

論文2002-39SD-6-5

새로운 부가 힘을 사용한 Quadratic Programming 기반의 표준셀 배치

(Quadratic Programming Based Standard-cell Placement with New Additional Force)

姜相求*, 林鐘錫*

(Sang Ku Kang and Chong Suck Rim)

요 약

본 논문은 quadratic programming(QP) 기반의 표준셀 배치에 대하여 다룬다. 본 논문은 QP 기반의 배치에서 발생하는 셀겹침을 제거하고 균등한 배치를 얻기 위하여 새로운 모델의 부가 힘을 제안한다. 부가 힘(additional force)이란, 셀 사이의 연결과는 관계없이 배치영역 내의 셀의 분포 밀도에 의해 받게되는 힘을 의미한다. 본 논문에서는 부가 힘을 가상 고정셀(dummy fixed cell)에 의해 발생하는 힘으로 모델화하여 그것을 이용한 개략배치 방법을 제안한다. 제안한 배치방법에 의한 최종 배치결과를 TimberWolf v7.0과 Itools v1.4와 비교하였다. 제안된 배치는 시간지연을 고려하지 않은 경우, 배선거리에서 TimberWolf v7.0에 비하여 평균 7.5% 향상된 결과를 얻었다. 시간지연을 고려했을 경우, 배선거리에서 TimberWolf v7.0에 비하여 평균 5.0% 향상된 결과를 얻었다. 그리고 Itools v1.4에는 비교할만한 결과를 얻었다.

Abstract

This paper deals with a standard cell placement which is based on a quadratic programming. This paper proposes a new additional force to reduce the cell overlap and to get a uniform distribution of cells. The additional force is not concerned with interconnections between cells, but it is determined by the density of a placement area. In this paper, we modelled that the new additional force is a force which is caused by the dummy fixed cell. And it is used for the global placement. Proposed placement method is compared with TimberWolf v7.0 and Itools v1.4. Proposed placer achieved 7.5% average reduction in wirelength in non timing driven mode, 5.0% average reduction in wirelength in timing driven mode compared to TimberWolf v7.0. And we got a comparable result to Itools v1.4.

I. 서론

표준셀 배치(standard-cell placement)는 여러 가지 목표들을 만족시키는 셀들의 최적위치를 찾는 것이다.

* 正會員, 西江大學校 컴퓨터學科

(Dept. of Computer Science and Engineering, Sogang University)

接受日字:2001年9月17日, 수정완료일:2002年4月9日

배치의 목표로는 면적의 최소화(배선거리의 최소화), 배선밀도의 최소화, 시간지연 최소화, 전력소모의 최소화 등을 들 수 있다. 위의 목표들을 동시에 모두 만족하게 하는 것이 매우 어려우므로, 지금까지 대부분의 배치방법들은 이들 중 한두 가지를 만족하는 수준으로 배치가 이루어진다. 반도체 제조기술의 발전으로 deep-submicron 공정이 가능하게 되면서 회로의 지연시간이 게이트(gate)에 의한 지연시간보다 배선(wire)에 의한 지연시간이 더 커지게 된다.^[20] 따라서 배치단계에서 배선거리와 시간지연을 함께 고려하는 것이 현재의 추세

이다.

대표적인 배치방법으로는 simulated annealing(SA)을 이용한 방법,^[19] min-cut을 이용한 분할기반 배치방법,^[7,10,11,18] quadratic programming(QP)을 이용한 방법^[4,6,12,15,17] 등이 있다. SA을 이용한 배치방법은 최적에 가까운 배치결과를 얻을 수 있지만, 많은 계산시간이 필요하다. Min-cut을 이용한 분할 배치방법은 빠른 시간 안에 배치결과를 얻을 수 있지만, 배치결과가 초기 배치에 의해 큰 영향을 받으므로 초기배치의 질이 나쁘면 배치결과도 좋지 않게 된다. 이에 반해 QP을 이용한 배치는 비교적 빠른 계산시간과 SA을 이용한 방법의 배치결과에 근접한 결과를 얻을 수 있다고 알려져 있다.

QP을 이용한 배치방법들을 살펴보면, GORDIAN,^[12] GORDIANL,^[17] SPEED^[15] 등과 같이 quadratic optimization과 분할을 번갈아 반복하는 방법(iterative method)과 GENERIC^[4]과 같이 부가 힘(additional force)을 사용하는 방법 등이 있다.

GORDIAN^[12]은 quadratic optimization과 분할을 주어진 조건까지 반복해서 배치를 수행한다. 분할단계에서 얻어진 분할정보는 다음 quadratic optimization 단계에서 무게중심 제약조건이 되어 셀을 균등하게 배치 영역에 분포시키기 위해 사용된다. GORDIANL^[17]은 목적함수를 선형화한 것을 제외하면 GORDIAN과 기본적으로 동일하다. SPEED^[15]에서는 net-based weighting 방법을 사용하여 시간지연을 고려한 배치방법을 제시한다. 즉, SPEED는 GORDIANL에 타이밍 분석을 통한 net-weight update를 추가한 것이다. GENERIC^[4]는 잘 알려진 force-directed 방법에 셀이 배치영역 내에 균등하게 분포하도록 하는 부가 힘을 덧붙여 사용한다. 부가 힘(additional force)이란, 셀 사이의 연결과는 관계없이 배치영역 내의 셀의 분포에 의해 결정되는 밀도에 의해 받게 되는 힘을 의미한다.

QP을 이용한 배치방법의 문제점 중 하나는 순수하게 QP만을 가지고 셀겹침이 없는 배치결과를 얻을 수 없다는 점이다. 따라서 셀겹침을 완전히 제거하는 과정이 필요하고 이를 상세배치(detailed placement)라 한다. GORDIAN, GORDIANL, SPEED, GENERIC 등에서는 DOMINO^[2]라는 상세배치를 사용하고, Attractor-Repeller^[6]에서는 Tabu-search 기반의 부분개선 알고리즘^[1]을 사용한다.

GENERIC^[4]에서 사용되는 부가 힘의 문제점은 적절

한 크기의 부가 힘을 찾기 어렵다는데 있다. 지나치게 큰 부가 힘을 적용하여 quadratic optimization을 수행하면 배치결과가 나빠지게 되고, 반면에 작은 부가 힘을 적용하여 quadratic optimization을 수행하면 배치결과는 좋아지지만 반복 횟수가 증가하여 실행시간이 증가하게 된다. 본 논문에서는 이 문제를 해결하기 위하여, 부가 힘을 가상 고정셀(dummy fixed cell)에 의해 발생하는 힘으로 모델화해서 개략배치를 수행한다.

본 논문에서는 셀겹침을 완전히 제거하고 부분개선(local improvement)을 위해 상세배치 단계를 수행하는데, 이것은 사분할 기반의 부분개선 과정과 셀 sliding 기법을 이용한 셀겹침 제거 과정으로 구성된다. 사분할 기반의 부분개선은 부분개선 과정에 사용된 방법은 Huang 등이 제안한 사분할 방법^[10]에 기초를 둔 것으로 MST(minimum spanning tree)를 비용함수로 사용한다. 셀겹침 제거 과정에는 완전히 셀겹침을 제거하기 위해 dynamic programming 기법을 사용한다. 본 논문에서 제안한 배치기의 배치 과정은 그림 1과 같다.

본 논문은 서론을 포함하여 4장으로 구성된다. 2장에서는 개략배치, 상세배치, 시간지연을 고려한 배치 등에 대해 기술한다. 3장에서는 실험결과를 보이고 4장에서는 결론과 향후 연구방향을 제시한다.

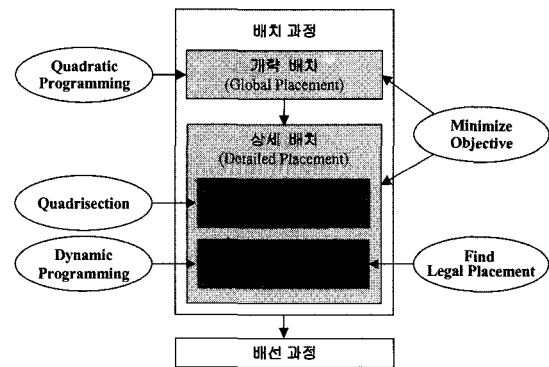


그림 1. 배치 과정
Fig. 1. Placement Process.

II. 배치 과정

이 장에서는 본 논문에서 구현된 배치 과정에 대해 기술한다. 먼저 1절에서는 본 논문에서 제안한 새로운 부가 힘을 이용한 개략배치 방법을 설명한다. 2절에서

는 상세 배치에 사용된 방법을 간략히 설명한다. 그리고 마지막으로 3절에서는 배치 과정에서 시간지연을 고려하기 위한 방법을 설명한다.

1. 개략배치 방법

본 논문에서 제안한 개략배치는 GENERIC^[4]에서 제안한 부가 힘(additional force)을 개선한 새로운 모델의 부가 힘을 적용한 QP 방법이 사용된다. 그림 2는 개략 배치를 위한 과정이다. 이것은 셀의 좌표를 계산하는 quadratic optimization 부분(그림 2의 7-9행)과 부가 힘을 계산하는 부분(그림 2의 10행)으로 구성된다. Quadratic optimization 부분은 기본적인 quadratic 모델의 목적함수(그림 2의 7행)에 새로운 부가 힘을 추가하여 만들어진 목적함수(그림 2의 8행)를 최소화하는 셀들의 좌표를 구한다(그림 2의 9행). 부가 힘을 계산하는 부분에서는 현재 셀들의 좌표에 의해 다음 단계의 quadratic optimization에 사용될 부가 힘을 계산한다. 이 과정을 주어진 수준까지 반복 수행하여 개략배치 결과를 얻는다(그림 2의 6행). 따라서 본 절에서는 개략배치에 필요한 이차 목적함수, 기존의 부가 힘, 새로운 부가 힘 등에 대해 설명한다.

```

1. Global_Placement( )
{
2. matrix matC; //system matrix
3. vector vecD; //system vector
4. vector vecE; //additional force vector
5. vector vecP; //placement coordinate vector
6. while( no empty square in a placement area )
{
//make matC, vecD
7. setup_objective();
//apply new model additional force
8. diagonal_update();
//Solve matC * vecP + vecD + vecE = 0
9. compute_placement();
//compute vecE
10. compute_additional_force();
}
}
    
```

그림 2. 개략배치 과정
Fig. 2. Global Placement Procedure.

(1) 이차 목적함수 (Quadratic Objective Function)

개략배치를 위한 quadratic optimization을 수행하기 위해서는 이차 목적 함수(quadratic objective function)가 요구된다. 본 항에서는 이차 목적 함수를 설명한다.

개략배치의 입력으로 주어지는 네트리스트는 다음과 같은 정보를 포함한다. 네트리스트에 속한 모든 셀들의

집합을 M , movable 셀들의 집합을 M_m , 고정셀들(io pads)의 집합을 M_f 라하고, 세 집합사이에는 사이에는 $M_m \cup M_f = M$, $M_m \cap M_f = \emptyset$ 과 같은 관계가 성립된다. 그리고 movable 셀의 수가 n ($\|M_m\| = n$)이라 하자. 셀 사이의 연결인 네트의 집합 N 이라 하면, 네트리스트 T 는 네트 ν 와 셀 μ 의 연결을 $(\nu, \mu) \in T$ 로 하는 이진관계 $T = N \times M$ 으로 표현가능하고, 네트 ν 에 연결된 셀들의 집합 M_ν 를 $M_\nu = \{\mu \in M | (\nu, \mu) \in T\}$ 로 정의된다.

네트 ν 의 중심좌표를 (x_ν, y_ν) , 셀 μ 의 중심좌표를 (x_μ, y_μ) 라 할 때, 네트 ν 의 squared Euclidean distance L_ν 는 다음과 같다.

$$L_\nu = \sum_{\mu \in M_\nu} [(x_\mu - x_\nu)^2 + (y_\mu - y_\nu)^2] \quad (1)$$

Critical 네트를 가까이 모이게 하기 위한 가중치 $w_\nu \geq 1$ 가 각 네트마다 부여되면, 모든 네트들에 대한 weighted squared Euclidean distance의 합이 목적함수 ϕ 가 된다.

$$\phi = \frac{1}{2} \sum_{\nu \in N} L_\nu \cdot w_\nu \quad (2)$$

각 네트의 좌표를 네트에 연결된 셀들의 중심좌표의 평균으로 치환한다. 그러면 목적함수를 네트에 대한 변수없이 셀의 중심좌표로만 표현할 수 있게 된다.

$$\phi(\vec{x}, \vec{y}) = \frac{1}{2} \vec{x}^T C \vec{x} + \vec{a}_x^T \vec{x} + \frac{1}{2} \vec{y}^T C \vec{y} + \vec{a}_y^T \vec{y} + const. \quad (3)$$

\vec{x} , \vec{y} 는 각각 movable 셀의 x 좌표벡터, y 좌표벡터를 의미한다. 행렬 C 는 연결 행렬로서 대각성분은 모든 셀 사이의 연결에 의해 생성되고 그 외의 성분들은 movable 셀 사이의 연결에 의해 생성된다. \vec{a}_x , \vec{a}_y 는 movable 셀과 고정셀 사이의 연결에서 생성된다. 상수(const.)부분은 고정셀 사이의 연결에서 생성된다.

식 (3)을 최소화하기 위해서는 다음과 같은 가정이 필요하다. 첫째, 적어도 하나 이상의 고정셀이 필요하다. 고정셀이 없는 경우에는 \vec{a}_x (\vec{a}_y)이 $\vec{0}$ 이 되어 \vec{x} , \vec{y} 는 모두 $\vec{0}$ 인 trivial solution을 갖게 된다. 둘째, 모든 movable 셀은 어떤 경로를 통해서든 모든 movable 셀이 고정셀과 직접 또는 간접적으로 연결되어 있다. 두 번째 가정은 행렬 C 가 positive definite이기 위한 조건이다. 각각의 셀들은 회로의 외부에서 접근(access)

할 수 있어야 하므로 위의 가정은 모든 유용한 네트리스트에 대해 적용할 수 있다.

식 (3)의 행렬 C 가 positive definite이므로, 식 (3)은 다음과 같은 선형방정식을 풀면 최소화된다.

$$C\vec{x} + \vec{d}_x = \vec{0} \quad (4-1)$$

$$C\vec{y} + \vec{d}_y = \vec{0} \quad (4-2)$$

식 (4)의 해를 구하기 위해서는 행렬 C 가 희소행렬 (sparse matrix)이고 symmetric positive definite인 특징을 고려하여, preconditioned conjugate gradient 방법^[9]을 사용한다. Preconditioning을 위해 incomplete Cholesky decomposition 방법^[9]을 사용한다.

(2) 부가 힘 (Additional Force)

식 (4)는 셀접침이나 배치영역의 밀도를 고려하지 않는다. 따라서 배치결과에 셀접침이 존재하고 셀들이 배치영역에 균등하게 분포되지 않는다. 부가 힘은 셀들이 배치영역에 균등하게 분포되도록 하기 위해 사용된다. 본 논문에서 제안한 새로운 모델의 부가 힘을 설명하기 전에 GENERIC^[4]에서 제안한 부가 힘에 대해 설명한다.

GENERIC에서 제안한 부가 힘은 다음의 네 가지 조건을 만족하여야 한다.

첫째, 부가 힘은 셀의 좌표에 의해서만 결정된다.

둘째, 배치영역에서 셀의 밀도가 높은 곳은 힘의 근원이 되고, 밀도가 낮은 곳은 힘의 소실점이 된다.

셋째, 힘은 원(circle)을 형성하지 않는다.

넷째, 무한에서 힘은 0이 되어야 한다.

이와 같은 조건을 만족하는 부가 힘의 수식을 얻기 위한 과정은 다음과 같다. 첫 번째 조건에 의해 셀 i 에 적용되는 부가 힘 \vec{f}_i 는 셀 i 의 좌표가 (x_i, y_i) 일 때 다음과 같다.

$$\vec{f}_i = \vec{f}(x, y)|_{x=x_i, y=y_i} \quad (5)$$

두 번째 조건에 의해, 주어진 한점에서 밀도를 기술하기 위해, 우선 rectangle 함수 $R(z)$ 를 다음과 같이 정의한다.

$$R(z) = \begin{cases} 1 & \text{if } -\frac{1}{2} < z < \frac{1}{2} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

셀 i 의 폭과 높이를 각각 w_i, h_i 라 하면 셀 i 의 영역

함수(area function) $a_i(x, y)$ 는 다음과 같이 정의된다.

$$a_i(x, y) = R\left(\frac{x-x_i}{w_i}\right) \cdot R\left(\frac{y-y_i}{h_i}\right) \quad (7)$$

$$= \begin{cases} 1 & \text{cell } i \text{가 } (x, y) \text{를 포함할 때} \\ 0 & \text{otherwise} \end{cases}$$

배치영역의 폭과 높이를 각각 W, H 라 하면, 배치영역의 영역함수 $A(x, y)$ 는 다음과 같다.

$$A(x, y) = R\left(\frac{x}{W}\right) \cdot R\left(\frac{y}{H}\right) \quad (8)$$

$$= \begin{cases} 1 & \text{placement area가 } (x, y) \text{를 포함할 때} \\ 0 & \text{otherwise} \end{cases}$$

밀도함수(Density function) $D(x, y)$ 를 다음과 같이 정의한다.

$$D(x, y) = \sum_i a_i(x, y) - s \cdot A(x, y), \quad s = \frac{\sum_i w_i \cdot h_i}{W \cdot H} \quad (9)$$

$\sum_i a_i(x, y)$ 는 셀에 의해 요구되는 면적, $s \cdot A(x, y)$ 는 s 에 의해 스케일된 배치영역으로부터 공급되는 면적을 의미하게 된다. 그림 3에서와 같이 $D(x, y) > 0$ 인 곳은 밀도가 높음을 의미하고, 그것은 셀들이 그 점에 몰려 있음을 뜻한다. 반대로 $D(x, y) < 0$ 인 곳은 밀도가 낮음을 의미하고, 그 점에 셀들이 거의 없음을 뜻한다. 따라서 두 번째 조건은 비례 상수 k 를 사용하여 다음과 같이 쓸 수 있다.

$$\vec{\nabla} \vec{f}(x, y) = k \cdot D(x, y) \quad (10)$$

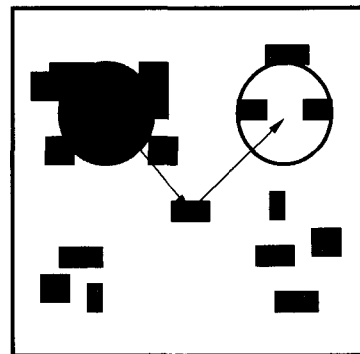


그림 3. 밀도 함수와 힘의 방향
Fig. 3. Density Function and Direction of Additional Force.

세 번째 조건은 $\vec{f}(x, y)$ 가 conservative함을 의미한다. 이것은 스칼라 함수 $\phi(x, y)$ 가 존재하여 다음을 만족함을 의미한다.

$$\vec{\nabla} \phi(x, y) = \vec{f}(x, y) \tag{11}$$

식 (5)와 식 (6)을 묶고 네 번째 조건에 의한 경계조건을 고려하면, 다음과 같은 Poisson 방정식을 얻게된다.

$$\Delta \phi(x, y) = k \cdot D(x, y) \tag{12}$$

$$\lim_{\vec{r} \rightarrow \infty} \|\vec{\nabla} \phi(x, y)\| = 0 \quad (\vec{r} = (x, y)^T) \tag{13}$$

이 standard problem은 $\vec{f}(x, y)$ 에 대해 다음과 같은 유일한 해를 갖는다.

$$\vec{f}(x, y) = \frac{k}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} D(x', y') \frac{\vec{r} - \vec{r}'}{\|\vec{r} - \vec{r}'\|^2} dx' dy' \tag{14}$$

그림 4과 같이 셀을 점로 모델화하고 배치영역을 단위 영역으로 작게 나누었을 경우, 식 (14)의 적분식을 다음과 같은 합(discrete sum)의 형태로 쓸 수 있게 된다.

$$\begin{aligned} \vec{f}(x, y) &= \frac{k}{2\pi} \cdot \Delta x \cdot \Delta y \cdot \sum_{y'=0}^{m-1} \sum_{x'=0}^{n-1} D(x', y') \frac{\vec{r} - \vec{r}'}{\|\vec{r} - \vec{r}'\|^2} \\ &= \frac{k}{2\pi} \cdot \frac{H}{n} \cdot \frac{W}{m} \cdot \sum_{y'=0}^{m-1} \sum_{x'=0}^{n-1} D(x', y') \frac{\vec{r} - \vec{r}'}{\|\vec{r} - \vec{r}'\|^2} \end{aligned} \tag{15}$$

식 (15)를 계산하면 적분을 계산하지 않고도 부가 힘을 계산할 수 있다. 벡터 \vec{e}_x, \vec{e}_y 는 각 셀의 x 방향, y 방향으로 작용하는 부가 힘을 포함하고, 식 (4)에 벡터 \vec{e} 를 더하여 식 (16)을 얻을 수 있다.

$$C \cdot \vec{x} + \vec{a}_x + \vec{e}_x = 0 \tag{16-1}$$

$$C \cdot \vec{y} + \vec{a}_y + \vec{e}_y = 0 \tag{16-2}$$

식 (16)은 배치문제를 부가 힘 \vec{e} 를 찾은 후, \vec{x}, \vec{y} 를 계산하는 문제로 변환한 것이다.

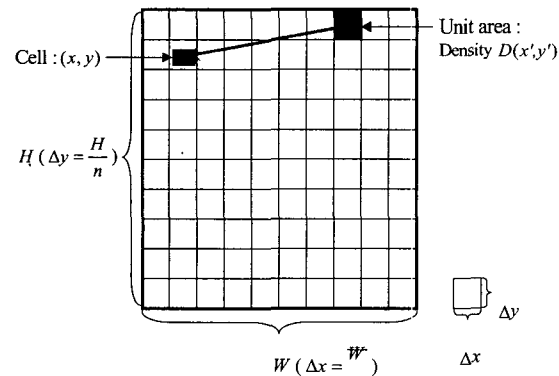


그림 4. 부가 힘 (discrete sum)
Fig. 4. Additional Force(discrete sum).

(3) 새로운 부가 힘(New Additional Force)

2 항에서 설명한 기존 부가 힘의 문제점은 적절한 크기의 힘을 찾기 어렵다는데 있다. 지나치게 큰 힘을 적용하여 quadratic optimization을 수행하면 배치결과가 나빠지게 된다. 특정 셀 i 에 작용하는 부가 힘은, 셀 i 가 다음 quadratic optimization 과정에서 현재 위치에서 어느 방향으로 움직일 것인지를 결정하는 요소가 된다. 따라서 셀 i 에 적정수준 이상의 부가 힘이 작용하게 되면 셀 i 는 배치 영역 밖으로 나가게 될 가능성이 발생한다(그림 5의 왼쪽). 이로 인해 셀 i 뿐만 아니라 셀 i 와 연결된 다른 셀들에게도 영향을 주어 배치결과를 나쁘게 하는 원인이 된다(그림 5의 오른쪽). 그림 5은 지나친 부가 힘이 적용된 예를 그림으로 보여주는 것이다.

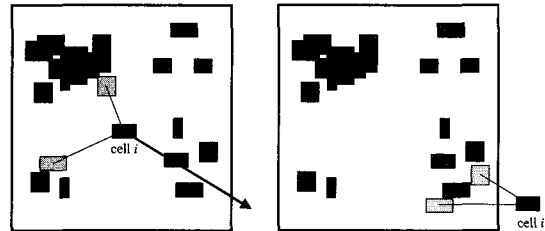


그림 5. 지나친 부가 힘이 적용된 예
Fig. 5. Example of Excessive Additional Force.

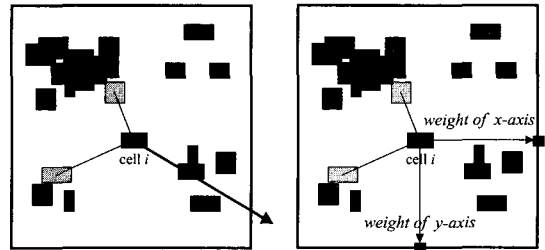


그림 6. 새로운 부가 힘의 개념
Fig. 6. Concept of New Additional Force.

본 논문에서는 이 문제를 해결하기 위해 부가 힘을 가상 고정셀에 의해 발생하는 힘으로 모델화하여 개략 배치를 수행한다. 먼저 모든 movable 셀에 대해 가상 고정셀이 x 축과 y 축에 하나씩 존재한다고 가정하고 가상 고정셀이 각 movable 셀과 연결됨으로써 발생하는 행렬 C 의 대각성분을 계산해서 새로운 연결 행렬 C_x, C_y 를 만든다.

행렬 $C_x(C_y)$ 는 다음과 같은 과정을 통해 계산할 수

있다. 부가 힘을 계산하는 식 (15)로 계산된 $e_x(e_y)$ 를 그림 6에서와 같은 모델을 적용하면, 셀 i 와 x 방향의 가상 고정셀 x_i (y 방향의 가상 고정셀 y_i)를 연결한 가상 네트에 부여된 가중치 $w_{x_i}(w_{y_i})$ 는 x_i 의 좌표가 (x_{x_i}, y_{x_i}) (y_i 의 좌표가 (x_{y_i}, y_{y_i}))일 때 다음과 같다.

$$w_{x_i} = \frac{e_{x_i}}{x_{x_i}}, \quad w_{y_i} = \frac{e_{y_i}}{y_{y_i}} \quad (17)$$

식 (17)에서 계산된 가상 네트의 가중치는 i 번째 대각 성분이 $w_{x_i}(w_{y_i})$ 이고 나머지 성분은 모두 0으로 하는 대각행렬(diagonal matrix) $W_x(W_y)$ 를 정의할 수 있다.

$$W_x = \begin{bmatrix} w_{x_1} & 0 & \dots & 0 \\ 0 & w_{x_2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & w_{x_n} \end{bmatrix}, \quad W_y = \begin{bmatrix} w_{y_1} & 0 & \dots & 0 \\ 0 & w_{y_2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & w_{y_n} \end{bmatrix} \quad (18)$$

행렬 $W_x(W_y)$ 와 행렬 C 로부터 얻어진 새로운 연결행렬 $C_x(C_y)$ 는 다음과 같다.

$$C_x = C - W_x \quad (19-1)$$

$$C_y = C - W_y \quad (19-2)$$

식 (19)에서 얻어진 행렬 C_x, C_y 를 식 (16)의 행렬 C 와 치환하면 다음과 같은 식을 얻을 수 있다.

$$C_x \cdot \vec{x} + \vec{a}_x + \vec{e}_x = 0 \quad (20-1)$$

$$C_y \cdot \vec{y} + \vec{a}_y + \vec{e}_y = 0 \quad (20-2)$$

이렇게 얻어진 식 (20)을 풀면 부가 힘의 지나친 적용을 막아서 좀더 안정적인 배치결과를 얻을 수 있다.

2. 상세배치 방법

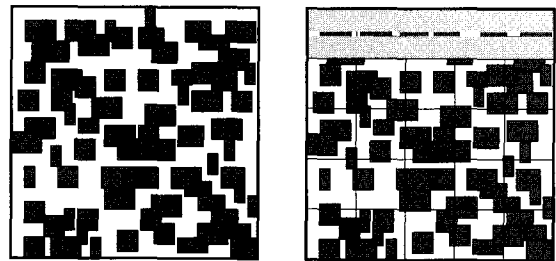
본 논문에서 구현된 상세배치는 사분할을 이용한 부분개선 과정과 셀 sliding을 이용한 셀겹침 제거 과정으로 구성된다.

부분개선 과정에 사용된 방법은 사분할 방법에 기초를 둔 것으로 MST(minimum spanning tree)를 비용함수로 사용한다. 분할기반의 배치는 회로 내의 셀들을 둘 또는 네 부분으로 분할영역 사이의 연결선(cut)이 작도록 분할하고 분할된 각 영역을 같은 방법으로 영역의 크기가 충분히 작아질 때까지 연속적으로 분할하여 이를 배치결과로 사용하는 방법이다.^[7,11] 특히, Huang 등이 발표한 배치방법^[10]은 사분할 방법을 기초

로 한 배치방법으로 성능이 우수한 것으로 알려져 있어, 본 논문의 상세배치 중 부분개선 과정에 사용한다.

부분개선을 위한 사분할은 개략배치 결과에 의한 셀 위치와 분할을 위한 영역 정보를 입력으로 한다(그림 7). 분할영역의 크기가 큰 경우에는 분할을 여러 번 수행해야 최종분할을 얻을 수 있고, 작으면 빠르게 최종 분할에 도달할 수 있으나 비용개선 효과가 적기 때문에 영역 정보는 부분개선을 어느 정도 수준에서 시작할 것인지를 결정하는 요소가 된다. 실험에 의하면 분할영역의 크기를 셀 row 크기의 2배 정도로 하는 것이 좋은 결과를 얻을 수 있다. 그림 7의 a는 개략배치의 결과를 나타내고 그림 7의 b는 개략배치로부터 부분개선의 입력으로 주어지는 분할영역을 나타낸 것이다.

셀겹침 제거 과정은 부분개선 과정으로부터 넘겨받은 셀의 현재 배치를 기반으로 셀들 사이의 겹침을 완전히 제거하고 MST를 최소로 하는 새로운 배치를 얻기 위한 과정이다. 배선을 위해 셀들 사이에 간격이 존재할 때, 그 폭은 feedthrough 너비의 정수배가 되도록 조정한다. 셀겹침 제거 과정에 사용된 셀 sliding은 dynamic programming으로 구현되었다.



a. 개략 배치 결과

b. 부분개선을 위한 분할영역 (row count = 2)

그림 7. 부분개선을 위한 입력

Fig. 7. Input for Local Improvement

3. 시간지연을 고려한 배치

시간지연을 고려한 배치를 수행하기 위해서는 타이밍 분석을 위한 부분과 분석결과를 배치에 적용하는 부분이 필요하다. 타이밍 분석을 위해서는 실제 배선결과가 필요하지만 배치과정 중에는 배선형태를 알 수 없으므로 star model^[15]을 사용한다. 즉, 네트의 각 셀 위치의 중점을 계산하고 네트의 소스(source)에서 타겟(target)까지 이 점을 통하여 연결된다고 가정하고 Elmore 시간지연^[5,16]를 계산한다. 모든 네트에 대한 시간지연을 계산한 후 시간지연 분석을 통하여 모든 네

트에 대하여 시간지연의 슬랙을 계산한다. 그 결과를 LBA(Limit Bumping Algorithm)^[8]를 사용하여 각 네트의 소스에서 타겟까지 시간지연의 상한을 계산한다. 시간지연과 상한과의 차이가 작은 네트들은 critical 네트로 분류하고, 이 네트들에 가중치를 부여하여 실제 배치에 적용한다.

```

1. Global_Placement( )
{ ...
  { ...
    //for delay driven
11.   if ( Timing_Driven )
    {
      //timing analysis
12.   timing_alalysis():
      //net weight update for critical net
13.   update_weight():
    }
  }
}

```

그림 8. 시간지연을 고려한 개략배치 과정
Fig. 8. Timing Driven Global Placement Procedure.

시간지연을 고려한 개략배치를 수행하기 위해서는 그림 2에서 보인 개략배치 과정에 시간지연을 위한 부분을 추가해야한다. 그림 8은 시간지연을 고려한 개략배치의 과정을 보인 것이다. 개략배치에서 시간지연을 고려한 배치(그림 8의 11-13행)를 수행하기 위해서는 기본적인 반복수행 과정에 시간지연 분석부분(그림 8의 12행)과 그 결과를 네트 가중치(weight)에 반영하는 부분(그림 8의 13행)을 추가함으로써 시간지연을 고려한 배치를 수행할 수 있다.

네트의 가중치를 갱신하기 위해서 각 네트마다 criticality c 를 정의하여 사용한다. m 번째 반복 수행 중일 때, 네트 j 에 대해서 criticality $c_j^{(m)}$ 를 다음과 같이 정의하고, $c_j^{(0)}$ 는 0으로 초기화한다.

$$c_j^{(m)} = \begin{cases} \frac{c_j^{(m-1)} + 1}{2} & \text{if net } j \text{ is critical.} \\ \frac{c_j^{(m-1)}}{2} & \text{otherwise} \end{cases} \quad (21)$$

이렇게 정의된 criticality를 사용하여 네트의 가중치를 갱신한다. m 번째 반복 수행 중일 때, 네트 j 는 가중치 $w_j^{(m)}$ 를 갖는다. 네트 가중치는 초기에 $w_j^{(0)} = 1$ 을 값으로 갖는다.

$$w_j^{(m)} = w_j^{(m-1)} \times (1 + c_j^{(m)}) \quad (22)$$

이렇게 구해진 가중치를 적용하여 식 (20)을 풀면 시간지연을 고려한 개략배치 결과를 얻을 수 있다.

여기서 식 (22)과 같은 네트 가중치 갱신 방법을 쓰는 이유는 criticality c 를 사용해서 급격한 네트 가중치의 변화를 막아서 진동효과(oscillation effect)를 방지하기 위한 것이다. 네트 가중치방법에서 진동효과란 이전단계의 critical 네트가 높은 가중치 적용에 의해 현재 단계에서는 critical이 아닌 네트가 될 경우 일반 네트와 같은 가중치를 적용했을 때 다음 단계에서 다시 critical 네트가 되는 현상을 의미한다.

시간지연을 고려한 상세배치를 위해 개략배치에 사용된 것과 같은 방법으로 시간지연 분석을 수행하고, critical 네트로 분류된 네트들에 대하여 식 (21)와 식 (22)에 사용된 방법과 유사한 방법으로 네트 벡터의 비중에 지연시간을 크게 하는 분할형태에 추가 비용을 부여한다. 이렇게 네트비용을 네트의 criticality에 따라 설정하면 차후 분할개선 단계에서 critical 네트에 연결된 셀들은 보다 가까이 배치되도록 하여 지연시간이 작아지도록 한다.

III. 실험결과

본 논문에서 제안한 표준셀 배치는 Microsoft Windows98기반의 Microsoft Visual C++6.0으로 구현되었다. 구현에 사용된 시스템은 256Mbyte의 주메모리와 하나의 Intel PentiumIII 500Mhz 프로세서를 가진 시스템이다. 결과를 비교하기 위해 배선거리와 시간지연을 측정하였는데, 배선거리를 측정하기 위해 Minimum Spanning Tree(MST)를 사용하였으며 시간지연은 2장에서 기술한 방법으로 계산하였다. 결과 비

표 1. MCNC 벤치마크 회로
Table 1. MCNC Benchmark Circuits.

circuit	cells	pads	nets	pins
fract	125	24	147	462
struct	1888	64	1920	5471
industry1	2271	814	2593	8513
biomed	6417	97	5742	26947
avqsmall	21854	64	22124	82601
avqlarge	25114	64	25385	82601

교를 위한 회로는 MCNC 벤치마크 회로^[13]를 사용하였으며, 그 특징은 표 1과 같다. 이 회로들 중 시간지연을 계산하기 위한 정보를 포함한 회로는 fract, struct, avqsmall 이다.

배치결과를 TimberWolf v7.0과 상용 버전의 TimberWolf(Itools v1.4)와 비교하였다. TimberWolf는 simulated annealing을 이용한 대표적인 배치/배선기로서 Itools v1.4 가 가장 최신 버전이다. 본 논문에서 제안한 배치기와 결과를 비교하기 위해 TimberWolf v7.0과 Itools v1.4의 배치기를 수행한 후, 그 출력을 다시 구현된 배치기로 읽어들이어 MST와 시간지연을 계산하였다. 최근의 공정들이 배선을 위한 N-layer metal (N ≥ 3)을 지원하기 때문에 배선을 위한 채널이 필요치 않게 되었다. 따라서 배치결과를 비교하기 위해 TimberWolf v7.0은 채널의 높이를 0으로 하여 비교하였고 Itools v1.4는 최소한의 채널 높이만을 주고 실행결과를 비교하였다. 본 논문에서 구현한 배치기의 결과는 시간지연을 고려한 것(TD)과 고려하지 않은 것(NTD) 두 가지 경우에 대해 결과를 측정하였다.

각 결과표에서 회로이름에서 괄호 안의 숫자는 배치에 사용된 row의 수이고, MST의 단위는 10⁻⁶ 미터(m)이고, 시간지연의 단위는 10⁻⁸초(sec)이다. 각 결과표의 MST 항목과 Delay항목의 괄호 안의 숫자는 TimberWolf의 결과를 1.0으로 한 배선거리와 시간지연의 상대적인 크기이다.

표 2는 시간지연을 고려하지 않은 배치를 TimberWolf v7.0과 비교한 것이다. 이 경우, TimberWolf v7.0에 비해 배선거리에서 평균 7.5%의 향상된 결과를 보인다. 표 3의 배치결과는 시간지연을 고려한 경우이다. 표 3에서 TimberWolf 결과는 시간정보를 포함한 회로를 사용하여 얻었다. 이 경우, TimberWolf v7.0에 비해 배선거리에서 평균 5.0%의 향상된 결과를 보이고, 시간지연에서 거의 같은 결과를 보인다.

Itools v1.4와 비교한 결과는 표 4과 같다. 제안된 배치는 Itools v1.4와 비교했을 때 배선거리 및 시간지연에서 비슷한 결과를 얻을 수 있다.

IV. 결론

본 논문에서는 새로운 모델의 부가 힘을 사용한 QP 기반의 표준셀 방식의 성능지향형 배치기를 제안하였다. 제안된 방법은 부가 힘을 가상 고정셀(dummy

fixed cell)에 의해 발생하는 힘으로 모델화하여 계산하였다. 이렇게 계산된 새로운 부가 힘은 기존의 부가 힘의 문제점을 개선하여 보다 안정적인 배치결과를 얻을 수 있도록 하였다. 구현된 배치는 3장의 실험결과에서 보였듯이 배선거리(MST) 및 시간지연에 있어서 TimberWolf v7.0 보다는 향상되고, 상용 버전의 TimberWolf (Itools v1.4)와는 비교 할 만 하거나 일부 더 나은 배치결과를 보여주었다.

표 2. 배치 결과(NTD)-TimberWolf v7.0.
Table 2. Placement Result(NTD)-TimberWolf v7.0.

circuit	TimberWolf 7.0 (NTD)		논문에서 구현된 배치기 (NTD)	
	MST		MST	
fract (8)	54572		53388 (0.978)	
struct (29)	630237		585823 (0.930)	
industry1 (29)	137756		127468 (0.925)	
biomed (62)	4530390		4095230 (0.904)	
avqsmall (112)	6684370		6017451 (0.900)	
avqlarge (118)	7228950		6606172 (0.914)	
average	-		- (0.925)	

표 3. 배치 결과(TD)-TimberWolf v7.0.
Table 3. Placement Result(TD)-TimberWolf v7.0.

circuit	TimberWolf 7.0 (TD)		논문에서 구현된 배치기 (TD)	
	MST	delay	MST	delay
fract (8)	57123	1.83	54763 (0.959)	1.88 (1.027)
struct (29)	636111	9.02	613703 (0.965)	8.75 (0.970)
avqsmall (112)	6807490	7.69	6305890 (0.926)	7.71 (1.002)
average	-	-	(0.950)	(1.000)

표 4. 배치 결과 - Itools v1.4
Table 4. Placement Result - Itools v1.4.

circuit	Itools 1.4		논문에서 구현된 배치기			
			(NTD)		(TD)	
	MST	delay	MST	delay	MST	delay
frac (8)	57627 (1.0)	1.91 (1.0)	56317 (0.977)	1.97 (1.031)	57340 (0.995)	1.81 (0.948)
struct (29)	628451 (1.0)	9.07 (1.0)	605616 (0.964)	8.85 (0.976)	639275 (1.017)	8.63 (0.951)
avqsmall (112)	6347220 (1.0)	7.70 (1.0)	6412360 (1.010)	7.99 (1.038)	6936680 (1.093)	7.83 (1.017)
average	- (1.0)	- (1.0)	- (0.984)	- (1.015)	- (1.035)	- (0.972)

QP 기반의 배치에서는 최소한 하나의 고정셀이 요구됨은 물론 고정셀의 위치에 따라 배치결과가 영향을 받게 된다. 따라서, 고정셀의 위치가 주어지지 않는 경우에 대해 최적의 고정셀 위치를 찾는 방법에 대한 연구가 필요하다.

참 고 문 헌

- [1] S. M. Areibi. "Towards Optimal Circuit Layout Using Advanced Search Techniques," PhD thesis, University of Waterloo, Ont. Canada, 1995.
- [2] K. Doll, F. M. Johannes, and K. J. Antreich, "Iterative placement improvement by network flow methods," *IEEE Trans. CAD*, Vol. 13, pp. 1190~1200, Oct. 1994.
- [3] A. Dunlop and B. Kernighan, "A Procedure for Placement of Standard-Cell VLSI Circuits," *IEEE Trans. CAD*, Vol. CAD-4, No. 1, pp. 92~98, Jan. 1985.
- [4] H. Eisenmann and F. M. Johannes, "Generic Global Placement and Floorplanning," *Proc. of ACM/IEEE DAC*, pp. 269~274, 1998.
- [5] W. C. Elmore, "The Transient Response of Damped Linear Network with Particular Regard to Wideband Amplifiers," *J. of Applied Physics*, Vol. 19, pp. 55~63, 1948.
- [6] H. Etawil, S. Areibi, and A. Vannelli, "Attractor-Repeller Approach for Global Placement," *Proc. of IEEE/ACM ICCAD*, pp. 20~24, 1999.
- [7] C. M. Fiduccia and R. M. Mattheyses, "A Linear-Time Heuristic for Improving Network Partitions," *Proc. of ACM/IEEE DAC*, pp. 175~181, 1982.
- [8] J. Frankle, "Iterative and Adaptive Slack Allocation for Performance Driven Layout and FPGA Routing," *Proc. of ACM/IEEE DAC*, 1992, pp. 536~542.
- [9] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3ed, Baltimore: Johns Hopkins University, 1996.
- [10] D. J. Huang and A. B. Kahng, "Partitioning-Based Standard-Cell Global Placement with an Exact Objective," *Proc. ACM/IEEE ISPD*, pp. 18~25, April 1997.
- [11] B. W. Kernighan and S. Lin. "An Efficient Heuristic Procedure for Partitioning Graphs," *Bell Syst. Tech. J.*, 49(2):291~307, 1970.
- [12] J. M. Kleinhans, G. Sigl, F. M. Johannes, and K. J. Antreich. "GORDIAN: VLSI Placement by Quadratic Programming and Slicing Optimization," *IEEE Trans. CAD*, Vol. CAD-10, No. 3, pp. 356~365, 1991.
- [13] K. Koziminski, "Benchmarks for Layout Synthesis - Evolution and Current Status," *Proc. of ACM/IEEE DAC*, pp. 265~270, 1991.
- [14] T. Ohtsuki, *Layout Design And Verification*, North Holland, 1985.
- [15] B. M. Riess and G. G. Ettl. "SPEED: Fast and Efficient Timing Driven Placement," *Proc. of the IEEE International Symposium on Circuits and Systems*, pp. 377~380, 1995.
- [16] J. Rubinstein, J. P. Penfield and H. Horowitz, "Signal Delay in RC-Tree Networks," *IEEE Trans. on CAD*, Vol. CAD-2, No. 3, pp. 202~211, 1983.
- [17] G. Sigl, K. Doll, and F. M. Johannes. "An analytical Placement: A Linear or a Quadratic Objective Function?" *Proc. of ACM/IEEE DAC*, pp. 427~432, 1991.
- [18] P. R. Suaris and G. Kedem, "An Algorithm for Quadrisection and Its Application to Standard Cell Placement", *IEEE trans. on CAS*, Vol. 35, No. 3, pp. 294~303, 1988.
- [19] W.-J. Sun and C. Sechen, "Efficient and effective placement for very large circuits", *IEEE Trans. CAD*, Vol. 14, No. 3, pp. 349~359, 1995.
- [20] D. Sylvester and K. Keutzer, "Rethinking Deep-Submicron Circuit Design", *IEEE Computer*, Vol. 32, No. 11, pp. 25~33, November 1999.

저 자 소 개



姜 相 求(學生會員)

1998년 서강대학교 수학과 학사.
2001년 서강대학교 컴퓨터학과 석사.
2001년 3월~현재 서강대학교 컴퓨터학과 박사과정. <주관심분야: 설계자동화 알고리즘, ASIC 설계>

林 鐘 錫(正會員)

현재 서강대학교 컴퓨터학과 정교수