

듣성한 그래프 [1]: 나무 그래프처럼 생긴 그래프 모임

2022년 3월 7일

권오정

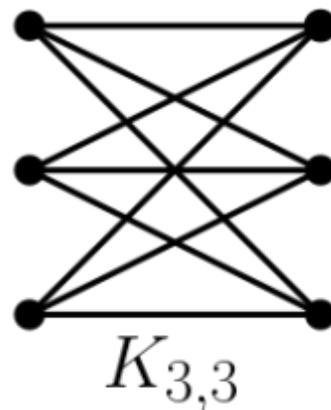
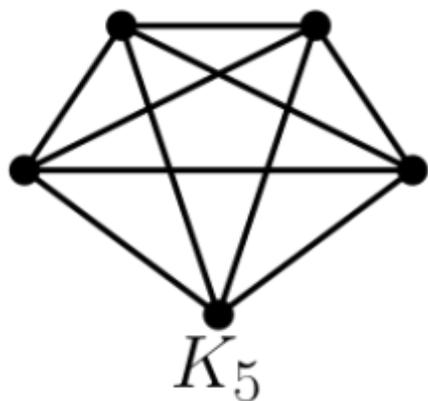


이번 연재 “듣성한 그래프”에서는 선이 듣성하게 있는 그래프 모임들의 분류에 관한 이야기를 하고자 합니다. 그래프는 유한개의 점들과 두 개의 점들을 잇는 몇몇 선들로 이루어진 구조물로 수학의 여러 분야에서 나타나며, 네트워크의 이름으로 응용 분야에서도 많이 사용됩니다. 주어진 그래프 모임에 대해 어떤 상수 c 가 존재하여 이 그래프 모임에 속한 점 n 개인 그래프가 선을 cn 개 이하만큼 가지게 되면, 이러한 그래프 모임을 듣성하다고 얘기합니다. 가장 잘 알려진 듣성한 그래프 모임으로는 평면 그래프 planar graph들의 모임이 있습니다. 평면 그래프는 평면에 선을 교차하지 않으면서 그릴 수 있는 그래프입니다. 평면 그래프에 대한 오일러 식 Euler's formula를 이용하면 점 n 개를 가지는 평면 그래프의 선의 개수는 항상 $3n$ 개 이하임을 알 수 있습니다.

현실의 여러 문제에서 나타나는 그래프 모임들은 듣성한 경우들이 많습니다. 예를 들어, 지하철역들을 점으로 표현하고 인접한 두 역을 선으로 연결하여 표현한 지하철 노선도는 하나의 평면 그래프가 됩니다. 모든 그래프에 대해서는 공통으로 쉽게 해결하는 방법을 찾기 어려운 문제일지라도 특정한 듣성한 그래프 모임에 대해서는 모임이 가진 성질을 이용하여 문제를 효율적으로 해결할 수 있지 않을까 질문해볼 수 있습니다. 이러한 맥락에서 듣성한 그래프 모임에는 어떤 것들이 있는지 알아보는 것과 이러한 모임들을 어떻게 사용할 수 있는지 알아보는 것은 그래프 이론에서 가장 흥미로운 주제 중 하나입니다.

1980-2000년대에 23편의 논문을 통해 로버트슨 N. Robertson과 시무어 P. Seymour는 그래프 구조정리 Graph structure theorem을 완성했고 [8], 이를 바탕으로 여러 그래프 문제를 해결하기 위한 방법들이 고안되었습니다. 그래프 구조정리에서는 어떤 특정한 그래프를 마이너 포함관계 graph minor로 포함하지 않는 그래프들이 어떻게 형성되는지 표현해 주

고 있습니다. 쿠라토스키(K. Kuratowski)는 1930년에 어떤 그래프가 평면 그래프일 필요충분조건은 K_5 나 $K_{3,3}$ 그래프를 마이너 포함관계로 갖지 않는 것임을 증명합니다 [3]. 그래프 구조정리는 쿠라토스키 정리를 확장하는 것으로 이해할 수 있습니다.



2008년에 네세트리(J. Nešetřil)와 오소나 데 멘데즈(P. Ossona de Mendez)는 특정한 그래프를 마이너로 포함하지 않는 모임들을 확장하기 위해 얇은 마이너(shallow minor) 개념을 이용하여 익스팬션이 상한된 그래프 모임(bounded expansion class)과 어느 곳에서도 조밀하지 않은 그래프 모임(nowhere dense class)을 제안하였습니다 [4]. 이렇게 표현되는 그래프 모임들은 평면 그래프 모임 외에도 모든 점에서 만나는 선의 개수가 어떤 상수 d 이하인 그래프 모임같이 다른 형태의 듚성한 그래프 모임들을 동시에 확장하는 그래프 모임들로 받아들여지고, 이를 기반으로 많은 연구가 진행되고 있습니다.

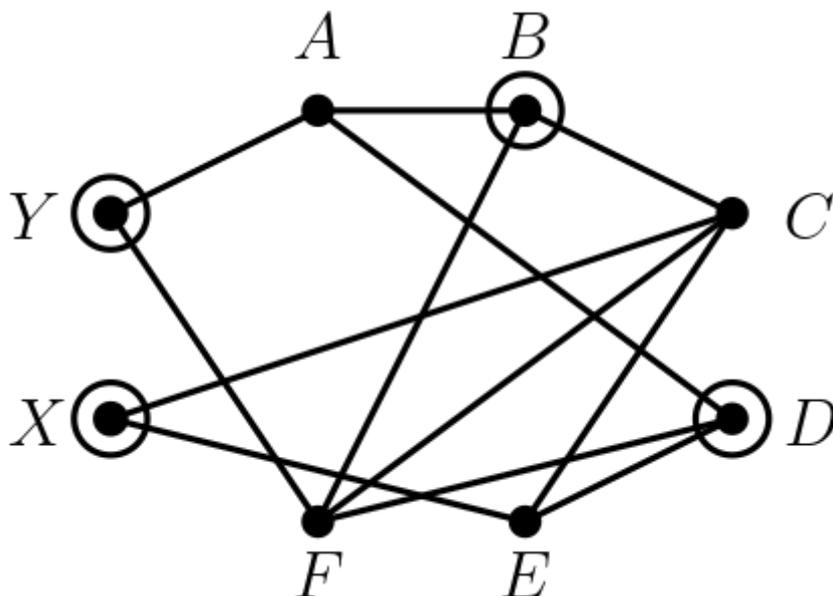
이번 글을 포함하여 앞으로 총 세 개의 글을 통해 듚성한 그래프에 관하여 알아보려고 합니다. 첫 번째 글에서는 듚성한 그래프 모임을 표현하기 위해 기본적으로 필요한 개념인 나무-폭(tree-width)에 대한 얘기를 해 보고, 알고리즘에 사용되는 예를 살펴볼 예정입니다. 두 번째 글에서는 그래프 구조이론의 구성에 관해 설명을 해 볼 예정이고, 세 번째 글에서는 익스팬션이 상한된 그래프 모임과 어느 곳에서도 조밀하지 않은 그래프 모임들이 어떻게 여러 가지 종류의 듚성한 그래프 모임을 확장하고 있는지를 관찰할 예정입니다.

최대 독립집합 문제

한 대학교에 신입생들에게 친목 도모를 위하여 스포츠 게임들을 마련하였다고 가정해 보겠습니다. 한 학생이 최대 두 개의 스포츠 게임에 참여 신청을 할 수 있다고 합니다. 다른 시간도 가능하지만, 토요일 2시가 가장 좋은 시간이어서 이날 가장 많은 스포츠 경기를 진행하고 싶다고 했을 때 토요일 2시에 동시에 진행할 수 있는 최대한의 경기 수는 몇 경기일지 알아보고 싶다고 생각해 보겠습니다.

이러한 일정 관련 문제는 그래프를 이용하여 해결하는 경우들이 많이 있습니다. 각 게임당 점을 하나씩 놓고 최소한 한 명의 사람이라도 동시에 두 게임에 지원하였을 경우 두 게임에 해당하는 점 사이에 선으로 연결을 하여 그래프를 만들어 줍니다. 어떤 사람이 두 게임을 지원하였다면 그 두 게임은 동시에 진행하지 못하기 때문에, 동시에 진행하고자 하는 게임의 목록을 보았을 때 어떤 두 게임 사이에도 선이 연결되지 않기를 원합니다. 즉, 풀고자 하는 문제는 주어진 그

래프에서 서로서로 연결되지 않은 점 집합 중 최대 크기를 갖는 집합을 찾는 문제가 됩니다. 서로서로 연결되지 않은 점 집합을 독립집합(independent set)이라고 부르고, 최대 크기의 독립집합을 찾는 문제를 최대 독립집합 문제(Maximum Independent Set Problem)이라고 부릅니다. 예를 들어, 아래와 같은 그래프에서 B, D, X, Y가 최대 독립집합을 이루게 됩니다.



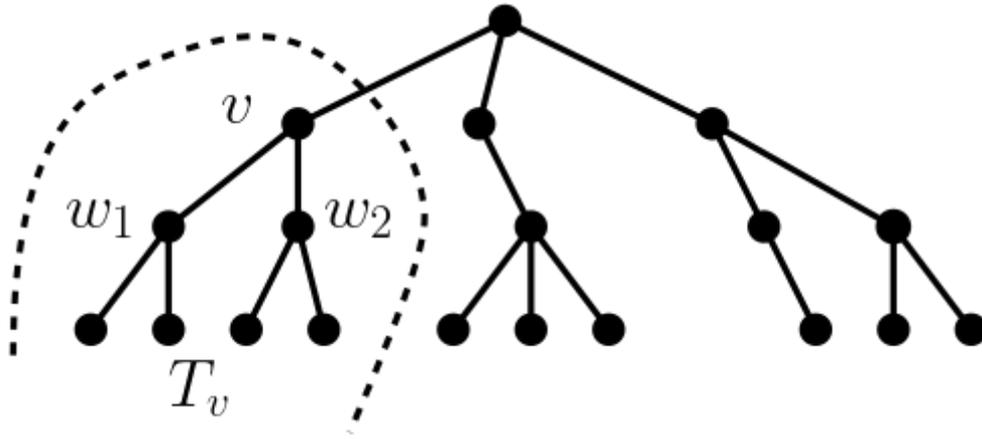
이러한 문제가 주어졌을 때 이를 효율적으로 해결할 수 있는 알고리즘이 있는지 궁금할 것입니다. 얼마나 효율적인지를 가능하는 여러 기준을 생각해볼 수 있지만, 기본적으로 생각할 수 있는 기준은 점의 개수가 n 개인 임의의 그래프가 주어졌을 때 n 에 대한 다항식 시간의 수행 시간을 통해 문제를 해결할 수 있는 것인지 보는 것입니다. 최대 독립집합 문제는 NP-hard 문제임이 알려져 있는데, 이는 다항식 시간의 알고리즘이 존재할 것이라고 믿기 어려운 문제라는 뜻입니다. 쉽게 구하는 방법은 주어진 그래프의 모든 점 부분집합을 택해서 확인하는 방법이지만 그런 경우 모든 부분집합을 고려하면서 2^n 정도의 시간이 기본적으로 들게 됩니다. 이는 점의 개수가 커지는 경우 매우 비효율적이라고 할 수 있습니다.

최대 독립집합 문제가 다항식 시간에 해결하기 어렵다고 하여서 효율적으로 풀 수 있는 알고리즘이 전혀 없는 것은 아닙니다. 여기에서 다항식 시간에 해결하기 어렵다고 말하는 의미는 모든 그래프에 대해서 공통으로 사용할 수 있는 알고리즘이 아직 발견되지 않았다는 의미입니다. 관심이 있는 그래프가 어떤 특정한 구조를 지니고 있다는 것을 미리 알고 있으면 이를 이용하여 문제를 해결할 수도 있습니다. 특히 듬성한 그래프 모임에서는 좀 더 유연하게 알고리즘을 구성할 수 있을 것으로 짐작해 볼 수 있습니다.

나무 그래프^{tree}와 나무-폭^{tree-width}

가장 간단하게 접할 수 있는 듬성한 그래프는 나무 그래프^{tree}입니다. 나무 그래프는 원 그래프^{cycle}를 가지지 않으면서 연결된 그래프입니다. 토너먼트 배치표 같은 형태에서 많이 나타나기 때문에 한 번 짚은 본 적이 있을 것 같습니다. 주어진 나무 그래프상에서 최대 독립집합 문제를 다항식 시간에 해결할 수 있음을 관찰하겠습니다. 아래와 같이 맨 꼭대

기에 있는 점을 기준으로 나무 그래프를 아래로 늘어뜨려 놓았다고 가정해 보겠습니다.



각 점 v 마다 v 아래에 있는 모든 점으로 이루어진 부분 그래프를 T_v 라고 하겠습니다. 이 때, 각 T_v 에서 점 v 를 포함하는 최대의 독립집합과 v 를 포함하지 않을 때 최대의 독립집합을 아래서부터 올라오면서 구할 것입니다. 맨 아래에 선 하나만 연결된 점 v 에 대해서는 T_v 가 v 로만 이루어져 있으므로, 점 v 를 포함하는 최대의 독립집합은 v 이고, v 를 포함하지 않을 때 최대의 독립집합은 공집합이 됩니다.

그러면 주어진 점 v 가 맨 아래에 있는 점이 아니라고 가정하고, v 바로 아래에 있는 점들 w_1, w_2, \dots, w_t 에 대해서는 위와 같은 정보를 알고 있다고 해 보겠습니다. v 를 포함하는 경우를 생각해 보면 각 w_i 를 독립집합에 동시에 넣을 수 없으므로 w_i 를 포함하지 않을 때 각각의 T_{w_i} 에서 최대의 독립집합 저장된 것을 가져와서 v 와 함께 합치면 최대의 독립집합이 됩니다. v 를 포함하지 않는 경우를 생각해 보면, 각 T_{w_i} 에서 w_i 포함 여부와 상관없이 나올 수 있는 최대의 독립집합을 가져와서 합치면 원하는 최대의 독립집합이 됩니다. 여기에서 중요한 성질은 v 아래에 나타나는 그래프들 T_{w_1}, \dots, T_{w_t} 들이 다시 나무 그래프가 되면서 이들 사이에 선이 없기 때문에 독립집합을 합치는 것을 쉽게 할 수 있다는 것입니다. 총 수행 시간은 $O(n)$ 정도의 시간이 들게 됩니다.

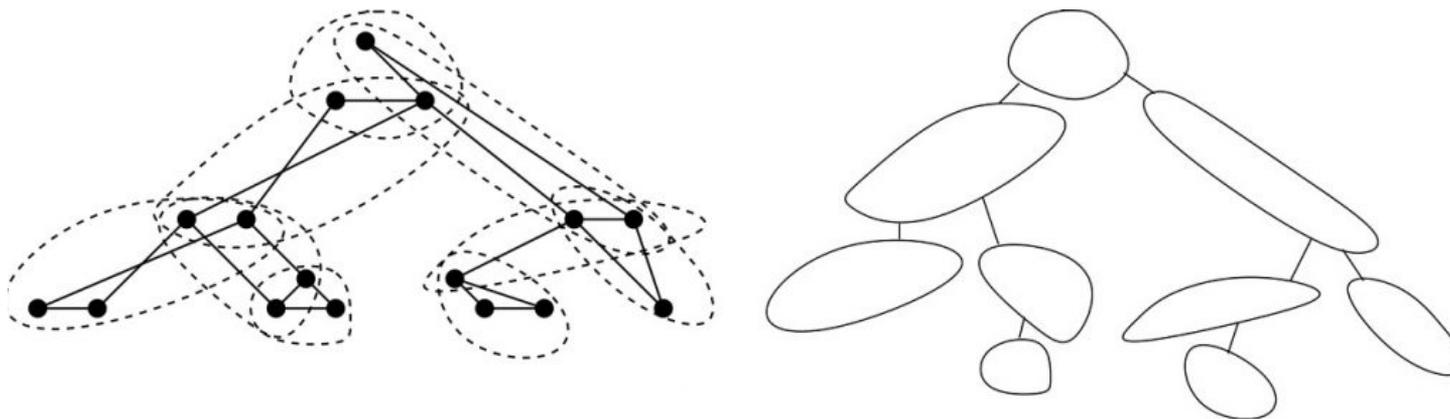
안타깝게도 임의로 주어진 그래프가 나무 그래프일 확률은 매우 희박합니다. 그래서 이러한 성질을 조금 더 일반화할 방법에 대해 생각해볼 수 있습니다. 나무 그래프와 비슷하게 '작은 개수'의 점을 지워서 비슷한 성질을 가지는 그래프들로 반복해서 분리해나갈 수 있는지를 판단하고자 고안된 매개변수가 나무-폭^{tree-width}입니다. 나무-폭을 정의하기 위해서는 나무-분해^{tree-decomposition} 개념이 필요합니다.

그래프 G 의 점 집합을 $V(G)$ 로 표기하겠습니다. 주어진 그래프 G 의 나무-분해는 어떤 나무 그래프 T 와 T 의 점 집합에서 G 의 점 부분집합으로 보내는 함수 $f: V(T) \rightarrow 2^{V(G)}$ 의 쌍 중에서 '특정한 조건'을 만족하는 것으로 정의합니다. 나무 그래프의 점 x 에 의해 대응되는 $f(x)$ 를 x 에 부여된 가방^{bag}으로 부릅니다. 특정한 조건들은 원래 그래프에서의 연결성을 잘 보존하게 하기 위한 설정으로 (1) 모든 점이 가방에 최소 한 번은 나타나야 하고, (2) 각 선에 대해 선의 양 끝점을 포함하는 가방이 하나는 있어야 하고, (3) G 의 각 점 v 마다 $\{t \in V(T) : v \in f(x)\}$, 즉, v 를 가지고 있는 가방들에 대응되는 나무 그래프의 점들이 연결되게 나타난다는 조건들입니다.

가장 간단하게 나무-분해를 만드는 방법은 한 점 v 로 이루어진 나무 그래프 T 에 $f(v) = V(G)$ 로 정의하는 것입니다. 이는 나무-분해가 맞지만, 한 가방이 가진 점의 개수가 매우 많게 됩니다. 관심이 있는 것은 모든 가방의 크기가 어떤 상수 c 보다 같거나 작게 나무-분해를 할 수 있는 것인 것입니다. 주어진 나무-분해에서 가방의 크기 중 가장 큰 가방

의 크기에서 1을 뺀 것을 폭^{width}으로 정의합니다. 주어진 그래프에서 만들어질 수 있는 모든 나무-분해 중 폭이 가장 작은 것의 폭을 주어진 그래프의 나무-폭^{tree-width}이라고 부릅니다. 예를 들어, 나무-폭이 5인 그래프는 폭이 5인 나무-분해가 존재한다는 것을 알려줍니다.

가령 아래의 예제는 주어진 그래프의 폭이 3인 나무-분해를 보여줍니다. 오른쪽 그림은 가방들이 어떻게 나무 그래프의 형태로 나열된 지를 나타내 줍니다.



이러한 나무-분해가 주어져 있을 때, 최대 독립집합 문제를 쉽게 풀 수 있다는 것을 관찰해보고자 합니다. 나무-분해의 한 가방 $f(v)$ 를 볼 때, 그 가방과 그 아래쪽에 나타나는 가방에 나타나는 모든 점으로 이루어지는 부분 그래프를 T_v 라고 해 보겠습니다. 어떤 독립집합이 있을 때 $f(v)$ 에서 교집합으로 나타날 방법은 단순히 $f(v)$ 의 모든 부분집합의 개수로 상한됩니다. 각각의 $f(v)$ 의 부분집합 S 에 대해 교집합으로 S 를 가지면서 T_v 에서 생기는 최대 크기의 독립집합을 아래쪽에서 저장한 정보를 이용하여 얻으려고 합니다. 이를 위해서 선택된 S 를 $f(v)$ 바로 아래에 있는 가방들로 S 를 제한시킨 후, 각 부분 그래프에서 저장된 최대 독립집합을 가져온 후에 이들을 합쳐서 원하는 정보를 얻을 수 있습니다.

연재글

등성한 그래프

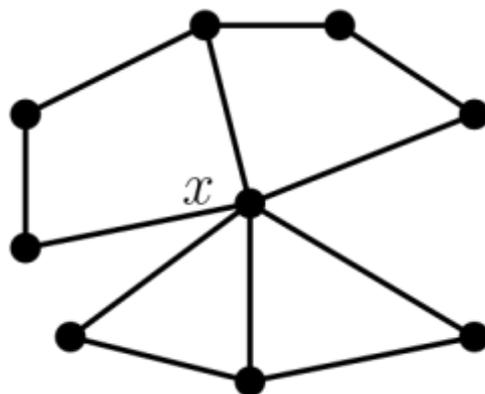
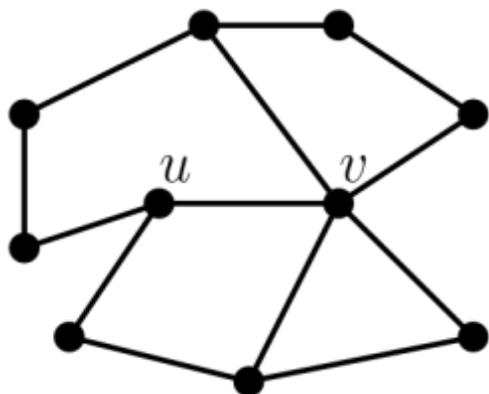
1. [등성한 그래프 \[1\]: 트리처럼 생긴 그래프 모임](#)
2. [등성한 그래프 \[2\]](#)
3. [등성한 그래프 \[3\]](#)

나무-분해를 이용하여 문제를 해결하기 위해서는 주어진 그래프의 나무-분해를 찾는 알고리즘이 필요합니다. 그래프의 나무-폭을 정확히 구하는 것은 NP-hard 문제입니다. 그러나 적절한 상수 c 가 있어서, 그래프 G 와 어떤 수 k 가 주어져 있을 때 G 의 나무-폭이 k 보다 크다고 얘기해 주거나 ck 이하의 폭을 가지는 나무-분해를 효율적으로 제공해주는 다양한 알고리즘들이 개발되어 있어서 사용할 수 있습니다. 수행 시간은 k 에 대해서는 지수적인 수행 시간을 가

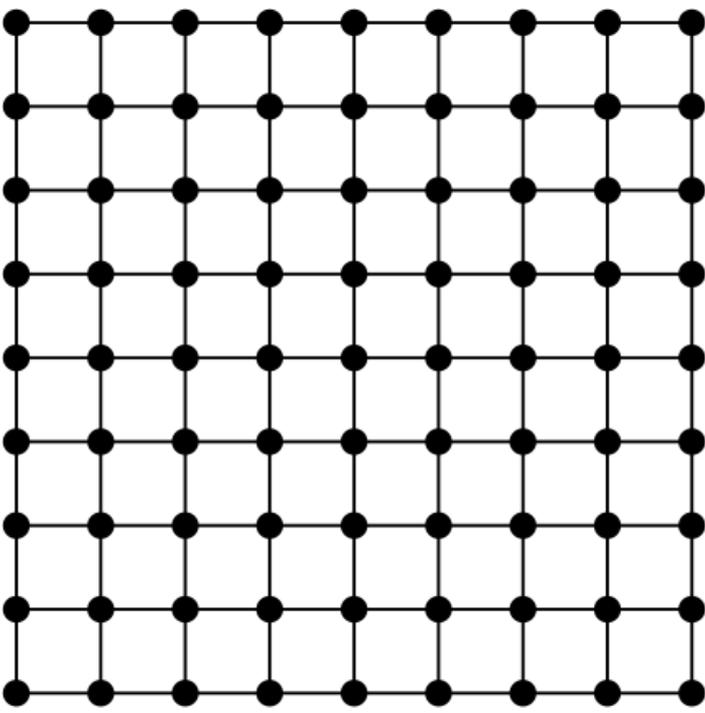
지나 n 에 대해서는 다항식 시간을 갖게 되는 알고리즘입니다. 비교적 간단하게 사용할 수 있는 알고리즘으로는 로버트슨^{N.Robertson}과 시무어^{P. Seymour}가 제안한 $(4k + 3)$ -근사 알고리즘을 사용할 수 있고 [Z], 이론적으로 더 효율적인 알고리즘을 찾는 연구가 계속해서 진행되고 있습니다.

격자그래프 마이너

나무-폭이 작은 그래프는 나무 그래프와 비슷한 형태로 반복적으로 분리가 되는 성질을 가지고 있기 때문에 이러한 성질을 알고리즘에 적용하여 사용할 수 있다는 것을 관찰하였습니다. 이와 반대의 상황을 생각하여 주어진 그래프의 나무-폭이 매우 크다면 어떤 정보를 얻을 수 있을지에 대해 궁금해할 수 있습니다. 나무-폭이 매우 큰 그래프에서는 항상 큰 격자 그래프를 마이너 포함관계로 가진다는 것을 로버트슨^{N.Robertson}과 시무어^{P. Seymour}가 증명하였고, 이를 격자 그래프 마이너 정리^{Grid minor theorem}라고 부릅니다. 그래프 H 가 다른 그래프 G 에서부터 선을 지우거나 점을 지우거나 선을 축약하는 연산을 반복적으로 시행함으로써 얻을 수 있으면 H 가 G 의 마이너라고 부릅니다. 다음 예는 선 uv 를 축약하는 연산을 보여줍니다.



다음은 가로, 세로가 각각 9개의 점을 가지는 (9×9) -격자 그래프를 나타냅니다.



(정리 1. [6]) 다음을 만족하는 어떤 함수 $f : \mathbb{N} \rightarrow \mathbb{N}$ 가 존재한다. 임의의 k 에 대해 주어진 그래프의 나무-폭이 $f(k)$ 이상이면 항상 $(k \times k)$ -격자 그래프를 마이너 포함관계로 가진다.

여기에서 필요한 $f(k)$ 함수를 줄이는 것에 관한 많은 연구가 진행되었습니다. 가장 최근의 결과로는 추조이^{Chuzhoy}와 탄^{Tan}이 2021년에 $f(k) = O(k^9 \text{polylog } k)$ 이면 된다는 것을 증명하였습니다 [1].

나무-폭과 비슷하게 나무 그래프와 비슷한 구조를 표현하되 행렬의 계수^{rank}를 이용하여 측정하는 매개변수로서 계수-폭^{rank-width}이라는 것이 있습니다. 계수-폭의 장점은 등성하지 않는 그래프 모임에 대해서도 나무 그래프와 비슷한 구조를 표현해준다는 것입니다. 가령 모든 점이 서로서로 연결된 완전 그래프는 나무-폭은 점의 개수에서 1을 뺀 것으로 매우 크지만 계수-폭은 1이 됩니다. 간단한 직관은 완전 그래프의 점 집합을 어떻게 두 개로 나누든지 그 두 집합 사이는 서로 완전히 연결되어 있고, 이는 모든 자리에 1이 들어 있는 행렬로 두 집합의 관계가 표현되며 이 행렬의 계수가 1로서 낮게 표현되므로 이 분리의 복잡도는 낮다고 생각할 수 있는 것입니다. 계수-폭이 작은 그래프는 이렇게 행렬의 계수가 낮게 측정되는 분리로 반복해서 분리 가능한 그래프를 표현합니다. 계수-폭을 고안한 엄상일 교수는 2005년에 계수-폭이 큰 그래프에서 반드시 나타나는 구조와 관련된 추측을 제시하였는데, 저는 길란^{Geelen}, 맥커티^{McCarty}, 윌란^{Wollan}과 함께 이 추측을 증명하였고, 이 논문은 2020년에 온라인 게재되었습니다 [2].

마지막으로, 나무-폭이 크다는 것은 전체 그래프에서 잘 분해가 되지 않는 부분이 국소적으로 존재한다는 것을 알려 줍니다. 하지만 이 사실이 전체 그래프가 어떻게 생겼다는 것을 이야기해주지는 않습니다. 격자 그래프 마이너 정리를 토대로 연구가 진행되어 그래프 전체의 구조를 설명하게 해 준 정리가 그래프 구조정리입니다. 이에 대해 다음 글에서 다루어 보도록 하겠습니다.

참고문헌

1. J. Chozhoy and Z. Tan. Towards tight(er) bounds for the Excluded Grid Theorem. *J. Combin. Theory Ser. B*, 146:219–265, 2021
2. J. Geelen, O. Kwon, R. McCarty and P. Wollan. The Grid Theorem for vertex-minors. *J. Combin. Theory Ser. B*, accepted.
3. K. Kuratowski. Sur le problème des courbes gauches en topologie. *Fund. Math.* (in French), 15: 271–283, 1930.
4. J. Nešetřil and P. Ossona de Mendez. Grad and classes with bounded expansion I. Decompositions. *European J. Combin.*, 29(3):760-776, 2008
5. S. Oum. Rank-width and vertex-minors. *J. Combin. Theory Ser. B*, 95(1):79-100, 2005
6. N. Robertson and P. D. Seymour. Graph minors. V. Excluding a planar graph. *J. Combin. Theory Ser. B*, 41(1):92–114, 1986
7. N. Robertson and P. D. Seymour. Graph minors. XIII. The disjoint paths problem. *J. Combin. Theory Ser. B*, 63(1):65–110, 1995
8. N. Robertson and P. D. Seymour. Graph minors. XX. Wagner’s conjecture. *J. Combin. Theory Ser. B*, 92(2):325–357, 2004