



## Öznelikli Ağlarda Topluluk Arama

Tuğçe Kağnıcı<sup>1\*</sup>, Arzu Gorgulu Kakışım<sup>2</sup>

<sup>1</sup> İstanbul Ticaret Üniversitesi, Fenbilimleri Enstitüsü, İstanbul, Türkiye (ORCID: 0000-0002-3541-9763)

<sup>2</sup> İstanbul Ticaret Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, İstanbul, Türkiye (ORCID: 0000-0001-6169-3486)

(International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA) 2021 – 11-13 June 2021)

(DOI: 10.31590/ejosat.947859)

**ATIF/REFERENCE:** Kağnıcı, T. & Kakışım, A. (2021). Öznelikli Ağlarda Topluluk Arama. *Avrupa Bilim ve Teknoloji Dergisi*, (26), 332-337.

### Öz

Sosyal ağların yoğun bir şekilde kullanımıyla birlikte, bir dizi sosyal nesnelere ve nesnelere ait sosyal etkileşimlerden oluşan geniş ölçekli karmaşık ağ yapıları ortaya çıkmaktadır. Son zamanlarda, bu büyük ağ yapılarını analiz etmek için ağlardaki anlamlı grupları diğer bir deyişle toplulukları arama yaklaşımları önerilmektedir. Topluluk arama, sorgu nesnelere kullanarak, karmaşık ağ yapılarından sorgu nesnelere ile ilişkili, belli özellikler açısından birbirleri ile benzerlik gösteren ve birbirleri ile yoğun etkileşimde olan toplulukları ortaya çıkarma problemidir. Öznelikli ağlar, düğümleri ağ nesnelere, kenarları bu nesnelere arasındaki ilişkileri temsil eden graf yapılarından ve ağdaki düğümleri karakterize eden içerik verisi diğer bir deyişle özneliklerden oluşmaktadır. Geleneksel topluluk arama yaklaşımları, düğümlere ait öznelik verisini göz ardı ederek, sadece ağ topolojisine odaklanmaktadır. Bu çalışmada, öznelikli ağlarda topluluk arama problemi için öznelik verisini ağ topolojisine entegre etme fikrine dayanan yeni bir yaklaşım öneriyoruz. Önerilen yöntem, her düğümün yalnızca kendi öznelikleriyle bağlı olduğu yeni bir graf temsili oluşturularak, ağ topolojisine içeriksiz bir boyut eklemektedir. Bu çalışmada, mevcut topluluk arama algoritmalarını öznelikli ağlar üzerinde uygulanabilir hale getirmek, bu algoritmaların başarısını artırmak ve öznelik verisini sorgu verisi olarak kullanılabilir hale getirmek amaçlanmaktadır. Gerçek dünya veri setleri üzerinden elde edilen deneysel sonuçlar, önerilen yöntemin temel yöntemler ile karşılaştırıldığında daha yüksek doğruluk değerleri elde ettiğini göstermektedir.

**Anahtar Kelimeler:** Öznelikli ağlar, topluluk arama, k-tras, k-çekirdek.

## Community Search on Attributed Networks

### Abstract

With the intense use of social networks and information networks, large-scale complex networks are emerging, consisting of a number of social objects (such as individuals or organizations) and social interactions of these social actors. Recently, many community search methods have been proposed to analyze these network structures and interactions between network objects. Given one or more query objects, community search is the problem of discovering a subset of objects, called as community, that are related to query objects which are similar to each other in terms of certain properties and intensely interact with each other. A network is characterized by a graph structure whose nodes represent the network objects and the edges refer to the relations between these objects. An attributed network contains contextual information that defines the interests of network objects as well as the graph structure. However, traditional community search approaches focus on the network topology, ignoring the attribute data. In this paper, we propose a new approach for the community search problem in attribute networks, based on the idea of integrating the attribute data into the network topology. The proposed method adds a contextual dimension to the network topology by creating a bipartite graph structure where each node is connected only by its attributes. In this study, we aim to make existing community search algorithms applicable on attribute networks, to increase the accuracy of these algorithms and to make the attribute data usable as query data. Experimental results on real world datasets show that the proposed method achieves higher accuracy than baseline methods.

**Keywords:** Attributed Networks, Community Search, k-truss, k-core.

\* Tuğçe Kağnıcı: İstanbul Ticaret Üniversitesi, Fenbilimleri Enstitüsü, İstanbul, Türkiye, ORCID: 0000-0002-3541-9763, [tugce.kagnici@istanbulticaret.edu.tr](mailto:tugce.kagnici@istanbulticaret.edu.tr)

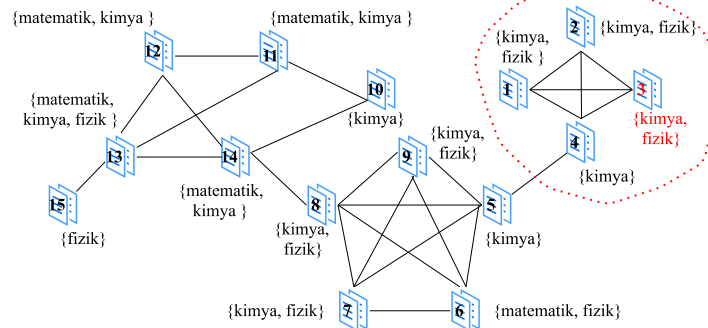
## 1. Giriş

Bilgi teknolojilerinin hızlı gelişimi ile birlikte sosyal medya ağları, protein-protein etkileşim ağları ve işbirliği ağları gibi birçok bileşenden ve bu bileşenlerin etkileşimlerinden oluşan kompleks ağ yapıları ortaya çıkmaktadır. Bu ağ yapılarını formülize etmek ve bu sayede ağdaki bileşenlere ait analizi kolaylaştırmak amacı ile graf yapıları kullanılmaktadır. Kompleks ağlar, ağdaki bileşenlerin düğümlere bileşenler arasındaki etkileşimlerin kenarlara karşılık geldiği graf yapıları ile temsil edilmektedir. Son zamanlarda, geniş ölçekli olarak gözlenen bu karmaşık ağ yapılarının daha yorumlanabilir bir özetini sunmak için ağımlı içerisindeki toplulukların tespiti problemi çalışılmaktadır (Lancichinetti ve Fortunato, 2009; Chunaev, 2020). Topluluk, belirli özelliklere göre birbirleri ile yoğun ilişkili ve benzerlik gösteren varlıklar kümesi olarak tanımlanır. Toplulukların keşfedilmesi, pazarlama, sosyoloji, biyoloji, güvenlik gibi birçok farklı alanda insanlar ya da nesnelere arasındaki ilişkilerin, ortak ilgi alanlarının ve hiyerarşinin analizi için oldukça önemli olmaktadır (Kakışim ve Sogukpinar, 2015). Literatürde, topluluk tespiti problemini adresleyen pek çok çalışma bulunmaktadır (He ve ark., 2018; Ma ve ark., 2020). Topluluk tespiti ile ağdaki tüm toplulukların ortaya çıkarılması amaçlanmakta ve böylece global bir ağ analizi sağlanmaktadır. Örneğin; sosyal medya ağlarında belirli bir kullanıcının dâhil olduğu ya da olabileceği sosyal çevrelerin keşfi, yakın çevredeki arkadaş gruplarına ait buluşmaları öneren ya da yakınlardaki bir restoranın kupon ve reklam paylaşımını kullanıcıya ileten sistemler için önemli olabilmektedir (Huang ve ark., 2017). Protein-protein etkileşimi gibi ağlarda birbirleri ile yüksek düzeyde etkileşime giren proteinlerinden oluşan toplulukları keşfetmek (Huang ve ark., 2019), belirli moleküler yapıların işlevselliğini ortaya çıkarmaya yardımcı olabilir. Fakat çok fazla düğümden ve yüksek sayıda kenardan oluşan geniş ölçekli ağlardaki tüm toplulukların keşfedilmesi maliyetli ve aynı zamanda zaman alıcı bir görev olarak bilinmektedir. Bu nedenle, son zamanlarda, birçok araştırmacı daha yerel bir analiz sağlayan "topluluk arama" problemine odaklanmaktadır. Topluluk arama, kişiselleştirilmiş sorgular kullanarak, hızlı ve verimli bir şekilde yoğun ilişkili, anlamlı ve kolay yorumlanabilen alt yapıları keşfetme problemi olarak tanımlanır (Huang ve Lakshmanan, 2017). Topluluk arama, sorgu tabanlı ve çevrimiçi bir arama özelliği sağladığından, özellikle gerçek zamanlı çalışan öneri

yada içerik sağlayıcı sistemler için daha hızlı sonuçlar elde edebilmek adına önemlidir (Shang ve ark., 2017).

Topluluk arama yöntemleri, sorgu düğümü/düğümünü içeren yoğun şekilde bağlı alt graf yapılarını diğer bir deyişle toplulukları bulmayı hedefler. Son zamanlarda,  $k$ -çekirdek ( $k$ -core) (Barbier ve ark., 2015),  $k$ -tras ( $k$ -truss) (Huang ve ark., 2014),  $k$ -klik ( $k$ -klick) (Yuan ve ark., 2015) ve  $k$ -kenar ( $k$ -edge) (Chang ve ark., 2015) olmak üzere birçok graf tabanlı topluluk arama yöntemi sunulmuştur. Bu yöntemler genellikle topluluk arama problemi için literatürde var olan graf alt yapılarını keşfetme yaklaşımlarını kullanmaktadırlar. Sorgu düğümünü temel alarak,  $k$ -çekirdek her bir düğümün en az bir  $k$  komşusu olduğu bir alt graf keşfi yaparken,  $k$ -tras her bir düğümün en az  $(k - 2)$  üçgen içinde bulunduğu bir alt grafı elde eder (Huang ve Lakshmanan, 2017).  $K$ -klik,  $k$  tane düğümün birbirine benzersiz bir kenarla bağlı olduğu bir alt grafı,  $k$ -kenar ise  $k - 1$  sayıda kenar çıkarıldığında bağlılığını kaybetmeyen bir alt grafı keşfeder (Zhang ve ark., 2020).

Gerçek hayatta gözlemlenen ağ nesnelere ise buldukları topluluğu temsil eden bazı özelliklere sahiptir. Örneğin, işbirliği ağlarındaki yazarlara ait uzmanlık alanları ya da sosyal ağlardaki kullanıcılara ait kişisel bilgiler düğümlere ait öznitelik verisi olarak elde edilmektedir. Graf yapısı ve öznitelik verisi olmak üzere iki farklı veriye sahip bu tür ağlar öznitelikli ağlar olarak adlandırılır. Graf verisi, ağ nesnesinin ağdaki yerel ve global konumunu temsil ederken, öznitelik verisi düğümü tanımlayan içeriksel veriyi temsil eder. Geleneksel graf tabanlı topluluk arama yaklaşımları, yalnızca ağ topolojisinin dikkate alır ve düğüm özniteliklerinin etkisini göz ardı eder. Fakat, düğümlere ait özniteliklerinin de topluluk oluşumu üzerinde bir etkiye sahip olduğu fikri ile son zamanlarda, öznitelik verisini de topluluk arama sürecine dahil eden yeni yaklaşımlar önerilmektedir (Huang ve Lankshmanan, 2017; Sun ve ark., 2020). Ayrıca, öznitelikleri de sorgu olarak kullanarak, sorgu öznitelikleri ile ilişkili topluluk arama süreci de içeriksel olarak benzer düğüm kümelerini ortaya çıkarmak için önemli olmaktadır. Şekil 1'de sorgu noktalarının ilişkili olduğu, öznitelikli bir graf yapısı örneklendirilmiştir. Şekilde, ağda ki üyeler, uzmanlık alanları (fizik, kimya, matematik) ve aralarındaki ilişkiler verilmektedir. Bu graf üzerinden sadece 3. üye sorgulanır ise 1., 2., 3. ve 4. üyenin olduğu öznitelik homojenliği yüksek bir alt graf topluluk olarak keşfedilir (Sun ve ark., 2020).



Şekil 1. Öznitelikli grafında topluluk aramasına bir örnek: Sorgulanan düğüm "3", sorgulanan öznitelikler {kimya, fizik}.

Bu çalışmada, öznitelikli ağlarda topluluk arama problemini adreslemek amacı ile düğümlere ait öznitelik verisini ağ topolojisine entegre etme fikrine dayanan yeni bir graf temsili öneriyoruz. Önerilen yöntem, öznitelikleri içeriksel bir düğüm olarak tanımlayarak ve her düğümün yalnızca kendi öznitelikleriyle ile bağlantılı olduğu bir graf oluşturarak ağ

topolojisine içeriksel bir boyut eklemektedir. Bu yeni graf temsili ile mevcut topluluk arama algoritmalarını öznitelikli ağlar üzerinde uygulanabilir hale getirmek hedeflenmiştir. Ayrıca, içerik verisini dâhil ederek topoloji tabanlı algoritmaların başarısını artırmak ve öznitelik verisini sorgu verisi olarak kullanılabilir hale getirmek amaçlanmaktadır. Gerçek dünya veri

setleri üzerinden elde edilen deneysel sonuçlar, önerilen yöntemin temel yöntemler ile karşılaştırıldığında daha yüksek doğruluk değerleri elde ettiğini göstermektedir.

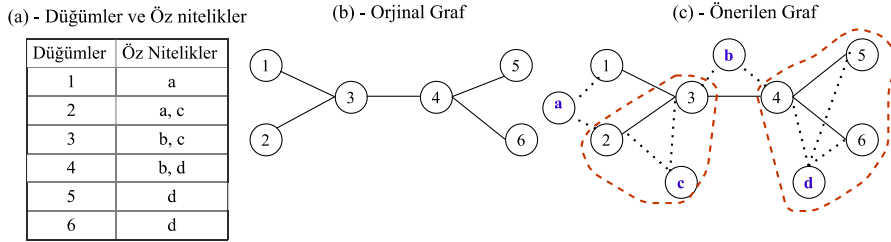
Bu yazının geri kalanı aşağıdaki şekilde düzenlenmiştir. İkinci bölümde önerilen yaklaşım ve kullanılan topoloji tabanlı algoritmalar ayrıntılı olarak sunulmaktadır. Üçüncü bölümde, deneysel sonuçlar verilmektedir. Son olarak dördüncü bölümde ise makaleye ait sonuçlar özetlenmekte ve gelecek çalışmalardan bahsedilmektedir.

## 2. Önerilen Yöntem

### 2.1. Önerilen Graf Yapısı

Ağlar, düğümleri ağlardaki nesnelere, kenarları nesnelere arasındaki ilişkilere karşılık gelen bir graf yapısı ile temsil edilmektedir. Öznitelikli ağlar ise ağın topolojisini temsil eden graf yapısının yanı sıra, ağdaki nesnelere ilgi alanlarını temsil eden içeriksel bir veriye de sahiptir. Örneğin, bir web sayfa ağında, düğümler ve yönlendirilmiş kenarlar sırasıyla web sayfalarına ve web sayfaları arasındaki hiper bağlantılara karşılık gelirken, web sayfasındaki kelimeler, başlıklar, etiketler, konular ağın içeriksel verisini diğer bir deyişle özniteliklerini temsil etmektedir. Bir ağ,  $V = \{v_1, v_2, \dots, v_n\}$  düğümler kümesi,  $E = (V \times V)$  kenarlar kümesi olmak üzere  $G = (V, E)$  grafi olarak tanımlanır. Düğüm  $v_1$  ve düğüm  $v_2$  arasında bir bağlantı var ise  $e_{12}$  kenar değeri 1, diğer durumda 0'dır. Öznitelikli ağlarda ise ağın topolojisini temsil eden  $v$  çalışmada yapısal graf olarak adlandırdığımız  $G$  grafının yanı sıra, her bir düğümün sahip olduğu öznitelikleri temsil eden bir  $F = \{f_1, f_2, \dots, f_n\}$  öznitelik kümesi bulunmaktadır. Burada,  $f_v$   $v$  düğümüne ait öznitelikleri içeren bir alt öznitelik kümesidir. Ağın topolojik yapısı bir graf ile, öznitelik verisi ise farklı bir küme ile tanımlandığından, graf tabanlı topluluk arama algoritmaları, ağın içeriksel verisi olan  $F$  öznitelik kümesini göz ardı etmektedir.

Bu çalışmada, ağın içeriksel verisi olan  $F$  öznitelik kümesinin ağın yapısal verisi  $G$  grafına entegre edildiği yeni bir



Şekil 2. Önerilen graf yapısı için örnek çizim.

### 2.2. Topluluk Arama Algoritmaları

Topluluk arama algoritmaları, bir sorgu isteğine bağlı olarak çevrimiçi bir şekilde sorgu ile ilişkili toplulukları elde etmeyi amaçlamaktadır. Bu çalışmada,  $k$ -tras (Huang vd., 2014) ve  $k$ -çekirdek (Barbier vd., 2015) tabanlı topluluk arama yöntemleri önerilen öznitelikli graf yapısı üzerinde çalıştırılacaktır. Bu nedenle, bu bölümde bu iki algoritma anlatılacaktır.

#### 2.2.1. $k$ -tras tabanlı topluluk arama

$k$ -tras topluluk arama yöntemi, bir  $G$  grafi ve bir sorgu düğümü  $q$  verildiğinde,  $q$  düğümünü içeren ve her kenarın en az  $k - 2$  üçgen içinde bulunduğu en büyük alt grafi  $S \subseteq G$  bulmayı amaçlar. Matematiksel olarak ifade edilir ise  $\forall$  kenar  $e \in E(S)$

graf temsili oluşturulmaktadır. Bu amaçla,  $F$  öznitelik kümesindeki her bir öznitelik bir düğüm olarak değerlendirilmiş ve  $G$  graf yapısı içeriksel düğüm olarak adlandırdığımız öznitelikler ile genişletilmiştir. Genişletilmiş bu graf yapısı  $G' = (V', E')$  olarak temsil edilmektedir. Düğüm kümesi  $V'$  yapısal düğümler ve içeriksel düğümler olmak üzere iki farklı kümeden gelen düğümlerin birleşiminden  $V \cup F$  oluşmaktadır. Kenar kümesi  $E'$  ise yapısal graftan gelen  $E$  kümesinin yanı sıra, yapısal düğümler ve o düğümlere ait öznitelikleri temsil eden içeriksel düğümler arasında oluşturulmuş yeni bağlantıları da kapsamaktadır. Eğer bir  $v$  düğümü  $u$  özniteliğini içeriyor ise  $v$  yapısal düğümü ve  $u$  öznitelik düğümü arasında bir bağlantı oluşturulur. Yapısal düğüm ile düğüme ait özniteliklere karşılık gelen içeriksel düğümler arasında yönsüz ve ağırlık değeri 1 olacak şekilde kenarlar oluşturulur. Bu süreç, ağdaki tüm yapısal düğümler için uygulanarak  $G'$  grafına ait  $E'$  kenar kümesi elde edilir. Şekil 2'de ağın orjinal yapısı ve önerilen graf yapısı örneklendirilmiştir. Şekil 2 (a) ve (b)' de sırasıyla altı adet ağ nesnesine ait öznitelikler ve bu ağ nesnelere arasındaki bağlantıları gösteren orjinal graf yapısı verilmektedir. Şekil 2 (c)' de ise altı adet ağ nesnesini temsil eden altı yapısal düğümleri ve bu ağ nesnelere ait dört farklı özniteliği temsil eden içeriksel düğümleri içeren yeni bir graf temsili sunulmaktadır. Yapısal düğüm ile sahip olduğu öznitelikler arasında doğrudan bir bağlantı oluşturulmuştur. Örneğin; yapısal düğüm 2 içeriksel düğümler  $a$  ve  $c$  ile yapısal düğüm 3 ise içeriksel düğümler  $b$  ve  $c$  ile doğrudan bağlıdır. Önerilen graf ile orjinal grafa yeni alt graf yapıları dahil edilmektedir. Örneğin; eklenen yeni içeriksel düğüm  $c$  nedeni ile düğüm 2 ve düğüm 3'ü içeren yeni bir üçgen yapı grafa eklenmiştir.

Yukarıdaki terminolojileri kullanılarak, problem, önerilen öznitelikli graf yapısı  $G' = (V', E')$  ve sorgu  $q$  verildiğinde,  $q$  düğümüne yada özniteliğine ait  $C'(q)$  topluluğunu arama olarak tanımlanır. Topluluk arama sürecinde, ağın topolojik yapısı üzerinde çalışan geleneksel topluluk arama yaklaşımları önerilen  $G' = (V', E')$  graf yapısına uygulanacaktır.

için  $sup(e, S) \geq (k - 2)$  olmalıdır. Burada,  $sup(e, S)$   $S$  alt grafindaki  $e$  kenarının desteği olarak tanımlanır ve  $S$  alt grafindaki  $e$  kenarını içeren üçgen sayısını ifade eder. Bir  $G(V, E)$  grafindaki üçgen yapısı  $\{(u, v), (u, w), (v, w)\} \in E$  ve  $\{u, v, w\} \in V$  olmak üzere 3 düğümü birbirine bağlayan 3 kenardan oluşan üçgen biçiminde bir alt graf  $T_{uvw}$  olarak tanımlanır. Bu durumda, örneğin  $T_{quv}$  ve  $T_{qvw}$  olmak üzere iki farklı üçgen  $e$  kenarını içeriyor ise kenar desteğinin değeri  $sup(e(q, v)) = 2$  olur. Bir alt graf  $S$ 'in tras derecesi  $\tau(S)$ , içerdiği en düşük kenar destek değerine sahip kenar tarafından belirlenir. Bir kenar  $e$ 'nin tras derecesi  $\tau(e)$  ise  $e$  kenarını içeren en yüksek tras derecesine sahip  $S$  alt grafinin tras derecesine  $\tau(S)$  eşittir.

$k$ -tras topluluk arama yöntemi (Huang ve ark., 2014; Shang ve ark., 2017), bir  $G$  grafiğine  $k$  ve  $q$  parametrelili verildiğinde, her bir kenar için tras derecesi hesaplamak amacı ile  $G$  grafi üzerinde tras ayrıştırma (truss decomposition) işlemi uygulanır. Tras ayrıştırma sürecinde, öncelikle her bir kenar  $e \in E$  için destek değeri  $sup(e)$  hesaplanır. Sonrasında,  $k=2$ ' den başlayarak, yinelemeli bir şekilde  $sup(e) \leq k - 2$  koşulunu sağlamayan en düşük destek değerli kenarları graftan kaldırır. Bu işlemden sonra, çıkarılan kenarlara ait maksimum tras derecesi hesabı yapılır ve bu değer  $k$  olarak belirlenir. Budanan kenarları içeren üçgen yapılarındaki diğer kenarlara ait kenar destek değeri bir azaltılır. Bu süreç,  $k - 2$  değerinden daha küçük ve eşit kenar destek değerine sahip kenarlar kaldırılincaya kadar devam eder. Algoritmanın sonucunda, her bir kenar  $e$  için bağ derecesi  $\tau(e)$  değeri hesaplanmış olur. Sonrasında, her bir düğüm  $v \in V$  için, düğüm  $v$ 'nin komşuları kenar tras derecelerine göre büyükten küçüğe sıralanır ve kenar tras derecesi  $k$ 'ya eşit olanlar işaretlenir. Bu sayede, düğüm  $v$ 'ye bağlı diğer kenarlar kenar tras derecelerine göre işaretlenmiş olur. Bir  $q$  düğümü sorgulandığında, algoritma  $q$ 'ye bağlı her bir kenarı kontrol ederek topluluk aramaya başlar. Verilen sorgu düğüm  $q$ 'dan başlayarak  $q$ 'ya bağlı her bir kenar  $(q, v) \in E$  için  $\tau(q, v) \geq k$  koşulunun sağlanıp sağlanmadığı kontrol edilir. Diğer bir deyişle, üçgen yapılar ile bağlı topluluklar aranır. Bu süreç,  $q$ 'yu içeren tüm bağlı kenarlar için devam eder ve sonuç olarak  $q$ 'yu içeren bir dizi  $k$ -tras topluluğu ortaya çıkarılır.

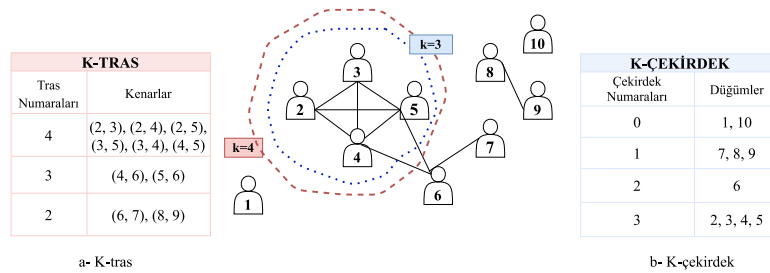
Şekil 3 (a)'da verilen bir graf üzerinde  $k$ -tras toplulukları örneklenmiştir. Örneğin; düğüm 1 veya düğüm 10 sorgulandığında, düğüm hiçbir üçgen yapının içinde olmadığından ve bu nedenle  $sup(1)$  değeri 0 olduğundan,  $k$ -tras yöntemi bu sorgu sonuç döndürmez. Düğüm 2 sorgulandığında, düğüm 2'ye bağlı kenarlar iki farklı üçgen yapısının içinde olduğundan bu kenarların destek değeri  $sup(1) = 2$  ve tras değeri ise 4 olur. Bu kenarların bağlı olduğu diğer düğümler 3, 4, 5' ait kenarların da tras değerleri 4 olduğundan,  $k=4$  parametresi için bu sorguya ait topluluk 2, 3, 4, 5 olarak belirlenir.

## 2.2.2. $k$ -çekirdek tabanlı topluluk arama

$k$ -çekirdek topluluk arama yöntemi, bir  $G$  grafi ve bir sorgu düğümü  $q$  verildiğinde,  $q$  düğümünü içeren ve her bir düğüm derecesinin en az  $k$  tamsayısı kadar olduğu diğer bir deyişle her bir düğümün en az  $k$  adet bağlantısının olduğu en büyük alt graf yapısını  $S \subseteq G$  bulmayı amaçlar. Matematiksel olarak ifade edilir ise  $\forall$  düğüm  $v \in V(S)$  için  $deg_S(v) \geq k$  olmalıdır. Bir düğüm  $v \in V$ 'ye ait çekirdek numarası  $\phi(v)$  ise  $v$  içeren en yüksek çekirdek dereceli alt grafin derecesine eşittir. Bu durumda, eğer bir düğüm bir  $k$ -çekirdek altgrafının içerisinde bulunuyor fakat bir  $(k+1)$ -çekirdek altgrafının içinde bulunmuyor ise,  $\phi(v) = k$  olur.

$k$ -çekirdek topluluk arama yöntemi (Barbier ve ark., 2015; Fang ve ark., 2020), bir  $G$  grafiğine  $k$  ve  $q$  parametrelili verildiğinde,  $G$  grafindaki tüm düğümler için  $k$ -çekirdek numarasını hesaplamak amacı ile  $G$  grafi üzerinde çekirdek ayrıştırma (core decomposition) işlemi uygulanır.  $k$ -çekirdek topluluk bulma algoritması öncelikli olarak graftaki düğümlerin düğüm derecelerini hesaplar. Düğümler düğüm derecesine göre artan bir sırada sıralanır. Graftaki minimum düğüm derecesine, örneğin  $d$ , sahip düğüm  $v$ 'nin çekirdek numarası  $\phi(v) = d$  olarak belirlenir ve bu düğüm ve düğüme bağlı kenarlar  $G$  grafindan kaldırılır. Sonrasında, graftaki tüm düğümler için düğüm dereceleri tekrar hesaplanır ve süreç yinelemeli bir şekilde tüm düğümlere ait çekirdek numaraları hesaplanana kadar devam eder. Bu sayede, düğüm  $v$ 'ye bağlı diğer düğümlerin de çekirdek numaraları işaretlenmiş olur. Bir  $q$  düğümü sorgulandığında,  $q$ 'ya bağlı her bir düğüm  $v \in V$  için  $\phi(v) \geq k$  koşulunun sağlanıp sağlanmadığı kontrol edilerek  $q$ 'yu içeren bir dizi  $k$ -çekirdek topluluğu ortaya çıkarılır.

Şekil 3 (b)'de verilen bir graf üzerinde  $k$ -çekirdek toplulukları örneklenmiştir. Örneğin;  $k=3$  olmak üzere 7. düğüm sorgulanır ise bu düğümün derecesi 1 olduğundan algoritma bir sonuç dönmeyecektir. Düğüm 2  $k=3$  için sorgulandığında ise, bu düğümün ve bağlı diğer düğümlerin düğüm dereceleri 3 ve 3'den büyük olduğundan algoritma 2, 3, 4, 5 düğümünü içeren bir topluluk keşfedecektir.



Şekil 3.  $K$ -tras ve  $K$ -çekirdek yöntemleri için örnek bir çizim.

## 3. Araştırma Sonuçları ve Tartışma

### 3.1. Veri Kümeleri ve Deneysel Kurulum

Deneyle, beş farklı karmaşık ağ veri kümesi üzerinde gerçekleştirilmiştir. Tablo 1'de  $|V|$ ,  $|E|$ ,  $|A|$  sırasıyla düğüm sayısı, kenar sayısı ve öznitelik sayısını temsil etmek üzere bu veri setlerine ait istatistikler verilmektedir. Pubmed, Cora ve Citeseer ağları, her düğümün bir bilimsel makaleye karşılık geldiği atf ağlarıdır (LINQS, 2021). Atf ağlarında, bir makalenin diğerine

atf vermesi durumunda bu makaleleri temsil eden düğümler arasında doğrudan bir bağlantı (kenar) oluşmaktadır. Her düğüm, makalenin başlığında bulunan kelimelerden oluşan öznitelikler ile ilişkilendirilmiştir. Facebook ve BlogCatalog ağları, aynı adlı sosyal ağ sitelerinden elde edilen bir sosyal ağ veri setleridir. Bu ağlarda, düğümler sosyal medya kullanıcılarını, kenarlar ise kullanıcılar arasındaki arkadaşlık ilişkilerini temsil etmektedir. Bu ağlarda düğümlere ait öznitelikler, sırasıyla kullanıcılara ait kişisel bilgilere ve bloglarda kullanılan etiketlere karşılık gelmektedir.

Tablo 1. Çalışmada kullanılan beş farklı veri kümesinin ait istatistikler

Veri Kümesi	$ V $	$ E $	$ A $	Topluluk Sayısı	Test Edilen Düğüm Sayısı
Cora	2708	5429	1433	7	700
Citeseer	3312	4598	3703	6	600
Blogcatalog	5196	171743	8189	6	600
Pubmed	19717	44324	500	3	300
Facebook	22470	11235	4714	4	400

Tüm yöntemler tarafından bulunan toplulukların kalitesini değerlendirmek için her bir ağa ait her bir sınıftan rastgele 100 düğüm seçilmiş ve sorgu düğümleri olarak kullanılmıştır. Sorgu düğümünün ait olduğu topluluk  $C$ , sorgu işleminden sonra elde edilen topluluk  $C'$  olmak üzere, her bir sorgudan elde edilen topluluk için keskinlik ( $Prec$ ) ve duyarlılık ( $Rec$ ) değerleri

$$Prec(C_q, C'_q) = \frac{|C_q \cap C'_q|}{|C'_q|}, \quad Rec(C_q, C'_q) = \frac{|C_q \cap C'_q|}{|C_q|} \quad (1)$$

$$Prec_{ort} = \frac{\sum_{q \in Q} Prec(C_q, C'_q)}{|Q|}, \quad Rec_{ort} = \frac{\sum_{q \in Q} Rec(C_q, C'_q)}{|Q|}, \quad F1 = \frac{2 * Prec_{ort}(C_q, C'_q) * Rec_{ort}(C_q, C'_q)}{Prec_{ort}(C_q, C'_q) + Rec_{ort}(C_q, C'_q)} \quad (2)$$

### 3.2. Performans Değerlendirilmesi

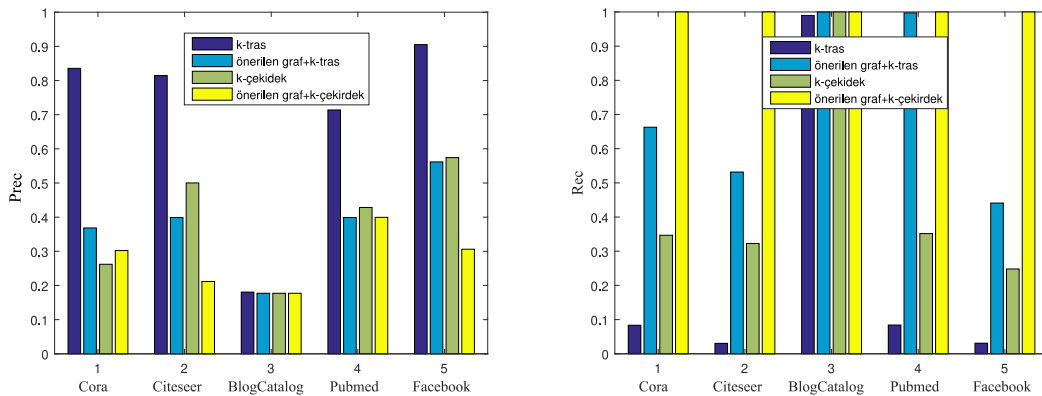
Bu bölümde, Bölüm 2'de detaylı olarak anlatılan  $k$ -tras ve  $k$ -çekirdek topluluk arama yöntemlerinin performansı orijinal graf yapısı ve önerilen graf yapısı için analiz edilmektedir. Tablo 2' de beş farklı veri seti üzerinde çalıştırılan yöntemlere ait F-ölçüm sonuçları verilmektedir. Bu sonuçlara göre, önerilen graf yapısı

hesaplanarak, F1-ölçümü değerleri elde edilmiştir. Sorgulanacak düğümlerinden oluşan düğüm kümesi  $Q$  ve sorgu düğümü  $q$  olmak üzere keskinlik ve duyarlılık değerleri Denklem 1 kullanılarak hesaplanmaktadır. Her bir veri seti için F1 değeri keskinlik ve duyarlılık değerlerinin ortalaması hesaplanarak Denklem 2'de gösterildiği şekilde elde edilmektedir.

üzerinden elde edilen  $k$ -tras sonuçlarının, tüm veri setleri için, orijinal graf yapısı üzerinden elde edilen  $k$ -tras performans sonuçlarından daha yüksek olduğu gözlemlenmektedir. Benzer şekilde,  $k$ -çekirdek yöntemi önerilen graf yapısı ile çalıştırıldığında Citeseer veri seti haricinde tüm veri setleri için daha yüksek doğruluk değerleri elde edilmiştir.

Tablo 2. Farklı veri setleri üzerinde yöntemlere ait F-ölçüm sonuçları ( $k=3$ ).

	Cora	Citeseer	BlogCatalog	Facebook	Pubmed
$k$ -truss	0.11450	0.05434	0.29812	0.05304	0.12964
Önerilen Graf + $k$ -truss	<b>0.36054</b>	<b>0.29166</b>	<b>0.30113</b>	<b>0.37883</b>	<b>0.56986</b>
$k$ -core	0.27851	<b>0.38035</b>	<b>0.30113</b>	0.34574	0.38623
Önerilen Graf + $k$ -core	<b>0.46411</b>	0.34936	<b>0.30113</b>	<b>0.46882</b>	<b>0.57082</b>



Şekil 4. Farklı veri setleri için karşılaştırılan yöntemlerin keskinlik ( $Prec$ ) ve duyarlılık ( $Rec$ ) değerleri

Şekil 4'de ise yöntemlere ait keskinlik ( $Prec$ ) ve duyarlılık ( $Rec$ ) değerleri sunulmaktadır. Keskinlik sonuçları temel alındığında, orijinal graf üzerinde çalıştırılan  $k$ -tras ve  $k$ -çekirdek algoritmalarının, neredeyse tüm veri setleri için önerilen yaklaşımdan daha iyi performans sergiledikleri görülmektedir. Bu durumun en temel nedeni, önerilen graf yapısındaki içeriksel

düğümün aynı grup içerisinde olmayan birçok düğüm arasında doğrudan bağlantı oluşturmasıdır. Böylece, ilgisiz veya gürültü veri olarak değerlendirilebilecek alt graf yapılarını topluluk arama sürecine dâhil etmektedir. Önerilen graf yapısı üzerinden bir düğüm sorgulandığında, orijinal grafa kıyasla, çok daha fazla sayıda düğüm içeren topluluklar elde edilmektedir. Şekil 4'de

verilen duyarlılık sonuçları dikkate alındığında ise, topluluk arama yöntemlerinin önerilen graf yapısı ile daha yüksek performans elde ettiği görülmektedir. Özellikle, önerilen graf yapısı  $k$ -çekirdek yöntemi ile çalıştırıldığında tüm veri setleri için ve test edilen tüm sorgu düğümleri için toplulukları %100 doğru bir şekilde bulmaktadır. Bu sonuçlara göre, orijinal grafa kıyasla, önerilen graf yapısı üzerinden, topluluk arama yöntemlerinin daha geniş bir ölçekte arama yapabildiği gözlemlenmiştir. Bu doğrultuda, önerilen yöntemin, tek bir sorgu düğümü üzerinden bir topluluğa ait birçok düğümü keşfedebilme eğiliminde olduğu analiz edilmiştir.

#### 4. Sonuç

Bu çalışmada, öznitelik verisini gözardı eden graf topolojisi tabanlı topluluk arama algoritmalarını öznitelikli ağlar için kullanılabilir hale getiren yeni bir graf temsili önerdik. Öznitelik verisini bir graf yapısına dönüştürerek, orijinal graf yapısına içeriksel bir boyut eklemeyi amaçladık. Deneysel sonuçlar, öznitelik verisi ile zenginleştirilmiş graf yapısı kullanarak topluluk arama süreci için genellikle daha yüksek başarı değerleri elde edildiğini göstermektedir. Gelecek çalışmalarda, öncelikle veri seti ve karşılaştırılan temel topluluk arama algoritmalarının sayısını artırarak deneysel sonuçları genişletmeyi planlıyoruz. Önerdiğimiz yaklaşıma ait duyarlılık sonuçlarını artırmak için önerilen graf yapısı kullanılarak elde ettiğimiz topluluklardaki ilgisiz düğümleri filtreleyen yeni bir yaklaşım geliştirmeyi hedefliyoruz. Bu amaçla, elde edilen topluluklar üzerinden, diğer bir deyişle  $k$ -tras ya da  $k$ -çekirdek alt grafları üzerinden sorgu düğümünün dahil olduğu daha anlamlı bir alt graf araması yapmayı amaçlıyoruz. Diğer bir gelecek çalışma ise, önerilen ağ temsili üzerinde daha hızlı ve başarılı sonuçlar üretecek, ağı yerel ve küresel komşuluklarını temel alarak topluluk arama süreci uygulayan yeni bir topluluk arama yaklaşımı önermektir.

#### Kaynakça

Lancichinetti, A., & Fortunato, S. (2009). Community detection algorithms: a comparative analysis. *Physical review E*, 80(5), 056117.

Chunaev, P. (2020). Community detection in node-attributed social networks: a survey. *Computer Science Review*, 37, 100286..

Kakisim, A., & Sogukpinar, I. (2015, May). Community detection in social networks using content and link analysis. In *2015 23rd Signal Processing and Communications Applications Conference (SIU)* (pp. 1521-1524). IEEE.

He, K., Li, Y., Soundarajan, S., & Hopcroft, J. E. (2018). Hidden community detection in social networks. *Information Sciences*, 425, 92-106.

Ma, T., Liu, Q., Cao, J., Tian, Y., Al-Dhelaan, A., & Al-Rodhaan, M. (2020). LGIEM: Global and local node influence based community detection. *Future Generation Computer Systems*, 105, 533-546.

Huang, X., Lakshmanan, L. V., & Xu, J. (2017, April). Community search over big graphs: Models, algorithms, and opportunities. In *2017 IEEE 33rd international conference on data engineering (ICDE)* (pp. 1451-1454). IEEE.

Huang, X., Lakshmanan, L. V., & Xu, J. (2019). Community search over big graphs. *Synthesis Lectures on Data Management*, 14(6), 1-206.

Huang, X., & Lakshmanan, L. V. (2017). Attribute-driven community search. *Proceedings of the VLDB Endowment*, 10(9), 949-960.

Shang, J., Wang, C., Wang, C., Guo, G., & Qian, J. (2017). An attribute-based community search method with graph refining. *The Journal of Supercomputing*, 1-28.

Barbieri, N., Bonchi, F., Galimberti, E., & Gullo, F. (2015). Efficient and effective community search. *Data mining and knowledge discovery*, 29(5), 1406-1433.

Huang, X., Cheng, H., Qin, L., Tian, W., & Yu, J. X. (2014, June). Querying  $k$ -truss community in large and dynamic graphs. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data* (pp. 1311-1322).

Yuan, L., Qin, L., Lin, X., Chang, L., & Zhang, W. (2016). Diversified top- $k$  clique search. *The VLDB Journal*, 25(2), 171-196.

Chang, L., Lin, X., Qin, L., Yu, J. X., & Zhang, W. (2015, May). Index-based optimal algorithms for computing steiner components with maximum connectivity. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data* (pp. 459-474).

Zhang, Q., Li, R. H., Yang, Q., Wang, G., & Qin, L. (2020, April). Efficient top- $k$  edge structural diversity search. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)* (pp. 205-216). IEEE.

Sun, H., Huang, R., Jia, X., He, L., Sun, M., Wang, P., ... & Huang, J. (2020). Community search for multiple nodes on attribute graphs. *Knowledge-Based Systems*, 193, 105393.

Fang, Y., Huang, X., Qin, L., Zhang, Y., Zhang, W., Cheng, R., & Lin, X. (2020). A survey of community search over big graphs. *The VLDB Journal*, 29(1), 353-392.

LNQS, (2021). Statistical Relational Learning Group, (<https://linqs.soe.ucsc.edu/data>), (Erişim Tarihi: 07.05.2021).