Utility of Genetic Algorithm and Grammatical Inference for Big Data Analytics: Challenges, Solutions and Applications in Big Data Analytics

Hari Mohan Pandey
Sr. Lecturer, Department of Computer Science
Edge Hill University, United Kingdom
Pandeyh@edgehill.ac.uk
“The art of war is science where everything must be charted”
Outline

1. Introduction and Motivation

2. Background and Related work.
   - Grammatical Inference Methods
   - Genetic Algorithm and Premature Convergence

3. Objectives of this talk

4. Discussion on algorithms

5. Application of GA and GI in Big data analytics

6. Concluding remark
Introduction and Motivation

• **Definition: Grammar Induction (GI)**
  - It deals with idealized learning procedures for **acquiring grammars** on the basis of the evidence about the languages.

• **GI and optimization has a great relevance in various domains:**
  - Compilation and translation
  - Human machine interaction
  - Graphic languages
  - Data mining
  - Computational biology
  - Software engineering
  - Machine learning
  - Big data Analytics

• **Key challenges:**
  - Handling huge search space for grammatical inference.
  - Alleviating premature convergence.
  - Parameter quantification.
Background and Related Work

• Two-fold studies are presented:
  a) Grammatical Inference (GI) Methods
  b) Approaches to avoid premature convergence in GA.
Grammatical Inference Models

Learning Model/Algorithms

- Gold's Learning Model (Identification in the limit)
- Probably Approximately Correct (PAC) Learning Model
- Automatic Distillation of Structure (ADIOS) Algorithm
- e-GRIDS
- Context Distribution Clustering Algorithm (CDC)
- Architecture for Learning Linguistic Structure (ALLiS)
- Genetic Algorithm Based Approach
- Teacher and Query or Oracle Learning Model
- Neural Network based Learning Model
- EMILE
- Computational Learning of Natural Languages (CLL)
- Language Agent (LAgent)
- Memtic Algorithm Based Approach "Magic"
## GI Methods: Working and Issues

<table>
<thead>
<tr>
<th>Method</th>
<th>Author(s)</th>
<th>Working</th>
<th>Issue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification of the limit</td>
<td>Gold 1967</td>
<td>Identify grammars from unknown languages</td>
<td>Next sample may invalidate the previous hypothesis</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Not fit for negative strings.</td>
</tr>
<tr>
<td>Teachers and Query Model</td>
<td>Angluin 1973</td>
<td>Target languages are known to the teacher</td>
<td>Not fit for negative strings</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Capable to answer a particular type of questions</td>
<td>Answer in terms of “yes” or “no”</td>
</tr>
<tr>
<td>Probably Approximate Correct</td>
<td>Valiant 1984</td>
<td>Combines the merits of the Gold’s and teachers and query model.</td>
<td>Inference algorithms must learn in polynomial time, which is practically not possible</td>
</tr>
</tbody>
</table>
## GI Methods: Working and Issues

<table>
<thead>
<tr>
<th>Method</th>
<th>Reference</th>
<th>Description</th>
<th>Issues</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neural Network based learning algorithms</td>
<td>Alex et al. 2009</td>
<td>Uses recurrent neural network, works as a internal short term memory.</td>
<td>Difficult to manage due to complex structure.</td>
</tr>
<tr>
<td>Automatic Distillation of Structure (ADIOS)</td>
<td>Solan, 1983</td>
<td>Statistical inference approach. Produces Context free grammar</td>
<td>Compromised in terms of accuracy, showed 60% accuracy.</td>
</tr>
<tr>
<td>EMILE</td>
<td>Adrian's 1993</td>
<td>Similar to teachers and query model. Follow a clustering approach to identify the rules</td>
<td>Slow learning approach. Compromised with accuracy.</td>
</tr>
<tr>
<td>E-GRIDS</td>
<td>Petasis 1998</td>
<td>Follow grammar inference by simplicity algorithm. Do not use Oracle</td>
<td>Not fit for negative samples.</td>
</tr>
</tbody>
</table>
# GI Methods: Working and Issues

<table>
<thead>
<tr>
<th>CLL</th>
<th>Watkinson, 1999</th>
<th>Text based supervised learning approach</th>
<th>Not fit for negative strings.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Parsing is difficult</td>
<td></td>
</tr>
<tr>
<td>Mimetic Algorithm based</td>
<td>Mernik, 2013</td>
<td>Developed for domain specific languages</td>
<td>Not suitable for global solution.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Good in finding local optima</td>
<td></td>
</tr>
<tr>
<td>Genetic Algorithm</td>
<td>Many</td>
<td>Most recent work is conducted by Dr. N.S. Choubey, 2013 for context free grammar induction using genetic algorithm employing elite mating pool.</td>
<td>Compromised in terms of generalization of the input.</td>
</tr>
</tbody>
</table>
Classification of GI Methods based on the learning types: text, informant, supervised, unsupervised and semi-supervised

<table>
<thead>
<tr>
<th>Learning model</th>
<th>Text based</th>
<th>Informant based</th>
<th>Supervised</th>
<th>Unsupervised</th>
<th>Semi-supervised</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification in the limit</td>
<td>Yes</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Teacher and Query/ Oracle</td>
<td></td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PAC learning model</td>
<td>Yes</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Neural Network Based learning model</td>
<td>Yes</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>ADIOS learning model</td>
<td>Yes</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>EMILE learning model</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>e-GRIDS learning model</td>
<td>Yes</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>CLL learning model</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CDC learning model</td>
<td>Yes</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>LAagent learning model</td>
<td>Yes</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>GA based learning</td>
<td>Yes</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>ALLiS learning model</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>ABL learning</td>
<td>Yes</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>TBL learning model</td>
<td>Yes</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>ITBL Learning model</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
</tbody>
</table>
### Classification based on the type training data/corpus/examples used

<table>
<thead>
<tr>
<th>Learning model</th>
<th>Positive examples only</th>
<th>Both positive and negative examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification in the limit</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Teacher and Query/ Oracle</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>PAC learning model</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Neural Network Based learning model</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>ADIOS learning model</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>EMILE learning model</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>e-GRIDS learning model</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>CLL learning model</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>CDC learning model</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>LAagent learning model</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>GA based learning</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>ALLiS learning model</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>ABL learning</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>TBL learning model</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>ITBL Learning model</td>
<td></td>
<td>Yes</td>
</tr>
</tbody>
</table>
## Classification Based on the Technique(s) Used

<table>
<thead>
<tr>
<th>Learning model</th>
<th>SM</th>
<th>EA</th>
<th>MDL</th>
<th>HM</th>
<th>GSM</th>
<th>CA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification in the limit</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Teacher and Query/ Oracle learning model</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PAC learning model</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Neural Network Based learning model</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADIOS learning model</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMILE learning model</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>e-GRIDS learning model</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>CLL learning model</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CDC learning model</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>LAagent learning model</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>GA based learning</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>ALLiS learning model</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>ABL learning</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>TBL learning model</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>ITBL Learning model</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>

SM: Statistical method,  EA: Evolutionary algorithm, MDL: Minimum description length
HM: Heuristic method, GSM: Greedy search method, CA: Clustering approach
Genetic Algorithm

• Genetic Algorithm is:
  – A search and optimization algorithm
  – Based on “Survival of the fittest”
  – Solves complex problems.
  – Suitable for the problem, where search space grows exponentially.
Working of the Classical GA
Factors Needs Special Attention

• GA follows Darwin’s theory says that evolution occurs if and only if three condition meets.
Factors Significantly affects the GA’s working
The Main Challenge in GA: Premature Convergence

- The major challenge with GA is handling premature convergence and slow finishing.

- Premature convergence is a situation when the diversity of the population decreases as it converges to a local optimum.
Addressing Mechanism (1)

• To overcome from this,
  
  – it is necessary to select the best solution of the current generation.

  – The tendency to select the best member of the current generation is known as selective pressure.

  – An approach to increase the population size may not be sufficient because any increase in the population size increases the computation cost.
Addressing Mechanism (2)

– Applying mutation alone converts GA into random search, whereas crossover alone generates sub-optimal solution.

– Also, selection, crossover and mutation if applied together may result GA to noise tolerant hill-climbing approach.

– An example of this view is the theoretical study conducted by Spears\(^1\) - argues that crossover and mutation having different properties that complement each other.

\(^1\)http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.75.6708&rep=rep1&type=pdf
Existing Approaches to Alleviate Premature Convergence

- Crowding Method
- Incest Prevention Algorithm
- Scheduled Sharing Approach
- Syntactic Analysis of Convergence
- Adaptive Probability Based Approach
- The Island Model Genetic Algorithm
- Shifting Balance Theory in Dynamic Environment
- Self-Adaptive Selection Pressure Steering Approach (SASEGASA)
- GA using Self-Organizing Maps
- Age-Layered Population Structure (ALPS) Approach
- Hybrid Particle Swarm Optimization (PSO) and GA
- Dynamic Application of Reproduction Operator (DARO)
- Frequency Crossover with Nine Different Mutations

The Big Question?

• What the researchers did in their approach?

• How they addressed the premature convergence?
Key Elements in Addressing Premature Convergence

• Maintain good balance between exploration and exploitation.

• Maintaining selection pressure: Select Best Individuals in each generation.

• Fitness function: Identify good fitness function.
Approaches to balance Exploration and Exploitation (1)

- The following approaches has been tried:
  - **Trial-and-error approach** - time-consuming and tedious method.
  - **Following general guidelines** (e.g., [De Jong 1975; Harik and Lobo 1999; Schaffer et al. 1989]), which are often inapplicable for specific cases. It is often found that recommended parameter settings from literature do not lead to the best solutions for particular cases (e.g., [Smith and Eiben 2009; Zielinski et al. 2009]);
  - **Using parameter-less EA** [Harik and Lobo 1999] or GAs without parameters [Back et al. 2000], which are robust but mostly less efficient approaches;
Approaches balance Exploration and Exploitation (2)

- **Using experiences from previous similar applications**, which is inapplicable when such experiences do not exist;

- **Statistical analysis of control parameter interactions** and their effect on algorithms’ performances [Czarn et al. 2004];

- **Using mathematical models**, which is important but often too simple to be realistic or too difficult to be understood by ordinary users;

- **Algorithmic parameter tuning**, where the search for the best control parameters is seen as an optimization problem that can be solved using specific methods.
Objectives

• Discuss a Genetic Algorithm for CFG induction and comparison with existing algorithms.

• Mechanism to address premature convergence.

• Applications of GA and GI for Big Data Analytics.
Approaches Proposed

• Bit Masking Oriented Genetic Algorithm (BMOGA)

• Bit Masking Oriented Genetic Algorithm with Minimum Description Length Principle (GAWMDL)

[Links]
Paper-2.pdf
Paper-3.pdf
Grammatical Inference from Chromosomes

Start

Generate Variable Length Chromosome

Evaluate Fitness

If Best Individual > Threshold OR Total Run = Max. Generation

Apply mask-fill crossover and mutation

Select Parent Pairs P1, P2

Set CM = Initialize Crossmask
Set MM = Initialize mutmask
Perform T1 = P1 AND CM
Perform T2 = P2 AND (NOT CM)
Perform T3 = P2 AND CM
Perform T4 = P1 AND (NOT CM)
Perform OS1 = T1 OR T2
Perform OS2 = T3 OR T4

Update OS1 = OS1 XOR MM
Update OS2 = OS2 XOR MM

Boolean based procedure (CM, MM, P1, P2)

Replacement to incorporate new population
Set New Population = Population after crossover and mutation

Merge the population and update the best individual

Substep-1

Selection Process

Substep-2

Evaluate Fitness

Substep-3

Bit Mask Oriented Genetic Algorithm

Substep-4

Substep-5

Substep-6

Yes

Step-4

Display CFG rules with highest fitness value

Step-5

Display total time elapsed in the implementation

Stop
The Basic Configuration of the BOMGA

- It uses “Bit-masking Oriented Data Structure”.
- Applies mask-fill reproduction operators:
  - Crossover
    - Cut crossover mask-fill operator
    - Bit-by-bit mask-fill crossover
    - Local cut crossover mask-fill operator
  - Mutation
    - Mutation Mask fill
- Uses Boolean based procedure mixture for offspring generations.
BMOGA based Model for Grammatical Inference

Training Data
Positive Samples  Negative Samples

Learning System

Context Free Grammars
Capable to describe the pattern of the sample strings

G=0
Random Initial Population Generator

Phase-1
Perform Mask Fill
Crossover
Perform Mask Fill
Mutation

Phase-2
Perform boolean_function (crossmask, mutmask, P1, P2)

Termination?
Replacement Strategy

No

Yes

Exit

Data Store Managing Historical Data

Parser to Validate CFG

Input
Finite State Controller
Accept/Reject

Stack

Working Video of BMOGA
Grammar Inference and GA

- **Mapping**: Mapping binary chromosome into terminals and non-terminals.

- **Representation in Backus Naur Form (BNF)**: Apply biasing on the symbolic chromosome to get non-terminal on the left-hand side.

- **Production rules generation**: Divide the symbolic chromosome to produce the production rules up to the desired length.

- **Resolve the insufficiency**: Addressing the issues such as unit production, left recursion, left factoring, multiple production, useless production, ambiguity etc.

Grammar Induction.docx
Fitness Calculation

\[ Fitness = \sum K \cdot ((APS + RNS) - (ANS + RPS)) + (2K - NPR) \]

S.T.

\( APS + RNS \leq \text{Number of positive samples in corpus data} \)
\( ANS + RPS \leq \text{Number of negative samples in corpus data} \)
\( NPR: \text{number of grammar rules} \)
\( K: \text{Constant} \)

- **Fitness Calculation.docx**
Reproduction Operators

• Crossover
  – Cut crossover mask-fill operator
  – Bit-by-bit mask-fill crossover
  – Local cut crossover mask-fill operator

• Mutation
  – Mutation Mask fill
Algorithm-1: cutcrossover (P1, P2)

S1 Set C = RND [1 to L-1]
S2 Initially Set Sum (i) = 0
S3 For i = 1 to Ndv do
S4 Set Sum (i) = Sum (i) + R (i)
S5 If Sum < C then
S6 Set Mask (i) = -1
S7 Else
S8 Set j = i
S9 If j ≠ Ndv then
S10 For k = j + 1 to Ndv
S11 Set Mask (k) = 0
S12 Else
S13 Set i = Ndv
S14 End for
S15 Set Mask (j) = 2^{R(j)} - 2^{Sum(i)-C}
**Algorithm-2: bitbybit (P1, P2)**

1. For i = 1 to Ndv do
2. \[\text{Set Mask} (i) = \text{INT} (\text{RND} \times R (i)) - 1\]
3. End for

**Algorithm-3: localcut (P1, P2)**

1. For I = 1 to Ndv do
2. \[\text{Set C} = \text{RND} \left( 1 \div R (I) - 1 \right)\]
3. \[\text{Set Mask} (I) = 2^R (I) - 2^C\]
4. End for
## Working of Algorithm-1

<table>
<thead>
<tr>
<th>Crossmask integer value</th>
<th>P1</th>
<th>P1 to P2 bit transition</th>
<th>P2</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>61440</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resolution</th>
<th>Var-1</th>
<th>Var-2</th>
<th>Var-3</th>
<th>Var-4</th>
<th>Var-5</th>
<th>Var-6</th>
</tr>
</thead>
<tbody>
<tr>
<td>8-bit</td>
<td>8-bit</td>
<td>4-bit</td>
<td>16-bit</td>
<td>16-bit</td>
<td>8-bit</td>
<td></td>
</tr>
</tbody>
</table>

| Binary Image | 11111111 | 11111111 | 1111 | 1110000100000000 | 0000000000000000 | 00000000 |

- **String Length (L) = 60**
- **Cut (C) = 24**
- **Remaining = 12**
- **Sum = Cut + Remaining = 24 + 12 = 36**
Working of Algorithm-1

- A binary string \(L = 60\) is taken
- Design variables \(\text{Ndv} = 6\)
- \(R[i] = \{8, 8, 4, 16, 16, 8\}, i = 1...6.\)
- A random cut \(C = 24\) is set at 24 (S1).
- Initially, \(\text{Sum (i)} = 0\) is set since none of the \(\text{Ndv}\) is selected (S2).
Working of Algorithm-1

- A loop is applied (S3) that updates Sum (i) in each iteration and if Sum (i) is less than C, set -1 in the CM, which will be true in first three iterations changes Sum (i)
  - 1st iteration: Sum (1) = 0 + 8 = 8 < C,
  - 2nd iteration: Sum (2) = 8 + 8 = 16 < C
  - 3rd iteration: Sum (3) = 16 + 4 = 20 < C),
  - but in 4th iteration Sum (4) = 20 + 16 = 36 > C.
Working of Algorithm-1

• Then set \( j = i \), i.e. \( j = i = 4 \) (S8).
• Check the condition specified at S9 and execute another loop given at S10 as:
  • For \( k = 4 + 1 \) to 6
  • For \( k = 5 \) to 6,
  • it will set the mask for P2 (Mask (5) = Mask (6) = 0).
Working of Algorithm-1

- The third section represents the P1-to-P2 transition and to fill this part equation (2) has been used (S15).

\[
CM_{tr}^C = 2^{R_{tr}} - 2^{\text{Sum}(tr)-C} \text{ with } tr \in [1 \div Ndv]
\]

\[
R_{tr} = R(4) = 16 \quad C = 24
\]

\[
\text{Sum}(4) = 20 + 16 = 36
\]

\[
CM_{4}^{24} = 2^{16} - 2^{36-24}
\]

\[
\Rightarrow CM_{4}^{24} = 65536 - 4096 = 61440
\]
Mutation Mask-fill Operator

• The mutation mask-fill operation is very similar to the classical random binary inversion driven,

• it performs the mutation based on the specific mutation rate.

\[ MM_i = \sum_{j=0}^{R_j-1} 2^j \cdot \delta_{lm} \quad i = 1, NdV \]

Where, \( \delta_{lm} = \begin{cases} l = m & \text{if } \text{rand} < \text{MutRate} \\ l \neq m & \text{otherwise} \end{cases} \)
Offspring Generation

Selection of Parent P1 and P2

OS1 = f1(P1, P2, CM, MM)

T1 = P1 AND CM

T2 = P2 AND (NOT CM)

OS1 = T1 OR T2

T4 = P1 AND (NOT CM)

OS2 = T3 OR T4

OSj = OSi XOR MM

Perform Mutation

Offspring Generation.docx
Minimum Description Length Principle

- Minimum Description Length Principle:
  - any regularity in a given set of data can be used to compress the data
  - Used to describe data using fewer symbols than needed to describe the data literally.
Minimum Description Length Principle and Grammar Inference

• A partial grammar $G$ is defined, which contains set of CFG rules for the training data.

• Two basic operations are performed:
  – Merge or merge for shorting the production rule.
  – Construction operation.
Merge Operation

Let

\[ g'_1 = \{g'_1 \rightarrow g'_2 g'_4/g'_3\} \in G \]

\[ g'_8 = \{g'_5 \rightarrow g'_7\} \in G \]

Then, these rules can be merged as:

\[ g_{new} = \{g'_1 \cup g'_8\} = \{g'_{new} \rightarrow g'_2 g'_4/g'_3/g'_7\} \]

And then remove: \( g'_1 \) and \( g'_8 \) from \( G \). Apply re-indexing to accommodate.
Construction Operation

• Let $g_l$ and $g_k$ are two classes, then applying construction operation we get:

$$g_{new} = \{g'_{new} \rightarrow g'_l g'_k \}$$
Working of MDL for Grammar Induction

Algorithm: MDL principle for GI

S1. Set a separate class for each string is present in the training set as:
    \[ g_1 = \{ g_1' \rightarrow w_1 \}, g_2 = \{ g_2' \rightarrow w_2 \} \ldots \]

S2. Perform merge operation
    \[ G = g_1 \cup g_2 \cup \ldots \]

S3. Perform the construction operation.

S4. Compute description length (DL)
    \[ DL = DL(trainingset) + DL(G) \]

S5. Determine the difference in DL that would result from merging of two classes.

S6. Determine the difference that would result from construction operation.

S8. If New_DL < Old_DL Then
    S9. Select the New_DL

S10. Else
    S11. Select the Old_DL
Standard Datasets

<table>
<thead>
<tr>
<th>L-id</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>(10)* over (0 + 1)*</td>
</tr>
<tr>
<td>L2</td>
<td>String not containing ‘000’ over (0+1)*</td>
</tr>
<tr>
<td>L3</td>
<td>Balanced Parentheses</td>
</tr>
<tr>
<td>L4</td>
<td>Odd binary number over (0 + 1)*</td>
</tr>
<tr>
<td>L5</td>
<td>(0<em>1) over (0+1)</em></td>
</tr>
<tr>
<td>L6</td>
<td>0(00)<em>1 over (0+1)</em></td>
</tr>
<tr>
<td>L7</td>
<td>Any string with even 0 and odd 1 over (0+1)*.</td>
</tr>
<tr>
<td>L8</td>
<td>Even binary number over (0+1)*</td>
</tr>
<tr>
<td>L9</td>
<td>{0^n1^n} over (0+1)*</td>
</tr>
<tr>
<td>L10</td>
<td>0* over (0+1)*</td>
</tr>
<tr>
<td>L11</td>
<td>All strings even number of 0 over (0+1)*.</td>
</tr>
<tr>
<td>L12</td>
<td>{0^n1^{2n}} over (0+1)*</td>
</tr>
</tbody>
</table>
## Standard Datasets

<table>
<thead>
<tr>
<th>L13</th>
<th>$(00)^<em>10^</em> \text{ over } (0+1)^*$.</th>
</tr>
</thead>
<tbody>
<tr>
<td>L14</td>
<td>$(00)^<em>(111)^</em> \text{ over } (0+1)^*$.</td>
</tr>
<tr>
<td>L15</td>
<td>Palindrome over ${a + b}$.</td>
</tr>
<tr>
<td>L16</td>
<td>$0^*1^<em>0^<em>1^</em> \text{ over } (0+1)^</em>$.</td>
</tr>
<tr>
<td>L17</td>
<td>No odd zero after odd ones over $(0+1)^*$.</td>
</tr>
<tr>
<td>L18</td>
<td>Palindrome over ${a + b + c}$.</td>
</tr>
<tr>
<td>L19</td>
<td>$12^*1+02^<em>0 \text{ over } (0+1+2)^</em>$.</td>
</tr>
<tr>
<td>L20</td>
<td>$(0^<em>+2^</em>)1 \text{ over } (0+1+2)^*$.</td>
</tr>
<tr>
<td>L21</td>
<td>$c^{n+1}d^{2n+1} \text{ over } (c+d)^*$.</td>
</tr>
<tr>
<td>L22</td>
<td>$ab^* \cup cb^* \text{ over } (a + b + c)^*$.</td>
</tr>
</tbody>
</table>
Parameter Selection and Tuning

- Orthogonal array-based approach
- Taguchi method

\[ SNR_i = -10 \log \left( \sum_{u=1}^{N} \frac{y_u^2}{N_i} \right) \]
Factors and Levels Selected

Table 1: Selected Factor and Levels

<table>
<thead>
<tr>
<th>Factors</th>
<th>Level 1</th>
<th>Level 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population Size (PS)</td>
<td>120</td>
<td>360</td>
</tr>
<tr>
<td>Chromosome Size (CS)</td>
<td>120</td>
<td>240</td>
</tr>
<tr>
<td>Production Rule Length (PRL)</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>Crossover Rate (CR)</td>
<td>0.6</td>
<td>0.9</td>
</tr>
<tr>
<td>Mutation Rate (MR)</td>
<td>0.5</td>
<td>0.8</td>
</tr>
</tbody>
</table>

Table 2: Selected Factor and Levels

<table>
<thead>
<tr>
<th>Level</th>
<th>PS</th>
<th>PRL</th>
<th>CS</th>
<th>CR</th>
<th>MR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-11.15</td>
<td>-11.31</td>
<td>-11.13</td>
<td>-10.65</td>
<td>-10.83</td>
</tr>
<tr>
<td>2</td>
<td>-10.82</td>
<td>-10.66</td>
<td>-10.84</td>
<td>-11.31</td>
<td>-11.13</td>
</tr>
<tr>
<td>Delta</td>
<td>0.33</td>
<td>0.65</td>
<td>0.29</td>
<td>0.66</td>
<td>0.30</td>
</tr>
<tr>
<td>Rank</td>
<td>3</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>
## Parameters Value Selection

<table>
<thead>
<tr>
<th>Ex. No.</th>
<th>PS</th>
<th>PRL</th>
<th>CS</th>
<th>CR</th>
<th>MR</th>
<th>Means</th>
<th>Coff. Variation</th>
<th>Std.dev.</th>
<th>SNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>120</td>
<td>5</td>
<td>120</td>
<td>0.6</td>
<td>0.5</td>
<td>3.56667</td>
<td>0.0428278</td>
<td>0.152753</td>
<td>-11.0506</td>
</tr>
<tr>
<td>2</td>
<td>120</td>
<td>5</td>
<td>120</td>
<td>0.9</td>
<td>0.8</td>
<td>4.06667</td>
<td>0.0375621</td>
<td>0.152753</td>
<td>-12.1889</td>
</tr>
<tr>
<td>3</td>
<td>120</td>
<td>8</td>
<td>240</td>
<td>0.6</td>
<td>0.5</td>
<td>3.26667</td>
<td>0.0467610</td>
<td>0.152753</td>
<td>-10.2884</td>
</tr>
<tr>
<td>4</td>
<td>120</td>
<td>8</td>
<td>240</td>
<td>0.9</td>
<td>0.8</td>
<td>3.56667</td>
<td>0.0901276</td>
<td>0.321455</td>
<td>-11.0687</td>
</tr>
<tr>
<td>5</td>
<td>360</td>
<td>5</td>
<td>240</td>
<td>0.6</td>
<td>0.8</td>
<td>3.50000</td>
<td>0.0285714</td>
<td>0.100000</td>
<td>-10.8837</td>
</tr>
<tr>
<td>6</td>
<td>360</td>
<td>5</td>
<td>240</td>
<td>0.9</td>
<td>0.5</td>
<td>3.59000</td>
<td>0.0460243</td>
<td>0.165227</td>
<td>-11.1080</td>
</tr>
<tr>
<td>7</td>
<td>360</td>
<td>8</td>
<td>120</td>
<td>0.6</td>
<td>0.8</td>
<td>3.30000</td>
<td>0.0606061</td>
<td>0.200000</td>
<td>-10.3809</td>
</tr>
<tr>
<td>8</td>
<td>360</td>
<td>8</td>
<td>120</td>
<td>0.9</td>
<td>0.5</td>
<td>3.50000</td>
<td>0.0494872</td>
<td>0.173205</td>
<td>-10.8884</td>
</tr>
</tbody>
</table>

➢ Parameters Tunining and Selection.docx
Effects of Parameters

Main Effects Plot for SN ratios

Data Means

Signal-to-noise: Smaller is better
Comparison with Existing Approaches

• Three-fold comparison has been done:

  ➢ Compared with approaches to **address premature convergence**.
    • Random offspring generation approach (ROGA)
    • Elite mating pool approach (EMPGA)
    • Dynamic application of reproduction operators (DARO)

  ➢ Tested against the other **global optimization algorithm**:
    • Compared with **Classical GA**
    • Particle swarm optimization (PSO)
    • Simulated Annealing (SA)

  ➢ Compared against the algorithm was **proposed for Context Free Grammar Induction**:
    • Compared with Improved Tabular Representation Algorithm (ITBL)
BMOGA is tested against the algorithms:

- Introduces diversity and prevent premature convergence.
- Study is conducted on 4-algorithms
  - Simple Genetic Algorithm (SGA)
  - Random off-spring generation genetic algorithm (ROGGA).
  - Elite mating pool genetic algorithm (EMPGA)
  - Dynamic allocation of crossover and mutation operators (DARO).
Experiment-1: Results and Analysis

- Quality measures
  - Average production rules
    \[ \text{APR} = \frac{\sum_{i=1}^{n} \frac{NPR_i}{10}}{\sum_{j=1}^{10} NR_j} \]
  - Success ratio
    \[ \text{Succ. Ratio} = \frac{\sum_{i=1}^{n} \frac{NOPR_i}{10}}{\sum_{j=1}^{10} NR_j} \times 100 \]

Results of Experiment-1.docx
Average Production Rule Comparison Chart

Languages L1 through L22

**Average Production Rule**

- SGA
- ROGGA
- EMPGA
- BMOGA
- DARO

Graph shows the comparison of average production rules across various languages from L1 to L22, with different algorithms represented by different colors.
Execution Time Comparison Chart
Experiment-1: Statistical Tests

• Statistical tests are conducted considering the hypothesis:

\[ H_0: \mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu_5 \]

\[ H_A: \text{At least one mean is different than others.} \]

• F-test is conducted, which is based on ANOVA.

• Posthoc tests are also conducted.
  – LSD Test
  – TukeyHSD Test
Estimated Marginal Mean Chart

Estimated Marginal Means of Fitness

- SGA
- EMPGA
- ROGGA
- DARO
- BMOGA
Experiment-2

• Performance of BMOGA is tested against the other global optimization algorithms.
  – Particle Swarm Optimization (PSO)
  – Simulated Annealing (SA)
  – Simple Genetic Algorithm (SGA).
Experiment-2: Results, analysis and Statistical Tests

• Results are collected considering the following factors:
  – Threshold value.
  – Execution time
  – Generation range.

• Statistical tests are conducted.
  – F-test based on ANOVA
  – Multiple comparison test.

Hypothesis:

\[ H_0: \mu_{SGA} = \mu_{PSO} = \mu_{SA} = \mu_{BMOGA} \]
\[ H_A: \text{At least one mean is different than others.} \]

Results of Experiment-2.docx
Experiment-2: Estimated Marginal Mean Chart

- BMOGA
- PSO
- SA
- SGA

Amity University Uttar Pradesh, India
Experiment-3

• The BMOGA with MDL is tested against the algorithm that was proposed for the CFG induction.

• Algorithms taken into consideration are:
  – Genetic Algorithm without MDL principle (GAWOMDL)
  – Genetic Algorithm with MDL principle (GAWMDL)
  – Improved Tabular Representation Algorithm (ITBL)
Results, Analysis and Statistical Test

• **Results of Experiment-3.docx**

• Statistical test is conducted creating the hypothesis:

\[ H_0: \mu_1 = \mu_2 = \mu_3 = \mu_4 \]

\[ H_A = \mu_1 \neq \mu_2 \neq \mu_3 \neq \mu_4 \]

Paired t-test is conducted creating three pairs of algorithms:

- **a) GAWOMDL-GAWMDL (Pair-1)**
- **b) EMPGA-GAWMDL (Pair-2)**
- **a) ITBL-GAWMDL (Pair-3)**
Experient-3: Estimated Marginal Mean Chart

![Graph showing the relationship between Mean of Fitness and Algorithms]
Some Notable Applications

- **Visual report generation** ([Unold et al. 2017](#))
  - A **grammar-based classifier system** is used to generate visual reports to analyze data.

- **Job shop Scheduling for Big Data Analytics** ([Lu et al. 2018](#))
  - A **GA based system** was proposed to improve the efficiency (by predicting the performance of clusters when processing jobs) of big data analytics.

- **Analysis of Big data models** ([Han et al. 2017](#))
  - **Dependency Grammar Induction** was used to analyze the impact of big data models in terms of training corpus size.

- **Mining of Big Data** ([A. A. Algwaiz, 2017](#))
  - Grammatical Inference method was used for mining big data.
Conclusions

• This talk have presented
  
  – Discussion of Grammatical Inference and Genetic Algorithm.
  
  – A key issue with the Genetic Algorithm has been discussed.
  
  – Two algorithms have been discussed to avoid premature convergence.
  
  – Applications of GA and GI for Big Data Analytics have been discussed.
Thank you