Implementation of Multi-model Approach for Estimation of Distribution Algorithms (EDA) to Solve Multi-Structural Problems

By Mohammad Reza Karimi Dastjerdi

Supervisor: Dr. Amin Nikanjam

Summer 2016
Contents

- Introduction
- Heuristic strategies
- Stochastic Heuristic Searches
- Genetic Algorithm
- Estimation of Distribution Algorithms
- EDAs Pseudo code
- Bayesian Optimization Algorithm
- BOAs Pseudo code
- Bayesian Networks
- Bayesian Dirichlet Metric
- Find Best Network
- Generate New Solutions

- Selection Operators
- Parent Selection
- Truncation Selection
- Tournament Selection
- Survivor Selection
- Worst Replacement
- Restricted Tournament Replacement
- Multi-model Approach
- Multi-structural Problems
- Conclusion & Suggestions
- References
Introduction

- Optimization is a mathematical discipline that concerns the finding of the extreme of numbers, functions, or systems.

- All the search strategy types can be classified as:
  - Complete strategies: All possible solutions of the search space
  - Heuristic strategies: Part of them following a known algorithm
Heuristic strategies

- **Deterministic**
  - Same Conditions → Same Solution
  - Main drawback → Risk of getting stuck in local optimum values

- **Non-deterministic**
  - Even in Same Conditions → Different Solutions
  - Escape from these local maxima by means of the randomness
Stochastic Heuristic Searches

- Store one solution: Simulated Annealing
Genetic Algorithms

- One of the examples of evolutionary computation is Genetic Algorithms (GAs).
- Inspired by the process of natural selection
- Relied on bio-inspired operators such as mutation, crossover and selection.
- The behavior of GAs depends on too many parameters like:
  - Operators and probabilities of crossing and mutation
  - Size of the population
  - Rate of generational reproduction
  - Number of generations
  - and so on.
Estimation of Distribution Algorithms

- All these reasons have motivated the creation of a new type of algorithms classified under the name of Estimation of Distribution Algorithms (EDAs).

- EDAs are:
  - Population-based
  - Probabilistic modelling of promising solutions
  - Simulation of the induced models to guide their search
The Estimation of Distribution Algorithm (EDA)

(1) set $t \leftarrow 0$
    randomly generate initial population $P(0)$
(2) select a set of promising strings $S(t)$ from $P(t)$
(3) estimate the distribution of the selected set $S(t)$
(4) generate a set of new strings $O(t)$ according to the estimate
(5) create a new population $P(t + 1)$ by replacing some strings from $P(t)$ with $O(t)$
    set $t \leftarrow t + 1$
(6) if the termination criteria are not met, go to (2)
Bayesian Optimization Algorithm

- By Pelikan, Goldberg, & Cantu-paz [1998]
BOAs Pseudo code

The Bayesian Optimization Algorithm (BOA)

1. set $t \leftarrow 0$
   randomly generate initial population $P(0)$
2. select a set of promising strings $S(t)$ from $P(t)$
3. construct the network $B$ using a chosen metric and constraints
4. generate a set of new strings $O(t)$ according to the joint distribution encoded by $B$
5. create a new population $P(t+1)$ by replacing some strings from $P(t)$ with $O(t)$
   set $t \leftarrow t + 1$
6. if the termination criteria are not met, go to (2)
Bayesian Networks

- A Bayesian network encodes the relationships between the variables contained in the modeled data.
- Bayesian networks can be used to:
  - Describe the data
  - Generate new instances of the variables with similar properties as those of given data
- Each node → one variable.

\[ p(X) = \prod_{i=0}^{n-1} p(X_i | \prod X_i) \]

\[ p(X) = p(X_0).p(X_1|X_0).p(X_2|X_0,X_1).p(X_3|X_1) \]
Bayesian Dirichlet Metric

- By Heckerman et al [1994]
- Prior knowledge about the problem + Statistical data from a given data set.
- The BD metric for a network B given a data set D and the background information $\xi$ denoted by $p(D, B | \xi)$ is defined as:

$$ p(D, B|\xi) = p(B|\xi) \prod_{i=0}^{n-1} \frac{m'(\pi_{X_i})!}{(m'(\pi_{X_i}) + m(\pi_{X_i}))!} \prod_{x_i} \frac{(m'(x_i, \pi_{X_i}) + m(x_i, \pi_{X_i}))!}{m'(x_i, \pi_{X_i})!} $$

$$ m(\pi_{X_i}) = \sum_{x_i} m(x_i, \pi_{X_i}) $$
Find Best Network

- **K = 0**
  - An empty network is the best one (and the only one possible).

- **K = 1**
  - special case of the so-called maximal branching problem

- **K > 1**
  - NP-complete
  - A simple greedy search
  - Only operations are allowed that:
    - keep the network acyclic
    - at most k incoming edges into each node
  - The algorithms can start with:
    - An empty network
    - Best network with one incoming edge for each node
    - Randomly generated network.
Generate New Solutions

The algorithm for the generation of a new instance

1. mark all variables as unprocessed
2. pick up an unprocessed variable $X_i$ with all parents processed already
3. set $X_i$ to $x_i$ with probability $p(X_i = x_i | \Pi_{X_i} = \pi_{X_i})$
4. mark $X_i$ as already processed
5. if there are unprocessed variables left, go to (2)

- Time complexity of generating the value for each variable: $O(k)$.
- Time complexity of generating an instance of all variables: $O(nk)$. 
Multi-model Approach

- Many real-world problems present several global optima.
- Most EDAs typically bypass the issue of global multi-modality.
- It is often preferable or even necessary to obtain as many global optima as possible.
- Used in ECGA by Chuang and Hsu [2010].
Implementation
## Results

<table>
<thead>
<tr>
<th>Problem Size</th>
<th>#Models</th>
<th>RTR?</th>
<th>Final Population size</th>
<th>Average algorithm time</th>
<th>Average model-building time</th>
<th>Average fitness calls</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>1</td>
<td>0</td>
<td>1312</td>
<td>0.028</td>
<td>0.019</td>
<td>6822</td>
</tr>
<tr>
<td>30</td>
<td>3</td>
<td>0</td>
<td>1812</td>
<td>0.015</td>
<td>0.021</td>
<td>16416</td>
</tr>
<tr>
<td>30</td>
<td>3</td>
<td>1</td>
<td>734</td>
<td>0.028</td>
<td>0.022</td>
<td>15995</td>
</tr>
</tbody>
</table>

**Different Approaches**

- **Traditional**
- **Multi-Models**
- **Multi-Models with RTR**

**Population Size**

**Average Fitness Calls**
Implementation

\[
N_{\text{model}_i} = \frac{\bar{f}_i}{\sum_{j=1}^{\text{Models}} \bar{f}_j}
\]

\[
N_{\text{model}_i} = \frac{e\bar{f}_i}{\sum_{j=1}^{\text{Models}} e\bar{f}_j}
\]
## Results

<table>
<thead>
<tr>
<th>Problem Size</th>
<th>#Models</th>
<th>RTR?</th>
<th>SoftMax?</th>
<th>Final Population size</th>
<th>Average algorithm time</th>
<th>Average model-building time</th>
<th>Average fitness calls</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1312</td>
<td>0.028</td>
<td>0.019</td>
<td>6822</td>
</tr>
<tr>
<td>30</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>15250</td>
<td>0.126</td>
<td>0.193</td>
<td>143689</td>
</tr>
<tr>
<td>30</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>578</td>
<td>0.039</td>
<td>0.039</td>
<td>26580</td>
</tr>
</tbody>
</table>

**Different Approaches**

- Traditional
- Multi-Models with Softmax
- Multi-Models with Softmax and RTR

### Average Fitness Calls

- Traditional
- Multi-Models with Softmax
- Multi-Models with Softmax and RTR

### Population Size

- Traditional
- Multi-Models with Softmax
- Multi-Models with Softmax and RTR
Implementation
Results

<table>
<thead>
<tr>
<th>Problem Size</th>
<th>#Models</th>
<th>RTR?</th>
<th>SoftMax?</th>
<th>Parent Displacement?</th>
<th>Final Population size</th>
<th>Average algorithm time</th>
<th>Average model-building time</th>
<th>Average fitness calls</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1312</td>
<td>0.028</td>
<td>0.019</td>
<td>6822</td>
</tr>
<tr>
<td>30</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1562</td>
<td>0.038</td>
<td>0.009</td>
<td>7731</td>
</tr>
<tr>
<td>30</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>921</td>
<td>0.071</td>
<td>0.012</td>
<td>7204</td>
</tr>
</tbody>
</table>

Different Approaches

- Traditional
- Multi-Models with Parent Displacement
- Multi-Models with Parent Displacement and RTR

Average Fitness Calls

Different Approaches

- Traditional
- Multi-Models with Parent Displacement
- Multi-Models with Parent Displacement and RTR

Population Size

- Traditional
- Multi-Models with Parent Displacement
- Multi-Models with Parent Displacement and RTR

Different Approaches
Implementation
## Results

<table>
<thead>
<tr>
<th>Problem Size</th>
<th>#Models</th>
<th>RTR?</th>
<th>SoftMax?</th>
<th>Parent Displacement?</th>
<th>Pop. Type</th>
<th>Final Population size</th>
<th>Average algorithm time</th>
<th>Average model-building time</th>
<th>Average fitness calls</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Single</td>
<td>1312</td>
<td>0.028</td>
<td>0.019</td>
<td>6822</td>
</tr>
<tr>
<td>30</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Single</td>
<td>1562</td>
<td>0.038</td>
<td>0.009</td>
<td>7731</td>
</tr>
<tr>
<td>30</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>Single</td>
<td>921</td>
<td>0.071</td>
<td>0.012</td>
<td>7204</td>
</tr>
<tr>
<td>30</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Multi</td>
<td>3125</td>
<td>0.058</td>
<td>0.015</td>
<td>14871</td>
</tr>
</tbody>
</table>

![Graph showing population size and average fitness calls for different approaches](image)

**Different Approaches**
- Traditional
- Single Pop.
- Single Pop. and RTR
- Multi Pop.

**Average Fitness Calls**
- Traditional
- Single Pop.
- Single Pop. and RTR
- Multi Pop.
Multi-structural Problems

- Deceptive Behavior!

\[
MSP1(x) = \begin{cases} 
\alpha + f_{\text{trap}(m,k)}(x_2, \ldots, x_n) & x_1 = 1 \\
(n - 1) - f_{\text{OneMax}}(x_2, \ldots, x_n) & x_1 = 0 
\end{cases}
\]

\[
MSP2(x) = \max \{\alpha + f_{\text{trap}(m,k)}(x), (n - 1) - f_{\text{OneMax}}(x)\}
\]
### MSP1 Test Results

The function $MSP1(x)$ is defined as:

$$MSP1(x) = \begin{cases} 
\alpha + f_{trap(m,k)}(x_2, ..., x_n) & x_1 = 1 \\
(n - 1) - f_{oneMax}(x_2, ..., x_n) & x_1 = 0 
\end{cases}$$

<table>
<thead>
<tr>
<th>Problem Size</th>
<th>#Models</th>
<th>RTR?</th>
<th>Parent Replacement?</th>
<th>Final Population size</th>
<th>Average algorithm time</th>
<th>Average model-building time</th>
<th>Average fitness calls</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>5750</td>
<td>0.093</td>
<td>0.066</td>
<td>24897</td>
</tr>
<tr>
<td>31</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>6875</td>
<td>0.1</td>
<td>0.022</td>
<td>27840</td>
</tr>
<tr>
<td>31</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>4875</td>
<td>0.163</td>
<td>0.022</td>
<td>23347</td>
</tr>
<tr>
<td>31</td>
<td>5</td>
<td>0</td>
<td>1</td>
<td>8250</td>
<td>0.311</td>
<td>0.042</td>
<td>33907</td>
</tr>
<tr>
<td>31</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>6250</td>
<td>0.218</td>
<td>0.019</td>
<td>31812</td>
</tr>
</tbody>
</table>

**Bar Charts**

- **Population Size**
  - Traditional
  - Three Models
  - Three Models and RTR
  - Five Models
  - Five Models and RTR

- **Average Fitness Calls**
  - Traditional
  - Three Models
  - Three Models and RTR
  - Five Models
  - Five Models and RTR
MSP2 Test Results

\[
MSP2(x) = \max \left\{ \alpha + f_{\text{trap}(m,k)}(x), (n-1) - f_{\text{oneMax}}(x) \right\}
\]

<table>
<thead>
<tr>
<th>Problem Size</th>
<th>#Models</th>
<th>RTR?</th>
<th>Parent Replacement?</th>
<th>Final Population size</th>
<th>Average algorithm time</th>
<th>Average model-building time</th>
<th>Average fitness calls</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>164000</td>
<td>4.231</td>
<td>2.949</td>
<td>596960</td>
</tr>
<tr>
<td>30</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>78000</td>
<td>2.327</td>
<td>0.379</td>
<td>238680</td>
</tr>
<tr>
<td>30</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>25000</td>
<td>0.697</td>
<td>0.089</td>
<td>201750</td>
</tr>
<tr>
<td>30</td>
<td>5</td>
<td>0</td>
<td>1</td>
<td>82000</td>
<td>2.797</td>
<td>0.237</td>
<td>256660</td>
</tr>
<tr>
<td>30</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>18500</td>
<td>0.87</td>
<td>0.067</td>
<td>185925</td>
</tr>
</tbody>
</table>

Different Approaches

![Population Size](image1)

![Average Fitness Calls](image2)
Conclusions and Suggestions

- The proposed method seems Nice!
- Method of learning: Sporadic or Incremental
- Number of Models
References

- Martin Pelikan’s personal website
- Wikipedia
Special Thanks to...

Dr. Nikanjam

Dr. Naser Sharif

Saeed Ghadiri
Special Thanks to...

Dr. Taghirad

KN2C Robotic Team
Q&A

You have Questions
We have Answers