Muş Alparslan Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi
1(1),16-21,2020

Minimum Baskın Küme Problemini Polinomsal Yöntemle Çözme

Ali KARCI
İnönü Üniversitesi, Bilgisayar Mühendisliği Bölümü, Malatya, Türkiye
ali.karci@inonu.edu.tr

Özet

Çizgelerde minimum baskın kümeyi elde etmek NP-Zor problem olup kesin çözümü bulan algoritmanın karmaşıklığı üstel artan bir bağıntıdır. Bu çalışmada minimum baskın kümeyi bulmak amacıyla çizgenin özel bir açılım ağacı elde edilmektedir ve o ağaç kullanılarak temel kesme kümeleri elde edilmektedir. Temel kesme kümeleri ile çizgenin düğüm dereceleri kullanılarak her düğümün baskınlık değeri elde edilir. Minimum baskın kümenin hepsi elde edilinceye kadar bu algoritma tekrar-tekrar uygulanır. Bu çalışmanın katkısı, bu algoritmanın geliştirilmiş olmasıdır.

Anahtar Kelimler: Baskın Kümeler, Temel Kesme-Kümeleri, Etkili Algoritmalar

Solving the Minimum Dominating Set Problem by Polynomial Method

Abstract

Obtaining the minimum dominating set in graphs is an NP-Hard problem, and the complexity of the algorithm that finds the exact solution is an exponentially increasing relation. In this study, in order to obtain the minimum dominating set, a special spanning tree of the graph is obtained and the fundamental cut-sets are obtained by using this tree. The dominance value of each node is obtained by using the fundamental cut-sets and node degree in given graph. This algorithm is applied over and over until the minimum dominating set is obtained. The contribution of this paper is to develop this algorithm.

Keywords: Dominating Sets, Fundamental Cut-Sets, Efficient Algorithms.

1. GİRİŞ

Çizgeler iki kümeden meydana gelirler ve bu kümeler denklem ve değişkenleri, nesne ve aralarındaki ilişkiler, bireyler ve akrabalıklar, işçi ve çalıştırdığı makineler gibi problemleri modellemek için kullanılan matematiksel modellerdir. Çizgeler kullanılarak yapılan modellemeler sonucunda bilgisayar ağları, Bayesian ağları, sosyal ağlar v.b. özel çizgeler elde edilmiştir. Çizge ile modellenebilecek çok sayıda problem ve uygulama bulunabilir.

Tanım 1: Bir çizge G = (V,E) şeklinde V düğümler kümesi ve E ayrıtlar (kenarlar) kümesinden oluşur. Paralel ve döngü ayrıt içermeyen çizgelere basit çizgeler denilir.

Çizgelerle ilgili olarak bağımsız kümeler, iki-parçalılık, baskın küme, düğüm kapsama, maksimum hizip, mükemmel eşleştirme, düğüm renklendirme v.b. gibi problemler bulunmaktadır ve aynı zamanda bu problemlerin birçoğunun sosyal hayatta karşılıkları vardır. G=(V,E) bir çizge olmak üzere $B \subseteq V$ gibi bir B düğümler kümesi bulunmak istenir. Çizgenin bütün düğümleri ya B kümesinin elemanıdırlar ya da V-B kümesinin elemanı olup en az B kümesinden bir düğüm ile komşudurlar. Bu durumda B kümesine baskın küme denilir. B kümesinin minimum elemanlı şekline minimum baskın küme denir.

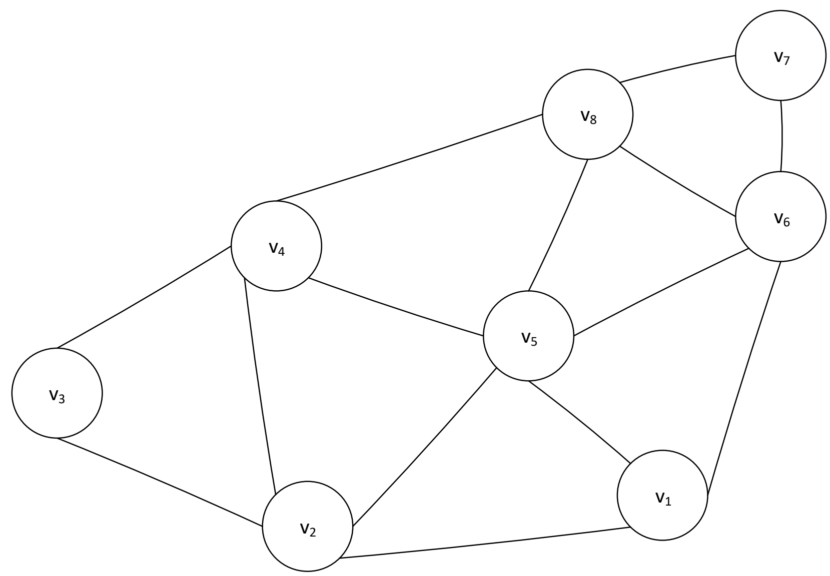
Minimum baskın küme üzerinde çok sayıda çalışmalar yapılmıştır. Bölünmüş çizgelerde baskın-χ-renk sayısının sınır üzerine çalışmalar yapılmıştır [2]. Verilen çizgenin baskın kümesini inşa etmeye çalışan yayınlar bulunmaktadır [1]. Minimum baskın küme sayısı üzerine yapılan çalışmalar bulunmaktadır [3]. Baskın kümeler ve tek tip dağınıklıklar [10], baskın küme karmaşıklığı için algoritma [11], bağımsız baskınlık [5], her tip çizge için tekil minimum baskın kümeler [6], belli tip çizgeler için etkili algoritmalara [4], herhangi bir çizge için minimum baskın kümeyi bulmak [7] gibi çalışmalar bulunmaktadır.

Bu çalışmanın amacı verilen çizge için minimum baskın kümeyi bulan etkili algoritmayı geliştirmektir. Bu çerçevede [7-9] çalışmalarındaki yaklaşımlar kullanılacaktır.

2. VERİLEN ÇİZGENİN BİR ÖZEL AÇILIM AĞACINI İNŞA ETME

Devre içermeyen ve bağlı olan çizgelere ağaç denilmektedir. Verilen çizgenin bütün düğümlerini barındıran ağaca ilgili çizgenin açılım ağacı denir. Minimum baskın kümeyi bulmak amacıyla çizgenin bir özel açılım ağacı tanımlanmaktadır ve bu ağaç ilk olarak [7-9] çalışmalarında verilmiştir.

Tanım 2: Bir açılım ağacı verilen çizgenin bütün düğümlerini içeren bağlı olmak kaydıyla minimum ayrıt içeren bir alt çizgedir.



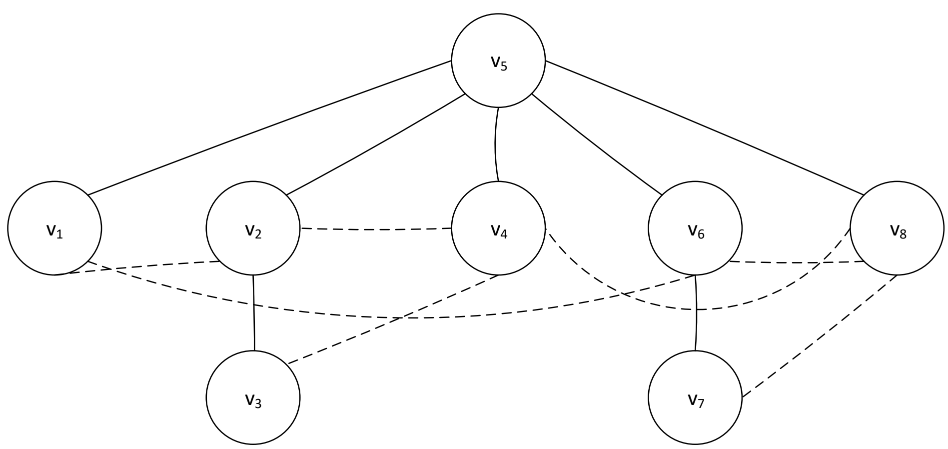
Şekil 1. 8 düğümlü pençesiz çizge.

Bu çalışmada Karcı [7-9] tarafından tanımlanan açılım ağacı kullanılacaktır. Karcı tarafından tanımlanan açılım ağacına Kmax ağacı denilmektedir. İlk olarak çizgenin düğüm derecelerine göre en yüksek dereceye sahip olan düğüm, kök düğüm olarak seçilir. Ondan sonra kök düğümün bütün komşuları ağaca eklenir. Kök düğümünü çocukları derecelerine göre değerlendirilir (açılmamış derecelerine göre). En yüksek dereceye sahip olan düğüm açılır (açılmamış komşuları ağaca eklenir). Bu şekilde üç seviyeli bir ağaç oluşmuş olur. Açılmamış düğümler arasında derecesi en yüksek olan düğüm açılır. En yüksek dereceye sahip birden fazla düğüm varsa, seviye olarak kök düğüme yakın olan düğüm açılır. Bu şekilde elde edilen ağaca Kmax (Karcı maksimum açılım ağacı) ağacı denilmektedir. Algoritma 1'de Kmax ağacını oluşturan algoritma görülmektedir.

Algoritma 1 kullanılarak Şekil 1'deki çizgenin Kmax ağacı oluşturulurken ilk olarak v_5 düğüm kök düğüm olarak seçilir, çünkü derecesi en yüksek olan düğüm v_5 düğümdür. Bunun sonucunda v_1, v_2, v_4, v_6 ve v_8 düğümler kök düğüme çocuk olarak eklenir. Bu yeni eklenen düğümlerin açılmamış dereceleri sırası ile $d(v_1)=0$, $d(v_2)=1$, $d(v_4)=0$, $d(v_6)=1$ ve $d(v_8)=0$ şeklindedir. Bir sonraki seviye için v_2 ve v_6 düğümler açılır. Bu şekilde Kmax ağacı elde edilmiş olur; elde edilen ağaç Şekil 2'de görülmektedir. Kesikli çizgi ile gösterilen ayrıtlar ağacın ayrıtları olmayıp tümler açılım ağacının ayrıtlarıdır (bu ayrıtlara kiriş denir).

Algoritma 1: Kmax ağacı oluşturma

```
Kmax_Ağacı(A,AT,D)
Q←∅
r←max(D) // D derece matrisi
T=(V,E1) and E1←∅
i←1,…, |V|
    j←1,…,|V|
        AT(i,j)←0
EnQueue(Q,r)
Q2←∅
for i←1,…,|N(r)|
    EnQueue(Q2,vi),    vi∈N(r)
    if AT(r,s)=0 then
       AT(r,s)=1, AT(s,r)=1
       EnQueue(Q2,s)
while Q2≠∅
    v←DeQueue_Max(Q2,A,AT)
    EnQueue(Q,v)
    for i←1,…,|N(v)|
        if not(vi∈Q2) and vi∈N(v)
            EnQueue(Q2,vi)
            AT(v,vi)=1, AT(vi,v)=1
    Remove_Zero(Q2,A,AT)
```



Şekil 2. Şekil 1’deki çizgenin Kmax ağacı (kesikli ayrıtlar kirişi temsil etmektedir).

Kmax ağacı 8 düğüme ve 7 ayrıta sahip bir açılım ağacıdır. Bu durumda 7 tane temel kesme kümesinin elde edilmesine sebep olur. Elde edilen temel kesme kümeleri Şekil 3’te görülmektedir.

3. DÜĞÜMLERİN BASKINLIK DEĞERİNİ HESAPLAMA

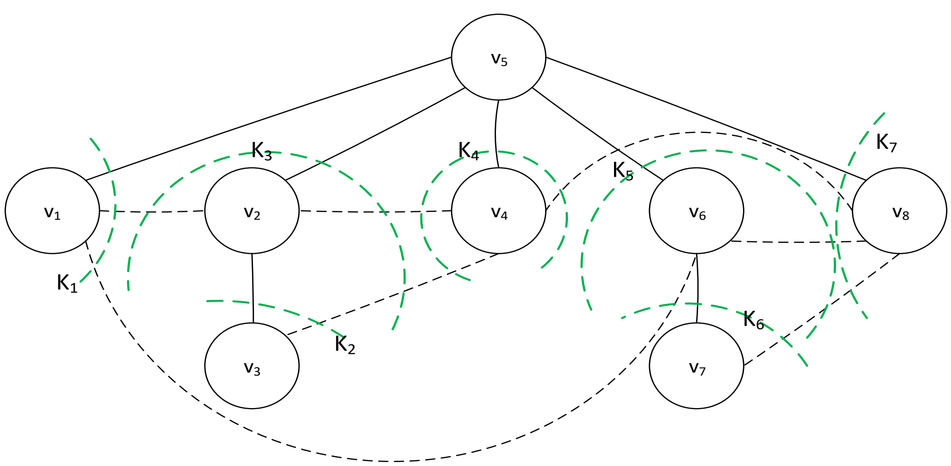
G=(V,E) bir çizge olmak üzere, Ec⊆E kümesindeki ayrıtların silinmesi ile çizge en az iki parçaya bölünüyorsa, bu kümeye kesme kümesi denir. Açılım ağacının sadece bir dalını ve geriye kalanları kiriş olarak içeren kesmelere temel kesme kümesi denir. Açılım ağacındaki dal sayısı kadar temel kesme kümesi elde edilir. Bu çalışmada temel kesme kümeleri iki gruba ayrılmaktadır. Birincisi yaprak temel kesme kümeleri olup sadece bir düğümün çizgeden koparılmasını sağlar. İkincisi ise dâhili temel kesme kümeleri olup çizgeyi iki alt çizgeye ayıran temel kesme kümeleridir. Bu kesme küme tanımları [7-9] çalışmalarında ilk kez olarak ileri sürülmüş ve algoritmaya dönüştürülmüştür.

Algoritma 2: Temel kesmeleri oluşturma

```
GenerateInternalCut(AT,A,C,B,e)
Assume e=(u,v)
Q2←∅
EnQueue(Q2,u)
while Q2≠∅
    u←DeQueue(Q2)
    EnQueue(Q,u)
    for i←1,…,n      //number of nodes in G=(V,E)
        if AT(u,i)=1 and i≠v
            EnQueue(Q2,i)
while Q≠∅
    u←DeQueue(Q)
    for i←1,…,n // n is the number of nodes in G=(V,E)
        if A(u,i)=1 and i∉Q
            C(u,k)=B(u,k) // k illustrates the edge e=(u,i)
```

Yaprak temel kesme kümeleri: K_1, K_2, K_4, K_6, K_7 şeklindedir.
$K_1=\{(v_1,v_2), (v_1,v_5),(v_1,v_6)\}$
$K_2=\{(v_2,v_3), (v_3,v_4)\}$
$K_4=\{(v_2,v_4), (v_3,v_4), (v_4,v_5),(v_4,v_8)\}$
$K_6=\{(v_6,v_7), (v_7,v_8)\}$
$K_7=\{(v_4,v_8), (v_5,v_8), (v_6,v_8), (v_7,v_8)\}$

Dahili temel kesme kümeleri: K_3, K_5 şeklindedir.
$K_3=\{(v_1,v_2), (v_2,v_5), (v_2,v_4), (v_3,v_4)\}$
$K_5=\{(v_1,v_6), (v_5,v_6), (v_6,v_8), (v_7,v_8)\}$



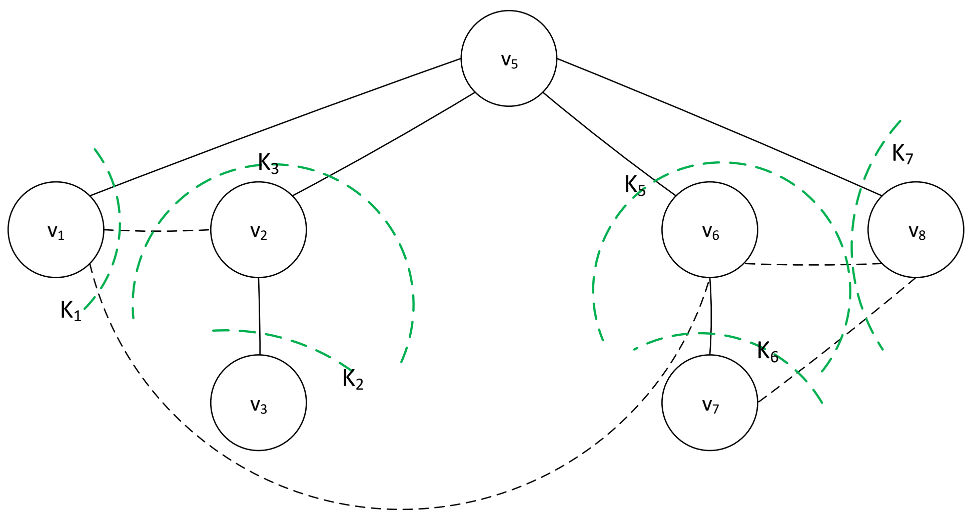
Şekil 3. Kmax ağacını kullanılarak oluşturulan yaprak ve dâhili temel kesme kümeleri.

Başlangıçta bütün düğümlerin baskınlık değeri sıfırlanır. Her temel kesmenin içerdiği ayrıtların uçlarındaki düğümlerin baskınlık değerine 1 eklenir. Bu şekilde bütün temel kesme kümeleri için bu işlem tekrarlanır. Bunun sonucunda her düğüm için elde edilen değere düğümün ilgili çizgedeki derecesi eklenerek düğümün baskınlık değeri (Γ) elde edilmiş olur. Elde edilen baskınlık değerlerinden en yüksek olan hangi düğüme aitse, o düğüm baskın kümenin elemanı olarak seçilir ve komşuları gri kümeye atılır. Seçilen düğüm çakışık olan ayrıtlarla beraber Kmax ağacından silinir. Tekrar baskınlık değerleri hesaplanır. Bunun sonucunda elde edilen baskınlık değerlerinden en yüksek olan hangi düğüme aitse, o düğüm baskın kümeye eklenir. Bu işlem baskınlık kümesi ile gri kümenin birleşiminin çizgenin düğümler kümesine eşit oluncaya kadar devam edilir. Algoritma 3 bu işlemleri gerçekleştirmektedir.

Algoritma 3: Baskınlık Değerini Hesaplama

1) G=(V,E) ve T=Kmax_Tree(A,AT,D)
2) B çakışım matrisi (G) ve C_{max} ise Kmax için kesme matrisi
3) $E_{max} = B * C_{max}^T$
4) i←1,…,n
5) $\pi(v_i) = 0$
6) j←1,…,m
7) $\pi(v_i) = \pi(v_i) + E_{max}(i,j)$
8) i←1,…,n
9) $\pi(v_i) = \pi(v_i) + d_G(v_i) + d_T(v_i)$

Şekil 1'de verilen çizgenin minimum baskın kümesini elde etmek için π baskınlık değer vektörü olmak üzere π=[9 10 6 12 10 11 6 12] olur. İki düğümün baskınlık değeri 12 olup bunlardan bir tanesi seçilecektir. v_4 düğümün seçildiği kabul edilirse, açılım ağacının v_4 düğüm silindikten sonraki hali Şekil 4'te görülmektedir.



Şekil 4. Kısmi Kmax ağacı.

Baskınlık kümesi D={v_4} ve N(D)={v_2,v_3,v_5,v_8} şeklinde olur. Şekil 4^teki kısmi Kmax ağacına göre tekrar baskınlık değerleri hesaplanır. Elde edilecek baskınlık değerleri π=[8 8 3 – 9 10 6 10] şeklinde olur. v_6 ve v_8 düğümleri en yüksek baskınlık değerlerine sahipler, fakat v_8 gri listede yer aldığı için seçilemez. Geriye sadece v_6 düğümü kalır. D={v_4,v_6} ve N(D)= {v_1,v_2,v_3,v_5,v_7,v_8} şeklinde olur. D∪N(D)= {v_1,v_2,v_3,v_4,v_5,v_6,v_7,v_8}=V olduğundan işlem sonlandırılır ve D kümesi çizgenin minimum baskın kümesi olur.

4. SONUÇLAR

Bu makale çalışmasında NP-Zor olan minimum baskın kümeyi bulma problemi için etkili bir algoritmanın önerilmesi ve örnek üzerinde çalışmasının gösterilmesini içermektedir. Bu amaçla geliştirilen çizgenin özel bir açılım ağacı olan Kmax ağacı (Karcı maksimum açılım ağacı) elde edilir ve bu ağaç kullanılarak çizgenin temel kesme kümeleri elde edilir. Temel kesme kümeleri düğümlerin baskınlık değerlerinin hesaplanmasında kullanılır ve onun sonucunda düğümlerin dereceleri eklenir. Nihai olarak düğümlerin baskınlık düğüm değerlerinin elde edilmesi için oluşturulan algoritmalar yeni algoritmalardır.

KAYNAKÇA

1. Alikhan, S., Peng, Y.-H., “Construction of Dominating Sets of Certain Graphs”, Journal of Discrete Mathematics, Vol:2013, Article ID:587196, 2013.
2. Bresar, B., Movarraei, N., “On the number of maximal independent sets in minimum colorings of split graphs”, Discrete Applied Mathematics, Vol:247, pp:352-356, 2018.
3. Connolly, S., Gabor, Z., Godbole, A., Kay, B., Kelly, T.,”Bounds on the Maximum Number of Minimum Dominating Sets”, Discrete Mathematics, Vol:339, pp:1537-1542, 2016.
4. Deng, Y.-P., Sun, Y.-Q., Liu, Q., Wang, H.-C.,”Efficient Dominating Sets in Circular Graphs”, Discrete Mathematics, Vol:340, pp:1503-1507, 2017.
5. Goddard, W., Henning, M.A., “Independent domination in Graphs: A Survey and Recent Results”, Discrete Mathematics, Vol: 313, pp:839-854, 2013.
6. Golovach, P.A., Heggernes, P., Kante, M.M., Kratsch, D., Villanger, Y.,”Enumerating Minimal Dominating Sets in Chordal Bipartite Graphs”, Discrete Applied Mathematics, Vol:199, pp:30-36, 2016.
7. Karci, A., Karci, Ş.,”Determination of Effective Nodes in Graphs”, International Conference on Science, Engineering & Technology, Mecca, Saudi Arabia, pp:25-28, 2020.
8. Karci, A., “Finding Innovative and Efficient Solutions to NP-Hard and NP-Complete Problems in Graph Theory”, Anatolian Science – Journal of Computer Science, Vol:5, pp:137-143, 2020a.
9. Karci, A., “New Algorithms for Minimum Dominating Set in Any Graphs”, Anatolian Science – Journal of Computer Science, Vol:5, pp:62-70, 2020b.
10. Marti-Farre, J., Mora, M., Ruiz, J. L.,”Uniform Clutters and Dominating Sets of Graphs”, Discrete Applied Mathematics, Vol:263, pp:220-233, 2019.
11. Rooij, J.van, Bodlaender, H.L., “Exact algorithms for dominating set”, Discrete Applied Mathematics, Vol:159, pp:2147-2164, 2011.