

## 최적 배치를 위한 유전자 알고리즘의 설계와 구현

송 호 정\* 이 범 근\*\*

### Design and Implementation of a Genetic Algorithm for Optimal Placement

Ho-jeong Song\* Beom-geun Lee\*\*

#### 요 약

배치(Placement)는 VLSI 회로의 physical design에서 중요한 단계로서 회로의 성능을 최대로 하기 위하여 회로 모듈의 집합을 배치시키는 문제이며, 배치 문제에서 최적의 해를 얻기 위해 클러스터 성장(cluster growth), 시뮬레이티드 어닐링(simulated annealing; SA), ILP(integer linear programming)등의 방식이 이용된다. 본 논문에서는 배치 문제에 대하여 유전자 알고리즘(genetic algorithm; GA)을 이용한 해 공간 탐색(solution space search) 방식을 제안하였으며, 제안한 방식을 시뮬레이티드 어닐링 방식과 비교, 분석하였다.

#### Abstract

Placement is an important step in the physical design of VLSI circuits. It is the problem of placing a set of circuit modules on a chip to optimize the circuit performance. The most popular algorithms for placement include the cluster growth, simulated annealing and integer linear programming. In this paper we propose a genetic algorithm searching solution space for the placement problem, and then compare it with simulated annealing by analyzing the results of each implementation.

---

\* 충북대학교 컴퓨터공학과 박사과정  
\*\* 경희대학교 전자공학과 박사과정

논문접수 : 2002. 6. 19  
심사완료 : 2002. 9. 17

## I. 서론

배치는 VLSI 회로의 physical design에서 중요한 단계로서 회로의 성능을 최적화하기 위해 주어진 제약을 고려하여 칩상에 회로 모듈들을 배치시키는 문제이다(1,2,3).

배치 문제에서 최적의 해를 얻기 위하여 클러스터 성장, 시뮬레이티드 어닐링, ILP등의 알고리즘들이 기존에 연구되었다.

시뮬레이티드 어닐링(SA)(4,5,6)은 조합 최적화 문제 해결에 적용되는 대표적 반복 휴리스틱 알고리즘으로 TimberWolf(9)등 실용 패키지뿐 아니라 알고리즘의 성능을 비교하는 벤치마크로 주로 사용된다.

유전자 알고리즘(genetic algorithm: GA)은 자연계의 진화를 기반으로 한 탐색 방법을 사용하여 시뮬레이티드 어닐링에 비해 효과적으로 최적해에 근접할 수 있다.

본 논문에서는 VLSI의 physical design 과정 중 회로 모듈의 집합을 상호연결의 길이와 칩 면적을 고려하여 배치시키는 배치 문제에 대하여 유전자 알고리즘(6)을 이용한 해공간 탐색 방식을 제안하였으며, 이 방식을 시뮬레이티드 어닐링 방식과 비교하여 분석하였다.

본 논문의 구성은 다음과 같다. II장에서 배치 문제에 대하여 알아보고, III장에서는 본 논문에서 제안한 최적 배치를 위한 유전자 알고리즘의 데이터 표현 방법과 알고리즘에 대하여 설명한다. IV장에서는 제안한 최적 배치를 위한 유전자 알고리즘과 시뮬레이티드 어닐링 방식을 비교하였고, 마지막으로 V장에서는 결론과 향후 연구 방향에 대하여 기술한다.

## II. 문제

배치는 칩 면적과 상호연결의 길이를 고려하여 n개의

모듈을 정해진 영역에 겹치지 않게 시키는 문제이다. 즉, 최적의 배치는 배치된 모듈들 사이의 상호연결의 길이와 칩 면적이 최소가 되도록 각각의 모듈들을 배치시키는 것으로서, 모듈들을 어떻게 배치하느냐에 따라 칩 면적과 모듈들 사이의 상호연결의 길이가 바뀔 수 있다(7).

그림 1은 배치의 간단한 예로서 (a),(b),(c)는 같은 칩 면적 안에 배치 되었지만, (d),(e)는 다른 결과를 얻는 것을 볼 수 있다.

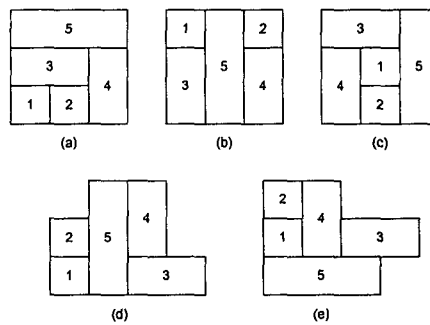


그림 2. 배치 예  
Fig. 1. Example of placement.

배치 문제를 해결하기 위하여, 기존에 연구된 각 알고리즘을 살펴보면, 클러스터 성장은 계산량이 적어 속도가 빠른 반면 지역해에 빠지기 쉬운 단점이 있으며, 시뮬레이티드 어닐링은 최적해에 가까운 해를 찾을 수 있으나, 온도에 따라 지역해를 벗어날 수 있는 가능성을 부여하므로 저온에서 해탐색 범위가 급격히 축소되는 경향을 보인다. 또한 ILP 알고리즘은 최적해를 반드시 찾을 수 있는 반면에 계산량이 많아 아주 작은 문제에만 적용 가능하고 큰 문제에서는 적용이 불가능한 단점이 있다.

## III. 최적 배치 유전자 알고리즘

유전자 알고리즘은 진화 과정에서 유도된 탐색방법으로, 염색체와 유사한 자료구조를 사용하여 해공간을 부호화하며, 부호화한 자료 구조에 제조할 연산자를 적용하여 염색체들을 진화시킨다.

유전자 알고리즘은 처음에 임의로 선택된 염색체 집단

(population of chromosome)에서 시작하며, 이러한 염색체 집단 중에서 일정한 방식으로 부모 염색체를 선택하고 이들 부모 염색체를 교배시켜 자식 염색체를 생성한다. 새로 생성된 자식 염색체는 평가함수에 의해 평가되며 좋은 평가 결과를 가지는 염색체가 다음 세대에 살아남을 확률이 높게 된다. 이와 같은 방식으로 유전자 알고리즘은 염색체 집단의 진화를 통하여 최적해에 근접할 수 있으므로, 최적 해를 구하기 어려운 여러 NP-문제에 적용될 수 있다.

1. 염색체의 표현

본 논문에서 제안한 최적 배치를 위한 유전자 알고리즘은 문제의 복잡도를 줄이기 위해 분할가능구조(slicing structure(8)), 모든 모듈의 크기 고정, 그리고 90° 회전 가능한 경우만을 가정하였다.

분할가능구조란 전체 사각형 구조의 모듈을 가로나 세로의 연속 분할만으로 부 모듈로 나눌 수 있는 구조를 말한다(그림 2).

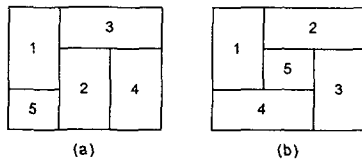


그림 2. (a) 분할가능 배치 (b) 분할불가능 배치  
Fig. 2. (a) Slicing placement. (b) Non-slicing placement.

분할가능 배치는 그림 3과 같이 분할가능트리(slicing tree)로 표현할 수 있고, 분할가능트리는  $E=e_1e_2\cdots e_{2n-1}$ . ( $e_i \in \{1, 2, \dots, n, H, V\}$ ,  $1 \leq i \leq 2n-1$ )의 식으로 표현할 수 있다. 여기서 연산자 H, V는 다음과 같은 의미를 갖는다.

- ijH : 모듈 j가 모듈 i의 위에 위치한다.
- ijV : 모듈 i가 모듈 j의 왼쪽에 위치한다.

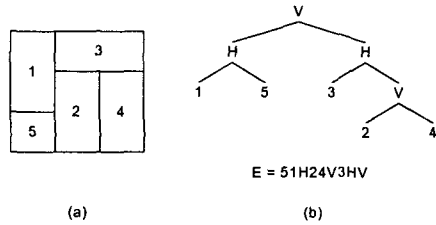


그림 3. (a) 분할가능 배치 (b) 분할가능트리와 분할가능트리의 표현  
Fig. 3. (a) Slicing placement. (b) Slicing tree and slicing tree representation.

하나의 배치는 그림 4와 같이 여러개의 분할가능트리로 표현할 수 있으므로 중복되는 해를 없애기 위하여 정규화된 분할가능트리의 표현을 사용하였다.

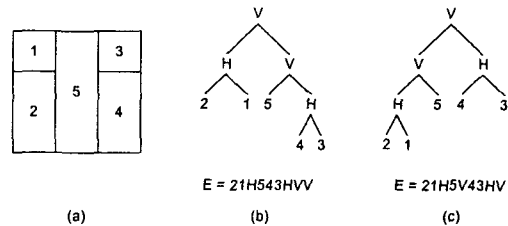


그림 4. 여러 가지 분할가능트리의 표현 (a) 배치 (b) 비정규화된 표현 (c) 정규화된 표현

Figure 4. several slicing tree representation. (a) Placement. (b) Not normalized expression. (c) Normalized expression.

정규화된 표현이란 같은 연산자가 연속적으로 나오지 않는 식을 말하며, 정규화된 표현을 사용하여 하나의 배치는 하나의 표현만을 가질 수 있도록 하였다.

그러므로 정규화된 표현식을 그림 5와 같이 유전자 알고리즘의 염색체로 사용하였다.

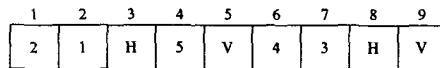


그림 5. 염색체의 표현  
Fig. 5. Representation of chromosome.

2. 교배 연산자(Crossover Operator)

유전자 알고리즘에서 각 세대의 모집단은 교배를 통해 각 부모의 유전 정보를 상속받게 된다. 이러한 교배로 생성되는 자손들 중 좋은 형질을 상속받은 자손은 다음 세

대에 살아남을 높은 확률을 가지게 되고, 그렇지 않은 자손은 낮은 확률을 가지게 된다.

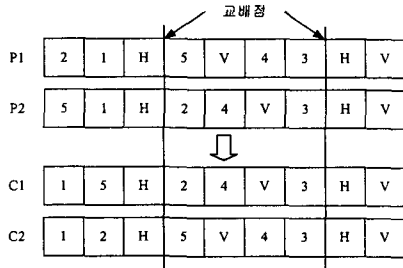


그림 7. 2-점 교배 연산  
Fig. 6. 2-point crossover operation.

본 논문에서 사용한 교배 연산자는 PMX(Partially Mapped Crossover)[6,7,8]의 한 형태로 볼 수 있으며, 구체적인 연산 방식은 다음과 같다.(그림 6)

- 1단계 : 부모 염색체 P1, P2에서 각 그룹에 속하는 교배점(cross point)을 하나씩 랜덤하게 선택한다.
- 2단계 : 부모 염색체 P1에서 두 개의 교배점에 의해 만들어진 부염색체(sub-chromosome)를 자식 염색체 C2의 같은 위치에 복사한다. 자식 염색체 C1도 같은 방식으로 복사한다.
- 3단계 : 부모 염색체 P2에서 자식 염색체 C2에 있는 유전인자를 삭제하고 P2의 나머지 유전인자를 C2의 빈곳에 순서대로 채운다. 자식 염색체 C1도 같은 방식으로 빈곳을 채운다. 이때, 연산자 H, V는 그대로 복사한다.

### 3. 돌연변이(Mutation)

유전자 알고리즘에서 각 세대의 모집단은 진화를 진행하면서 얻고자 하는 해에 가까운 염색체들로 구성된 모집단으로 수렴하게 되지만, 그 결과가 최적해가 아닌 지역해로 수렴할 수도 있으며, 이러한 지역해로의 수렴을 막기 위하여 돌연변이 연산을 수행하게 된다.

본 논문에서 제안된 돌연변이 연산자는 다음과 같이 4가지의 연산자를 정의하였다. 여기서 모듈 1, ..., n을 피연산자, H, V를 연산자라고 가정한다.

M1 : 두 개의 이웃된 피연산자를 교환한다.

M2 : 임의의 개수의 연속된 연산자를 반전시킨다. 즉,  $\overline{V} = H, \overline{H} = V$ 이다.

M3 : 두 이웃된 피연산자와 연산자를 교환한다.

M4 : 피연산자를 90° 회전시킨다.

즉, 12V4H35VH에서 5는 5번 피연산자를 90° 회전시킨 것을 의미한다.

여기서 돌연변이 연산자의 경우, M1, M2, M4 이동은 항상 정규화된 표현을 얻는 반면, M3 이동은 비정규화된 표현을 얻을 수 있으며, 교배 연산자의 경우도 비정규화된 표현을 얻을 수 있으므로 다음의 경우에 해당하는 결과를 나타내는 교배나 돌연변이 연산 결과는 취소되어야 한다.

- (1) 같은 연산자의 연속 표현 : 12V43H5HV에서 e7과 e8을 교환하였을 경우 12V43HH5V의 비정규화된 결과를 얻게 된다.
- (2) Balloting property[8]의 위배 : 피연산자  $e_i$ 와 연산자  $e_{i+1}$ 을 M3 이동하였을 때  $2N_i + 1 \geq i$ 이면 피연산자의 수보다 연산자의 수가 많아져 비정규화된 표현이 된다(여기서  $N_k$ 는 정규화된 표현  $E = e_1e_2 \dots e_k (1 \leq k \leq 2n-1)$ 에서의 연산자의 수로 정의한다).  
즉, 12V4H35VH에서 e7와 e7+1를 M3 이동하면  $2N_7 + 1 (=6) < 7$ 이지만, e4와 e4+1를 M3 이동하면  $2N_4 + 1 (=4) = 4$ 이므로 balloting property를 위배하게 된다.

### 4. 최적 배치를 위한 유전자 알고리즘

알고리즘 A는 본 논문에서 제안한 최적 배치를 위한 유전자 알고리즘을 나타낸다.

알고리즘 A. 최적 배치를 위한 유전자 알고리즘  
Algorithm A. Genetic algorithm for optimal placement

단계 0 : 파라미터들의 설정

객체들의 수를 나타내는 pop\_size, 돌연변이율을 나타내는 Pm, 그리고 최대 생성 횟수를 나타내는 max\_gen을 설정한다.

단계 1 : 초기 모집단의 생성

pop\_size 만큼의 각기 다른 객체  $S_i(i=1, \dots, pop\_size)$ 를 임의로 생성한다.

단계 2 : 교배

모집단으로부터 두 개의 부모 염색체 P1, P2를 임의로 선택하고, 2-점 교배 연산을 수행하여 새로운 자식 염색체 C1, C2를 생성한다.

단계 3 : 돌연변이

3.1 랜덤하게  $\epsilon \in (0,1)$ 을 생성하고 돌연변이율 Pm 과 비교한다. 만일  $\epsilon < Pm$ 이면 3.2를 수행한다.

3.2 랜덤하게 M1, M2, M3, M4중의 하나를 선택하여 돌연변이 연산을 수행한다.

단계 4 : 만일 생성된 자식 염색체 C1 또는 C2가 모집단 내의 어느 객체와 같다면, C1 또는 C2가 같은 객체가 없는 새로운 객체가 될 때까지 돌연변이 연산을 수행한다.

단계 5 : 새로운 세대의 구성

모집단 내의 최대 F(G)값을 갖는 두 개의 염색체를 삭제하고 생성된 자식 염색체 C1과 C2를 모집단 내에 추가한다.

단계 6 : 새로운 염색체의 생성 횟수가 max\_gen이 될 때까지 단계 1부터 5까지를 반복한다. 최적해는 모집단 내에서 가장 작은 Cost를 갖는 염색체이다.

### 5. 비용함수

배치의 목적은 최소의 칩 면적에 최단의 상호연결의 길이를 갖는 해를 찾는 것이므로, 비용함수는 전체 모듈이 차지하는 영역과 각 모듈들의 상호연결의 길이의 합으로 나타낸다. 모듈이 차지하는 영역은 가로, 세로의 비를 고려하여 차이가 너무 커지지 않도록 할 수 있다(식 1).

$$Cost(F) = \alpha A(y/x) + \lambda W \text{ (식 1)}$$

여기서 가로, 세로의 길이 중 짧은 부분을 x로 긴 부분을 y로 가정하였으며,  $\alpha$  와  $\lambda$ 는 각각 칩 면적과 상호연결의 길이의 중요도를 나타내는 파라미터로서,  $\alpha$  와  $\lambda$  값을 적절히 조절함으로써 다른 해를 얻을 수 있다.

## IV. 시뮬레이션

본 논문에서 제안한 최적 배치를 위한 유전자 알고리즘을 시뮬레이션 하기 위하여 일정한 모듈의 개수를 갖는 블록 데이터를 생성하고, 동일 블록 데이터에 시뮬레이티드 어닐링과 유전자 알고리즘을 적용하여 그 시뮬레이션 결과를 분석하였다. 또한 배치 문제에 사용되는 벤치마크 회로인 MCNC 회로에 대하여 동일 알고리즘을 적용하여 결과를 비교하였다.

본 시뮬레이션에서는 유전자 알고리즘의 돌연변이율 Pm=0.1, 모집단내의 개체의 수 pop\_size=20, 최대 수행 횟수 max\_gen=10000을 사용하였고, 시뮬레이티드 어닐링의 한 온도에서의 알고리즘수행 횟수 M=100, 최대 수행 횟수 max\_time=10000, 초기 온도 T=10, 냉각 파라미터  $\alpha=0.9$ , 그리고  $\beta=1.0$ 의 값을 사용하였다. 또한 비용 계산의 중요한 파라미터인  $\alpha=1, \lambda=0$ 을 사용하여 칩 면적을 최소화하는 해를 구하도록 하였다.

표 1은 임의로 생성된 블록 데이터에 시뮬레이티드 어닐링과 유전자 알고리즘 적용한 결과의 최적값, 최악값, 평균값을 가지고 각 알고리즘을 비교한 것을 보이고 있다.

표 1. GA와 SA의 테스트 결과  
Table 1. Test result for GA, SA.

문제 종류	모듈 개수	시뮬레이티드 어닐링			유전자 알고리즘		
		최적	최악	평균	최적	최악	평균
1	10	96	224	151	89	99	89.35
2	12	140	315	223.16	130	142	135.3
3	14	256	580	394.57	205	241	227.56
4	16	280	648	452.79	256	296	276.12
5	18	324	816	578.17	292	359	321.26
6	20	350	986	689.6	341	413	371.49

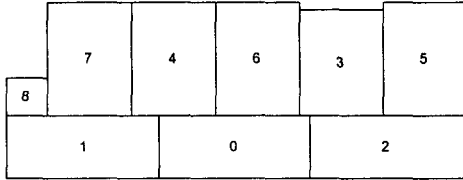
또한, 표 2는 배치 문제에 사용되는 벤치마크 회로인 MCNC 회로에 대하여 동일 알고리즘을 적용한 결과를 보이고 있다.

표 2. SA와 GA의 벤치마크 테스트 결과  
Table 2. Benchmark test result for SA, GA

벤치마크	모듈 개수	면적	
		시뮬레이티드 어닐링	유전자 알고리즘
apte	9	47528748	47301557
xerox	10	24911208	21181518
hp	11	14868168	10428453
ami33	33	2379636	2354727
ami49	49	91773864	87411393

그림 8은 MCNC 벤치마크 회로중 "apte"의 최적 배

치를 위한 유전자 알고리즘 적용 후의 배치 결과이다.



$$E = 10 V2 \sqrt{87} \sqrt{46} \sqrt{3} \sqrt{5} VVH \quad \text{Cost} = 47313280$$

그림 8 MCNC "apte" 회로의 배치 결과  
Fig. 7. Placement result for MCNC "apte"

두 알고리즘의 시뮬레이션 결과를 분석해보면 최적 배치를 위한 유전자 알고리즘이 시뮬레이티드 어닐링보다 더 좋은 결과를 얻는 것을 알 수 있다.

시뮬레이티드 어닐링은 초기에는 지역해에서 벗어날 수 있는 확률이 높으나, 냉각과정이 진행되면서 지역해에서 벗어날 수 있는 확률이 낮아져서 일정 횟수를 수행한 후에는 더 좋은 해를 찾을 가능성이 작아지게 된다. 반면에 최적 배치를 위한 유전자 알고리즘은 교배와 돌연변이 연산을 사용한 이웃해로의 이동범위 광역화에 의한 효과적인 해공간 탐색으로 시뮬레이티드 어닐링에 비해 더 좋은 최적해를 찾게 된다.

## V. 결론

본 논문에서는 VLSI의 physical design 과정중 배치 문제에 대하여 최적 배치를 위한 유전자 알고리즘을 제안 하였으며, 제안한 방식을 시뮬레이티드 어닐링 방식과 비교, 분석하였다.

제안한 최적 배치를 위한 유전자 알고리즘과 시뮬레이티드 어닐링 방식을 자동 생성한 회로와 벤치마크 회로에 적용하여 비교, 분석한 결과 제안한 최적 배치를 위한 유전자 알고리즘이 시뮬레이티드 어닐링 방식보다 더 효과적으로 최적해에 근접하는 것을 알 수 있었다.

앞으로 각 모듈간의 상호 연결을 고려한 배치와 각 모듈의 모양이 사각형이 아닌 다각형인 경우에 대한 유전자 알고리즘의 연구가 필요하다고 생각된다.

## 참고문헌

- [1] S. M. Sait, H. Youssef, VLSI Physical Design Automation Theory and Practice, World Scientific Publishing, 2001.
- [2] Naveed A. Sherwani, Algorithms for VLSI Physical Design Automation. 3rd Edition, Kluwer Academic Publishers, 2001.
- [3] F.Y. Young, D.F. Wong, Hannah H. Yang, ON Extending Slicing Floorplan to Handle L/T-Shaped Modules and Abutment constraints, IEEE Transaction on Computer-Aided Design, pp. 800-807. June 2001.
- [4] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi. Optimization by Simulated Annealing, Science, vol. 220, no. 4598, pp.671-680, 1983.
- [5] S. M. Sait, H. Youssef, Iterative Computer Algorithms with Applications in Engineering, Computer Society, 1999.
- [6] S. M. Sait, H. Youssef, Iterative Computer Algorithms with Applications in Engineering, Computer Society, 1999.
- [7] K.Shahookar and P.Mazumder. VLSI cell placement techniques. ACM Computing Surveys, 23(2):143-220, June 1991.
- [8] D.F.Wong and C.L.Liu. A new algorithm for floorplanning design. Proc. of the 23rd DAC, pages 101-107, 1986.
- [9] C.Sechen and A.L.Sangiovanni-Vincentelli. Timberwolf3.2: A new standard cell placement and global routing package. Proceedings of 23rd Design Automation Conference, pp. 432-439, 1986.

저 자 소개



송 호 정

1994년 배재대학교 물리학과  
졸업(학사)  
1996년 청주대학교 전자공학과  
졸업(공학석사)  
2001년 충북대학교 컴퓨터공학과  
박사수료  
2000년3월~현재 극동정보대학  
전자통신과 겸임강사  
관심분야 : VLSI 설계,  
High-level Synthesis,  
Genetic Algorithm



이 범 근

1973년 2월 14일생  
1995년 2월 청주대학교 전자공  
학과 졸업(공학사)  
1997년 2월 청주대학교 전자공  
학과 졸업(공학석사),  
2001년 8월 경희대학교 전자공  
학과 박사수료  
2001년 3월~2002년2월 극동  
정보대학 전자통신과  
겸임강사  
2002년3월~현재 극동정보대학  
전산정보처리과 초빙전  
임강사  
관심분야 : Micro Display  
HDTV, CAD,  
Internet Applications