트리의 최적 경로 분할을 위한 다항시간 알고리즘

(A Polynomial-time Algorithm to Find Optimal Path Decompositions of Trees)

안 형 찬 †
(Hyung-Chan An)

요 약 트리의 최소단말경로분할이란 트리를 에지가 서로 소인 단말 노드 간 경로들로 분할하되, 가장 긴 경로의 길이를 최소화하는 문제이다. 본 논문에서는 트리의 최소단말경로분할을 $O(|V|^2)$ 시간에 구하는 알고리즘을 제시한다. 이 알고리즘은 주어진 최적화 문제를 이에 대응하는 결정 문제, 즉 최소단말경로분할의 비용이 l 이하인지를 결정하는 문제를 이용한 이진 탐색으로 환원한다. 결정 문제는 트리를 한 차례 순회하는 동안 동적 계획법에 의해 해결된다.

키워드 : 그래프 이론, 트리, 동적 계획법, 경로 분할

Abstract A minimum terminal path decomposition of a tree is defined as a partition of the tree into edge-disjoint terminal-to-terminal paths that minimizes the weight of the longest path. In this paper, we present an $O(|V|^2)$-time algorithm to find a minimum terminal path decomposition of trees. The algorithm reduces the given optimization problem to the binary search using the corresponding decision problem, the problem to decide whether the cost of a minimum terminal path decomposition is at most l. This decision problem is solved by dynamic programing in a single traversal of the tree.

Key words : graph theory, tree, dynamic programming, path decomposition

1. 서 론

트리에서의 최적화 문제는 일반적인 그래프에 대한 NP-hard 문제의 특수한 경우로서[1,2] 뿐만 아니라, 그 자체로서도 중요한 문제의 범주이다. 트리에서의 최적화 문제를 해결하는 자연스러운 접근법 중 하나로, Chen et al.[1]에서 보듯 동적 계획법을 고려할 수 있다. 그러나 어떤 문제들은 전체 최적해가 부트리에 밀접하게 연관되어 있어, 동적 계획법을 사용하기 위해 필수적인 최적 부구조(optimal substructure)가 명확히 드러나지 않기도 한다.

이러한 문제의 한 예로, 18회 한국 정보 올림피아드[3]에 출제된 "버스 노선" 문제를 들 수 있다. 이 문제는 $2k$ 개의 단말 노드를 갖는 트리를 에지가 서로 소인 k 개의 단말 노드간 경로들로 분할하되, 이들 분할 중 주어진 목적함수를 최소화하는 것을 찾는 최적화 문제이다. 그러나 주어진 목적함수의 특성 상, 전체 트리의 최적해가 각 부트리의 최적해의 함수로 주어지지 않기 때문에 동적 계획법을 그대로 적용하기는 곤란한 문제이다. 이 문제의 엄밀한 정의는 다음 장에 주어진다.

기존에 Guruswami et al.[4]을 비롯한 많은 연구가 이루어져 잘 알려져 있는 EDP(edge-disjoint paths) 문제나, Lukes[5]의 트리의 노드 집합의 최적 파티션에 관한 연구 등이 이 문제와 다소 정의(定義)상의 유사점을 가지나, 이 문제에 대한 직접적인 과거의 연구는 찾지 못하였다. 본 논문에서는 동적 계획법을 직접적으로 적용하기 힘든 트리에서의 최적화 문제인 이 문제를 변형하여 동적 계획법을 사용하는 다항시간 알고리즘을 제시하고자 한다. 구체적으로, 주어진 최적화 문제에 대응하는 결정 문제(decision problem)를 고려하여([6]) 이 결정 문제가 최적 부구조를 가지므로 동적 계획법으로 해결될 수 있음을 보이고, 이진 탐색을 통해 최적해를 구하는 알고리즘을 제안하고자 한다.

·본 연구의 일부는 서울대학교 컴퓨터공학부 재학 중 수행하였음
† 학생회원 : Cornell University Field of Computer Science.
Supported by graduate school university fellowship
anhc@cs.cornell.edu
논문접수 : 2005년 8월 9일
심사완료 : 2007년 3월 16일

본 논문의 2장에는 사용되는 용어와 문제의 엄밀한 정의를 서술한다. 이어서 3장에서는 알고리즘 설계에 필요한 트리의 최소단말경로분할의 성질을 살피고, 이에 기초해 주어진 문제를 해결하는 알고리즘을 4장에 제시한다. 5장에서는 가중치가 없는 트리에서의 특수한 경우를 검토한다.

2. 용어 및 문제의 정의

본 논문에서 사용되는 정의의 일부는 Cormen et al.[7]과 Chartrand et al.[8]을 따른다.

트리를 사이클이 없는 연결 무향 그래프로 정의하고, 트리에서 차수가 1인 정점을 단말 (노드), 한 단말에서 시작하여 다른 단말에서 끝나는 경로를 단말경로라 하자. 경로 P를 문맥에 따라 적절히 경로 상의 에지의 집합, 혹은 원 트리의 부그래프로 보기로 하자.

가중치가 주어진 그래프의 에지 e에 대하여 그 가중치를 $w(e)$라 하고, 정점 u, v에 인접한 에지를 (u, v)와 같이 나타낸다. 루트가 있는 트리에서 노드 v의 부모 노드를 parent(v)로 표시하며, 자식 노드가 없는 노드를 리프라 하자. 경로 P의 가중치는 경로를 이루는 에지의 가중치의 합, 즉 $\sum_{e \in P} w(e)$로 정의한다.

트리 $T=(V,E)$와 그 단말경로들 P_i에 대하여, 그 경로들이 에지가 서로 소이고($P_i \cap P_j = \phi \quad (\forall i \neq j)$), 합하면 T가 될 때($\cup P_i = E$), 이들 P_i의 집합을 $T=(V,E)$의 단말경로분할이라고 정의하자. 단말경로분할의 비용은 이 단말경로분할을 이루는 경로의 가중치 중 최대값으로 정의된다.

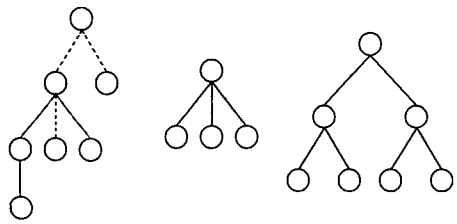

그림 1 단말경로분할의 예. 첫 번째 트리는 각각 점선과 회색선으로 표시된 두 개의 단말경로로 분할되었다. 두 번째와 세 번째의 트리는 단말경로분할이 존재하지 않는다.

어떤 트리에 대하여, 최소의 비용을 갖는 단말경로분할을 최소단말경로분할이라 하고, l 이하의 비용을 갖는 단말경로분할을 l-단말경로분할이라 하자.

이제 본 논문에서 다룰 문제를 정의하면, 다음과 같다.

문제 1. 트리 $T=(V,E)(|V| \geq 2)$와 그 에지의 가중치가 주어질 때, 단말경로분할이 존재하는지 결정하고, 존재할 경우 최소단말경로분할을 구하라.

주어진 단말경로분할에 대하여, 같은 단말경로의 에지는 같은 색, 다른 경로의 에지는 다른 색으로 칠한 것을 단말경로분할의 채색이라 하고, 두 채색이 같은 단말경로분할을 나타낼 때 이들 두 채색이 동등하다고 한다.

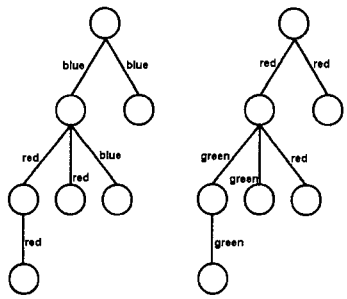

그림 2 단말경로분할의 채색. 두 채색은 동등하다.

루트가 있는 트리의 루트 아닌 노드 v_c의 부모가 v_p일 때, 다음의 용어를 정의한다. v_c의 확장부트리 T는 v_c와 그 자손들, 그리고 v_p에 의해 유도(induce)되는 부그래프이다. 다시 말하여, T는 v_c의 부트리에 v_p를 추가한 트리이다. 이 때에 추가되는 에지, 즉 (v_c, v_p)를 v_c의 (혹은 T의) 확장에지라고 하자. v_c의 확장부트리의 단말경로분할을 생각할 때, 확장경로는 확장에지를 포함하는 단말경로를 가리키며, 확장비용은 확장경로의 가중치이다. 마지막으로, v_c의 확장부트리의 최소확장 l-단말경로

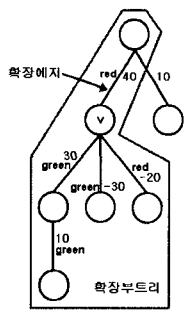

그림 3 정점 v의 확장부트리와 확장에지. 확장부트리는 v의 부트리에 그 부모노드를 더한 것으로, 회색 주머니로 표시되어 있다. 확장부트리의 최소확장 10-단말경로분할(의 채색)이 함께 표시되어 있으며, 'red'로 표시된 두 에지가 확장경로에 해당한다.

분할은, 확장경로 외의 단말경로의 가중치가 l을 넘지 않으면서 확장비용을 최소화하는 단말경로분할로 정의된다. 최소확장 l-단말경로분할의 비용이란 그 확장비용을 말한다.

짝수개의 실수 $a_1, a_2, \cdots, a_{2n}$에 대해, 이들 실수를 2개씩 n개의 쌍으로 짝짓는 것을 생각하자. 짝지어진 실수의 합이 l을 넘지 않도록 하는 짝짓기 방법이 존재할 때, $\{a_i\}$를 'l 이하로 짝지을 수 있다'고 한다. 길이 0인 수열은 l 이하로 짝지을 수 있는 것으로 정의한다.

$G_1=(V_1,E_1)$, $G_2=(V_2,E_2)$, $G_d=(V_1-V_2,(E_1-E_2)\cap((V_1-V_2)\times(V_1-V_2)))$, $G_i=(V_1\cap V_2,E_1\cap E_2)$에 대하여 $G_d=G_1-G_2$, $G_i=G_1\cap G_2$과 같이 표기하자. 또한 $G_1=G_2$는 $(V_1=V_2)\wedge(E_1=E_2)$, $G_1\neq G_2$는 $\neg(G_1=G_2)$로 정의된다.

3. 단말경로분할의 성질

트리와 확장부트리의 단말경로분할과 관련하여 다음의 성질을 알 수 있다. 명백한 정리에 대해서는 증명을 생략한다.

보조정리 1. 트리 $T=(V,E)$의 단말경로에 해당하는 부그래프에서, 이 단말경로의 양 끝 정점의 차수는 1, 다른 모든 정점의 차수는 2이다.

따름정리 1. 트리 $T=(V,E)(|V|\geq 2)$가 단말경로분할을 갖는다면, 모든 $v\in V$에 대해 v의 차수는 1 혹은 짝수이다.

증명. v가 단말이 아니면 모든 v를 포함하는 단말경로는 v에 인접한 에지를 둘 갖는다. 단말경로분할이 존재하므로 모든 에지는 하나의 단말경로에 속하며, 따라서 v의 차수는 짝수이다.

v가 단말이면 차수는 1이다. □

따름정리 2. 루트가 있는 트리 $T=(V,E)(|V|\geq 2)$가 단말경로분할을 갖는다면, 모든 루트도 리프도 아닌 노드 v의 자식 노드는 홀수 개이다.

보조정리 2. 루트가 있는 트리 T와 그 임의의 단말경로분할의 채색 C를 생각하자. T의 노드 v의 확장부트리 T'에 대하여, C의 T'부분을 따로 취한 C'는 T'의 단말경로분할의 채색이다.

보조정리 3. 루트 있는 트리 T와 그 한 노드 v의 확장부트리 T'를 생각하고, T'의 확장에지를 e라 하자. T의 임의의 단말경로분할의 채색을 C, T'의 임의의 단말경로분할의 채색을 C'라고 하자. e의 색이 C와 C'에서 같고, 그 색 외에는 C와 C' 모두에 공통적으로 사용된 색이 없을 때, C의 T'부분을 C'로 치환한 C''는 T의 단말경로분할의 채색이다.

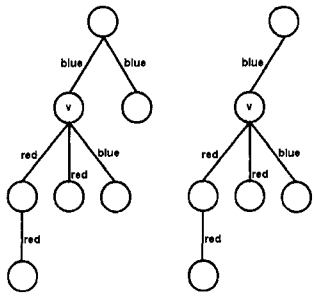

그림 4 보조정리 2의 예. 루트 있는 트리의 단말경로분할의 채색의 한 예와, 그 채색에서 노드 v의 확장부트리 부분을 따로 취한 것. 오른쪽에 나타난 v의 확장부트리 부분이 그 단말경로분할로 채색되어 있음을 알 수 있다.

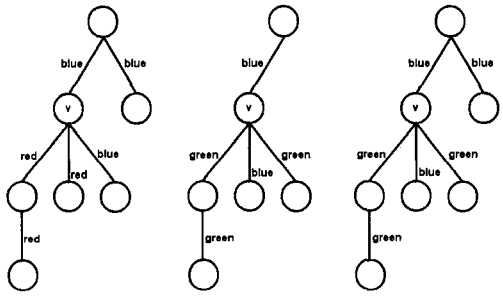

그림 5 보조정리 3의 예. 가장 왼쪽은 전체 트리의 단말경로분할의 채색이며, 가운데는 v의 확장부트리의 단말경로분할의 채색이다. 왼쪽 채색에서 확장부트리 부분을 가운데에 표시된 채색으로 대체하면, 오른쪽에 표시된 채색을 얻는다. 이 채색은 전체 트리의 단말경로분할의 채색임을 알 수 있다.

보조정리 4. 어떤 루트 있는 트리의 루트도 리프도 아닌 노드 v와, 그 확장부트리 T를 생각하자. v의 각 자식노드 u_i에 대해, u_i의 확장부트리를 T_i', 그 임의의 최소확장 l-단말경로분할의 채색을 C_i'라 하자. T의 최소확장 l-단말경로분할이 존재하는 경우, 다음 조건을 만족하는 T의 최소확장 l-단말경로분할의 채색 C_0가 존재한다.

조건: 모든 i에 대하여, C_0의 T_i'부분이 C_i'와 동등하다.

증명. T의 임의의 최소확장 l-단말경로분할의 채색 C를 생각하자. 각 자식 노드 u_i와 그 확장부트리 T_i', 최소확장 l-단말경로분할의 채색 C_i'에 대하여 도움정리의 조건이 만족되도록 C를 변형해 나가는 방법을 제시하여

증명한다 (constructive proof). T의 확장에지를 e, T_i'의 확장에지를 e'라 하자.

C에서 $e'=(v,u_i)$는 i) e와 같은 색이거나, ii) $j\neq i$에 대해 (v,u_j)와 같은 색이다. ($\because v$가 단말이 아니므로 도움정리 1에 의하여)

경우 i).

C_i'의 색상을 바꾸어 이와 동등한 C''를 얻되, 1) C''에서의 e'의 색상이 C에서의 e'의 색상과 같도록 하고, 2) 나머지 색상은 C와 겹치지 않도록 하자.

C에서 T_i' 부분을 C''로 바꾸어 얻어지는 채색 C'''은 보조정리 3에 의해 T의 단말경로분할의 채색이다. 이 때, C'''에서의 단말경로는 1) e를 제외한 $T-T_i'$의 에지로만 이루어지거나, 2) e'를 제외한 T_i'의 에지로만 이루어지거나 3) e와 e'를 포함한다. C, C''가 최소확장 l-단말경로분할의 채색이므로 1), 2)에 해당하는 C'''에서의 단말경로는 가중치가 l 이하이다.

e'를 포함하는 C'''에서의 단말경로를 P''', C에서의 단말경로를 P라 하자. 이 때, $P'''\cap(T-T_i')=P\cap(T-T_i')$이다. 한편, C''가 T_i'의 최소확장 l-단말경로분할의 채색이므로, 정의에 의해 경로 $P'''-(T-T_i')$의 가중치가 경로 $P-(T-T_i')$의 가중치보다 작거나 같다 (보조정리 2). 따라서 P'''의 가중치는 P의 가중치보다 작거나 같다.

따라서 C'''는 T의 최소확장 l-단말경로분할의 채색이며, 그 비용은 P'''의 가중치로, C에서와 같다.

경우 ii).

i)에서와 마찬가지 방식으로 C_i'의 색상을 바꾸어 이와 동등한 C''를 얻자. C에서 T_i' 부분을 C''로 바꾸어 얻어지는 채색 C''' 역시 앞에서와 마찬가지로 T의 단말경로분할의 채색이다.

이 때, C'''에서의 단말경로는 1) $T-T_i'$의 에지로만 이루어지거나, 2) e'를 제외한 T_i'의 에지로만 이루어지거나 3) e'를 포함한다.

1)에 해당하는 C'''에서의 단말경로는 C와 C'''에서 동일하다. T의 확장경로도 여기에 해당한다.

2)에 해당하는 C'''에서의 단말경로는 C''의 정의에 의해 l 이하의 가중치를 갖는다.

3)에 해당하는 C'''에서의 단말경로 P'''는 하나가 존재하며 T의 확장경로가 아니고, T_i'의 에지로 이루어진 부분과 나머지 부분으로 나눌 수 있다. C에서도 e'를 포함하는 경로 P가 존재하며, 마찬가지로 T_i'의 에지로 이루어진 부분과 나머지 부분으로 나눌 수 있다. P'''의 T_i'의 에지로 이루어진 부분의 가중치를 l_1, 나머지 부분의 가중치를 l_2라 하자. P에 대해서도 T_i'의 에지로 이루어진 부분의 가중치를 l_1', 나머지 부분의 가중치를 l_2'라 하자.

C가 최소확장 l-단말경로분할의 채색이므로 $l_1'+l_2'\leq l$이 성립한다. 또한 P'''와 P의 $T-T_i'$의 에지로 이루어진 부분은 동일하므로 $l_2=l_2'$이다. 마지막으로, C''가 T_i'의 최소확장 l-단말경로분할의 채색이므로 $l_1\leq l_1'$이 성립한다. 그러므로 l_1+l_2로 주어지는 P'''의 가중치는 l 이하이다.

따라서 C'''는 T의 최소확장 l-단말경로분할의 채색이며, 그 비용은 C에서와 동일하다.

임의의 최소확장 l-단말경로분할의 채색 C에 경우 i), ii)를 반복 적용하여, 주어진 조건을 만족하는 T의 최소확장 l-단말경로분할의 채색을 얻을 수 있다. □

4. 알고리즘

최소단말경로분할을 구하는 대신 그 비용만을 구하는 다음의 문제를 생각하자.

문제 2. 트리 $T=(V,E)(|V|\geq 2)$의 단말경로분할이 존재하는지 결정하고, 존재할 경우 최소단말경로분할의 비용을 구하라.

이 문제를 풀기 위해, 문제 2의 해가 l 이하인지를 묻는 결정 문제를 생각할 수 있다[6]. l-단말경로분할의 정의에 의해, 이 결정 문제는 다음과 같이 표현될 수 있다.

문제 3. 트리 $T=(V,E)(|V|\geq 2)$에 대하여, l-단말경로분할이 존재하는지 결정하라.

문제 2의 해가 될 수 있는 후보들의 유한 집합이 주어질 때, 문제 3을 반복적으로 풀어 이진 탐색함으로써 문제 2를 풀 수 있다[6]. 문제 2의 해가 될 수 있는 후보들은 T의 모든 단말경로의 가중치들이다.

보조정리 5. 문제 2의 해는 T의 모든 단말경로들의 가중치 중 하나로 주어진다.

보조정리 6. 트리 $T=(V,E)(|V|\geq 2)$에 대하여, T의 모든 단말경로들의 가중치의 집합은 $O(|V|^2)$ 시간에 구할 수 있다.

증명. 루트가 있는 트리를 순회하면 루트에서 다른 모든 정점으로 가는 경로의 가중치를 구할 수 있다. 순회는 $O(|V|)$ 시간에 할 수 있고, T의 모든 단말에 대해 각각을 루트로 순회하기를 반복하면 $O(|V|^2)$ 시간에 모든 단말경로들의 가중치의 집합을 구할 수 있다. □

둘 이상의 노드를 갖는 트리는 단말 노드를 가지므로, 일반성을 잃지 않고 주어진 입력 T가 루트의 차수가 1인 루트 있는 트리라고 가정할 수 있다. 따라서,

문제 4. 루트 있는 트리 $T=(V,E)(|V|\geq 2)$와 그 루트가 아닌 노드 v에 대하여, v의 확장부트리의 최소확장 l-단말경로분할이 존재하는지 결정하고, 존재하면 그 비용을 구하라.

에 대한 알고리즘이 주어지면, 문제 3을 해결할 수 있다. 즉, 이 알고리즘을 루트의 자식 노드에 대하여 수행하여, 최소확장 l-단말경로분할이 존재하고 그 비용이 l 이하이면 문제 3에 대한 답은 '예'이며, 그렇지 않으면 '아니요'가 된다.

이제 문제 4를 푸는 알고리즘을 생각하자. 루트 아닌 노드 v가 최소확장 l-단말경로분할을 가질 경우, 따름정리 2에 의해 v는 홀수 개의 자식노드를 갖는다. 뿐만 아니라, 보조정리 4에 의해 그 채색은 v의 자식노드들의 확장부트리의 최소확장 l-단말경로분할의 채색으로부터 구할 수 있으며, 다만 결정할 것은 v의 어떤 두 자식노드 v_1, v_2에 대하여 (v,v_1), (v,v_2)가 같은 색을 가지며, 어느 한 자식노드 v_c에 대하여 v의 확장에지와 v_c의 확장에지가 같은 색을 가질 것인가이다. 확장에지가 같은 색을 가지는 두 자식노드의 확장경로는 하나로 이어지며, 따라서 문제 4는 결국, 모든 자식노드의 확장부트리의 최소확장 l-단말경로분할의 비용으로 주어지는 수열 중 가능한 한 작은 값의 하나를 제외한 나머지를 l 이하로 짝지을 수 있는지 판단하는 문제로 환원(reduce)된다.

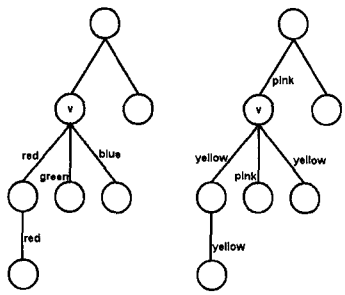

그림 6 노드 v의 확장부트리에 대한 알고리즘 1의 동작. 왼쪽은 v의 각 자식노드의 확장부트리의 최소확장 l-단말경로분할의 확장경로들의 예이다. 즉, 각각 red, green, blue로 표시된 세 개의 경로가 각 자식노드의 확장경로이다. 만약 이들 중 green을 v의 확장경로에 포함시키고, 나머지 두 확장경로는 같은 경로로 묶기로 하였다면 오른쪽과 같은 채색이 된다. (green은 v의 확장에지와 함께 pink가 되었고, 나머지 두 경로는 공히 yellow가 되었다.) 이와 같이 어떤 자식노드를 확장경로에 포함할 것인지를 결정하는 것이 알고리즘 1의 (특히, 단계 8의) 동작이다.

이에 따라 다음의 알고리즘이 주어진다.

알고리즘 1.

```
1   d ← v의 자식노드의 수
2   if d = 0 then
        min_extended_cost ← w((v, parent(v)))
3   elseif d가 짝수이면 then
    min_extended_cost ← ∞
4   else
5       for each child v_i of v,
6           v_i의 확장부트리에 대하여 재귀적으로
            같은 문제를 풀고 그 결과로 얻은
            min_extended_cost를 l_i라 한다
7       {l_k}를 오름차순으로 정렬한다
8       {l_m}_{m≠i}을 l 이하로 짝지을 수 있는 최소의 i
        를 구한다
9       if 그러한 i가 없으면 then
            min_extended_cost ← ∞
10      else
            min_extended_cost ← l_i+w((v, parent(v)))
```

보조정리 7. 알고리즘 1은 v의 확장부트리의 최소확장 l-단말경로분할이 존재할 경우 그 비용을 min_extended_cost에 구하며, 그렇지 않을 경우 min_extended_cost를 ∞로 한다.

증명. 알고리즘 1의 단계 2는 정의에 의해 옳다. 단계 3은 따름정리 2에 의해 옳다.

주어진 보조정리를 증명하기 위해 트리의 각 노드에 대한 귀납법을 사용하자. 리프노드를 처리하는 단계 2가 출발점(basis)이 된다. 귀납 가정에 의해, 단계 5~6에서 각 자식노드의 확장부트리의 최소확장 l-단말경로분할의 비용이 구해진다. 각 자식노드의 확장에지 중, v의 확장에지와 같은 경로에 들어갈 것을 고르고(단계 8), 이 때에 다른 단말 경로의 가중치가 l이 넘지 않는지는 $\{l_m\}_{m\neq i}$을 l 이하로 짝지을 수 있는지를 검사함으로써 확인한다. $\{l_m\}_{m\neq i}$을 l 이하로 짝지을 수 있는 i가 여럿 있을 때에는 이들 중 v_i의 확장경로가 가장 짧은 i를 선택한다. 이렇게 해서 구한 최소 확장비용이 최소확장 l-단말경로분할의 비용이 된다. 이상이 옳음은 보조정리 4에서 안다. □

알고리즘 1의 단계 8을 수행하기 위해서는 주어진 수열을 l 이하로 짝지을 수 있는지 판단할 수 있는 알고리즘이 필요하다. 다음 보조정리가 알고리즘을 제공한다.

보조정리 8. $a_1, a_2, \cdots, a_{2n}(a_1 \leq a_2 \leq \cdots \leq a_{2n})$을 l 이하로 짝지을 수 있는 필요충분조건은 $\forall k \leq n (k \in \mathbb{N})$ $a_k + a_{2n+1-k} \leq l$이다.

증명. 충분조건은 자명하다.

필요조건을 증명하기 위해, 짝지어진 실수의 합이 l을 넘지 않도록 하는 짝짓기 방법이 존재할 때, 그 중 a_1과 a_{2n}을 짝짓는 방법이 항상 존재함을 증명하자.

짝지어진 실수의 합이 l을 넘지 않도록 하는 짝짓기 방법 하나를 생각하자. 이 때 a_1과 $a_i (i \neq 1)$가 짝지어지고, $a_j (j \neq 2n)$와 a_{2n}이 짝지어진다고 하자. $i = 2n$이면 증명하고자 하는 명제가 성립한다. $i \neq 2n$이면 $i \neq j$, $a_j + a_{2n} \leq l$, $a_1 \leq a_j$, $a_i \leq a_{2n}$이 성립하므로, $a_1 + a_{2n} \leq l$, $a_i + a_j \leq l$이 성립하여 주어진 짝짓기 방법에서 a_1, a_j의 짝을 서로 바꾸어도 짝지어진 실수의 합이 l을 넘지 않는다.

증명한 명제와 수학적 귀납법에 의해 필요조건도 성립한다. □

따름정리 3. $a_1, a_2, \cdots, a_{2n+1}(a_1 \leq a_2 \leq \cdots \leq a_{2n+1})$에 대하여, $a_1, \cdots, a_{i-1}, a_{i+1}, \cdots, a_{2n+1}$을 l 이하로 짝지을 수 있으면 모든 정수 j $(i < j \leq 2n+1)$에 대해 $a_1, \cdots, a_{j-1}, a_{j+1}, \cdots, a_{2n+1}$도 l 이하로 짝지을 수 있다.

증명. 수열 $a_1, \cdots, a_{i-1}, a_{i+1}, \cdots, a_{2n+1}$을 $b_1, b_2, \cdots, b_{2n}$으로 첨자를 다시 매기자.

$a_1, \cdots, a_{j-1}, a_{j+1}, \cdots, a_{2n+1}$ 역시 마찬가지로 $c_1, c_2, \cdots, c_{2n}$으로 정의하면, a가 비감소수열이고 $i < j$이므로 모든 $k (1 \leq k \leq 2n)$에 대해 $b_k \geq c_k$가 성립한다. 따라서 보조정리 8에 의하여, $b_1, b_2, \cdots, b_{2n}$을 l 이하로 짝지을 수 있으면 $c_1, c_2, \cdots, c_{2n}$도 l 이하로 짝지을 수 있다. □

더욱이 따름정리 3에 의해, 단계 8에서 최소의 i를 선택할 때 이진 탐색을 사용할 수 있다.

보조정리 9. 문제 3을 $O(|V|\log|V|)$ 시간에 풀 수 있다.

증명. 앞에서 살펴본 바와 같이, 문제 3은 루트의 (하나뿐인) 자식 노드 v_0에 대하여 알고리즘 1을 수행하고 여기에 상수 시간 처리를 추가하여 해결할 수 있다.

알고리즘 1의 수행 시간을 살펴보면, 단계 5~6의 재귀호출 부분을 제외하고 $O(d\log d)$시간 (d는 노드의 차수)이 소요되는데, 이는 단계 7이 $O(d\log d)$ 시간에 수행 가능하며, 단계 8 역시 보조정리 8, 따름정리 3에 의해 $O(d\log d)$ 시간에 수행 가능하기 때문이다.

루트의 자식 노드 v_0에 대하여 알고리즘 1을 호출하면, 단계 5~6의 재귀호출에 의해 모든 루트 아닌 노드에 대하여 알고리즘 1이 최대 1회 호출되게 된다. 따라서 $v \in V$의 자식노드의 수를 d_v라 할 때, 문제 3을 최대 $O(|V| + \sum_{v \in V}(d_v \log d_v))$ 시간에 해결할 수 있으며, 이는 $O(|V|\log|V|)$를 넘지 않는다. □

정리 1. 트리 $T=(V,E)(|V| \geq 2)$에 대하여 문제 2를 $O(|V|^2)$ 시간에 풀 수 있다.

증명. 보조정리 5에 의해 문제 2의 해는 $O(|V|^2)$개의 실수 중 하나로 주어진다. 따라서 $O(|V|^2)$개의 후보에 대하여 이진 탐색을 수행하면, 문제 2의 해를 구할 수 있다.

후보들의 목록을 준비하는 데에는 보조정리 6에 의해 $O(|V|^2)$ 시간이 소요된다. 이진 탐색을 위해 중간값을 찾는 데에는 $O(|V|)$ 시간이 소요되며[9], 이진 탐색을 위한 비교연산은 문제 3을 푸는 것에 해당하므로 보조정리 9에 의해 $O(|V|\log|V|)$ 시간이 소요된다. 비교연산의 결과에 따라 중간값보다 큰 (혹은 작은) 후보들의 부리스트를 만드는 데에는 $O(|V|)$ 시간이 소요된다.

따라서 문제 2를 $O(|V|^2 + \log(|V|^2) \cdot (|V|+|V|\log|V|+|V|)) = O(|V|^2)$ 시간에 풀 수 있다. (중간값을 찾는 시간과 부리스트를 만드는 시간이 보수적으로 평가되었지만 전체 시간 복잡도에는 영향을 미치지 않는다.) □

시간 복잡도에 영향을 주지 않고, 알고리즘 1이 단말경로분할을 기록하도록 변경할 수 있으며, 따라서 동일한 알고리즘을 사용해 문제 1을 해결할 수 있다.

이상에서, 2개 이상의 노드를 갖는 트리 T의 최소단말경로분할을 구하는 $O(|V|^2)$ 시간 알고리즘을 얻었다.

5. 가중치가 없는 경우

[3]에서 제안된 원래의 문제는 트리에 가중치가 없으며, 경로의 가중치는 그 경로가 포함하는 에지의 개수로 주어진다. 이는 문제 1에서 $\forall e \in E\ w(e) = 1$인 특수한 경우이다.

정리 2. 트리 $T=(V,E)(|V| \geq 2)$에 대하여 모든 에지의 가중치가 동일한 경우, 문제 1을 $O(|V|\log^2|V|)$ 시간에 풀 수 있다.

증명. 이 경우, 문제 2의 해가 될 수 있는 후보의 집합은 $\{nc \mid n \in \mathbb{N},\ 1 \leq n < |V|\}$ $(c \in \mathbb{R})$로 주어진다. 그러므로 가중치가 있는 경우와는 달리 후보들의 목록을 준비할 필요가 없으며, 중간값을 결정하고 부리스트를 만

드는 조작을 상수시간에 수행할 수 있다.

따라서 문제 2를 $O((\log|V|)\cdot|V|\log|V|)=O(|V|\log^2|V|)$ 시간에 풀 수 있으며, 시간복잡도에 주는 영향 없이 알고리즘 1이 단말경로분할을 기록할 수 있으므로 문제 1 역시 같은 시간에 풀 수 있다. □

감사의 글

본 연구에 많은 도움을 주신 부산대학교 조환규 교수님과 논문을 개선할 수 있도록 조언해 주신 심사위원님들께 감사드립니다. 특히, 논문에서 제시한 알고리즘은 한 심사위원님의 의견에 따라 수행시간이 개선된 것입니다.

참 고 문 헌

[1] Chen, G. H., Kuo, M. T. and Sheu, J. P., "An Optimal Time Algorithm for Finding a Maximum Weight Independent Set in a Tree," *BIT Numerical Mathematics*, Vol.28, pp. 353-356, 1988.

[2] Garey, M. R. and Johnson, D. S., *Computers and Intractability: A Guide to the Theory of NP-Completeness*, pp. 194-195. W. H. Freeman and Co., New York, 1979.

[3] 한국정보문화진흥원, "버스 노선", 제 18회 한국 정보 올림피아드, 2001.

[4] Guruswami, V., Khanna, S., Rajaraman, R., Shepherd, B. and Yannakakis, M., "Near-optimal hardness results and approximation algorithms for edge-disjoint paths and related problem," *Proceedings of the 31st ACM STOC*, pp. 19-28, 1999.

[5] Lukes, J. A., "Efficient Algorithm for the Partitioning of Trees," *IBM Journal of Research and Development*, Vol.18, No.3, pp. 217-224, 1974.

[6] Neapolitan., R. and Naimipour, K., *Foundations of Algorithms*, pp. 363-392, D. C. Heath and Co., Lexington, 1996.

[7] Cormen, T. H., Leiserson, C. E. and Rivest, R. L., *Introduction to Algorithms*, pp. 86-94, the MIT Press, Cambridge, 1996.

[8] Chartrand, G. and Oellermann, O. R., *Applied and Algorithmic Graph Theory*, pp. 1-88, McGraw-Hill, New York, 1993.

[9] Blum, M., Floyd, R. W., Pratt, V., Rivest, R. L. and Tarjan, R. E., "Linear Time Bounds for Median Computations," *Proceedings of the 4th ACM STOC*, pp. 119-124, 1972.

안 형 찬

2006년 서울대학교 컴퓨터공학부 학사
2006년~현재 Cornell University Computer Science 박사과정. 관심분야는 알고리즘