Genetic Algorithms
Parameter Settings

Piero P. Bonissone

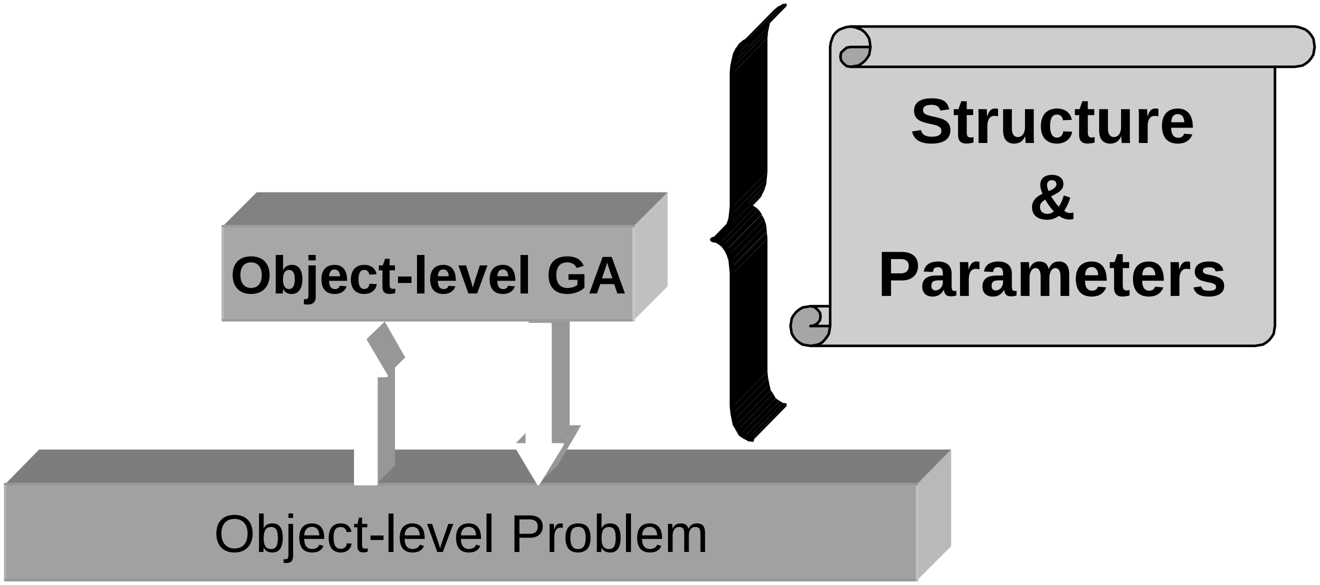
GE Corporate Research & Development

Bonissone@crd.ge.com

GA Parameter Setting - Outline

- **GA Model:** Structure, Parameters
- **GA Parameter Setting**
  - GA Parameter Tuning: Off-line study, Off-line Meta-GA
  - GA Parameter Control: Deterministic Control, Self-Adaptation, Control (FLC)
- **GA Parameter Setting: An Application to Agile Mfg.**
  - Object-level Representation and Complexity
- **First Solution Architecture (U-TGA, TGA, FLC-GA)**
  - FLC-GA: KB, I/O, Control Parameters, Rule base and Scaling Factors
  - Results - Performance Tradeoff
  - Problems with First Solution
  - Summary and Analysis of 30 Experiments
  - Addressing Problems with First Solution
- **Second Solution**
  - FLC: KB, Inputs, Outputs, Termsets, Rule Sets, Control Parameters
  - Statistical Experiments: Set-Up & Measures
  - Summary and Analysis of 1200 Experiments
- **Conclusions and Next Steps**

GA Model



Object-level GA

Structure & Parameters

Object-level Problem

GA Structure

GA Structural Design Selections:

- GA Type:
  - {Simple, Steady-State, Niche,…}
- Chromosome Encoding:
  - {Binary, Integer, Real,...}
- Constraints Representation:
  - {Penalty function, data structure, filters, …}
- Fitness Function:
  - {Scalar function (weighted aggregation of multiple functions) …}
- Selection Method:
  - {Proportional Roulette, Tournament, Rank, Uniform, ... }
- Crossover Operator:
  - {Once-cut, Two-cuts, Uniform, BLX, Parent Weighted, ...}
- Mutation Operator:
  - Mutation Rate: {Exponentially Decreasing, Uniform, ..}
  - Value: {Exp. Decreasing, Uniform, Normally Distr., …}

GA Parameters

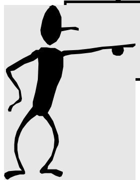
Adjustable parameters for a GA

- **N** = Population size - large pop. prevent premature convergence
- **P_c** = Crossover rate: P_{cr} * N = # crossovers per generation
- **P_m** = Mutation rate: P_m * N * L = # mutations per generation
- **G** = Generation Gap - percentage of population to be replaced
- **W** = Scaling Window Size =[1, 7]
- **S** = Selection Strategy = {Elitist, Non-Elitist}

Other possible parameters (that could be adjusted):

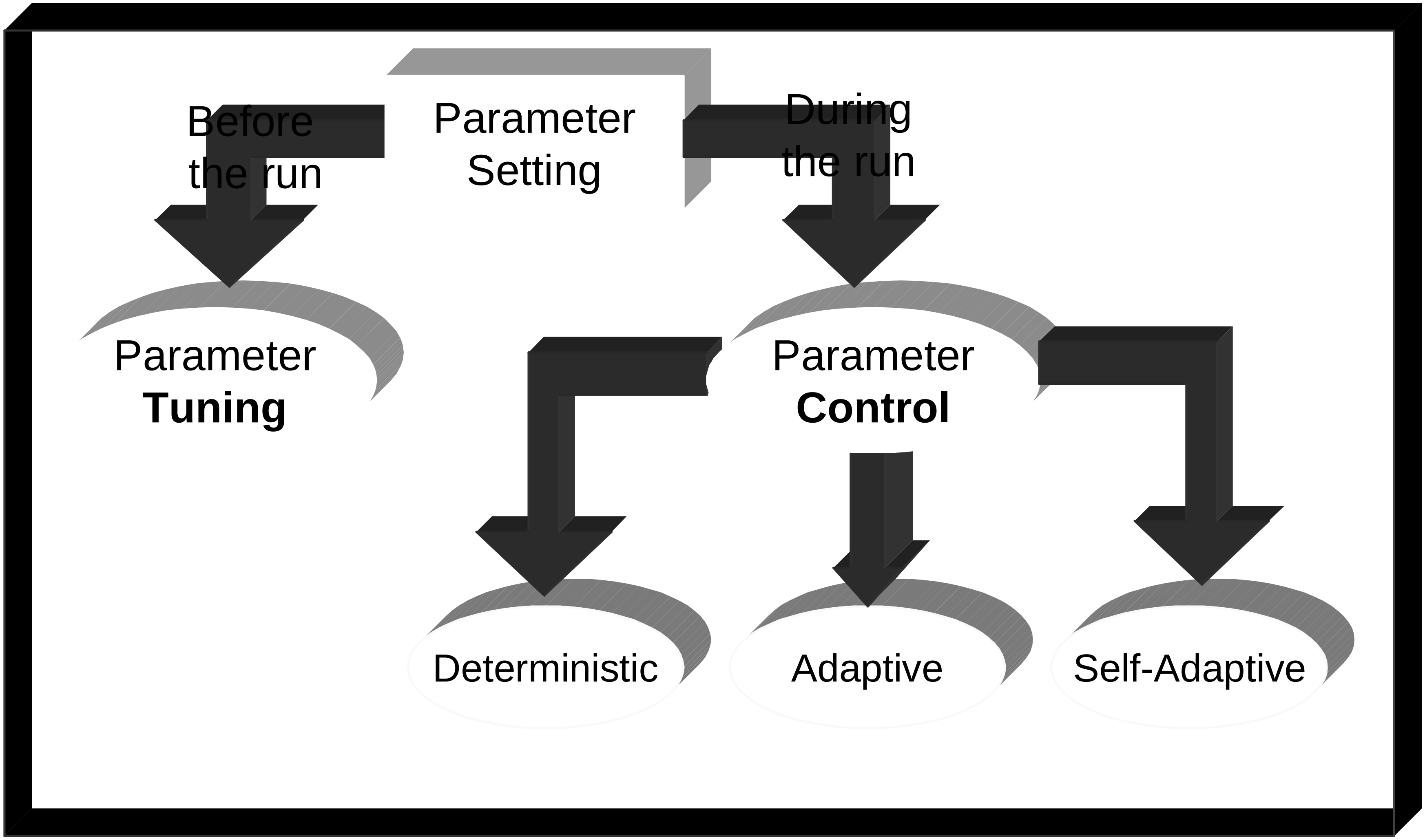
- **T** = Number of Trials = ΣN_i - where i= 1, Max_Gen
- **S_m** = Mutation step - (σ in Normally distrib. Mutation value)
- **P_S** = Probability of Selection - (Parametrized slope of prob distrib.)
- **A_S** = Arity of Parents - number of parents in recombination

GA Parameter Setting - Outline



- GA Model: Structure, Parameters
- GA Parameter Setting
  - GA Parameter Tuning: Off-line study, Off-line Meta-GA
  - GA Parameter Control: Deterministic Control, Self-Adaptation, Control (FLC)
- GA Parameter Setting: An Application
  - Object-level Representation and Complexity
- First Solution Architecture (U-TGA, TGA, FLC-GA)
  - FLC-GA: KB, I/O, Control Parameters, Rule base and Scaling Factors
  - Results - Performance Tradeoff
  - Problems with First Solution
  - Summary and Analysis of 30 Experiments
  - Addressing Problems with First Solution
- Second Solution
  - FLC: KB, Inputs, Outputs, Termsets, Rule Sets, Control Parameters
  - Statistical Experiments: Set-Up & Measures
  - Summary and Analysis of 1200 Experiments
- Conclusions and Next Steps

GAs Parameter Setting



Parameter Setting

Before the run

During the run

Parameter **Tuning**

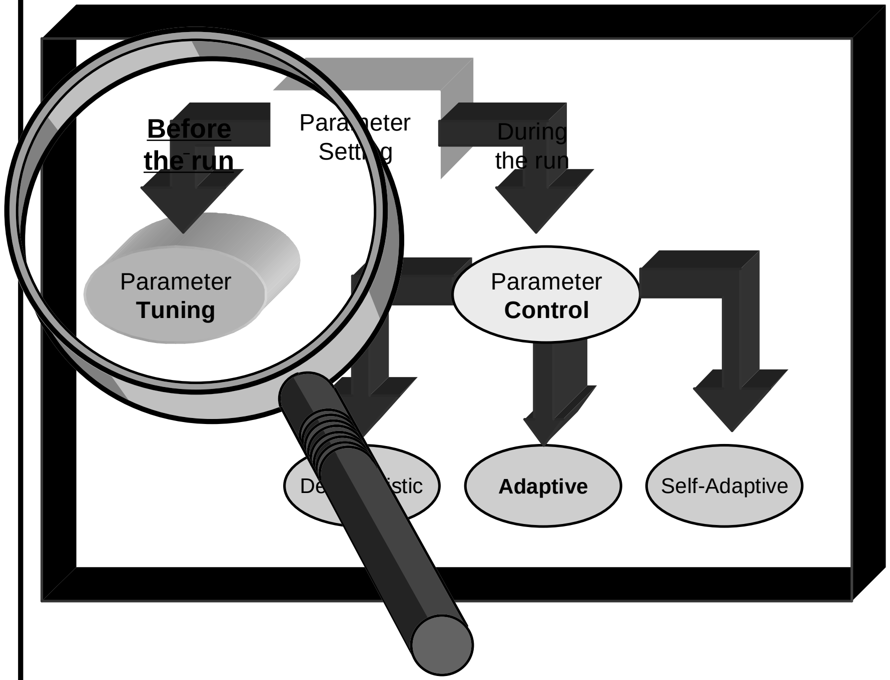
Parameter **Control**

Deterministic

Adaptive

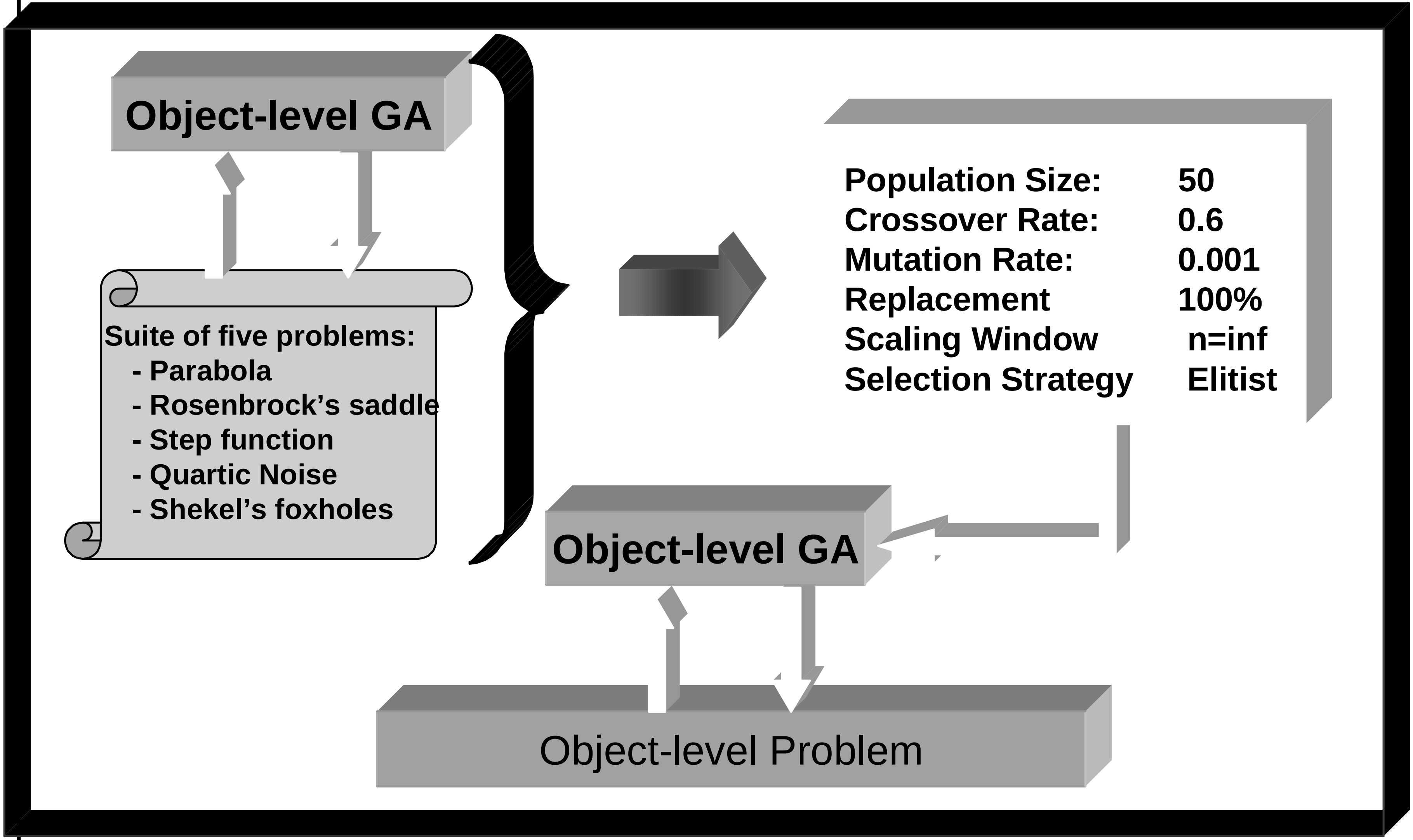
Self-Adaptive

GAs Parameter Setting: Parameter Tuning



- ***<u>Off-line Tuning</u>***
- **Determined before running the GAs on the object-level problem by**
  - Studying a subset of five diverse problems (*DeJong, 1975*)
  - Running a Meta-Genetic Algorithm (*Grefenstette, 1986*)

Off Line Tuning of GA Parameters *(DeJong, 1975)*



Object-level GA

Suite of five problems:

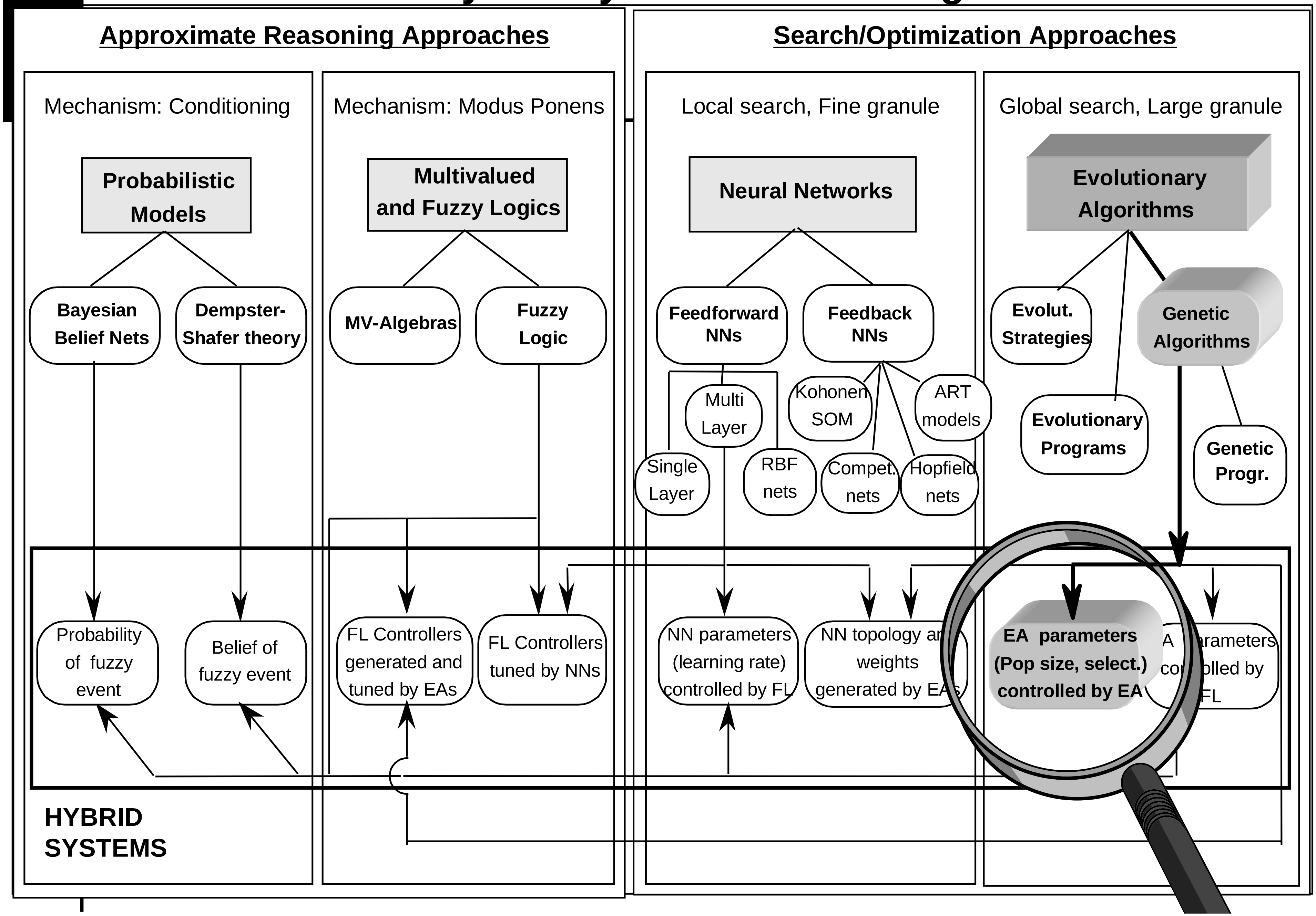
- Parabola
- Rosenbrock’s saddle
- Step function
- Quartic Noise
- Shekel’s foxholes

Population Size:	**50**
Crossover Rate:	**0.6**
Mutation Rate:	**0.001**
Replacement	**100%**
Scaling Window	**n=inf**
Selection Strategy	**Elitist**

Object-level GA

Object-level Problem

SC Hybrid Systems: EA Tuning EA



Approximate Reasoning Approaches

Mechanism: Conditioning

Probabilistic Models

Bayesian Belief Nets

Dempster-Shafer theory

Mechanism: Modus Ponens

Multivalued and Fuzzy Logics

MV-Algebras

Fuzzy Logic

Search/Optimization Approaches

Local search, Fine granule

Neural Networks

Feedforward NNs

Feedback NNs

Multi Layer

Single Layer

RBF nets

Kohonen SOM

ART models

Compet. nets

Hopfield nets

Global search, Large granule

Evolutionary Algorithms

Evolut. Strategies

Genetic Algorithms

Evolutionary Programs

Genetic Progr.

Probability of fuzzy event

Belief of fuzzy event

FL Controllers generated and tuned by EAs

FL Controllers tuned by NNs

NN parameters (learning rate) controlled by FL

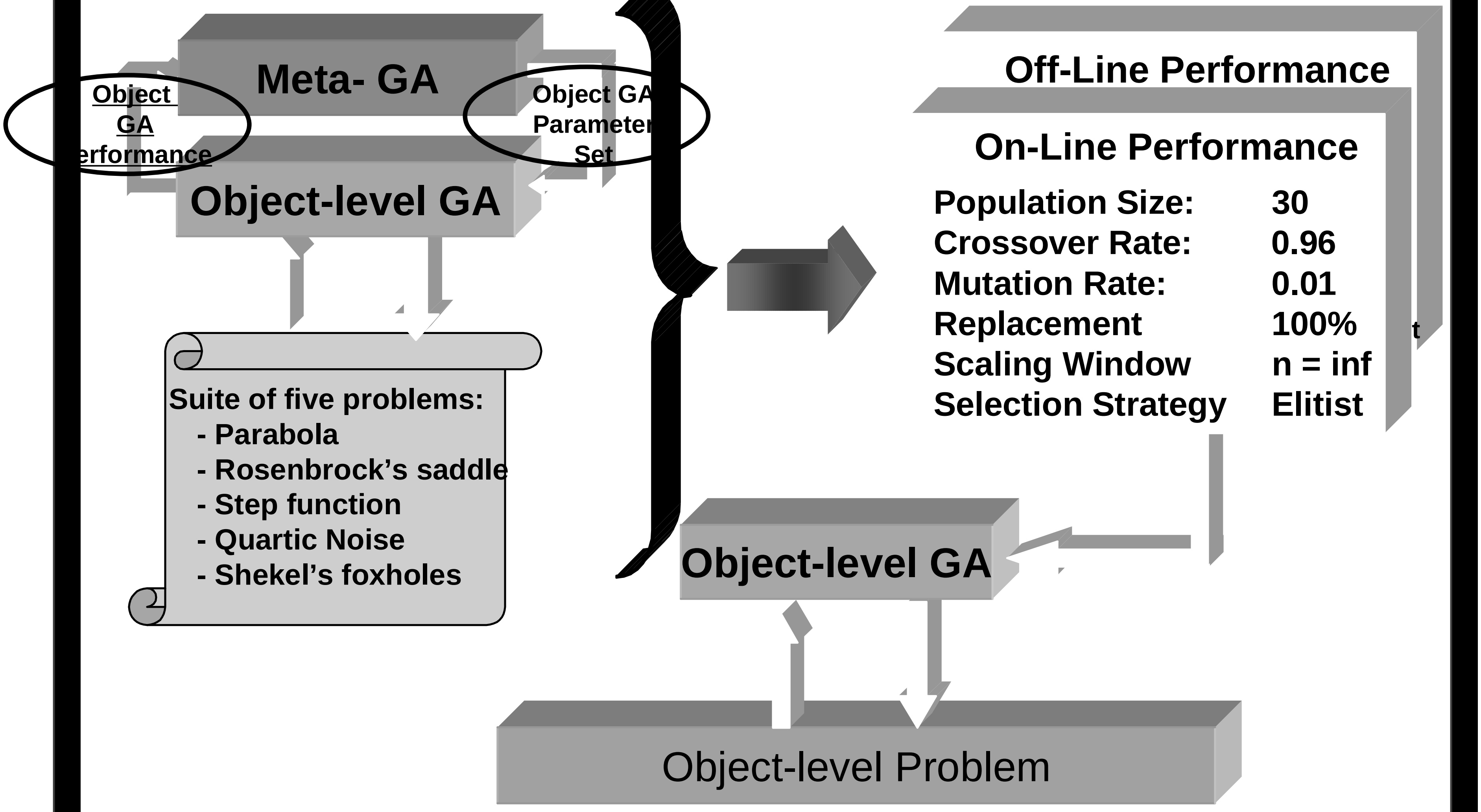
NN topology ar weights generated by EAs

EA parameters (Pop size, select.) controlled by EA

A arameters co lled by FL

HYBRID SYSTEMS

Off Line Tuning of GA Parameters *(Grefenstette, 1986)*



Suite of five problems:

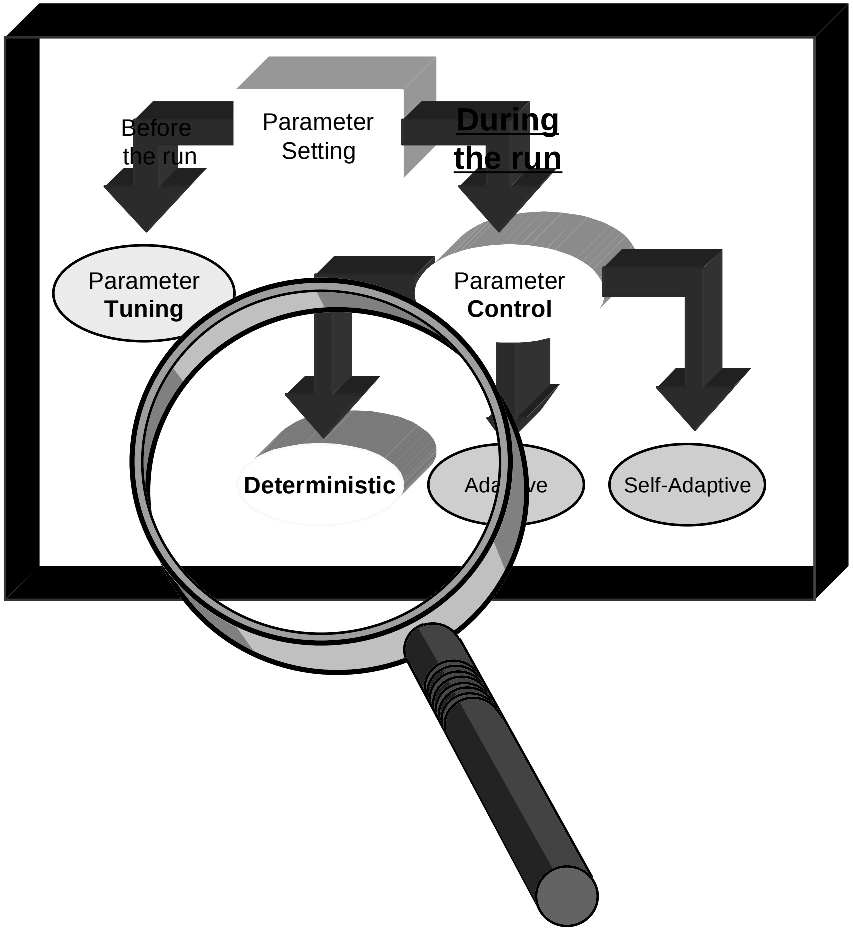
- Parabola
- Rosenbrock's saddle
- Step function
- Quartic Noise
- Shekel's foxholes

Off-Line Performance

On-Line Performance

Population Size:	30
Crossover Rate:	0.96
Mutation Rate:	0.01
Replacement	100%
Scaling Window	n = inf
Selection Strategy	Elitist

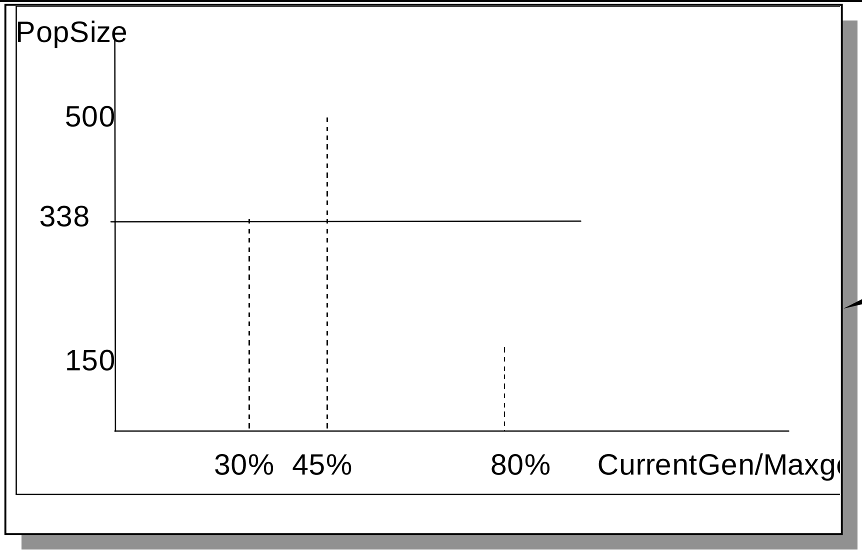
GAs Parameter Setting: Deterministic Control



- ***No feedback*** **information is used.**
- **A time-varying schedule is used to modify a GA parameter *p***
- ***p is replaced by p(t)***
- **Correct design of p(t) is very difficult**

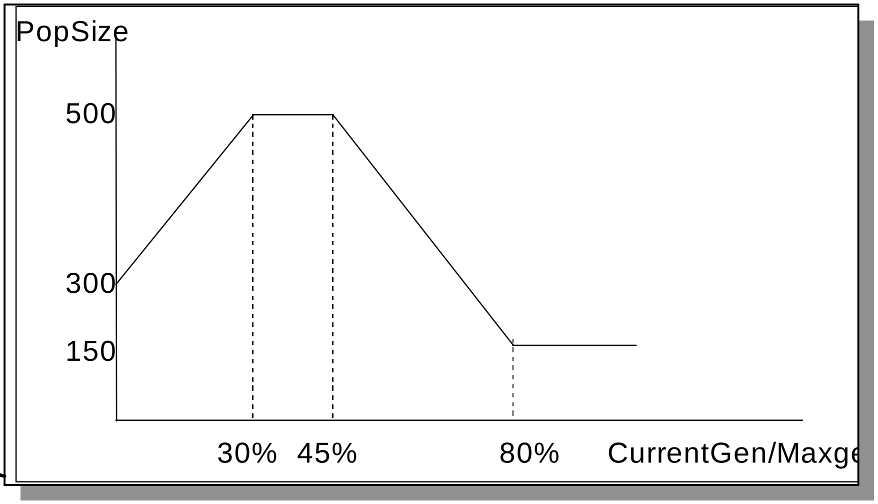
GAs Parameter Setting: Deterministic Control - Example

- **Control of Population size**
  - By decreasing Population Size toward the last part of the Evolution we are trying to improve the solution refinement (e.g., more generations with same number of trials)



Constant Population size: N = 338

Number of trials = 338 * MaxGen

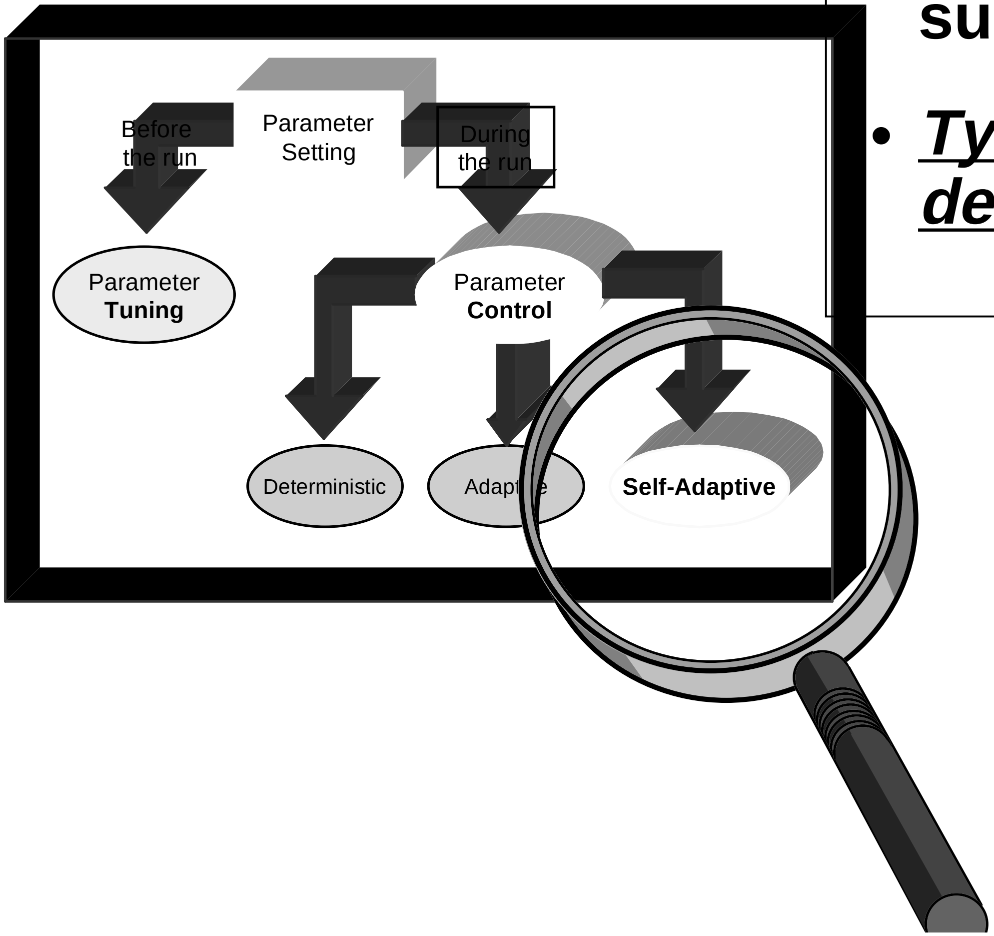


Variable Population size: N(t)

Number of trials = 338 * MaxGen

GAs Parameter Setting: Self-Adaptive Control

- ***Incorporate parameters into chromosome*** **making them subject to evolution**
- ***Typically used by ES to determine Mutation Step S:***



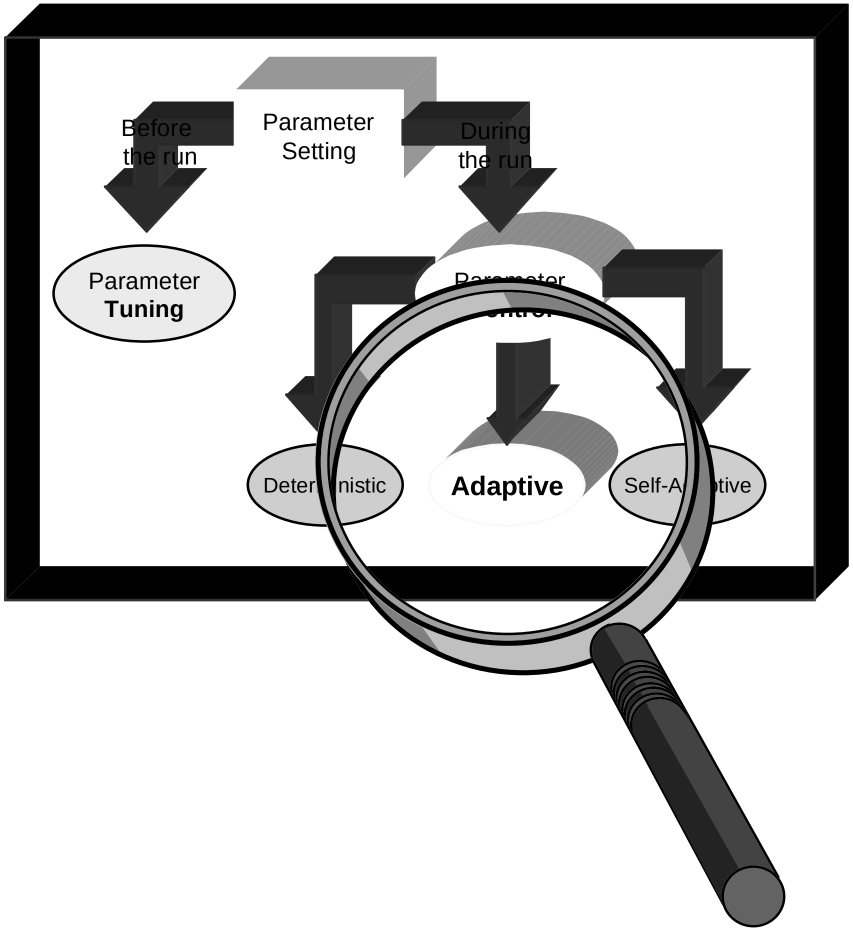
Mutation Step for Entire Genome

$$[g_1\, g_2 \dots g_n\, S]$$

or

$$[g_1\, g_2 \dots g_n\, \underbrace{S_1\, S_2 \dots S_n}_{\text{Mutation Steps for Each Genome Value}}]$$

GAs Parameter Setting: Adaptive Control



- ***Feedback*** **from the search is used to determine the direction and/or magnitude of the change in the parameter value.**

- ***A Fuzzy Logic Controller***

  is used to obtain parameter changes in

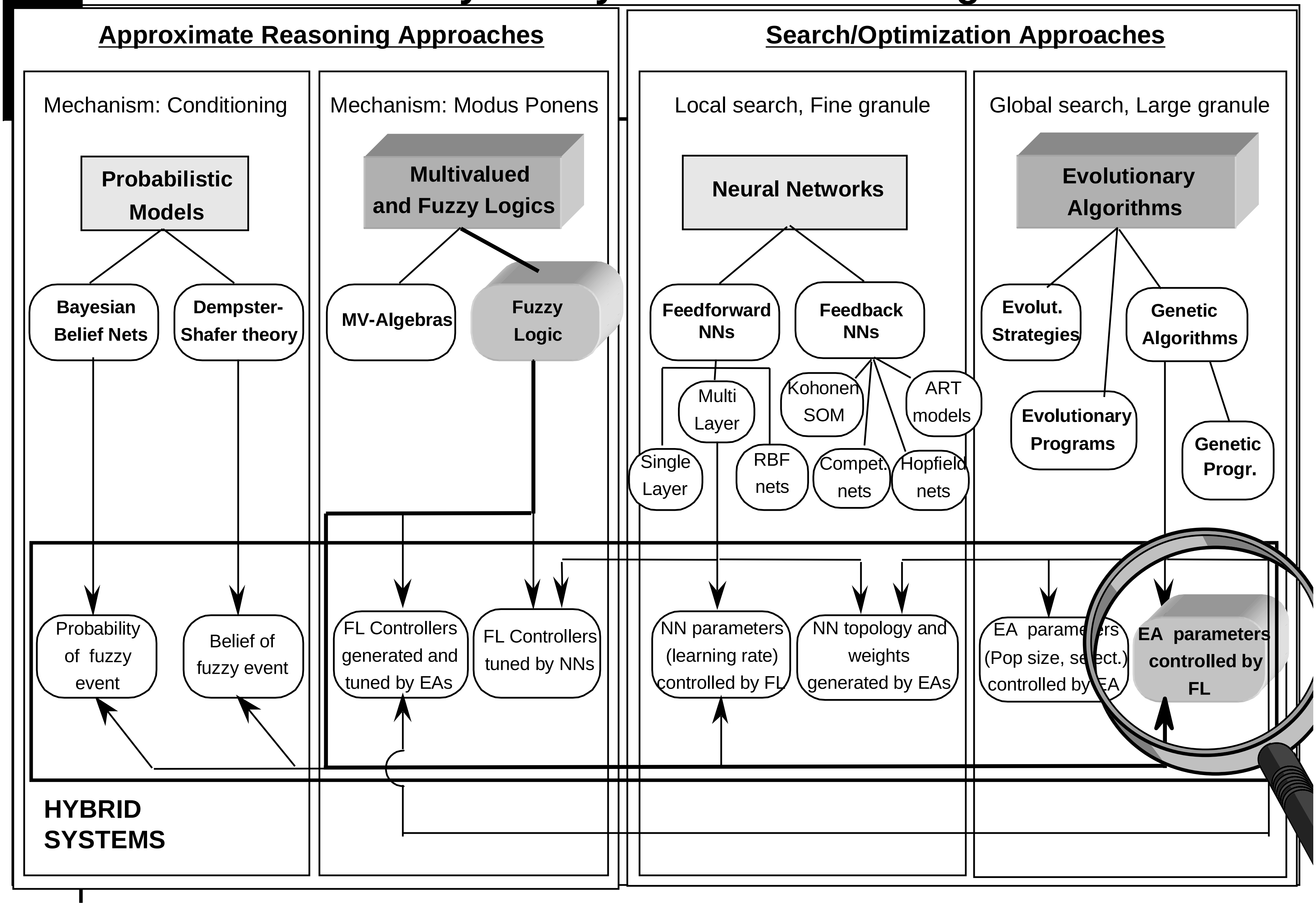
  - Population Size
  - Mutation Rate

  as a function of

  - Genotypic Diversity
  - Percentage Completed Trials

Soft Computing: Evolutionary Comp.: GA parameters control

SC Hybrid Systems: FLC Tuning EA



Approximate Reasoning Approaches

Mechanism: Conditioning

Probabilistic Models

Bayesian Belief Nets

Dempster-Shafer theory

Mechanism: Modus Ponens

Multivalued and Fuzzy Logics

MV-Algebras

Fuzzy Logic

Search/Optimization Approaches

Local search, Fine granule

Neural Networks

Feedforward NNs

Feedback NNs

Multi Layer

Single Layer

RBF nets

Kohonen SOM

ART models

Compet. nets

Hopfield nets

Global search, Large granule

Evolutionary Algorithms

Evolut. Strategies

Genetic Algorithms

Evolutionary Programs

Genetic Progr.

Probability of fuzzy event

Belief of fuzzy event

FL Controllers generated and tuned by EAs

FL Controllers tuned by NNs

NN parameters (learning rate) controlled by FL

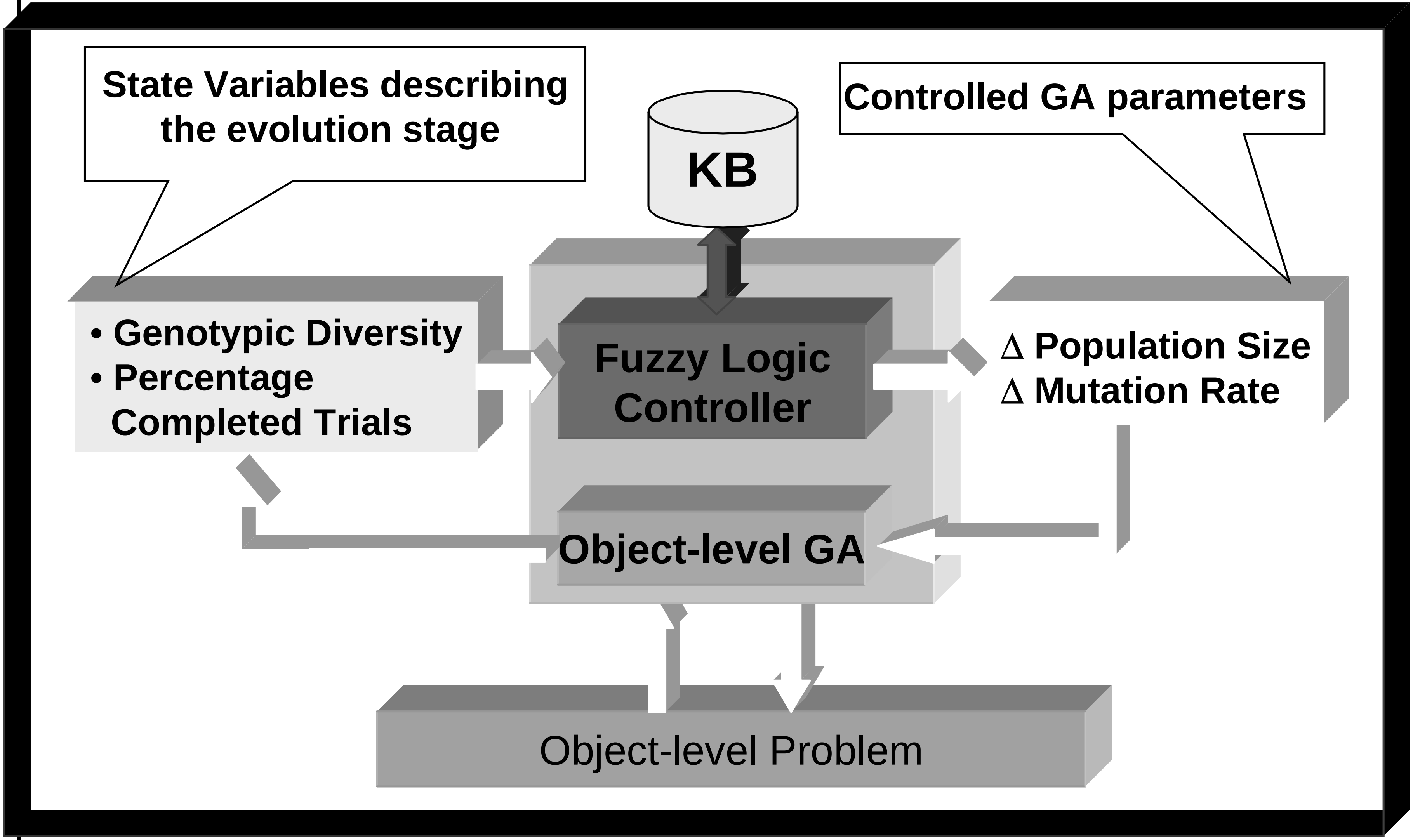
NN topology and weights generated by EAs

EA parame ers (Pop size, se ect.) controlled by EA

EA parameters controlled by FL

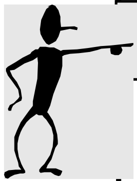
HYBRID SYSTEMS

Fuzzy Logic Controlled GA (FLC-GA)



GA Parameter Setting - Outline

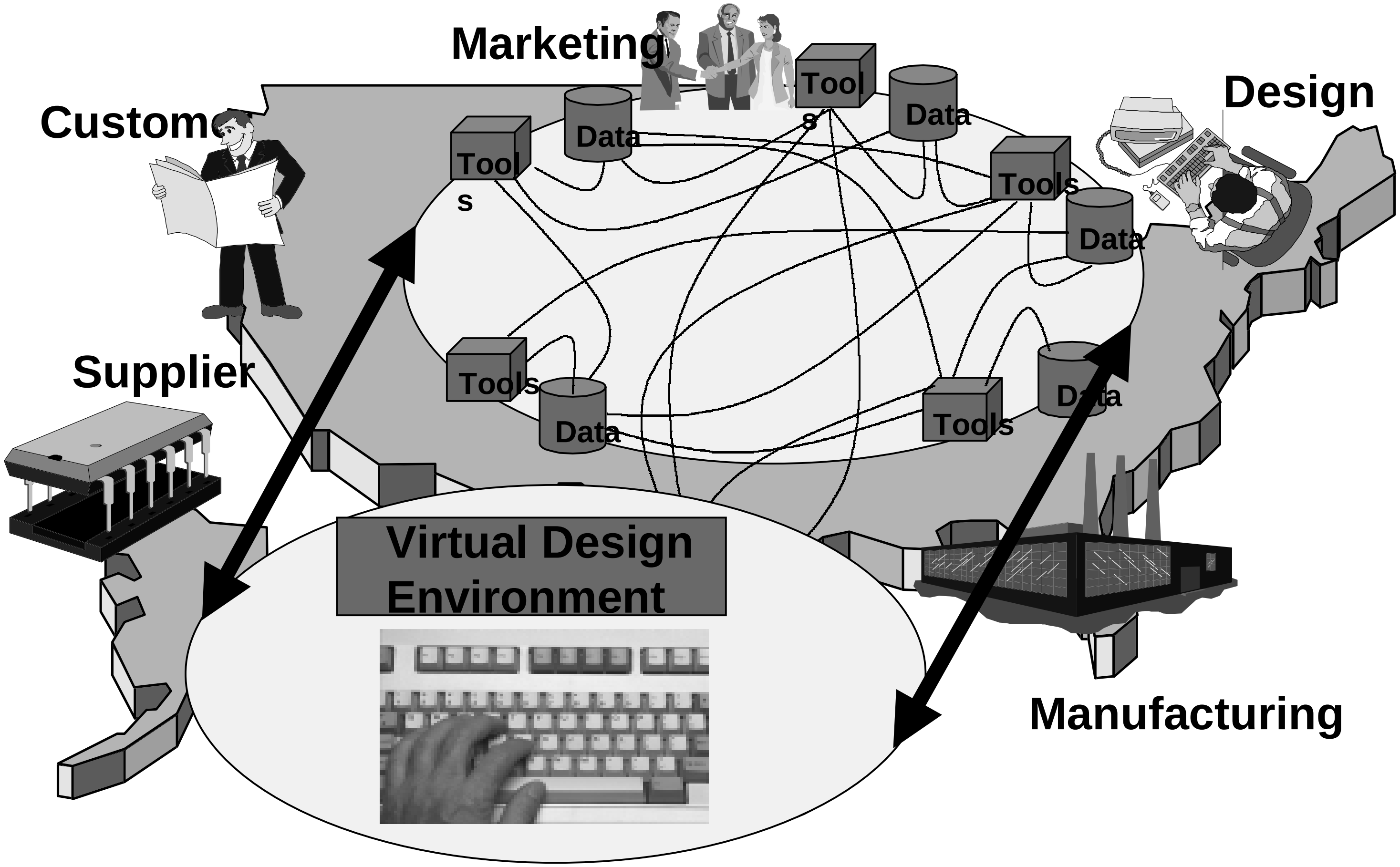
- GA Model: Structure, Parameters
- GA Parameter Setting
    - GA Parameter Tuning: Off-line study, Off-line Meta-GA
    - GA Parameter Control: Deterministic Control, Self-Adaptation, Control (FLC)



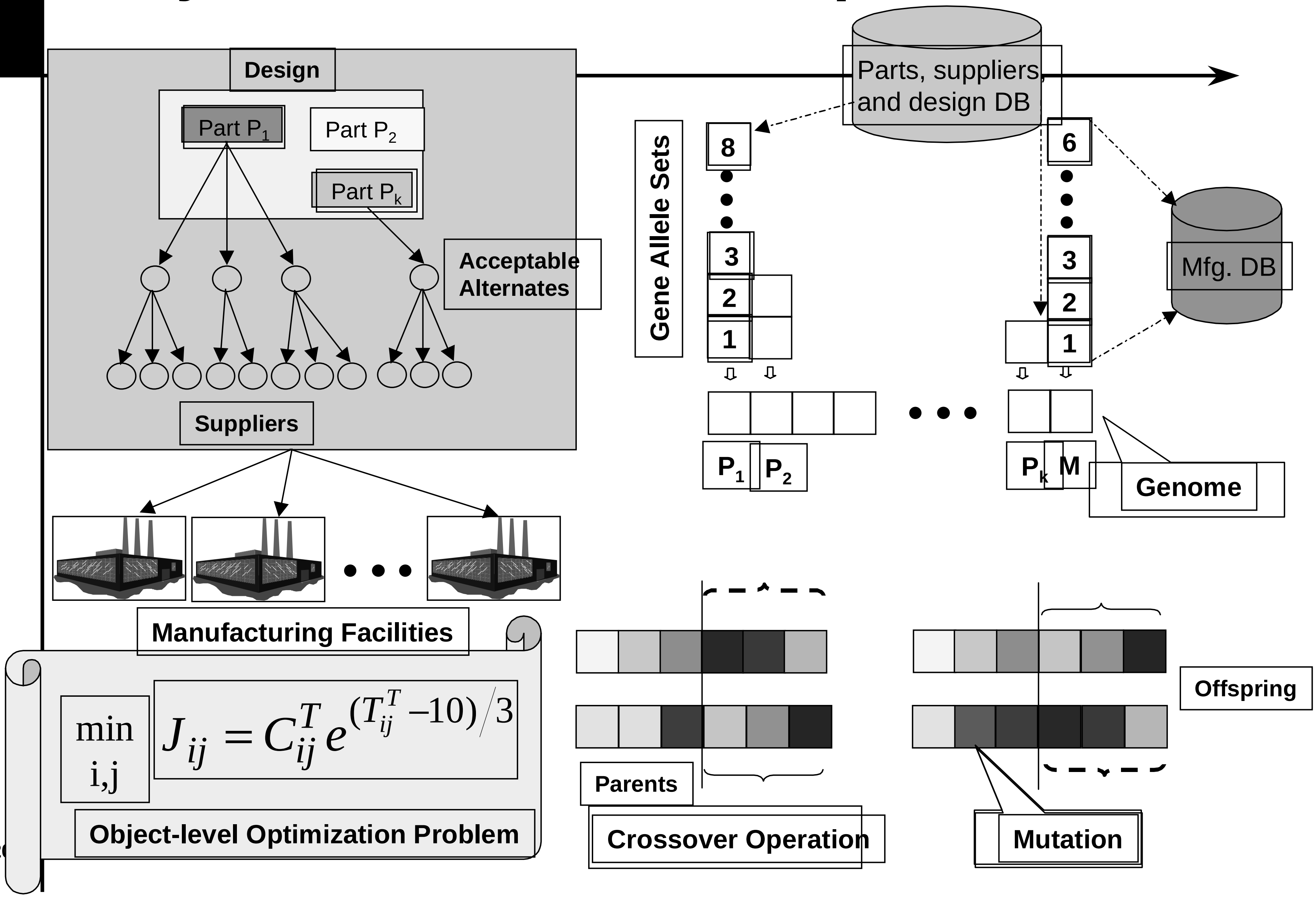
- GA Parameter Setting: An Application
    - Object-level Representation and Complexity
- First Solution Architecture (U-TGA, TGA, FLC-GA)
    - FLC-GA: KB, I/O, Control Parameters, Rule base and Scaling Factors
    - Results - Performance Tradeoff
    - Problems with First Solution
    - Summary and Analysis of 30 Experiments
    - Addressing Problems with First Solution
- Second Solution
    - FLC: KB, Inputs, Outputs, Termsets, Rule Sets, Control Parameters
    - Statistical Experiments: Set-Up & Measures
    - Summary and Analysis of 1200 Experiments
- Conclusions and Next Steps

GA Parameter Setting: An Application

Global optimization of design, manufacturing, supplier planning decisions in a distributed manufacturing environment



Object-level Problem Representation



$$\min_{i,j} \quad J_{ij} = C_{ij}^{T} e^{(T_{ij}^{T} - 10)/3}$$

Object-level Problem Complexity

Search Space Size

- **For off-line parameter tuning and system identification (Data Set 1): O(10^7)**

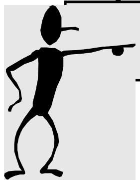
- **For GA performance validation: O(10^{18}) and O(10^{21})**

Computer (to evaluate run-time performance)

- SUN ULTRA SPARC-1

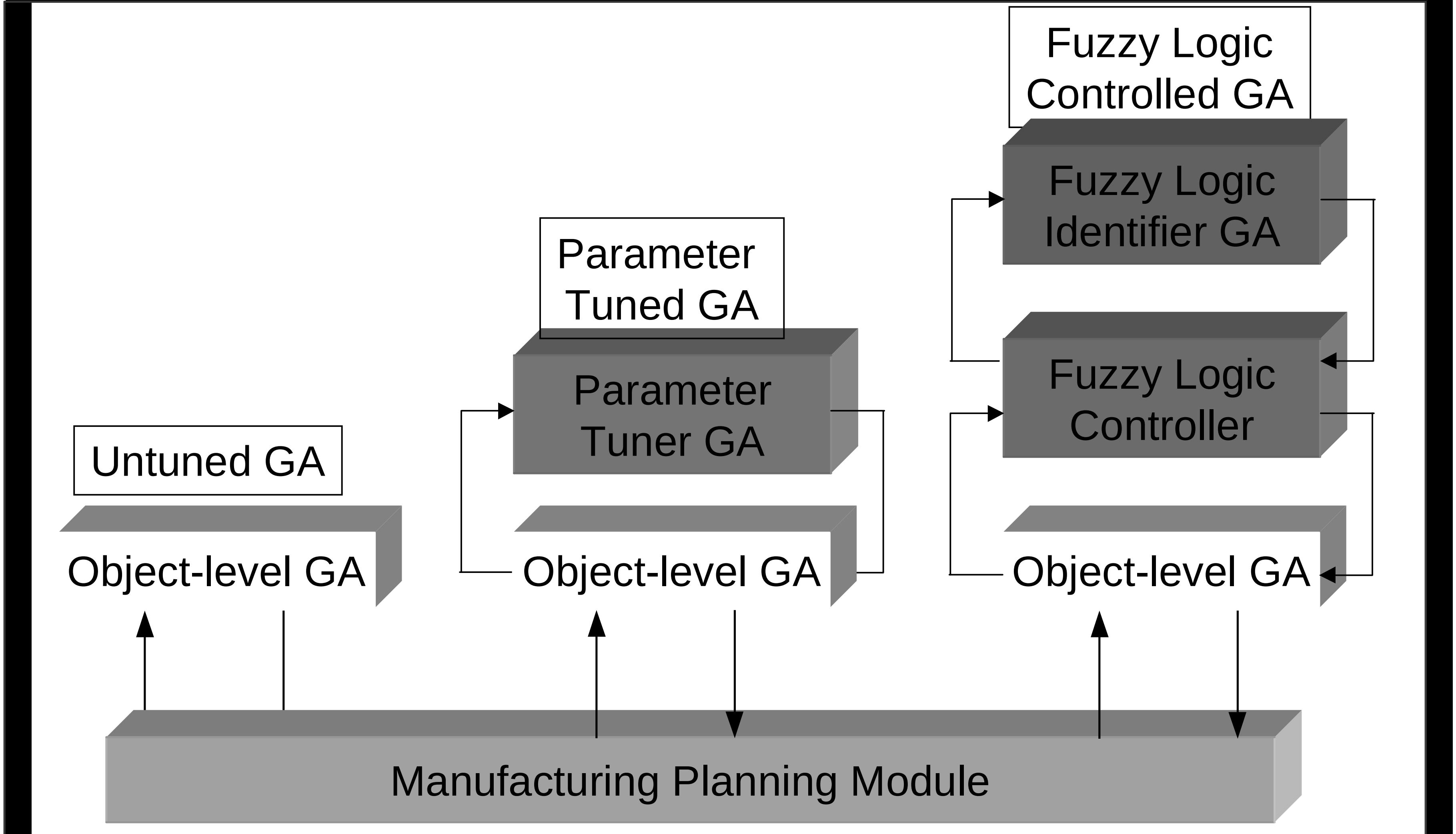
GA Parameter Setting - Outline

- GA Model: Structure, Parameters
- GA Parameter Setting
  - GA Parameter Tuning: Off-line study, Off-line Meta-GA
  - GA Parameter Control: Deterministic Control, Self-Adaptation, Control (FLC)
- GA Parameter Setting: An Application
  - Object-level Representation and Complexity

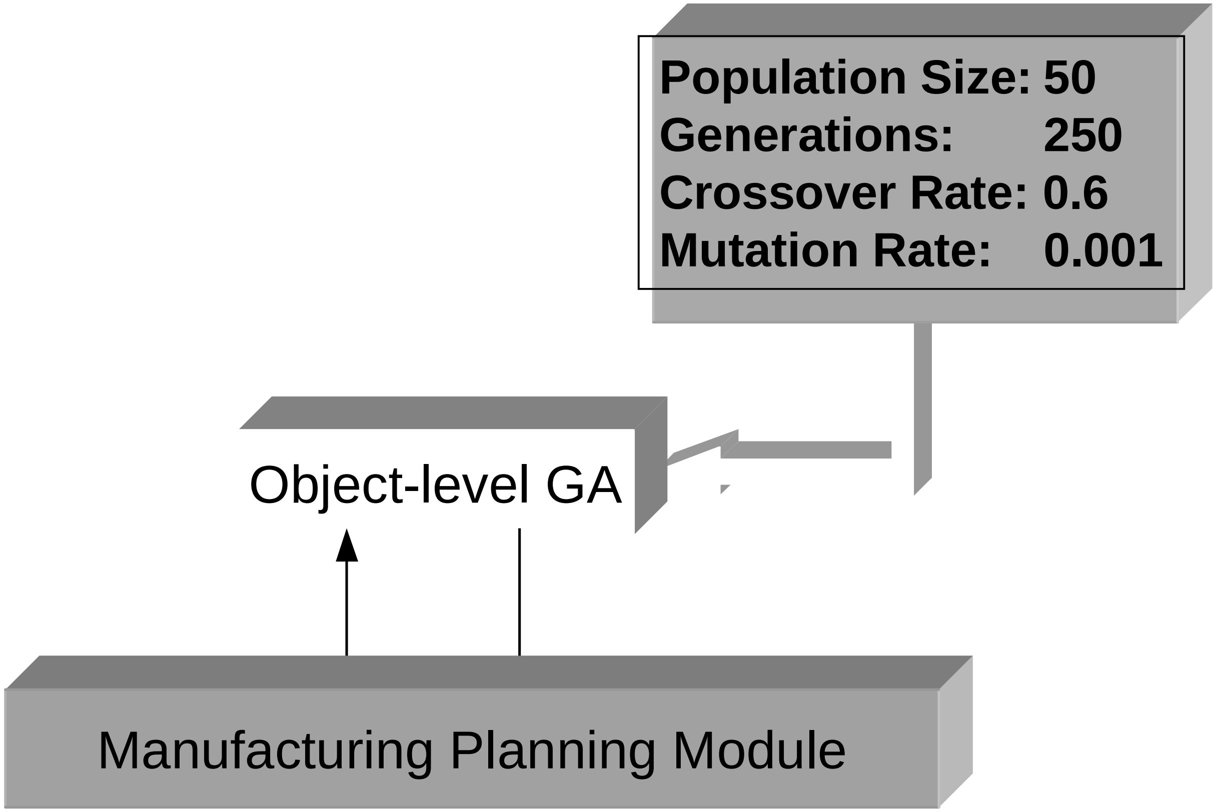


- First Solution Architecture (U-TGA, TGA, FLC-GA)
  - FLC-GA: KB, I/O, Control Parameters, Rule base and Scaling Factors
  - Results - Performance Tradeoff
  - Problems with First Solution
  - Summary and Analysis of 30 Experiments
  - Addressing Problems with First Solution
- Second Solution
  - FLC: KB, Inputs, Outputs, Termsets, Rule Sets, Control Parameters
  - Statistical Experiments: Set-Up & Measures
  - Summary and Analysis of 1200 Experiments
- Conclusions and Next Steps

First Solution: Architecture

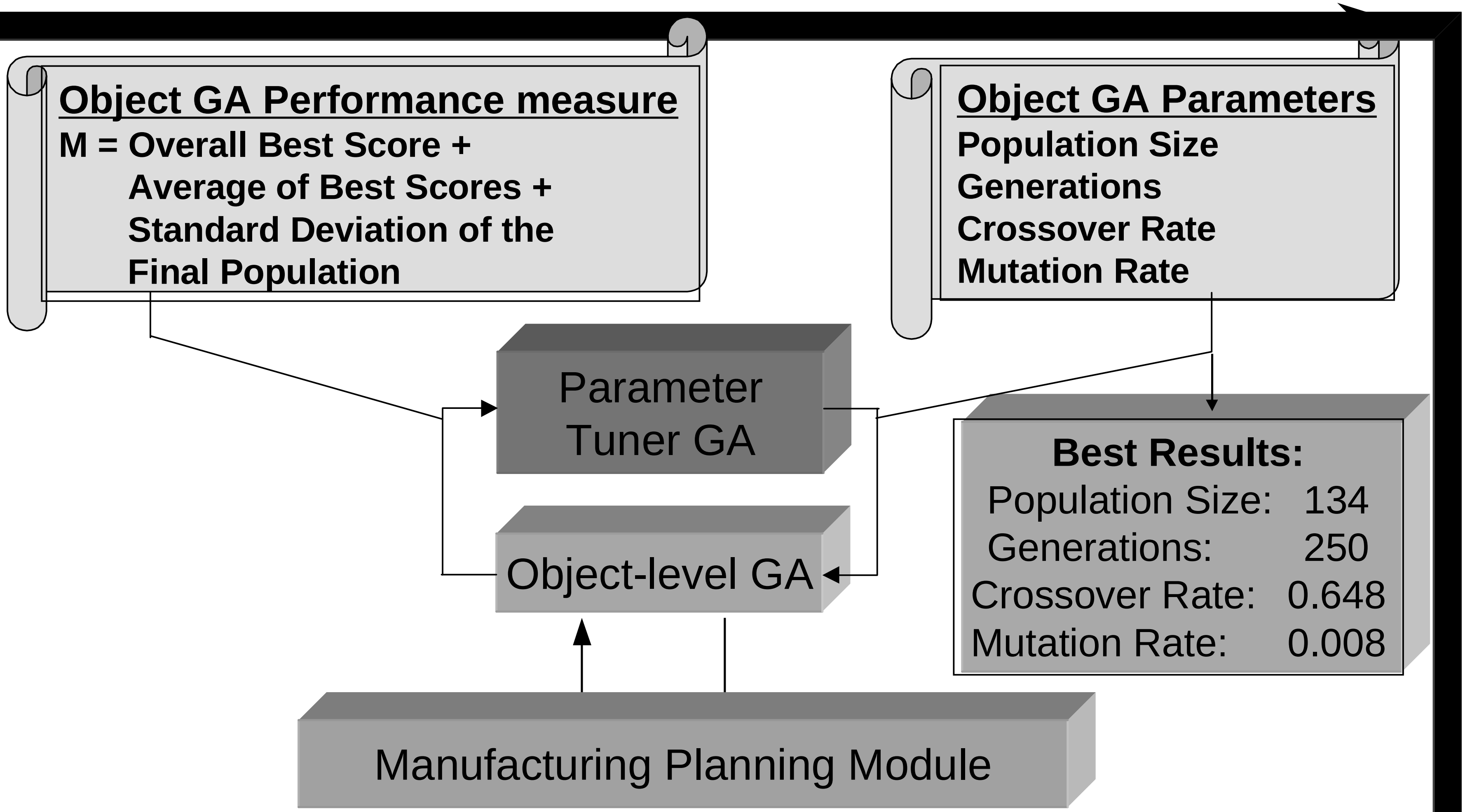


Untuned GA (U-TGA)



Population Size: 50
Generations: 250
Crossover Rate: 0.6
Mutation Rate: 0.001

Parameter Tuned GA (TGA)



Object GA Performance measure

M = Overall Best Score +
Average of Best Scores +
Standard Deviation of the
Final Population

Object GA Parameters

Population Size
Generations
Crossover Rate
Mutation Rate

Best Results:

Population Size:	134
Generations:	250
Crossover Rate:	0.648
Mutation Rate:	0.008

Fuzzy Controller for ΔN, ΔP_c, ΔP_m: Inputs and Outputs

Inputs

GD = Genotypic Diversity (Normalized Average Hamming Distance)

$$GD = \frac{2}{n(n-1)} \sum_{i=1,\, j=i+1}^{n} \frac{d_{ij}}{Genome\ Length}$$

where dij is the Hamming Distance

GD range is [0, 1] == [Low, High]

PD = Phenotypic Diversity:

$$PD = \frac{f_b}{f_\mu}$$

f_b = Best Fitness

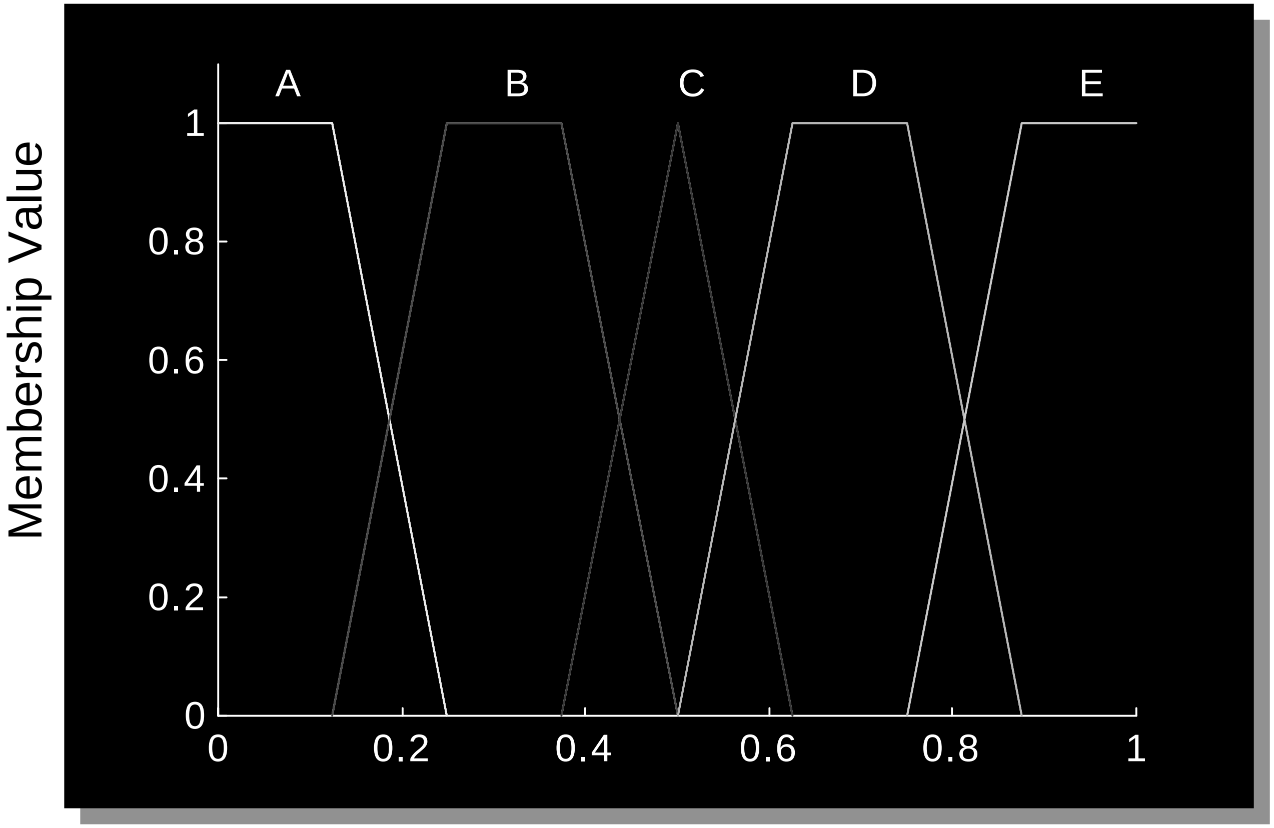
f_μ = Average Fitness

PD range is [0, 1] == [Low, High]

Outputs

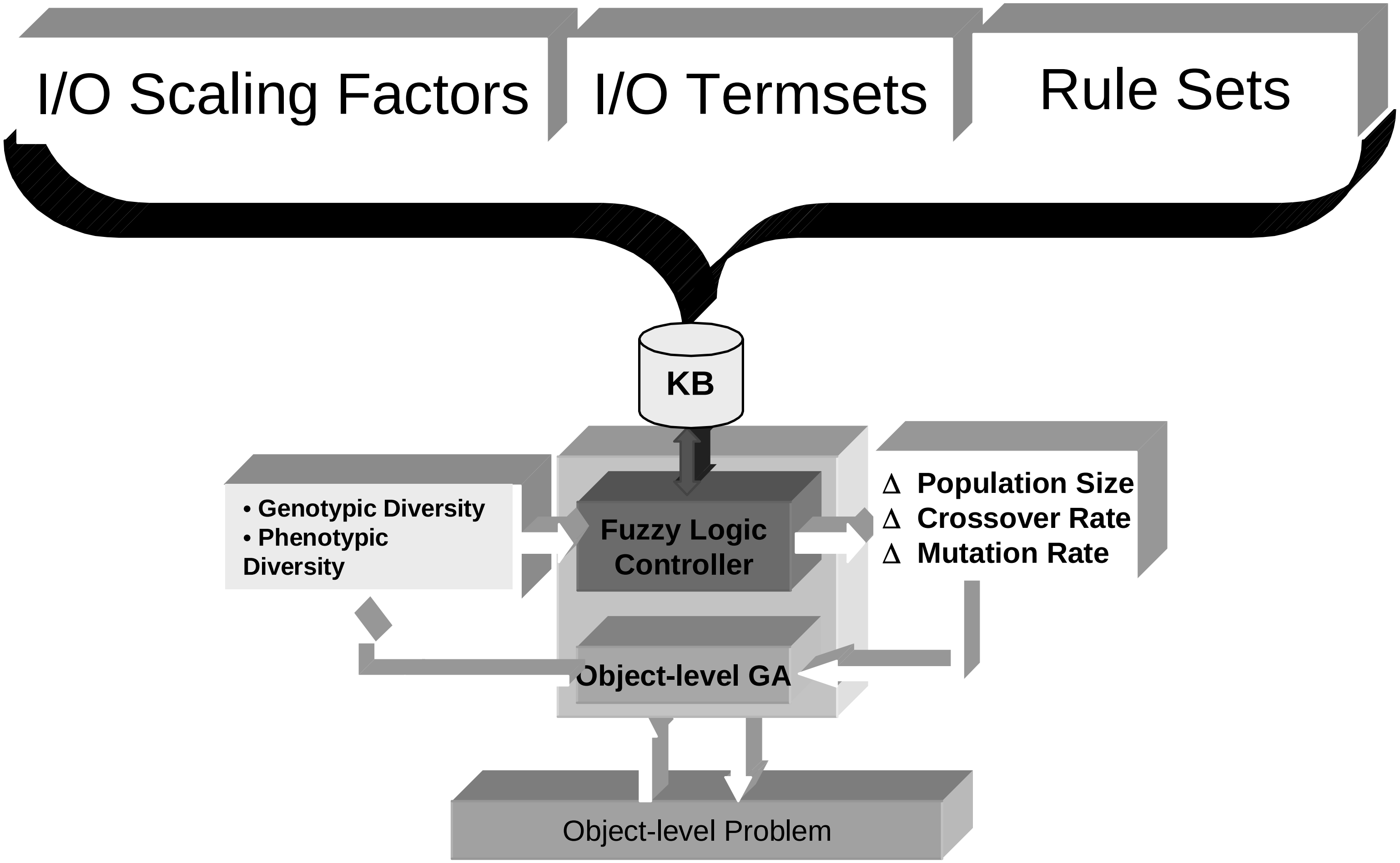
Δ Population Size Δ Crossover Rate Δ Mutation Rate

Termset for State and Control Variables



GD: A(Very Low), B(Low), C(Medium), D(High), E(Very High)
PD: A(Very High), B(High), C(Medium), D(Low), E(Very Low)
Outputs: A(Neg. High), B(Neg. Medium), C(No Change), D(Pos. Medium), E(Pos. High)

Fuzzy Logic Controller for GAs: Knowledge Base



I/O Scaling Factors

I/O Termsets

Rule Sets

KB

- Genotypic Diversity
- Phenotypic Diversity

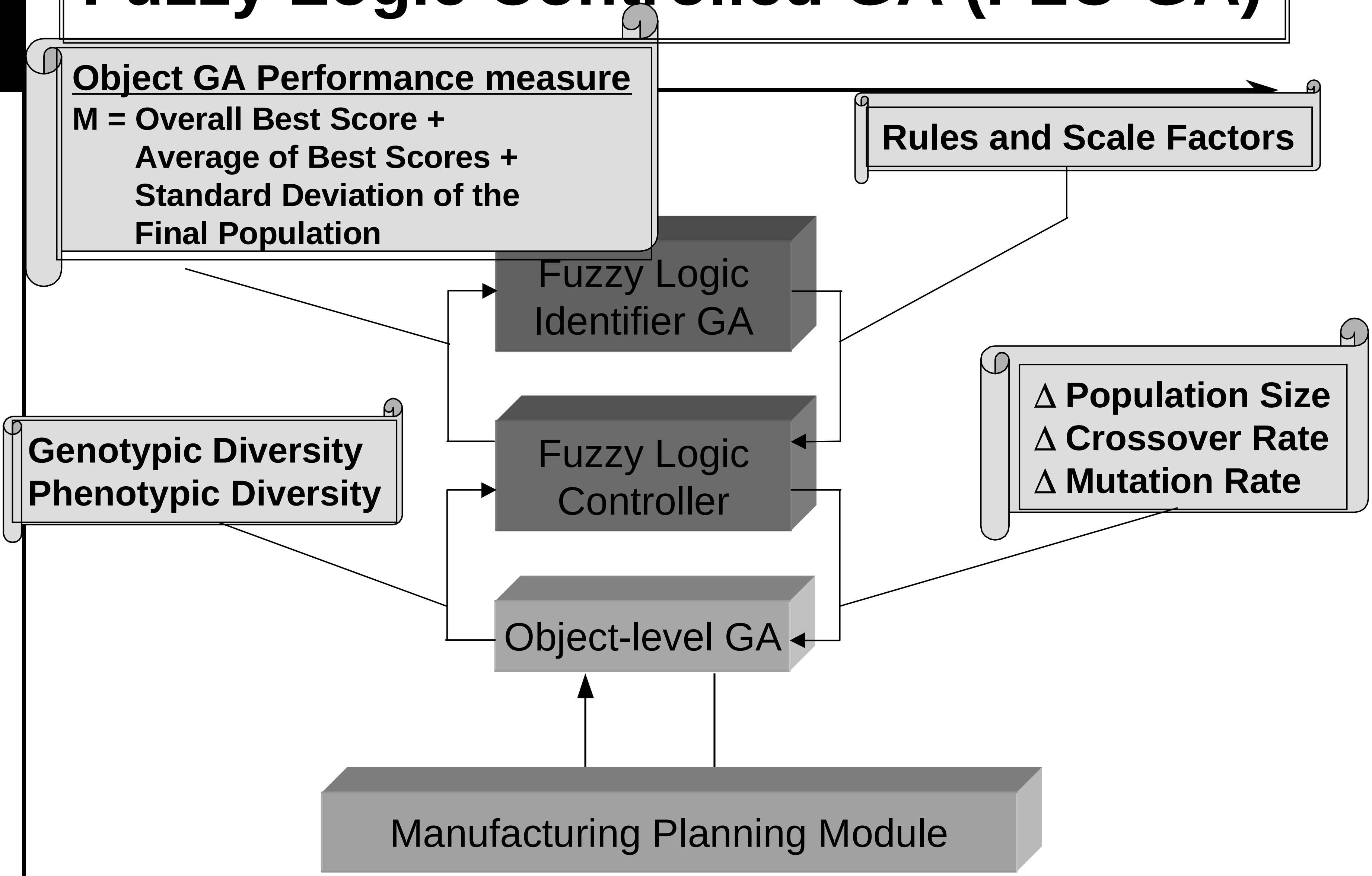
Fuzzy Logic Controller

Δ Population Size
Δ Crossover Rate
Δ Mutation Rate

Object-level GA

Object-level Problem

Fuzzy Logic Controlled GA (FLC-GA)



Object GA Performance measure

M = Overall Best Score +
Average of Best Scores +
Standard Deviation of the
Final Population

Rules and Scale Factors

Genotypic Diversity
Phenotypic Diversity

Δ Population Size
Δ Crossover Rate
Δ Mutation Rate

Fuzzy Rule Base and Scaling Factors

Phenotypic Diversity

Genotypic Diversity

	VH	H	M	L	VL
VL	NH	PM	NM	NC	PM
L	NH	PM	NM	NM	PM
M	NH	PM	PH	PM	PH
H	PM	PH	PM	NH	NM
VH	PH	NM	PH	PH	NC

Δ Population Size

Phenotypic Diversity

Genotypic Diversity

	VH	H	M	L	VL
VL	NC	PM	NC	NH	NM
L	NC	PH	NC	NH	PM
M	PM	PM	PH	PH	NM
H	NC	PH	NC	NC	PM
VH	PH	PM	PM	PM	NC

Δ Crossover Rate

Phenotypic Diversity

Genotypic Diversity

	VH	H	M	L	VL
VL	NH	PH	NM	PH	PH
L	NM	PM	PH	NH	NM
M	NC	PH	NH	NM	PH
H	NM	NH	NC	NC	NH
VH	NC	NH	PM	PH	NM

Δ Mutation Rate

NH	NM	NC	PM	PH

Output Terms Adjacency Key

Scale Factors

Δ Pop. Size = [0.1, 1.9]
Δ Cross. Rate = [0.6, 1.4]
Δ Mut. Rate = [0.4, 1.6]

Fuzzy Controller for ΔN, ΔP_c, ΔP_m : Control Parameters

Frequency of Control Actions

Control Action applied every 5 generations

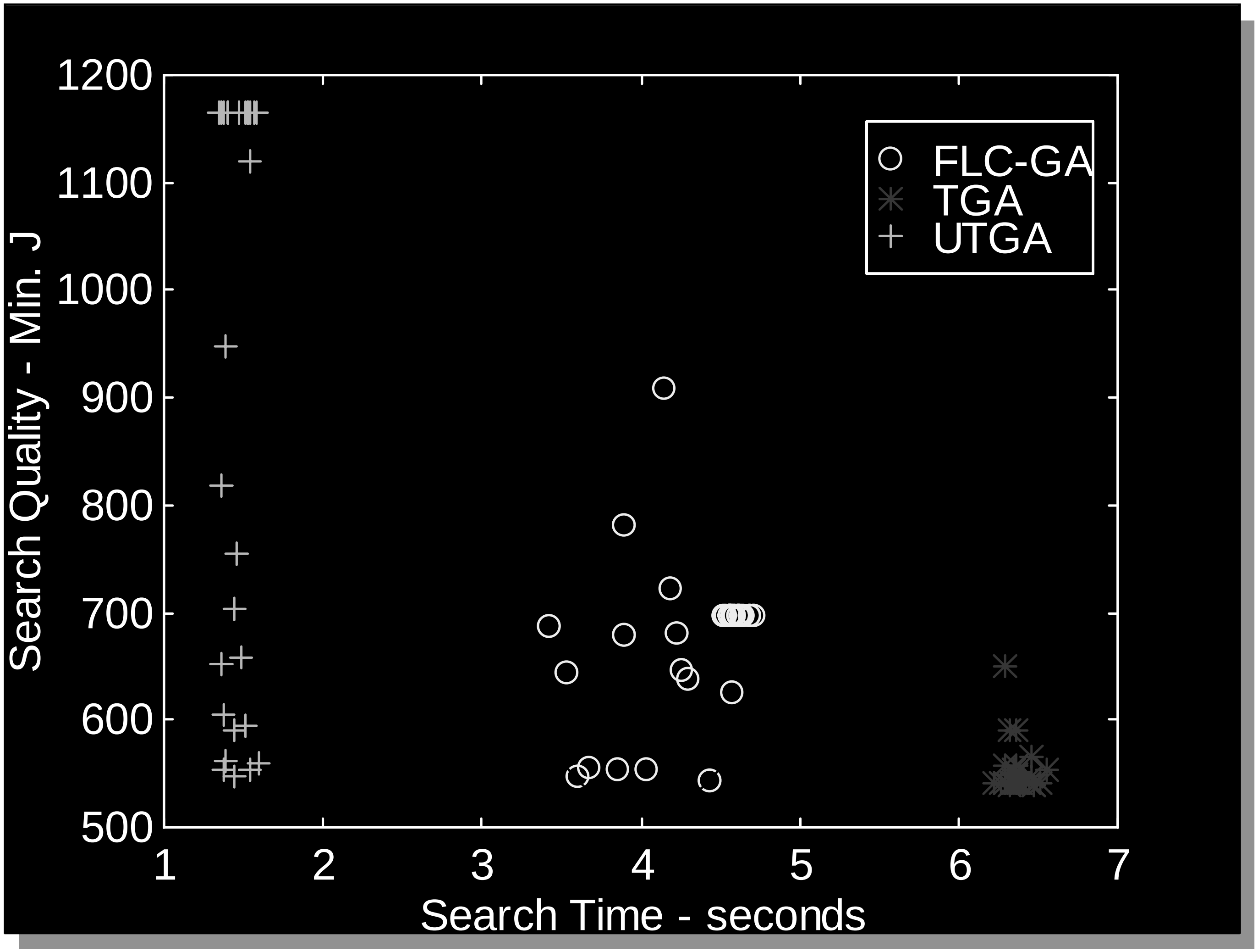
Mutation Rate

Mutation rates drops exponentially after a control action that increases it

Inference Engine Parameters

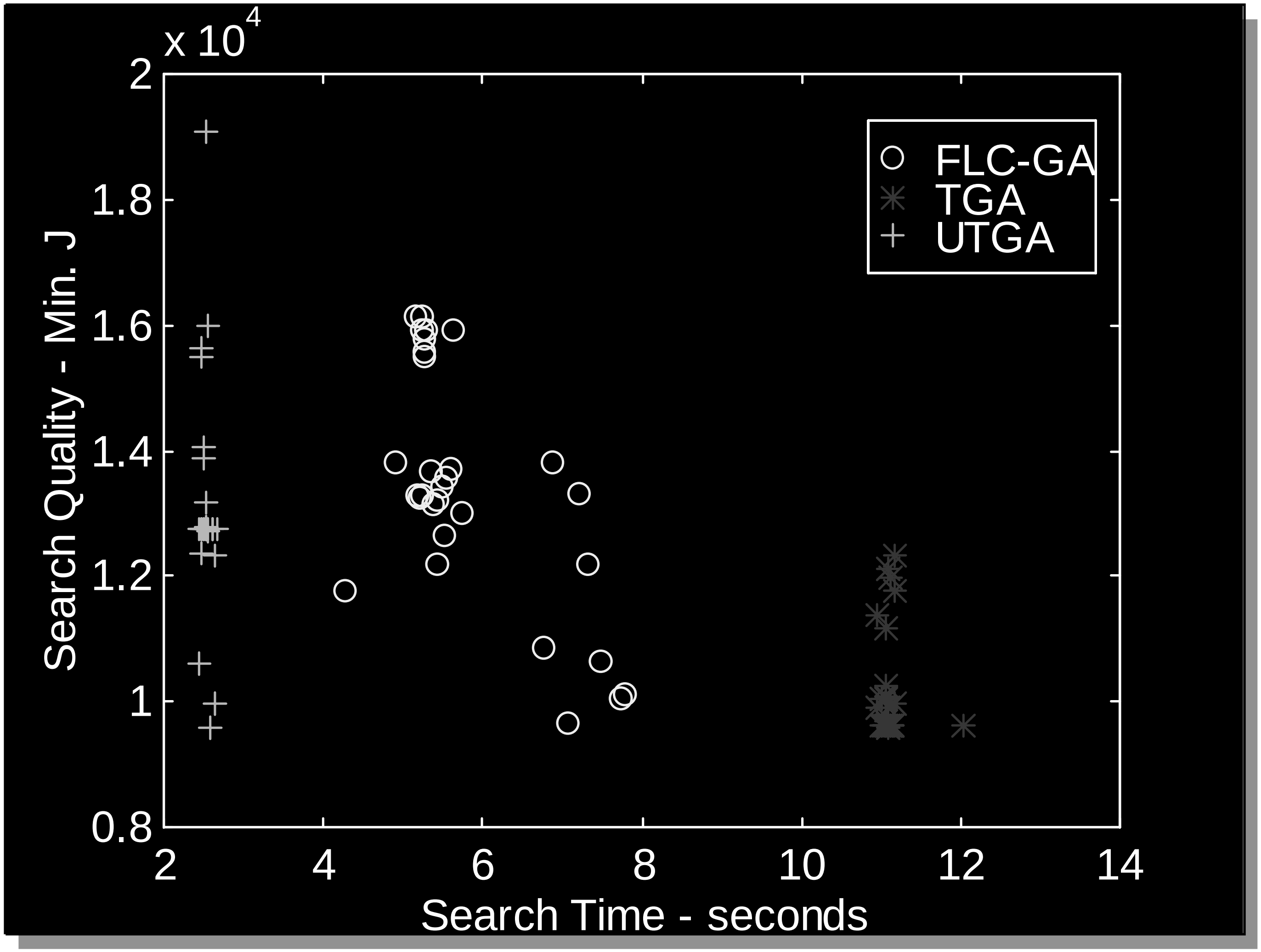
Left Hand Side (LHS) evaluation:	*Minimum operator*
Rule Firing:	*Minimum operator*
Rule Output Aggregation:	*Maximum operator*
Defuzzification:	*Center of Gravity (COG)*

Results - Performance Tradeoff - O(10^7)



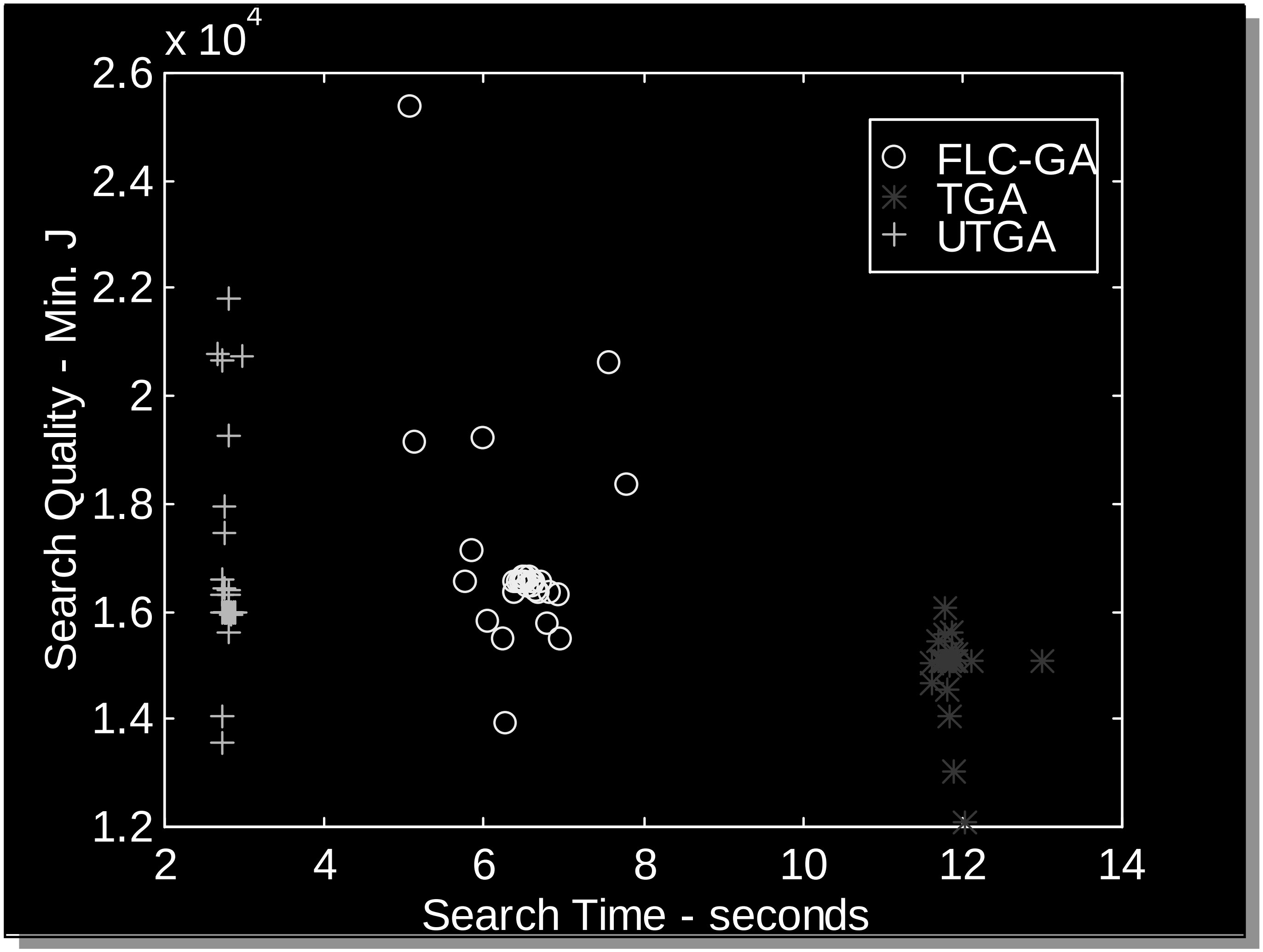
Benchmarked on search space of size O(10^7)

Results - Performance Tradeoff - O(10^{18})



Benchmarked on search space of size O(10^{18})

Results - Performance Tradeoff - O(10^{21})



Benchmarked on search space of size O(10^{21})

Problems with First Solution

FLC design parameter space:

- Rule Set:

75 output rules - each rule can have one out of five values

Complexity (possibly different # rules): $5^{75} \cong$ **2.64 x10^{52}**

- Scaling Factors:

9 scaling factors - each one can have one out of nine values

Complexity (possibly different # scaling factors): $9^{9} \cong$ **3.87 x10^{8}**

- Combining Complexity:

5^{75} X $9^{9} \cong$ **1.02 x10^{61}**

Larger than the object-level problem complexity!

Summary of 30 Experiments

Data Sets	GA		Min J		Time	
[Complexity]	Strategy	μ	σ	σ/μ	μ	σ
1	FLC-GA	674	74	0.110	4.30	0.39
O(10^7)	TGA	551	23	0.042	6.37	0.08
	UTGA	923	272	0.295	1.45	0.08
2	FLC-GA	13380	1911	0.143	5.83	0.95
O(10^{18})	TGA	10154	871	0.086	11.12	0.18
	UTGA	13048	1761	0.135	2.53	0.06
3	FLC-GA	17001	2006	0.118	6.46	0.56
O(10^{21})	TGA	14939	738	0.049	11.86	0.24
	UTGA	16793	1946	0.116	2.78	0.06

Analysis of Results

- ***Offline computation time - using Data set 1 - for:***
  - FLC rule set and scaling factors: 14 hrs
  - Meta-GA: 2 hrs
- ***Statically tuned Meta-GA (TGA):***
  - slowest one but has the best search quality performance (J)
- ***Untuned GA (UTGA):***
  - is the fastest but has the lowest search quality
- ***FLC controlled GA (FLC-GA) :***
  - has large spread of search quality but it is faster than TGA

Identified problems with FLC-GA approach that need to be solved

Addressing the Problems with First Solution

- ***Minimize high-level search space size for FLC-GA by***

  - Identify primary drivers (influences) of GA search
  (use DOE to identify main effects and control only those drivers, minimizing number of controllers, hence number of parameters)

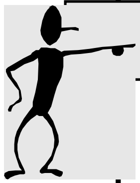
  - Control primary drivers by few simple heuristic rules
  (eliminating learning of rule set and scaling factors)

- ***Optimal FLC firing rate?***
  - Reduction of firing frequency to reduce overhead

- ***Extensive & statistically significant empirical evidence***
  - Let's use something we learned in 6σ !

GA Parameter Setting - Outline

- GA Model: Structure, Parameters
- GA Parameter Setting
  - GA Parameter Tuning: Off-line study, Off-line Meta-GA
  - GA Parameter Control: Deterministic Control, Self-Adaptation, Control (FLC)
- GA Parameter Setting: An Application
  - Object-level Representation and Complexity
- First Solution Architecture (U-TGA, TGA, FLC-GA)
  - FLC-GA: KB, I/O, Control Parameters, Rule base and Scaling Factors
  - Results - Performance Tradeoff
  - Problems with First Solution
  - Summary and Analysis of 30 Experiments
  - Addressing Problems with First Solution



- Second Solution
  - FLC: KB, Inputs, Outputs, Termsets, Rule Sets, Control Parameters
  - Statistical Experiments: Set-Up & Measures
  - Summary and Analysis of 1200 Experiments
- Conclusions and Next Steps

Second Solution

- ***Minimize high-level search space size for FLC-GA by***

  - Identify primary drivers (influences) of GA search

  DOE determined that the two main drivers were:

  Population Size (N) and Mutation Rate (P_m)

  - Control primary drivers by few simple heuristic rules

  Built two FLC controllers with heuristic rule sets and SF
  Changed on input (state variable) to capture evolution stage

- ***Optimal FLC firing rate?***

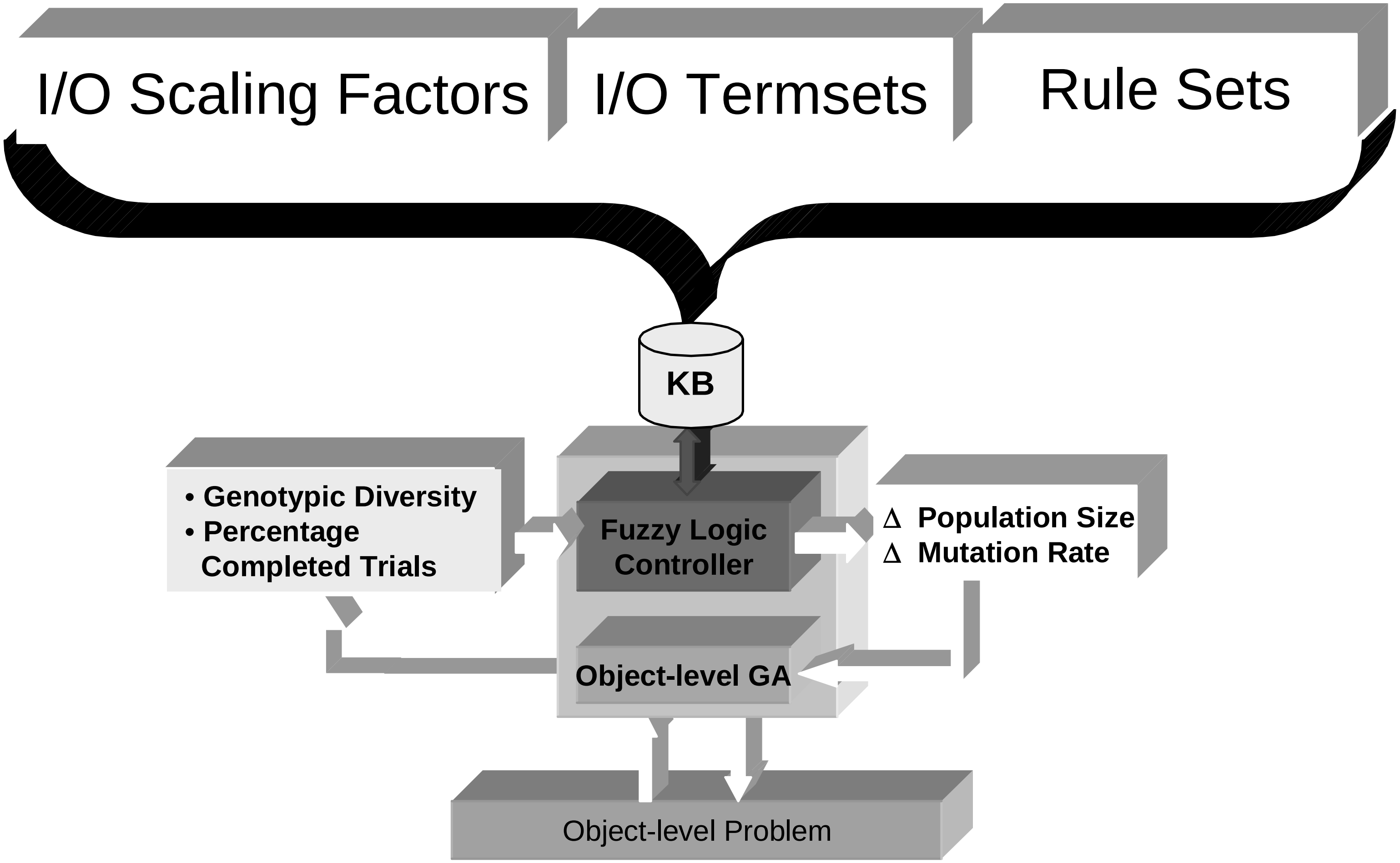
  - Reduction of firing frequency

  From an action every 5 to one every 10 generation

- ***Extensive & statistically significant empirical evidence***

  - Use t-test and F-tests to anlyze μ and σ improvements

Fuzzy Logic Controller for GAs: Knowledge Base



Fuzzy Controller for ΔN and ΔP_m: Inputs

Inputs

GD = Genotypic Diversity:

Normalized Average Hamming Distance

$$GD = \frac{2}{n(n-1)} \sum_{i=1,\, j=i+1}^{n} \frac{d_{ij}}{Genome\ Length}$$

where dij is the Hamming Distance
GD range is [0, 1] == [Low, High]

PFE = Percentage Fitness Evaluations:

(Completed # Trials) / (Max Allocated # Trials)

PFE range is [0, 1] == [Low, High]

Fuzzy Controller for ΔN and ΔP_m: Outputs

Outputs

ΔN = Change in Population Size (Mult. Factor)

Δ *N* range is [0.5, 1.5] == [Neg High, Pos High]
so that NC corresponds to 100% of previous Pop Size

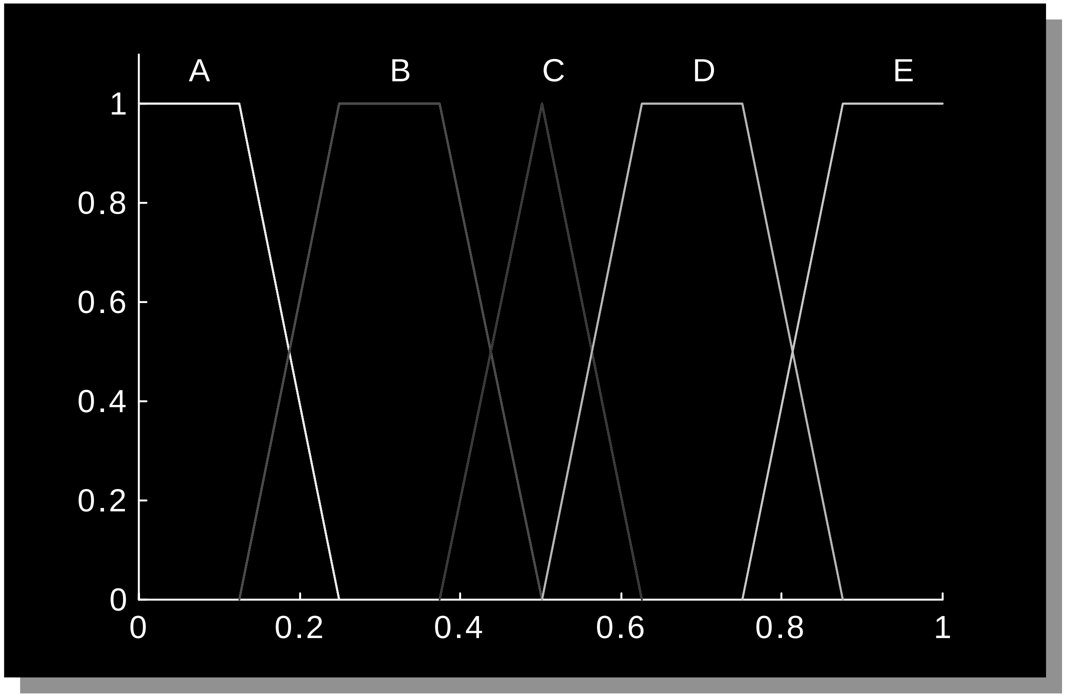
Population Size is *clamped* within [25, 150]

ΔP_m = Change in Mutation Rate (Mult. Factor)

Δ *Pm* range is [0.5, 1.5] == [Neg High, Pos High]
so that NC corresponds to 100% of previous Pm

Mutation Rate is *clamped* within [0.005, 0.10]

Fuzzy Controller for ΔN and ΔP_m: Termsets



Inputs:

GD**:** **A(Very Low), B(Low), C(Medium), D(High), E(Very High)**

PFE**:** **A(Very Low), B(Low), C(Medium), D(High), E(Very High)**

Outputs (for both ΔN and ΔP_m):

A(Neg. High), B(Neg. Medium), C(No Change), D(Pos. Medium), E(Pos. High)

Fuzzy Controller for Population Size: Rule Set

GD, PFE–>ΔN

GD = Genotypic Diversity:
Normalized Average Hamming Distance
PFE = Percentage Fitness Evaluations:
(Completed # Trials) / (Max Allocated # Trials)
ΔN = Change in Population Size

Genotypic	**Percentage Fitness Evaluation (PFE)**				
Diversity (GD)	*Very Low*	*Low*	*Medium*	*High*	*Very High*
Very Low	Pos High	Pos High	Pos High	Pos Medium	No Change
Low	Pos High	Pos High	Pos Medium	No Change	Neg Medium
Medium	Pos High	Pos Medium	No Change	Neg Medium	Neg High
High	Pos Medium	No Change	Neg Medium	Neg High	Neg High
Very High	No Change	Neg Medium	Neg High	Neg High	Neg High

Exploration Stage
Increase population/ broaden search

Exploitation Stage
Reduce population/ Refine best

Fuzzy Controller for Mutation Rate: Rule Set

GD, PFE–>ΔP_m

GD = Genotypic Diversity:
Normalized Average Hamming Distance

PFE = Percentage Fitness Evaluations:
(Completed # Trials) / (Max Allocated # Trials)

ΔP_m = Change in Mutation Rate

Genotypic	**Percentage Fitness Evaluation (PFE)**				
Diversity (GD)	*Very Low*	*Low*	*Medium*	*High*	*Very High*
Very Low	Pos High	Pos High	Pos Medium	Pos Medium	No Change
Low	Pos High	Pos Medium	Pos Medium	No Change	No Change
Medium	Pos Medium	Pos Medium	No Change	No Change	No Change
High	Pos Medium	No Change	No Change	No Change	No Change
Very High	No Change	No Change	No Change	No Change	No Change

Fuzzy Controller for ΔN and ΔP_m: Control Parameters

Frequency of Control Actions

Control Action:

- mutation rate changed every 10 generations
- population size change every generation

Mutation Rate

Mutation rates drops exponentially after a control action that increases it

Inference Engine Parameters

Left Hand Side (LHS) evaluation:	*Minimum operator*
Rule Firing:	*Minimum operator*
Rule Output Aggregation:	*Maximum operator*
Defuzzification:	*Center of Gravity (COG)*

Statistical Experiments: Set-Up

Data Set for Experiments

- Seven part classes corresponding to a complexity of O(10^7)

GA Structure:

- Type: {Simple, Steady-State}
- Chromosome Encoding: Integer
- Fitness Function: Three type of cost functions
- Selection Method: Proportional Roulette
- Crossover Operator: Uniform
- Mutation Operator: Exponentially Decreasing

GA Parameters

- N = Base Population size: 50
- P_c = Crossover rate: 0.6
- P_m = Mutation rate: 0.005
- G = Generation Gap 100% replac. - Simple GA (SGA)
  25% replac. - Steady State GA (SSGA)
- **S** = Selection Strategy: Elitist

Statistical Experiments: Set-Up

Set-Up for 1200 experiments:

- We defined 4 GA configurations
  - (a) Untuned Simple GA (U-SGA)
  - (b) FL Controlled Simple GA (FLC-SGA)
  - (c) Untuned Steady State GA (U-SSGA)
  - (d) FL Controlled Steady State GA (FLC-SSGA)
- For each configuration we performed 300 experiments:
  - 20 runs for each pair of (Cost function, Max number of Trials)
  - 15 different pairs of (Cost function, Max number of Trials)
    - Three types of *cost functions*:

(1) $J = C*T$; (2) $J = C*T^2$; (3) $J = C*e^{(T-10)/3}$

    - Five values of maximum *number of Trials* (to evaluate effect of control for different evolution lengths:

(i) 3,000; (ii) 5,000 (iii) 7,000 (iv) 9,000 (v) 11,000

Statistical Experiments: Measures

- **For each of the four configuration (a-d) we ran 20 experiments with the same parameters**
- **Then we considered the following measures:**
  - $\hat{B}$ = sample average *over 20 experiments* of Best score frequency (number of time cost function J reached its minimal value - known a priori for small size experiment)
  - $\hat{\mu}$ = average of population best
  - $\hat{\sigma}$ = standard deviation of population best
- **We performed an ANOVA test (using both t and F test -**using the usual value of $p < 0.05$ to identify statistical significance**) to see how many times the following statements were true:**
  - Cost (U-SGA) significantly higher than cost (FLC-SGA)
  - Cost (U-SGA) significantly higher than cost (U-SSGA)
  - Cost (U-SSGA) significantly higher than cost (FLC-SSGA)
- **Finally we verified if the FLC caused the GA to perform any worse than its corresponding untuned GA, i.e.:**
  - Cost (U-SGA) significantly lower than cost (FLC-SGA)
  - Cost (U-SSGA) significantly lower than cost (FLC-SSGA)

Summary of 1200 Experiments

J = C*T

Max Numb		U-SGA			FLC-SGA				U-SSGA				FLC-SSGA			
Trials	B	μ	σ	σ/μ	B	μ	σ	σ/μ	B	μ	σ	σ/μ	B	μ	σ	σ/μ
3000	0%	1788.8	71	0.040	20%	1729	81	0.047	70%	1685	63	0.037	80%	1677	58	0.034
5000	5%	1767.9	103	0.058	35%	1705	74	0.043	75%	1682	63	0.037	80%	1673	47	0.028
7000	35%	1710.3	81	0.047	45%	1680	41	0.025	60%	1739	108	0.062	95%	1665	45	0.027
9000	20%	1748.8	102	0.058	50%	1676	46	0.027	80%	1695	82	0.048	85%	1689	70	0.041
11000	50%	1719.5	88	0.051	75%	1668	40	0.024	75%	1709	98	0.058	95%	1665	45	0.027

J = C*T^2

Max Numb		U-SGA			FLC-SGA				U-SSGA				FLC-SSGA			
Trials	B	μ	σ	σ/μ	B	μ	σ	σ/μ	B	μ	σ	σ/μ	B	μ	σ	σ/μ
3000	5%	352.5	20.9	0.059	15%	341.3	12.0	0.035	50%	343.8	23.0	0.067	75%	332.4	4.6	0.014
5000	30%	338.6	8.0	0.024	30%	337.7	7.9	0.023	60%	336.2	11.6	0.034	85%	331.8	4.1	0.012
7000	20%	339.7	1.9	0.005	30%	338.8	1.7	0.005	70%	341.3	5.2	0.015	70%	336.3	3.4	0.010
9000	30%	343.2	15.74	0.046	50%	333.9	5.0	0.015	80%	335.2	15.3	0.046	60%	334.6	5.6	0.017
11000	65%	337.0	15.3	0.045	60%	331.7	3.5	0.010	65%	336.9	15.3	0.045	65%	336.9	15.3	0.045

J = C*e$^{(T-10)/3}$

Max Numb		U-SGA			FLC-SGA				U-SSGA				FLC-SSGA			
Trials	B	μ	σ	σ/μ	B	μ	σ	σ/μ	B	μ	σ	σ/μ	B	μ	σ	σ/μ
3000	0%	655.05	90.2	0.138	5%	638.2	87.7	0.137	15%	592.0	53.0	0.090	80%	554.3	14.9	0.027
5000	10%	625.1	95.0	0.152	25%	600.8	47.6	0.079	35%	597.1	91.5	0.153	55%	570.8	24.7	0.043
7000	20%	606.84	97.9	0.161	20%	566.4	22.3	0.039	70%	563.6	22.8	0.040	65%	566.0	23.7	0.042
9000	30%	569.14	29.8	0.052	50%	573.9	41.8	0.073	85%	556.3	17.7	0.032	50%	573.2	24.9	0.043
11000	25%	608.35	129.4	0.213	40%	573.0	35.7	0.062	60%	568.4	24.4	0.043	70%	563.6	22.7	0.040

Significant change in μ

Significant change in σ

		7%				47%	60%			7%				20%	47%	

Statistical Experiments: Summary

Analysis of $\hat{\mu}$ for controlled GAs with Untuned GAs:

- FLC-SGA significantly better than U-SGA: **47%** of the times (7/15)
- FLC-SSGA significantly better than U-SSGA: **20%** of the times (3/15)
- FLC-SGA significantly worse than U-SGA: **7%** of the times (1/15)
- FLC-SSGA significantly worse than U-SSGA: **7%** of the times (1/15)

Analysis of $\hat{\sigma}$ for controlled GAs with Untuned GAs:

- FLC-SGA significantly better than U-SGA : **60%** of the times (9/15)
- FLC-SSGA significantly better than U-SSGA: **47%** of the times (7/15)
- FLC-SGA significantly worse than U-SGA: **Never**
- FLC-SSGA significantly worse than U-SSGA : **Never**

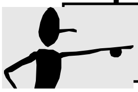
Comparison of Untuned GAs (Simple vs SS):

$\hat{\mu}$ U-SSGA significantly better than U-SGA: **15%** of the times

$\hat{\sigma}$ U-SSGA significantly better than U-SSGA: **30%** of the times

GA Parameter Setting - Outline

- GA Model: Structure, Parameters
- GA Parameter Setting
  - GA Parameter Tuning: Off-line study, Off-line Meta-GA
  - GA Parameter Control: Deterministic Control, Self-Adaptation, Control (FLC)
- GA Parameter Setting: An Application
  - Object-level Representation and Complexity
- First Solution Architecture (U-TGA, TGA, FLC-GA)
  - FLC-GA: KB, I/O, Control Parameters, Rule base and Scaling Factors
  - Results - Performance Tradeoff
  - Problems with First Solution
  - Summary and Analysis of 30 Experiments
  - Addressing Problems with First Solution
- Second Solution
  - FLC: KB, Inputs, Outputs, Termsets, Rule Sets, Control Parameters
  - Statistical Experiments: Set-Up & Measures
  - Summary and Analysis of 1200 Experiments



- Conclusions and Next Steps

Conclusions

FLC State Representation: [Evolution Stage]

- **Since the objectives of the GA are different in the earliest and latest stages, we need time to be an explicit state variable**
- **Diversity is an essential measure of the evolutionary stage:**
  - Percentage Fitness Evaluations (PFE)
  - Genotypic Diversity (GD)

FLC Control Variables: [GA Adaptable Parameters (Subset)]

- **ΔN = Change in Population Size**
- **ΔP_m = Change in Mutation Rate**

Main Result

- **By using the FLC with the above State and Control variables, we achieved a good improvement of the population *average* and an even better improvement of the population *variance*.**
- **No major negative effects on GA performance using FLC**

Next Steps: Multi-Criteria Fitness Functions

- **Scalar-Valued Fitness Functions**
  - Aggregated fitness function focuses on one tradeoff point in frontier
    - Use of ***variable weights*** in multi-criteria aggregating function
    - **Dynamic Control of weights over time**
- **GA Generation of Pareto Frontier**
  - **VE-GA:** Vector Evaluated GA (*Shaffer & Grefenstette 1985*)
    - **Multi-sexual GA** (*Lis & Elben 1997*)
  - **VOW-GA:** Variable Objective Weighting GA (*Hajela & Lin 1992*)
  - **MO-GA:** Multiobjective Optimization GA (*Fonseca & Fleming 1994*)
  - **NP-GA:** Niched Pareto GA (*Horn, Nafpiolitis, Goldberg, 1994*)
  - **NS-GA:** Non Dominated Sorting GA (*Srinivas & Deb, 1995*)
  - **RW-GA:** Random Weights GA (*Ishibushi & Murata, 1998*)
- **Great parameter control opportunity!**

Next Steps: Multi-Criteria Fitness Functions

- **GA Generation of Pareto Frontier (Past Work)**
  - **VE-GA:** Vector Evaluated GA (*Shaffer & Grefenstette 1985*)
    - *N/k sub-populations* created (where *k* is the number of criteria; *N* = Pop-Size)
    - Modified selection operator performs proportional selection for each sub-population according to each objective function
    - Sub-population shuffled to generate population of size N
    - Crossover and Mutation applied to population

Problems with “speciation” (sub-population excelling in only one aspect of fitness)

Problems with maintaining “middlings” (well-round individual not outstanding in any specific aspect of fitness)

Suggested: Heuristic selection to protect middling + heuristic crossbreeding of species

Equivalent to linear combination of objectives - Cannot generate Pareto-optimal solutions in the presence of non-convex search spaces.

Next Steps: Multi-Criteria Fitness Functions

- **GA Generation of Pareto Frontier (Past Work)**
  - **VE-GA:** Vector Evaluated GA (*Shaffer & Grefenstette 1985*)
    - **Multi-sexual GA** (*Lis & Elben 1997*)

      A variation of VE-GA:

      - A fitness function for each criterion
      - Gender-tagged genotypes - a gender for each criterion
      - Multiple-parents recombination operator
        - Replace aggregation of multiple fitness functions with aggregation of multiple genetic material

NEEDS

- **Dynamic determination of offspring gender**
- **Dynamic control of genders distribution in population**

Next Steps: Multi-Criteria Fitness Functions

- **GA Generation of Pareto Frontier (Past Work)**
  - **VOW-GA:** Variable Objective Weighting GA (*Hajela & Lin 1992*)
    - Each objective is assigned a (non-negative, normalized) weight
    - Scalar fitness function is computed with weighted average of objectives
    - Weights are encoded in genotype
    - Diversity of weight combinations promoted by phenotypic fitness “sharing”
    - EA evolves solutions and weights simultaneously
  - **RW-GA:** Random Weights GA (*Ishibushi & Murata, 1998*)
    - Each pair of parents is randomly assigned a weight vector used in their offspring evaluation
    - The (non-negative, normalized) weights represent a direction in the search
    - Each individual ‘s fitness is computed by a weighted average of objectives
    - Each offspring is locally improved by a k-neighborhood search around it
    - At each generation, a provisional set of non-dominated solutions is identified, stored, and updated.
    - A subset of these non-dominated solutions are re-introduced in population (elitism)

Next Steps: Multi-Criteria Fitness Functions

- GA Generation of Pareto Frontier (Past Work)
  - **MO-GA:** Multiobjective Optimization GA (*Fonseca & Fleming 1994*)

    {description - to be continued}
  - **NP-GA:** Niched Pareto GA (*Horn, Nafpiolitis, Goldberg, 1994*)

    {description - to be continued}
  - **NS-GA:** Non Dominated Sorting GA (*Srinivas & Deb, 1995*)

    {description - to be continued}

More Next Steps: Controlling Other Parameters

- **Run-time Controlled GAs Parameters:**
  - DONE **Population size:**
    - larger size: increase parallel search in solution space
    - smaller size: focus on current existing regions
  - DONE **Probability mutation:**
    - Higher prob. of mutation disrupts current solutions - exploration
    - Lower probability of mutation favors current solutions - exploitation
- **Other Possible Run-time Controllable GAs Parameters:**
  - **Customized mutation operators:**
    - Variable amount of changes
      - smaller for good solutions, larger for bad ones
  - **Fitness function**:
    - Evolving fitness function (variable weights in multi-criteria aggregating function)

GAs controlled by FL (cont.)

- **Probability of Selection**:
  - Parametrized slope distribution ranging from:
    - **Uniform** probability: ignore fitness function and perform random selection of parents - extreme case of exploration, to
    - **Proportional** selection with rescaling and other intermediate strategies - compromise between exploration and exploitation cases, and
    - **Ranking**: always select the best N and ignore the rest - extreme case of exploitation
  - Probability as function of fitness and genotypical distance with other solutions - enforcing diversity and favoring exploration
- **Probability of crossover:**
  - Constraints applicability to mostly good solutions
- **Customized-crossover operators** (for real-coded GAs):
  - Selection of crossovers based on T-norms and T-conorms causes offsprings to take more extreme values (exploration)
  - Selection of crossovers based on aggregating operators causes offsprings to take average values (exploitation)