

A Hybrid Estimation of Distribution Algorithm for the Minimal Switching Graph Problem

Maolin Tang

Faculty of Information Technology
Queensland University of Technology
2 George Street, Brisbane, Australia
m.tang@qut.edu.au

Raymond Y. K. Lau

Department of Information Systems
City University of Hong Kong
Tat Chee Avenue, Kowloon, Hong Kong
raylau@cityu.edu.hk

Abstract

Minimal Switching Graph (MSG) is a graphical model for the constrained via minimization problem — a combinatorial optimization problem in integrated circuit design automation. From a computational point of view, the problem is NP-complete. In this paper we present a new approach to the MSG problem using hybrid Estimation of Distribution Algorithms (EDAs). This approach uses a Univariate Marginal Distribution Algorithm (UMDA) to sample start search points and employs a hill-climbing algorithm to find a local optimum in the basins where the start search points are located. By making use of the efficient exploration of the UMDA and the effective exploitation of the hill-climbing algorithm, this hybrid EDA can find an optimal or near-optimal solution efficiently and effectively. The hybrid EDA has been implemented and compared with the UMDA and the hill-climbing algorithm. Experimental results show that the hybrid EDA significantly outperforms both the UMDA and the hill-climbing algorithm.

1. Introduction

Minimal Switching Graph (MSG) is a graph-theoretic model for the constrained via minimization problem — a combinatorial optimization problem in integrated circuit design automation [1]. From a computational point of view, the problem is NP-complete [2].

Estimation of Distribution Algorithm (EDA) is a new paradigm of evolutionary computation. It was introduced into the field of evolutionary computation for the first time by Mühlenbein and Paaß in 1996 [3]. 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 from previous generation. In EDAs the problem specific in-

teractions among the variables of individuals are expressed explicitly through the joint probability distribution associated with the variables in the selected individuals.

Different approaches have been proposed for the estimation of probability distribution. One of the approaches is the Univariate Marginal Distribution Algorithm (UMDA) introduced by Mühlenbein in 1998 [4]. The pseudocode for the UMDA is as follow:

1. randomly generate M individuals (the initial population) and store them in D_0 ;
2. $g = 1$;
3. while the stopping criterion has not met
 - (a) select $N \leq M$ individuals from D_{g-1} according to the selection method and store them in D_{g-1}^S ;
 - (b) estimate the joint probability distribution using $p_g(x) = p(x|D_{g-1}^S) = \prod_{i=1}^n p_g(x_i) = \prod_{i=1}^n \frac{\sum_{j=1}^N \delta_j(X_i=x_i|D_{g-1}^S)}{N}$;
 - (c) sample M individuals (the new population) from $p_g(x)$.

where

$$\delta_j(X_i = x_i|D_{g-1}^S) = \begin{cases} 1, & \text{if in the } j^{th} \text{ case of } D_{g-1}^S, \\ & X_i = x_i; \\ 0, & \text{otherwise.} \end{cases}$$

This paper presents a hybrid EDA for the MSG problem. The hybrid algorithm uses a UMDA to sample start search points and employs a hill-climbing algorithm to find a local optimum in the basins where the start search points are located. By making use of the efficient exploration of the UMDA and the effective exploitation of the hill-climbing algorithm, this hybrid EDA can find an optimal or near-optimal solution efficiently and effectively. The hybrid EDA has been implemented and compared with both

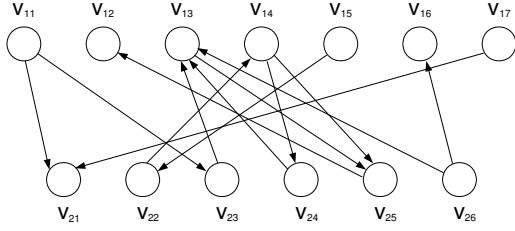


Figure 1. A directed bigraph

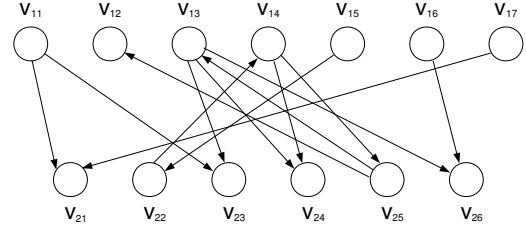


Figure 2. A switching graph

the UMDA and the hill-climbing algorithm. Experimental results show that the performance of the hybrid EDA is significantly better than that of the UMDA and the hill-climbing algorithm.

The remaining paper is organized as follows. In Section 2 the MSG problem is formalized. Then an encoding scheme for the MSG problem, which is used by the algorithms, is presented in Section 3. Section 4 and Section 5 present the UMDA and the hill-climbing algorithm respectively. Section 6 elaborates the hybrid EDA. Empirical comparisons between the hill-climbing algorithm, the UMDA and the hybrid EDA are conducted and experimental results are presented in Section 7. Finally we draw conclusions and discuss the hybrid EDA in Section 8.

2. The Minimal Switching Graph Problem

A *directed bigraph* is a directed graph whose vertices can be partitioned into two disjoint sets such that no vertices within the same set are adjacent. A directed bigraph can be denoted by $G = (V_1 \cup V_2, E)$, where V_1 and V_2 are two disjoint vertex sets, and E is a set of directed edges, or arcs. Figure 1 shows a directed bigraph, where $V_1 = \{v_{11}, v_{12}, v_{13}, v_{14}, v_{15}, v_{16}, v_{17}\}$, $V_2 = \{v_{21}, v_{22}, v_{23}, v_{24}, v_{25}, v_{26}\}$, and $E = \{ \langle v_{11}, v_{21} \rangle, \langle v_{11}, v_{23} \rangle, \langle v_{12}, v_{22} \rangle, \langle v_{13}, v_{23} \rangle, \langle v_{13}, v_{25} \rangle, \langle v_{14}, v_{24} \rangle, \langle v_{14}, v_{25} \rangle, \langle v_{15}, v_{25} \rangle, \langle v_{16}, v_{26} \rangle, \langle v_{17}, v_{21} \rangle \}$.

A directed bigraph G can be represented in a matrix M defined in Equation 1.

$$M = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1|V_2|} \\ a_{21} & a_{22} & \cdots & a_{2|V_2|} \\ \vdots & \vdots & \ddots & \vdots \\ a_{|V_1|1} & a_{|V_1|2} & \cdots & a_{|V_1||V_2|} \end{bmatrix} \quad (1)$$

where

$$a_{ij} = \begin{cases} 1, & \text{if } \langle v_i, v_j \rangle \in E, v_i \in V_1, \text{ and } v_j \in V_2; \\ -1, & \text{if } \langle v_i, v_j \rangle \in E, v_i \in V_2, \text{ and } v_j \in V_1; \\ 0, & \text{otherwise.} \end{cases}$$

For example, the directed bigraph shown in Figure 1 is represented by the following matrix:

$$M = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & -1 & -1 & 1 & -1 \\ 0 & -1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Given a directed bigraph G , a *switching graph* of G , denoted by $G(S)$, is defined as the directed graph obtained by reversing the direction of all the arcs being incident to the vertices in $S \subseteq V_1$. Figure 2 displays a switching graph of G , $G(\{v_{13}, v_{16}\})$.

The corresponding matrix representation of a switching graph $G(S)$ can be obtained from the matrix representation of G through multiplying all values in the corresponding rows of S by -1 . For switching graph $G(\{v_{13}, v_{16}\})$, the corresponding matrix representation can be obtained through multiplying the values in the 3rd and 6th by -1 as below:

$$M = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 1 & -1 & 1 \\ 0 & -1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

It is pointed out that a directed bigraph G itself is a special switching graph of G in which $S = \phi$. A directed bigraph $G = (V_1 \cup V_2, E)$ has $2^{|V_1|}$ switching graphs.

The vertices in V_2 are categorized into two types: *desirable vertices* and *undesirable vertices*. Desirable vertices are those vertices whose either in-degree or out-degree is zero, and undesirable vertices are those vertices whose both in-degree and out-degree are not zero. Desirable and undesirable vertices can be easily counted from the corresponding matrix representation. If a vertex is desirable vertex if and only if all non-zero values in the corresponding column are either all 1 or all -1 (not both); otherwise, the vertex is a undesirable vertex.

Given a directed bigraph G , the objective is to find a switching graph $G(S)$ such that the number of desirable vertices is maximal. This is so-called MSG problem.

3. Encoding Scheme for the MSG Problem

Suppose $G = (V_1 \cup V_2, E)$ is a directed bigraph, and $G(S)$ is a switching graph of G . Let $m = |V_1|$ and $n = |V_2|$. Then, a switching graph $G(S)$ can be represented as a sequence of m bits, $x_1x_2 \cdots x_m$, where

$$x_i = \begin{cases} 1, & \text{if } v_i \in S; \\ 0, & \text{otherwise.} \end{cases}$$

and $1 \leq i \leq n_1$.

For example, the switching graph $G(\{v_{13}, v_{16}\})$ shown in Figure 2 is encoded as 0010010, and $G = G(\phi)$ is encoded 0000000. In the hill-climbing algorithm, the UMDA and the hybrid EDA, a solution (switching graph) is encoded by such a sequence.

Given an MSG problem represented by the matrix $M_{|V_1| \times |V_2|}$, there is a one-to-one correspondence between the switching graphs and the set of binary strings of m bits.

The fitness value of $x_1x_2 \cdots x_m$ is defined in Equation 2.

$$f(x_1x_2 \cdots x_m) = |V_d| \quad (2)$$

In the above equation, V_d is the set of the desirable vertices in the corresponding switching graph $G(S)$. This fitness function is used by the hill-climbing algorithm, the UMDA, and the hybrid EDA presented in the following sections.

4. The Hill-Climbing Algorithm

Given an MSG problem G , the hill-climbing algorithm starts its search from the switching graph $G(\phi)$, and iteratively explores the neighboring switching graphs of the current switching graph one by one to see if it is better than the current one. If it is, then the current switching graph is replaced by the neighboring switching graph. The search process is repeated until no better switching graph can be found. The final switching graph is the solution to the MSG problem.

According to the encoding scheme, the starting switching graph $G(\phi)$ is encoded in $00 \cdots 0$. The neighboring switching graphs of a switching graph $x_1x_2 \cdots x_m$ are those switching graphs whose Hamming distance to the switching graph is 1. The Hamming distance between two switching graphs $x_1^1x_2^1 \cdots x_m^1$ and $x_1^2x_2^2 \cdots x_m^2$ is defined in Equation 3.

$$H(x_1^1x_2^1 \cdots x_m^1, x_1^2x_2^2 \cdots x_m^2) = \sum_{i=1}^m |x_i^1 - x_i^2| \quad (3)$$

A switching graph $x_1x_2 \cdots x_m$ has m neighbors, $\overline{x_1}x_2 \cdots x_m, x_1\overline{x_2} \cdots x_m, \cdots, x_1x_2 \cdots \overline{x_m}$.

The description of the hill-climbing algorithm is as below. The input of the algorithm is a directed bigraph G represented as a matrix M , and the output is a switching graph $G(S)$ of G , which is also represented by a matrix.

1. set current search point $x = 00 \cdots 0$;
2. use M to calculate the total number of desirable vertices in x , d_x ;
3. improved = true;
4. while (improved = true) do
 - (a) improved = false;
 - (b) for each of the neighbors of x , x^n :
 - i. get the corresponding matrix of x^n , M^n ;
 - ii. use M^n to calculate the total number of desirable vertices in x^n , d_{x^n} ;
 - iii. if $d_{x^n} > d_x$, then
 - A. $x = x^n$;
 - B. $d_x = d_{x^n}$;
 - C. improved = true;
 - D. go to (a) ;
5. output the corresponding matrix of x .

The hill-climbing algorithm is a local optimization algorithm. It can exploit the information about the current search point effectively. However, the search may be trapped at a local optimum.

5. The Univariate Marginal Distribution Algorithm

Given an MSG problem G , the UMDA starts with randomly generating an initial population of $PopSize$ individuals, each of which is encoded in a binary string of $|V_1|$ bits, by sampling the probability distribution: $p_0 = \prod_{i=1}^{|V_1|} p_0(X_i)$, where $p_0(X_i) = 0.5$ for $i = 1, 2, \cdots, |V_1|$. Thus, the joint probability distribution is factorized as a product of $|V_1|$ univariate marginal probability distributions, each of which follows Bernoulli distribution with parameter value equaling to 0.5.

Once the initial population is created, the UMDA starts evolving the population for Gen generations. In each generation, the best $PopSize/2$ individuals are selected from the population and the selected individuals are characterized using the joint probability distribution

$$p_g(x) = p(x|D_{g-1}^S) = \prod_{i=1}^n p_g(x_i)$$

$$= \prod_{i=1}^n \frac{\sum_{j=1}^N \delta_j(X_i = x_i | D_{g-1}^S)}{N}$$

Then, a new population is obtained by sampling the joint probability $p_g(x)$. The selecting, learning and sampling process is repeated for Gen times to produce a satisfactory switching graph.

The UMDA is described as below:

1. randomly generate $PopSize$ individuals (the initial population) and store them in D_0 ;
2. for $g = 1$ to Gen :
 - (a) select the best $PopSize/2$ individuals from D_{g-1} and store them in D_{g-1}^S ;
 - (b) estimate the joint probability distribution using $p_g = p(x | D_{g-1}^S) = \prod_{i=1}^n p_g(x_i) = \prod_{i=1}^n \frac{\sum_{j=1}^N \delta_j(X_i = x_i | D_{g-1}^S)}{N}$;
 - (c) sample $PopSize$ individuals from $p_g(x)$.

In the above algorithm,

$$\delta_j(X_i = x_i | D_{g-1}^S) = \begin{cases} 1, & \text{if in the } j^{th} \text{ case of } D_{g-1}^S, \\ & X_i = x_i; \\ 0, & \text{otherwise.} \end{cases}$$

6. The Hybrid of the UMDA and the Hill-Climbing Algorithm

The hybrid EDA uses the UMDA to sample start search points and employs the hill-climbing algorithm to find a local optimum in the basins where the start search points are located. By making use of the efficient exploration of the UMDA and the effective exploitation of the hill-climbing algorithm, this hybrid EDA can find an optimal or near-optimal solution efficiently and effectively.

The hybrid EDA is as follow:

1. randomly generate $PopSize$ individuals (the initial population) and store them in D_0 ;
2. use the hill-climbing algorithm to optimize the individuals in the population;
3. for $g = 1$ to Gen :
 - (a) select the best $PopSize/2$ individuals from D_{g-1} and store them in D_{g-1}^S ;
 - (b) estimate the joint probability distribution using $p_g(x) = p(x | D_{g-1}^S) = \prod_{i=1}^n p_g(x_i) = \prod_{i=1}^n \frac{\sum_{j=1}^N \delta_j(X_i = x_i | D_{g-1}^S)}{N}$;
 - (c) sample $PopSize$ individuals from $p_g(x)$;
 - (d) use the hill-climbing algorithm to optimize the individuals in the population.

7 Empirical Comparisons between the Hill-Climbing Algorithm, the UMDA and the Hybrid EDA

The hill-climbing algorithm, the UMDA and the hybrid EDA have been implemented in C#. In order to perform fair comparisons, we have developed a program to randomly generate SMG test problems. The program takes two parameters, $|V_1|$ and $|V_2|$, and generates a bigraph $G = (V_1 \cup V_2, E)$. The program firstly generates two vertex sets V_1 and V_2 . Then, for each of the vertex v in V_2 the program generates a degree $d(v)$, where $2 \leq d(v) \leq 4$, and then randomly generates $d(v)$ directed edges associated with v .

Using the program we have generated 20 test problems. When we generated the 20 test problems using the program, we set the first parameter $|V_1| = 100$, and the second parameter $|V_2|$ from 50 to 145 with increment of 5. Thus, for all the randomly generated test problems the search spaces are all 2^{100} , but the numbers of local optima are different. Generally, the more mutual constraints among the vertices in V_1 in a test problem, the more local optima the test problem has, and the more challenging the test problems is (it is more likely trapped in a local optimum). The characteristics of the randomly generated test problems are shown in Table 1. In the table, $|V_1|$ and $|V_2|$ represent the number of vertices in V_1 and V_2 respectively, and $|V_u|$ is the number of undesirable vertices.

We have tested the three implemented algorithms on the test problems. For each of the test problems, we use the hill-climbing algorithm, the UMDA, and the hybrid EDA to optimize it, and we record the solutions. For the UMDA and the hybrid EDA, we run them for 60 seconds for each of the test problems. The population sizes used by the UMDA and EDA are both 100. The hill-climbing algorithm outputs its solution when the search converges. The test results are shown in Table 2. In the table, the number of undesirable vertices $|V_u|$ in the test problem is displayed in the second column. The number of undesirable vertices after optimization, together with the reduction of undesirable vertices in percentage, for each of the algorithms, are also presented. All the experiments are performed in a computer which has an Intel Pentium M 1000MHz CPU with 768MB of RAM.

It can be seen from the test results that the hybrid EDA always generates better results than the UMDA and the hill-climbing algorithm except for the second test problem which the hybrid EDA produces a result that is as good as that of the hill-climbing algorithm but is better than that of the UMDA. The statistic data show that the hill-climbing algorithm, the UMDA and the hybrid EDA can reduce the number of undesirable vertices by 68.36%, 64.18% and 75.15%, respectively. In other word, overall the hybrid EDA outperforms the hill-climbing algorithm and the

Table 1. The characteristics of the test problems

<i>Test Problems</i>	$ V_1 $	$ V_2 $	$ V_u $
1	100	50	35
2	100	55	36
3	100	60	32
4	100	65	34
5	100	70	45
6	100	75	49
7	100	80	52
8	100	85	60
9	100	90	55
10	100	95	61
11	100	100	67
12	100	105	73
13	100	110	61
14	100	115	76
15	100	120	73
16	100	125	74
17	100	130	78
18	100	135	77
19	100	140	96
20	100	145	95

Table 2. Comparisons of the hill-climbing algorithm, the UMDA and the hybrid EDA

<i>Test Problems</i>	<i>Before optimization</i> $ V_u $	<i>Hill-climbing</i>		<i>UMDA</i>		<i>Hybrid EDA</i>	
		$ V_u $	%	$ V_u $	%	$ V_u $	%
1	35	5	85.71	6	82.86	3	91.43
2	36	1	97.22	4	88.89	1	97.22
3	32	7	78.13	11	65.63	5	84.38
4	34	9	73.53	9	73.53	5	85.29
5	45	12	73.33	11	75.56	7	84.44
6	49	11	77.55	1	77.55	8	83.67
7	52	20	61.54	17	67.31	11	78.85
8	60	17	71.67	22	63.33	15	75.00
9	55	18	67.27	19	65.45	14	74.55
10	61	16	73.77	17	72.13	15	75.41
11	67	23	65.67	26	61.19	18	73.13
12	73	25	65.75	27	63.01	22	69.86
13	61	20	67.21	29	52.46	16	73.77
14	76	28	63.16	31	59.21	24	68.42
15	73	30	58.90	32	56.16	25	65.75
16	74	32	56.76	36	51.35	26	64.86
17	78	32	58.97	31	60.26	25	67.95
18	77	38	50.65	43	44.16	29	62.34
19	96	33	65.63	44	54.17	32	66.67
20	95	43	54.74	48	49.47	38	60.00

UMDA by 6.79% and 10.97%, respectively, for the test problems.

8 Conclusion and Discussion

In this paper we have presented a new approach to the MSG problem using hybrid EDAs. This approach uses a UMDA to sample start search points and employs a hill-climbing algorithm to find a local optimum in the basins where the start search points are located. By making use of the efficient exploration of the UMDA and the effective exploitation of the hill-climbing algorithm, this hybrid EDA can find an optimal or near-optimal solution efficiently and effectively.

In order to test the performance of the hybrid EDA, we have implemented the hybrid EDA, the UMDA and the hill-climbing algorithm, and have done empirical comparisons between the three algorithms for 20 randomly generated SMG problems of the same size, but of different levels of mutual constraints. Experimental results have shown that the performance of the hybrid EDA is significantly better than both of the UMDA and the hill-climbing algorithm.

The performance of the hybrid EDA could be further improved by two means. Firstly, the joint probability distribution could be factorized by a more sophisticated model reflecting the linkage among those variables [5]. For example, the Factorized Distribution Algorithm (FDA) [6] might be applied on the MSG problem and might produce better results. Secondly, incorporation of domain-specific knowledge into the hybrid EDA could also contribute to improvement of the performance of the hybrid EDA.

References

- [1] M. Tang, K. Eshraghian, and H.N. Cheung, "An efficient approach to constrained via minimization for two-layer VLSI routing," *Proc. IEEE Asia and South Pacific Design Automation Conference*, IEEE Press, Hong Kong, 1999, pp.149-152.
- [2] N.J. Naclerio, S. Masuda, and K. Nakajima, "The via minimization problem is NP-complete," *IEEE Transactions on Computers*, vol.38, no.11, 1989, pp.1604-1608.
- [3] H. Mühlenbein and G. Paaß, "From recombination of genes to the estimation of distributions I. binary parameters," In: A. Eiben, T. Bäck, M. Shoenauer, and H.-P. Schwefel (eds), *Parallel Problem Solving from Nature - PPSN IV*, Springer-Verlag, Berlin, 1996, pp.178-187.
- [4] H. Mühlenbein, "The equation for response to selection and its use for prediction," *Evolutionary Computation*, vol.5, no.3, 1997, pp.303-346.
- [5] M. Pelikan, D.E. Goldberg, and K. Cantu-Paz, "Linkage Problem, distribution estimation, and Bayesian networks," *Evolutionary Computation*, vol.8, no.3, 2000, pp.311-340.
- [6] H. Mühlenbein and T. Mahnig, "The factorized distribution algorithm for additively decomposed functions," *Proc. Congress on Evolutionary Computation*, IEEE Press, Piscataway, 1999, pp.752-759.
- [7] H. Mühlenbein, T. Mahnig, and A. Ochoa, "Schemata, distributions and graphical models in evolutionary computation," *Journal of Heuristics*, vol.5, no.2, pp.215-247.
- [8] Q. Zhang and H. Mühlenbein, "On the convergence of a class of estimation of distribution algorithms," *IEEE Transactions on Evolutionary Computation*, vol.8, no.2, pp.127-136.
- [9] K. Sastry, H.A. Abbass, D.E. Goldberg, and D.D. Johnson, "Estimation of distribution algorithms: sub-Structural niching in estimation of distribution algorithms," *Proc. Conference on Genetic and Evolutionary Computation*, ACM Press, Washington, 2005, pp.671-678.
- [10] M. Tang, "An adaptive genetic algorithm for the minimal switching graph problem," *Lecture Notes in Computer Science*, vol.3448, pp.224-233.