

## 효과적인 OTC 채널 라우터의 구현

장승규<sup>°</sup>, 장호운  
숭실대학교 대학원 전자계산학과

### An Implementation of Efficient OTC(Over-The-Cell) Channel Router

Seung-Kew JANG<sup>°</sup>, Hoon Chang  
Dep't of Computer Science, SoongSil Univ.

**Abstract** - As evolution of the process technology, we proposed the Over-The-Cell channel routing algorithm for the advanced three-layer process. The proposed algorithm simplifies the channel routing problem, and then makes use of Simulated Annealing Technique to converge at global optimal solution. Also, we proposed a new method to remove the cyclic vertical constraints which are known to be the hardest element in the channel routing problem and a way to detect the local minimal solution and escape from it successfully.

#### 1. 서론

채널 라우팅 문제는 VLSI 기술과 더불어 지난 수십 년간 연구되어 왔지만, 대부분 2-레이어에 국한되어 왔고 라우팅의 해 역시 크게 변화하지 못했다[6]. 하지만 90년대부터 관심을 갖기 시작한 새로운 표준셀 모델과 프로세스로 인해서 OTC 라우터가 등장하게 되었다.

본 논문에서 제시하는 알고리즘은 시대의 흐름에 따라서 Advanced Three-Layer Process를 위한 라우팅 알고리즘으로 기존의 OTC 채널 라우터와는 다른 과정으로 라우팅을 수행하게 된다. 제시한 라우터는 SA(Simulated Annealing)을 이용한 라우터로서 최적에 가까운 해를 얻게 된다. 기존에 연구된 많은 라우팅 알고리즘들은 대부분 그래프를 이용한 알고리즘으로 최적의 해를 보장하기 힘들다 [9].

제시한 알고리즘은 문제의 정의를 간단히 하여 기존의 2-레이어 채널 라우팅 문제와 3-레이어 OTC 채널 라우팅 문제에 모두 적용할 수 있고, 기존의 방법에서 사용되는 채널 라우터는 필요하지 않는다. 논문에서는 CVC를 제거하는 방법과, 지역해에서 탈출하는 방법을 제시할 것이다.

#### 2. 라우터의 구현

##### 2.1. 문제의 정의

그림 1에서 보듯이, 모든 네트이 CVC가 제거된 2-터미널 네트이라면 임의의 네트은 하나의 수평 세그먼트와 두개의 수직 세그먼트를 가지게 된다. 수직 세그먼트의 열(Column) 위치는 고정되고 그 길이만 변하게 되고, 수평 세그먼트는 길이는 고정되지만 트랙의 위치만 채널 안에서 수직 방향으로 움직일 수 있게 된다. 이를 수식으로 나타내면 식 1과 같다.  $N_i$ 는 네트이고  $h_{channel}$ 은 채널의 높이이며  $y(h_i)$ 는  $hi$ 의  $y$  좌표의 값이다.

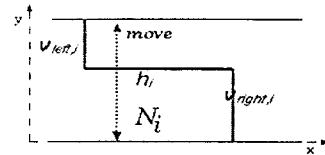


그림 1: 라우팅 문제

$$\begin{aligned} N &= \{N_1, N_2, \dots, N_n\} \quad N_i = \{v_i, h_i\} \\ V &= \{v_1, v_2, \dots, v_n\}, \quad v_i = \{v_{left,i}, v_{right,i}\} \\ H &= \{h_1, h_2, \dots, h_n\} \\ 0 < y(h_i) < h_{channel} \end{aligned} \quad (1)$$

#### 2.2. 채널 라우팅 알고리즘

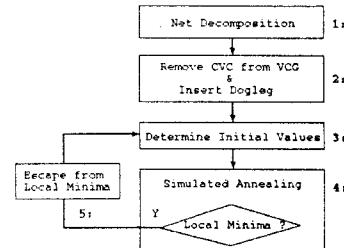


그림 2: 전체적인 알고리즘

본 논문에서 제시하는 알고리즘은 전체적으로 그림 2와 같은 구조를 갖는다. 본 채널 라우터는 Restricted Dogleg과 Unrestricted Dogleg를 모두 허용한다 (1).

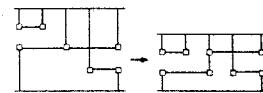


그림 3: Restricted  
Dogleg의 적용

##### 2.2.1. 터미널로 분해(Decomposition)

1: 에서는  $n$ -터미널 네트을 2-터미널 네트으로 바꿈으로서 문제를 간단히 한다. 각각의 네트에 대해 왼쪽의 터미널 부터 순서대로 두개씩 묶어 새로운 넷(서브넷)으로 한다. 따라서  $n$ -터미널 네트은  $n-1$ 의 서브넷을 갖게 된다. 이렇게 함으로써 Restricted Dogleg을 허용하게 된다 [3]. 그림 3에서는 Restricted Dogleg을 적용함으로써 채널이 감소함을 볼 수 있다.

##### 2.2.2. CVC의 제거

본 연구는 한국과학재단 특정기초연구비 961-0100-001-2 지원으로 수행되었으며 지원에 감사드립니다.

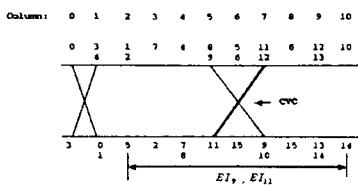


그림 4: CVC를 갖는 채널 예

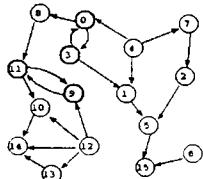


그림 5: 그림 4의  
VCG

2: 에서는 2-터미널 넷으로 바뀐 상태에서 Locally Optimal Breaking을 시도한다 [5]. 이 과정을 거치는 이유는 처음에 주어진 문제에 CVC가 존재하면 바로 SA를 적용하기 힘들기 때문이다. 본 알고리즘에서는 VCG(Vertical Constraint Graph)를 형성한 후에 CVC를 찾아 없앨 수 있도록 하나의 넷을 두개 이상의 넷으로 분리(Split)하는 과정을 거친다. 이 과정에서 Unrestricted Dogleg가 필연적으로 삽입되고, 그 위치는 CVC가 전혀 없도록 확실히 결정되어야 한다.

그림 4은 [7]에서 사용된 예로써 제시한 알고리즘의 동작을 확인하기 위해서 보였다. 이 그림은 2-터미널 넷으로 분해된 상태이다. 터미널 번호와 열 번호은 편의상 0부터 시작하였다. 그림 5의 VCG를 만든 후에 탐색 알고리즘에 의해서 CVC를 형성하는 넷 리스트(Directed Circuits)를 얻을 수 있다  $\{(0,3), (9,11)\}$ . 발견한 DC 중에서 CVC를 제거할 수 있는 넷을 찾아내기 위한 효율적인 알고리즘이 요구된다 [0,11].

구한 각각의 DC에서 넷을 선택하는 문제는 다음 같은 휴리스틱으로 수행하게 된다. 넷  $N_i$ 의 Interval을  $(l_i, r_i)$  라 할 때,  $l_i$ 와  $r_i$ 를 제외한  $(l_{i-k}, r_{i+k})$ 를 확장된 Interval  $EI_i$ 라 정의하자.

$$N_{vacant}(i) : EI_i \text{ 안에서 Vacant 터미널의 개수}$$

$$N_{abutment}(i) : EI_i \text{ 안에서 Vacant Abutment의 개수}$$

$n$ -터미널 넷이 2-터미널 넷으로 분해되었기 때문에 그 넷의 터미널 이외의 터미널에 같은 넷이 존재한다면, 그 터미널은 Vacant 터미널로 볼 수 있다.

하나의 CVC를 제거하기 위해서는 두개의 트랙이 소요되기 때문에 이러한 넷들은 채널의 밀도를 증가 시킬 소지가 크므로 가능하면 셀 위로 옮겨야 한다.

$$Cost(n^*) = \max_{n \in DC} [Cost(n)] \quad (2)$$

$$Cost(n) = \alpha \cdot N_{vacant}(n) + \beta \cdot N_{abutment}(n)$$

수식 1에 따라 DC에 속한 넷( $n$ ) 들에 대해서 평가함수를 적용해 그 중에서 최대의 값을 갖는 넷( $n^*$ )을 선정한다. 이렇게 선택된 넷은 두개의 서브넷으로 분해되는데, 그 위치 즉 Dogleg(Channel Switching)의 위치는  $EI_n^*$ 에 속한 열에 대해서 Local Channel Density의 값과 Vacant Abutment와 Vacant 터미널

의 개수에 따라 식 3의 휴리스틱으로 위치( $c^*$ )를 정하게 된다.

$$Cost(c^*) = \max_{c \in EI_n^*} [Cost(c)] \quad (3)$$

$$Cost(c) = -\alpha \cdot M_{vacant}(c) + \beta \cdot M_{abutment}(c) + \gamma \cdot M_{terminal}(c)$$

수식 3에서  $M_{local}(c)$ 은 컬럼  $c$ 에서의 Local Channel Density를,  $M_{abutment}(c)$ 는 Vavant Abutment의 개수를,  $M_{vacant}(c)$ 은 Vacant 터미널의 개수를 각각 나타낸다. 그림 4의 네트 11의  $EI_{11}$ 에 대해 적용하면 열 9가 선택 된다.

### 2.2.3. 초기값 결정

3: 에서는 SA를 적용하기 전에 초기값들을 정하게 된다. 각 넷의 초기 위치는 임의로 배치되고, 채널의 초기 높이는 이론적인 최소의 값을 갖게 된다. 즉, 채널의 초기 높이는 Horizontal Constraint Graph의 가장 긴 경로인  $v_{max}$ 로 결정된다 [9]. 그리고 2:의 결과를 이용해 새롭게 2-터미널 넷의 문제로 간신히 개선해야 한다.

### 2.2.4. Simulated Annealing 적용

4: 에서는 Simulated Annealing을 적용한다. 이때 가장 중요한 것은 상태의 생성과 평가함수와 온도  $T$ 의 스케줄링이다.

상태를 생성에는 임의로 하나의 넷을 선택하여 그 넷의 트랙을 제약조건 안에서 변경하는 Operator와 임의로 선택된 두개의 넷간의 트랙을 서로 교환하는 Operator를 정의하였고, 두 Operator간의 선택은 확률적으로 선택하도록 하였다.

제시한 알고리즘에서는 식 4과 같은 평가함수를 갖는다.  $OL$ 은 넷들의 Overlap의 빈도수이고,  $NL$ 은 넷들의 길이로 수직 세그먼트의 총 길이의 합이다.

$$Cost = \alpha \cdot OL + \beta \cdot NL \quad (4)$$

일반적인 SA에서는 온도  $T$ 의 스케줄링이 새로운 상태의 이동 시 상수 배로 감소하지만, 제시한 알고리즘에서는 지역해에 빠졌을 때  $T$ 의 값을 증가시킴으로써 지역해에서 탈출하도록 한다.

$$T_{i+1} = \alpha \cdot T_i : \text{if not local minima}$$

$$T_{i+1} = T_i + \gamma : \text{if local minima} \quad (5)$$

### 2.2.5. 지역해의 탈출

5:는 제시한 알고리즘의 가장 중요한 부분으로써 SA 알고리즘의 수행 중에 지역해에 빠졌음을 감지해서 빠르게 탈출하게 만든다. 매번 상태의 이동에서 평가함수의 변화가 일정기간 없으면 지역해로 판단을 내리고 탈출을 위해 먼저 온도  $T$ 를 증가 시킨다. 그리고 일정기간 기다려 보다가 평가함수에 역시 변화가 관찰되지 않으면 깊은 지역해로 판단하여 채널의 높이를 하나 증가시켜 탈출을 시도한다. 두단계로 지역해를 구분함으로써 유연하게 대처하도록 하였다. 주어진 문제에는 CVC가 존재하지 않기 때문에 전역 해로의 수렴을 보장할 수 있다. 그림 6는 지역해에서 탈출하기 위해서 시도 되는 두개의 방법을 보여주고 있다.

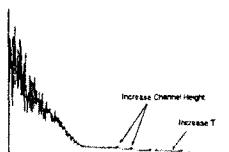


그림 6: 지역해의 탈출

### 3. OTC 채널 라우터로의 확장

OTC 채널 라우터는 표준셀의 라우팅 모델에 따라 알고리즘이 달라지게 된다 [8]. 본 논문에서 제시한 알고리즘이 다른 모델은 HCVD-BTM으로 셀 위에서 Via를 허용하는 Advanced Three-Layer Process를 위한 라우팅 알고리즘이다.

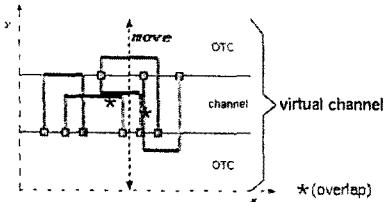


그림 7: OTC 라우터를 위한 문제의 정의

예제	이전 알고리즘		제안한 알고리즘	
	# density	# via	# density	# via
ex1	6	21	1	23
ex2	3	12	3	27
ex3	19	349	18	326
ex4	12	67	4	67

표 1: 실험 결과

#### 3.1. 문제의 재정의

$$0 \leq y(h_i) \leq h_{\text{channel}} + s \cdot h_{\text{OTC}} \quad (6)$$

본 논문에서 제시하는 OTC 채널 라우터는 앞서 제시한 채널 라우팅 알고리즘을 확장함으로써 구현할 수 있다. 이를 위해서 문제를 다시 정의하면, 그림 7에서 보듯이 수평의 세그먼트가 셀 위까지 올라갈 수 있도록 식 6과 같이  $y$  축의 자유도를 높여주고, 채널의 초기 높이를 변경해야 한다. 즉, 초기 채널의 높이는 식 7로 주어지는데 앞서 제시한 알고리즘의 초기 채널에 비해 확연히 감소하게 된다.

$$h_{\text{channel}} = v_{\max} - h_{\text{cell}} + h_{\text{power}} \quad (7)$$

이를 통해 얻어진 해에서, 프로세스와 전기적 특성을 고려해 채널의 레이어 3을 모두 1로 바꾸어 준다.

#### 3.2. 구현 및 결과

제안한 알고리즘은 유닉스 시스템에서 C++과 Motif를 이용하여 구현하였다. 프로그램의 성능을 높이기 위한 방법으로, 평가함수의 계산에서 상당히 많은 계산시간이 소요되므로, 상태의 변화시 변화하는 곳의 Cost만을 최소로 계산하여 이전의 Cost에 변화를 가하는 방법을 사용하였다.

다음의 결과들은 OTC 영역위에 5개의 트랙이 있다고

가정하고 얻은 결과이다. 그림 8과 9은 제안한 알고리즘을 구현하여 [4]와 Deutschs Difficult 예제에 적용한 결과이다. 표 1은 [4], [7], [2], [1]에서 사용되었던 예제에 대해서 적용해본 결과이다.

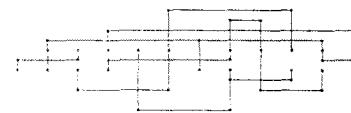


그림 8: 라우팅 결과 I

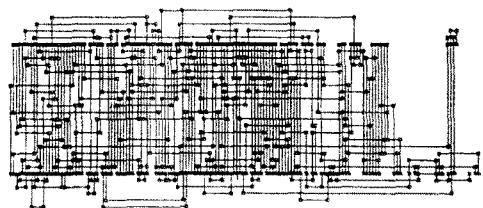


그림 9: 라우팅 결과 II

#### 4. 결론

시대의 추세에 따라 본 논문에서는 Advanced Three-Layer Process에 기초한 OTC 채널 라우팅 알고리즘을 제시하였다.

제안한 알고리즘에서는 문제를 단순화시켜 최적에 가까운 해를 얻기 위해 SA를 이용하였다. CVC를 제거하기 위한 휴리스틱을 제시하였고, SA의 수행에 있어서 지역해에 빠졌음을 감지하여 탈출을 보장하는 방안을 제시하였다. 그 외에 빠르고 최적화된 해로의 수렴을 위한 다양한 휴리스틱을 제안하였다. 이로써, 기존의 라우터에 비해 상대적으로 간단한 알고리즘으로 만족할 만한 결과를 얻을 수 있었다.

#### (참 고 문 헌)

- [1] Randall Jay Brouwer, "Experience with Serial and Parallel Algorithms for Channel Routing using Simulated Annealing", Master Thesis, Univ. of Illinois, 9-17, 1985.
- [2] Michael Burstein, "Hierarchical channel router", INTEGRATION, 1983.
- [3] David N. Deutschs, "A DOGLEG Channel Router", DAC, 1976.
- [4] D. F. Wong Jingsheng Cong and C. L. Liu, "A New Approach to Three- or Four-Layer Channel Routing", IEEE Transactions on CAD, 1988.
- [5] Anthony D. Johnson and Rongchung Sun, "A Genetic Algorithm for VLSI Channel Routing in the Presence of Cyclic Vertical Constraints", Midwest Symposium on Circuits and Systems, 1996.
- [6] Jaewon Kim and Sung-Mo Kang, "A New Triple-Layer OTC Channel Router", IEEE Transaction on CAD, 1996.
- [7] Chi-Yi Hwang Min-Siang Lin, Hourng-Wern Perng and Youn-Long Lin, "Channel Density Reduction By Routing Over The Cells", DAC, 1991.
- [8] Siddhart Bhingarde Naveed Sherwanu and Anand Panyam, "Routing in the Third Dimension", IEEE Press, 1995.
- [9] Naveed Sherwanu, "Algorithms for VLSI Physical Design Automation", Toppan Kluwer, 1996.