Linear and Combinatorial Optimizations by Estimation of Distribution Algorithms

Topon Kumar Paul
Department of Frontier Informatics
The University of Tokyo

Contents

Genetic Algorithms

- **Genetic Algorithms are popular because**
  -Easy to use
  -Can be applied without prior knowledge
  -Parallelization can used
- **Problems in Genetic Algorithms**
  -Complex interactions among variables
  -Sometimes very difficult to get solutions

Estimation of Distribution Algorithm (EDA)

- Use probabilistic models for recombination
- Learn and sample that probabilistic models to generate new solutions
- Selection and replacement strategy of GA can be used
- Allow adaptation and improve expressiveness

Other Names of EDAs

- Probabilistic Model Building Genetic Algorithms (PMBGAs)
- Distribution Estimation Algorithms (DEAs)
- Iterated Density Estimation Algorithms (IDEAs)

Pseudocode of EDA

1. Generate initial population
2. Select some promising individuals
3. Build probabilistic model using selected individuals
4. Sample the model to generate new individuals
5. Replace old population by new individuals
6. If Termination_Criteria not satisfied go to 2

Discrete EDAs

- **Univariate**: no dependency among variables
  Example:**UMDA,PBIL,CGA**
- **Bivariate**: pairwise dependency
  Example:**MIMIC,COMIT,BMDA**
- **Multivariate**:multiple dependencies
  Example:**BOA,EBNA, ECGA,FDA,LFDA**

Univariate EDAs

- **Univariate Marginal Distribution Algorithm(UMDA)** (Mühlenbein et.al, 1996)
- **Population Based Incremental Learning (PBIL)** (Baluja, 1994)
- **Compact Genetic Algorithm (CGA)** (Harik et.al,1998)
- **PBIL and CGA use probability vector (*p*) while UMDA uses population**
- **PBIL and CGA use different update rules for *p***

UMDA

- **Uses a probability vector** $p=(p_1,p_2,\ldots,p_n)$
  - p_i is the probability of *1* in *i*th position
- **Compute *p* from selected individuals**
- **Generate *1* in the position *i* with prob.** p_i
- **Approximates the behavior of uniform crossover of GA**
- **Problem**: Joint probability may be *0 due to* some $p_i=0$ and may converge to local optimal

Laplace Correction for UMDA

- **Now joint probability distribution>0**
- **Marginal probability of each variable** X_i **is calculated as**

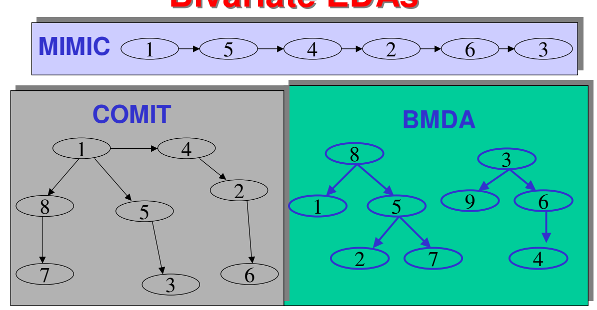
$$p(x_i \mid D^{Se}_{l-1}) = \frac{\sum_{j=1}^{N} \delta_j (X_i = x_i \mid D^{Se}_{l-1}) + 1}{N + r_i}$$

\- r_i is the number of different values X_i may take

Bivariate EDAs

- **Mutual Information Maximization for Input Clustering (MIMIC)** (De Bonet et al.,1997)
- **Combining Optimizers with Mutual Information Trees (COMIT)** (Baluja et al., 1997)
- **Bivariate Marginal Distribution Algorithm (BMDA)**(Pelikan et al. 1999)
- They learn two order structures and are easy to use

Graphical Representations of Bivariate EDAs

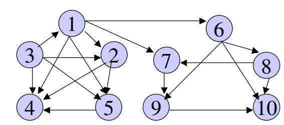


Multivariate EDAs

- **Extended Compact Genetic Algorithm (ECGA)** (Harik 1999)
- **Bayesian Optimization Algorithm (BOA)** (Pelikan et al.,2000)
- **Estimation of Bayesian Networks Algorithm (EBNA)** (Etxeberria et al., 1999)
- **Factorized Distribution Algorithm (FDA)** (Mühlenbein et al.,1999)
- **Learning Factorized Distribution Algorithm (LFDA)** (Mühlenbein et al.,1999)

BOA,EBNA and LFDA

- Uses Bayesian Network but different score metrics:
  - **BOA**: Bayesian Dirichlet Equivalence (BDe) metric
  - **EBNA**: BIC, K2+Penalization scores
  - **LFDA**: BIC but restriction on no. of parents



Structure Learning in Bayesian Network

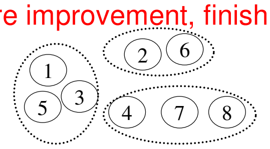
1. **Start with empty graph(no edges)**
2. **Apply primitive graph operators**
3. **Pick the operation that increases score**
4. **Perform that operation**
5. **If no improvement or constraints violates, stop else go to 2.**

FDA

- Applicable to **Additively Decomposable Problem**
- Uses **Boltzmann Seletion** and calculate Boltzmann distribution
- Needs **structure learning** as well as **parameters learning**

ECGA

- Uses **Greedy Search** and **BIC score**
- For Search
  - Start with one bit group
  - Merge two groups if improves
  - No more improvement, finish



Summary of Multivariate EDAs

- **Bayesian Network is mostly used**
- **But learning model structure is very difficult (NP hard)**
- **Can successfully solve problems decomposable into sub problems of bounded difficulty**

Different Problems Considered

1. **OneMax**
2. **Subset Sum**
3. **N-Queen**

- **Algorithm used**: **UMDA with Laplace correction**
- **Selection(for UMDA): Truncation Selection (best half of the population)**
- **Replacement**: **Elitism**

OneMax Function

- **Return sum of the bits**
- **Optimum**: All 1s ({1,1,…,1})
- **Fitness**: sum of variables in an individual
- **Terminate**: when fitness=problem size
- **GA**:One point crossover and mutation

Subset Sum Problem

- **Definition**: What subset of a set of integers has the sum equal to expected weight
- **Example**: Given

  I={1,3,5,6,8,10}, W=14

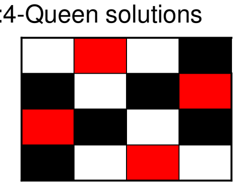
  Solution: *{1,3,10},{3,5,6},{6,8},{1,5,8}*
- **Trivial case**: Sum of all integers=Weight
- **Random case**: Sum of all integers>Weight

UMDA and GA for Subset Sum Problem

- **Fitness: Absolute difference between sum of variables in an individual and expected weight**
- **Termination: Sum of variables in an individual equal to expected weight**
- **GA: one point crossover and mutation**

n-Queen Problem

- **Definition**: place n queens on a $n \times n$ chessboard so that no two queens attack each other i.e. they are not on the same row,column or diagonal
- **How?** If $abs(row_i - row_j) = abs(column_i - column_j)$
- **Example** :4-Queen solutions



Solution of n-Queen Problem

- **Solution Encoded**:$\{X_1, X_2, \ldots, X_n\}$

  -X_i indicates the column position of ith queen at row i
- **Probability Vector**: 2-dimensional

  -Each p_{ij} indicates the probability of variable X_i taking its jth value at position i
- **Fitness**: no. of queens at non-attacking positions

GA for n-Queen Problem

- **Crossover**: Partially Matched Crossover(PMX)
- **Mutation**:Greedy Swap Mutation
- **Termination**:When all the queens are at non attacking positions
- **Very efficient than UMDA**

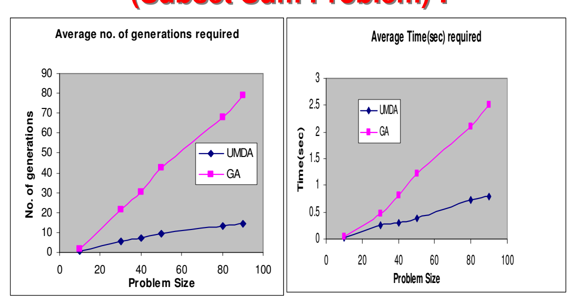
Local Heuristics for n-Queen

- To improve performance of UMDA 2-opt algorithm as local heuristic used
- But GA with PMX still better

Probabilistic Modification for n-Queen

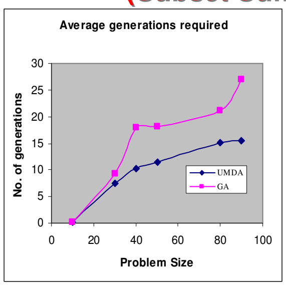
- When a variable is selected for a position, its probability for other positions must be zero
- If Roulette Wheel selection is used probabilistic modification ensure distinct values for different positions

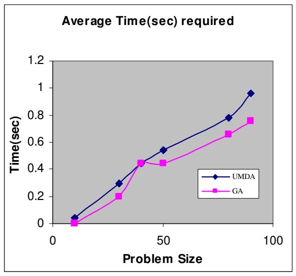
Experimental Results (Subset Sum Problem)-I



Average no. of generations and time(sec) for trivial solution

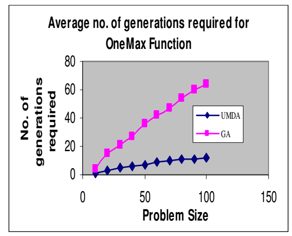
Experimental Results (Subset Sum Problem)-II

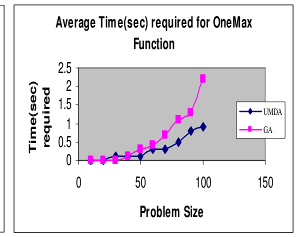




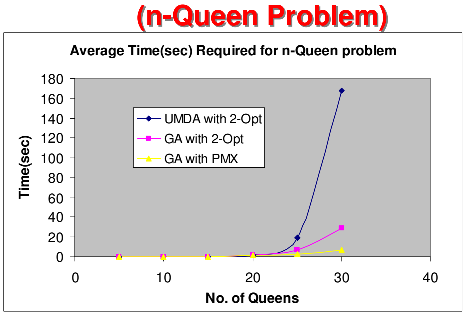
Average no. of generation and time required for random sum

Experimental Results (OneMax Function)





Experimental Results (n-Queen Problem)



Summary

- UMDA better than GA for some linear problems of independent variables
- To Capture higher order dependency we have to consider BOA, ECGA, FDA or EBNA
- BOA and EBNA may be general approach for a problem

Future Works

- Understand the behavior of BOA, EBNA, FDA etc.
- Apply EDA to permutation domains
- Develop an EDA for discrete and continuous domain