Yüksek-Derece Çizge Yapıları ile Maksimum Klik Sayma Problemine Yönelik Buluşsal Yöntemleri Öğrenme
Learning Heuristics for the Maximum Clique Enumeration Problem Using Higher-Order Graph Structures

Ali Baran Taşdemir
Bilgisayar Mühendisliği Bölümü
Hacettepe Üniversitesi
Ankara, Türkiye
alibaran@tasdemir.us

Lale Özkahya
Bilgisayar Mühendisliği Bölümü
Hacettepe Üniversitesi
Ankara, Türkiye
ozkahya@cs.hacettepe.edu.tr

***Özetçe* —Günümüzde birçok NP-zor kombinatoryal optimizasyon sorusu karmaşık öğrenme modellerini kullanarak öğrenen buluşsal yöntemler ile çözülmektedir. Özellikle, çizgelerde düğüm sınıflandırma yöntemi, optimal kümede bulunan ve bulunmayan düğümler arasındaki karar sınırını bulmaya yönelik kullanılagelmiştir. Bu çalışmada, bir düğümü içeren lokal çizgecik sayılarının çizge özniteliği olarak, maksimum klikleri sayma problemini çözmedeki rolü araştırılmaktadır. Çizgecikler, lokal ve global frekansları ağ analizinde önemli öznitelikleri arasında olan küçük ölçekli geren altçizgelerdir. Burada, herhangi bir maksimum klik içinde bulunan düğümleri diğerlerinden ayırdetme işlemi bir öğrenme çerçevesi içinde yapılmaktadır. Bunun sonucunda, bu yöntem ile maksimum klik sayma probleminin hesaplama süresinin azaltılmasına yönelik bir budama işlemi gerçekleşmektedir. Elde edilen sonuçların yüksek doğruluk oranı yanında sağlam ve ölçülebilir olduğu gözlenmiştir. Burada sunulan yöntem farklı ölçekte her ağa uygulanabilir olmakla beraber, maksimum klik sayma dışında çizge yapılarını aramaya yönelik karmaşıklığı yüksek başka sorular için de yaklaşık çözümler bulmada kullanılabilir.**

Anahtar Kelimeler—maksimum klik sayma problemi, düğüm sınıflandırma, makine öğrenme modeli, çizgecik.

***Abstract*—Recently, various NP-hard combinatorial optimization problems have been solved by learned heuristics using complex learning models. In particular, node classification in graphs has been a helpful method towards finding the decision boundary to distinguish nodes in an optimal set from the rest. In this work, we investigate the role of local graphlet counts surrounding a node as graph features in the node classification towards solving the maximum clique enumeration problem. Graphlets are small induced subgraphs, whose local and global frequencies have been important features in the analysis of networks. We use a learning framework to identify the nodes that belong to some maximum clique of the network. Consequently, this idea is used in a pruning process to reduce the runtime of the maximum clique enumeration problem. Besides the high accuracy of the results, the performance of this framework is shown to be scalable and robust. The method presented here is applicable to networks from all sizes and can be used in estimating the solution of other graph search problems with high complexity.**

Keywords—Maximum clique enumeration problem, node classification, machine learning model, graphlet.

I. GİRİŞ

Gezgin satıcı problemi (traveling salesman problem), maksimum klik problemi gibi birçok kombinatoryal optimizasyon probleminin NP-zor olduğu bilinmektedir. Klik, düğümlerin tümünün birbiriyle komşu olduğu altçizgedir ve maksimum klik ise çizgede olası en fazla düğüme sahip kliktir. Örnek olarak, üçgen üç düğümlü bir kliktir ve Facebook ağı gibi sosyal ağlarda, arkadaşlık ilişkileri hakkında bilgi edinmeye yönelik sıklığı ve dağılımı en fazla incelenen yapılardan biridir. Maksimum klik problemi, [1], [2]'deki örneklerde olduğu gibi, yaygın olarak çalışılan bir NP-tam (NP-complete) problemdir ve verilmiş bir çizgedeki maksimum klik büyüklüğünü bulmayı amaçlar. Genel olarak, bir karmaşık ağda, birden fazla maksimum klik bulunmaktadır ve *maksimum klikleri sayma (MKS) problemi*, maksimum klik problemini çözmeye ek olarak bu kliklerin her birinin bulunmasını hedefler. Bu anlamda maksimum klik probleminden daha da zordur. Maksimum klikleri bulmak, ağın içinde nerede yoğunluk olduğu ve düğümler arasındaki ilişkiler hakkında da bilgi verdiği için, ağın zamanla değişimini anlamak ve tahminlerde bulunmak açısından da araştırılmaktadır. Bu problemin gerçek hayat uygulamaları sosyal [3], davranışsal [4], finansal [5] ve dinamik [6] ağlarda görülmektedir.

Hesaplama açısından NP-zor olan bu tür kombinatoryal sorulara makine öğrenmesi yöntemiyle getirilen yaklaşım ilk olarak [7]–[9] çalışmalarında kullanılmıştır. Bu çalışmalarda, makine öğrenmesi tekniklerinin düğüm sınıflandırmasında kullanılmasıyla, MKS problemi için oldukça pahalı arama algoritmalarının gerektirdiği hesaplama zamanını azaltabilmesinin mümkün olabileceği gösterilmiştir.

Bu çalışmada da benzeri şekilde, maksimum klik sayma (MKS) problemine benzeri şekilde buluşsal (heuristic) bir yöntemle çözüm bulmak için düğüm sınıflandırmasını amaçlayan bir öğrenme modeli kullanılmaktadır. Bir çizgede, maksimum klik boyutunun ne olduğu, düğümlerin bir maksimum klikte yer alıp almamasına göre ikili sınıflandırma (binary classification) problemi olarak tanımlanmıştır. Bu amaçla, her düğüm için belirli çizgesel özniteliklerin bulunduğu bir öznitelik vektörü tanımlanmış ve makine öğrenmesi yöntemleriyle, sınıflandırıcı algoritmalar gerçek ağ kümeleri üzerinde eğitilmiştir. Bu yaklaşımdaki çizgesel öznitelikler kümesinde temel olarak, çizgenin yapısal özellikleri hakkında daha önemli bilgileri taşıyan ve Bölüm II'de bahsedilen çizgecik dağılımı bilgisi kullanılmaktadır. Çizgecik (graphlet), küçük (genelde 4 ile 10 arasında düğüm içeren) ve gerili (induced) bir altçizgedir. Çizgecikler, karmaşık ağlarda modelleme ve tahmin sorularına yönelik biyoinformatik [10], keminformatik [11], görüntü işleme and bilgisayarlı görü [12], [13] gibi birçok alanda kullanılmaktadır. Örnek olarak, çizgecik yoğunluğu çizge sınıflandırma problemi ve çizge örüntülerini bulmaya yönelik birçok farklı amaç için önemli bilgi sağlar. Literatürden verilen örneklerde de görüldüğü gibi, çizgeciklerin yerel olarak nerelerde yoğun olduğu bilgisi, birçok makine öğrenmesi ve ağ analizi çalışmasında performansı artırmıştır. Bu açıdan, kullandığımız yöntem MKS probleminde düğüm sınıflandırma yaklaşımı kullanılarak getirilmiş çözümleri geliştirmektedir.

İzlediğimiz yöntemde, herhangi bir maksimum klik içinde bulunan düğümleri diğerlerinden ayırdetme işlemi, makine öğrenme yöntemleri çerçevesinde ikili sınıflandırma sorusu olarak tanımlanmaktadır. Sınıflandırıcılar, verilen her çizge için, çizgenin büyüklüğünü klik bulma işleminden önce küçültmeye yarayan ön işleme (preprocessing) aşamasında da kullanılarak, MKS'nin hesaplama zamanını kısaltacaktır. Bu süreçte, çizgecik olarak tanımlanmış, yüksek-dereceli çizge yapılarının lokal (bir düğüm etrafındaki) frekanslarının, doğruluk oranına (accuracy) etkisi analiz edilmiş ve bu oranı artırdığı gözlenmiştir. Elde edilen sonuçların yüksek doğruluk oranı yanında sağlam (robust) ve ölçülebilir (scalable) olduğu gösterilmiştir. Burada sunulan yöntem farklı ölçekte her ağa uygulanabilir olmakla beraber, maksimum klik sayma dışında başka ağ davranışlarını incelemeye ya da çizge yapılarını aramaya yönelik karmaşıklığı yüksek başka soruların da çözümünde kullanılabilir.

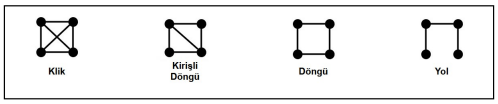
II. METOT

A. Ön İşleme (Budama) ve Sınıflandırma İşlemi

Ön işleme yöntemi, başlangıç çizgesinde maksimum büyüklükteki kliklerden hiçbirinde bulunmadığı öngörülen düğümlerin çizgeden silinmesi ile gerçekleşir. Bu yöntem kısaca budama (pruning) olarak adlandırılır. Bu sayede, çizgenin boyutu %50-60'lara varan miktarlarda küçültülerek hesaplama ve arama zamanından kazanılır. Bilinen yöntemlerden biri olan Derece Yöntemi'nde, k büyüklüğündeki klik (k-klik) aranması durumunda, derecesi k-1'den küçük düğümler herhangi bir k-klikte yer alamayacağı için silinir ve kalan çizge işleme alınır [14]. Öte yandan, [7]'de derece yöntemiyle silinen düğümlerin oranının, aşağıda sunulan olasılıksal sınıflandırma yardımıyla yapılan budamayla silinenlerin oranına göre düşük kaldığı gösterilmiştir. Bu sebeple, ikinci yöntem tercih edilecektir.

Olasılıksal sınıflandırma işlemiyle budama yönteminde amaç, $G = (V, E)$ çizgesinde, V düğüm kümesi üzerinde $\beta : V \rightarrow \{0, 1\}$ olarak ikili (binary) bir sınıflandırıcı elde etmek ve düğümleri bir maksimum klikte yer alıp almama olasılığına göre ikiye ayırmaktır. Eğitme işlemi için düğümlerin bir altkümesi örnek olarak seçilip, eğitme kümesi, $T = \{< f(v_i), y_i >\}$ oluşturulur. Burada, y_i her v_i düğümü için 0 ya da 1 olan sınıf değerini, $f : V \rightarrow \mathbb{R}^d$ ise her düğüm için, d elemanlı öznitelik vektörünü temsil eder. Öncelikle, bir *güvenlik eşik değeri (confidence threshold)* $q \in [0, 1]$ tanımlanır. Her $u \in V$ düğümü için elde edilen $f(u)$ vektörüne bir olasılık değeri verilecek şekilde, bir olasılıksal sınıflandırıcı P tarafından tüm düğümler için olasılık dağılımı verilir. Bu dağılım sonucu, her $u \in V$ için $f(u) > q$ olduğunda, u düğümü V' adında bir altkümeye yerleştirilir. Bu işlem tüm düğümler için yapıldıktan sonra, V''de toplanan düğümler çizgeden silinir. Silinen düğümler için $\beta = 0$, kalanlar için de $\beta = 1$ değeri verilerek sınıflandırma yapılır. Bu şekilde maksimum bir klikte yer alması öngörülmeyen düğümlerin silinmesiyle çizgeyi küçültme işlemine *budama (pruning)* denir ve *budama oranı* $|V'|/|V|$'ye eşittir. Eşik değeri q ile budamanın oranı kontrol edilebilmekte, q küçüldükçe budanan düğüm sayısı da artmaktadır. Ön işlemede, herhangi bir maksimum klikte yer alan bir köşenin silinmesindense, hiçbir maksimum klikte yer almayan köşelerin kalması tercih edilir. Bu anlamda, q değerinin gerekenden küçük seçilmemesi de önemlidir.

Sınıflandırıcının eğitiminde, Destek Vektör Makinesi (SVM), Lojistik Regresyon, Rastgele Orman ve Stokastik Gradyan İnişi (SGD, Stochastic Gradient Descent) algoritması olmak üzere dört farklı makine öğrenmesi algoritması denenmiştir. Bu sınıflandırıcıların uygulama aşamasında, sklearn otomatik sisteminden faydalanılarak parametreler optimize edilmiştir. F1-skorları aracılığıyla, doğruluk oranı (accuracy) değerlerinin SVM, Lojistik Regresyon, Rastgele Orman ve SGD algoritmaları için sırasıyla 0.82, 0.91, 0.90, ve 0.87 olduğu gözlemlenmiştir. Bu sonuçlar doğrultusunda, bu çalışmadaki öğrenme işleminde 5 katlamalı çapraz doğrulama kullanılarak lojistik regresyon yöntemi ile sınıflandırma yapılması uygun görülmüştür.



Şekil 1: Lokal frekansları öznitelik olarak kullanılan çizgecikler.

B. Çizgesel Öznitelikler:

Çizgesel öznitelikler, (F1) düğüm sayısı, (F2) kenar sayısı, (F3) düğüm derecesi, (F4) yerel kümeleşme katsayısı, ve (F5) özmerkeziliktir. İlk iki öznitelik adından da anlaşılmaktadır. (F3) bir düğümün derecesi sahip olduğu komşu sayısıdır. Yerel kümeleşme katsayısı, (F4), burada 3.dereceden olup her v düğümü için, v'yi içeren üçgen (3 düğümlü klik) sayısının, v'nin komşu çiftlerine oranıdır. Bu öznitelik, v'nin komşularının kendi aralarındaki kenar yoğunluğu hakkında bilgi verir. Son olarak, (F5) bir düğüm için ne kadar büyük olursa, o düğümün çizgeye bağı (connectedness), başka deyişle diğer düğümlerle arasındaki mesafenin kısalık ölçüsü, o kadar yüksektir. Bir G çizgesinin *özvektör merkeziliği (eigenvector centrality),* G'nin komşuluk (adjacency) matrisi A'nın en büyük özdeğeri s için,

$A\mathbf{v} = s\mathbf{v}$ eşitliğini sağlayan $\mathbf{v}$ özvektörüdür. Komşuluk matrisinde, x düğümüne karşılık gelen indeks i ise, $\mathbf{v}$ vektörünün i hanesindeki değer, x'in *özmerkeziliğidir (eigencentrality).* Bu değer, çizgenin yüksek yoğunluklu altçizgelerindeki düğümler için daha büyük olmaktadır.

Lokal frekanslarının öznitelik olarak kullanılacağı çizgecikler Şekil 1'de listelenmiştir: 4-klik (4-clique), kirişli 4-döngü (chordal 4-cycle), kuyruklu üçgen (tailed triangle), 4-döngü (4-cycle), 4-yol (4-path). Bir H *çizgeciğinin lokal frekansı*, H'nin konu edilen çizge bölgesindeki yoğunluğunu ölçmeye yarayan bir parametredir. Buradaki lokal frekans, her v düğümü için, onu içeren tüm altçizgelerin içinde H'nin kopyası olanların sayısıdır ve *H'nin v derecesi* olarak adlandırılmaktadır. Bu frekanslar [15], [16]'daki algoritma temel alınarak hesaplanmış, onların elde ettiği kenarlar üzerinden yapılan sayım, düğümler üzerindeki frekansları verecek şekilde adapte edilmiştir.

III. Deneysel Sonuçlar

Her ağın içindeki kliklerin bulunması işlemi, halihazırda en yaygın kullanılan ve "branch-and-bound" [17] yöntemine dayalı cliquer [18] algoritması ile gerçekleştirilmiştir. Bu amaç için 16GB RAM'e sahip Debian 9 sistemiyle çalışan bir bilgisayar, diğer tüm işlemler için ise 8GB RAM'e sahip Windows 10 işletim sistemine sahip bir bilgisayar kullanılmıştır. Bölüm III-B'deki sonuçlar için maksimum klik büyüklüğü, Bron-Kerbosch algoritmasını [19] kullanan igraph [20] tarafından Windows 10 sisteminde 8GB RAM'e sahip Intel Core i5-7300HQ (2.50GHz) işlemciyle hesaplanmıştır.

A. Düğüm sınıflandırma doğruluğu

Sınıflandırıcı algoritma, 36 ağdan oluşan biyolojik ağlar [21] kümesi üzerinde eğitilmiş, test aşaması için bio-WormNet-v3 ağı kullanılmıştır. Eğitim kümesindeki her bir G_i ağı için, bütün maksimum klikler $\mathcal{C}_i = \{C_1, C_2, .., C_n\}$ olarak sıralanmıştır. Maksimum klik içerisinde bulunan bütün düğümler sınıflandırıcı için sınıf-1 olarak etiketlenir. Ayrıca dengeli bir veri seti oluşturmak amacıyla, her ağ için maksimum klik düğüm sayısının 1.5 katı kadar rastgele klik içerisinde bulunmayan, $G_i \setminus \mathcal{C}_i$'den alınan örnek düğümler eklenecektir. Ve bu düğümler sınıflandırıcı için sınıf-0 olarak etiketlenir. Bunun sonucunda, biyolojik ağlar için elde edilen eğitim kümesi 2522 öznitelik vektöründen oluşmaktadır. Budama işleminde $q = 0.55$ olarak alınmıştır.

ω	ω_t	ω [7]	P_d	P_t	P [7]
121	98	90	0.68	0.94	0.90

TABLO I: Budama aşamasında $q = 0.55$ kullanılarak bio-WormNet-v3 ağı ($|V| = 16K$, $|E| = 763K$) üzerinde elde edilen değerler.

Test aşamasında, bio-WormNet-v3 ağı üzerinde elde edilen değerler, Tablo I'de sunulmuştur. Karşılaştırıldığında, sınıflandırıcımızın bulduğu ω_t'nin gerçek değer olan ω'ya [7]'de tahmin edilen değerden çok daha yakın olduğu görülmektedir. Ek olarak, bu performansa ulaşırken %94 oranında silme işlemi sonucu, [7]'dekinden (silme oranı: %90) daha küçük bir ağda işlem yapılmıştır. Kullandığımız önişleme yönteminin, *derece yöntemi* olarak bilinen (Fast Max-Clique Finder (fmc) [14]) ve derecesi belirli bir k sayısının altında olan düğümlerin silinmesiyle elde edilen budama oranından da ($P_d = \%68$) çok daha iyi performansa sahip olduğu gözlemlenmektedir.

B. Sağlamlık ve Ölçülebilirlik

Bu bölümde, kullandığımız sınıflandırma modelinin ne kadar sağlam (robust) ve ölçülebilir (scalable) olduğunu değerlendirmeye yönelik, rastsal çizgeler üzerinde doğruluk oranı (accuracy) ve hız artışı analiz edilmektedir. Bir rastsal $G(n, p)$ çizgesinin tanımında, n düğüm sayısını ve p de her düğüm çifti arasında bir kenarın olma olasılığını verir [22]. Klik yerleştirilmesi probleminde (planted clique problem), $G(n, p)$ çizgesinin, yine rastsal olarak k tane düğüm seçilip, bu düğümlerdeki tüm çiftlerin arasına kenar eklenmesiyle klik olması sağlanır ve amaç bir k-klik bulmaktır.

Rastsal çizge örneklerinde k büyüklüğünde bir klik yerleştirmek için k tane düğüm eşit olasılıkla (uniformly at random) seçilir ve aralarındaki tüm kenarlar çizgeye eklenerek bir *k-klik* oluşturulur. Olasılık hesabıyla, $k \leq log_2 n$ koşulu sağlandığında $G(n, p)$ rastsal çizgesinde neredeyse her zaman (almost surely) k büyüklüğünde bir klik olduğu görülebilir. Bu gözlemden yola çıkarak, eğitim aşamasında $(n, k) \in \{(64, 10), (128, 12), (256, 13)\}$, çiftleri kullanılmış ve her biri için 100K öznitelik vektörü elde edilecek şekilde $G(n, 1/2)$ çizgeleri üretilmiştir. Bu şekilde, her (n, k) çifti için ayrı bir sınıflandırıcı eğitilmiştir. Test aşamasında kullanılan rastsal çizgelere $k' = k + 1, k + 2, k + 3$, olarak üç farklı büyüklükte klik yerleştirilmiş, her k' değeri için 100 rastsal çizge örneği üretilmiştir.

Test sonuçları, Tablo II ve Şekil 2 üzerinden analiz edilmiştir. *Klik doğruluğu* (clique accuracy), budanma sonucu oluşan G''de tespit edilen $\omega(G)$ (G için maksimum) büyüklüğündeki kliklerin G'dekine oranıdır ve yöntemin başarısını ölçmedeki en önemli parametredir. Tablo II'deki klik doğruluğu ve hız artışı değerleri, her farklı (n, k, k') seçiminde test için kullanılan 100 rastsal çizgedeki sonuçların ortalaması alınarak hesaplanmıştır. Yöntemin performansını daha sağlıklı görebilmek açısından, Tablo II'de budama oranı ve hız artışı iki farklı q değeri kullanılarak karşılaştırılmıştır.

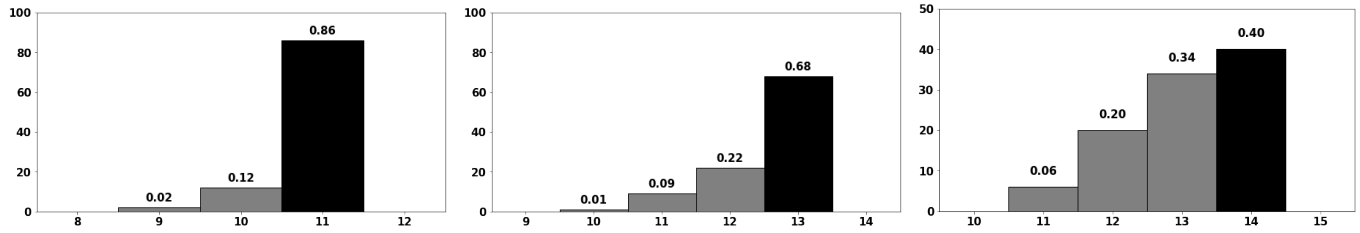
a) Budama oranı ve hız artışı: Tablo II'de, budamanın göreceli olarak n'in küçük değerleri için çok düşmediği gözlemlenmektedir. Hesaplama hızında budama oranına bağlı olarak her durumda en az 10 katlık artış görülmektedir.

b) Sağlamlık: Beklenildiği gibi, sınıflandırıcıların doğruluk performansı yerleştirilen klik büyüklüğü olan k''nin değeri yükselirken artmıştır. Öte yandan Şekil 2'de görüldüğü gibi n artarken, rastsal olarak hesaplanan k değeriyle ω'nın gerçek değeri arasındaki fark arttığından, klik doğruluğu azalmıştır. Örnek olarak, $q = .75$ için $(n, k) = (64, 11)$ durumunda sonuçların % 86'sı doğru tahminde bulunurken, $(n, k) = (128, 13)$ durumunda bu oran % 68'e düşmüştür.

Tablo II'de sunulan düğüm doğruluk değerlerinin, $n = 128$ iken bile % 70'in altına düşmediği, ama klik doğruluğunun % 40'larda olduğu görülmektedir. Bu durum, düğüm doğruluğundaki hata miktarının maksimum kliklerdeki düğümlerin de silinme olasılığının yükselmesine sebep olmasındandır. Buna rağmen, Şekil 2b'deki örnek çizgelerin % 99'unda 13-düğümlü kliklerden en fazla iki düğüm silindiği gözlemlenmiştir.

TABLO II: İki farklı q değerinin her biri için kullanılan üç sınıflandırıcının (n, k) değerleri ile $k' = k + 1$, $k + 2$ ve $k + 3$ yerleştirilmiş kliklerin büyüklükleri olmak üzere ayrılan durumlar için elde edilen sonuçlar.

($q = 0.55$)		k + 1				k + 2				k + 3			
n	k	Budama oranı	Klik Doğr.	Düğüm Doğr.	Hız artışı	Budama oranı	Klik Doğr.	Düğüm Doğr.	Hız artışı	Budama oranı	Klik Doğr.	Düğüm Doğr.	Hız artışı
64	10	0.8820	0.67	0.8800	77.5660	0.8504	0.94	0.8717	64.7478	0.8368	1.0	0.8748	9.3992
128	12	0.9211	0.38	0.8478	134.5886	0.9106	0.4	0.8435	123.9356	0.8978	0.77	0.8426	99.1681
256	13	0.9228	0.05	0.7766	255.7411	0.9134	0.1	0.7747	214.6767	0.9133	0.15	0.7798	217.4318
($q = 0.75$)		k + 1				k + 2				k + 3			
64	10	0.8100	0.86	0.8009	18.5902	0.7796	0.99	0.7886	7.8114	0.7330	1.0	0.7995	17.2541
128	12	0.8107	0.68	0.7088	24.0887	0.8034	0.8	0.7105	23.0164	0.7926	0.94	0.7105	20.9245
256	13	0.6568	0.4	0.4909	10.3529	0.6452	0.59	0.4880	9.7368	0.6471	0.59	0.4927	9.8585



(a) (64, 11) (b) (128, 13) (c) (256, 14)

Şekil 2: Gösterilen (n, k) değerlerinin her biri için 100 örnek çizge üzerinde $q = 0.75$ için gözlemlenen maksimum klik büyüklüklerinin dağılımı.

IV. SONUÇLAR

Bu çalışmada, bir düğümü içeren lokal çizgecik sayılarının çizge özniteliği olarak, maksimum klikleri sayma problemini çözmedeki rolü araştırılmış ve makine öğrenme algoritmaları kullanılarak buluşsal (heuristic) bir yöntem geliştirilmiştir. Elde edilen sonuçların yüksek doğruluk oranı yanında sağlam ve ölçülebilir olduğu gözlenmiştir. Burada sunulan yöntem farklı ölçekte her ağa uygulanabilir olmakla beraber, maksimum klik sayma dışında başka ağ davranışlarını incelemeye ya da çizge yapılarını aramaya yönelik karmaşıklığı yüksek başka soruların da çözümünde kullanılabilir.

BILGILENDIRME

Bu çalışma, kısmen TÜBİTAK 120E443 no'lu proje kapsamında desteklenmiştir.

KAYNAKLAR

[1] J. Chen, X. Huang, I. A. Kanj, and G. Xia, "Strong computational lower bounds via parameterized complexity," *Journal of Computer and System Sciences*, vol. 72, no. 8, pp. 1346–1367, 2006.

[2] D. Zuckerman, "Linear degree extractors and the inapproximability of max clique and chromatic number," in *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, 2006, pp. 681–690.

[3] S. Wasserman, K. Faust, and D. Iacobucci, "Social network analysis: Theory and methods," 1995.

[4] H. R. Bernard, P. D. Killworth, and L. Sailer, "Informant accuracy in social network data iv: A comparison of clique-level structure in behavioral and cognitive network data," *Social Networks*, vol. 2, no. 3, pp. 191–218, 1979.

[5] V. Boginski, S. Butenko, and P. M. Pardalos, "Statistical analysis of financial networks," *Computational statistics & data analysis*, vol. 48, no. 2, pp. 431–443, 2005.

[6] V. Stix, "Finding all maximal cliques in dynamic graphs," *Computational Optimization and applications*, vol. 27, no. 2, pp. 173–186, 2004.

[7] J. Lauri and S. Dutta, "Fine-grained search space classification for hard enumeration variants of subset problems," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 2314–2321.

[8] M. Grassia, J. Lauri, S. Dutta, and D. Ajwani, "Learning multi-stage sparsification for maximum clique enumeration," *arXiv preprint arXiv:1910.00517*, 2019.

[9] J. Lauri, S. Dutta, M. Grassia, and D. Ajwani, "Learning fine-grained search space pruning and heuristics for combinatorial optimization," *arXiv preprint arXiv:2001.01230*, 2020.

[10] N. Shervashidze, S. Vishwanathan, T. Petri, K. Mehlhorn, and K. Borgwardt, "Efficient graphlet kernels for large graph comparison," in *Artificial Intelligence and Statistics*, 2009, pp. 488–495.

[11] H. Kashima, H. Saigo, M. Hattori, and K. Tsuda, "Graph kernels for chemoinformatics," in *Chemoinformatics and advanced machine learning perspectives: complex computational methods and collaborative techniques*. IGI Global, 2011, pp. 1–15.

[12] L. Zhang, R. Hong, Y. Gao, R. Ji, Q. Dai, and X. Li, "Image categorization by learning a propagated graphlet path," *IEEE transactions on neural networks and learning systems*, vol. 27, no. 3, pp. 674–685, 2015.

[13] L. Zhang, M. Song, Z. Liu, X. Liu, J. Bu, and C. Chen, "Probabilistic graphlet cut: Exploiting spatial structure cue for weakly supervised image segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 1908–1915.

[14] B. Pattabiraman, M. M. A. Patwary, A. H. Gebremedhin, W.-k. Liao, and A. Choudhary, "Fast algorithms for the maximum clique problem on massive sparse graphs," in *Algorithms and Models for the Web Graph (WAW)*, 2013, pp. 156—-169.

[15] N. K. Ahmed, J. Neville, R. A. Rossi, and N. Duffield, "Efficient graphlet counting for large networks," in *2015 IEEE International Conference on Data Mining*. IEEE, 2015, pp. 1–10.

[16] N. K. Ahmed, T. L. Willke, and R. A. Rossi, "Estimation of local subgraph counts," in *2016 IEEE International Conference on Big Data (Big Data)*. IEEE, 2016, pp. 586–595.

[17] P. R. Östergård, "A fast algorithm for the maximum clique problem," *Discrete Applied Mathematics*, vol. 120, no. 1-3, pp. 197–207, 2002.

[18] S. Niskanen and P. R. Östergård, *Cliquer User's Guide: Version 1.0*. Helsinki University of Technology Helsinki, Finland, 2003.

[19] D. Eppstein, M. Löffler, and D. Strash, "Listing all maximal cliques in sparse graphs in near-optimal time," in *International Symposium on Algorithms and Computation*. Springer, 2010, pp. 403–414.

[20] G. Csardi, T. Nepusz *et al.*, "The igraph software package for complex network research," *InterJournal, complex systems*, vol. 1695, no. 5, pp. 1–9, 2006.

[21] R. A. Rossi and N. K. Ahmed, "The network data repository with interactive graph analytics and visualization," in *AAAI*, 2015. [Online]. Available: http://networkrepository.com

[22] P. Erdös and A. Rényi, "On random graphs i," *Publicationes Mathematicae (Debrecen)*, vol. 6, pp. 290–297, 1959.