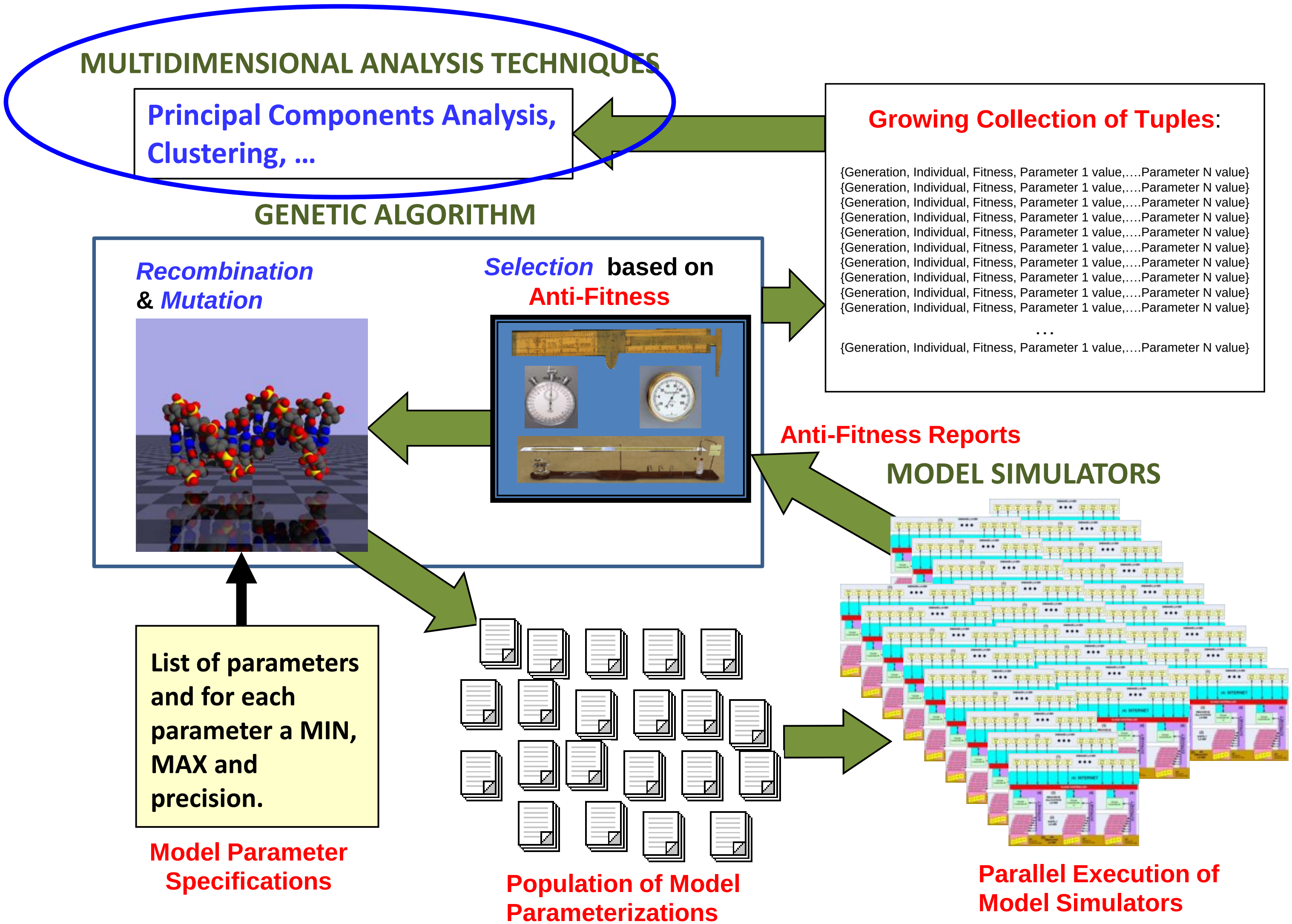
Generations	500
Population Size	200 Individuals
Elite Per Generation	16 Individuals
Reboot After	200 Generations
Selection Method	Stochastic Uniform Sampling
# Crossover Points	3
Mutation Rate	0.001 $\leq$ Adaptive $\leq$ 0.01







Act 6: Analysis Method



Differential Probability Analysis

Let $\boldsymbol{C}$ be the set of collected tuples, each containing a vector of parameter value (PV) pairs and a corresponding anti-fitness value, $\boldsymbol{f}$

Segment $\boldsymbol{C}$ into high-pass ($\boldsymbol{H}$) and low-pass ($\boldsymbol{L}$) subsets, where:

$$\boldsymbol{H} = \{\boldsymbol{x} \in \boldsymbol{C} \mid \boldsymbol{f_x} > 0.70\} \text{ and } \boldsymbol{L} = \{\boldsymbol{x} \in C \mid \boldsymbol{f_x} < 0.15\}$$

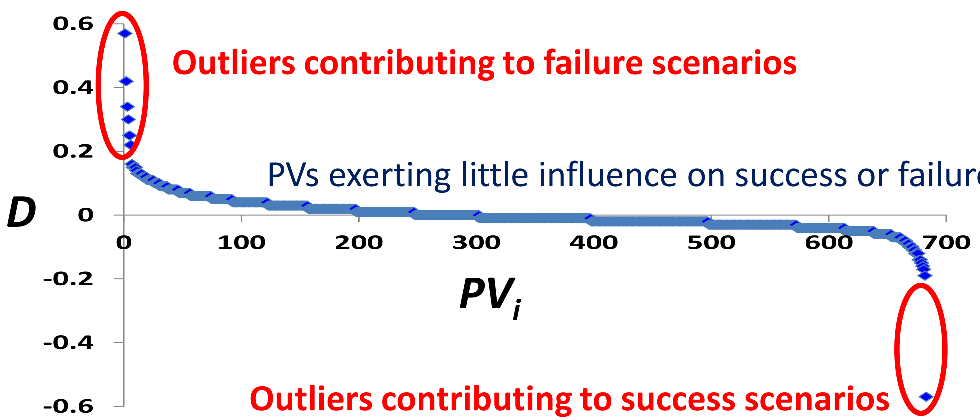
For each PV estimate the probability of occurrence in $\boldsymbol{H}$ and $\boldsymbol{L}$:

$$\mathrm{P}(\mathrm{PV}_i \mid \boldsymbol{f} > 0.70 = \mid \mathrm{PV}_i \in \boldsymbol{H} \mid / \mid \boldsymbol{H} \mid \text{ and } \mathrm{P}(\mathrm{PV}_i \mid \boldsymbol{f} > 0.15) = \mid \mathrm{PV}_i \in \boldsymbol{L} \mid / \mid \boldsymbol{L} \mid$$

Then compute the estimated differential probability:

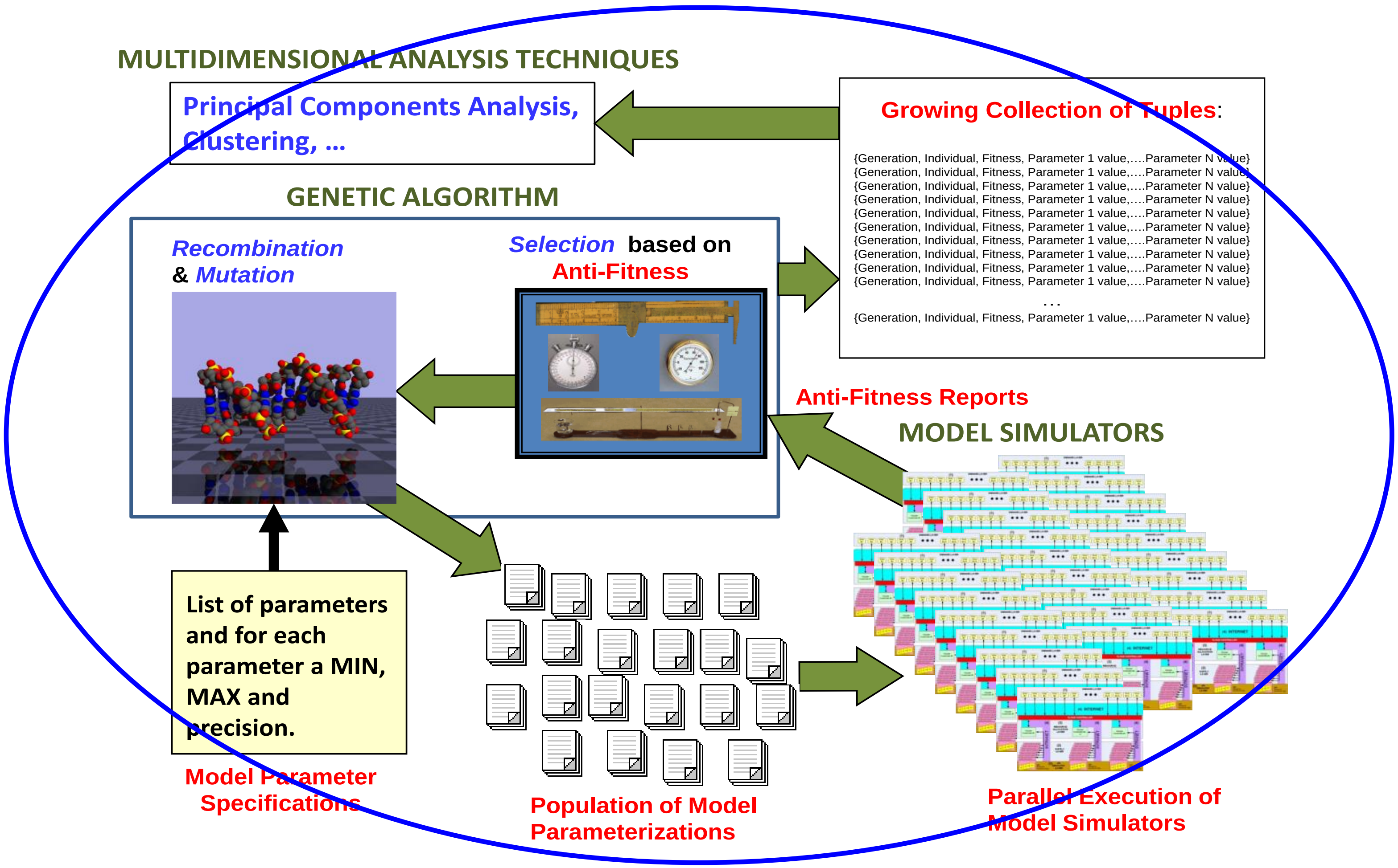
$$\boldsymbol{D} = \mathrm{P}(\mathrm{PV}_i \mid \boldsymbol{f} > 0.70) - \mathrm{P}(\mathrm{PV}_i \mid \boldsymbol{f} < 0.15)$$

Plot $\boldsymbol{D}$ for each PV pair



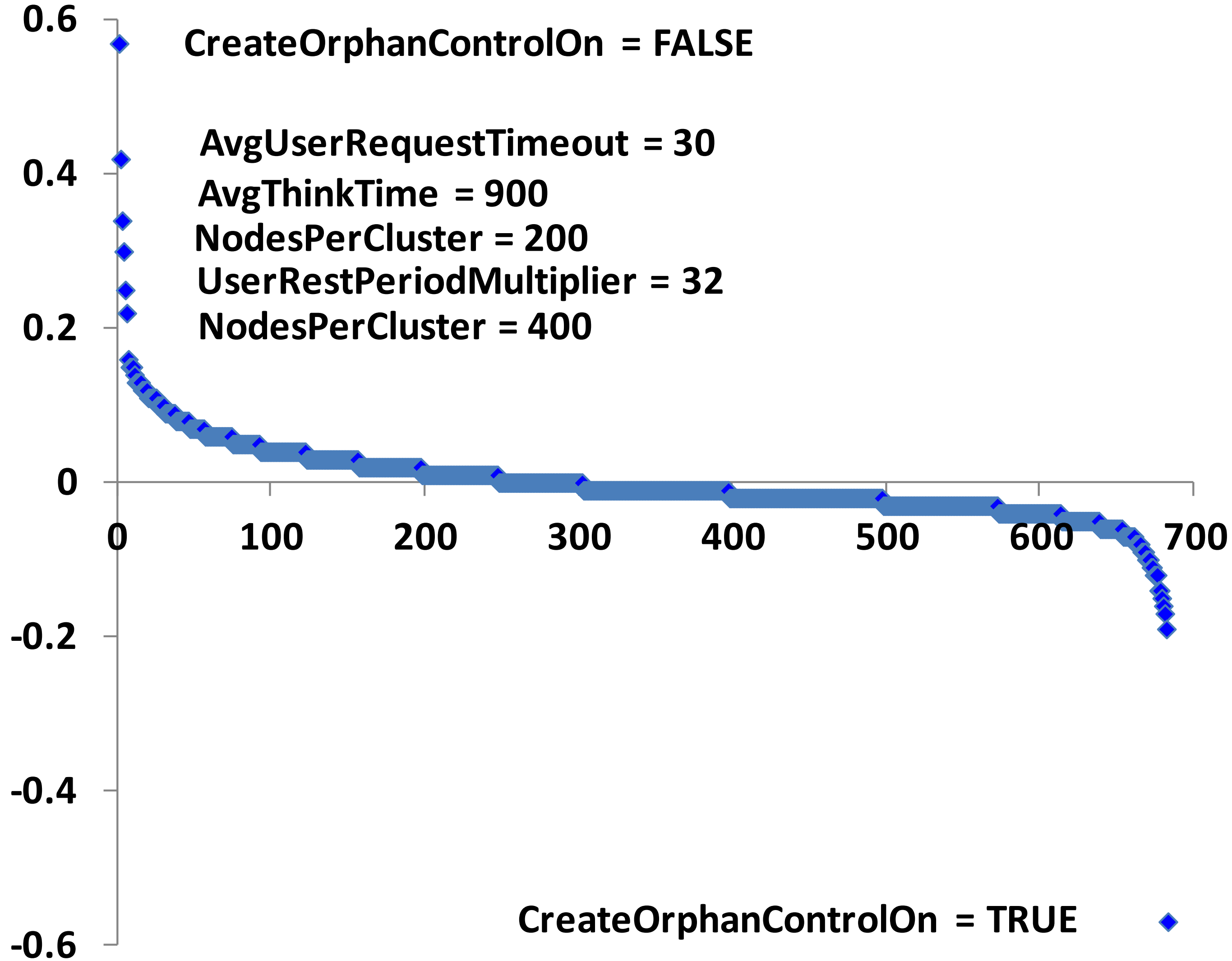
PV pairs sorted by decreasing *D*

Act 7: Results from Four GA Searches



Analysis of Results from GA Search 1 – 500 Generations

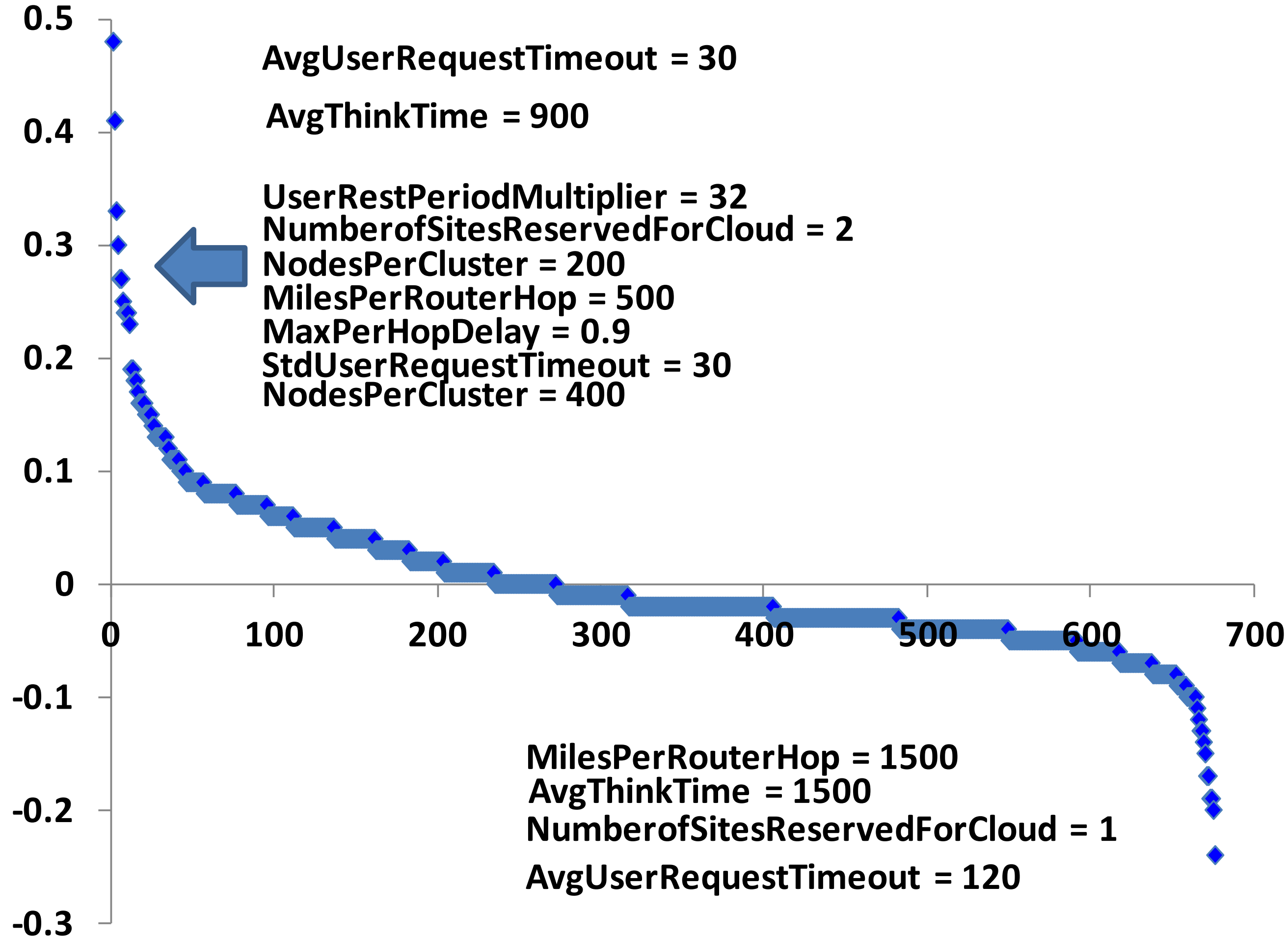
Seeking Known Failure Scenario – search duration 30 days



D (*y*-axis) for 684 PV pairs (*x*-axis) for first GA search–outlier PV pairs labeled.

Analysis of Results from GA Search 2 – 205 Generations

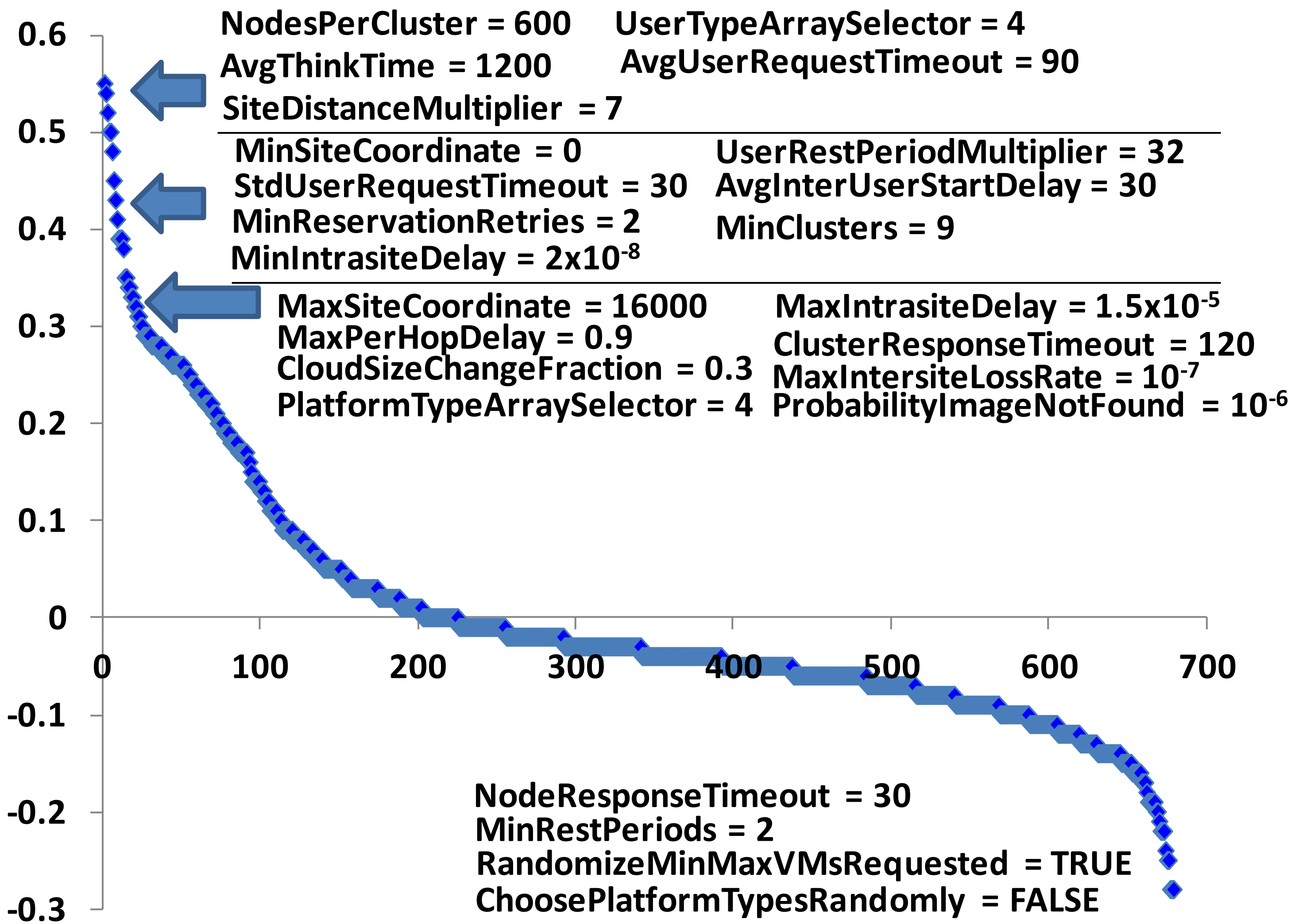
Seeking Previously Unknown Failure Scenarios – search duration 14 days



D (*y*-axis) for 677 PV pairs (*x*-axis) for first GA search–outlier PV pairs labeled.

Analysis of Results from GA Search 3 – 209 Generations

Nudging up Some Parameter Ranges and Seeking Additional Failure Scenarios - search duration 16 days



D (*y*-axis) for 680 PV pairs (*x*-axis) for first GA search–outlier PV pairs labeled.

Conclusions

SUMMARY:

- Defined a design-time method, combing GA search with simulation, to seek failure scenarios in system models
- Applied the method in a case study, seeking (and finding) a known failure scenario in an existing cloud simulator
- Iterated search to reveal previously unknown failure scenarios

FINDINGS:

- GA searches explored predominantly non-duplicative scenarios with high anti-fitness
- Uncovered evidence that GA search can reveal insights about optimal parameters settings, while simultaneously searching for failure scenarios
- GA search should be pursued only for systems with sufficient schedule time, and where failure scenarios have high cost