Estimation of Distribution Algorithm for Optimization of System Reliability

Shang Gao, Ling Qiu

1 School of Computer Science and Engineering, Jiangsu University of Science and Technology, Zhenjiang 212003, China, Email: gaoshang@sohu.com
2 Artificial Intelligence of Key Laboratory of Sichuan Province, Sichuan University of Science and Engineering, Zigong 643000, China

Abstract

As an important branch of evolutionary algorithms, an Estimation of Distribution Algorithm (EDA) uses some selected individuals to build a probability model to produce offspring by sampling the probability model, which is a process of statistically learning from the selected individuals and then probabilistically sampling the probability model. A redundancy optimization model is given and many optimization methods to solve optimum model and their advantages and shortages are analyzed. A new method based on estimation distribution algorithm for optimization of system reliability is put forward and better population selection proportions are analyzed. Compared with simulated annealing algorithm, genetic algorithm and particle swarm algorithm, the estimation distribution algorithm is more effective through result. This method has good scalability, and it can be modified solve general combinatorial optimization problems.

Keywords Reliability Optimization, Estimation Distribution Algorithm, Simulated Annealing Algorithm, Genetic Algorithm, Particle Swarm Algorithm

1. Introduction

Reliability engineering is an engineering field that deals with the study, evaluation, and life-cycle management of reliability. The reliable performance of a system for a predefined mission time under various conditions is very important in many industrial and military applications. In more complex system, if the reliability is less than the higher index, the greater the losses will cause. The reliability problem is one of the key factors that must be considered in system design, research and running. In the formulation of a series-parallel system problem, for each subsystem, multiple component choices (assuming an unlimited supply of each component) are used in parallel. For those systems, with known cost, reliability, and weight, system design and component selection becomes a combinatorial optimization problem. The problem is then to select the optimal combination of parts and redundancy levels to meet cost and weight constraints collectively while maximizing system reliability.

Estimation of distribution algorithms (EDAs) are stochastic optimization techniques that explore the space of potential solutions by building and sampling explicit probabilistic models of promising candidate solutions. This explicit use of probabilistic models in optimization offers some significant advantages over other types of metaheuristics. EDAs were successfully applied to optimization of large spin glass instances in two-dimensional and three-dimensional lattices, military antenna design, multiobjective knapsack, groundwater remediation design, aminoacid alphabet reduction for protein structure prediction, identification of clusters of genes with similar expression profiles, economic dispatch, forest management, portfolio management, cancer chemotherapy optimization, environmental monitoring network design, and others. In this paper, a new method for knapsack problem is put forward based on estimation distribution algorithm and better population selection proportions are analyzed. Estimation of Distribution Algorithms (EDAs) is a new area of Evolutionary Computation. In EDAs there is neither crossover nor mutation operator. New population is generated by sampling the probability distribution, which is estimated from a database containing selected individuals of previous generation.

Since Estimation of Distribution Algorithms (EDAs) were proposed by Baluja in 1994 [1-3], EDAs quickly become an important branch of evolutionary algorithms because they have better mathematical foundation than other evolutionary algorithms. On the basis of statistical learning theory, EDAs use
some individuals selected from the population at the current evolutionary generation to build a
probability model and then produces offspring for the next generation by sampling the probability
model in a probabilistic way. A lot of investigations in [1-3] show that EDAs have good optimization
performance in both combinatorial problems and numeric optimization problems. Until now there are
many studies about EDAs, but EDAs mainly consist of several types: Population based incremental
learning (PBIL) [102], univariate marginal distribution algorithm (UMDA), compact genetic algorithm
(CGCA), mutual-information-maximizing input clustering algorithm (MIMIC), bivariate marginal
distribution algorithm (BMDA), factorized distribution algorithm (FDA), Bayesian optimization
algorithm (BOA), extended compact genetic algorithm (ECGA) and estimation of Bayesian network
algorithm (EBNA). UMDA works well only in the solution of linear problems with independent
variables, so it requires extension as well as application of local heuristics for combinatorial
optimizations. PBIL uses vector probabilities instead of population and has good performance for
solving problems with independent variables in binary search space. CGA independently deals with
each variable and needs less memory than simple genetic algorithm. MIMIC searches the best
permutation of the variables at each generation to find the probability distribution through using
Kullback-Leibler distance. BMDA is mainly based on the construction of a dependency graph, which
is acyclic but does not necessarily have to be a connected graph. FDA integrates evolutionary
algorithms with simulated annealing. This method requires additively decomposed function and the
factorization of the joint probability distribution remains same for all iterations. BOA applies Bayesian
network and Bayesian Dirichlet metric to estimate joint probability distributions, thus, it can take
advantage of the prior information about the problem. ECGA factorizes the joint probability
distribution as a product of marginal distributions of variable size. EBNA employs Bayesian network
for the factorization of the joint probability distribution and BIC score. In lecture [4], a novel EDA
(nEDA) is proposed by simultaneously and appropriately introducing Gaussian and Cauchy probability
models in a single algorithm. In lecture [5], different probability models have different characteristics
with respect to exploration and exploitation. An estimation of distribution algorithm (EDA) is proposed
to optimize the combination of the rules. The superiority of this approach is verified by extensive
computational experiments and comparisons.

2. Redundancy optimization model

![Figure 1 Series-parallel system configuration](image)

We assume a mixed series-parallel system has n stages. Our aim is to optimal the cost of system
subject to constraint on system reliability . The schematic is in Figure 1. Find the \( x_i \), \( i = 1, 2, \ldots, n \)

\[
\text{minimize } C_s = \sum_{i=1}^{n} c_i x_i
\]

subject to : \( \prod_{i=1}^{n} R_i(x_i) \geq R_0 \) (1)

where \( C_s \)=cost of system, \( R_i \)=reliability of system, \( c_i \)=cost of component \( i \), \( x_i \)=number of
redundancies of component \( i \), \( x_i \geq 1 \), \( R_i(x_i) \)=reliability of stage \( i \), \( R_i(x_i) = 1 - (1 - p_i)^x_i \),
\( R_0 \)=lower bound on \( R_s \), \( p_i \)=reliability of component \( i \).

The problem is NP-hard, only some special cases can be solved efficiently. Thus, many approaches
have been proposed to find optimal design in reasonable time. Tillman, Hwang, and Kuo (1977)[6]
provide a thorough review related to optimal system reliability with redundancy. They divided optimal system reliability models with redundancy into series, parallel, series-parallel, parallel-series, standby, and complex classes. They also categorized optimization methods into integer programming, dynamic programming, linear programming, geometric programming, generalized Lagrangian functions, and heuristic approaches. The authors concluded that many algorithms have been proposed but only a few have been demonstrated to be effective when applied to large-scale nonlinear programming problems. Also, none has proven to be generally superior. Fyffe, Hines, and Lee (1968) [7] provide a dynamic programming algorithm for solving the system reliability allocation problem. As the number of constraints in a given reliability problem increases, the computation required for solving the problem increases exponentially. In order to overcome these computational difficulties, the authors introduce the Lagrange multiplier to reduce the dimensionality of the problem. Nakagawa and Miyazaki (1981) [8] show a more efficient algorithm compared to dynamic programming using the Lagrange multiplier. In their algorithm, the authors use surrogate constraints obtained by combining multiple constraints into one constraint. Misra and Sharma (1991) [9] present a simple and efficient technique for solving integer programming problems such as the system reliability design problem. Hwang, Tillman and Kuo (1979) [10] use the generalized Lagrangian function method and the generalized reduced gradient method to solve nonlinear optimization problems for reliability of a complex system. The same authors (1977) [11] also present a mixed integer programming approach to solve the reliability problem. They maximize the system reliability as a function of component reliability level and the number of components at each stage. Using a genetic algorithm (GA) approach, Coit and Smith (1996) [12] provide a competitive and robust algorithm to solve the system reliability problem. The authors use a penalty guided algorithm which searches over feasible and infeasible regions to identify a final, feasible optimal, or near optimal, solution. For a fixed design configuration and known incremental decreases in component failure rates and their associated costs, Painion and Campbell (1995) [13-15] also use a GA to find a maximum reliability solution to satisfy specific cost constraints. Gao Shang (2004) [16] use ant colony algorithm to solve optimum model. Particle swarm optimization algorithm, which combines with idea of genetic algorithm, is put forward (Gao shang 2006) [17-19]. In this paper, a new method for optimization of system reliability is put forward based on estimation distribution algorithm and better population selection proportions are analyzed. Compared with simulated annealing algorithm, genetic algorithm and particle swarm algorithm, the estimation distribution algorithm is more effective through result.

3. Basic estimation of distribution algorithms

Estimation of distribution algorithms (EDAs), sometimes called probabilistic model-building genetic algorithms (PMBGAs), are stochastic optimization methods that guide the search for the optimum by building and sampling explicit probabilistic models of promising candidate solutions. Optimization is viewed as a series of incremental updates of a probabilistic model, starting with the model encoding the uniform distribution over admissible solutions and ending with the model that generates only the global optima.

EDAs belong to the class of evolutionary algorithms. The main difference between EDAs and most conventional evolutionary algorithms is that evolutionary algorithms generate new candidate solutions using an implicit distribution defined by one or more variation operators, whereas EDAs use an explicit probability distribution encoded by a Bayesian network, a multivariate normal distribution, or another model class. In EDAs the new population of individuals is generated without using neither crossover nor mutation operators. Instead, the new individuals are sampled starting from a probability distribution estimated from the database containing only selected individuals from the previous generation. Figure 1 illustrates the flowchart of EDA.

The general procedure of an EDA is outlined in the following (Paul T K, 2002):

Step 1 \( t = 0 \);
Step 2 initialize model \( M(0) \) to represent uniform distribution over admissible solutions
Step 3 while (termination criteria not met)
  Step 3.1 \( P = \) generate \( N \geq 0 \) candidate solutions by sampling \( M(t) \)
  Step 3.2 \( F = \) evaluate all candidate solutions in \( P \)
  Step 3.3 \( M(t+1) = \) adjust model\((P,F,M(t))\)
  Step 3.4 \( t = t + 1 \)
Randomly generate an initial individual

Select the number of individuals

Estimate the probability distribution among the selected individuals

Move the particles in the search space and evaluate their fitness

Generate the next generation by probabilistically selecting particles to produce offspring

Maximum number of iteration?

Output the optimal individual

**Figure 1.** illustrates the flowchart of EDA.

Using explicit probabilistic models in optimization allowed EDAs to feasibly solve optimization problems that were notoriously difficult for most conventional evolutionary algorithms and traditional optimization techniques, such as problems with high levels of epistasis. Nonetheless, the advantage of
EDAs is also that these algorithms provide an optimization practitioner with a series of probabilistic models that reveal a lot of information about the problem being solved. This information can in turn be used to design problem-specific neighborhood operators for local search, to bias future runs of EDAs on a similar problem, or to create an efficient computational model of the problem.

4. Solving optimization of system reliability by estimation of distribution algorithm

Firstly, we transform (1)(constrained problem) into a single unconstrained problem.

\[
\min \sum_{i=1}^{n} c_i x_i + M \left\{ \min \left\{ 0, \prod_{i=1}^{n} \left(1 - R_i\right)^{y_i} - R_0 \right\} \right\}^{1/2} \tag{2}
\]

where \( M > 0 \) is a large number. The vector \( X = (x_1, x_2, \cdots, x_n)^T \) is the solution of reliability optimization problem. For simple convenience, we assume that the scope of the variables were within the range \([1, \max x]\). \( p_{ij} \) is the probability of \( x_i = j \), so probability matrix of \( X = (x_1, x_2, \cdots, x_n)^T \) is \( P = (p_{ij})_{n \times \max x} \).

The estimation of distribution algorithm for reliability optimization problem is as follows:

Step 1 Using the uniform design technique, for each variable are the probability of random values within the interval \([1, \max x]\). Generate \( N \) individuals constitute the initial population.

Step 2 Assess the fitness of all individuals in the initial population, and retain the best solution.

Step 3 Order the population by fitness in descending sorting, and choose the optimal \( m \) individuals (\( m \leq N \)).

Step 4 Build a probability model \( p_{ij} \) based on the statistical information extracted from the selected \( m \) solutions in the current population.

Step 5 Sample \( N \) new solutions from this build probability models.

Step 6 If the given stopping condition (up to the required number of iterations \( n_{\text{max}} \)) is not met, go to step 2.

The estimation of distribution algorithm's time complexity is estimated as follows: The time to calculate the fitness operation is the longest, so the time complexity of algorithm is about \( O(N n_{\text{max}}) \).

5. Numerical example

Consider a system that is composed of 5 components. The reliability of components are \( p_1 = 0.96 \), \( p_2 = 0.93 \), \( p_3 = 0.85 \), \( p_4 = 0.80 \) and \( p_5 = 0.75 \). The cost of components are \( c_1 = 3 \), \( c_2 = 12 \), \( c_3 = 8 \), \( c_4 = 5 \) and \( c_5 = 10 \). The lower bound on \( R_s \) is \( R_0 = 0.9 \). If we don’t redundant the system, the reliability of system is 0.455328. It can’t meet the requirements, so it must be redundant.

When \( N = 100 \), \( m = 0.4 * N \) and \( \max x = 6 \), the probability matrix is changes shown in table 1. The procession of value is shown in Figure 2. The optimal solution is \((2, 2, 2, 3, 2)\), and the total cost of 81. From Table 1, the probability matrix gradually converges. It converges to the optimal solution with probability 1 when it iterate to 13 times. Since we retain the best solution at each iteration, so it find the optimal solution at the 4th iteration.
Table 1 Process of probability matrix

<table>
<thead>
<tr>
<th>iteration</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.1667</td>
<td>0.1667</td>
<td>0.1667</td>
<td>0.1667</td>
<td>0.1667</td>
<td>0.1667</td>
</tr>
<tr>
<td>1</td>
<td>0.1667</td>
<td>0.1667</td>
<td>0.1667</td>
<td>0.1667</td>
<td>0.1667</td>
<td>0.1667</td>
</tr>
<tr>
<td></td>
<td>0.2750</td>
<td>0.1500</td>
<td>0.1000</td>
<td>0.2000</td>
<td>0.2500</td>
<td>0.0500</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0.2000</td>
<td>0.2750</td>
<td>0.1000</td>
<td>0.1500</td>
<td>0.2500</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0.1750</td>
<td>0.2000</td>
<td>0.2500</td>
<td>0.1250</td>
<td>0.1500</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0.2250</td>
<td>0.2500</td>
<td>0.2250</td>
<td>0.1250</td>
<td>0.1750</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0.3250</td>
<td>0.2750</td>
<td>0.1500</td>
<td>0.1750</td>
<td>0.0750</td>
</tr>
</tbody>
</table>

...  

12  

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

13

0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

The time of iteration

Cost

Figure 2 The procession of value

Compared with simulated annealing algorithm, genetic algorithm and particle swarm algorithm, 100 rounds of computer simulation are conducted for each algorithm, and the results are shown in Table 2. From table 2, the estimation distribution algorithm is more effective through result.

Table 2 Comparison of various algorithms

<table>
<thead>
<tr>
<th>algorithms</th>
<th>Mean</th>
<th>The worst solution</th>
<th>the best solution</th>
<th>Number of the best solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>SA[16]</td>
<td>87.6</td>
<td>109</td>
<td>81</td>
<td>16</td>
</tr>
<tr>
<td>GA[17]</td>
<td>87.3</td>
<td>102</td>
<td>81</td>
<td>24</td>
</tr>
<tr>
<td>basic PSO[12]</td>
<td>87.2</td>
<td>98</td>
<td>81</td>
<td>30</td>
</tr>
<tr>
<td>improved PSO[12]</td>
<td>86.1</td>
<td>97</td>
<td>81</td>
<td>46</td>
</tr>
<tr>
<td>EDA</td>
<td>81.48</td>
<td>87</td>
<td>81</td>
<td>89</td>
</tr>
</tbody>
</table>

The main parameters affecting the performance of the EDA are the number N of the population and selected population number m. When N = 100, it test 100 times, and the statistics are shown in Table 3.
Table 3 The results with different m / N

<table>
<thead>
<tr>
<th>m/N</th>
<th>Mean</th>
<th>The worst solution</th>
<th>the best solution</th>
<th>Number of the best solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>10%</td>
<td>82.58</td>
<td>92</td>
<td>81</td>
<td>67</td>
</tr>
<tr>
<td>20%</td>
<td>81.66</td>
<td>87</td>
<td>81</td>
<td>86</td>
</tr>
<tr>
<td>30%</td>
<td>81.47</td>
<td>86</td>
<td>81</td>
<td>89</td>
</tr>
<tr>
<td>40%</td>
<td>81.48</td>
<td>87</td>
<td>81</td>
<td>89</td>
</tr>
<tr>
<td>50%</td>
<td>81.31</td>
<td>86</td>
<td>81</td>
<td>93</td>
</tr>
<tr>
<td>60%</td>
<td>81.87</td>
<td>87</td>
<td>81</td>
<td>83</td>
</tr>
</tbody>
</table>

From Table 3, if the ratio of m/N is the greater, the effect is the worse. Of course, the ratio m/N is too small, it is easy to fall into local minima. So the ratio of m/ N is 40%-50%, the results were quite good.

6. Conclusions

Reliability optimization problem is a nonlinear mixed-integer planning problem. The estimation of distribution algorithm is applied to optimization of system reliability. The other reliability optimization model can be similar to solved by EDA. Estimation of distribution algorithm can be slightly modified to solve similar nonlinear mixed integer programming problem. The estimation of distribution algorithm can be further improved, such as adding the crossover operators and mutation operators, so the performance may be better.

7. Acknowledgements

This work was partially supported by Artificial Intelligence of Key Laboratory of Sichuan Province (2012RYJ04) , Jiangsu 333 Project and Qing Lan Project.

8. References