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

<table>
<thead>
<tr>
<th></th>
<th>X1</th>
<th>X2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Very Low</strong></td>
<td>PH</td>
<td>PH</td>
</tr>
<tr>
<td><strong>Low</strong></td>
<td>PM</td>
<td>PL</td>
</tr>
<tr>
<td><strong>Medium</strong></td>
<td>PL</td>
<td>ZE</td>
</tr>
<tr>
<td><strong>High</strong></td>
<td>ZE</td>
<td>NL</td>
</tr>
<tr>
<td><strong>Very High</strong></td>
<td>NL</td>
<td>NM</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>X1</th>
<th>X2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Very Low</strong></td>
<td>PH</td>
<td>PH</td>
</tr>
<tr>
<td><strong>Low</strong></td>
<td>PM</td>
<td>PL</td>
</tr>
<tr>
<td><strong>Medium</strong></td>
<td>PL</td>
<td>ZE</td>
</tr>
<tr>
<td><strong>High</strong></td>
<td>ZE</td>
<td>NL</td>
</tr>
<tr>
<td><strong>Very High</strong></td>
<td>NL</td>
<td>NM</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>X1</th>
<th>X2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Very Low</strong></td>
<td>PH</td>
<td>PH</td>
</tr>
<tr>
<td><strong>Low</strong></td>
<td>PM</td>
<td>PL</td>
</tr>
<tr>
<td><strong>Medium</strong></td>
<td>PL</td>
<td>ZE</td>
</tr>
<tr>
<td><strong>High</strong></td>
<td>ZE</td>
<td>NL</td>
</tr>
<tr>
<td><strong>Very High</strong></td>
<td>NL</td>
<td>NM</td>
</tr>
</tbody>
</table>

### 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 (f₁, f₂, f₃)**

\[
\begin{align*}
    f_1 &= \min(\sum_i |\text{notch}_i - \text{notch}_{i-1}| + |\text{dynbrake}_i - \text{dynbrake}_{i-1}|) \\
    f_2 &= \min(\sum_i |v_i - v_{i}^{d}|) \\
    f_3 &= \min(w_1 \frac{\sum_i |\text{notch}_i - \text{notch}_{i-1}|}{K_1} + w_2 \frac{\sum_i |v_i - v_{i}^{d}|}{K_2})
\end{align*}
\]
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 $K_e$, $K_{edot}$, $K_{\Delta u}$
  - MF $(21-9) = 12$ values
    - 21 parameters: $[(\text{Left}_i, \text{Center}_i, \text{Right}_i) \text{ for } i=1, ..., 7]$
    - 9 dependent values: $[(\text{Left}_i = \text{Right}_{i+1}) \text{ for } i=1, ..., 6]$
      + $[\text{Center}_1 = \text{Center}_7] + [\text{Right}_1 = \text{Left}_7 = 0]$
  - Constraints to maintain 0.5 terms overlap, for best interpolation

Experiments Results

- **Experiment Results with $f_1$**

<table>
<thead>
<tr>
<th>Description</th>
<th>Time</th>
<th>Journey</th>
<th>Fuel</th>
<th>Fitness</th>
<th>Gen.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial SF; Initial MF</td>
<td>26.5</td>
<td>14.26</td>
<td>878</td>
<td>73.2</td>
<td></td>
</tr>
<tr>
<td>EA tuned SF; Initial MF</td>
<td>27.8</td>
<td>14.21</td>
<td>857</td>
<td>15.15</td>
<td>4</td>
</tr>
<tr>
<td>Initial SF; EA tuned MF</td>
<td>26.00</td>
<td>14.18</td>
<td>879</td>
<td>70.93</td>
<td>20</td>
</tr>
<tr>
<td>EA tuned SF; EA tuned MF</td>
<td>28.3</td>
<td>14.12</td>
<td>823</td>
<td>14.64</td>
<td>10</td>
</tr>
</tbody>
</table>

- **Experiment Results with $f_3$**

<table>
<thead>
<tr>
<th>Description</th>
<th>Time</th>
<th>Journey</th>
<th>Fuel</th>
<th>Fitness</th>
<th>Gen.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial SF; Initial MF</td>
<td>26.5</td>
<td>14.26</td>
<td>878</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>EA tuned SF; Initial MF</td>
<td>27.2</td>
<td>14.35</td>
<td>871</td>
<td>0.817</td>
<td>4</td>
</tr>
<tr>
<td>Initial SF; EA tuned MF</td>
<td>26.26</td>
<td>14.18</td>
<td>871</td>
<td>0.942</td>
<td>20</td>
</tr>
<tr>
<td>EA tuned SF; EA tuned MF</td>
<td>27.3</td>
<td>14.35</td>
<td>872</td>
<td>0.817</td>
<td>10</td>
</tr>
</tbody>
</table>
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

![Graphs showing performance comparison between manually and GA tuned controllers on a simulated 43 mile track.](image)
Results of EA Tuned PI on 43 mile Track

Manually tuned controller

GA tuned controller

NOTCH POSITION

Mile

Results of EA Tuned PI on 43 mile Track
Results of EA Tuned PI on 43 mile Track

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
Off-line Meta-Heuristics for Digital Underwriting

Evolutionary Alg.

Performance of Classifier

FLE or F-CBR Classifier

Design Parameters of Classifier

Run-time Parameters for FLE or F-CBR Classifier

FLE or F-CBR Classifier

Underwriting Problem

Run-time Environment

3,000 cases with Standard Ref. Decisions

Off-line Tuning

Example: Digital Underwriting for GE Insurance

Developing an automated system to underwrite ~50% of GEFA Fixed Life Applications

Past

New Client

Data Collection

Critical X’s

Underwriter

Processes Info

Makes Decision

Y_1 = Preferred

Y_2 = Preferred

Y_3 = Standard

Y_n = Declined

Present

Data Base

Captures Ys and Associated Xs

Underwriter

Makes Decision

Y_1 = Preferred

Y_2 = Preferred

Y_3 = Standard

Y_n = Declined

GE Proprietary models provide uniformity & confidence in decision
Path 1: Clean Cases (Rule Based)

- 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
  - $MS_1 (r) = \min_{i=1}^{n} C_{i,r} (x_i)$
  - $MS_2 (r) = \sum_{i=1}^{n} [1 - C_{i,r} (x_i)] = n - nC_r = n(1 - C_r)$
- Compare $MS_1 (r)$ and $MS_2 (r)$ against their corresponding thresholds $\tau_1$ and $\tau_2$
- Assign the most competitive rate class for which $MS_1 (r) \leq \tau_1$ and $MS_2 (r) \leq \tau_2$
- Parametric representation of $C_{i,r} (x_i) = (a_{i,r}, b_{i,r})$

Fuzzy Logic Engine (FLE)

IEEE Argentina, 11 Marzo 2004  PPB

Page 12
**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)
    \[
    MS_2(r) = \sum_{i=1}^n [1 - C_{i,r}(x_i)] = n - nC_r = n(1 - C_r)
    \]
- Compare \( MS_1(r) \) and \( MS_2(r) \) against their corresponding thresholds \( \tau_1 \) and \( \tau_2 \)
- Assign the most competitive rate class for which \( MS_1(r) \leq \tau_1 \) and \( MS_2(r) \leq \tau_2 \)
- Parametric representation of \( C_{i,r}(x_i) = (a_{i,r}, b_{i,r}) \)

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

---

**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)

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

Fuzzy Case Based Engine (FCBE) - Step 1:
Represent New Case with an SQL Query

\[ X_{\text{New, Case}} = [7.5, 3.8, \ldots] \text{ Impairment=} \text{Hypertension} \]

\[ Q1: [f1(x_1), f2(x_2), \ldots, fn(x_n)] \text{ AND } \text{Impairment=} \text{label} \]

For instance:
\[ Q1: \text{ [Support(Around (7.5;x_1)), Support(Around (3.8;x_2)), } \]
\[ \text{ Support(NORMAL(X_1)), \ldots, Support(NORMAL(X_n))]} \]
\[ \text{ AND } \text{Impairment=} \text{Hypertension} \]
Fuzzy Case Based Engine (FCBE) - Step 2:
Retrieve relevant cases from the case library

Retrieved N Cases \( C_i \)

where \( 1 \leq i \leq N \)

\[
C_i = [7.4, 4.3, ...] \text{ Impairment} = \text{Hypertension}
\]

Table VI

\[
C_i = [7.2, 3.3, ...] \text{ Impairment} = \text{Hypertension}
\]

Table II

\[
C_i = [7.9, 4.1, ...] \text{ Impairment} = \text{Hypertension}
\]

Table IV

\[ \Delta \text{Weight/Height} \]

\[ \text{Systolic BP} \]

\[ \text{Cholesterol Ratio} \]

\[ \text{NEW_CASE} \]

\[ \text{STANDARD} \]

\[ \text{TABLE II} \]

\[ \text{TABLE IV} \]

\[ \text{TABLE VI} \]

\[ \text{TABLE VIII} \]

Distribution of N Retrieved Cases

Fuzzy Case Based Engine (FCBE) - Step 3:
Evaluate and Rank the retrieved cases - using similarity measure

Selection of MF family

\[ \text{Trap}(x; a, b, c, d) = \text{Trap}(x; 10, 20, 60, 100) \]

\[ \text{Sq Trap}(x; a, b, c, d) = \text{Sq Trap}(x; 10, 20, 60, 100) \]

\[ \text{GBF}(x; a, b, c) = \text{GBF}(x; 15, 3, 50) \]

Thresholded GBF:

\[
\text{GBF}(x; a, b, c) = \begin{cases} 
\frac{1}{1 + \left| \frac{x - c}{a} \right|^b} & \text{if } \text{GBF} > 10^{-3} \\
0 & \text{otherwise}
\end{cases}
\]

- The membership value is a function of the inverse of the distance respect of the reference (probe) upon which the membership is centered.
- A linear slope compensates too much (One close point would be averaged out with 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.
Generalized Bell Function: Effect of changing Parameters \( \{a, b, c\} \)

\[
\mu_x(c) = \frac{1}{1 + \left(\frac{x - c}{a}ight)^2}
\]

- (a) Changing 'a'
- (b) Changing 'b'
- (c) Changing 'c'
- (d) Changing 'a' and 'b'

Fuzzy Case Based Engine (FCBE) - Step 3:
Evaluate and Rank the retrieved cases - using similarity measure

\[ C_N = [7.9, 4.1, \ldots] \text{ AND Impairment=Hypertension} \]

\[ f_1: \text{Around}(7.5;x_1) \]

\[ f_2: \text{Around}(3.8;x_2) \]

\[ Q_i(C_N) = [0.65, 0.6, \ldots] \text{ AND } [1] \]

\[ S(C_N, Q_i) = 0.6 \]

\[ S = \cap_{i=1}^n GBF(x_i; a_i, b_i, c_i) = \min_{i=1}^n GBF(x_i; a_i, b_i, c_i) \]
Fuzzy Case Based Engine (FCBE) - Step 4:

Select best cases

- Histogram of N retrieved cases
- Preferred, Best, Standard, Table IV, Table VIII
- Weighed Histogram
- Preferred, Best

Fuzzy Case Based Engine (FCBE) - Step 5:

5. Derive solution for current case from (adapted) retrieved solutions

- Histogram Aggregation (e.g., Mode, First Moment, Median)
- Design configuration of FCBE \[ (a_i, b_i), \ldots \] \( i=1, n \)
Fuzzy Case Based Engine (CBE) - Step 6a:

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

---

**Step 6a:**

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

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

1. **Selected Case**
2. **Generate Histogram & Decision**
3. **K-1 Past Applications with Reference Rate Class**
4. **Case Base**
5. **Loop through all K cases in CB**
6. **Place probe back in CB and select another case**

---

**Confidence Function for cardinality of retrieved cases**

\[ f(Bin_i) = \sum_{j=1}^{k} w_j D(i,k) \]

**Similar to fitness function used in optimization**

---

**Confidence Function**

**X**: Ranges of values of \( N \)

Distance | Number of retrieved cases for each \( N \) |
---|---|
\([-3,-2,-1,0,1,2,3]\) | |
**Total % within bins**

---

**IEEE Argentina, 11 Marzo 2004 35PPB**

---

**IEEE Argentina, 11 Marzo 2004 36PPB**

---

**Page 18**
Fuzzy Case Based Engine (CBE) - Step 6b:
Evaluate solution’s quality - Generate a Confidence Factor (on-line)

Parameter Vector $\mathbf{X}$:
\[ [39, \ldots, 73\%] \]

Soft Constraints Evaluation: $SCE = [0.85, \ldots, 0.77]$  
Confidence Factor: $CF = Min(SCE) = 0.77$

IEEE Argentina, 11 Marzo 2004

RBDE Parameter Tuning

Decision Engine Parameter Tuning/Optimization

Evolutionary Optimization Algorithm

Create initial population $\langle \mathbf{x}_i \rangle$

Proportional Selection

Intermediate population $\langle \mathbf{x}_i \rangle' = S(\mathbf{x}_i)$ (exploitation)

Stochastic Variation

New population $\langle \mathbf{x}_{i+1} \rangle = V(\mathbf{x}_i)'$ (exploration)

Problem Definition

\[ \min_{\mathbf{x} \in \chi} \psi(\mathbf{x}) \quad \chi \subseteq \mathbb{R}^n \quad \psi: \chi \rightarrow \mathbb{R} \]

Space of $n$ dimensional positive real vectors

More superior population of trial solutions

Space of decision engine designs

Space of positive real numbers

Decision mismatch penalty function
Decision Engine Parameter Tuning/Optimization

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

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

Evaluation Process

- Decision Engine Instance → DE Decision
- Case Base → Past Validated Cases

Decision Engine Parameter Tuning/Optimization: Matrix M

Computation of Rate Class Decision Mismatch Penalties

\[ \text{Score} (k) = \psi (Ind_k) \sum_i \sum_{j 
eq i} P(i,j) M(j,i) \]

Decision Comparison: Correct/Incorrect

Loop through all cases in Case Base

Score Table:

<table>
<thead>
<tr>
<th>Penalty</th>
<th>PB</th>
<th>P</th>
<th>S+</th>
<th>S</th>
<th>S-</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stricter</td>
<td>1.4%</td>
<td>33.0%</td>
<td>...</td>
<td>51.5%</td>
<td>...</td>
</tr>
<tr>
<td>More Liberal</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>4.5%</td>
</tr>
</tbody>
</table>
Penalty Matrix P

- Used Actuarial studies to develop 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

\[
P = \begin{pmatrix}
A & B & C & D & E & \\
\$0 & -$3 & -$11 & -$15 & & \\
\$-6 & \$0 & -$5 & & & \\
\$-18 & & $0 & & & \\
\$-22 & & & & & \\
\$-26 & & & & & \\
\end{pmatrix}
\]

- 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

- Chromosome Decoder
- Mutation
- Fitness
- Selection
- Elitist
- Original and Best
- Uniform Mutation
- Gaussian Mutation
- Fitness Function: Quantify Quality of Chromosome
  \[f(Ind_j) = \sum_{i=1}^{T} \sum_{j=1}^{T} P(i, j)M(i, j)\]

- XML Config.
- File
- Instance of Fuzzy Logic Engine
- Case
- FLE Decision
- FUZZY RULE EVALUATION
- EVOLUTIONARY ALGORITHM

- Standard Reference Data Set
- Engine Parameters
**Case Based Reasoning Optimization**

**EVOLUTIONARY ALGORITHM**

- Chromosome Decoder
- Mutation
  - Uniform Mutation
  - Elitist Mutation
- Fitness Function: Quantify Quality of Chromosome
  \[ f(\text{Ind}_i) = \sum_{j=1}^{T} k(i,j)M(i,j) \]

**CBR EVALUATION**

XML Config File

**RBE Parameter Tuning via Optimization**

**RBE Optimization Results (Before and After)**

<table>
<thead>
<tr>
<th></th>
<th>Coverage (not UW)</th>
<th>Relative Accuracy (no UW)</th>
<th>Global Accuracy with UW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manually Tuned</td>
<td>90.30%</td>
<td>2539 out of 2920</td>
<td>90.30%</td>
</tr>
<tr>
<td>GEFA</td>
<td>90.30%</td>
<td>2539 out of 2920</td>
<td>90.30%</td>
</tr>
</tbody>
</table>

**Rule-Based Engine Decision**

<table>
<thead>
<tr>
<th></th>
<th>Coverage (not UW)</th>
<th>Relative Accuracy (no UW)</th>
<th>Global Accuracy with UW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct (UW)</td>
<td>90.30%</td>
<td>2539 out of 2920</td>
<td>90.30%</td>
</tr>
<tr>
<td>PB</td>
<td>90.35%</td>
<td>2539 out of 2920</td>
<td>90.30%</td>
</tr>
<tr>
<td>P</td>
<td>90.30%</td>
<td>2539 out of 2920</td>
<td>90.30%</td>
</tr>
<tr>
<td>Sel</td>
<td>90.30%</td>
<td>2539 out of 2920</td>
<td>90.30%</td>
</tr>
<tr>
<td>Std</td>
<td>90.30%</td>
<td>2539 out of 2920</td>
<td>90.30%</td>
</tr>
<tr>
<td>Std+Nic</td>
<td>90.30%</td>
<td>2539 out of 2920</td>
<td>90.30%</td>
</tr>
<tr>
<td>Std+Nic</td>
<td>90.30%</td>
<td>2539 out of 2920</td>
<td>90.30%</td>
</tr>
<tr>
<td>Std Nic</td>
<td>90.30%</td>
<td>2539 out of 2920</td>
<td>90.30%</td>
</tr>
<tr>
<td>UW</td>
<td>90.30%</td>
<td>2539 out of 2920</td>
<td>90.30%</td>
</tr>
</tbody>
</table>

**Rule-Based Engine Decision**

<table>
<thead>
<tr>
<th></th>
<th>Coverage (not UW)</th>
<th>Relative Accuracy (no UW)</th>
<th>Global Accuracy with UW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct (UW)</td>
<td>90.30%</td>
<td>2539 out of 2920</td>
<td>90.30%</td>
</tr>
<tr>
<td>PB</td>
<td>90.30%</td>
<td>2539 out of 2920</td>
<td>90.30%</td>
</tr>
<tr>
<td>P</td>
<td>90.30%</td>
<td>2539 out of 2920</td>
<td>90.30%</td>
</tr>
<tr>
<td>Sel</td>
<td>90.30%</td>
<td>2539 out of 2920</td>
<td>90.30%</td>
</tr>
<tr>
<td>Std</td>
<td>90.30%</td>
<td>2539 out of 2920</td>
<td>90.30%</td>
</tr>
<tr>
<td>Std+Nic</td>
<td>90.30%</td>
<td>2539 out of 2920</td>
<td>90.30%</td>
</tr>
<tr>
<td>Std+Nic</td>
<td>90.30%</td>
<td>2539 out of 2920</td>
<td>90.30%</td>
</tr>
<tr>
<td>Std Nic</td>
<td>90.30%</td>
<td>2539 out of 2920</td>
<td>90.30%</td>
</tr>
<tr>
<td>UW</td>
<td>90.30%</td>
<td>2539 out of 2920</td>
<td>90.30%</td>
</tr>
</tbody>
</table>

**The Optimization improved the global accuracy from 90.1% to 93.6%**
Evolutionary Tuning Results

Fuzzy Logic Engine

<table>
<thead>
<tr>
<th>Metric</th>
<th>Sub-Optimal Parameters</th>
<th>Optimized Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coverage</td>
<td>90.38%</td>
<td>91.71%</td>
</tr>
<tr>
<td>Relative Accuracy</td>
<td>92.99%</td>
<td>95.52%</td>
</tr>
<tr>
<td>Global Accuracy</td>
<td>90.07%</td>
<td>93.63%</td>
</tr>
</tbody>
</table>

Fuzzy Case-Based Engine

<table>
<thead>
<tr>
<th>Metric</th>
<th>Sub-Optimal Parameters</th>
<th>Optimized Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coverage</td>
<td>47.97%</td>
<td>98.86%</td>
</tr>
<tr>
<td>Relative Accuracy</td>
<td>92.10%</td>
<td>90.80%</td>
</tr>
<tr>
<td>Global Accuracy</td>
<td>44.18%</td>
<td>89.77%</td>
</tr>
</tbody>
</table>

Coverage: 
# Decisions / # Cases

Relative Accuracy: 
#Correct Decisions / # Decisions

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

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

Five segmentations with disjoint validation sets each containing 20% of the original data
RBDE Parameter Tuning: Robustness
Conclusions

Coverage:

<table>
<thead>
<tr>
<th>Pair</th>
<th>Training Set</th>
<th>Validation Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>91.94%</td>
<td>92.11%</td>
</tr>
<tr>
<td>2</td>
<td>91.73%</td>
<td>90.39%</td>
</tr>
<tr>
<td>3</td>
<td>91.68%</td>
<td>93.14%</td>
</tr>
<tr>
<td>4</td>
<td>91.47%</td>
<td>92.97%</td>
</tr>
<tr>
<td>5</td>
<td>92.24%</td>
<td>90.41%</td>
</tr>
</tbody>
</table>

Relative Accuracy:

<table>
<thead>
<tr>
<th>Pair</th>
<th>Training Set</th>
<th>Validation Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>94.03%</td>
<td>93.85%</td>
</tr>
<tr>
<td>2</td>
<td>95.28%</td>
<td>93.93%</td>
</tr>
<tr>
<td>3</td>
<td>94.72%</td>
<td>95.21%</td>
</tr>
<tr>
<td>4</td>
<td>93.81%</td>
<td>92.80%</td>
</tr>
<tr>
<td>5</td>
<td>94.75%</td>
<td>92.23%</td>
</tr>
</tbody>
</table>

Global Accuracy:

<table>
<thead>
<tr>
<th>Pair</th>
<th>Training Set</th>
<th>Validation Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>92.41%</td>
<td>92.11%</td>
</tr>
<tr>
<td>2</td>
<td>93.66%</td>
<td>90.22%</td>
</tr>
<tr>
<td>3</td>
<td>92.97%</td>
<td>90.74%</td>
</tr>
<tr>
<td>4</td>
<td>91.98%</td>
<td>90.41%</td>
</tr>
<tr>
<td>5</td>
<td>92.67%</td>
<td>91.44%</td>
</tr>
</tbody>
</table>

- **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:
    \[(1 - \text{Global Accuracy})\]
  - 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
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 Model Diagram](image)

- **Object-level GA**
- **Object-level Problem**
- **Structure & Parameters**

### 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:
    - \( P_{cr} \times N = \# \) crossovers per generation
  - **P_m** = Mutation rate:
    - \( P_{m} \times N \times L = \# \) 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 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
EAs Parameter Setting

Parameter Tuning

Before the run

Parameter Setting

During the run

Parameter Control

Deterministic

Adaptive

Self-Adaptive

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)

Object GA Performance

Object-level GA Parameter Set

Meta-GA

Selection Strategy: NonElitist

Replacement: 90%

Population Size: 80

Crossover Rate: 0.45

Mutation Rate: 0.01

Scaling Window: n = 1

Object-level Problem

Object-level GA

On-Line Performance

Population Size: 30

Crossover Rate: 0.96

Mutation Rate: 0.01

Replacement: 100%

Scaling Window: n = inf

Selection Strategy: Elitist

Object-level GA

Run-time Environment

Object-level Parameter Setting

Off-line Automated Tuning

Suit of 5 problems:
- Parabola
- Rosenbrock’s saddle
- Step function
- Quartic Noise
- Shekel’s foxholes

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
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)

- Variable Population size: \( N(t) \)
- Number of trials = \( 338 \times \text{MaxGen} \)

EAs Parameter Setting: Self-Adaptive Control

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

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

or

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

- Approximate Reasoning Approaches
  - Probabilistic Models
  - Multivalued & Fuzzy Logics

- Search/Optimization Approaches
  - Neural Networks
  - Evolutionary Algorithms

- MV-Algebras
  - Fuzzy Logic
  - Fuzzy Controller

- Evolution Strategies
  - Evolutionary Programs

- Genetic Algorithms
  - Genetic Progr.

- EA parameters controlled by FLC

IEEE Argentina, 11 Marzo 2004

Page 32
Fuzzy Logic Controlled GA (FLC-GA)

State Variables describing evolution stages
- Genotypic Diversity
- Percentage Completed Trials

Off-line Design

Controlled GA parameters
- Population Size
- Mutation Rate

Run-time Environment

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

**Object-level Problem Representation**

\[
\min_{i,j} J_{ij} = C_{ij}^T e^{T_j^T(T_j^T - 10)/3}
\]

Object-level Optimization Problem

**Gene Allele Sets**

<table>
<thead>
<tr>
<th>Part P_1</th>
<th>Part P_2</th>
<th>Part P_k</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Mfg. DB**

**Genome**

**Parents**

**Offspring**

**Crossover Operation**

**Mutation**
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

- Untuned GA
- Object-level GA
- Manufacturing Planning Module

Fuzzy Logic Controlled GA (Online Control)

- Fuzzy Logic Controller
- Object-level GA

Untuned GA (U-TGA)

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

Object-level GA

Manufacturing Planning Module
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 $\mu$ and $\sigma$ improvements

Fuzzy Logic Controller for EAs: Knowledge Base
Fuzzy Controller for $\Delta N$ and $\Delta P_m$: Inputs

- **Inputs**
  - GD = Genotypic Diversity:
    - Normalized Average Hamming Distance
    
    \[
    GD = \frac{2}{n(n-1)} \sum_{i=1, j=i+1}^{n} d_{ij} \]
    
    where $d_{ij}$ is the Hamming Distance
    
    GD range is $[0, 1] = [\text{Low, High}]
  
  - PFE = Percentage Fitness Evaluations:
    
    \[
    \frac{\text{Completed # Trials}}{\text{Max Allocated # Trials}}
    \]
    
    PFE range is $[0, 1] = [\text{Low, High}]

Fuzzy Controller for $\Delta N$ and $\Delta P_m$: Outputs

- **Outputs**
  - $\Delta N$ = Change in Population Size (Mult. Factor)
    
    $\Delta N$ range is $[0.5, 1.5] = [\text{Neg High, Pos High}]
    
    so that NC corresponds to 100% of previous Pop Size
    
    Population Size is clamped within $[25, 150]$
  
  - $\Delta P_m$ = Change in Mutation Rate (Mult. Factor)
    
    $\Delta Pm$ range is $[0.5, 1.5] = [\text{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

- Genotypic Diversity
- Percentage Completed Trials

Fuzzy Logic Controller

Δ Population Size
Δ Mutation Rate

Genotypic Diversity
Percentage Completed Trials

Fuzzy Controller for ΔN and ΔPₘ:
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 ΔPₘ):
- A(Neg. High), B(Neg. Medium), C(No Change), D(Pos. Medium), E(Pos. High)
### Fuzzy Logic Controller for EAs: Knowledge Base

- **I/O Scaling Factors**
- **I/O Termsets**
- **Rule Sets**

- **Genotypic Diversity**
- **Percentage Completed Trials**

**Fuzzy Logic Controller**

**Object-level EA**

**Object-level Problem**

**Population Size**

**Mutation Rate**

### Fuzzy Controller for Population Size: Rule Set

**GD** = Genotypic Diversity:
- Normalized Average Hamming Distance

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

**ΔN** = Change in Population Size

<table>
<thead>
<tr>
<th>Genotypic Diversity (GD)</th>
<th>Percentage Fitness Evaluation (PFE)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Very Low</strong></td>
<td>Pos High</td>
</tr>
<tr>
<td><strong>Low</strong></td>
<td>Pos High</td>
</tr>
<tr>
<td><strong>Medium</strong></td>
<td>Pos High</td>
</tr>
<tr>
<td><strong>High</strong></td>
<td>Pos Medium</td>
</tr>
<tr>
<td><strong>Very High</strong></td>
<td>No Change</td>
</tr>
</tbody>
</table>

**Exploitation Stage**
- Reduce population/ Refine best

**Exploration Stage**
- Increase population/ broaden search
**Fuzzy Controller for Mutation Rate: Rule Set**

\[
\text{GD}, \text{PFE} \rightarrow \Delta P_m
\]

\(\text{GD} \) = Genotypic Diversity:
Normalized Average Hamming Distance

\(\text{PFE} \) = Percentage Fitness Evaluations:
\((\text{Completed} \# \text{Trials}) / (\text{Max Allocated} \# \text{Trials})\)

\(\Delta P_m \) = Change in Mutation Rate

<table>
<thead>
<tr>
<th>Genotypic Diversity (GD)</th>
<th>Very Low</th>
<th>Low</th>
<th>Medium</th>
<th>High</th>
<th>Very High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very Low</td>
<td>Pos High</td>
<td>Pos High</td>
<td>Pos Medium</td>
<td>Pos Medium</td>
<td>No Change</td>
</tr>
<tr>
<td>Low</td>
<td>Pos High</td>
<td>Pos Medium</td>
<td>Pos Medium</td>
<td>No Change</td>
<td>No Change</td>
</tr>
<tr>
<td>Medium</td>
<td>Pos Medium</td>
<td>Pos Medium</td>
<td>No Change</td>
<td>No Change</td>
<td>No Change</td>
</tr>
<tr>
<td>High</td>
<td>Pos Medium</td>
<td>No Change</td>
<td>No Change</td>
<td>No Change</td>
<td>No Change</td>
</tr>
<tr>
<td>Very High</td>
<td>No Change</td>
<td>No Change</td>
<td>No Change</td>
<td>No Change</td>
<td>No Change</td>
</tr>
</tbody>
</table>

**Fuzzy Controller for \(\Delta N\) and \(\Delta P_m\): 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 (FLC-SEA)
  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

#### Statistical Experiments: Results

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

<table>
<thead>
<tr>
<th>J = Cost Function</th>
<th>U-SEA</th>
<th>FLC-SEA</th>
<th>U-SSGA</th>
<th>FLC-SSGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>( J = C \cdot T )</td>
<td>(-)</td>
<td>(-)</td>
<td>(-)</td>
<td>(-)</td>
</tr>
<tr>
<td>( J = C \cdot T^2 )</td>
<td>(-)</td>
<td>(-)</td>
<td>(-)</td>
<td>(-)</td>
</tr>
<tr>
<td>( J = C \cdot e^{(T-10)/3} )</td>
<td>(-)</td>
<td>(-)</td>
<td>(-)</td>
<td>(-)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Max Numb Trials</th>
<th>U-SSGA</th>
<th>FLC-SSGA</th>
<th>U-SSGA</th>
<th>FLC-SSGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>3000</td>
<td>0% 1788.8 71 0.040</td>
<td>20% 1729 81 0.047</td>
<td>70% 1685 63 0.037</td>
<td>80% 1677 58 0.034</td>
</tr>
<tr>
<td>5000</td>
<td>5% 1767.8 103 0.058</td>
<td>35% 1705 74 0.043</td>
<td>75% 1682 65 0.037</td>
<td>80% 1673 47 0.038</td>
</tr>
<tr>
<td>7000</td>
<td>35% 1710.3 81 0.047</td>
<td>45% 1680 41 0.025</td>
<td>60% 1739 108 0.062</td>
<td>95% 1665 45 0.027</td>
</tr>
<tr>
<td>9000</td>
<td>20% 1748.8 102 0.058</td>
<td>50% 1676 46 0.027</td>
<td>80% 1695 82 0.048</td>
<td>85% 1689 70 0.041</td>
</tr>
<tr>
<td>11000</td>
<td>50% 1719.5 88 0.051</td>
<td>75% 1688 40 0.024</td>
<td>75% 1703 98 0.055</td>
<td>95% 1665 46 0.027</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Max Numb Trials</th>
<th>U-SSGA</th>
<th>FLC-SSGA</th>
<th>U-SSGA</th>
<th>FLC-SSGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>3000</td>
<td>5% 352.5 20.9 0.059</td>
<td>15% 341.3 22.0 0.056</td>
<td>50% 343.8 28.0 0.067</td>
<td>75% 332.4 4.8 0.014</td>
</tr>
<tr>
<td>5000</td>
<td>30% 338.6 8.0 0.024</td>
<td>30% 337.7 7.9 0.023</td>
<td>60% 336.2 11.6 0.024</td>
<td>85% 318.1 4.1 0.012</td>
</tr>
<tr>
<td>7000</td>
<td>20% 339.7 1.9 0.005</td>
<td>30% 338.8 1.7 0.005</td>
<td>70% 341.3 5.2 0.015</td>
<td>75% 336.3 3.4 0.010</td>
</tr>
<tr>
<td>9000</td>
<td>35% 334.2 15.74 0.046</td>
<td>50% 339.8 9.8 0.015</td>
<td>60% 335.2 15.3 0.044</td>
<td>60% 334.8 3.5 0.017</td>
</tr>
<tr>
<td>11000</td>
<td>65% 337.0 15.3 0.045</td>
<td>65% 331.7 3.5 0.010</td>
<td>65% 335.9 15.3 0.045</td>
<td>65% 336.3 15.3 0.045</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Max Numb Trials</th>
<th>U-SSGA</th>
<th>FLC-SSGA</th>
<th>U-SSGA</th>
<th>FLC-SSGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>3000</td>
<td>0% 655.05 90.2 0.136</td>
<td>5% 638.2 87.7 0.130</td>
<td>15% 582.0 53.0 0.084</td>
<td>40% 554.3 73.8 0.023</td>
</tr>
<tr>
<td>5000</td>
<td>10% 625.1 95.0 0.152</td>
<td>25% 600.8 46.7 0.079</td>
<td>35% 597.1 91.5 0.193</td>
<td>55% 570.8 24.7 0.043</td>
</tr>
<tr>
<td>7000</td>
<td>20% 606.84 97.9 0.161</td>
<td>20% 566.4 22.3 0.039</td>
<td>70% 563.6 22.8 0.040</td>
<td>65% 566.0 23.7 0.042</td>
</tr>
<tr>
<td>9000</td>
<td>30% 586.16 29.8 0.058</td>
<td>50% 573.9 41.8 0.073</td>
<td>85% 568.2 17.7 0.022</td>
<td>50% 573.2 24.9 0.043</td>
</tr>
<tr>
<td>11000</td>
<td>25% 608.35 129.4 0.213</td>
<td>45% 573.0 38.2 0.062</td>
<td>60% 563.8 24.4 0.043</td>
<td>70% 563.6 22.7 0.043</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Max Numb Trials</th>
<th>U-SSGA</th>
<th>FLC-SSGA</th>
<th>U-SSGA</th>
<th>FLC-SSGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>3000</td>
<td>0% 47% 60% 7%</td>
<td>7% 20% 47%</td>
<td>47% 60% 7%</td>
<td>7% 20% 47%</td>
</tr>
</tbody>
</table>

Significant changes in \( \mu \) and in \( \sigma \)
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

- **FLC State Representation:** [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)

- **FLC Control Variables:** [EA Adaptable Param.]
  - $\Delta N = \text{Change in Population Size}$
  - $\Delta P_m = \text{Change in Mutation Rate}$
Remarks (cont.)

• 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 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
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.