□特 輯□

VLSI 설계와 그래프 알고리즘

부산대학교 전자계산학과 조 환 규*

●목 차●

Ⅰ. 서 론
Ⅱ. 설계 자동화와 그래프 알고리즘
Ⅲ. 회로배치에 관한 이론적 문제
Ⅳ. 회로배선에 관한 이론적 문제
Ⅴ. 최근의 연구동향과 결론

Ⅰ. 서 론

전산학의 이론을 이루고 있는 분야는 여러가지가 있지만 그 중에서도 가장 중요한 한 분야중의 하나가 그래프이론(Graph Theory)이라고 생각된다. 그래프가 가지는 다양성과 그것으로 인해서 표현되는 여러가지 상태는 복잡한 컴퓨터의 구조와 그에 의거해서 운용되는 프로그램의 상태를 나타내기에 다른 어떤 도구보다도 적합하다. 예를 들어 인공지능에서 보이는 의미네트워크(Semantic Network)나 데이타 베이스에서 보여지는 여러 문제들이 그래프로 모델링되고 그 문제는 결국 그래프 이론의 한 문제로 귀납되어 해결되는 것은 매우 일반적인 일이다.

최근 들어 급격한 발전 속도를 보이고 있는 병렬처리 컴퓨터나 프로그래밍언어 이론 역시 그래프이론의 도움을 크게 받고 있다. 특히 병렬 처리기의 속도한계나, 어떤 문제가 얼마나 효율적인 정도로 병렬화 되는가하는 것이 중요한 문제라고 생각될 때, 이를 그래프문제화 시키면 궁극적으로는 대부분 어떤 클래스의 그래프의 특성을 탐구하는 문제, 예를 들어 가장 긴 경로(path)나 Clique, 최대의 갯수를 가지는 독립 경로(Maximal Disjoint Paths) 등의 문제로 귀납됨을 볼 수 있다.

이 중에서도 비교적 최근들어 발전하기 시작한 설계 자동화(Design Automation)분야는 가장 그래프 이론적인 도움을 필요로 하고 있다. 그것은 우리가 흔히 보는 회로도 그 자체가 바로 그래프 구조를 그대로 닮고 있기 때문이다. 좀더 정확히 이야기하자면 보통의 회로도는 그래프 중에서도 보다 일반적인 하이퍼 그래프 구조와 같다. 따라서 회로도를 이용해서 칩을 설계하는 문제는 대부분 이전 많이 연구되어 온 그래프 문제와 밀접하고도 직접적인 연관을 가지게 되므로, 설계 자동화의 여러 알고리즘을 다루는 문제는 응용적인 의미 뿐만이 아니라 그래프구조를 탐구한다는 데에도 큰 의미가 있다고 보여진다. 따라서 설계 자동화는 이론 전산학에서 발생하는 다양한 결과가 가장 직접적으로 작용되며, 또한 이론 전산학의 최대 난제인 NP-Complete류의 문제를 우리가 어떻게 대처해 나가야 하는가에 대해서 많은 전망을 보여주는데에서 의미가 있다. 따라서 본고에서의 주안점은 이론 전산학의 한 분야, 그 중에서도 그래프 이론 분야의 많은 이론적 문제들이 가장 직접적으로 나타나는 설계 자동화의 문제에 어떻게 응용되며 이로부터 새롭게 나타나는 이론적 문제들에는 어떤 것들이 있는가를 살펴보는데 있다.

이를 위해서 우리는 먼저 기존의 회로도가 어떻게 그래프 이론적인 문제로 변환되는 그 과정을 살펴본다. 그리고 본고의 3, 4장에서는 설계 자동화의 과정 중에서도 주된 과정이면서 그래프이론과 가장 밀접한 배치(Placement)문제와 배선(Routing)문제만을 그래프 알고리즘을 중심으로해서 살펴본다. 마지막으로는 최근의

* 정회원

연구동향과 함께 나타난 연구문제를 소개하고 각 문제에서 어떤 점들이 그 해결을 어렵게 하는지 살펴보기로 하자.

II. 설계 자동화와 그래프 알고리즘

설계자동화란 주로 초집적회로 설계시에 나타나는 여러가지 문제를 사람의 개입없이 컴퓨터의 도움만으로 해결해 보고자하는 전산학의 한 분야이다. 그 단계는 회로도 구성 Circuit Network, 칩 면적 예상 Area Estimation, 배치 설계 Floor Plan, 배치 Placement, 배선 Routing, 검증 Verification, 공정 Physical Design 등의 단계를 거쳐 시제품을 만든 뒤 비로소 대량생산의 단계로 가게 되는 상당히 복잡한 과정이다. 그러나 최근 들어 반도체 기술의 빠른 발전 속도로 말미암아 그 공정은 자동화를 필수적인 단계로 받아들이고 있다. 예를 들어 어떤 회로의 부품수가 백여개 이하라면 사람의 경험과 지식만으로 설계하여 최종의 공정까지 마칠수 있게 되지만 그 수가 만여개를 넘어간다면 사람의 경험적인 지식만으로 해결하기에 불가능할 것이다. 현재의 대부분 고집적회로는 만여개 이상의 소자를 가지고 있다. 따라서 이러한 경우에 어떤 자동화된 과정, 예를 들면 상위단계 합성(high-level synthesis)이나 실리콘 컴파일러와 같은 도구의 도입이 필수적이며, 그 결과물 역시 사람의 수작업에 비교될 수 있을 정도로 좋아야 할 것을 요구하고 있다.

그래프에 관한 연구는 크게 그래프의 특성을 규명하는 것(Characterizing)과 보다 효율적인 그래프 알고리즘을 개발하는 일로 대별된다[13]. 이중에서도 그래프에 관한 알고리즘은 매우 다양하지만 대개는 특정한 성질을 만족시키는 부그래프(Subgraph) 탐색 문제, 네트워크를 모델로 하여 최단 경로나 다양한 조건의 스패닝 트리(Spanning Tree)를 찾는 문제, 해밀턴 회로(Hamiltonian Cycle)찾기, 매칭(Matching), 버텍스나 에지에 색칠하기(Vertex Coloring, Edge Coloring), 평면그래프 테스팅(Planarity Testing), 두 그래프가 얼마나 같은지를 검사하는 그래프 Isomorphism와 이에 관한 각종 근사 알고리즘(Approximation Algorithm) 등이 주된 연구과제이다[33].

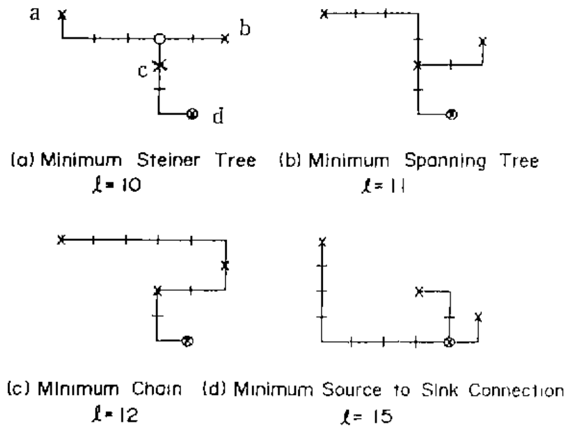
그래프 알고리즘으로서 설계 자동화에 응용되는 것은 그래프 분할문제, 그래프에서 버텍스의 정렬 문제, 그래프에서 에지 혹은 버텍스의 레이블링 문제로 구별된 수 있다. 이 문제들은 보통 배치문제, 혹은 배선 문제라고 불리워진다. 배선문제는 어떤 모듈을 보드의 어떻게 배치하면 가장 좋은가를 다루는 문제인데 각 알고리즘에서 다루는 목적함수(Objective function)는 매우 다양하고 각 응용마다 다르므로 한가지로 고정시켜 말할 수는 없다. 따라서 회로 설계와 연관된 그래프 이론적 문제가 해결된다고 하더라도 실제 회로 설계에 있어서의 전체의 목적함수는 여러가지 목적함수가 혼합된 형태이므로 모든 공학적인 문제를 완벽히 고려한 이상적인 형태의 대수적 목적함수를 발견하기는 매우 힘들다.

III. 회로배치에 관한 이론적 문제

회로를 적당한 위치에 배치하기 위해서는 먼저 회로를 뭉쳐진 몇 단계의 덩어리로 분리해야 한다. 이 작업을 보통 회로분할(Circuit Partition)이라고 부른다. 이 작업의 정도에 따라서 이후에 이어지는 배치와 배선의 성능이 결정되므로 매우 중요하다. 분할이 끝난 뒤 배치, 배선의 작업이 이어지고, 이어서 검증단계를 거치게 된다. 이제 배치에 관한 그래프 이론적 문제를 살펴보기로 하자. 배치문제를 형식적으로 쓰면 다음과 같다.

[일반적인 회로 배치 문제(Placement Problem)] n개의 모듈을 2차원 그리드 평면에 위치시키려 한다. 이때 회로상으로 연결된 두 모듈 (X_i, X_j) 연결하는 선의 길이를 d(X_i, X_j)로 표시할 때, 연결된 모든 모듈들간의 연결 길이 Aij*d(Xi, Xj)의 합을 최소가 되도록 각 위치를 결정하라. 단 여기서 Aij는 모듈 X_i, X_j를 연결하는 선의 수이다.

그런데 모듈의 연결 정도와 계산의 간편정도에 따라서 여러가지 d( )함수가 사용된다. d( ) 함수에는 보통 L_1 메트릭(metric), 즉 두점 (xi, yi)와 (xj, yj)의 거리를 $|xi-xj|+|yi-yj|$로 측정하는 방법을 사용한다. 또는 보통의 유클리디안 메트릭이 사용되거나 임계경로(critical path)가 중요한 문제라면 $\max\{|xi-xj|, |yi-yj|\}$을 사용하기도 한다. 특히 앞에서 언급한데로 각 에지가 하이퍼 그래프의 에지 형태로 되어 있기 때문에 이단 터미날 네트(two terminal net)가 아닌 3개 이상의 네트인 경우에 그 연결될 버텍스를 지나가는 에지의 전체 길이를 정확히 예상하는 문제도 쉽지 않으므로 보통의 경우는 Half-Perimeter나 또는 2차원 평면상에서의 스패닝 트리를 measuraing function으로 대신한다[7,10,11]. 따라서 이 문제는 궁극적으로는 스타이너(Steiner) 트리를 구하는 문제로 귀납된다. 그런데 이 스타이너(Steiner) 트리를 찾는 문제는 그래프 버젼이나, 계산 기하학(Computational Geometry)에서도 NP-Complete이므로 그 해결을 어렵게 한다[30,31]. 그래프문제에서 스타이너 트



(그림 3-1) 4점 {a, b, c, d}의 배선길이를 예상하기 위한 다양한 측정함수들

리를 구하는 문제는 다음과 같다.

[그래프에서 스타이너(Steiner) 트리][31] 그래프 G (V, E)가 있고, 각 에지는 양수의 가중치가 지정되어 있다. 이때 R⊆V인 부분집합 R과, 정수 한계치 B가 있다. 그런데 G의 부분 트리(Subtree) 이면서 R의 모든 버텍스를 포함하고 그 트리의 가중치의 합이 B를 넘지 않는 트리가 있는가?

이 문제는 모든 가중치가 같은 값 일때나, 또한 G가 이분그래프일 때에도 NP-Complete로 남는다. 다음은 기하학적인 스타이너 트리를 제시한다.

[기하학적인 스타이너 트리(Geometric Steiner Tree)] 2차원의 정수 그리드 공간에 있는 집합 P와 양의 정수 K가 있다. 이때 P에 추가의 정수 그리드위의 집합 Q를 첨가하여 스패닝 트리를 구성하는데 그 가중치의 합이 K 이하인 트리가 존재하는가? 단 거리는 Discretized Eucledian Metric으로 측정한다[32].

이 문제는 NP-Complete이며[5] 그 거리를 보통의 Eucledian Metric으로 측정하면 NP-Hard이라고 믿어지고 있다[32]. 거리를 rectilinear, 즉 L_1으로 측정하여도 마찬가지이다. 기하학적인 스타이너 트리는 실제적이 회로설계 문제에 응용되므로 많이 연구되어 왔다[5]. 그리고 몇 가지 극히 제한된 경우에는 polynomial 시간에 풀림이 증명되어 있다. 그림 3-1에는 스타이너 트리, 스패닝 트리, Half-Perimeter 방식으로 측정한 한 네트의 연결 상태가 나타나 있다. 그런데 보통의 집적 회로에서 대부분의 신호선들이 5개보다 적은 터미널을 가지고 있다는 가정에서, 즉 입력의 크기가 작을 경우에 이 문제는 완전 탐색(exhaustive search)에 의해서 쉽게 polynomial시간 안에 해결된다.

3.1 그래프 분할문제

그래프 분할 문제[Minimum Cut Graph Partition Problem]: 단순 그래프 G(V, E)가 있고 각 에지에는 양수인 가중치가 있다. G의 V를 크기가 B 이하인 두개의 부분집합 V1 과 V2로 나눈다. 단 V1과 V2의 합집합은 원래 그래프의 V와 같아야 한다. 이 때, V1과 V2에 걸쳐있는 에지, 즉 (x, y), x∈V1, y∈V2의 weight의 합이 k 이하인 분할이 있는가?

이 문제는 전형적인 NP-Complete문제이다[2,32]. 그리고 이 문제의 제약된 형태로서 B=|V|/2인 경우 즉 같은 버텍스수를 가지도록 V를 균등하게 나누는 문제 역시 NP-Complete이다. 또한 모든 에지의 weight가 1인 제한된 경우나, 위 문제의 역인 최대 분할(Maximum Cut Partition) 역시 NP-Complete로 남아 있다[32].

최근에는 좀 더 복잡한 형태의 분할 문제인 비율 분할문제(Ratio Minimum Cut)가 있다. 비율분할 컷이란 분할의 컷의 값이 나누어진 분할의 크기를 반영하도록 Cut(V1, V2)=(V1과 V2를 연결하는 에지의 가중치의 합)/(|V1|*|V2|)로 설정한 것이다. 이와같은 비선형적인 목적함수는 실용적이긴 하나 최적해를 구해주는 데에는 더욱 어려움을 가중시킨다. 이런 직관적인 관점에서도 비율분할 문제는 NP-Complete라고 생각되고 최근에는 그것이 직접 증명되었다. 비율분할은 실제 초집적회로를 설계하는데 여러가지 의미를 지닌다. 이 비율분할은 배치해야할 면적이 직사각형이 아닌 일반적인 사각형일 경우를 고려해서 한곳에 배선이 너무 밀집하지 않도록 하기 위한 것이다.

분할을 위한 휴리스틱은 일찌기 많은 사람들로부터 연구되어 왔다. 그 중에서도 Kernighan-Lin의 알고리즘이 보편적으로 많이 사용되고 있다[6]. 그들의 알고리즘은 어느 정도 Local Minima에서부터 벗어날 수도 있고, 평균 시간 복잡도도 $O(n^3)$인데 비하여 결과도 비교적 좋으므로 쓸만하다고 할것이다. 그리고 Fiduccia에 의하여 O(n)인 선형시간에 해결되는 분할 알고리즘도 소개되었다[34]. 그들의 알고리즘은 각 에지의 가중치가 그다지 크지 않은 가중치일 경우에 bucketing과 같은 자료구조를 이용하여 시간을 줄였다. 따라서 아주 일반적인, 제약이 없는 그래프의 경우에는 그들의 알고리즘은 선형시간을 보장하지 못하는 단점이 있다. 이외에도 다양한 분할 방법이 있고, 최근 들어서는 Simulated Annealing방식까지 회로 분할에 쓰이긴 하지만 기존의 Kernighan-Lin 알고리즘이나 훌륭한 휴리스틱 방식에 비해서 큰 성능차이를 보이지 못하고 있다[27].

3.2 그래프 정열 문제

그래프 정열문제는 여러가지 형태가 있는데 그 목적함수에 따라서 각기 이름이 다르다[29]. 실제 집적회로 설계시의 배치 문제는 2차원 상에서 모듈을 위치지우는 것이지만 그의 한 제약형태인 일차원의 선형 위치로 제한하여도 많은 문제가 NP-Complete로 남게 된다. 따라서 평면 상에 그래프를 고정시키는 문제 역시 대부분 NP-Complete로 남게 된다. 이제 각 문제를 하나씩 살펴보기로 하자.

[밴드너비 최소화 문제 Bandwidth Minimization] 그래프 G(V, E)와 정수 K≤|V|가 있을 때 f: V→{1, 2, 3, …|V|}이면서 모든 에지 (u, v)에 대해서 |f(u)−f(v)|≤K가 되도록 하는 일대일 함수 f가 존재하는가? [32]

위 문제는 전형적인 NP-Complete이다[18,32]. G(V, E)를 인접행렬(Adjacency Matrix)로 바꾸어서 행, 렬 교환(Column, Row Operation)을 했을 때, 그 행렬의 밴드, 즉 1로 채워진 대각선 방향의 두께를 줄이는 문제와 동일하므로 밴드너비 최소화 문제라는 이름을 가지고 있다. 이 문제는 주어진 그래프가 방향그래프일 때는 물론이거니와, 심지어는 트리일 때에도 NP-Complete이다. 더 나아가 그 트리의 각 버텍스의 분기수(Degree)가 3을 넘지 않다는 조건을 가지더라도 NP-Complete로 남아 있을 정도로 해결하기 힘든 문제이다. 집적회로 설계시의 한 회로의 밴드 너비는 그 회로의 최대지연시간(Maximum Delay Time)과 동일하므로 큰 의미를 가진다고 하겠다. 즉 시스템 클럭의 빠를 수 있는 정도는 가장 늦게 전달될 신호의 정도에 달려 있으므로 되도록이면 가장 긴 배선의 길이를 짧게 할 필요가 있으며 그 길이는 바로 그래프의 밴드 넓이와 같다.

[최소절단 선형정렬 Min Cut Linear Arrangement] 그래프 G(V, E)와 양의 정수 K가 있다. 이때 다음과 같은 일대일 함수가 존재하는가? f: →V {1, 2, 3, …|V|}이며 모든 i에 대하여 1<i<|V|, |{(u, v)⊆E: f(u)≤i<f(v)}|≤K.

이 문제는 그래프가 트리일 경우에도 NP-Complete로 남는다[32,33]. 최소절단의 의미는 각 버텍스를 일렬로 늘어 놓았을 때, 그들을 연결하는 선의 최대두께를 최소화 하는 것이고 이는 결국 배선 문제중에서도 단층열 배선문제(Single Row Routing Problem)와 직접적인 연관을 가지고 있는 것이다. 그리고 외부평면 그래프(outerplanar graph)에서는 divide-conquer를 이용하여 이 문제를 polynomial 시간에 해결할 수 있음을 보였다.[29]

[최적 선형 정열 Optimal Linear Arrangement] 그래프 G(V, E)와 정수 K가 있다. 이때, f: V→{1, 2, 3, …|V|}이며 그 모든 에지의 거리의 합, Σ|f(u)−f(v)|≤K인 함수 f가 존재하는가?

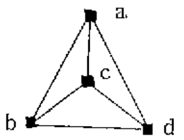
위 문제의 목적함수가 집적회로 설계시에 가지는 의미는 회선 길이의 총합이라고 하겠다. 회로설계시에 전체 회선길이를 짧게 하는 것은 먼저 배선단계의 문제를 단순하게 만든다는 것과 이와 더불어 회선을 통한 신호의 전달과정에서 나타나는 열 발생율(power dissipation)의 감소, 그리고 신호의 전달시간 단축을 가능케 하는 등 여러가지 장점이 있으므로 대개의 설계 자동화 단계에서 배선길이를 줄이려는 노력이 가장 보편적이라 하겠다. 이 문제에서 그래프를 이분 그래프로 제한하더라도 역시 NP-Complete로 남아있게 된다[32]. 그러나 한 가지 특기할 만한 사실은 G가 트리일 경우에는 polynomial 시간에 해결된다. 역시 방향 그래프(Directed Graph)에서도 NP-Complete로 남아 있음을 볼 수 있다[32].

이상에서 제시한 배치문제가 대부분 NP-Complete이므로 2차원 배치를 위한 많은 문제들은 위한 휴리스틱 알고리즘이 많이 개발되었다. 그중에서 대표적인 것은 Min-Cut을 배치문제에 적용한 것[2], 선형계획으로 수식화하여 해결하는 것, Steinberg Assignment, Force-Directed Relaxation, Neighborhood Exchange 등이 있고 이들을 적당히 조합하여 해결한 여러 방식들이 있다[16].

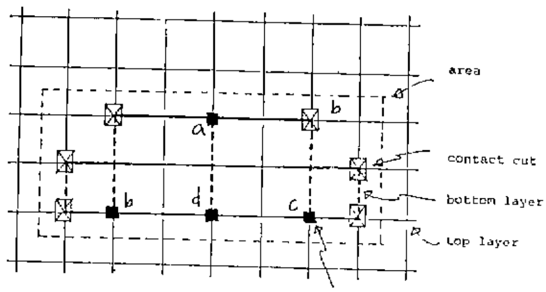
3.3 그래프 설치(Layout) 문제

앞서 소개한 그래프의 정열 문제가 일차원적인 문제라면 그래프의 레이아웃(Layout)을 구하는 문제는 그래프 정열의 2차원적인 문제라고 하겠다[1]. 그래프의 레이아웃이란 주어진 그래프를 그리드 그래프(grid graph)에 배치하는 문제이다. 이 그래프는 일명 톰슨(Thomson) 모델이라고도 불린다. 그리고 많은 집적회로에서의 수학적인 모델은 톰슨의 모델을 사용한다[12]. 그 특징은 다음과 같다.

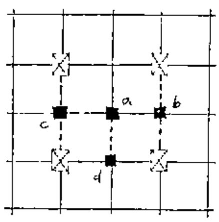
먼저 칩의 그리드는 등간격(evenly spaced)의 수직 수평으로 나누어져 있고, 그 수평선분을 보통 트랙(track)이라고 부른다. 그래프의 버텍스에 해당되는 능동소자, 예를 들어 트랜지스터는 그리드의 교점에 놓인다. 배선은 두개의 층(layer)에 그려지게 되는데 수직 선분은 윗층(top layer), 수평선분은 아랫층(bottom layer)에 놓이는



circuit G



layout A of G



layout B of G

(그림 3-3-1)

것이 보통이며 서로 같은 방향으로 선분이 두층 사이에서 겹칠수 없다. 그 이유는 같이 겹치게 되는 경우 전자파에 의한 간섭 현상으로 신호의 왜곡이 나타나기 때문이다. 그리고 via라고 불리는 contact cut이 존재해서 아레 윗층을 연결시켜 준다. 이러한 배선 방식은 일명 멘하탄 배선(Manhattan routing)이라고도 불린다. 그림 3-3-1에 톰슨의 모델, 그리고 한 회로와 그를 위한 두가지의 상이한 레이아웃 A와 B가 있다. 위와 같은 레이아웃에서 고려해야 할 측정치(Measure)는 다음과 같다[1].

[레이 아웃 면적 문제] 주어진 그래프를 그리드 그래프에 위치 시키는데 그 그래프가 차지하는 면적이 최소로 되게 하라.

레이 아웃 면적은 각 버텍스와 에지를 포함하는 최소의 사각형으로 정의된다. 위의 예에서 A는 21이고 B는 9이다. 이것을 최소화 시키려는 작업은 보통의 집적회로의 목표로 매우 자연스런 것이다. 왜냐하면 적은 면적의 칩은 공정면에서나 가격면에서 유리하기 때문이다. Dolev, Leighton과 Trickey는 트리의 집합인 숲(forest)를 최적의 면적에 위치시키는 문제도 역시 NP-Complete였음을 밝혔다[8]. 그리고 그 하한치와 상한치에 관한 많은 연구가 지금까지 진행되어 왔다[1].

[Via(Contact Cut)수 최소화 문제]

via수 관점에서 볼 때, A는 6이며 B는 4이다. 보통 칩 설계시 via는 선보다 많은 자리(약 1.5배 이상)를 차지하며 전선의 캐퍼시턴스(capacitance)를 증가시키는 단점이 있으므로 최소화 할수록 유리하다.

[배선 크로싱 최소 문제]

이 문제는 그래프 이론적 관점에서 보면 에지 크로스(Edge Cross)최소화 문제이다[13]. 에지 크로스는 병렬계산시, cross-talking이라는 현상이 발생하게 된다. 또한 많은 수의 크로싱은 공정과정을 어렵게 만들기도 한다. 이 문제의 본류는 주어진 그래프를 평면상에 그리는데 에지가 서로 겹치게되는 횟수를 가장 작게 하는 문제와 같은 류이다. 최근 들어 관심을 받고 있는 문제가 바로 에지 크로스를 최소화 시키는 문제이다. 이 문제는 설계 자동화 뿐만이 아니라, 사용자 인터페이스 등의 문제에 자주 응용되고 있다. 즉 그래프 구조를 가지는 여러가지 오브젝트들, 예를 들어 프로그램의 흐름도, Make file과 같이 복잡하게 얽힌 화일 시스템의 구조를 일목요연하게 보여주는 일까지 다양하게 쓰이고 있다[15,26].

이런 경우 항상 그런 것은 아니지만 일반적으로 사람에게는 에지 크로스가 없는 것이 더욱 단순해 보이므로 같은 그래프를 그리더라도 에지 크로스가 없는 것이 더 선호된다고 하겠다. 이와 관련된 많은 문제들이 새로이 발견되고 그들 대부분이 NP-Complete문제라고 밝혀지고 있다[30,31,32]. 비교적 단순한 형태라고 보여지는 이분 그래프(Bipartite Graph)의 최소 에지 크로스의 수를 구하는 문제는 Garey와 Jonshon에 의해서 NP-Complete 임이 밝혀져 있다[32]. 그리고 아직도 완전 그래프와 완전 이분 그래프의 최소 에지 크로스의 수도 그 한계값만 구해져 있을 뿐 정확한 값은 밝혀져 있지 않다. 완전 그래프의 에지 크로싱 값 v(G)는 다음과 같다.[13]

$$v(K_p) \leq 1/4[p/2][(p-1)/2][(p-2)/2][(p-3)/2]$$

$$v(K_{m,n}) \leq [m/2][(m-1)/2][n/2][(n-1)/2],$$

위 식에서 [ ]기호는 floor 함수이다. 즉 [2.3]=2, [4.9]=4로 계산된다 그리고 오랫동안 Hypercube의 크로싱 수는 1970년, Eggleton, Guy에 의해서[17]

$$v(Qn) < 0.15625\ 4^n - [(n^2+1)/2]2^{n-2}$$

로 알려져 있었으나 최근 한 연구에 의해서 그 한계치가

크게 개선되었고, 이는 상당히 의미있는 일이라 하겠다. 그 연구에 의하면 Hypercube Qn의 크로싱 한계는 다음과 같다[17].

$$\Omega(n(\log\ n)^{c\log\log\ N}) \leq v(Qn) \leq 1/6\ n^2$$

이 한계치는 Subcube와 homeomorphic한 family를 단계적으로 배치하는 방법으로 얻어진 것이다[17]. 그러나 그래프를 사람에게 알기 쉽도록 그리는 문제는 단순히 에지 크로싱만을 최소화 시키는 일만이 최선은 아니고 도리어 적당한 크로싱을 허용하더라도 대칭성과 같은 모양의 부그래프가 규칙적으로 드러나도록 하는일이 더 중요하다. 이 문제는 궁극적으로 복잡한 군론(Group)의 문제로 귀납되므로 아주 어려운 계산문제라고 하겠다[15,26].

[최대배선 길이 최소화 문제] 주어진 그래프를 그리드 그래프에 놓을 때 가장 긴 에지의 길이가 최소가 되게 하라.

밴드너비 문제와 마찬가지로 긴 배선은 신호 전달을 느리게 한다. 전체 배선의 길이의 제곱은 전체 신호의 전달 시간과 비례하며 저항과 캐퍼시턴스의 곱에 비례한다. 즉 Delay ∝ Registance ∝ L*L이다.

이와 같은 톰슨의 모델은 간편하고 그래프 이론적인 전개를 가능케 한다는 점에서 좋지만 몇 가지의 단점도 가지고 있다. 예를 들어 Power나 I/O 패드, 그라운드 신호선을 처리하기에는 문제가 있고 버텍스의 사이즈가 그리드의 교점에 비해서 좀 크다는 단점도 있다. 레이아웃에 관한 많은 문제가 NP-Complete이므로 대개는 분할-정복(Divide-Conquer)방식에 의한 휴리스틱 방식으로 해결한다. 그것의 기본이 되는 이론은 평면 분할자 정리(Planar Separator Theorem)라는 유명한 정리에 주로 의존해서 전개되고 분석된다. 먼저 그래프의 분할자(Separator)를 정의해 보면 다음과 같다.

[정의] (1) 어떤 그래프 G(V, E)의 k-분할자(separator)는 집합 U⊆V이며 다음과 같은 조건을 만족시킨다. G에서 U를 제거하면 G는 각각 최대 k개의 버텍스를 가지는 A와 B로 분리되고 A와 B 사이에는 에지가 전혀 없다. (2) 어떤 그래프 G(V, E)의 k-에지 분할자(edge separator)는 에지 집합 F⊆E이며 다음과 같은 조건을 만족시킨다. G에서 F를 제거하면 G는 각각 최대 k개의 버텍스를 가지는 A와 B로 분리되고 A와 B 사이에는 에지가 없게 된다.

평면(planar) 분할자는 분할자가 평면 그래프에 적용될 때이다. 특정한 그래프에 대해서는 분할자가 알려져 있다. 트리일 경우에는 1-분할자이며[35]. n개의 버텍스를 가진 평면 그래프의 분할자는 $O(N^{1/2})$이다. 그리고 Leiserson에 의해서 아래의 사실이 증명되었다[1,8].

x^a-분할자 정리	배치 면적의 상한치
$a<1/2$	$O(N)$
$a=1/2$	$O(N\log^2\ N)$
$a>1/2$	$O(N^{2a})$

그러나 위의 한계치를 가지는 그래프는 존재하지만 [1], 일반적으로 모든 그래프에 관해서는 실제 개별적인 레이아웃 면적과 위에서 보인 한계치와는 큰 차이를 보이고 있다. 예를 들어 N개의 노드를 가진 그리드인 메쉬 형태의 그래프를 생각해보면 그 면적이 O(N) 임을 당연하다. 그러나 위 Leiserson의 한계치에 의하면 그리드 그래프가 $N^{1/2}$의 분할자를 가지므로 $O(N\log^2\ N)$의 한계치를 보이고 이는 최적의 값과 큰 차이를 보인다. 그리고 분할자가 클수록 이론치와 실제값은 큰 차이를 보이고 있다. 이 문제는 그래프 클래스를 세분하여 각 클래스마다 합당한 분할자를 찾아내는 과정이 필요할 것이다[1].

평면 분할자 정리가 말하는 것은 평면 그래프는 상당히 적은 크기의 분할자를 가지고 있다는 것이다. 예를 들어 모든 k에 대해서 크기가 $O((n/\sqrt{k})\log^{3/2}\ k)$인 k-분할자가 있음이 밝혀졌다. 그리고 Lipton과 Tarjan은 평면 그래프에서는 항상 $\sqrt{8n}$ 보다 작은 2n/3-분할자가 있음을 보였다. 그리고 이것은 선형 시간에 발견되는 특징이 있다. 그리고 Dijidev는 6n보다 적은 크기의 2n/3-분할자가 있음을 보였다. 나아가서 평면 그래프에는 항상 $4\sqrt{\varepsilon/n}$ 크기인 평면 εn-분할자를 선형시간내에 찾을 수 있음을 밝혔다[1]. 이후에도 분한자에 관한 많은 연구가 있다.

이러한 분할자나 bifurcator를 이용하면 문제를 재귀적으로(recursive) 해결하려 할 때 각 문제의 특성을 유지시킬수가 있으므로 일관된 방법의 순차적인 사용을 가능하게 한다. 즉 부그래프(subgraph)로 내려감에 따라서 각 문제가 같은 방식으로 해결된다. 이 사실을 이용해서 우리는 회로의 배치 면적에 관해서는 주요한 한 사실을 알 수 있다. 그것은 한 그래프의 분할너비(Bisection width)가 B일 때, 그 넓이 A는 $A>\Omega(B^2)$가 된다는 사실이다. 그리고 면적이 A이고 시간이 T이라면 N개의 변수를 가진 trnasitive 함수는 $AT^2 \geq \Omega(N^2)$임이 알려져 있다[12].

IV. 회로배선에 관한 이론적 문제

배치가 끝난 단계에서 배선은 각 모듈의 터미널을

디자인 룰(Design Rule)을 지키면서 연결하는 것이다. 즉 이것은 각 에지를 그리드 스페이스 상에서 어떻게 연결시켜주어야 하는가의 문제이다. 배선은 전체배선(Global Routing)과 상세배선(Detailed)으로 나누어 지는데 이 둘은 수학적으로는 동등하다. 전체 배선은 각 선들이 지나갈 길을 대강 정해두는 것이다. 이것의 목적은 한 쪽으로 너무 많은 선이 지나가므로 밀집된 선이 허용된 트렉의 용량을 넘어서게 되는 것을 미리 어느 정도 방지하기 위해서이다. 그리고 이어지는 상세배선은 실제 각 선이 어느 채널의 어떤 트렉위로 달려야하고 어디에 via를 뚫어야 하는지를 정하는 것이다. 그러므로 배치가 최적이라고 해도 전체배선이 완성되는 것이 아니며, 전체배선이 최적이라고해도 상세배선이 완성되어 칩의 설계가 끝나는 것이 아니다. 만일 각 단계에서 실패한다면 그 이전의 단계에서부터 새로 해야만하는 어려움이 있다. 이제 우리는 각 단계에서의 문제가 그래프 알고리즘에서는 어떤 형태로 바뀌어 지는지를 살펴보기로 하자.

4.1 전체배선의 이론적 특성

수학적으로, 전체배선에서 셀 안쪽의 배선 방향에 대해 고려하지 않는다는 점을 제외하면, 전체배선과 상세배선는 근본적으로 같다. 일단 이단 터미날 네트일 경우에는 아래와 같은 최단거리 문제를 해결해야 한다. 최단거리 문제는 다음과 같다. 그래프 G(V, E)를 고려해보자. 각각의 에지 e_{ij}는 길이 d_{ij}를 갖는 데. 이것은 배선될 때의 경비를 나타낸다.

$(v_s, e_{si}, v_i, e_{ij}, v_j, \cdots, v_t)$는 v_s에서 v_t로 가는 경로라고 한다. 경로의 길이는 경로 상에 있는 모든 간선의 길이 합이다. 최소 경로 문제는 "임의의 주어진 두 정점을 연결하는 최소 길이를 갖는 경로를 찾는 문제"이다. 보통 v_i에서 v_j로 가는 경로가 존재하지 않으면 $d_{ij}=\infty$라 두고 존재할 때는 $d_{ij}=0$라 둔다. 다음 3가지를 가정하면, 1) 모든 d_{ij}는 양수이고 2) 간선들은 방향을 가지며 3) 많은 간선을 가지는 경로는 작은 간선을 가지는 경로보다 더 짧은 거리를 가질 수 있다. 이것을 다음과 같은 조건으로 표기할 수 있다.

1) $d_{ij}>0$ 모든 i, j에 대해
2) $d_{ij}\neq d_{ji}$ 어떤 i, j에 대해
3) $d_{ij}+d_{jk}<d_{ik}$ 어떤 i, j, k에 대해

여기서 조건 3)를 허용하지 않는다면 이 문제를 매우 쉽다. 이 세 조건하에서 Dijkstra는 처음으로 최소경로 알고리즘을 개발했다. 이 알고리즘은 임의로 주어진 v_s에서 다른 v_t로 가는 최소 경로를 정확히 찾지는 못한다. 그것은 v_s에서 네트워크 내에 있는 다른 모든 정점으로 가는 최소경로를 제공해 준다. 그래서 이 알고리즘을 single-source 최단거리 알고리즘이라 한다.

네트워크내에서 모든 정점간의 경로 중 최소 경로를 찾기 위해서는 $O(n^3)$ 시간을 갖는 Warshall-Floyd의 알고리즘을 사용할 수 있다. 이 알고리즘은 네트워크내에 음수 길이를 갖는 사이클이 존재하지 않는 한 $d_{ij}>0$인 모든 것이 다 필요하지는 않다. 거대한 네트워크에 대해서는 분리 알고리즘을 사용할 수 있는데, 이는 네트워크를 한번에 처리할 수 있는 부분으로 분리하여 처리한다. 대부분 single-source 최단거리 알고리즘은 기본적으로 네트워크내의 v_s에서 다른 모든 버텍스에로의 루트있는 트리를 구축한다. 트리내의 모든 노드는 루트로부터 유일한 경로를 가지고, 그 유일한 경로는 루트로부터 최소 경로가 된다. 이 문제를 일반적으로 바꾸어보면 여러 개의 터미날 쌍을 동시에 연결하되 서로 그 경로가 겹치지 않는 안전한 길을 찾는 문제로 볼 수 있다. 이것의 기본적인 그래프 문제는 다음과 같다.

[독립 연결 경로문제(disjoint connecting path problem)] 그래프 G(V, E)에서 서로 독립된 k개의 버텍스 쌍을 생각해보자. 즉 (s1, t1), (s2, t2), ⋯, (sk, tk)가 있을때, 각 쌍들의 버텍스를 서로 연결하는 서로 독립적인 경로가 k개 존재하는가?

이 문제는 주어진 그래프가 평면 그래프라고 하여도 역시 NP-Complete로 남는다[32]. 그런 한가지 특기할 만한 사실은 G가 평면 그래프이거나 코덜(chordal)그래프에서 k=2인 문제는 polynomial time에 해결된다. 그리고 평면 방향 그래프(directed planar graph)이거나, 반회로 그래프(acyclic graph) 일 때에도 역시 polynomial 시간에 문제가 해결된다. 이보다 좀더 제한적인 형태로서는 다음과 같은 문제가 있으며 이 역시 NP-Complete문제이다[32].

[최대 길이 제한 독립 경로(Maximum length-bounded disjoint path problem)] 그래프 G(V, E)에 특정한 버텍스 s와 t가 있고, 정수 J, K<|V|가 있다. 이때 s에서 t에 이르는 K개 이하의 에지를 지나는 버텍스 독립인 길이 J개 존재하는가?

배선 문제가 실제 회로에서 나타나는 경우는 채널배선, 강둑배선(river routing)[9], 스위치박스 배선 등의 문제가 있다. 이중 가장 간단한 채널 배선에서 최소의 트랙으로 배선을 완성하는 문제가 역시 NP-Complete로 남아 있다. 그런데 정수 선형 프로그램(Integer Linear Programming)을 이용하여 전체배선을 해결하는 방법도 있다.

앞에서 설명하였듯이 네트를 연결하는 데는 많은 방법이 있는데, 각각의 연결은 네트안의 주어진 터미널을 연결하는 Steiner 트리이다. 네트에 연결되는 트리에 변수 y_j를 부여한다. 특별한 트리가 사용되면 y_j를 1이라 두고, 사용되지 않으면 0이라 둔다. 예를 들어, 첫 네트에 3가지 연결이 있고 두번째 네트에 5가지 연결이 있으면 다음과 같이 표기할 수 있다.

$$y_1+y_2+y_3=1$$
$$y_4+y_5+y_6+y_7+y_8=1$$

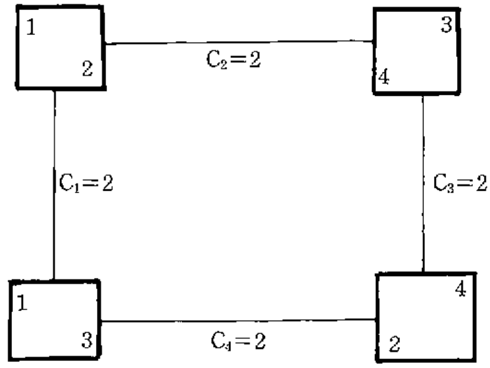
단 하나의 트리에 주어진 네트에 연결되어질 필요가 있기 때문에 각 네트에 대해서 하나의 방정식이 존재한다. 따라서 주어진 네트에 여러 가지 다른 연결이 존재하는 경우에 대해 살펴보자.

원칙적으로 주어진 행렬 $[a_{ij}]$에 (0, 1)로써 가능한 모든 방법을 표현할 수 있다. m개의 에지를 가지는 그리드 그래프에 대해, i번째 행은 i번째 에지에 해당되고 각 열은 주어진 네트에 연결되는 서로 다른 방법이 되는, $[a_{ij}]$안에 m행이 될 것이다. 만약 1000개의 네트와 각 네트에 10개의 다른 연결이 있다면, 행렬 내에 만개의 열이 존재하고 실제 모든 네트를 연결하는 가능한 에지의 개수를 헤아릴 수 없다. 그것은 선형 점수 프로그램을 해결하기 위하여 행렬의 모든 원소를 다 알 필요는 없기 때문이다.

어떤 열 j에 i번째 값 a_{ij}는 특별한 연결에서 i번째 에지를 사용한다면 1이 되고 아니면 0이 된다. 어떤 에지에서 많은 선은 위와 같이 제한되어야 하므로 에지의 용량은 다음과 같이 표기할 수 있다.

$$\Sigma a_{ij} y_j \leq c_i \qquad (i=1, \cdots, m;\ j=1)$$

여기서 c_i는 i번째 에지의 용량이다. 그림 4-1-1에서 4개의 정점과 4개의 간선, $c_i=2(i=1, \cdots, 4)$인 그리드 그래프를 나타내고 있다. 여기서 4개의 네트는 연결되어 있고 모든 네트들은 2개의 터미널을 가진다. 각 네트는 두 가지의 다른 방법으로 연결될 수 있고 네트 j에 연결될 수 있는 두 가지 다른 방법을 y_i와 y_j'으로 표현할 수 있을 것이다. 이 그림 4-1-1에 있듯이 8가지의 서로 다른 연결을 다음 (0, 1)행렬에서 보여 주고 있다. p개의 네트가 있고 그 네트에 연결될 수 있는 방법이 n가지라 가정하자. N_k에 k번째 연결되는 여러 가지 방법이 y_j들의 집합으로 나타날 것이다. 이렇게 전체 배선을 모델링하는 것은 배선의 문제를 완벽히 기술할 수 있다는 장점이 있지만 선형계획 자체가 NP-Complete 문제인 단점을 가진다. 특별한 경우를 제외하고는 polynomial 시간에



	y_1	y_1'	y_2	y_2'	y_3	y_3'	y_4	y_4'
c_1	1	0	0	1	0	1	0	1
c_2	0	1	1	0	0	1	0	1
c_3	0	1	1	0	1	0	1	0
c_4	0	1	0	1	1	0	0	1

(그림 4-1-1) 정수계획법을 이용한 전체배선 과정

NP-Complete문제를 푸는 것과 다르기 때문에 이 문제를 최적의 해에 근접하는 polynomial 시간을 갖는 휴리스틱한 알고리즘을 구하기 위한 연구가 진행되어 왔다.

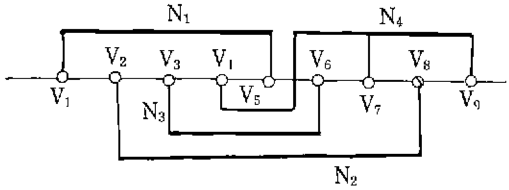
4.2 상세배선의 이론적 특성

먼저 단일층 배선 문제를 고려해보자. 단일층 배선문제는 배선문제 중에서 가장 단순한 형태이다. 그런데 이것 역시 많은 제약조건이 있다고 하여도 NP-Complete로 남게 되므로 우리가 일반적으로 고려해야 할 많은 실제적인 문제들이 얼마나 까다로운 문제인가하는 것을 알려 준다고 할 것이다[31,32].

[단일층 배선문제(Single Row Routing Problem)] V={1, 2, 3, …, n}인 버텍스 집합이 수직선 상에 균등하게 나열되어 있다. 그리고 네트의 집합 L={N1, N2, N3, …, Nm}이 있는데 각 네트는 전위적으로 같이 연결되어야 할 버텍스를 가지고 있다. 이들을 단일층을 가진 기판위에 수직 수평 방향으로 서로 겹치게 하지않고 배선시키려 한다. 그 가능한 경로가 존재하는가?

그림 4.2.1에 한 단일층 배선의 예가 있다. 이 문제는 NP-Hard로 알려져 있으나 많은 휴리스틱 알고리즘이 알려져 있다[7,25].

[그리드상에서 평면 연결 문제] 정수 좌표 그리드 상의 연결되지 않은 점들의 쌍의 집합 P=[(x_1, y_1), …,



$N_1 = \{V_1,\ V_5\}$

$N_2 = \{V_2,\ V_8\}$

$N_3 = \{V_3,\ V_6\}$

$N_4 = \{V_4,\ V_7,\ V_9\}$

(그림 4-2-1) 상층에는 1개의 트랙, 하층에는 3개의 트랙을 사용하여 single Row 배선이 끝난 상태

$(x_n,\ y_n)$]이 주어졌을 때, 그리드 세그먼트로 구성된 경로에 의해 두 경로가 연결되지 않도록 P의 그리드 점들의 각 쌍을 연결하는 방법이 있는가?

이 문제는 Raghavan 등에 의해서 각각 NP-complete 임이 증명되어졌다. 그런데 수직, 수평의 연결이 허용되거나 또는 연결하는 경로가 수평 방향으로 단조증가 또는 단조감소가 되더라도 여전히 NP-Complete이다.

[2-계층 다중 모듈 배선 문제] 다음의 (i), (ii), (iii)의 3조건이 주어져 있다. (i) 연결되지 않은 모듈들의 집합 [M_1, …, M_n] 사각형들은 4개의 모서리의 정수 좌표에 의해 표시된다. (ii) 네트 N_1, …, N_m으로 분리되는 터미날들의 집합 P. (iii) 정수의 한계치 B가 있을 때, 영역 B의 사각형 R내에서 네트에 대한 요구된 계층의 배선이 가능한가? [32]

이 문제는 다음과 같다. (i) 스타이너 트리의 어떤 그리드 세그먼트도 그 모듈에 포함되지 않도록, (ii) 스타이너 트리의 모든 모듈과 세그먼트가 사각형 R에 포함된다, (iii) 직교하지 않는 연결을 가지는 두 트리는 없다, 의 조건을 만족시키는 직교 스타이너 트리에 의해 각 네트에 있는 핀을 연결하기 위한 것이다. 이 문제는 Szmanski에 의해 NP-complete 임이 증명되었다. 단 2개 모듈이 있고 각 네트는 단 2개 terminal들을 포함한다 하더라도 NP-Complete로 남는다. 그리고 휴리스틱 알고리즘과 특별한 경우의 polynomial 시간의 알고리즘도 있다.

[2-계층 채널배선 문제] 아래 (i), (ii)의 두 조건이 주어져 있다. (i) 직선 $y=0$과 $y=W$에 한계를 가지는 사각영역으로 생각될 수 있는 채널 (ii) 네트 N_1, …, N_m으로 분리되는 그리드 점들의 집합 $P=[(a_i,\ W)$: $1\le i\le p] \cup [(b_j,\ 0)$: $1\le j\le q]$이 있다. 이 때 그 네트에 대한 2-계층 채널 배선이 가능한가?

이 문제는 다음과 같다. (i) 스타이너 트리의 모든 그리드 세그먼트는 $y>0$, $y<W$인 영역에 존재한다. (ii) 직교연결이 아닌 연결을 가지는 2개의 트리는 없다. (i), (ii)의 조건을 만족하는 직교 스타이너에 의해 각 네트에서 핀들을 연결하는 문제이다. 이 문제 역시 NP-complete임이 증명되었다. 모든 네트가 $(a_i,\ W)$ 형태의 한 점과 $(b_i,\ 0)$ 형태의 한 점인 단 두 점을 구성하더라도 역시 NP-Complete로 남는다. 심지어 모든 점이 2-터미날 네트이고 T_i의 각 네트가 하나의 수평선 세그먼트를 포함하도록 허용하더라도 여전히 NP-Complete이다.

[3-계층 채널배선 문제] 채널 라우팅 예에서 2-터미날 네트를 연결하는 에지-비연결 경로의 전체가 주어졌을 때, 분명한 S-계층 Knock-Knee 배선이 얻어지도록 경로 세그먼트에 계층들을 할당하는 방법이 있는가?

이 문제는 Lipski에 의해 NP-complete으로 증명되었다. 이상에서 살펴본 바와 같이 톰슨의 것과는 다른 다양한 배선 모델에 관하여 많은 문제가 NP-Complete이며 스위치 박스 문제 등도 NP-Complete이다[9]. 이외에도 via를 최소화한 배선, 최대 배선길이를 최소화 시키는 문제들도 어려운 문제이다. 최근에는 가장 보편적으로 쓰이는 표준배선 문제를 최적의 결과로 보여주는 휴리스틱이 속속 발표되고 있다[3,14]. 그리고 TimberWolf와 같은 프로그램은 배선까지도 Simulated Annealing으로 해결한다.

4.3 그래프의 두께

최근에는 다층기판 기술이 발달함에 따라서 이전의 톰슨 모델에서 더욱 확장된 형태의 배선 기술이 필요하게 되었다 즉 에지 크로스를 몇개 이상 허용하는 다층에서 어떻게 배선하는가 하는 것이 큰 연구문제이다[24]. 그런데 이 문제의 한 단순한 형태인 주어진 그래프를 몇개의 층으로 배선하면 전혀 에지 크로스 없이 할 수 있는가는 결국 그래프의 두께(graph thickness)를 구하는 문제로 귀납된다고 볼 수 있다. 그래프의 두께에 관한 문제이다. 그래프의 두께란 주어진 그래프를 평면 그래프로 분리해내기 위한 최소의 갯수이다. 따라서 평면 그래프의 두께는 1이며, 5개의 비텍스를 가진 완전 그래프의 두께는 2이다. 이론적인 면에서 그래프의 두께를 알아내기란 쉽지 않으나 완전 그래프에 관해서는 그 한계치 th(G)가 밝혀져 있다. 그것은 다음과 같고 그것을 일반 그래프에서 찾아내는 것도 NP-Complete임이 밝혀

져 있다[13].

$th(K_p) > [(p(p-1)/2)/3(p-2)]$

그래프의 두께를 구하더라도 각 에지를 어떤 층에 어떻게 연결해야만 via 등을 최소화 시킬 수 있는지는 알 수 없으므로 그를 구하는 것도 한 연구 문제이다.[4]

V. 최근의 연구동향과 결론

최근의 연구동향으로서는 첫째, 배선시에 각 모듈들의 특성인 터미날의 위치가 어느 정도 자유롭다는 사실에서 보다 나은 배선 결과를 얻기위한 터미날 이동기법이 많이 연구되고 있다. 예를 들어 Lateral Shift 기법이라든지 [19], 표준셀에서 셀의 위치를 뒤집는 방법을 이용하면 추가의 배선상의 이득을 볼 수 있다. 또는 배치시에 각 터미날의 위치가 충분히 움직일 수 있다는 사실에서 배치초기부터 배선의 문제를 고려하는 것이다. 이는 궁극적으로 배치, 배선의 문제를 동시에 고려하는 것이다 [19-22].

둘째로는 병렬 알고리즘을 이용하여 문제를 해결하는 시도가 있다. 이는 그래프에서 병렬 알고리즘의 기초적인 성질이 규명되는 일과 같은 맥락에서 이해되어야 할 것이다. 마지막으로는 앞에서도 설명한 다층 기판내에서의 배선, 배치 문제를 들 수있다. 배치 문제는 결국 앞서 소개한 그래프의 두께와 밀접한 관계를 가진다. 그러나 다층기판의 더욱 일반화된 3차원 공간에서의 배선 문제는 아직 명확하게 밝혀진 실제적인 결과가 없으므로 많은 연구가 필요할 것이다. 즉 참고 문헌[30,31]에 나타나지 않은 여러 기초적인 그래프 문제의 복잡도에 관한 연구가 필요하다. 그리고 아직 많은 알고리즘은 입력 그래프가 단순 그래프의 형태임을 가정하고 있어서 실제 회로가 하이퍼 그래프라는 특성을 잘 살리지 못하고 있다. 앞으로도 이러한 하이퍼 그래프의 특성과 그에 따른 알고리즘을 개발하는 문제에 많은 연구가 필요할 것으로 본다.

우리는 이상에서 그래프 알고리즘적 문제들이 직접회로 설계에 어떻게 응용되는지를 살펴 보았다. 결론적으로 우리는 그래프상의 다양한 문제들이 어떻게 설계 자동화에 응용되고 수 많은 NP-Complete 문제들이 어떤 휴리스틱 등으로 해결되는지를 보았다. 이상의 사실에서 우리는 실제 그래프 알고리즘이 NP-Complete라고 해도 각 문제의 특성, 예를 들면 버텍스의 분기수가 5 이하라든지, 한 네트가 연결해야할 터미날의 수가 충분히 적다든지 할때, 최적이거나 최적에 매우 가까운 해를 구해 수 있음을 여러 앞선 연구에서 볼 수 있었다. 따라서 실제적인 많은 문제를 분리하고 그 제약조건에 대한 연구와 그래프 특성을 규명하는 연구가 유기적으로 결합될 때 공학적으로나, 이론적인 큰 발전을 기대할 수 있다고 할 것이다.

참 고 문 헌

1. S. N. Bhatt and F. T. Leighton, "A Framework for solving VLSI Graph Layout Problems," J. of Computer and System Sciences, Vol. 28, pp. 300~343, 1984.
2. M. A. Breuer, "Min-cut placement," J. Design Automation and Fault Tolerant Computing, Vol. 1, no. 4, pp. 343~362, 1977.
3. M. Burstein and P. Pelavin, "Hierarchical Channel Routing," IEEE Trans. on CAD, Vol. CAD-2, no. 2, pp. 156~163, 1984.
4. Y. K. Chen and M. L. Liu, "Three-Layer Channel Routing," IEEE Trans. on CAD. Vol. CAD-3, no. 2, pp. 156~163, 1984.
5. M. R. Garey and D. S. Johnson, "The Rectilinear Steiner Tree Problem is NP-Complete," SIAM J. Applied Mathmatics. Vol. 32, pp. 826~834, 1977.
6. B. W. Kernighan and S. Lin, "An Efficient Heuristic Procedure for Partitioning Graphs." bell System Technical J., Vol. 49, no. 2, pp. 291~307, 1970.
7. M. R. Karmer and J. van Leeuwen, "Wiring-Routing is NP-complete," Tech. Report RUU-CS-82-4, Dept. of Computer Science, Univ. of Utrecht, Utrecht, the Netherlands, 1982.
8. C. E. Leiserson, "Area-efficient VlSI Computation," ph.D Thesis, Carnegie-Mellon University, Pittsburg, PA, 1981.
9. C. E. Leiserson and R. Y. Pinter, "Optimal Placement for river routing," in VLSI systems and Computations, ed. H. T. Kung. R. Spoull and G. Steels, pp. 126~142, Computer Science Press. Rockville, Maryland, 1981.
10. J. MagGregor-Smith, D. T. Lee, and J. S. Leibman, "An O(n logn) Heuristic Algorithm for the Rectilinear Steiner Minimal Tree Problem," Engineering Optimization, Vol. 4, pp. 179~192. 1980.
11. M. Servit, "Heuristic Algorithms for Rectilinear Steiner Trees," Digital Processes. Vol. 7, pp 21~32, 1981.
12. C. D. Thompson, "A Complexity Theory for VLSI," ph.D. Thesis. Carnegie-Mellon University, Pittsbu-

rgh, PA., 1980.
13. F. Harary, Graph Theory, Addison-Wesley, Apr, 1971.
14. T. Yoshimura and E.Kuh. "Efficient Algorithms for Channel Routing," IEEE Trans. on CAD, Vol. CAD-1, no. 1, 1982.
15. E. B. Messinger, "Automatic Layout of Large Directed graphs," Technical Report, No. 88-07-08, July, 1988.
16. T. M. J. Fruchterman and E. M. Reingold, "Graph Drawing by Force-directed Placement," Software-Practice and experocnce, Vol. 21(11), pp 1129~1164, Nov., 1991.
17. T. Madej, "Bounds for the Crossing Number of the N-Cube," Journal of Graph Theory, Vol. 15, pp. 81~97, 1991.
18. P. Z. Chinn, J. Chvatalova, A. K. Dewdney and N. E. Gibbs, "The Bandwidth Problem for Graphs and Matrices-A Survey," Journal of Graph Theory, Vol. 6, pp. 223~254, 1982.
19. R. Tamassia and I. G. Tollis, "On Improving Channel Routability by Lateral Shifting of the Shores," SIGDA Newsletter, Vol. 18, No. 1, pp. 18~29.
20. Thang Nguyen Bui and Sing Ling Lee, "On the Mincut Bipartite Arrangement Problem," The Pennsylvania State Univ., CS-87-15, April.
21. Masaki Yamada and C. L. Liu, "An Analytical Method for Optimal Module Orientation," ISCAS '88, pp. 1679~1682, 1988.
22. Sartaj Sahni and San-Yuan Wu, "Two Np-hard Interchangeable Terminal Problems," IEEE Trans. on CAD. Vol. 7, No. 4, April, 1988.
23. Xianjin Yao, Massaki Yamada, and C. L. Liu, "A New Approach to thr Pin Assignment Problem," IEEE Trans. on CAD, Vol. 8, No. 9, Sept., 1989.
24. Li-Shin and Sartaj Sahni, "Maximum Alignment of Interchangeable Terminals," IEEE Trans. on Computers, Vol. 37, No. 10, October. 1988.
25. S. Han and Sartaj Sahni, "Single-Row Routing in Narrow Streets", IEEE trans. Computer-Aided Design, Vol. CAD-3, No. 3, pp. 235~241, 1984.
26. R. D. Ringeisen, S. K. Stueckle and B. L. Piazza, "Subgraphs and Bounds on Maximum Crossings", Dept. of Mathematical Sciences Clemson Univ., pp. 1~24, July, 1990.
27. B. Krishnamurty, "An Improved Min-Cut Algorithm For Partitioning," IEEE Trans. on Computers, Vol. c-33. No. 5, pp. 438~446, May, 1984.
28. K. C. Chang and D. H. Du, "Efficient Algorithm for Layer Assignment Problem," IEEE Trans. on CAD, Vol. CAD-6, No. 1, pp. 67~87, Jan., 1987.
29. G. Fredrickson and S. Hambrusch, "Planar Linear Arrangements of Outerplanar Graphs," IEEE Trans. on Circuit and Systems, Vol. 35, no. 3, pp. 323~333, Mar. 1988.
30. D. S. Johnson, "The NP-Completeness Column: On Going Guide, (Embedding Problem)" J. Algorithm, 3, pp. 89~99, 1982.
31 D. S. Johnson, "The NP-Completeness Column: On Going Guide," J. Algorithm, 3, pp. 381~395, 1982.
32. M.R. Garey and D. Johnson, Computers and Intractability: A Guide to the theory of NP-Completeness, Freeman, San Fransicco, 1979.
33. K. Mehlhorn, Graph Algorithms and NP-Completeness, Data Structures and Algorithms 2, Springer-Verlag, Berlin, 1984.
34. C. M. Fiduccia and R. M. Mattheyses, "A Linear TIme Heuristic for Improving Network Partitions," Proc. 19th DAC, pp. 175~181, 1982.

조 환 규

1984 서울대학교 계산통계학과(학사)
1986 한국과학기술원 전산학과(석사)
1990 한국과학기술원 전산학과(박사)
1990 ~현재 부산대학교 자연과학대학 전자계산학과 조교수
관심분야 : Graph Theory, Design Automation Computer Graphics, Philosophy of Science