JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지(J. Korea Inst. Inf. Commun. Eng.) Vol. 18, No. 8 : 1981~1986 Aug. 2014

P-가지 색을 가진 점들의 할당에 대한 밀도 최소화

김재훈*

Density Minimization for the Assignment of *P*-color Points

Jae-hoon Kim*

Department of Computer Engineering, Busan University of Foreign Studies, Busan 609-815, Korea

요 약

본 논문에서 다루는 문제는 채널의 위쪽 행에 위치한 *P*가지 색을 가지는 점들을 아래쪽 행의 점들에 밀도가 최소가 되도록 연결하는 채널 라우팅 문제이다. 위쪽 행에 위치한 점들이 동일한 색을 가지거나 단지 2가지 색을 가지는 경우는 [1, 2]에서 다루어졌다. 본 논문에서는 *P*가지 색을 가지는 경우로 일반화한다. 우선 임의의 값 *d*가 주어질 때, *d*이하의 밀도를 가지는 할당이 존재하는지 결정하는 문제를 $O(p(n+m)\log(n+m))$ 시간에 풀 수 있음을 보인다. 이를 이용해서 최소 밀도 값의 할당을 찾는 문제를 해결할 수 있음을 보인다.

ABSTRACT

The problem studied in this paper is the channel routing problem to assign points with *p* colors on the upper row of the channel to points on the lower row in order to minimize its density. The case that the points on the upper row has an identical color or only two colors is studied in [1, 2]. This paper generalizes that to the points with *p* colors. First, we consider the problem to determine whether there is an assignment with density less than or equal to *d,* when an arbitrary *d* is given. We show that the problem is solved in $O(p(n+m)\log(n+m))$ time. Using this result, we resolve the problem to fine an assignment with a minimum density.

키워드 : 채널, 채널 라우팅, 밀도, 할당, 결정문제

Key word : channel, channel routing, density, assignment, decision problem

접수일자 : 2014. 05. 08 심사완료일자 : 2014. 05. 14 게재확정일자 : 2014. 05. 27

* **Corresponding Author** Jae-Hoon Kim(E-mail: jhoon@bufs.ac.kr, Tel:+82-51-509-6226)
Department of Computer Engineering, Busan University of Foreign Studies, Busan 609-815, Korea

Open Access http://dx.doi.org/10.6109/jkiice.2014.18.8.1981 print ISSN: 2234-4772 online ISSN: 2288-4165

Ⅰ. 서 론

채널에서 특정 위치들을 선으로 연결하는 채널 라우팅(channel routing) 문제는 집적 회로의 여러 레이아웃 문제에서 발생한다. 특정 위치들을 어디에 놓을 것이지 또는 어떻게 연결할 위치들을 선택할 지와 같은 결정은 라우팅 이전에 이루어져야 한다. 본 논문에서는 두 개의 수평선 상에 위치들이 놓여 있을 때, 이 위치들을 연결하는 문제를 다룰 것이다.

평면상에 일정한 간격을 두고 두 개의 수평선 U와 L이 주어지는데 U는 L보다 위쪽에 위치한다. 위쪽 수평선 U에는 시작점이라 불리는 n개의 점들이 존재하고 아래쪽 수평선 L에는 끝점이라 불리는 m개의 점들이 존재한다. 그리고 항상 $n \leq m$을 만족한다. 또한 위쪽 수평선 U상의 점들은 p가지의 색으로 구별된다.

색을 $k \in \{1, 2, ..., p\}$인 정수로 나타낼 때, U상의 점 a_i^k는 색이 k인 i번째 점을 나타내고 L상의 점은 b_j로 나타낸다. $1 \leq a_i^k, b_j \leq m$ 이고 색 k인 점 a_i^k의 총 수를 n_k로 나타낸다. 우리는 U상의 점들의 집합 P_U에서 L상의 점들의 집합 P_L로의 일대일 함수 $f: P_U \rightarrow P_L$를 찾고자 한다. 점 a_i^k가 점 $b_j = f(a_i^k)$로 대응되었다는 것은 점 a_i^k에서 점 b_j로 선분이 그어졌음을 의미한다. 여기서 함수 f에 의해서 생긴 선분들에 대해서 색 k인 시작점을 연결하는 선분을 k-색 선분(k-color segment)이라고 부른다. 그러면 k-색 밀도(k-color density)란 수직선을 왼쪽에서 오른쪽으로 이동하면서 만나는 k-색 선분들의 개수를 세었을 때, 만나는 k-색 선분들의 개수가 최대가 됐을 때 그 때의 선분 개수를 말한다. 다시 말해서, k-색 밀도란 모든 실수 x에 대해서 $a_i^k \leq x < b_j$ 또는 $a_i^k > x \geq b_j$를 만족하는 k-색 선분들의 개수의 최대값으로 정의한다. 결과적으로 우리는 선분들에 대한 밀도(density)를 k-색 밀도들의 최대값으로 정의할 수 있다. 문제는 이 밀도가 최소가 되도록하는 일대일 함수 f를 찾는 것이다.

본 논문에서는 최소 밀도를 주는 일대일 함수 f를 찾기 이전에 임의의 상수 d가 주어질 때, 밀도가 d이하인 일대일 함수 f가 존재하는지 여부를 묻는 결정 문제를 해결할 것이다. 이 결정 문제 알고리즘을 이용해서 최소 밀도 값 d^*를 계산하고 이 밀도 값을 주는 일대일 함수 f 를 찾을 것이다.

Ⅱ. 관련 연구

본 논문과 가장 밀접한 관련이 있는 논문들은 [1, 2]이다. [1]에서는 수평선 U상의 점들이 모두 같은 색을 가지는 경우를 다룬다. 3장에서 다룰 결정문제에 대해서 저자들은 $O(n+m)$시간 알고리즘을 제안하였다. 최소 밀도를 찾는 최적화 문제에 대해서도 또한 $O(n+m)$시간 알고리즘을 제안하였다.

[2]에서는 수평선 U상의 점들이 단지 두 가지 종류의 색을 갖는 문제를 연구하였다. 저자들은 이 경우에 대한 결정문제에 대해서 $O((n+m)\log(n+m))$시간 알고리즘을 제안하였다. 또한 최소 밀도 값을 $O(n+m)$시간에 찾을 수 있음을 보였고 이 최소 밀도 값을 주는 일대일 함수 f를 $O((n+m)\log(n+m))$ 시간에 찾을 수 있음을 보였다. 본 논문은 이 논문의 모델을 확장해서 U상의 점들이 p가지의 서로 다른 색을 가지는 경우로 일반화할 것이다.

우리는 채널 라우팅 문제를 연구한 여러 다양한 논문들을 찾을 수 있다. [3]에서는 점들을 연결하는 선분들이 수직과 수평 선분 부분들로 이루진 경우를 다룬다. 이 때 수평 선분 부분들의 수를 최소로 하는 레이아웃을 찾는 것이 목적이다.

[4]에서는 L상의 점들을 순서는 유지하면서 오른쪽으로 이동시키는 회전(rotation) 연산을 생각한다. 이 때 최소 밀도 값을 주는 회전 연산을 찾는 문제를 연구하였다.

[5]에서는 U 또는 L상의 점들이 요소(component) 안에 놓여 있는 경우를 다룬다. 그리고 이 요소들은 일직선상에서 왼쪽 또는 오른쪽으로 이동 가능하다. 이 요소들을 이동시켜 밀도 값이 최소가 되도록 하는 문제를 연구하였다.

Ⅲ. 결정 문제

이 장에서는 다음과 같은 결정문제 P1을 다룰 것이

다:

P1 : 임의의 정수 d가 주어질 때, 밀도가 d보다 작거나 같은 선분들을 생성하는 일대일 함수 f가 존재하는가?

순서쌍 $v=(i_1, i_2, ..., i_p)$에 대해서, 부분할당 $f_{v,\ell}$을 정의한다. 이것은 각각의 색 $k=1,...,p$에 대해서, 점 $a_1^k, ..., a_{i_k}^k$가 벌써 L상의 점과 연결되었고, 점 b_ℓ이 연결된 L상의 가장 오른쪽 점이란 것을 의미한다. 이 때, 다음을 만족하는 부분할당 $f_{v,\ell}$을 d-할당이라고 부른다: 각각의 색 k에 대해서, $f_{v,\ell}$은 색 k인 시작점들이 k-색 밀도가 $\leq d$를 만족하는 할당으로 확장될 수 있다.

d-할당 $f_{v,\ell}, \ell \leq h$, 이 주어질 때, 끝점 b_{h+1}의 k-레이블 j_k를 정의한다. 각 k에 대해서, $f_{v,\ell}$에 b_{h+1}과 $a_{i_k+1}^k$을 연결하는 선분을 추가할 때 밀도가 $d+1$이 되면 $j_k=0$. $j_k \neq 0$이면, $j_k=1$ 또는 $j_k=2$. d-할당 $f_{v,\ell}$가 그것의 k-색 밀도가 $\leq d$를 만족하는 할당으로 확장하기 위해서 b_{h+1}가 반드시 $a_{i_k+1}^k$와 연결되어야 하면, $j_k=1$. 그렇지 않다면, $j_k=2$.

아래의 표 1에서 결정 알고리즘이 주어진다. 이 알고리즘에서 함수 $label$은 끝점 b_{h+1}의 모든 k-레이블들의 순서쌍 $w=(j_1,...,j_p)$를 출력한다. 여기서 함수 $label$의 구현을 구체적으로 설명할 것이다. 각 색 k에 대해서, $j_k=0$, 다시 말해서, b_{h+1}과 $a_{i_k+1}^k$을 연결하면 k-색 밀도가 $d+1$가 되는지 결정하는 것은 $O(1)$시간에 할 수 있다. $j_k=1$ 또는 $j_k=2$ 인지 결정하기 위해서 우리는 [2]에서와 같은 균형 이진트리 T_k를 사용할 것이다.

점 $a_1^k, ..., a_{n_k}^k$와 $b_{f(1)}, ..., b_{f(i_k)}, b_{h+1}, ..., b_m$을 정렬한 후, 각 점들을 T_k의 잎노드에 왼쪽에서 오른쪽으로 저장한다. T_k의 내부노드 v는 부가적인 정보들을 저장해서 자신이 루트가 되는 서브트리의 잎노드들에 대한 k-밀도를 계산할 수 있도록 한다. 그러면 이 트리 T_k에서 b_{h+1}을 제거하는 질의를 수행한다. 결과로 k-밀도가 $d+1$로 증가하면 $j_k=1$이고 아니면 $j_k=2$. 이 질의는 $O(\log(n_k+m))$시간에 수행된다. 따라서 $label$ 함수의 출력을 얻기위해서 모든 k에 대해서 이것을 수행하면 $O(p\log(n+m))$시간이 소요된다. 마지막으로 결정 알고리즘이 끝점 b_{h+1}을 어떤 색 c인 시작점과 연결하기로 결정한다면 $k \neq c$인 모든 색 k에 대해서 트리 T_k에서 b_{h+1}을 제거해서 트리를 업데이트 한다.

표 1. 결정 알고리즘
Table. 1 Decision algorithm

```
i_1,..., i_p, h := 0;
F := ∅;
w := label(b_{h+1}); (w = (j_1, ...., j_p))
while (w 안에 많아야 하나 이하의 j_k가 1) and
      (i_1 < n_1 or ... or i_p < n_p) do
{
  if w 안에 하나의 j_k = 1 then
     F := F+ (a_{i_k+1}, b_{h+1});
     i_k := i_k + 1;
  else
     w 안에서 j_k = 2 인 가장 작은 k에 대해서,
     F := F+ (a_{i_k+1}, b_{h+1});
     i_k := i_k + 1;
  h := h + 1;
  w := label(b_{h+1});
}
if w 안에 적어도 두 개 이상의 j_k가 1 then
  write("P1의 함수 f는 존재하지 않는다“);
else output F;
```

결정 알고리즘은 색 k인 시작점들과 끝점들을 각각 왼쪽에서 오른쪽으로 고려하면서 수행한다. $label$의 결과 순서쌍 $w=(j_1,....,j_p)$에서 하나 이하의 j_k가 1이면, 현재 고려되는 끝점 b_{h+1}과 색 k인 시작점 $a_{i_k+1}^k$을 연결하는 선분 (a_{i_k+1}, b_{h+1})을 추가한다. 그렇지 않으면, 다시 말해서, 모두가 1이 아니면, $j_k=2$인 색 k중에 가장 작은 k의 시작점과 끝점을 연결한다. 알고리즘 진행 중에 w안에 적어도 두 개 이상의 j_k가 1

이 되면, 일대일 함수 f는 존재하지 않는다. 결과적으로 위 결정 알고리즘은 $O(p(n+m)\log(n+m))$시간에 수행됨을 알 수 있다.

Ⅳ. 최적화 문제

이 장에서는 다음과 같은 최적화문제 P2를 다룰 것이다:

P2 : 일대일 함수 f에 의해 생성되는 선분들의 최소 밀도를 찾으시오.

이 문제는 3장의 결정 알고리즘을 이용해서 풀 것이다. 다시 말해서, 결정 알고리즘을 이용해서 밀도 값 d에 대해서 이분검색(binary search)을 수행할 것이다.

$1 \le x, y \le M$인 모든 x, y에 대해서, $up_k(x,y)$는 위치가 $\ge x$이고 $\le y$인 색 k인 시작점들의 수로 정의하고, $down(x,y)$는 위치가 $\ge x$이고 $\le y$인 끝점들의 수로 정의한다. 또한 $out_k(x,y)$는 $up_k(x,y) - down(x,y)$로 정의한다. 이 정의로부터 $\psi_1(x,y) = \max\{out_1(x,y), ..., out_p(x,y)\}$, $\psi_2(x,y) = \sum_{k=1}^{p} up_k(x,y) - down(x,y)$라고 하자. 그러면 Φ_1을 다음과 같이 정의한다:

$$\Phi_{11}(x) = \max\left\{\psi_1(1,x), \frac{1}{p}\psi_2(1,x)\right\},$$
$$\Phi_{12}(x,y) = \max\left\{\frac{1}{2}\psi_1(x,y), \frac{1}{2p}\psi_2(x,y)\right\},$$
$$\Phi_{13}(x) = \max\left\{\psi_1(x,M), \frac{1}{p}\psi_2(x,M)\right\},$$
$$\Phi_1 = \max\{\max_x \Phi_{11}(x), \max_{x,y}\Phi_{12}(x,y), \max_x \Phi_{13}(x)\}$$

Φ_1가 최적 밀도 d^*의 하한이 된다는 것, 다시 말해서, $d^* \ge \Phi_1$ 을 만족함을 보이기는 어렵지 않다.

보조정리 4.1. $d^* \ge \Phi_1$.

또한 d^*의 상한을 얻기 위해서 다음과 같이 Φ_2를 정의한다:

$$\Phi_{21}(x) = \max\left\{\psi_1(1,x), \frac{1}{2}\psi_2(1,x)\right\},$$
$$\Phi_{22}(x,y) = \max\left\{\frac{1}{2}\psi_1(x,y), \frac{1}{4}\psi_2(x,y)\right\},$$
$$\Phi_{23}(x) = \max\left\{\psi_1(x,M), \frac{1}{2}\psi_2(x,M)\right\},$$
$$\Phi_2 = \max\{\max_x \Phi_{21}(x), \max_{x,y}\Phi_{22}(x,y), \max_x \Phi_{23}(x)\}$$

다음에서 우리는 Φ_2가 d^*의 상한임을 보인다.

보조정리 4.2. $d^* \le \Phi_2$.

증명. 밀도 Φ_2인 일대일 함수 f가 존재함을 보임으로 충분하다. 다시 말해서, 이전 장의 결정문제 P1에 대해서 $d = \Phi_2$일 때, 결정 알고리즘의 답이 항상 참임을 보이면 된다.

결론이 아니라고 가정하면, 다시 말해서, 결정 알고리즘이 거짓을 출력하였다고 가정하자. 알고리즘의 동작에서 점 $a^1_{i_1+1}, ..., a^p_{i_p+1}, b_{h+1}$을 고려할 때 실패하였다면 b_{h+1}의 레이블 w안에는 적어도 둘 이상이 1이다. 하지만 이런 일은 일어나지 않음을 증명할 것이다.

레이블 w안에서 1의 값을 갖는 임의의 두 개의 인덱스를 j_α와 j_β라고 하자. 다시 말해서, 색 α와 β 각각에 대해서, 밀도 $d = \Phi_2$ 를 얻기 위해서는 b_{h+1}가 각각 $a_{i_\alpha+1}$과 $a_{i_\beta+1}$에 연결되어야만 한다.

$j_\alpha = 1$ 또는 $j_\beta = 1$이 되는 다음과 같은 두 가지 이유가 있다:

1) b_{h+1}가 없으면, 시작점들과 연결되기 충분한 개수의 끝점이 남아 있지 않는 경우.
2) b_{h+1}가 없어도 충분한 끝점들이 남아있지만 밀도가 $\le \Phi_2$이기 위해서 b_{h+1}와 반드시 연결되어야 하는 경우.

먼저 $j_\alpha = 1$, $j_\beta = 1$의 적어도 한 경우가 1) 때문인 경우를 생각하자. 남아있는 색 α와 β인 점들의 총 수

가 남아있는 끝점들의 수를 넘어가는 순간을 생각한다. 이것이 점 a^{α}_{r+1}, a^{β}_{s+1}, b_{t+1}를 고려할 때 처음으로 나타난다고 가정하자. 그러면, $n_{\alpha}-r+n_{\beta}-s$은 $m-t+1$과 같다. $b_{t+1-\lambda}$를 $\leq b_{t+1}$ 위치에 있는 연결되지 않은 가장 오른쪽 끝점이라고 하자. 그러면 이 점에서의 레이블 $w=0$. 따라서 $b_{t+1-\lambda} < a^{\alpha}_{r+1}$이고 $b_{t+1-\lambda} < a^{\beta}_{s+1}$. 또한 $\leq b_{t+1-\lambda}$에 위치한 끝점들에 연결된 $> b_{t+1-\lambda}$인 각각 Φ_2개의 색 α와 β인 점들이 존재한다. Δ를 $b_{t+1-\lambda}$과 b_{t+1} 사이에 있는 끝점들과 연결된 시작점들의 수라고 하자. 그러면,

$$\frac{1}{2}\psi_2(b_{t+1-\lambda}, M) \geq \frac{1}{2}\{\Phi_2 + \Delta + (n_{\alpha}-r) + \Phi_2 + (n_{\beta}-s) - (\Delta + (m-t))\} > \Phi_2.$$

이것은 모순이다. 따라서 $j_{\alpha}=1$, $j_{\beta}=1$의 어떤 경우도 1) 때문은 아니다.

남은 것은 $j_{\alpha}=1$, $j_{\beta}=1$ 가 모두 2) 때문인 경우이다. S_{α}를 색 α인 점들만을 고려해서 $a_1, \ldots, a_{i_{\alpha}}$은 f와 같이 연결하고 남은 점들을 $b_{h+1}, \ldots, b_m$에 연결하는 밀도 Φ_2의 해답이라고 하자. 그러면 S_{α}에서 적어도 하나의 좌표 $x_{\alpha} \geq b_{h+1}$에 대해서 $\leq x_{\alpha}$에 위치한 시작점과 $> x_{\alpha}$인 끝점을 연결하는 Φ_2개의 선분들이 존재한다. 따라서 위 조건을 만족하는 가장 왼쪽의 좌표를 x_{α}이라고 하자. 그러면 S_{α}에서 $\geq b_{h+1}$이고 $\leq x_{\alpha}$인 위치의 끝점들은 모두 α색의 시작점들과 연결된다. 비슷하게 색 β에 대한 S_{β}를 생각할 수 있고 x_{β}를 정의한다. 일반성의 손실 없이, $x_{\alpha} \leq x_{\beta}$.

위치가 $> b_{h+1}$이고 $\leq x_{\alpha}$끝점들의 개수를 δ_1이라 하고 $> b_{h+1}$이고 $\leq x_{\beta}$인 끝점들의 개수를 δ_2라고 하자. 그러면 $a^{\alpha}_{i_{\alpha}+1+\delta_1+\Phi_2}$는 위치 $\leq x_{\alpha}$인 가장 오른쪽 색 α 점이고 $a^{\beta}_{i_{\beta}+1+\delta_2+\Phi_2}$는 위치 $\leq x_{\beta}$인 가장 오른쪽 색 β 점이다. δ_3를 $x_{\alpha}+1$과 x_{β} 사이의 색 α 점들의 수라고 하면, $a^{\alpha}_{i_{\alpha}+1+\delta_1+\Phi_2+\delta_3}$는 위치 $\leq x_{\beta}$인 가장 오른쪽 색 α 점이다.

우선 b_{h+1} 왼쪽의 모든 끝점들이 연결되어 있다고 가정하자. 그러면,

$$\begin{aligned}\frac{1}{2}\psi_2(1, x_{\beta}) &\geq \frac{1}{2}\{(i_{\alpha}+1+\delta_1+\Phi_2+\delta_3) \\ &+ (i_{\beta}+1+\delta_2+\Phi_2) - (i_{\alpha}+i_{\beta}+1+\delta_2)\} \\ &= \frac{1}{2}\{\delta_1+\delta_3+2\Phi_2+1\} > \Phi_2.\end{aligned}$$

이것은 모순이다.

b_{h+1} 왼쪽의 모든 끝점들이 연결되어 있지는 않다고 가정하자. b_z를 그러한 연결되지 않은 가장 오른쪽 끝점이라고 하면, $< b_z$인 끝점들과 연결된 각각 Φ_2개의 색 α와 β인 시작점들이 $> b_z$에 존재한다. 위와 비슷한 이유에 의해서 다음을 만족한다:

$$\begin{aligned}\frac{1}{4}\psi_2(b_z, x_{\beta}) \geq \frac{1}{4}\Big\{&up_{\alpha}(b_z, x_{\beta}) + up_{\beta}(b_z, x_{\beta}) \\ &- down(b_z, x_{\beta})\Big\} > \Phi_2\end{aligned}$$

그러나 이것은 모순이다. 위의 두 보조정리로부터 $\Phi_1 \leq d^* \leq \Phi_2$임을 알 수 있다. 따라서 밀도 값 d에 대한 이분검색을 수행할 때, 위의 범위에서 수행하면 $O(p(n+m)\log(n+m)\log(\Phi_2-\Phi_1))$시간에 풀 수 있다.

Ⅴ. 결 론

본 논문에서는 [1, 2]에서 다루었던 문제에서 수평선 U상의 점들이 p가지의 색을 가지는 경우로 확장하였다. 임의의 상수 d가 주어질 때, 밀도가 d이하인 일대일 함수 f가 존재하는지 여부를 묻는 결정문제에 대해서 $O(p(n+m)\log(n+m))$시간 알고리즘을 제안하였다. 또한 최소 밀도 값을 d^*를 계산하고 이 밀도 값을 주는 일대일 함수 f 를 찾는 $O(p(n+m)\log(n+m)\log(\Phi_2-\Phi_1))$시간 알고리즘을 제안하였다. 앞으로 채널 라우팅 문제에 대한 여러 변형 문제들에 대해서도 연구할 수 있을 것이다.

감사의 글

본 연구는 2014년도 부산외국어대학교 학술연구조성비에 의해 연구되었으므로, 관계부처에 감사 드립니다.

REFERENCES

[1] M. J. Atallah and S. E. Hambrusch, "On terminal assignments that minimize the density," Purdue University, Technical Report CSD-TR-468, 1984.

[2] M. J. Atallah and S. E. Hambrusch, "An assignment algorithm with applications to integrated circuit layout," *Discrete Applied Mathematics,* vol. 13, no. 1, pp. 9-22, 1986.

[3] T. Yoshimura and E. S. Kuh, "Efficient algorithms for channel routing," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems,* vol. CAD-1, no. 1, January, pp. 25-35, 1982.

[4] M. J. Atallah and S. E. Hambrusch, "Optimal rotation problems in channel routing," Purdue University, Technical Report CSD-TR-386, 1984.

[5] D. S. Johnson, A. S. LaPaugh, and R. Y. Pinter, "Minimizing channel density by lateral shifting of components," in *Proceeding of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 122-131, 1994.

김재훈(Jae-Hoon Kim)

1994년 서강대학교 수학과 이학사
1996년 KAIST 수학과 이학석사
2003년 KAIST 전산과 공학박사
2003년 - 현재 부산외국어대학교 컴퓨터공학과 교수
※관심분야 : 알고리즘, 최적화, 스케줄링