FLC Tuned by EA - Outline

- **Components & Historical Approaches**
- **Application to Automatic Train Handling (ATH)**
- **Solution Architecture**
- **Analysis of Results**
- **Remarks**

FL Controllers Tuned by EAs

- **FLC**
  - **FLC = KB + Inference Engine (with Defuzz.)**
  - **KB parameters:**
    - » Scaling factors (SF)
    - » Membership Functions (MF)
    - » Rule set (RS)
- **EA**
  - Encoding: binary or real-valued
  - Chromosome: string or table
  - Fitness function: Sum quadratic errors, entropy
  - Operators: one-point crossover, max-min arithmetical crossover, point-radius crossover.

FL Controllers tuned by EAs (cont.)

- **Historical Approaches:**
  - **Karr 91-93:**
    - » Chromosome = concatenation of all termsets.
    - » Each value in a termset was represented by 3 binary-encoded parameters.
  - **Lee & Takagi 93:**
    - » Chromosome = 1 TSK rule (LHS: memb. fnct. RHS pol.)
    - » Binary encoding of 3-parameter repr. of each term
  - **Surman et al: 93:**
    - » Fitness function with added entropy term describing number of activated rules

FL Controllers tuned by EAs (cont.)

- **Historical Approaches (cont.):**
  - **Kinzel et al. 94:**
    - » Chromosome = Rule Table
    - » Point-radius crossover changing 3x3 rule window (similar to a two-point crossover for string representation)
    - » Order of tuning:
      - – Initialize rulebase according to heuristics
      - – Apply GAs to find best rule table
      - – Tune membership function of best rule set
  - **Herrera et al. 95:**
    - » Chromosome = concatenation of all rules
    - » Real-valued encoding, Max-min arithmetical crossover

SC in Train Handling: An Example

- **Problem Description: *Automated Train Handling***
  - **Control a massive, distributed system with little sensor information**
  - **Freight trains consist of several hundred heavy railcars connected by couplers (train length up to two miles)**
  - **Couplers have a dead zone and a hydraulically damped spring, causing railcars to move relative to each other and train length to change by 50 – 100 ft.**
  - **The position of the cars and couplers cannot be electronically sensed**

SC in Train Handling: An Example

- **Solution Requirements**
  - An automated system has to satisfy multiple goals:
    - Tracking a velocity reference (defined over distance) to enforce speed limits and respect the train schedule
    - Providing a degree of train-handling uniformity across all crews
    - Operating the train in fuel-efficient regimes
    - Maintaining a smooth ride by avoiding sudden accelerations or brake applications (slack control)

Multi-body regulation problem, subject to proper slack management, without sensors for most of the state

SC in Train Handling: An Example

- **Description of Our Approach**
  - **Use a Velocity Profile externally generated (using classical optimization or Evolutionary Algorithms)**
  - **Use a Fuzzy Logic Control (FLC) to track the velocity reference (Fuzzy PI Control)**
  - **Use an Evolutionary Algorithms to tune the FLC parameters to minimize velocity tracking error and number of throttle changes**
  - **Implement control actions with fuzzy rule set to maintain slack control**

FLC tuned by EAs: Our Approach

- **Chromosome (real-valued encoding)**
  - Chr. 1 = Scaling factors;
  - Chr. 2 = Termsets;
  - Chr. 3 = Rules (not used)
- **Order of tuning (as in Zheng '92):**
  - Initialize rulebase with standard PI structure and termsets with uniformly distributed terms
  - Apply EAs to find best scaling factors
  - Apply EAs to find best termsets
  - Apply EAs to find best rule set (not used)
- **Transition from large to small granularity**

FLC Sensitivity to Parameter Changes

Changing a Scaling Factor

X1			X2		
	Very Low	*Low*	*Medium*	*High*	*Very High*
Very Low	PH	PH	PM	PL	ZE
Low	PH	PM	PL	ZE	NL
Medium	PM	PL	ZE	NL	NM
High	PL	ZE	NL	NM	NH
Very High	ZE	NL	NM	NH	NH

Changing a Term in X1

X1			X2		
	Very Low	*Low*	*Medium*	*High*	*Very High*
Very Low	PH	PH	PM	PL	ZE
Low	PH	PM	PL	ZE	NL
Medium	PM	PL	ZE	NL	NM
High	PL	ZE	NL	NM	NH
Very High	ZE	NL	NM	NH	NH

Changing a Rule

X1			X2		
	Very Low	*Low*	*Medium*	*High*	*Very High*
Very Low	PH	PH	PM	PL	ZE
Low	PH	PM	PL	ZE	NL
Medium	PM	PL	ZE	NL	NM
High	PL	ZE	NL	NM	NH
Very High	ZE	NL	NM	NH	NH

Architecture: Modules, Fitness Funct.

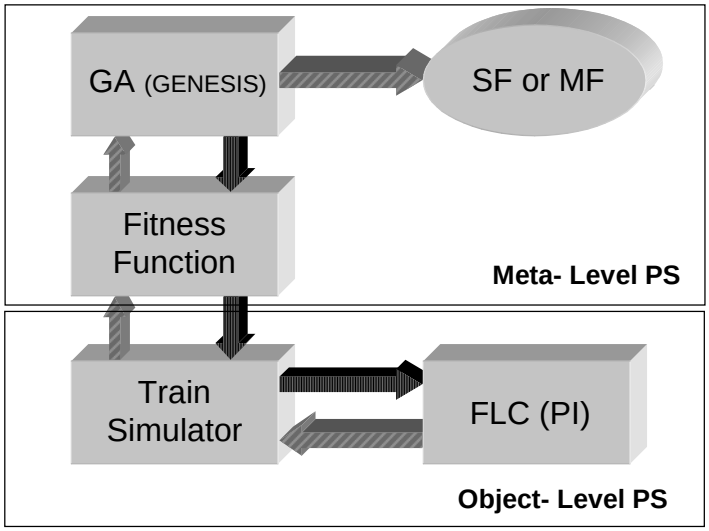
- **Architecture**
  - EA: pop.size=50; P(cross)=.6; P(mut)=.001
  - Three Types of fitness functions
  - Train Simulator: NSTD (STD+TEM)
  - Fuzzy PI (Ke, Kedot, KΔu)
- **Fitness functions (f_1, f_2, f_3)**

$$f_1 = \min\left(\sum_i |notch_i - notch_{(i-1)}| + |dynbrake_i - dynbrake_{(i-1)}|\right)$$

$$f_2 = \min\left(\sum_i |v_i - v_i^d|\right)$$

$$f_3 = \min\left(w_1 \frac{\sum_i |notch_i - notch_{(i-1)}|}{K_1} + w_2 \frac{\left(\sum_i |v_i - v_i^d|\right)}{K_2}\right.$$

FLC tuned by GAs



Experiment Design

- **12 test (4 for each fitness function)**
  - **Initial SF with initial MF;**
  - **EA tuned SF with Initial MF**
  - **Initial SF with EA tuned MF;**
  - **EA tuned SF with EA tuned MF**
- **Train Simulation:**
  - **14 miles long flat track**
  - **1 uniformly heavy train with 100 cars and 4 locomotives**
  - **Analytically computed velocity profile**

Experiment Design

- **Representation:**
  - **SF: 3 floating point values for Ke, Kedot, KΔu**
  - **MF (21-9) = 12 values**
    - » 21 parameters: [($Left_i$,$Center_i$, $Right_i$) for i=1, ..., 7]
    - » 9 dependent values: [($Left_i$ = $Right_{(i+1)}$) for i=1, ..., 6] + [$Center_1$= $Center_7$]+[$Right_1$ = $Left_7$ = 0]
  - **Constraints to maintain 0.5 terms overlap, for best interpolation**

Experiments Results

- **Experiment Results with f_1**

Description	Time	Journey	Fuel	Fitness	Gen.
Initial SF; Initial MF	26.5	14.26	878	73.2	
EA tuned SF; Initial MF	27.8	14.21	857	15.15	4
Initial SF; EA tuned MF	26.00	14.18	879	70.93	2
EA tuned SF; EA tuned MF	28.3	14.12	82[illegible]	14.64	[illegible]

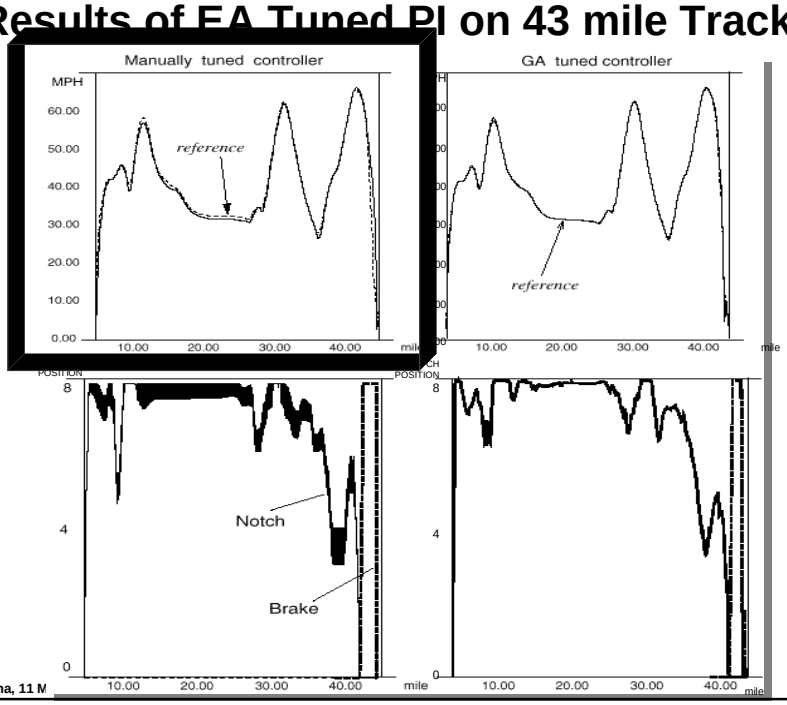
- **Experiment Results with f_3**

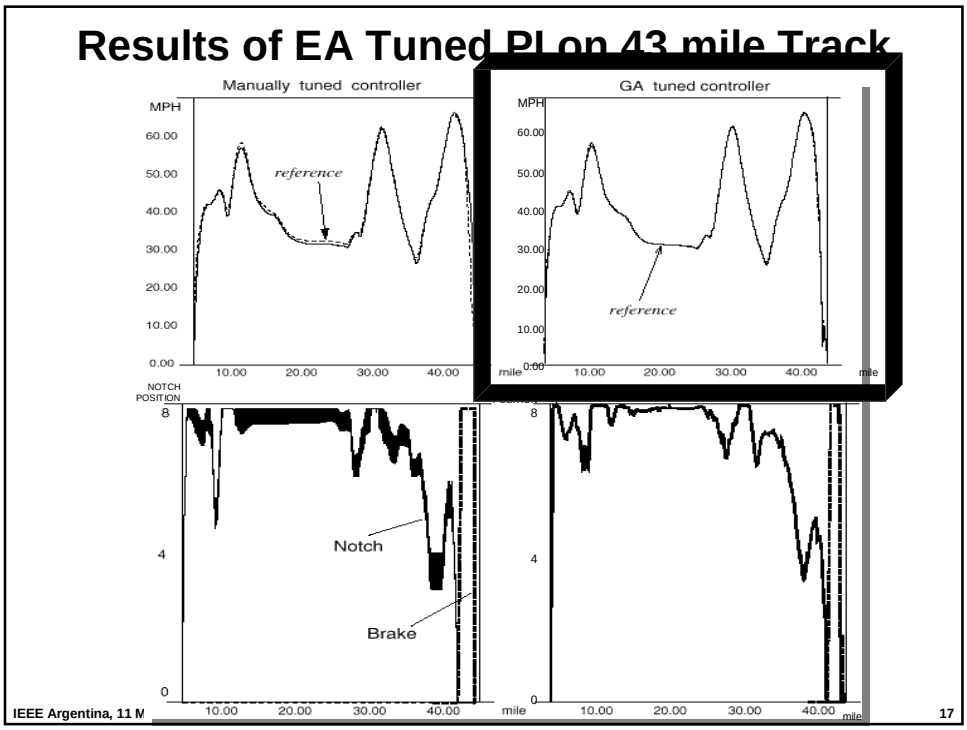
Description	Time	Journey	Fuel	Fitness	Gen.
Initial SF; Initial MF	26.5	14.26	878	1	
EA tuned SF; Initial MF	27.2	14.35	871	0.817	4
Initial SF; EA tuned MF	26.26	14.18	871	0.942	20
EA tuned SF; EA tuned MF	27.3	14.35	872	0.817	10

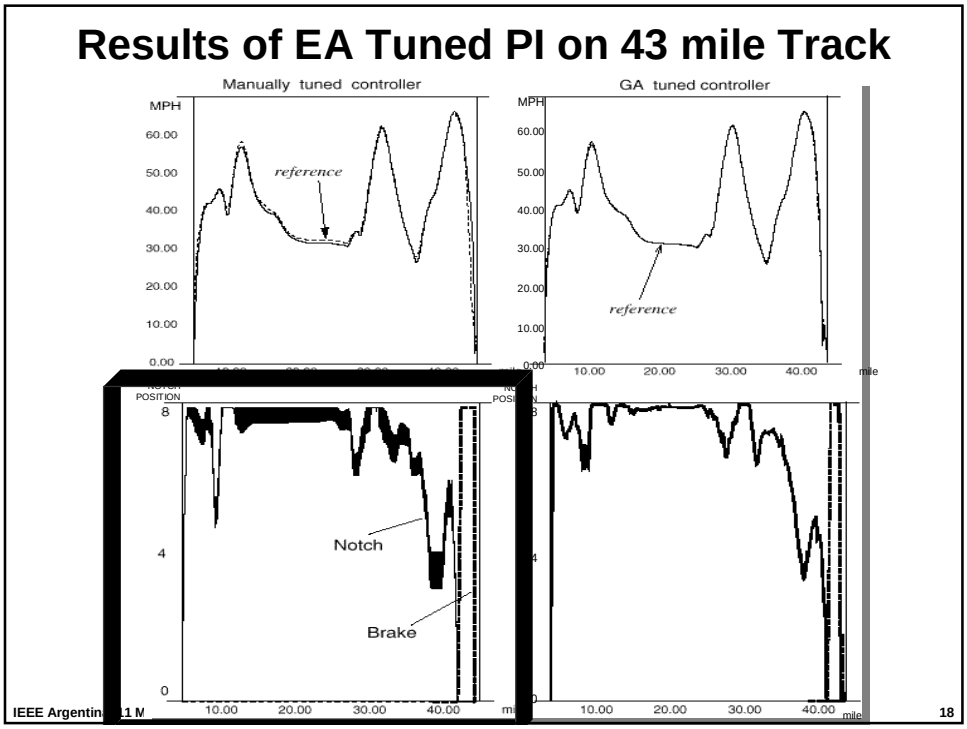
Tuning of FLC with EA: Remarks

- Verified tuning order proposed by Zheng (92)
  - » SF tuning: major impact
  - » MF tuning: minor impact
  - » RS tuning: almost no impact
- For both f1 and f3, fuel minimization is implicitly derived from throttle jockeying minimization
- Complex fitness function (requiring simulation run - 23 sec for each chromosome evaluation) limited trials number - with no apparent impact
- Successfully tested on simulated 43 mile long track with altitude excursions
  - » (Selkirk, NY->Framingham, MA)

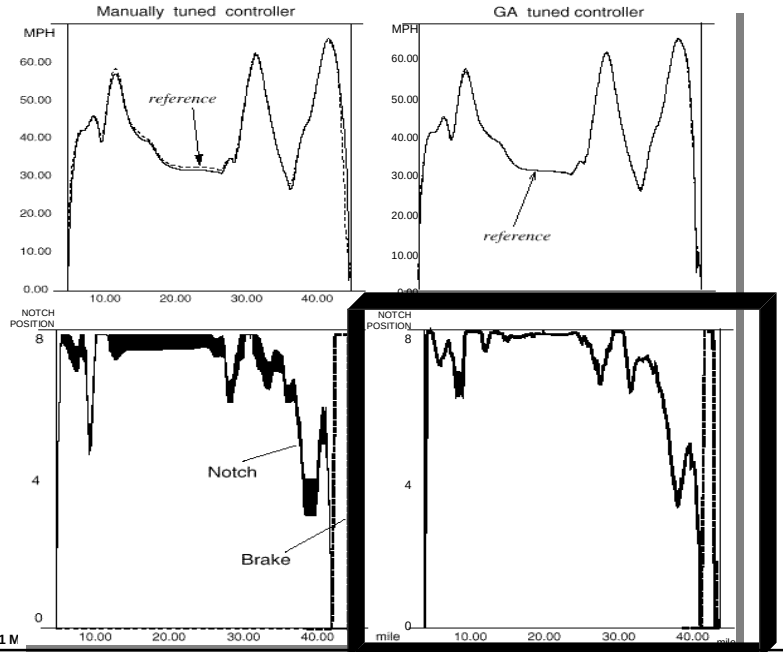
Results of EA Tuned PI on 43 mile Track




Results of EA Tuned PI on 43 mile Track
Manually tuned controller
GA tuned controller
MPH
reference
NOTCH POSITION
Notch
Brake
mile
IEEE Argentina, 11 M
17


Results of EA Tuned PI on 43 mile Track
Manually tuned controller
GA tuned controller
MPH
reference
POSITION
Notch
Brake
mile
IEEE Argentina, 11 M
18

Results of EA Tuned PI on 43 mile Track



IEEE Argentina, 11 M 19

NFL, Meta-Heuristics & Hybrid SC: Outline

- The NFL
- Tuning or Controlling the Object-Level Problem Solver (PS) with Meta-Heuristics
- Soft Computing Overview
  - SC Components: PR, FL, NN, EA
- Using SC to implement the Meta-Heuristics: Modeling with FL and EA

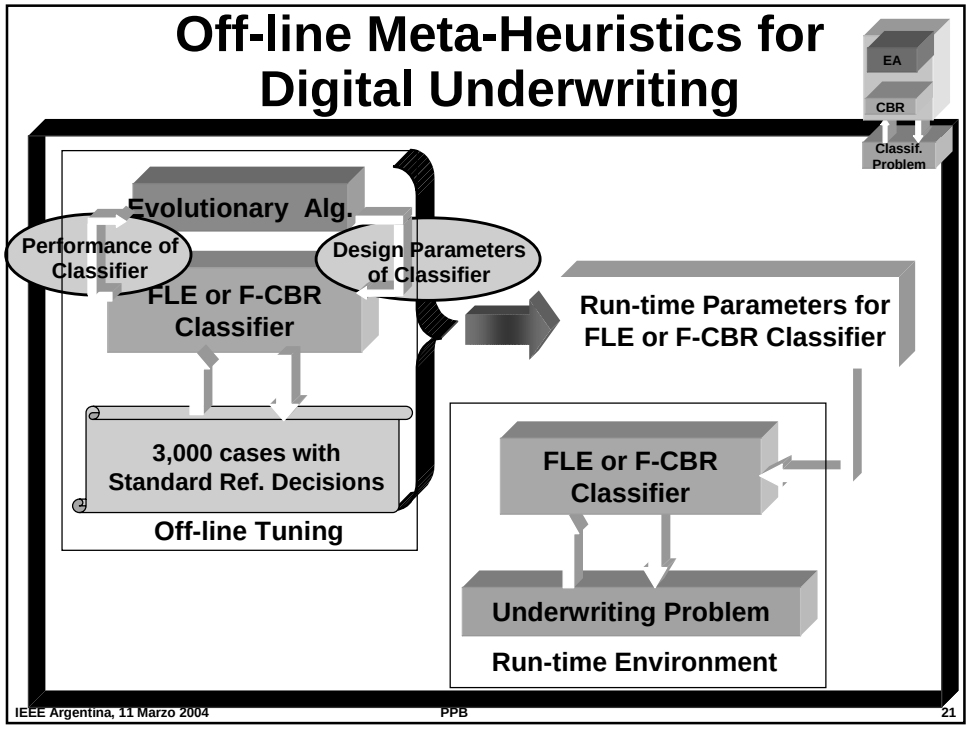
Example of Hybrid SC Systems

- FLC Parameter Tuning by EA
- **FLR and CBR Parameter Tuning by EA**
- EA Parameter Setting (by EA) or Control (by FL)

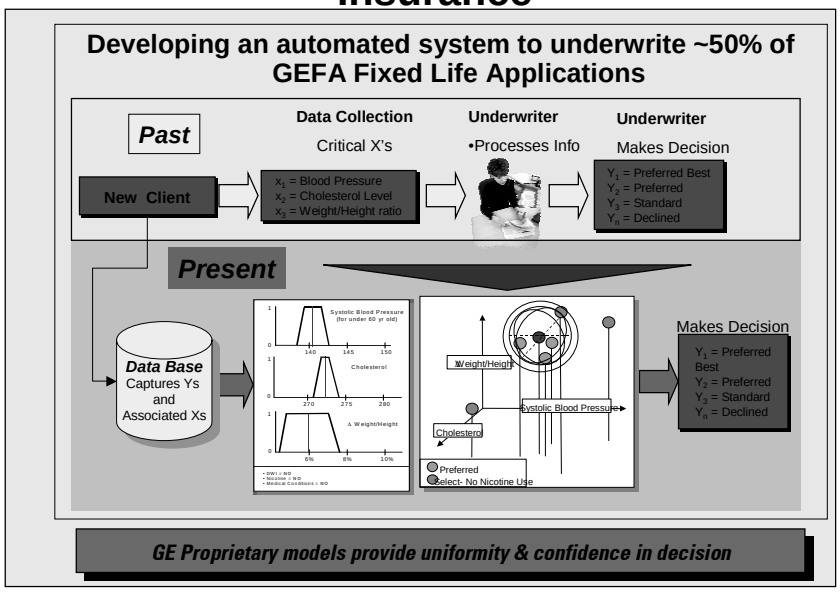
- Conclusions

IEEE Argentina, 11 Marzo 2004 PPB 20

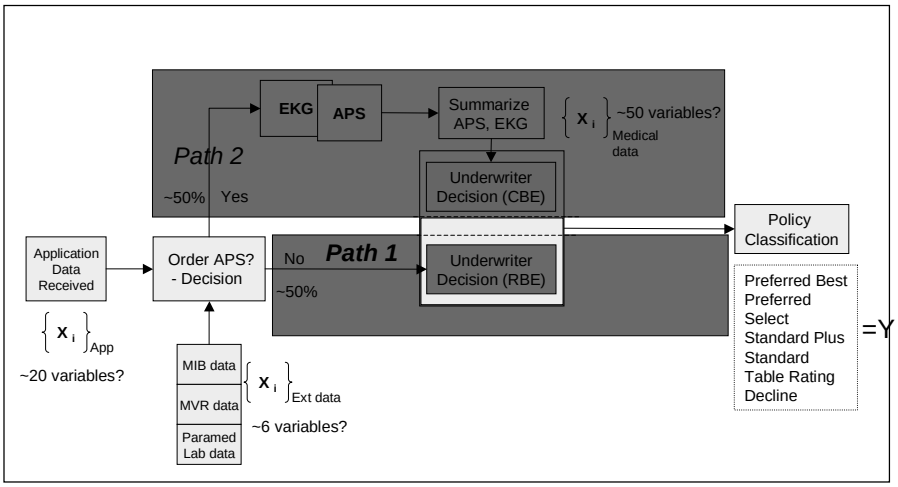
Off-line Meta-Heuristics for Digital Underwriting



Example: Digital Underwriting for GE Insurance



Path 1: Clean Cases (Rule Based)



Establish Confidence & Start Digital Underwriting

Fuzzy Logic Engine (FLE)

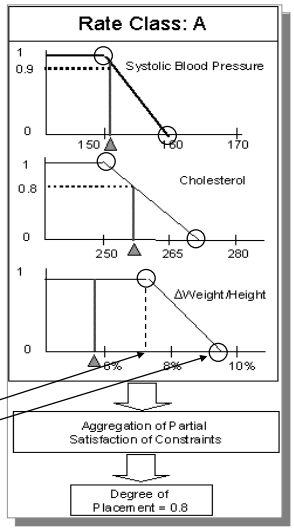
- **Use Fuzzy Logic rules to codify UW decision process**
- **Identify significant decision factors by interviewing UW's and encode rules using fuzzy membership functions**
- **Compute two Measures of Satisfaction (MS) for each rate class**
  - **$MS_1(r)$: Intersection of soft constraints $C_{i,r}(x_i)$**

$$MS_1(r) = Min_{i=1}^{n} C_{i,r}(x_i)$$

  - **$MS_2(r)$: Sum of Missing Points (Cumulative measure of constraint satisfaction)**

$$MS_2(r) = \sum_{i=1}^{n} [1 - C_{i,r}(x_i)] = n - n\overline{C}_r = n(1 - \overline{C}_r)$$

- **Compare $MS_1(r)$ and $MS_2(r)$ against their corresponding thresholds τ_1 and τ_2**
- **Assign the most competitive rate class for which** $MS_1(r) \le \tau_1$ **and** $MS_2(r) \le \tau_2$
- **Parametric representation of $C_{i,r}(x_i) = (a_{i,r}, b_{i,r})$**



Fuzzy Logic Engine (FLE)

- **Use Fuzzy Logic rules to codify UW decision process**
- **Identify significant decision factors by interviewing UW's and encode rules using fuzzy membership functions**
- **Compute two Measures of Satisfaction (MS) for each rate class**
  - **$MS_1(r)$: Intersection of soft constraints $C_{i,r}(x_i)$**

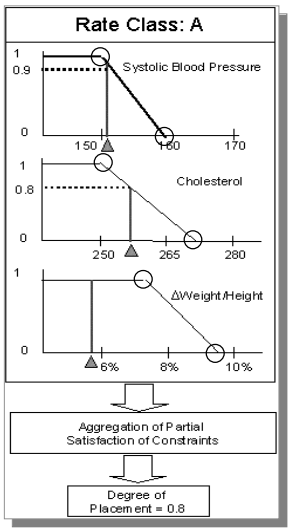
$$MS_1(r) = Min_{i=1}^{n} C_{i,r}(x_i)$$

  - **$MS_2(r)$: Sum of Missing Points (Cumulative measure of constraint satisfaction)**

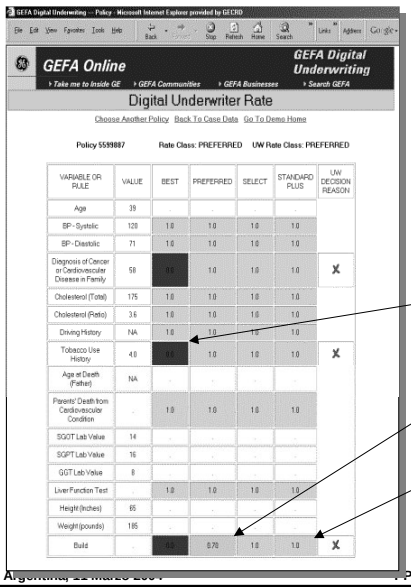
$$MS_2(r) = \sum_{i=1}^{n} [1 - C_{i,r}(x_i)] = n - n\overline{C}_r = n(1 - \overline{C}_r)$$

- **Compare $MS_1(r)$ and $MS_2(r)$ against their corresponding thresholds τ_1 and τ_2**
- **Assign the most competitive rate class for which** $MS_1(r) \le \tau_1$ **and** $MS_2(r) \le \tau_2$
- **Parametric representation of $C_{i,r}(x_i) = (a_{i,r}, b_{i,r})$**

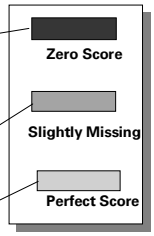
FLE Design configuration $[\ \ldots (a_{i,r}, b_{i,r}), \ldots, \tau_1, \tau_2\ ]$



Fuzzy Logic Engine



- If the APS is not required, the RDE system rates the policy by the Rule Based engine.
- The results are stored in the DB.
- The results for the case under review are retrieved and displayed via the web.
- The reasons are presented in both color coded tables and plain English for better understanding.



Fuzzy Case Based Engine (FCBE)

On-line CBE

1. *Represent New Case* with a Query
2. *Retrieve relevant cases* from the Case Base (CB)
3. *Evaluate and Rank the retrieved cases* - using similarity measure
4. *Select best cases*
5. *Derive solution to current case* from (adapted) retrieved solutions
6. *Evaluate solution's quality* - using a confidence factor
7. *Store newly solved case* in the case memory

Off-line CB Maintenance

IEEE Argentina, 11 Marzo 2004 PPB 27

Fuzzy Case Based Engine (FCBE) - Step 1:

Represent New Case with an SQL Query

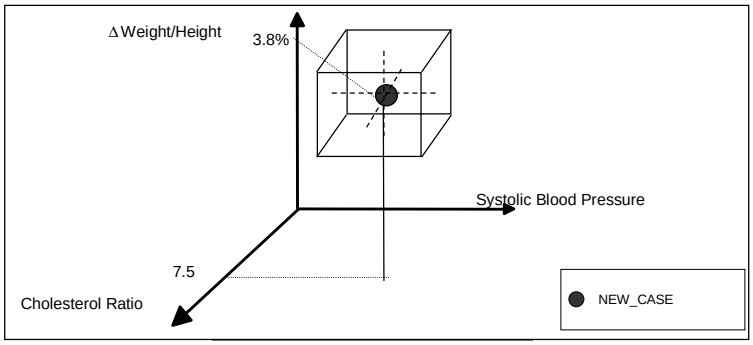
GEFA Insurance

X_{New_Case} = *[7.5, 3.8, …,] Impairment=* ***Hypertension***

Q1: [f1(x_1), f2(x_2), …, fn(x_n)] AND [Impairment=label]

For instance:

Q1: [Support(*Around (7.5;x_1)*), Support(*Around (3.8;x_2)*), Support(NORMAL(X_i)),…, Support(NORMAL(X_n))] *AND* [Impairment=Hypertension]



IEEE Argentina, 11 Marzo 2004 PPB 28

Fuzzy Case Based Engine (FCBE) - Step 2:

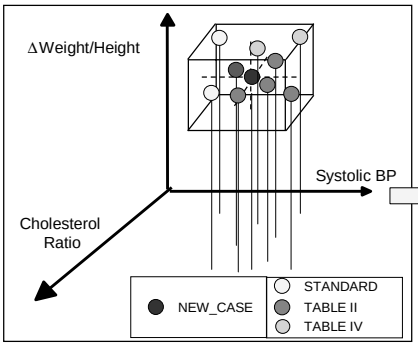
Retrieve relevant cases from the case library

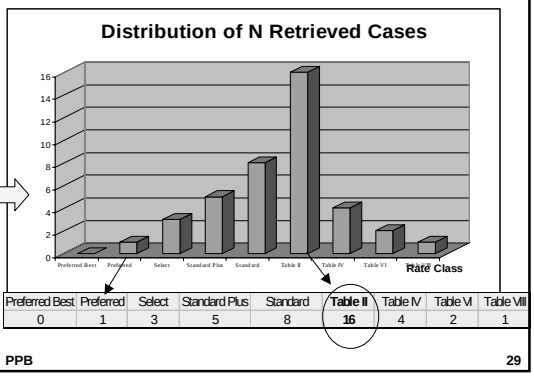
GEFA Insurance

Rate Classes Decision

Retrieved N Cases C_i $\quad where\ (1 \le i \le N)$

C_1 = [7.4, 4.3, …,] Impairment= **Hypertension**	Table VI
⋮	
C_i = [7.2, 3.3, …,] Impairment = **Hypertension**	Table II
⋮	
C_N = [7.9, 4.1, …,] Impairment = **Hypertension**	Table IV





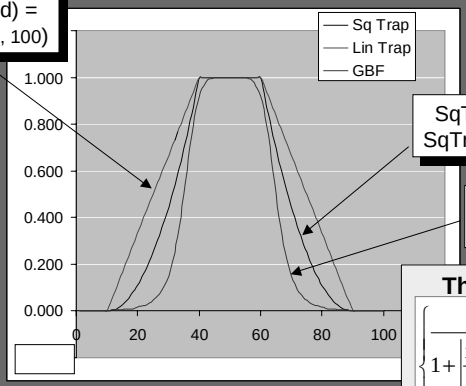
Preferred Best	Preferred	Select	Standard Plus	Standard	Table II	Table IV	Table VI	Table VIII
0	1	3	5	8	16	4	2	1

Fuzzy Case Based Engine (FCBE) - Step 3:

Evaluate and Rank the retrieved cases - using similarity measure

Selection of MF family

Trap(x; a, b, c, d) = Trap(x; 10, 20, 60, 100)



SqTrap(x; a, b, c, d) = SqTrap(x; 10, 20, 60, 100)

GBF(x; a, b, c) = **GBF(x; 15, 3, 50)**

Thresholded GBF:

$$\begin{cases} \dfrac{1}{1+\left|\dfrac{x-c}{a}\right|^{2b}} & \text{if } GBF > 10^{-5} \\ o & \text{otherwise} \end{cases}$$

• The membership value is a ***function of the inverse of the dis*** respect ot the reference (probe) upon which the membership is

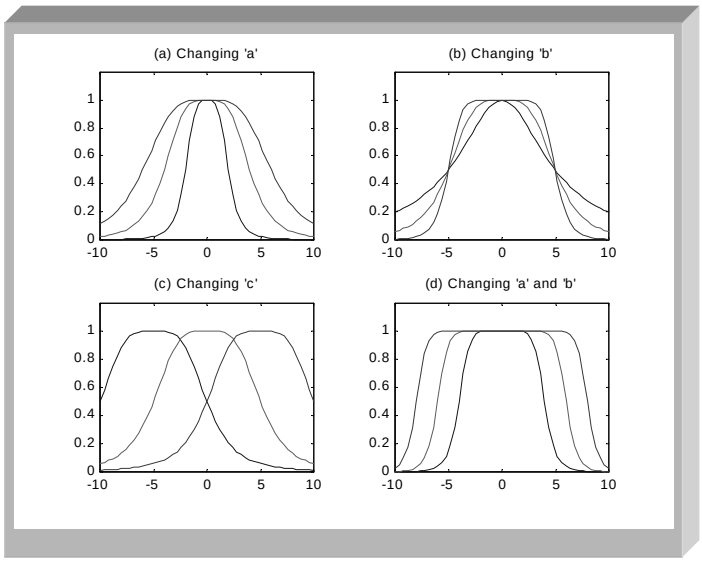
• A linear slope compensates too much (One close point would be averaged out w points located at n-times the distance – but before saturation)

• A non-linear slope is preferable (***Generalized Bell Function***) – we should use a threshold to force a finite support.

IEEE

Generalized Bell Function: Effect of changing Parameters {a,b,c}

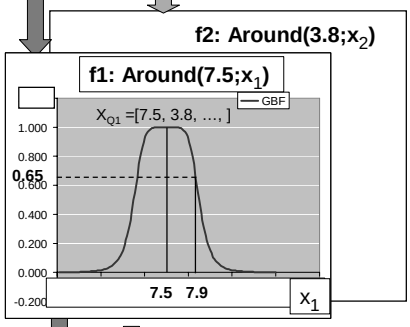
$$\mu_A(x) = \frac{1}{1 + \left| \frac{x - c_i}{a_i} \right|^{2b}}$$



Fuzzy Case Based Engine (FCBE) - Step 3:

Evaluate and Rank the retrieved cases - using similarity measure

C_N** = [7.9, 4.1, …,] AND Impairment=**Hypertension | Table IV



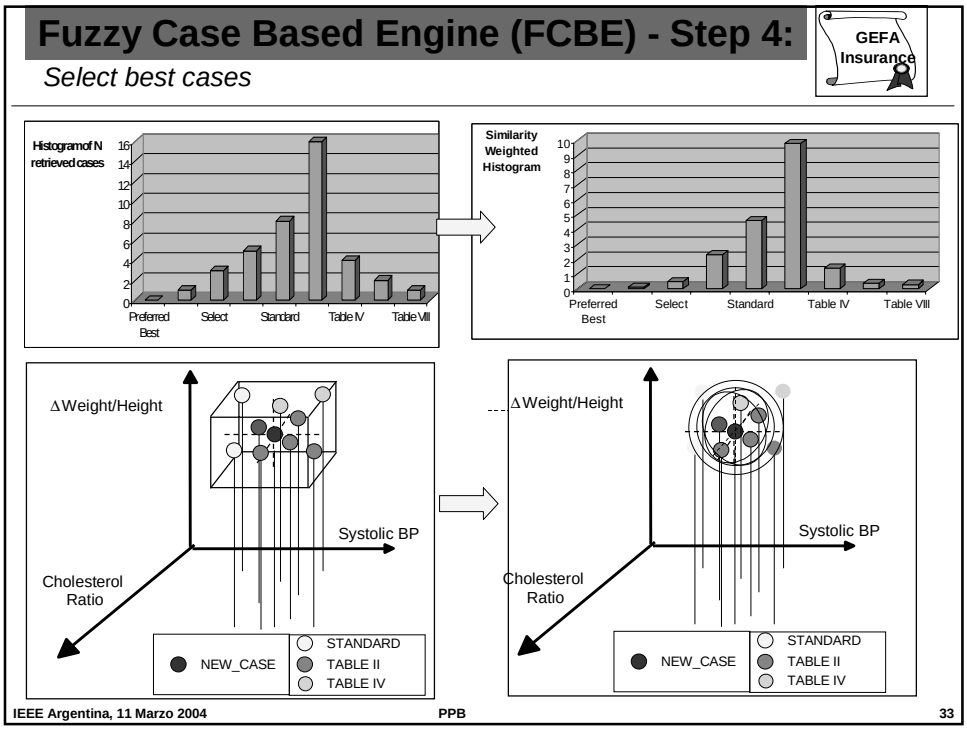
Preferred Best	Preferred	Select	Standard Plus	Standard	Table II	Table IV	Table VI	Table VIII
0	1	3	5	8	16	4	2	1
	0.1	0.1	0.2	0.2	0.3	0.2	0.2	0.3
		0.2	0.3	0.5	0.4	0.3	0.2	
		0.2	0.5	0.5	0.4	0.3		
			0.6	0.6	0.5	0.6		
			0.7	0.6	0.5			
				0.6	0.5			
				0.8	0.5			
				0.8	0.6			
					0.6			
					0.7			
					0.7			
					0.7			
					0.8			
					0.8			
					0.9			
					0.9			

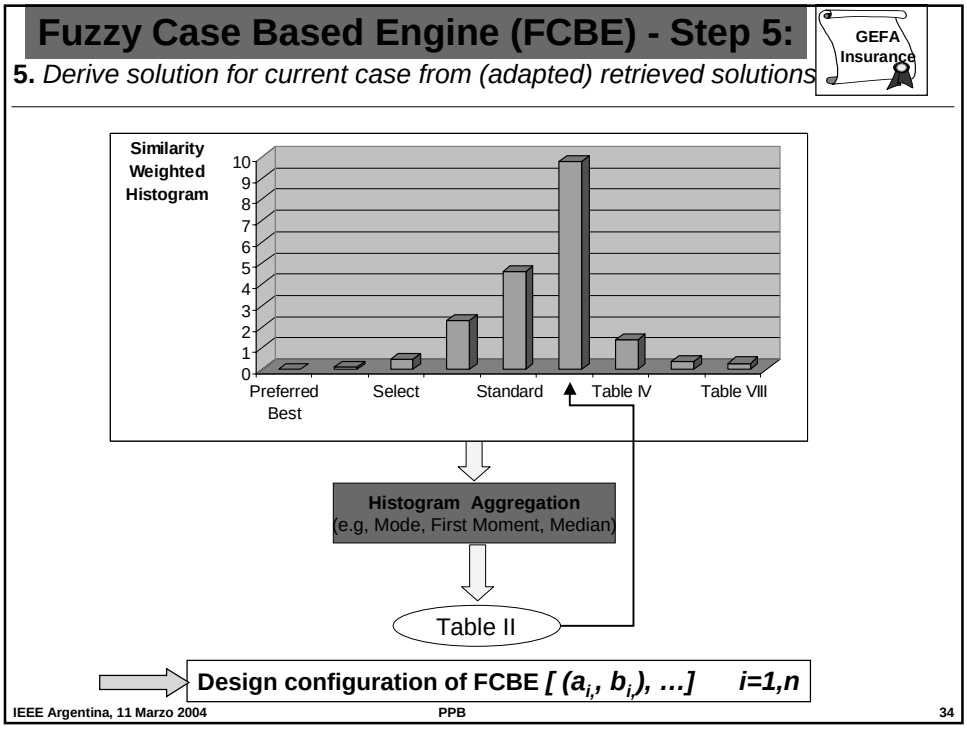
$Q_1(C_N)$ = [0.65, **0.6,** …,] AND [1] → Preference Aggregation → $S(C_N, Q_1) = 0.6$

Preference Vector for Case

Similarity

$$S = \cap_{i=1}^{n} GBF(x_i; a_i, b_i, c_i) = Min_{i=1}^{n} GBF_i(x_i; a_i, b_i, c_i)$$

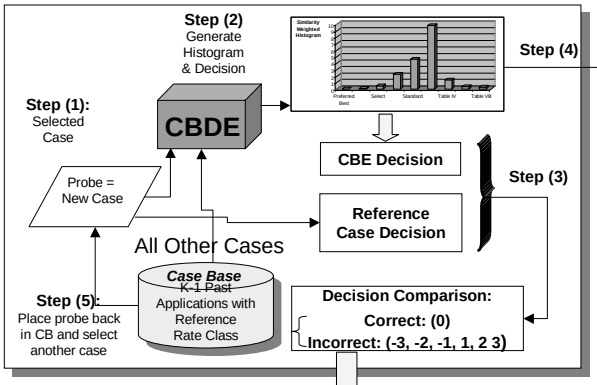

Fuzzy Case Based Engine (FCBE) - Step 4:
Select best cases
GEFA Insurance
Histogram of N retrieved cases
Preferred Best
Select
Standard
Table IV
Table VIII
Similarity Weighted Histogram
Δ Weight/Height
Systolic BP
Cholesterol Ratio
NEW_CASE
STANDARD
TABLE II
TABLE IV
IEEE Argentina, 11 Marzo 2004
PPB
33


Fuzzy Case Based Engine (FCBE) - Step 5:
5. Derive solution for current case from (adapted) retrieved solutions
GEFA Insurance
Similarity Weighted Histogram
Preferred Best
Select
Standard
Table IV
Table VIII
Histogram Aggregation
(e.g, Mode, First Moment, Median)
Table II
Design configuration of FCBE [(a_i, b_i), …] i=1,n
IEEE Argentina, 11 Marzo 2004
PPB
34

Fuzzy Case Based Engine (CBE) - Step 6a:

GEFA Insurance

Evaluate solution's quality - Develop Confidence Functions (off-line)



Query-Dependent Parameters [x_1-x_3]

X_1= number of cases retrieved
X_2= measures of dispersion
X_3= percentage of cases in mode

Step (6)
Loop through all
K cases in CB

Classification	X1: N = Number of retrieved cases for each probe									
Distance	[1-4]	[5-8]	[9-12]	[17-20]	[21-24]	[25-28]	[29-32]	[33-36]	[37-40]	40+
-3	12.20%	8.51%	6.25%	3.08%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
-2	21.95%	25.53%	14.06%	10.77%	10.45%	9.52%	10.00%	7.27%	3.77%	3.45%
-1	14.63%	19.15%	25.00%	26.15%	28.36%	20.63%	15.00%	16.36%	15.09%	13.79%
0	12.20%	12.77%	23.44%	30.77%	37.31%	49.21%	53.33%	60.00%	66.04%	68.97%
1	14.63%	17.02%	20.31%	23.08%	19.40%	17.46%	20.00%	14.55%	13.21%	12.07%
2	19.51%	14.89%	10.94%	6.15%	4.48%	3.17%	1.67%	1.82%	1.89%	1.72%
3	4.88%	2.13%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Total % within bins	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%

Fuzzy Case Based Engine (CBE) - Step 6a:

GEFA Insurance

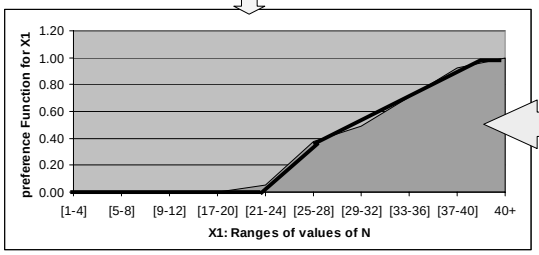
Evaluate solution's quality - Develop Confidence Functions (off-line) cont.

Weights
-11
-6
-1
4
-1
-6
-11

Classification	X1: N = Number of retrieved cases for each probe									
Distance	[1-4]	[5-8]	[9-12]	[17-20]	[21-24]	[25-28]	[29-32]	[33-36]	[37-40]	40+
-3	12.20%	8.51%	6.25%	3.08%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
-2	21.95%	25.53%	14.06%	10.77%	10.45%	9.52%	10.00%	7.27%	3.77%	3.45%
-1	14.63%	19.15%	25.00%	26.15%	28.36%	20.63%	15.00%	16.36%	15.09%	13.79%
0	12.20%	12.77%	23.44%	30.77%	37.31%	49.21%	53.33%	60.00%	66.04%	68.97%
1	14.63%	17.02%	20.31%	23.08%	19.40%	17.46%	20.00%	14.55%	13.21%	12.07%
2	19.51%	14.89%	10.94%	6.15%	4.48%	3.17%	1.67%	1.82%	1.89%	1.72%
3	4.88%	2.13%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Total % within bins	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%

$$f(Bin_k) = \sum_{i=1}^{7} w_i D(i,k)$$

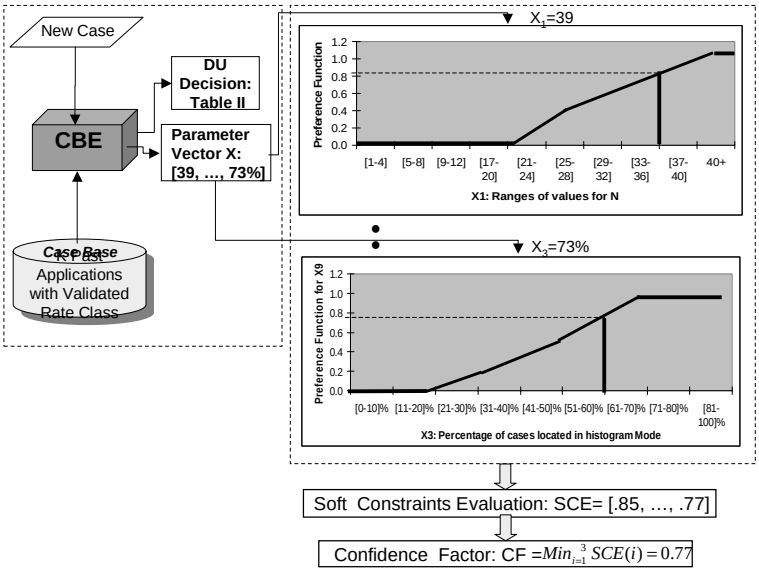
Similar to fitness function used in optimization



Confidence Function for cardinality of retrieved cases

Fuzzy Case Based Engine (CBE) - Step 6b:

Evaluate solution's quality - Generate a Confidence Factor (on-line)



Soft Constraints Evaluation: SCE= [.85, …, .77]

Confidence Factor: CF $= Min_{i=1}^{3} SCE(i) = 0.77$

RBDE Parameter Tuning

Decision Engine Parameter Tuning/Optimization

Evolutionary Optimization Algorithm

Create initial population $\langle x_i \rangle$

Proportional Selection

Intermediate population $\langle x_i \rangle' = S\langle x_i \rangle$ (exploitation)

Population of trial solutions

Stochastic Variation

New population $\langle x_{i+1} \rangle = V\langle x_i \rangle'$ (exploration)

More superior population of trial solutions

Problem Definition

$$\min_{x \in \chi} \psi(x) \quad \chi \subset \Re^n \quad \psi : \chi \rightarrow \Re_+$$

Space of decision engine designs

Space of positive real numbers

Decision mismatch penalty function

Space of n dimensional positive real vectors

Decision Engine Parameter Tuning/Optimization

GEFA Insurance

Approach: Iterative Process

Space of Decision Engine Designs → mapping → Space of Decision Mismatch Matrices → mapping → Space of Decision Mismatch Penalties

Trial Solutions (Search Probes) ← Evolutionary Algorithm ← Search Feedback

Intelligently probe for those decision engine designs that minimize decision mismatch penalties

Design Parameter Representation

100 Solutions

[(8%, 9.2%, 12%, 14%), ... ,

Population of Trial Solutions

[(7%, 14%, 18%, 21%), ... ,

Each individual in a given population represents a possible decision engine design parameter set

Evaluation Process

Design Parameters → Decision Engine Instance → DE Decision

Case

Case Base: Past Validated Cases

Decision Engine Parameter Tuning/Optimization: Matrix M

GEFA Insurance

Computation of Rate Class Decision Mismatch Penalties

Individual k → Decision Engine Instance → DE Decision

Case → Decision Engine Instance

Case → Validated Case Decision (All Other Cases)

Case Base: 2,000-3,000 Past Applications with Validated Rate Class

Decision Comparison: Correct/Incorrect

Loop through <u>all</u> cases in Case Base

$$\text{Score (k)} = \Psi(Ind_k) = \sum_{i=1}^{T} \sum_{j=1, j \neq i}^{T} P(i,j) M(i,j)$$

Penalty

Comparison Matrix M (TxT)		DU Decisions PB	P	S	St+	St	Tab 2		Tab 32
	PB	45.0%	3.0%	1.5%	0.0%				
	P	1.4%	23.0%	...					
Case	S		...	6.0%					
Decisions	St+				...				
	St								
	Tab 2								
								...	
	Tab 3								4.5%

Stricter

More Liberal

GEFA Insurance

Penalty Matrix P

- Used Actuarial studies to developed a Penalty Matrix P for the Confusion Matrix based on financial impact of misclassifications.
- Matrix P shows an example of Minimal Regret of Net Present Value

P =

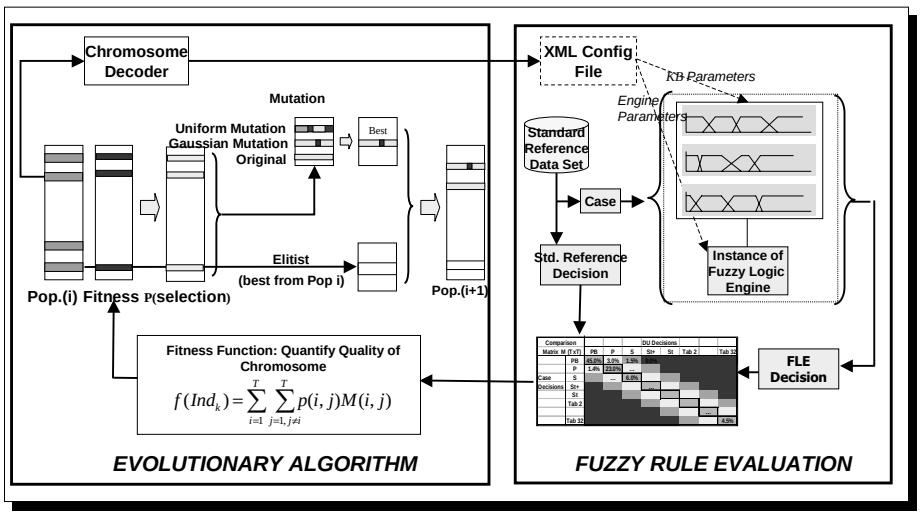
Correct Decision Rate Class	Engine Decision Rate Class					
RC	A	B	C	D	E	…
A	$0	-$3	-$11	-$15	…	
B	-$6	$0	-$5			
C	-$18	…	$0			
D	-$22					
E	-$26					
…						

- Tuning goal:

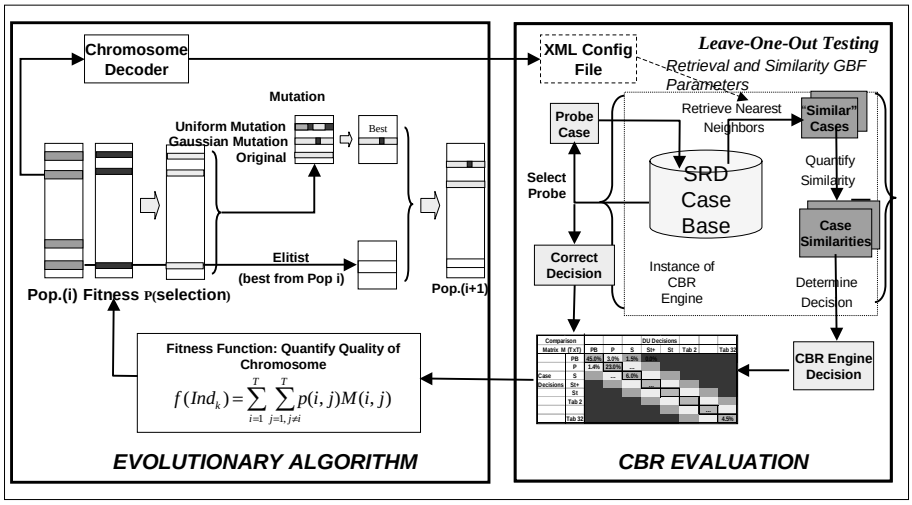
  Minimize cost of misclassification

$$\sum_{i=1}^{T}\sum_{j=1}^{T} P(i,j)\times M(i,j)$$

Fuzzy Rule Based Optimization



Case Based Reasoning Optimization



RBE Parameter Tuning via Optimization
RBE Optimization Results (Before and After)

GEFA Insurance

Manually Tuned RBE

		Rule-Based Engine Decision									
		PB	P	Sel	Std+	Std	P Nic	Std+Nic	Std Nic	UW	
Correct (UW) Decision	PB	489	30	2	0	0				0	521
	P	1	332	38	1	1				1	374
	Sel	0	5	379	27	2				2	415
	Std+	0	0	0	297	34				27	358
	Std	0	0	0	2	202				24	228
	P Nic						284	17	0	0	301
	Std+Nic						1	321	16	19	357
	Std Nic						0	2	150	32	184
	UW	0	2	0	0	0	2	1	1	176	182
		490	369	419	327	239	287	341	167	281	2920

Coverage (not UW)		90.38%	2639	out of	2920
Relative Accuracy (no UW)		92.99%	2454	out of	2639
Global Accuracy		90.07%	2630	out of	2920
RELATIVE MISCLASSIFICATION					
+ 1 rate class	(no UW)	6.14%	162	out of	2639
- 1 rate class	(no UW)	0.45%	12	out of	2639
+ 2 rate class	(no UW)	0.19%	5	out of	2639
- 2 rate classes	(no UW)	0.04%	1	out of	2639
+ 3(+) rate class	(no UW)	0.04%	1	out of	2639
- 3(+) rate class	(no UW)	0.15%	4	out of	2639

Genetic Algorithms-Tuned RBE

		Rule-Based Engine Decision									
		PB	P	Sel	Std+	Std	P Nic	Std+Nic	Std Nic	UW	
Correct (UW) Decision	PB	501	20	0	0	0				0	521
	P	1	360	11	0	1				1	374
	Sel	0	9	394	10	2				0	415
	Std+	0	0	8	304	24				22	358
	Std	0	0	0	4	212				12	228
	P Nic						296	5	0	0	301
	Std+Nic						5	327	12	13	357
	Std Nic						0	2	164	18	184
	UW	0	2	0	0	0	2	1	1	176	182
		502	391	413	318	239	303	335	177	242	2920

Coverage (not UW)		91.71%	2678	out of	2920
Relative Accuracy (no UW)		95.52%	2558	out of	2678
Global Accuracy (with UW)		93.63%	2734	out of	2920
RELATIVE MISCLASSIFICATION					
+ 1 rate class	(no UW)	3.06%	82	out of	2678
- 1 rate class	(no UW)	1.12%	30	out of	2678
+ 2 rate class	(no UW)	0.07%	2	out of	2678
- 2 rate classes	(no UW)	0.04%	1	out of	2678
+ 3(+) rate class	(no UW)	0.04%	1	out of	2678
- 3(+) rate class	(no UW)	0.15%	4	out of	2678

The Optimization improved the global accuracy from 90.1% to 93.6%

Evolutionary Tuning Results

GEFA Insurance

Fuzzy Logic Engine

Metric	Sub-Optimal Parameters	Optimized Parameters
Coverage	90.38%	91.71%
Relative Accuracy	92.99%	95.52%
Global Accuracy	**90.07%**	**93.63%**

Fuzzy Case-Based Engine

Metric	Sub-Optimal Parameters	Optimized Parameters
Coverage	47.97%	98.86%
Relative Accuracy	92.10%	90.80%
Global Accuracy	**44.18%**	**89.77%**

Coverage:
 # Decisions / # Cases

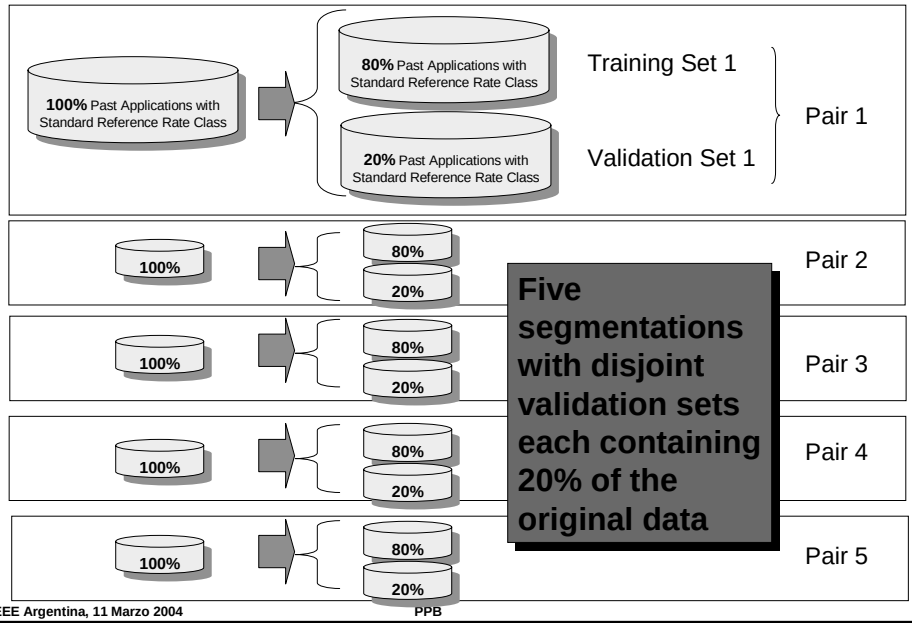
Relative Accuracy:
#Correct Decisions / # Decisions

Global Accuracy:
[# Correct Decisions + # Correct No-decisions (UW)] / # Cases

IEEE Argentina, 11 Marzo 2004 PPB 45

RBDE Parameter Tuning: Robustness & Cross-Validation
Data Set Segmentation in Training and Validation

GEFA Insurance



Five segmentations with disjoint validation sets each containing 20% of the original data

IEEE Argentina, 11 Marzo 2004 PPB 46

RBDE Parameter Tuning: Robustness Conclusions

GEFA Insurance

Coverage		
Pair	Training Set	Validation Set
1	91.94%	92.11%
2	91.73%	90.39%
3	91.68%	93.14%
4	91.47%	92.97%
5	92.24%	90.41%
min	**91.47%**	**90.39%**
max	**92.24%**	**93.14%**

Relative Accuracy		
Pair	Training Set	Validation Set
1	94.03%	93.85%
2	95.28%	93.93%
3	94.72%	95.21%
4	93.81%	92.80%
5	94.75%	92.23%
min	**93.81%**	**92.23%**
max	**95.28%**	**95.21%**

Global Accuracy		
Pair	Training Set	Validation Set
1	92.41%	92.11%
2	93.66%	90.22%
3	92.97%	93.48%
4	91.98%	90.74%
5	92.67%	91.44%
min	**91.98%**	**90.22%**
max	**93.66%**	**93.48%**

Coverage:
 # Decisions / # Cases

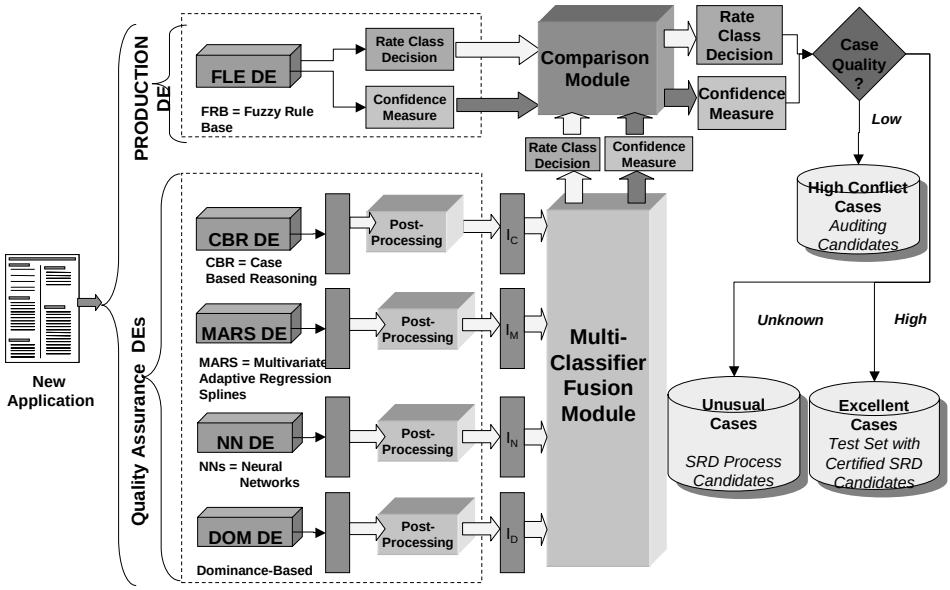
Relative Accuracy:
#Correct Decisions / # Decisions

Global Accuracy:
[# Correct Decisions + # Correct No-decisions (UW)] / # Cases

- **Conclusions:**
  - ***Remarkable robustness for both training and validation sets across all five models (as evidenced from above ranges)***
- **Notes regarding the Off-line Certification tollgate :**
  - ***The RBDE (un-scaled) Rate of Deviation (6.37%) was computed as: (1 - Global Accuracy)***
  - ***The financial impact was computed from the Relative Accuracy***

Fusion of Multiple Models to Validate RBE Output and Verify Quality of Cases to be Stored in CB

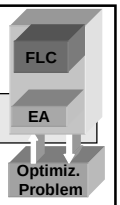
GEFA Insurance



NFL, Meta-Heuristics & Hybrid SC: Outline

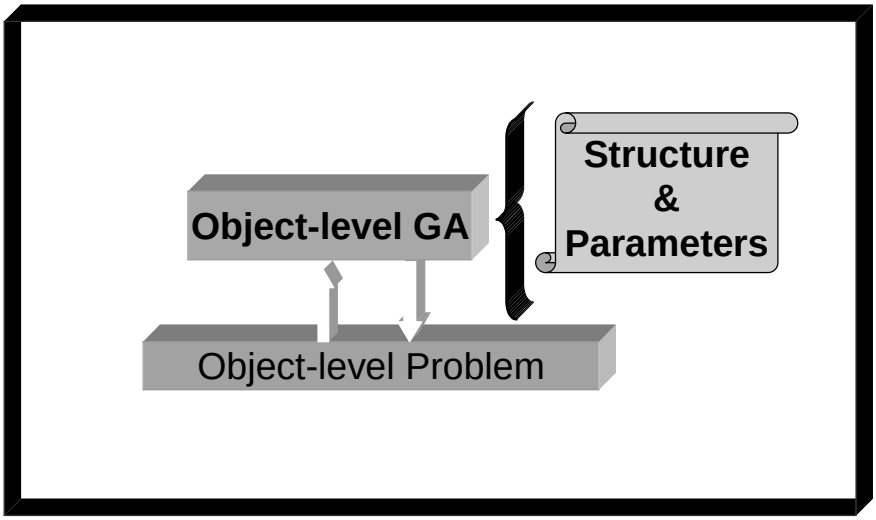
- The NFL
- Tuning or Controlling the Object-Level Problem Solver (PS) with Meta-Heuristics
- Soft Computing Overview
  - SC Components: PR, FL, NN, EA
- Using SC to implement the Meta-Heuristics: Modeling with FL and EA
- **Example of Hybrid SC Systems**
  - FLC Parameter Tuning by EA
  - FLR and CBR Parameter Tuning by EA
  - **EA Parameter Setting (by EA) or Control (by FL)**
- Conclusions

EA Parameter Setting



- **EA Model:**
  - Structure, Parameters
- **EA Parameter Setting**
  - EA Parameter Tuning
  - EA Parameter Control
- **An Application to Agile Manufacturing**
  - Object-level Representation and Complexity
- **Solution**
  - FLC KB
  - Statistical Experiments
  - Analysis and Summary of 1200 Experiments
- **Remarks**

EA Model



EA Structure

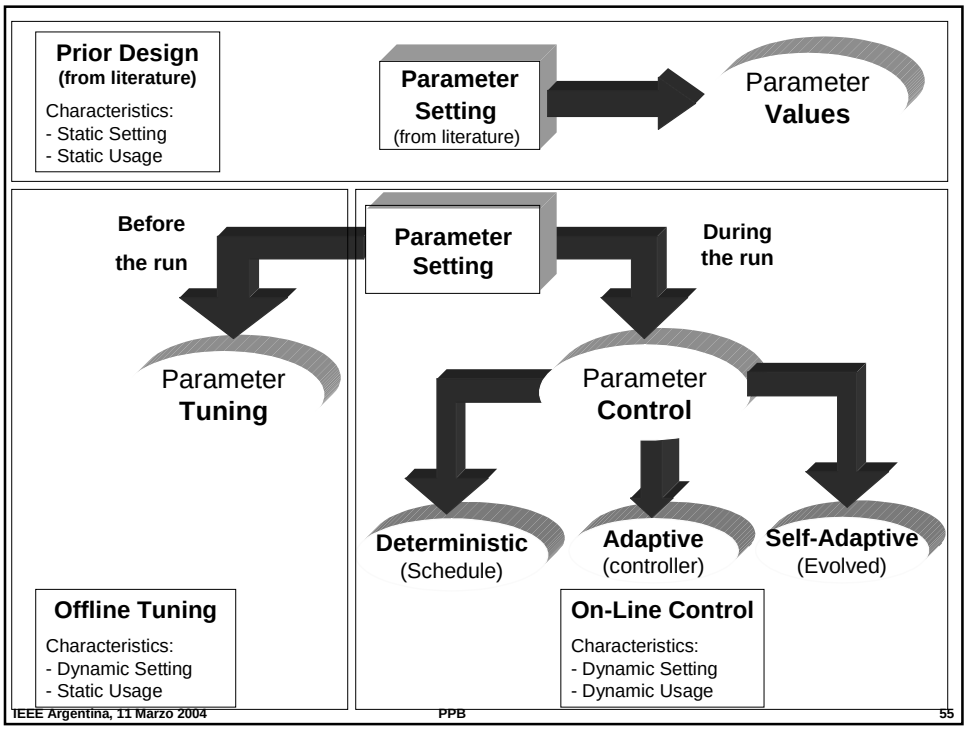
- **GA Structural Design Selections:**
  - GA Type:
    - »{Simple, Steady-State, Niche,…}
  - Chromosome Encoding:
    - »{Binary, Integer, Real,...}
  - Constraints Representation:
    - »{Penalty function, data structure, filters, …}
  - Fitness Function:
    - »{Scalar function, Weighted aggregation of multiple functions, Vector-valued function, …}

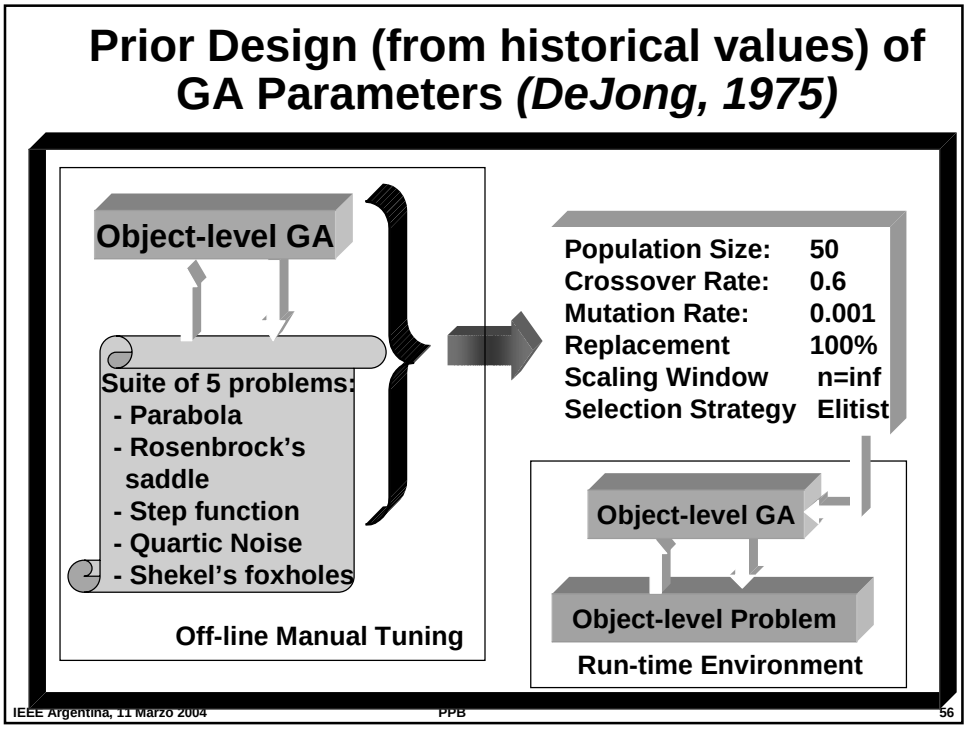
EA Parameters

- **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:

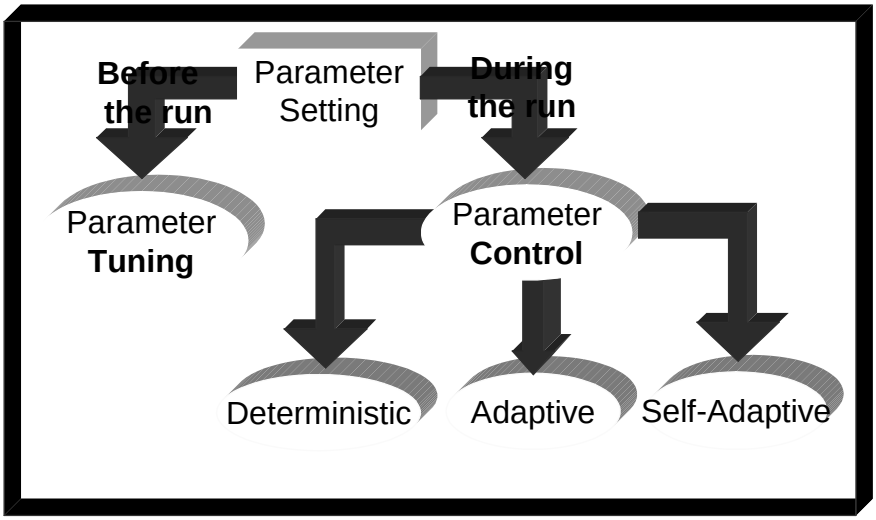
EA Parameter Setting - Outline

- **EA Model:**
  - Structure, Parameters
- **EA Parameter Setting**
  - EA Parameter Tuning
  - EA Parameter Control
- **An Application to Agile Manufacturing**
  - Object-level Representation and Complexity
- **Solution**
  - FLC KB
  - Statistical Experiments
  - Analysis and Summary of 1200 Experiments
- **Remarks**

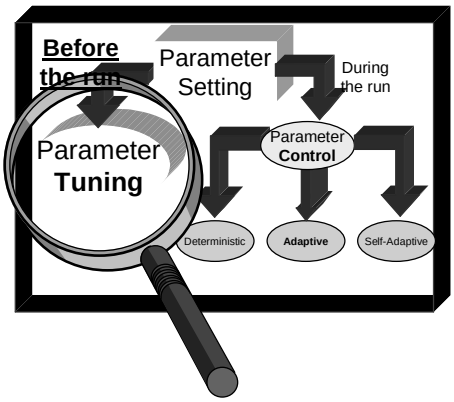

Prior Design
(from literature)
Characteristics:
- Static Setting
- Static Usage
Parameter Setting
(from literature)
Parameter Values
Before the run
Parameter Setting
During the run
Parameter Tuning
Parameter Control
Deterministic
(Schedule)
Adaptive
(controller)
Self-Adaptive
(Evolved)
Offline Tuning
Characteristics:
- Dynamic Setting
- Static Usage
On-Line Control
Characteristics:
- Dynamic Setting
- Dynamic Usage
IEEE Argentina, 11 Marzo 2004
PPB
55


Prior Design (from historical values) of GA Parameters (DeJong, 1975)
Object-level GA
Suite of 5 problems:
- Parabola
- Rosenbrock's saddle
- Step function
- Quartic Noise
- Shekel's foxholes
Off-line Manual Tuning
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
Run-time Environment
IEEE Argentina, 11 Marzo 2004
PPB
56

EAs Parameter Setting

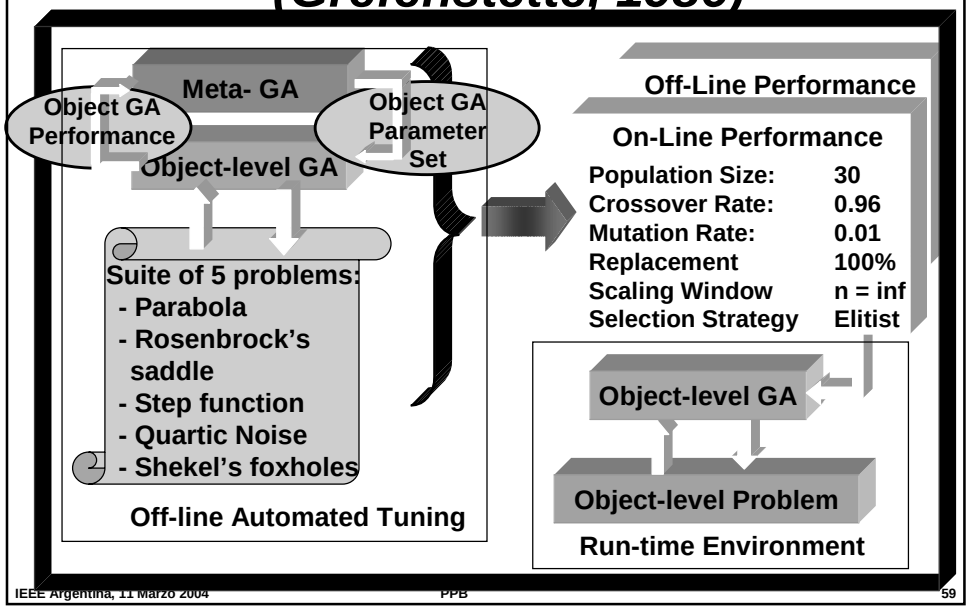


EAs Parameter Setting: Parameter Tuning

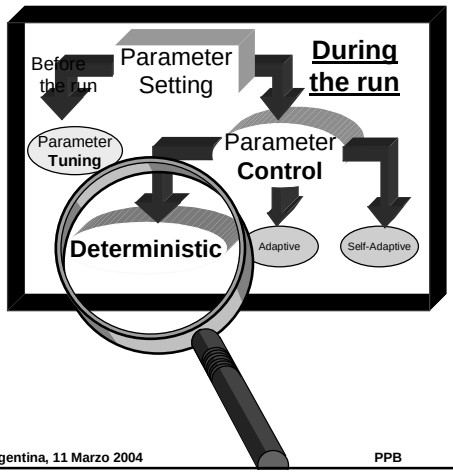


- ***Off-line Tuning***
- **Determined before running the GAs on the object-level problem by**
  - » Running a Meta-Genetic Algorithm *(Grefenstette, 1986)*

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



GAs Parameter Setting: Deterministic Control

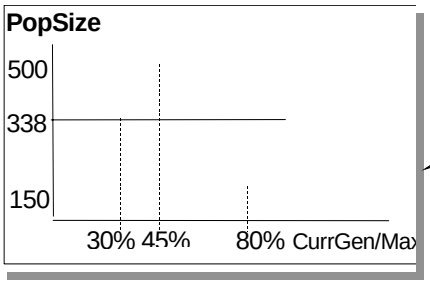


- ***No feedback*** **information is used.**
- **A time-varying schedule is used to modify a GA parameter *p***
- ***p is replaced by p(t)***
- **Correct design of p(t) is very difficult**

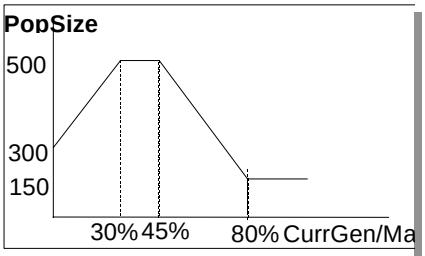
EAs Parameter Setting: Deterministic Control - Example

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)

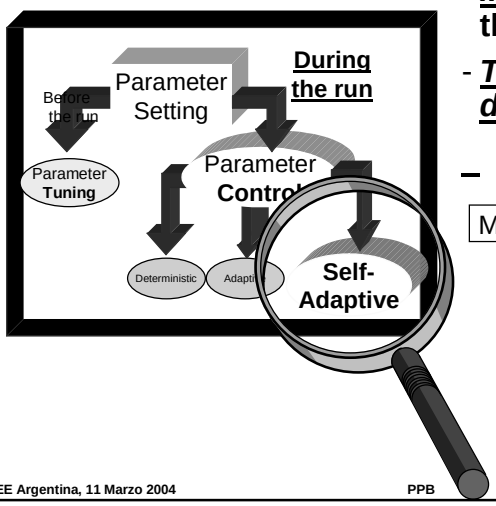


- Constant Population size: N = 338
- Number of trials = 338 * MaxGen



- Variable Population size: N(t)
- Number of trials = 338 * MaxGen

EAs Parameter Setting: Self-Adaptive Control



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

$$[g_1\ g_2 \ldots g_n\ S]$$

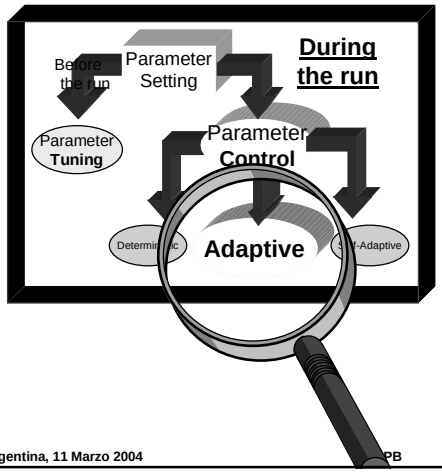
Mutation Step for Entire Genome

or

$$[g_1\ g_2 \ldots g_n\ S_1\ S_2 \ldots S_n]$$

Mutation Steps for Each Genome Value

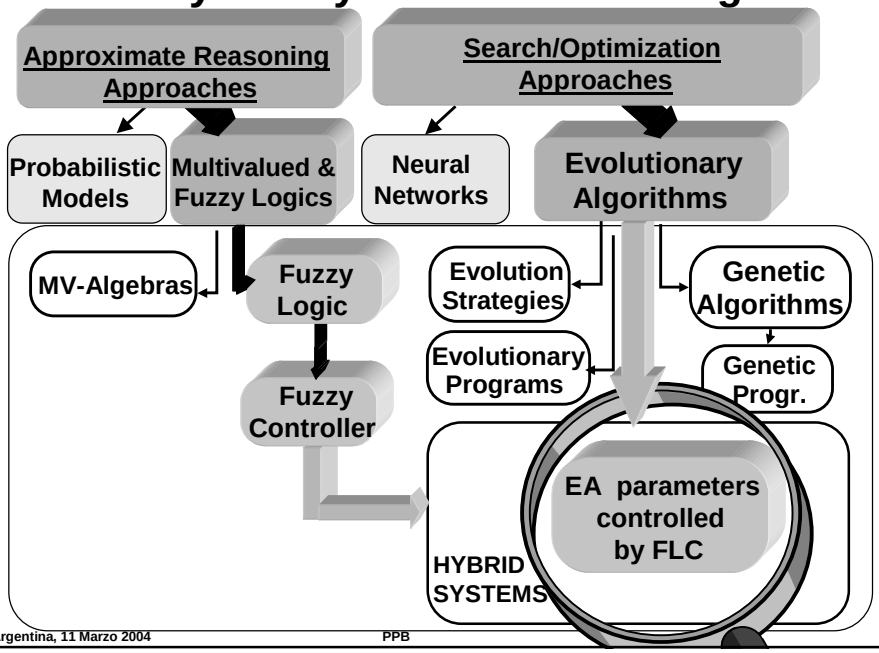
GAs Parameter Setting: Adaptive Control



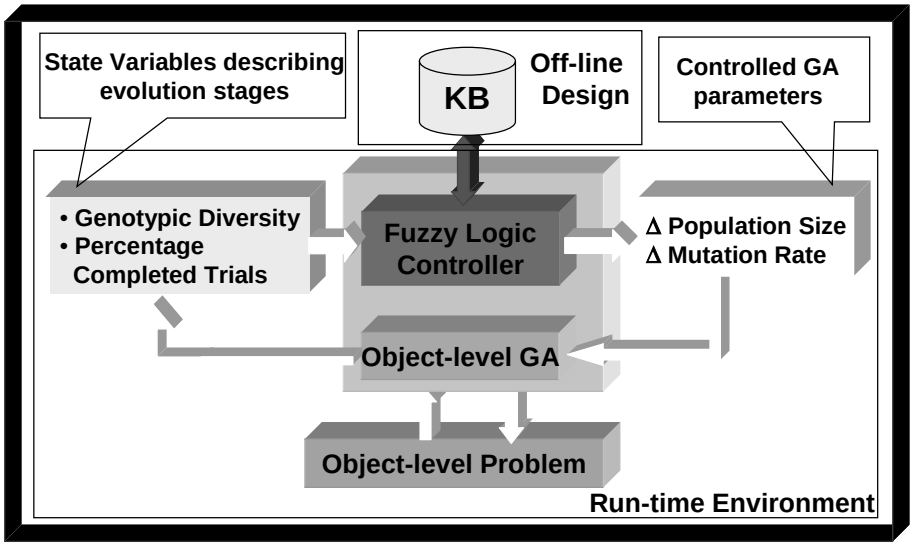
- ***Feedback*** **from the search is used to determine the direction and/or magnitude of the change in the parameter value.**
- ***A Fuzzy Logic Controller*** **is used to obtain parameter changes in**
  - » Population Size
  - » Mutation Rate

  as a function of
  - » Genotypic Diversity
  - » Percentage Completed Trials

SC Hybrid Systems: FLC Tuning EA



Fuzzy Logic Controlled GA (FLC-GA)

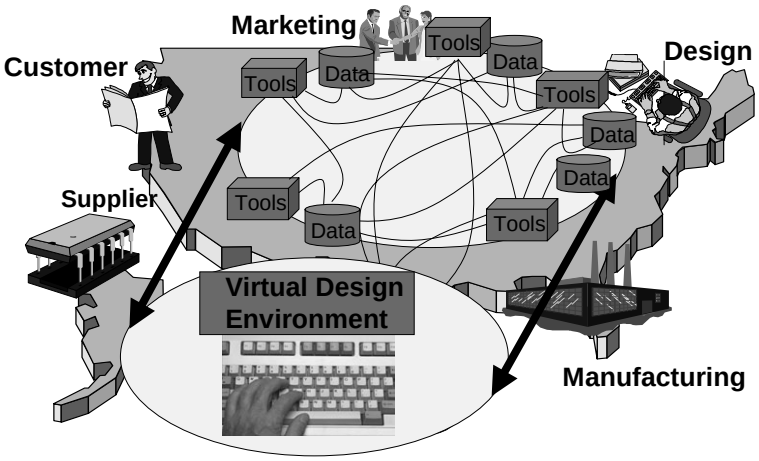


EA Parameter Setting

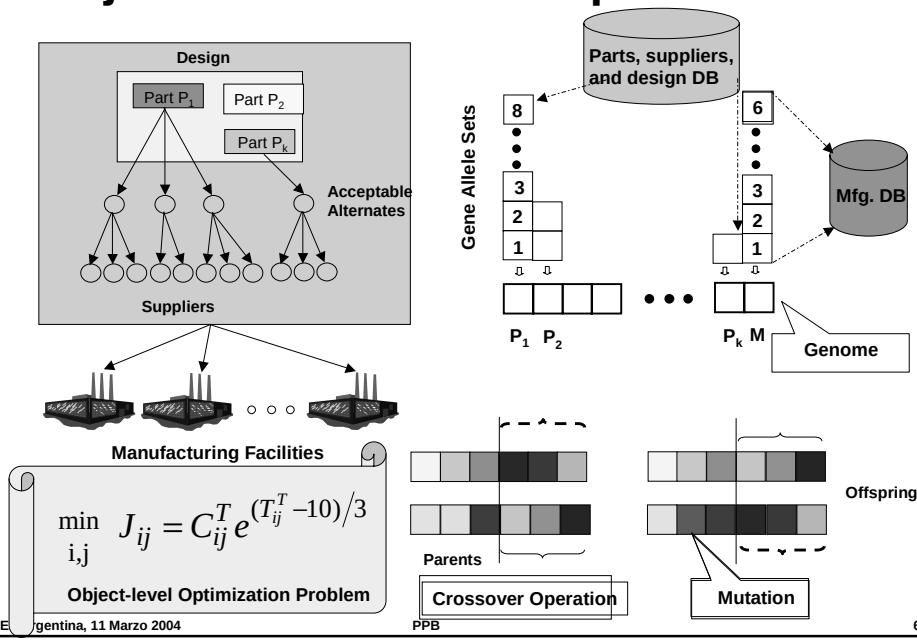
- **EA Model:**
  - Structure, Parameters
- **EA Parameter Setting**
  - EA Parameter Tuning
  - EA Parameter Control
- **An Application to Agile Manufacturing**
  - Object-level Representation and Complexity
- **Solution**
  - FLC KB
  - Statistical Experiments
  - Analysis and Summary of 1200 Experiments
- **Remarks**

EA Parameter Control: An Application

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



Object-level Problem Representation



Object-level Problem Complexity

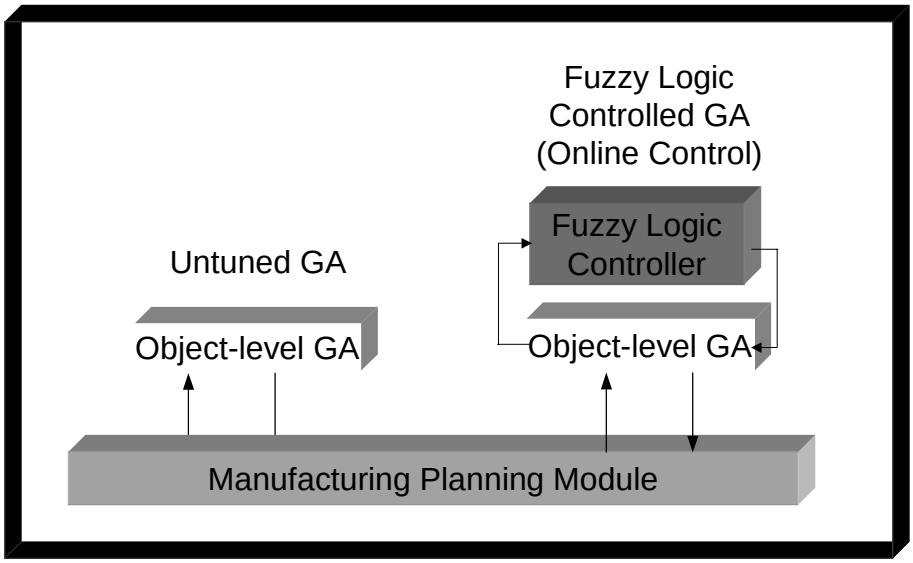
Search Space Size

- **For EA Statistical Analysis: $O(10^7)$**
- **For EA Performance Validation: $O(10^{18})$ and $O(10^{21})$**

EA Parameter Setting - Outline

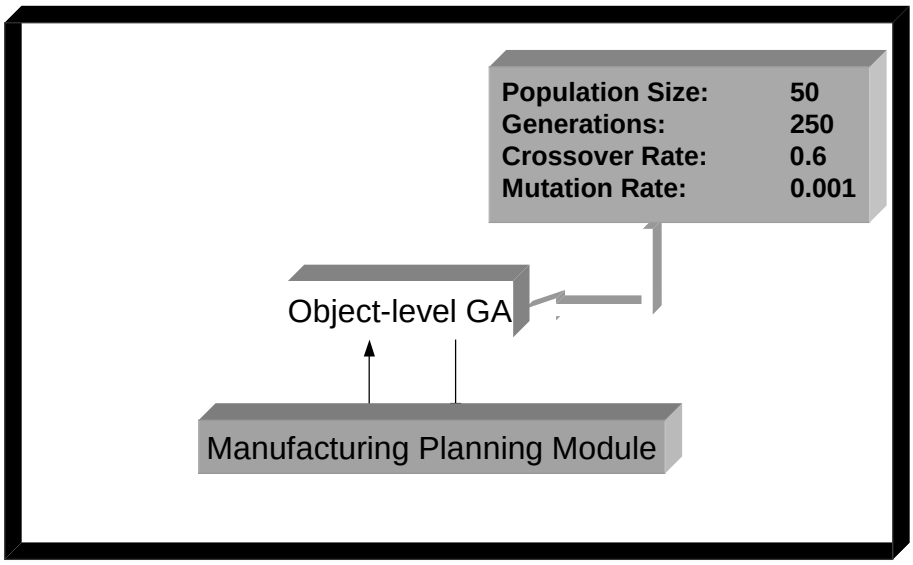
- **EA Model:**
  - Structure, Parameters
- **EA Parameter Setting**
  - EA Parameter Tuning
  - EA Parameter Control
- **An Application to Agile Manufacturing**
  - Object-level Representation and Complexity
- **Solution**
  - FLC KB
  - Statistical Experiments
  - Analysis and Summary of 1200 Experiments
- **Remarks**

Solution Architecture



IEEE Argentina, 11 Marzo 2004 PPB 71

Untuned GA (U-TGA)



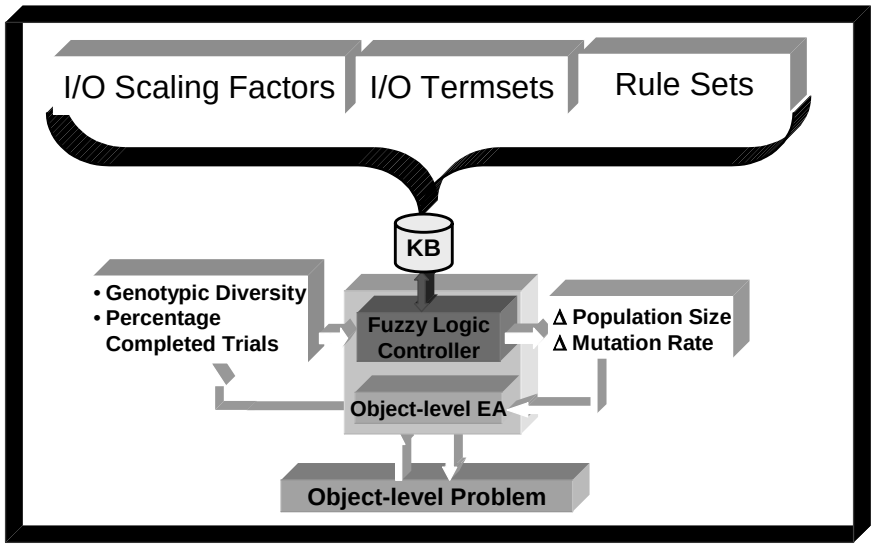
IEEE Argentina, 11 Marzo 2004 PPB 72

Guidance for Experiments

- ***Minimize high-level search space size for FLC-EA by***
  - ***Identify primary drivers (influences) of EA 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
- ***Determining FLC firing rate***
  - ***Take a control* action every 10 generation**
- ***Extensive & statistically significant empirical evidence***
  - ***Use t-test and F-tests to analyze μ and σ improvements***

Fuzzy Logic Controller for EAs: 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

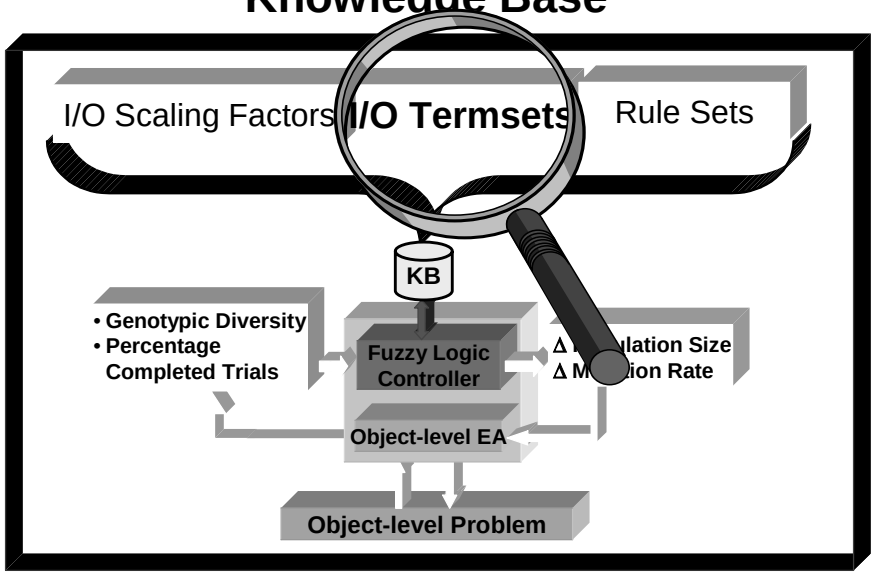
Population Size is *clamped* within [25, 150]

ΔP_m = Change in Mutation Rate (Mult. Factor)

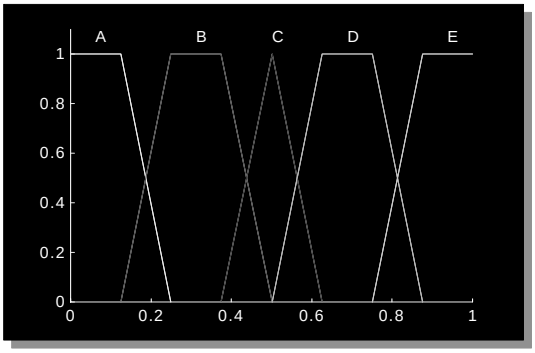
Δ Pm range is [0.5, 1.5] == [Neg High, Pos High]
so that NC corresponds to 100% of previous Pm

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

Fuzzy Logic Controller for EAs: Knowledge Base

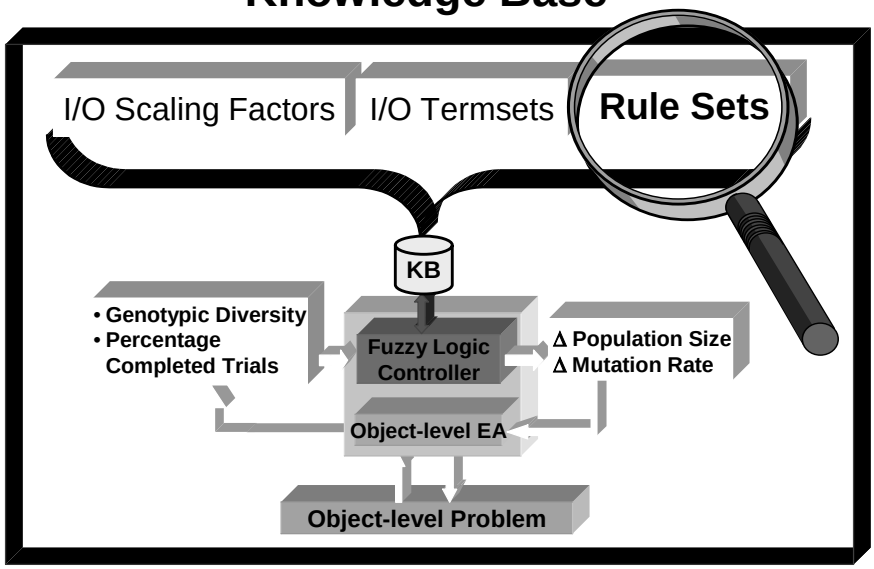


Fuzzy Controller for ΔN and ΔP_m: Termsets



- ***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 Logic Controller for EAs: Knowledge Base



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 Diversity (GD)	Percentage Fitness Evaluation (PFE)				
	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: EA Structure

- **Data Set for Experiments**
  - Seven part classes corresponding to a complexity of $O(10^7)$
- **EA 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

Statistical Experiments: Set-Up

- We defined 4 EA configurations
  - (a) Untuned Simple EA (U-SEA)
  - (b) FL Controlled Simple EA (FLC-SEA)
  - (c) Untuned Steady State EA (U-SSEA)
  - (d) FL Controlled Steady State EA (FLC-SSEA)
- 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 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 configurations (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

Statistical Experiments: Analysis

- **We performed an ANOVA test (both t and F test - with $p < 0.05$) to see if:**
  - **Cost (U-SEA) >> cost (FLCSEA)**
  - **Cost (U-SEA) >> cost (U-SSEA)**
  - **Cost (U-SSEA) >> cost (FLC-SSEA)**
- **We verified if the FLC caused the controlled EA to perform worse than its corresponding untuned EA, i.e.:**
  - **Cost (U-SEA) << cost (FLC-SEA)**
  - **Cost (U-SSEA) << cost (FLC-SSEA)**

Summary of 1200 Experiments

J = C*T

Max Numb Trials	U-SGA B	U-SGA μ	U-SGA σ	U-SGA σ/μ	FLC-SGA B	FLC-SGA μ	FLC-SGA σ	FLC-SGA σ/μ	U-SSGA B	U-SSGA μ	U-SSGA σ	U-SSGA σ/μ	FLC-SSGA B	FLC-SSGA μ	FLC-SSGA σ	FLC-SSGA 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*T²

Max Numb Trials	U-SGA B	U-SGA μ	U-SGA σ	U-SGA σ/μ	FLC-SGA B	FLC-SGA μ	FLC-SGA σ	FLC-SGA σ/μ	U-SSGA B	U-SSGA μ	U-SSGA σ	U-SSGA σ/μ	FLC-SSGA B	FLC-SSGA μ	FLC-SSGA σ	FLC-SSGA 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 Trials	U-SGA B	U-SGA μ	U-SGA σ	U-SGA σ/μ	FLC-SGA B	FLC-SGA μ	FLC-SGA σ	FLC-SGA σ/μ	U-SSGA B	U-SSGA μ	U-SSGA σ	U-SSGA σ/μ	FLC-SSGA B	FLC-SSGA μ	FLC-SSGA σ	FLC-SSGA 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: Results

J = Cost Function	U-SEA B	U-SEA μ	U-SEA σ	FLC-SEA B	FLC-SEA μ	FLC-SEA σ	U-SSEA B	U-SSEA μ	U-SSEA σ	FLC-SSEA B	FLC-SSEA μ	FLC-SSEA σ
C*T	-	-	-	-	4	3	-	-	-	-	1	2
C*T²	-	-	-	-	2	3	-	-	-	-	1	3
C*e^(T-10)/3	-	1	-	-	-	3	-	1	-	-	1	2
Total		7%			47%	60%		7%			20%	47%

Significant changes in μ and in σ

Significant changes in σ

- For each cost function we ran 400 experiments (100 x EA type)
- For each EA type we ran 20 experiments for 5 different pop. sizes
- The entry in each cell is the number of significant changes found in the statistics of each of these five groups of experiments

EA Parameter Setting

- **EA Model:**
  - Structure, Parameters
- **EA Parameter Setting**
  - EA Parameter Tuning
  - EA Parameter Control
- **An Application to Agile Manufacturing**
  - Object-level Representation and Complexity
- **Solution**
  - FLC KB
  - Statistical Experiments
  - Analysis and Summary of 1200 Experiments
- **Remarks**

Remarks

- **<u>FLC State Representation</u>: [Evolution Stage]**
  - **Evolution time needs to be an explicit state variable since we have different control goals during the EA's stages.**
  - **Diversity measures the evolutionary stage:**
    - » Percentage Fitness Evaluations (PFE)
    - » Genotypic Diversity (GD)
- **<u>FLC Control Variables:</u> [EA Adaptable Param.]**
  - **ΔN = Change in Population Size**
  - **ΔP_m = Change in Mutation Rate**

Remarks (cont.)

- **Main Result**

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

NFL, Meta-Heuristics & Hybrid SC: Outline

- The NFL
- Tuning or Controlling the Object-Level Problem Solver (PS) with Meta-Heuristics
- Soft Computing Overview
  - SC Components: PR, FL, NN, EA
- Using SC to implement the Meta-Heuristics: Modeling with FL and EA
- Example of Hybrid SC Systems
  - FLC Parameter Tuning by EA
  - FLR and CBR Parameter Tuning by EA
  - EA Parameter Setting (by EA) or Control (by FL)

- **Conclusions**

Knowledge and the NFL

- We **need to exploit domain knowledge**, embedding it in meta-heuristics, to tune or control object-level problem solvers, as we try to overcome some theoretical limitations derived from the NFLT.
- ***Knowledge is power*** was a common AI slogan of the eighties.
- ***Meta-knowledge is power*** is still quite relevant nowadays, as we try to overcome some theoretical limitations derived from the No Free Lunch Theorems.
- ***Embed meta-knowledge internally, in the algorithm's data structure***
- ***Embed meta-knowledge externally, using Meta-Heuristics***

Meta-Heuristics and the NFLT (cont.)

- **The use of meta-heuristics will improve the performance of optimization algorithms for a subset of problems**
- **The meta-heuristics will not be universal, but specific for a problem or a class of problems.**
  - **Therefore, even the knowledge base (KB) used by the online adaptation scheme cannot be considered of general applicability**
- ***Two types of Meta-heuristics:***
  - ***On-line*** heuristics when we want to generate run-time corrections for the behavior of the object-level problem solver
  - ***Offline*** meta-heuristics when we want to define the best structural and/or parametric configuration for the model that is working on the object-level task

Meta-Heuristics and SC

- **If *meta-knowledge is power*,**
  then *proper meta-knowledge representation is key*
- **Soft computing is a new paradigm that provides a natural framework to represent such knowledge.**
  - Leverages ***tolerance for imprecision***, uncertainty, and incompleteness - intrinsic to the problems to be solved
  - Generates *tractable, low-cost,* ***robust*** solutions to such problems ***by integrating knowledge and data***
- **Modeling Meta-heuristics with SC**
  - ***Data-driven*** Tuning of Knowledge-derived Models
    - » Translate domain knowledge into initial structure, parameters, encoding, variational operators
    - » Use global or local data search to tune parameters
  - ***Knowledge-driven*** Search Control
- **Use global or local data search to derive models (Structure + Parameters)**
    - » Translate domain knowledge into an algorithm's controller to improve/manage solution convergence and quality

Increased efficiency achieved by object-level problem solvers when guided by *offline* or *on-line* meta-heuristics that leverage the relevant meta-knowledge.