

Yön-Kesme Yöntemi ve Kare Izgarada Adreslenmiş Temel Arşimet Latislerinde İki-Boyutlu Bağ Perkolasyonu

Alptekin Yıldız¹

İstanbul Teknik Üniversitesi, Uçak ve Uzay Bilimleri Fakültesi, Uçak Mühendisliği Bölümü, İstanbul, Türkiye (ORCID: 0000-0002-4801-2209)
İstanbul Teknik Üniversitesi, Havacılık ve Uzay Teknolojileri Uygulama ve Araştırma Merkezi, İstanbul, Türkiye (ORCID: 0000-0002-4801-2209)

(İlk Geliş Tarihi 3 Şubat 2020 ve Kabul Tarihi 11 Mart 2020)

(DOI: 10.31590/ejosat.682106)

ATIF/REFERENCE: Yıldız, A. (2020). Yön Kesme Yöntemi ve Kare Izgarada Adreslenmiş Temel Arşimet Latislerinde İki-Boyutlu Bağ Perkolasyonu. *Avrupa Bilim ve Teknoloji Dergisi*, (18), 515-530.

Öz

Perkolasyon teorisi, ağ yapıları üzerinde faz geçişleri sırasında gerçekleşen olayları anlamayı ve tanımlamayı hedefleyen istatistiksel bir yaklaşımdır. Yapılan bu çalışmada, geliştirilen *yön-kesme yöntemi* tanıtılarak, perkolasyon teorisinin en temel ve yaygın uygulaması olan iki-boyutlu izgaralar üzerinde, Monte Carlo temelli bir benzetimde, perkolasyon eşiği ve kritik üstellerin tespit edilebileceği gösterilmiştir. Latis üzerindeki herhangi bir düğüm noktasına ait komşuları gözetererek, bu noktaya ait bağ potansiyelini bir dizi ile tanımlayan yöntem, kurulmuş bağların tekrarlanmaması ve tanımlı olmayan bağların kurulmaması için ilgili yönlerin kesilerek diziden çıkartılmasına dayanmaktadır. Ayrıca, referans numaralarına dayalı kümelenme sistemiyle, küme davranışlarını incelemeye olanak sağlayan algoritma, bu yöntemle evrensel davranışların ve kritik üstellerin de tespit edilmesini sağlamaktadır. Çalışmada, yöntemin sınanması adına, kare izgara üzerine adreslenen üç farklı temel Arşimet latislerinde, perkolasyon benzetimleri yapılmıştır. İki-boyutta bağ perkolasyonu için, üçgen, kare ve bal peteği latislerinde, gerçek değerleri sırasıyla yaklaşık olarak 0,3473, 0,5 ve 0,6527 olan perkolasyon eşikleri, 1000×1000 boyutlarına sahip bir izgara üzerinde üçgen latis için $0,3469 \pm 0,0016$, kare latis için $0,4992 \pm 0,0022$ ve bal peteği latis için $0,6510 \pm 0,0027$ olarak tespit edilmiştir. Yine değerleri iki-boyutta evrensel olan kritik üsteller β , γ , ν ve fraktal boyut D , gerçek değerleri sırasıyla $5/36$ (~0,1389), $43/18$ (~2,3889), $4/3$ (~1,3333) ve $91/48$ (~1,8958) olmak üzere, üçgen, kare ve bal peteği latislerinde sırasıyla, β değeri 0,1389, 0,1386 ve 0,1390, γ değeri 2,3886, 2,3272 ve 2,3275, ν değeri 1,3326, 1,3392 ve 1,3168 ve son olarak D değeri 1,8801, 1,8729 ve 1,8932 şeklinde tespit edilmiştir. Her üç farklı latis için de oldukça uygun sonuçların elde edildiği benzetimler ışığında gösterilmiştir ki yön-kesme yöntemi farklı latis tiplerinde ve potansiyelde farklı boyutlarda, gerçekleştirilmesi hedeflenen perkolasyon benzetimlerine etkin ve nispeten kolay bir algoritma olmaya aday durumdadır.

Anahtar Kelimeler: Perkolasyon Teorisi, Perkolasyon Eşiği, Kritik Üsteller, Yön-Kesme Yöntemi.

Direction-Cut Method and Two-Dimensional Bond Percolation in Basic Archimedes Lattices Addressed on Square Latis

Abstract

Percolation theory is a statistical approach that aims to understand and identify events that occur during phase transitions on networks. In this study, with the developed *direction-cut method*, it has been shown that percolation threshold and critical exponents can be

¹ Sorumlu Yazar: İstanbul Teknik Üniversitesi, Uçak ve Uzay Bilimleri Fakültesi, Uçak Mühendisliği Bölümü, İstanbul, Türkiye, ORCID: 0000-0002-4801-2209, alptekin_yildiz@hotmail.com

determined with Monte Carlo-based simulation on two-dimensional grids that are the most basic and common application of percolation theory. By considering the neighbors of any site on the lattice, the method of defining the bond potential of this site with an array is based on cutting the related directions from the array so that the established bonds are not repeated and the infeasible bonds are not established. In addition, the algorithm, which allows examining the cluster behavior with the cluster system based on reference numbers, enables the detection of universal behaviors and critical exponents. In order to test the method, percolation simulations have been performed in three different Archimedean lattices addressed on square grids. In two-dimensional bond percolation, on a 1000×1000 grid, the percolation thresholds for triangular, square, and honeycomb lattices, exact values of which are approximately 0.3473, 0.5 and 0.6527 respectively, have been found as 0.3469 ± 0.0016 for the triangular lattice, 0.4992 ± 0.0022 for the square lattice, and 0.6510 ± 0.0027 for the honeycomb lattice. Also, critical exponents such as β , γ , ν , and fractal dimension D , values of which are universal in two-dimensions and the exact values are $5/36$ (~ 0.1389), $43/18$ (~ 2.3889), $4/3$ (~ 1.3333), and $91/48$ (~ 1.8958), respectively, have been found as 0.1389, 0.1386, and 0.1390 for β ; 2.3886, 2.3272, and 2.3275 for γ ; 1.3326, 1.3392, and 1.3168 for ν ; and finally 1.8801, 1.8729, and 1.8932 for D in triangular, square and honeycomb lattices, respectively. The results of the simulations which are quite convenient for all three different lattices showed that the direction-cut method is a strong candidate to be an effective and relatively easy algorithm for percolation simulations targeted to different lattice types and potentially in different dimensions.

Keywords: Percolation Theory, Percolation Threshold, Critical Exponents, Direction-Cut Method.

1. Giriş

Staudinger'e ait tezin kabul görmesinin ardından (Feldman, 2008) başlayan polimer çağının en heyecanlı dönemlerinde, Flory'nin meşhur üç makalesiyle filizlenen perkolasyon teorisine ait serüven (Flory, 1941a, 1941b, 1941c), teoremin isim babaları Broadbent ve Hammersley'in gaz maskeleri üzerine yaptıkları çalışma ile gelişimine devam etmiştir (Broadbent ve Hammersley, 1957). Kristaller ve labirentler alt başlıklı matematik temelli bu çalışma, perkolasyon yaklaşımının farklı alanlarda da söz sahibi olup, Araújo ve arkadaşlarının da ifade ettiği şekilde (Araújo vd., 2014), ölmeye niyeti olmayan dokuz canlı bir kedi gibi, 21. yüzyıla kadar hız kaybetmeden gelişeceğini gösteren ilk işaret kabul edilebilir. Özellikle fizikçi ve matematikçilerin ilgisini sıklıkla çeken teori, iletken malzemeler, kıyı şeritlerine ait fraktallar, çok çeşitli ağ yapıları, türbülans, manyetik modeller, koloitler, büyüme modelleri, tutma kapasitesi ve peyzaj havzaları, spin kuantum Hall geçişi, kafes ayar kuramı, örgülerin cebirsel ve topolojik özelliklerine olasılıksal yaklaşımları gibi çok çeşitli alanlarda, farklı problemlere cevap verecek şekilde, büyük çaplı bir uygulama alanı bulmuştur (Saber, 2015).

1960'lı yılların başında, Fisher ve Essam tarafından yapılan çalışmalarla (Fisher, 1961; Fisher ve Essam, 1961), bilim camiasının gündemine giren küme ölçüleri ve perkolasyon eşikleri, bugün de halen farklı örgüler ve yapılar için en önemli konular arasında yer almaktadır. Ele alınan örgü veya yapının sahip olduğu ağ mimarisi ve boyutuna göre değişen perkolasyon eşikleri ve bununla birlikte, sistemin sadece boyutuna bağlı kalan kritik üsteller konusundaki çalışmalar, bilim insanları tarafından, teorik ve hesaplamalı düzlemde paralel şekilde ele alınmıştır. Bu bağlamda, Stauffer ve arkadaşlarının perkolasyon alanındaki katkıları oldukça önemli olup, yıllar içerisinde, meslektaşları için de büyük bir motivasyon kaynağına sebebiyet vermişlerdir (Stauffer, 1979; Stauffer ve Aharony, 1985; Stauffer vd., 1982). Günümüzdeyse, özellikle mühendislik çalışmalarında uygulama alanı bulan perkolasyon teorisi, ağ yapısı içeren ve/veya faz geçişlerinin görüldüğü sistemleri anlamak, yorumlamak ve şekillendirmek için biçilmiş kaftan rolünü devam ettirmektedir. Yüksek isterlere sahip teknolojilerde gereksinimi her geçen gün artan çoklu fonksiyona sahip kompozit malzemelerin geliştirilmesinden (Trompeta vd., 2019), bu malzemelerin en uygun özelliklerde üretimi adına teorik ve deneysel çalışmalara kadar (Kaynan vd., 2020; Li vd., 2007), ve hava trafiği modellemelerinde dahi (Cook vd., 2015) perkolasyon teorisi, bugün de başvurulması gereken yöntemler arasında halen ilk sıralardadır.

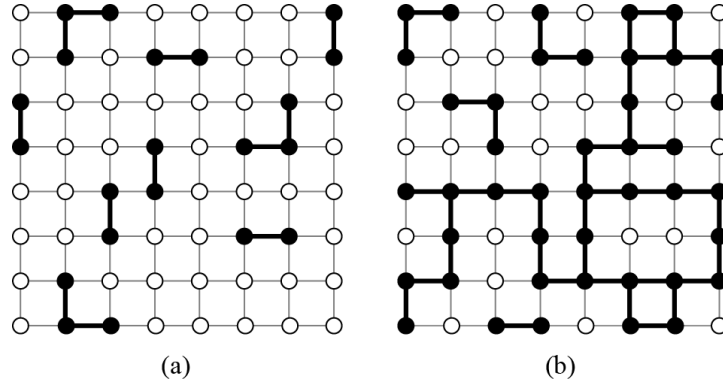
İki-boyutlu perkolasyon çalışmaları, stokastik Loewner evrimi, konformal dönüşüm ve ölçek değişmezliği, mknatıslanma, Ising modeli gibi genel itibarıyla matematiksel modeller ve teorik fizik çalışmaları için tercih edilse de (Saber, 2015), farklı mühendislik alanları (Fernandez-Anez vd., 2017; Winterfeld vd., 1981; Zeng vd., 2019) ve sosyal bilimler (Solomon vd., 2000) için de kullanım alanı sağlamıştır. Teoriye ait gelişimin ilk zamanlarından itibaren oldukça değerli çalışmaların yapıldığı iki-boyutta perkolasyon, kritik perkolasyon eşiklerinin tam değerlerinin bulunduğu ve kritik üstellerin tespit edildiği teorik (Sykes ve Essam, 1963, 1964) ve benzetim (Hoshen ve Kopelman, 1976; Hoshen vd., 1978) çalışmalarına sıklıkla konu olmuş ve halen olmaya da devam etmektedir (Newman ve Ziff, 2001; Vogel vd., 2010; Zerko vd., 2012).

Bu çalışmada, yön-kesme yöntemi olarak isimlendirilen bir algoritma, Monte Carlo temelli benzetim çalışmalarında kullanılmak üzere tanıtılmıştır. Yöntem, iki-boyut özelinde temel Arşimet latisleri üzerinde sınanarak, alınan sonuçlar evrensel davranışlar gözetilerek yorumlanmıştır. Bu yöntemle, iki- ve üç-boyutlu sistemler başta olmak üzere, giriş seviyesinden, daha yüksek derecede karmaşıklık içeren yapılara kadar perkolasyon benzetimleri yapılabilmektedir. Ana yöntem ek olarak, referans numaralarını temel alan kümelenme yapısıyla, oldukça etkili bir çalışma sistemine sahip olan algoritmalar, kümelenme ile ilgili yapılacak çalışmaların da önünü açmakta ve evrensel davranışlarını takip etmeyi kolaylaştırmaktadır. Yöntemi tanıtmak ve perkolasyon benzetimlerine uygun olduğunu göstermek amacıyla, iki-boyutlu Arşimet latisleri üzerinde bağ perkolasyonu seçilmiş olsa da yön-kesme yöntemi, esnek doğası nedeniyle nokta ve bağ perkolasyonu için de kullanılabilir. Latis üzerindeki herhangi bir düğüm noktasına, istenilen sayıda yön veya komşuluk tanımlanabilmesiyle, ihtiyaç olduğu takdirde, yüksek dereceli komşuluklara sahip latislere ve/veya iki-boyut üstündeki sistemlere de cevap verebilmektedir.

2. Perkolasyon Eşiği ve Kritik Üsteller

Perkolasyon teorisi, herhangi bir örgü üzerinde tanımlanmış düğüm noktaları ve bu noktalar arasındaki bağlar üzerinden iki farklı şekilde ele alınabilir. İlgili yaklaşım, eğer düğüm noktalarının dolu olup olmadıkları üzerinden tanımlanıyorsa nokta (site) perkolasyonu, eğer bu düğüm noktaları arasındaki bağlar üzerinden tanımlanıyorsa da bağ (bond) perkolasyonu olarak isimlendirilir. Fakat temel tutum her iki yaklaşım içinde aynıdır. Teoriye göre, herhangi bir anda, işgal edilen noktaların veya kurulan bağların sahip oldukları var olma olasılıkları p ise, boş noktaların veya kurulmamış bağların da olasılığı $1-p$ şeklindedir. p değerinin artmasıyla, yani noktaların dolması veya bağların kurulmasıyla oluşan kümelenmeler sayesinde, öyle bir an gelecektir ki bir küme, diğer oluşan kümelenmeleri de bünyesine katarak, tüm örgüde baskın pozisyona geçecektir. Sonsuz küme (s_∞) olarak isimlendirilen bu küme, sahip olduğu son eleman ile örgünün herhangi karşılıklı iki kenarına dokunacak kadar büyüyecektir. Bir faz geçişinin yaşandığı bu kritik noktadaki olaya perkolasyon, bu özel p değerine ise perkolasyon eşiği (p_c) ismi verilmektedir. Perkolasyonun gerçekleştiği eşik değeri her ne kadar örgüye bağlı olarak değişiklik gösterse de kümelenme ve faz geçişi ile ilgili fenomenler evrensel nitelikte olup, örgünün şeklinden bağımsız olarak sadece örgü boyutuna bağlı olacak şekilde meydana gelmektedirler.

Şekil 1’de, iki-boyutlu bir kare ızgara üzerinde, perkolasyon aşamaları tasvir edilmektedir. Bağ ve nokta perkolasyonu yaklaşımının her ikisi için de örnek olan bu tasvirde, zamanla kurulan bağlar veya işgal edilen noktalar görülmektedir. Başlangıçta küçük p değeri için oluşan kümeler az sayıda ve sahip oldukları eleman sayıları az miktarda iken (Şekil 1a), kritik eşik sonrası sonsuz küme oluşmuş ve sahip olduğu eleman sayısı tüm ızgarayı bir uçtan diğerine kapsayacak şekilde çoğalmıştır (Şekil 1b).



Şekil 1. Kare ızgara üzerinde kümelenme aşamaları; p (a) eşik değerinden küçük bir değere ve (b) eşik değerinin üstünde bir değere sahipken.

ızgara üzerindeki düğüm noktalarının, zamanla birbirleriyle bağ kurmaları, Şekil 1a’da görüldüğü gibi küçük kümelenmelerin başlaması anlamına gelmektedir. Matematiksel olarak bu kümeler ifade edilmek istenirse, s büyüklüğündeki kümelerin dağılım fonksiyonu n_s olmak üzere, kümelerin ortalama büyüklüğü,

$$S(p) = \frac{\sum_s s^2 n_s(p)}{\sum_s s n_s(p)} \quad (1)$$

değerine eşittir. Bu büyüklük, perkolasyon eşiğine yaklaştıkça sonsuza doğru yaklaşırken, davranışı ise,

$$S(p) \propto (p_c - p)^{-\gamma}, \quad p \rightarrow p_c^- \quad (2)$$

şeklinde orantılıdır. Rastgele bir bağın sonsuz kümeye ait olma olasılığı olarak tanımlanan sonsuz kümenin gücü ise,

$$P(p) \propto (p - p_c)^\beta, \quad p \rightarrow p_c^+ \quad (3)$$

olarak verilir. Verilen β ve γ , perkolasyon teorisinde tanımlanan kümelenme dinamikleri ile ilgili kritik üstellerdir. (Stauffer ve Aharony, 1985).

Buraya kadar tartışılan ifadeler için herhangi bir sınırlama konulmamıştır ve verilen denklemler herhangi bir yapı için genel geçer olacak şekilde düzenlenmiştir. Her ne kadar Denk. 2 ve Denk. 3, ızgaranın büyüklüğünden bağımsız olsa da ele alınan ızgaraya ait kenar uzunluğu L olmak üzere, yapılan kabuller $L=\infty$ değerine sahip olduğu durumlar için tanımlanmıştır. $L<\infty$ olduğu durumlar içinse farklı evrensel davranışlar tanımlanabilmektedir.

Korelasyon uzunluğu ζ , aynı kümeye ait iki noktanın ortalama mesafesini temsil eder. Bu ifade, bir mikroskobik değişkenin dalgalanmalarının diğerinin dalgalanmaları ile ilişkili olduğu tipik uzunluk ölçeğinin bir ölçüsünü verir. Sınırlı bir ızgara yapı için ($L<\infty$), korelasyon uzunluğu L 'ye yaklaştığında,

$$|p_c(L) - p_c| \propto L^{-1/\nu} \quad (4)$$

eğilimi geçerlidir (Vogel vd., 2010). Aynı şekilde, $s_\infty(L)$ perkolasyona uğrayan sonsuz kümenin kütlesi olmak üzere,

$$s_{\infty}(L) \propto L^D \quad (5)$$

şeklindedir. Burada da ν ve D kritik üsteller olup, D sistemin fraktal boyutudur ve

$$D = \frac{1}{\sigma\nu} \quad (6)$$

değerine sahiptir. Eşitlikteki σ ise, korelasyon uzunluğu ζ ile ilgili kritik üsteldir (Stauffer ve Aharony, 1985). Son olarak, kritik üstel α 'nın değeri, kümelenme ile ilgili kritik üsteller olan β ve γ ile tespit edilebilir. Aralarındaki ilişki,

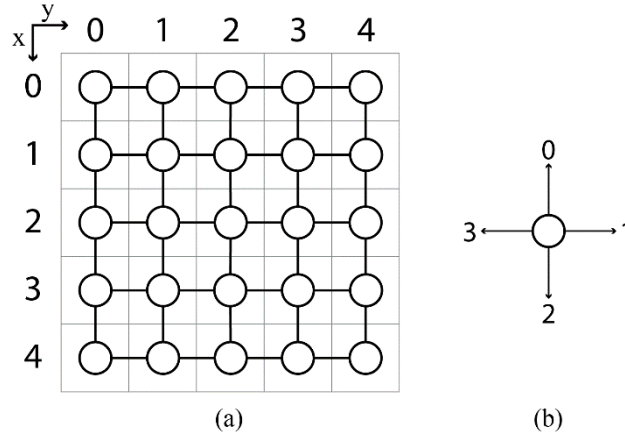
$$2 - \alpha = \gamma + 2\beta \quad (7)$$

eşitliği ile verilebilir (Stauffer, 1979).

3. Temel Yöntem ve Algoritmalar

3.1. Yön-Kesme Yöntemi

Bir düzlemde tanımlanan kare ızgara iki temel elemana sahiptir. Bunlardan ilki, ızgara üzerindeki düğüm noktaları iken, ikincisi ise bu düğüm noktaları arasındaki bağlardır (Şekil 2a). Kare ızgaranın basit tanımı gereği, her bir düğüm noktası kendisine yakın komşu olan diğer dört düğüm noktası ile bağ yapabilmektedir. Herhangi bir düğüm noktasına ait bağlanma yönleri Şekil 2b'de görülebilir. Bu yakın komşularla tanımlanan bağlar birinci dereceden bağlar olup, iki üstü yüksek dereceli, başka bir deyişle uzak komşulara tanımlanan bağlar, kare latis üzerinde tanımlanmamıştır. Fakat ileride de görüleceği üzere, farklı latis yapıları için, kullanılan yöntem ile bir düğüm noktasına dört bağdan fazlası veya daha azı tanımlanabilmektedir.



Şekil 2. (a) Kare ızgara üzerindeki düğüm noktaları ve noktalar arasında tanımlı bağlar. (b) Bir düğüm noktasının komşuları ile arasında tanımlı bağlanma yönleri.

Sınırlı büyüklükte iki-boyutlu bir dizi ile tanımlanmış bir kare ızgara için (ki eğer sayısal bir hesaplamadan bahsediyorsak bilgisayarın sonlu hafızası böyle bir engel teşkil etmektedir), izin verilen birinci dereceden bağ sayıları, her düğüm noktası için artık eşit değildir. Periyodik sınır koşullarıyla tanımlanmamış, dört farklı kenarda konumlanan düğüm noktaları, sınırlarda olmaları nedeniyle artık bir bağdan yoksun kalmışlardır. Üst kenardakiler 0-yönünde, sağ kenardakiler 1-yönünde, alt kenardakiler 2-yönünde ve sol kenardakiler 3-yönünde bir bağa sahip değildir (Şekil 2). Bunlara ilaveten, tam köşelerde bulunan düğüm noktaları, iki kenara denk geldiklerinden, bir komşu daha kaybederek sadece 2 izinli bağa sahip olmaktadır.

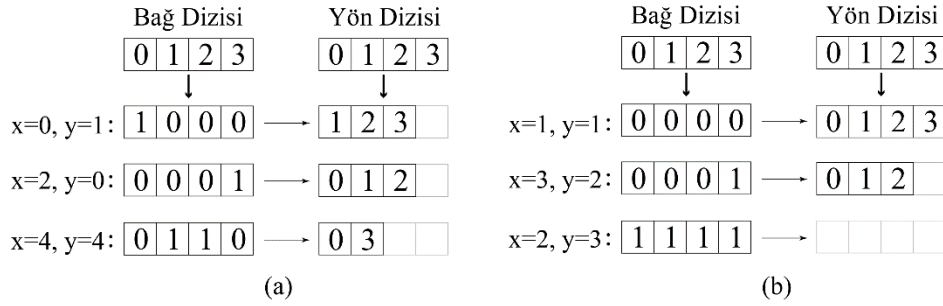
Bu şartlarda tanımlanmış kare ızgara üzerinde, Monte Carlo yöntemi esas alınarak yapılan perkolasyon benzetimi kısaca şu döngüsel aşamaları içermektedir: (1) İki boyutlu bir dizide tanımlanan kare ızgara üzerinde rastgele bir düğüm noktası seç, (2) bu düğüm noktası ile rastgele bir komşusu arasında bağ kur, (3) sistemin herhangi iki kenar arasında perkolasyona uğrayıp uğramadığını kontrol et, (4) eğer perkolasyon gerçekleşmemişse tekrar yeni bir bağ kur. Benzetim bu adımları izlerken dikkat edilmesi gereken durumlar mevcuttur. Bunlardan ilki, rastgele seçilen düğüm noktasının ızgaranın sınırlarında konumlanmış olmasıdır. Böyle bir nokta için mevcut olmayan komşuların ve bağların yok sayılması gerekmektedir. İkincisi, kümelerde yapılmış aynı bağların tekrar kurulmaması ve son olarak da kurulan bağlar sonucu oluşmakta olan sonsuz kümeyi takip edecek, kümenin perkolasyona uğradığı anı tespit edecek bir mekanizmanın kurulması gerekmektedir.

Geliştirilen yön-kesme yöntemi ile, herhangi bir düğüm noktasına ait mevcut komşu bağ yönleri bir dizi ile saklanmaktadır. Kare ızgara için (Şekil 2b) bu dizinin dört elemanı $\{0, 1, 2, 3\}$ şeklindedir. Fakat bu dizinin boyutu istenildiği kadar büyütülerek, yüksek dereceli komşuluklar da tanımlanabilir. Sınırlardaki noktalarında mevcut olmayan bağların ve daha önceden yapılmış olan bağların, kısacası artık izinli olmayan yönlerin, diziden çıkartılması bu algoritmanın en temel yaklaşımıdır. İlgili yönler kesilerek diziden

çıkartılmış, bu sayede izinli ve/veya tanımlı olmayan bağların ve daha önce yapılmış olan bağların tekrar kurulmaya çalışılması engellenmiştir.

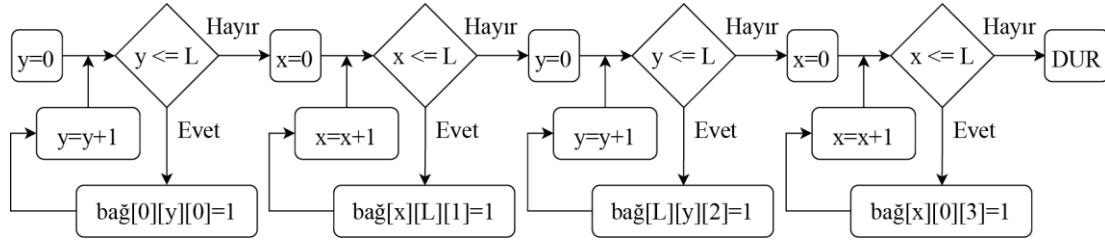
Bu yöntemin uygulanabilmesi için ızgara üzerindeki noktaların her birine ait bir bağ dizisi tanımlanmalıdır. Bu dizide, dört farklı yön içinde yasaklı olan bir yön var ise işaretlenir. Hafıza gözetilerek ilgili dizi sadece dört bit kullanılarak tanımlanabilir. 0 değeri bağlanma izni verirken, 1 değeri ilgili yöndeki bağın yasaklandığını simgeler. Sınırlarda konuşlanmış noktalara ve ızgaranın farklı konumlarındaki noktalara ait örnek dizi düzenlemeleri Şekil 3'te görülebilir. Şekil 3a, kenarlarda bulunan noktaların, izinli olmayan yönlerinin kesilmesine örnek olarak verilmektedir. Sınırlardaki bu işlemler, benzetim başlatılmadan önce gerçekleştirilmelidir.

Benzetim boyunca kurulan bağlar karşılıklı olarak bağın sahipleri iki düğüm noktasınca da işaretlenmelidir. Aralarından bağ kurulmuş düğüm noktalarından ilk seçilen, yön-kesme sonucu seçilen yöndeki bağı işaretlerken, bağ kurmak için seçilen düğüm noktası ise bu yönün zıddına tekabül eden yönü işaretlemelidir. Bu işaretleme kolaylık açısından bir ters yön fonksiyonu tanımlanarak basitçe gerçekleştirilebilir. Şekil 3b ise benzetim sırasında karşılaşılabilecek olan, bu şekilde farklı yönlerden bağlar kurmuş ve henüz hiç bağ yapmamış olan düğüm noktalarından örnekler içermektedir. Her iki örnekte de bağ dizilerine karşılık kesilen yönler sonucu yeniden kurgulanan yön dizileri açıkça görülmektedir. Eğer kesilen yönler sonucu yön dizisi tamamen boşaltılmışsa (Şekil 3b, son örnek), bu ilgili nokta için tüm bağların yapıldığı anlamına gelmekte olup, yeni bir bağ kurulması için rastgele yeni bir nokta seçilmesi gerekmektedir.



Şekil 3. 5 x 5 boyutlarına sahip kare ızgaranın (a) kenarlarında tanımlı düğüm noktalarına ait bağ dizileri ve yön kesme sonucu tanımlanan yön dizileri. (b) Aynı ızgara üzerinde bulunan benzetim sırasında oluşan bağlar sonucu şekillenmiş bağ dizine karşılık gelen yön dizileri.

Bağ dizisi temel alınarak yönlerin kesildiği bu yöntemde, $L \times L$ büyüklüğünde kare ızgara üzerindeki düğüm noktalarına ait dört elemanlı bağ dizilerini şekillendiren algoritma Şekil 4'te görülebilir. Bu işlemlerin sadece sınırlardaki düğüm noktaları için yapıldığına dikkat edilmelidir.

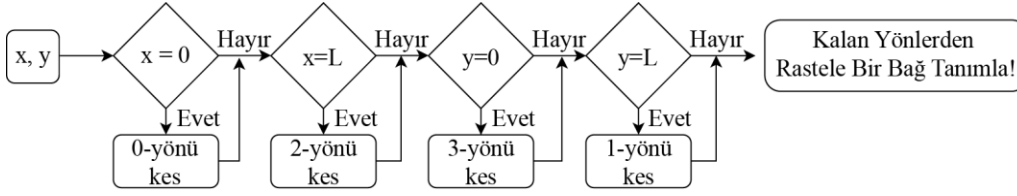


Şekil 4. $L \times L$ büyüklüğünde tanımlanmış kare ızgara üzerinde, toplamda dört döngü ile bağ dizisini şekillendiren algoritma.

Şekil 4'te verilen yöntem ile hazırlanmış bağ dizisi kullanılarak tanımlanacak olan bağ için aşağıdaki adımlar izlenmelidir.

- (1) Yön dizisi $\leftarrow \{0, 1, 2, 3\}$ olarak tanımla.
- (2) Rastgele bir x ve y değeri üret.
- (3) Seçilen adresteki noktaya ait bağ dizisine göre yönleri kes.
- (4) Yeni yön dizisinden rastgele bir komşu seçerek bağ tanımla.
- (5) Yeni kurulan bağı, bağ dizisinde işaretle

Kare ızgara için tanımlanan bu yöntemde, bağ dizilerini oluşturmak, sadece sınırlarda konuşlanan düğüm noktaları için bu tanımlamayı yapmak anlamına gelmektedir. Sınırlar dışında geriye kalan düğüm noktaları, Şekil 3b'de tanımlanan tüm bağları içereceği için ek bir kontrole ihtiyaç duymamaktadırlar. Sayısal olarak ifade etmek gerekirse, $L \times L$ büyüklüğünde bir kare ızgarada, $4L$ kez bağ dizisini şekillendirmek gerekmektedir. Fakat, içerilerde kalan düğüm noktaları, farklı dereceden bağlar içeren ve/veya eksik bağları olan latisler için bu şekillendirmenin her noktayı kapsayacak şekilde $L \times L$ kez yapılması gerekir. Böyle bir yükün benzetim sırasında ortadan kaldırılması için, kare latisin aksine diğer Arşimet latislerinde, bağ dizisini şekillendirmek için kullanılan yön-kesme algoritması, doğrudan bağ kurmak için kullanılabilir. Bu durumda bağ dizileri, kenar işaretleme dışında, benzetimin başında tanımlanmamışken, benzetim sırasında yön-kesme algoritması ile kalan bağlar, bağ dizisine işaretlenecektir. Sırası bu şekilde tanımlanan yöntem ile yön-kesme yöntemi doğrudan bağ kurmak için kullanılırken, benzetim sırasında sürekli şekillenen bağ dizileri ise önceden yapılmış bağların tekrar kurulmasını engelleme kontrolünde kullanılacaktır. Eğer kare ızgarada böyle bir yöntem kullanılmak istenirse, yön-kesme algoritması Şekil 5'te görüldüğü gibidir.



Şekil 5. $L \times L$ büyüklüğünde kare ızgara için yön-kesme algoritması.

3.2. Kümeler ve Perkolasyon Takibi

Izgara yapı üzerinde iki düğüm noktası arasında bağ kurulması demek, esasen bir kümelenme oluşması anlamına gelmektedir. Benzetim boyunca, her seferinde yeni bir düğüm noktasının bu kümeye eklenmesi ve farklı konumlarda oluşan kümelerin birbirleriyle birleşmesi sonucu, nihayetinde sonsuz kümenin oluşarak perkolasyona uğraması gerçekleşecektir (Stauffer ve Aharony, 1985). Monte Carlo benzetimi yapılırken, bu oluşan bu kümelerin tanımlanması için, ızgara yapıya ait iki-boyutlu dizinin elemanları kullanılabilir. Bu yöntemde, birden başlayarak artan küme numaraları, düğüm noktalarını temsil eden dizi değerleri olarak atanır (Hoshen ve Kopelman, 1976). Böylelikle rastgele seçilen bir düğüm noktasının herhangi bir kümeye ait olup olmadığı, eğer dahil ise hangi kümenin üyesi olduğu tespit edilebilir.

Benzetimin başlangıç koşulunda kare ızgara tanımlanırken, düğüm noktalarını temsil eden diziyeye ait her bir değer sıfır olarak atanır. Bu işlem, başlangıçta kümelemenin olmadığını veya tüm noktalarının serbest ve hiçbir bağa sahip olmadığını temsil eder. İlk noktanın seçilip, ilk bağın yapılmasının ardından kümeleme işlemi başlamış ve 1 numaralı küme kurulmuştur. Böylelikle rastgele seçilmiş olan düğüm noktasının ve yine rastgele seçilmiş olan bağ yaptığı düğüm noktasının, iki-boyutlu dizideki yeni değerleri 1 değerine eşitlenir. Bir sonraki kurulacak olan bağ, eğer bu kümeden uzak bir noktada ise yeni 2 numaralı kümenin doğması, aynı kümenin elemanlarından birinin yapacağı bir bağ ise de 1 numaraları kümeye yeni bir üye katılacağı anlamına gelmektedir (Şekil 6).

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

(a)

0	0	0	0	0
0	1	0	0	0
0	1	0	0	0
0	0	0	0	0
0	0	0	0	0

(b)

0	0	0	2	0
0	1	0	2	0
0	1	0	0	0
0	0	3	3	0
0	0	0	3	0

(c)

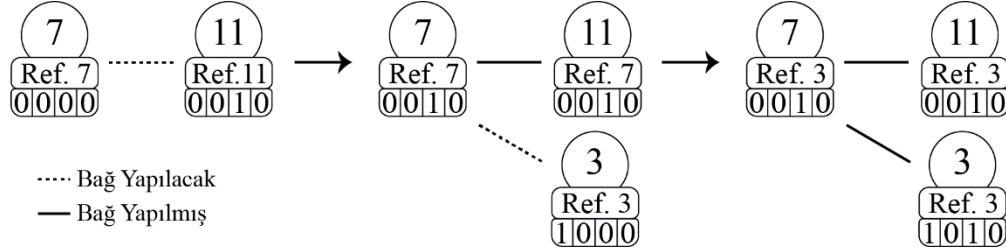
Şekil 6. 5×5 boyutlarına sahip kare ızgaranın (a) başlangıç koşulundaki, (b) ilk yapılan bağ sonrası ve (c) birden çok kümelenmenin oluştuğu sıradaki iki-boyutlu dizi değerleri.

Yapılan bağlar sonucu perkolasyona uğrayacak sonsuz kümenin takibi ve perkolasyona uğradığı anı tespit edebilmek için bir yöntem daha ihtiyaç duyulmaktadır. Yüksek ihtimalle burada uygulanabilecek en verimsiz yol, her yeni bağ sırasında, ızgarayı temsil eden dizinin baştan sona taranarak tüm kümelere ait bilgilerin çıkarılmasıdır. Özellikle iki farklı kümenin birleşmesi sırasında, kümeleri temsil eden iki farklı numaranın tek bir değere çevrilmesi için, tüm dizideki elemanların tek tek elden geçirilmesi, benzetimin süresi konusunda oldukça ciddi gecikmelere sebep olacaktır. Böyle bir yaklaşım, benzetimde ızgara boyutuyla orantılı şekilde yavaşlamalara neden oluşturacaktır.

Bir küme için, bu olumsuz etkileri bertaraf edebilmek adına, kümenin sahip olduğu bağ sayısı, kare ızgaranın kenarlarına değme bilgisi ve referans numarasını içeren bir sınıf tanımlanabilir. Bu sınıf kullanılarak, tanımlanacak dizinin her bir elemanı ızgara üzerinde oluşacak kümelere ait tüm bu bilgileri içerecektir.

Kenar değme bilgileri, bir kümenin ızgaranın dört farklı kenarlarından hangisine komşu olduğunu veren bilgidir. Bağ dizisinde olduğu gibi, sadece dört bitlik bir dizi ile saklanabilir olan bu değerler, ilgili kümenin perkolasyona uğrama durumu konusunda karar verilmesini sağlamaktadır. Kenar değme verileri, kare ızgarayı temsil eden iki-boyutlu dizinin de bir sınıf olarak tanımlanması ile elde edilebilir. Bu durumda, kare ızgaranın dört sınırında yer alan düğüm noktalarına ait kenar değme bilgileri, benzetimin başında tanımlanarak, tüm işlemler boyunca saklanır. Bu düğüm noktaları herhangi bir kümenin elemanı oldukları andan itibaren kenar değme bilgilerini kendi kümelerine aktararak küme sınıfında tanımlı değme dizisini yeniden şekillendirirler. Sadece sınırlardaki düğüm noktalarına konulması gereken bu işaretlemeler, Şekil 4'te verilen bağ dizilerinin tanımlanması sırasında gerçekleştirilebilir. Her bir düğümde bağ dizisine ek olarak aynı şekilde kenar işaretlemelerini içeren değme dizisi de işaretlenir.

Küme sınıfında referans numarası tanımlanmasının sebebi ise kümelerin birleşmeleri sırasında sağlatılmak istenilen kolaylıktır. İki farklı kümenin birleşmesi durumunda, tüm ızgaranın taramasını yapmak yerine, kümeleri referans numarasıyla temsil etmek, birleşim sırasında kümeler arasında sadece referans numarası alışverişini yapılabilmesini mümkün kılar. Böylelikle düğüm noktasına kaydedilen küme numarası hiç değiştirilmeden sadece referans numarası üzerinden kontrol sağlanacaktır. Düğüm noktası farklı bir küme numarası ile işaretlemiş olsa da mensubu olduğu kümenin özellikleri esasen referans numarasında saklı olan gerçek kümenin özelliklerini taşıyabilecektir. İki küme arasında gerçekleşen bağ sonucu referans numaralarının taşınmasına dair bir örnek Şekil 7’de görülebilir. Şekilde aynı zamanda kenar değme bilgilerinin aktarılışı da betimlenmiştir. Tercih edilen yöntem, iki farklı numaradan küçük olanın büyük olana referans olması ve referans numarasını teslim etmesidir. Nasıl ki bir küme, yeni bir düğüm noktasını üye olarak kabul ederken kenar değme bilgilerini de alıyorsa, iki küme birleşirken de birbirlerine kenar değme bilgilerini miras bırakırlar.



Şekil 7. Farklı küme ve referans numaralarına sahip kümelerin bağlanması. Şekilde 7 numaralı küme, önce 11 sonra 3 numaralı küme ile bağ kurmaktadır. Kurulan bağlar sonucu tüm kümeler en küçük numara olan 3 referans numarasına sahip olurlar. Yapılan tüm bağlarda alttaki kenar değme dizisine ait bilgileri de aktarırlar.

Benzetim sırasında karşılaşılabilecek bağlanma türleri üç şekildedir: (1) Hiçbir kümeye ait olmayan iki düğüm noktasının bağlanması (yeni bir küme tanımlanır), (2) bir küme ile herhangi bir küme elemanı olmayan bir düğüm noktasının (veya tam tersi) bağ kurması ve (3) iki farklı kümenin aralarında bağ kurarak tek bir kümeye dönüşmesi. Bir önceki başlıkta tanıtılan yön-kesme yöntemi ile rastgele seçilen komşu düğüm noktasının ardından burada sayılan üç durumdan birine hazırlıklı olmak gerekmekte olup, algoritma kurulurken bu ihtimaller değerlendirilerek hazırlanmalıdır. Ayrıca, bağ dizisinin önceden şekillendirilemediği, yani yön-kesme algoritmasının doğrudan bağ kurma için kullanıldığı durumda, bu ihtimallere bir yenisi daha eklenecektir. Önceden yapılmış bağın tekrar kurulmak istendiği bu durum, aynı küme içerisinde iki düğüm noktasının tekrar bağ yapmaları anlamına gelir. Bu olasılıkta ise bağ dizisinde işaretli olan bu bağın tekrar yapılması engellenerek yeni bir düğüm noktası bağ kurması için seçilmelidir.

Yapılan her bağ sonucu sistemin perkolasyona uğrayıp uğramadığı kontrol edilebilir. Fakat, özellikle büyük sayıda düğüm noktası içeren ızgaralar üzerinde, hiçbir kümeye ait olmayan iki düğüm noktasının bağ kurarak sistemi perkolasyona uğratma ihtimali yoktur. Bu nedenle böyle bağlar kurulduktan sonra perkolasyon sorgusunun yapılmasına gerek yoktur. Fakat diğer iki ihtimalde mutlaka sorgulamanın yapılması gerekmektedir. Zira sistem mutlaka bir kümenin yeni bir düğüm noktasına sahip olması veya daha yüksek bir ihtimalle de iki farklı kümenin bağlanması sonucu perkolasyona uğrayacaktır.

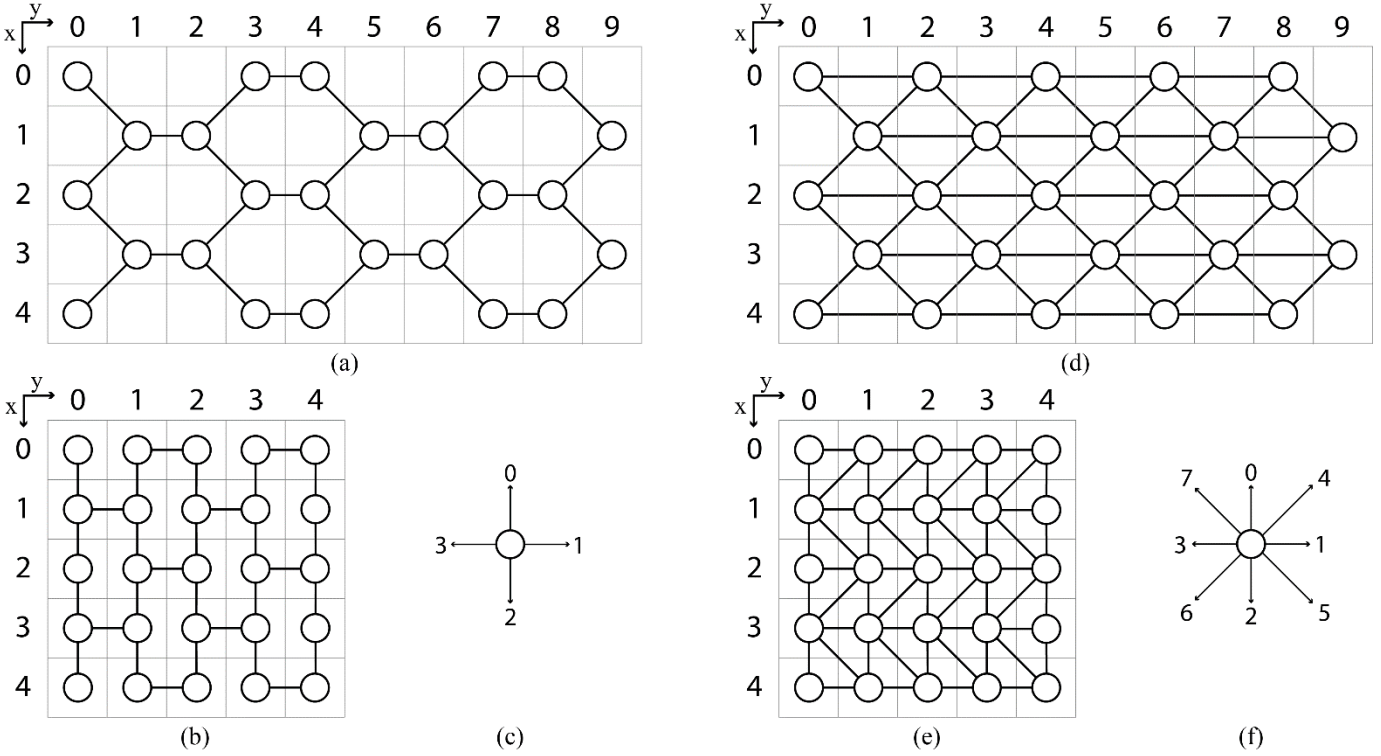
3.3. Kare Iızgarada Adresleme

Arşimet latisleri, her yüzün düzenli bir çokgen olacağı şekilde, düzlem üzerinde tanımlanabilen tepe (düğüm) noktaları geçişli (vertex-transitive) grafiklerdir. Gündelik hayatta en çok karşılaşılan, kare, üçgen ve bal peteği latisler aslen en temel ve bilinen Arşimet latisleridirler. Bu latisler, bir düğüm noktasına denk gelen yüzlerin boyutuna göre, yani ilgili çokgenin kenar sayısına göre isimlendirilirler. Yüz boyutları, alfabetik sırasına göre en küçük olacak şekilde bir yüzle başlayarak listelenir (Parviainen, 2007). Buna göre kare latis (4, 4, 4, 4), üçgen latis (3, 3, 3, 3, 3, 3) ve bal peteği latis ise (6, 6, 6) şeklinde isimlendirilirler. Fakat oldukça uzun olan bu gösterimler yerine genellikle üstel kısaltmalar kullanılmaktadır. Nihayetinde yaygın kullanımda, kare latis (4^4), üçgen latis (3^6) ve bal peteği latis (6^3) gösterimleri ile temsil edilmektedirler.

Arşimet latisleri üzerinde perkolasyon benzetimleri yapmak adına, bu latislerin kare ızgara üzerinde adreslenmesi literatürde de yaygın bir yöntemdir (Parviainen, 2007; Suding ve Ziff, 1999). İlgili latis üzerinden bulunan düğüm noktalarının bir kare ızgara üzerine kaydırılmasıyla adresleme gerçekleştirilir. Kaydırma sonucu yüksek dereceli komşuluklar arası bağlar tanımlanabilir ve/veya tanımlı bağlar iptal edilebilir.

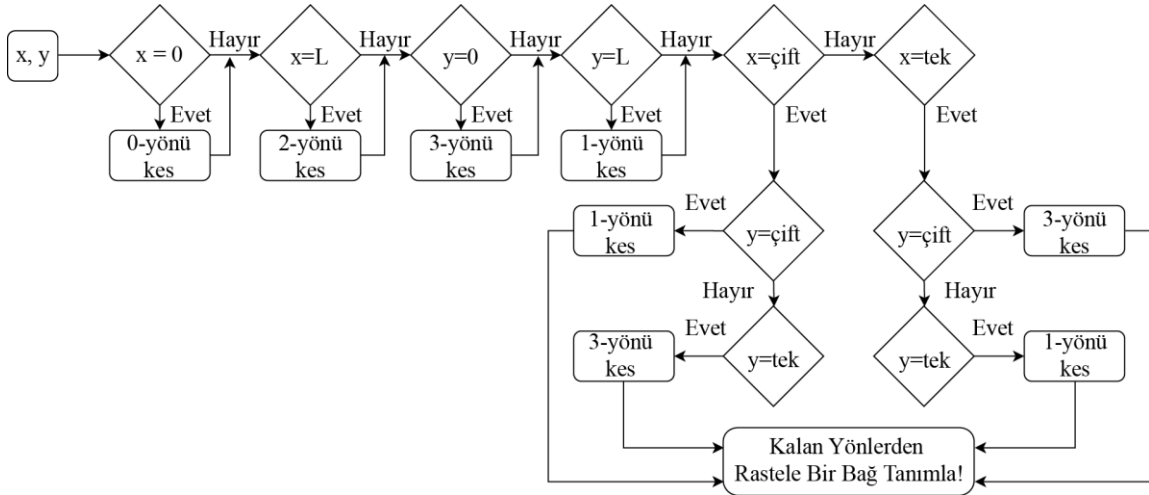
Şekil 8, bal peteği ve üçgen latis için kare ızgara üzerinde yapılan adresleme işlemini sergilemektedir. Kare ızgara üzerine Şekil 8a ve Şekil 8d’de görüldüğü gibi konumlanan latisler, ilgili düğüm noktalarının aralarında herhangi bir boşluk kalmayacak şekilde, sağdan sola doğru taşınması ile yeniden tanımlanmışlardır (Şekil 8b ve Şekil 8e). Düğüm noktalarıyla birlikte taşınan bağlar sayesinde ilgili latisin kare ızgara üzerinde yapılabilir bağları ortaya çıkmıştır. Bal peteği latis için bazı bağlar iptal olurken, üçgen latisde doğası gereği kare latisine ait bağlar korunup, bunlara ek olarak bazı düğüm noktaları arasında ikinci dereceden bağlar tanımlanmıştır.

Her ne kadar bal peteği için kare ızgaraya ait yön dizisi kullanılsa da (Şekil 2b ve Şekil 8c), üçgen latis için dört elemanlı yön dizisi yetersiz kalacağından artık sekiz elemanlı bir yön dizisi kullanılmalıdır. Buna göre, üçgen latis için tanımlanacak olan bağ dizinde Şekil 8f ile gösterilen yönler kullanılabilir.



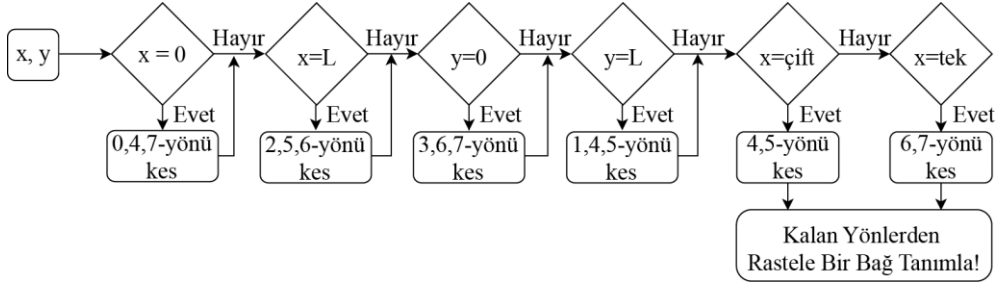
Şekil 8. Bal peteği (a) ve üçgen (d) latislerin kare ızgara üzerinde adreslenmesi. Kare ızgara üzerinde (b) bal peteğine ait, (e) üçgen latis için düğüm noktalarıdır. Adresleme sonrası bağ yönleri ise bal peteği için (c) ve üçgen latis için (f) şeklindedir.

Şekil 8b’de görüldüğü gibi, bir kare ızgara üzerinde tanımlanmış olan bal peteği latisi için yön-kesme algoritması, şekildeki yönüne göre, 90° açıyla döndürülerek adreslenmiş bir bal peteği latis için farklılık gösterecektir. Bu nedenle tanımlanan yöntemler Şekil 8’deki adreslenmiş bir latis için geçerli olacaktır ve iki farklı olasılıktan hangisi seçilirse seçilsin, alınan sonuçlar seçimden etkilenmeyecektir. Buna göre, Şekil 8b’de gösterildiği gibi bir kare ızgarada adreslenmiş, dört elemanlı yön dizisine sahip bir bal peteği latisi için, yön-kesme algoritması Şekil 9’da görüldüğü gibidir. Rastgele üretilen x ve y noktalarının ardından bağ yapılması için izinli olmayan yönleri keserek, izinli yönlerden biriyle bağ kurulmasını sağlayan bu algoritma, benzetim sırasında rastgele seçilen x ve y değerine göre şekillenmektedir.



Şekil 9. $L \times L$ büyüklüğünde kare ızgara üzerinde adreslenmiş bal peteği latisine ait yön-kesme algoritması. Bu algoritma Şekil 8b’de görüldüğü üzere, bir kare latis üzerinde adreslenen bal peteği latisi için geçerlidir.

Şekil 10 ise, Şekil 8e’de gösterildiği gibi bir kare ızgarada adreslenmiş bir üçgen latis için yön-kesme algoritması görülmektedir. Burada sekiz elemana sahip bir yön dizisi temel alınmıştır ve yöntem, kare ızgara üzerinde Şekil 8e’de görüldüğü gibi adreslenen bir üçgen latis için geçerlidir.



Şekil 10. $L \times L$ büyüklüğünde kare ızgara üzerinde adreslenmiş üçgen latisin ait yön-kesme algoritması. Bu algoritma Şekil 8'e'de görüldüğü üzere, bir kare latis üzerinde adreslenen bal peteği latisi için geçerlidir.

Tanımlanan üç farklı latisin sahip oldukları toplam bağ sayıları ise, doğaları gereği birbirinden farklıdır. Toplam bağ (TB) sayıları büyükten küçüğe sırasıyla üçgen, kare ve bal peteği şeklindedir. Şekil 2'de görülen kare latis ve Şekil 8'da görülen kare ızgarada adreslenmiş bal peteği ve kare latis için, bir kenarlarındaki düğüm nokta sayısı L olmak üzere,

$$TB_{\text{Kare}} = 2 \times L \times (L - 1) \quad (8a)$$

$$TB_{\text{Petek}} = L \times (L - 1) + \left(\frac{L - 1}{2} \times L\right) = \frac{3}{2}(L^2 - L) \quad (8b)$$

$$TB_{\text{Üçgen}} = L \times (L - 1) \times 2 + (L - 1)^2 = (3L - 1) \times (L - 1) \quad (8c)$$

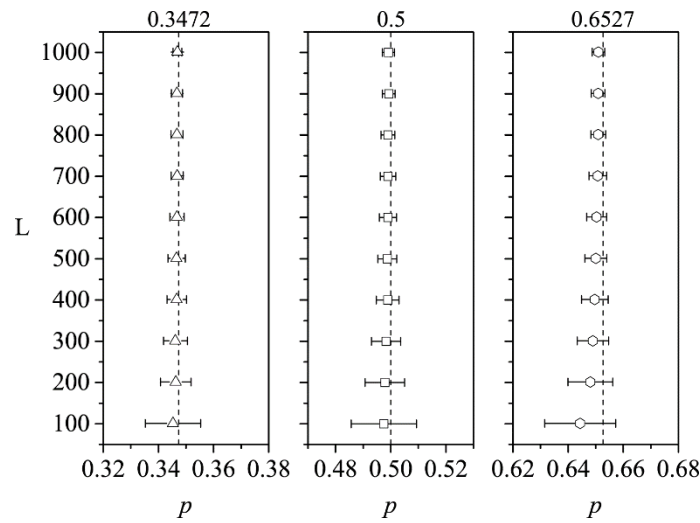
şeklinde ifade edilebilir.

4. Benzetim Sonuçları ve Tartışma

4.1. Perkolasyon Eşikleri

İki-boyutta gerçekleştirilen benzetim çalışmalarında, üçgen, kare ve bal peteği latisler için, her birine ait L değeri 100'den başlamak üzere, 100'er miktarda artırılarak, en son 1000 değerine varacak şekilde farklı ızgaralar kullanılmıştır. 100×100 ve 1000×1000 düğüm noktasına sahip ızgaralar arasında toplamda 10 farklı ölçüde latislerden sonuçlar alınmıştır. Bu farklı ölçülere sahip latisler üzerinde, her benzetim en az 100 (küçük latisleri için en az 1000) kez tekrarlanarak, bulunan sonuçların ortalama değerleri hesaplanmıştır. Kare latis için, perkolasyon eşikini tespit eden örnek bir kod, çalışmanın ekinde görülebilir.

p değerleri, her bir latisin kendisine ait toplam bağ sayıları göz önünde bulundurularak (Denk. 8), kurulan bağ sayılarının, toplam bağ sayılarına oranı şeklinde hesaplanmıştır. Yapılan benzetimlerde, bal peteği ve üçgen latis, kare ızgara üzerinde adreslenerek gerçekleştirilmiştir. Şekil 11'de üç farklı Arşimet latisi için ortalama perkolasyon eşikleri ve standart sapmaları görülebilir.



Şekil 11. Üçgen, kare ve bal peteği latisler için tespit edilen perkolasyon eşikleri ve standart sapma değerleri. Farklı büyüklükteki her latis için en az 100 benzetim yapılmış olup, grafikte bulunan ortalama değerler ve bu değerlere ait standart sapma çizgilerine yer verilmiştir.

Açıkça görüldüğü üzere, her latis türü için, düşük boyutlara sahip ızgaralar üzerinde yapılan benzetim sonuçları, nispeten yüksek sapma ile tespit edilse de ızgara yapı büyüdükçe bu sapmalar giderek azalmaktadır. Benzer şekilde, ölçkleme yasalarına uygun olarak,

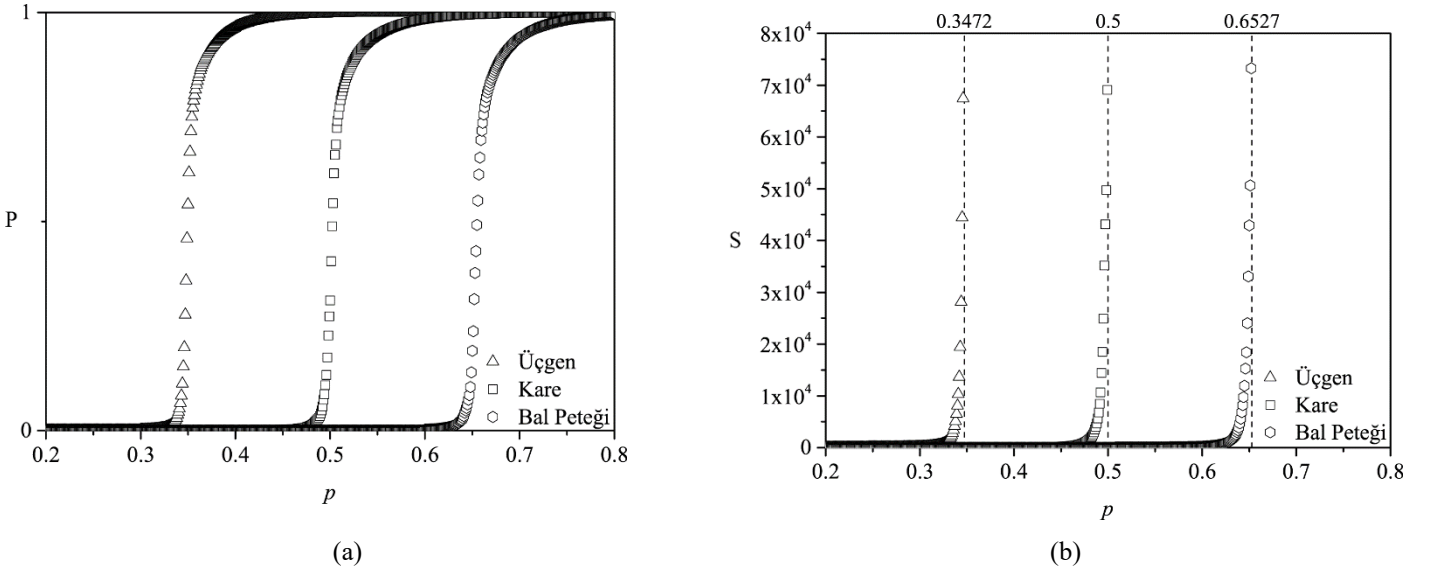
ortalama değerler de bilinen tam değerlere giderek yaklaşmaktadır. Tablo 1, üç farklı latis için, 1000×1000 büyüklüğündeki ızgaralar üzerinde, elde edilen perkolasyon eşiklerine ait değerleri içermektedir.

Tablo 1. Arşimet latisleri için tespit edilen perkolasyon eşikleri. Bu değerler 1000×1000 büyüklüğündeki ızgaralar üzerinde elde edilmiştir.

	p_c (Benzetim Sonucu)	Tam Değer (Stauffer ve Aharony, 1985)
Üçgen	0,3469	$2\sin(\pi/18) \approx 0,3473$
Kare	0,4992	$1/2 = 0,5$
Bal Peteği	0,6510	$1 - 2\sin(\pi/18) \approx 0,6527$

4.2. Evrensel Davranışlar ve Kritik Üsteller

Tamamlanan benzetimlerde latisler üzerindeki kümelenmelere ait dinamikler de incelenmiştir. Benzetim sırasında, latisler üzerinde oluşan sonsuz kümelerin güçleri (P) ve kümelerin ortalama büyüklüğü (S), kurgulanan küme sınıfına ait bağ sayılarını tutan değişkenler sayesinde takip edilebilmiştir. Şekil 12’de, 1000×1000 büyüklüğündeki ızgaralar üzerinde P ve S değerlerinin, her yeni kurulan bağa göre (p 'ye karşılık) değişimleri verilmektedir.



Şekil 12. 1000×1000 büyüklüğündeki ızgaralar üzerinde, üçgen, kare ve bal peteği latisler için, (a) oluşan sonsuz kümelerin güçleri (P) ve (b) kümelerin ortalama büyüklüğü (S).

Tüm latisler için, benzetimler sırasında şekillenen kümelenmelerle ilgili her iki eğilim $P(p)$ ve $S(p)$, perkolasyon teorisinde öngörüldüğü gibi gerçekleşmektedir. Perkolasyon anında oluşan sonsuz kümenin gücü, latislere ait perkolasyon eşiklerinden itibaren hız kazanarak, zamanla tüm kümeyi kapsayacak değere ilerlemektedir. Bununla birlikte, kümelerin ortalama büyüklükleri ise, perkolasyon değerine kadar hızla artarak sonsuza doğru gitmektedir.

$P(p)$ ve $S(p)$ eğrileri beklendiği gibi değişse de davranışlarına ait doğru kriter, kritik üstellerin eğriler üzerindeki değerlerinin tespit edilmesidir. Bu eğrilerden, ilgili kritik üstellerin ölçülebilmesi için Denk. 2 ve Denk. 3 kullanılmalıdır. İki ifade arasındaki orantısal ilişkiyi içeren bu denklemler kullanılarak kritik üstellerin elde edilebilmesi adına, her iki taraftaki ifadelerin logaritması alınabilir. Bu işlem sonucu denklemler, $y = mx + a$ şeklinde bir lineer eşitliğe dönecektir ve buradaki eğim değeri ilgili kritik üstele eşit olacaktır. Aşağıda Denk. 2 ve Denk. 5 arasındaki ifadeler için elde edilen eşitlikler görülebilir.

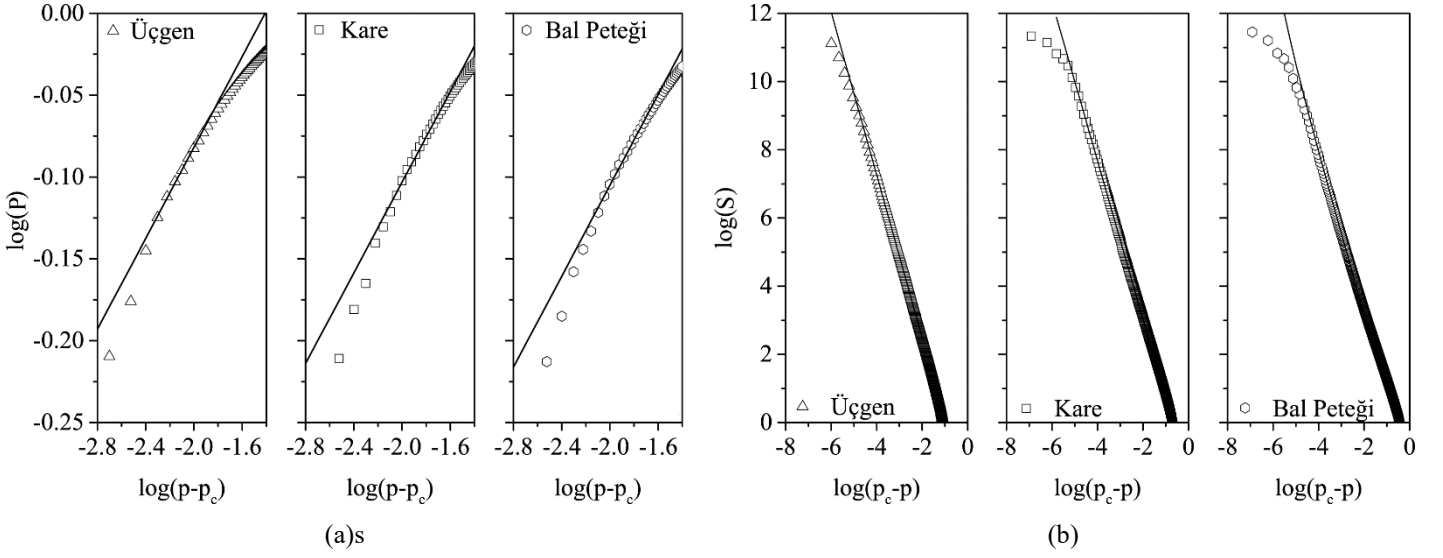
$$\log S \sim \log A_1 + \beta \cdot \log(p - p_c) \quad (9a)$$

$$\log P \sim \log A_2 - \gamma \cdot \log(p_c - p) \quad (9b)$$

$$\log \Delta p_c \sim \log A_3 - \frac{1}{\nu} \cdot \log L \quad (9c)$$

$$\log s_\infty \sim \log A_4 + D \cdot \log L \quad (9d)$$

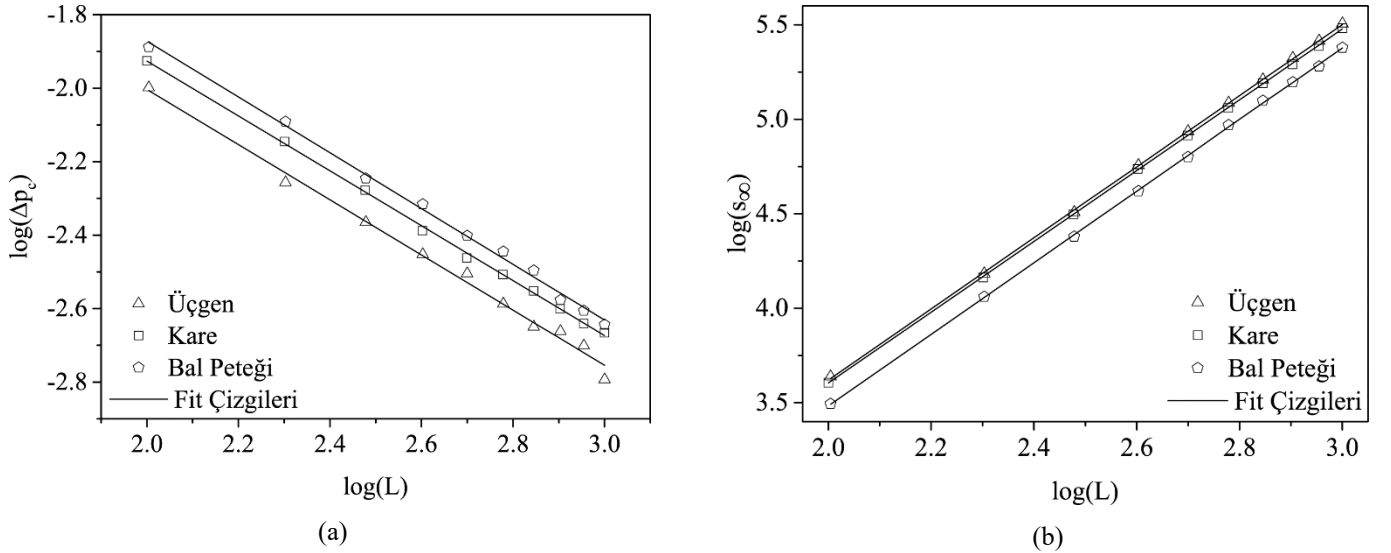
Buna göre, Denk. 9a ve Denk. 9b özelinde, logaritmik değerler ile çizdirilen $P(p)$ ve $S(p)$ değerleri ve uydurulan fit eğrileri Şekil 13'te görülebilir. Bu eğrilerde kullanılan p_c değerleri, literatürde belirtildiği gibi, sonsuz kümenin gücüne ait birinci türevin tepe noktasına karşılık gelen p_m değeri alınarak çizilmiştir (Tüzel vd., 2000).



Şekil 13. 1000×1000 büyüklüğündeki ızgaralar üzerinde, üçgen, kare ve bal peteği latisler için, (a) P ve (b) S değerlerine ait log-log grafikleri. Kritik üstellerin tespit edilebilmesi için yapılan fitler de grafikler üzerinde gösterilmektedir.

Kritik üstelleri bulmak için yapılan eğri uydurma işlemleri sonucu elde edilen β ve γ değerleri Tablo 2'de verilmiştir. Bulunan bu değerler, iki-boyutta tam değerleri bilinen kritik üstel değerleri ile büyük bir oranda uyum içerisindedir.

Ölçeklendirme ile ilgili diğer üsteller için Denk. 4 ve Denk. 5'ten faydalanılmıştır. İlk iki üstelde olduğu gibi, logaritmik yaklaşım bu üstelerde de uygulanmıştır. Latislerde, korelasyon uzunluğu ile ilgili kritik üstellerin tespit edilebilmesi adına, bulunan perkolasyon eşiklerine ait sapma değerleri, latis büyüklüklerine karşı kullanılmıştır (Denk. 9c). Fraktal boyutun tespiti içinse, perkolasyon anında oluşan sonsuz kümelerine ait eleman sayıları, latis büyüklüklerine karşı çizdirilmiştir (Denk. 9d). Şekil 14'te çizdirilen log-log grafikleri ve fit eğrileri görülebilir.



Şekil 14. Üçgen, kare ve bal peteği latisler için, farklı ızgara boyutlarında elde edilen (a) eşik değerlerinin sapma değerleri ve (b) sonsuz kümelerin eleman sayıları kullanılarak çizdirilen log-log grafikleri.

Yapılan fit eğrilerine ait eğimler ile tespit edilen kritik üsteller ve fraktal büyüklükleri, her bir latis için Tablo 2'de verilmiştir. Ek olarak, benzetimler sonucu elde edilen değerler Denk. 6 ve Denk. 7'de kullanılarak diğer üsteller de hesaplanmıştır. Bu sonuçlar da aynı tablo üzerinde verilmektedir. Görüldüğü üzere, ölçeklendirme ile ilgili bu kritik üsteller de tam değerlere oldukça yakın olarak tespit edilmiştir.

Tablo 2. Arşimet latisleri için tespit edilen kritik üstel değerleri. Bu değerler 1000×1000 büyüklüğündeki ızgaralar üzerinde elde edilmiştir.

	Üçgen	Kare	Bal Peteği	Tam Değer (Stauffer vd., 1982)
β	0,1389	0,1386	0,1390	5/36 (0,1389)
γ	2,3886	2,3272	2,3275	43/18 (2,3889)
ν	1,3326	1,3392	1,3168	4/3 (1,3333)
D	1,8801	1,8729	1,8932	91/48 (1,8958)
α	-0,6664	-0,6044	-0,6055	-2/3 (-0,6667)
σ	0,3991	0,3987	0,4011	36/91 (0,3956)

5. Sonuç

Gerçekleştirilen çalışmada, yön-kesme olarak isimlendirilen yöntem tanıtılmış ve iki-boyutlu Arşimet latisleri üzerinde sınınanarak, alınan sonuçlar ile, yöntemin perkolasyon benzetimleri üzerindeki başarısı kanıtlanmıştır. Latis üzerindeki düğüm noktalarına ait potansiyel bağları bir dizi üzerinde tutan yöntem, esnek yapıyla ihtiyaç duyulduğu kadar bağ veya komşuluk tanımlanmasına izin vermektedir. Çalışılan latis yapısına uygun doğru algoritmalar yardımıyla, ilgili yönlerin kesilerek dizinin şekillendirilmesi sayesinde, kullanım senaryolarına ait ihtimaller giderek artmaktadır. Bu bağlamda, her ne kadar çalışmada ele alınan latis türü 3 ile sınırlandırılrsa da diğer Arşimet latisleri de kare ızgara üzerinde adreslenerek, perkolasyon benzetimleri için kullanılabilir. Burada, literatürde yapılan çalışmalarda, diğer Arşimet latislerinin kare ızgara üzerinde adreslenmesi ile ilgili örneklerle bakılabilir (Parviainen, 2007; Suding ve Ziff, 1999).

Nokta perkolasyonu, özellikle perkolasyon eşiği açısından tamamen farklı sonuçlar içerse de esasen aynı örgünün sadece farklı bir şekilde ele alınması durumudur. Kare latis için örneğin, bağ perkolasyonunda, bir düğüm noktasında toplamda dört potansiyel bağ mevcuttur ve her bir bağ kurulabilme iznine sahiptir. Fakat nokta perkolasyonunda, göz önüne alınan durum, ilgili noktanın işgal edilmediği olduğu için, bir düğüm noktası sadece bir bağ kurabilecektir. Böylelikle, dört farklı yönden herhangi biri kullanılarak gerçekleştirilen bağ ile diğer bağ ihtimalleri ortadan kalkacaktır. Bu nedenle, L kenar uzunluğuna sahip bir ızgarada, nokta perkolasyonu için L^2 (toplam nokta sayısı) kadar bağ kurulabilirken, bağ perkolasyonu için $2L^2-2L$ kadar bağ kurulabilecektir (Denk. 8a). Görüldüğü üzere, bu şartları sağlayacak şekilde kurgulanmış bir kodlama ile yön-kesme yöntemi rahatlıkla nokta perkolasyonu için de kullanılabilir.

İki-boyutun üzerindeki benzetimler adına, ilgili ızgaraya özgü olarak, yön dizisinin eleman sayısı artırılarak ve yön tanımları değiştirilerek perkolasyon benzetimleri gerçekleştirilebilir. Üç boyutta kare latislerle bezenmiş bir küp için, Şekil 2'de verilen ızgaraya bir z eksen ve bu eksen üzerinde tanımlanan iki yeni yön sisteme dahil edilmelidir. Sonradan sisteme eklenen eksen nedeniyle yeni gelen sınır koşullarının (kübün altı yüzeyi) kodlamaya dahil edilmesiyle, yön-kesme yöntemi, iki-boyutta olduğu gibi, üç-boyutta da rahatlıkla işlevini yerine getirmeye devam edecektir.

Yön-kesme yönteminde, bağların kurulması için kullanılan yön dizisinin, bağ dizilerine göre şekillendirmesi, üzerinde perkolasyon ve/veya kümelenme benzetimleri yapılmak istenilen latis ağını özgürce şekillendirme imkânı da sunmaktadır. Her bir düğüm noktasına ait bağ sayısı ve komşuluk derecesi, ilgili noktaya özel olmak üzere, bağ dizisine işlenebilir ve bu yolla tanımlanmış bir dizi düğüm noktasıyla, özgün ağ yapısına sahip bir latis tanımlanabilir. Ardından her bir düğüm noktasına göre kesilen yönler sayesinde, benzetim çalışmaları rahatlıkla tamamlanabilir. Bu tür özelleştirilmiş ağ yapıları, özellikle mühendislik alanlarında yön-kesme yönteminin kullanılmasının önünü açık hale getirmektedir.

Ek. Kaynak Kod

Aşağıdaki kaynak kod, iki sütun şekilde olmak üzere Python 3 dilinde yazılmış olup, 1000×1000 ölçeğinde bir kare latis için perkolasyon eşliğini tespit etmektedir. L değeri (LatisBoy) değiştirilerek, istenilen büyüklükteki ızgara üzerinde perkolasyon eşliği tespiti yapılabilir.

```
import numpy as np

import time

#-----Global Degiskenler-----

class GlbDeg:

    LatisBoy = 1000 # L degeri

    LatisXY = LatisBoy * LatisBoy

    toplamBag = LatisBoy * (LatisBoy - 1) * 2

    kume_say = 1

    bag_say = 0.

#-----

#-----Latis Sinifi-----

class Latis:

    m = None

    degme = None

    bagYon = None

#-----

#-----Kume Sinifi-----

class Kumeler:

    bag = None

    ref = None

    degme = None

#-----

#-----Değer Sifirlama-----

def sifirSet():

    glb.kume_say = 1

    glb.bag_say = 0.

    glb.m = np.zeros((glb.LatisBoy, glb.LatisBoy), dtype=int)

    glb.degme = np.zeros((glb.LatisBoy, glb.LatisBoy, 4),
dtype=bool)

    glb.bagYon = np.zeros((glb.LatisBoy, glb.LatisBoy, 4),
dtype=bool)

    kume.bag = np.zeros(glb.LatisXY, dtype=int)

    kume.degme = np.zeros((glb.LatisXY, 4), dtype=bool)

    kume.ref = np.arange(glb.LatisXY, dtype=int)

# Kenar degme ve bag dizisi isaretleme

for y0 in range(glb.LatisBoy):

    g.degme[0][y0][0] = True

    g.bagYon[0][y0][0] = True

for y0 in range(glb.LatisBoy):

    g.degme[glb.LatisBoy - 1][y0][2] = True

    g.bagYon[glb.LatisBoy - 1][y0][2] = True

for x0 in range(glb.LatisBoy):

    g.degme[x0][0][3] = True

    g.bagYon[x0][0][3] = True

for x0 in range(glb.LatisBoy):

    g.degme[x0][glb.LatisBoy - 1][1] = True

    g.bagYon[x0][glb.LatisBoy - 1][1] = True

#-----

#-----Ters Yon Fonksiyonu-----

def tersYon(y):

    yonler = np.array([2, 3, 0, 1])

    return yonler[y]

#-----

#-----Rastgele Bag Kurma-----

def randAdim():

    x0 = np.random.randint(glb.LatisBoy)

    y0 = np.random.randint(glb.LatisBoy)
```

```
x, y = x0, y0
yon = np.array([0, 1, 2, 3]) # yon dizisi
git = -1
yon_kont = 0

for i in yon: # Yonler kesiliyor
    if g.bagYon[x0][y0][i]:
        yon = np.delete(yon, np.argmax(np.where(yon == i)), 0)
        yon_kont += 1

if yon_kont == 4: # Tum baglar yapilmis
    return 0 # Kenara degme kontrolu yapilmiyor.

adim = np.random.choice(yon)

if adim == 0: # 0 yonu secildi
    x, git = x - 1, adim

if adim == 1: # 1 yonu secildi
    y, git = y + 1, adim

if adim == 2: # 2 yonu secildi
    x, git = x + 1, adim

if adim == 3: # 3 yonu secildi
    y, git = y - 1, adim

# Hicbir kumeye ait olmayan iki nokta
if g.m[x0][y0] == 0 and g.m[x][y] == 0:
    g.m[x0][y0] = glb.kume_say
    g.m[x][y] = glb.kume_say
    g.bagYon[x0][y0][git] = True
    g.bagYon[x][y][tersYon(git)] = True

# Kenar degme isaretleri kaydediliyor
```

```
for i in range(4):
    if g.degme[x][y][i] or g.degme[x0][y0][i]:
        kume.degme[kume.ref[glb.kume_say]][i] = True

glb.kume_say += 1
glb.bag_say += 1
kume.bag[kume.ref[g.m[x0][y0]]] += 1
return 0 # Kenara degme kontrolu yapilmiyor.

# Herhangi bir kumeye veya farkli kumelere ait iki nokta
if g.m[x0][y0] != 0 and g.m[x][y] != 0:
    # Ayni kumeye ait iki nokta
    if kume.ref[g.m[x0][y0]] == kume.ref[g.m[x][y]]:
        g.bagYon[x0][y0][git] = True
        g.bagYon[x][y][tersYon(git)] = True
        glb.bag_say += 1
        kume.bag[kume.ref[g.m[x0][y0]]] += 1
        return kume.ref[g.m[x0][y0]]

# Iki farkli kumeye ait noktalar, kumeler birlesecek!!!
else:
    if kume.ref[g.m[x0][y0]] > kume.ref[g.m[x][y]]:
        buyuk_kume = kume.ref[g.m[x0][y0]]
        kucuk_kume = kume.ref[g.m[x][y]]
    else:
        buyuk_kume = kume.ref[g.m[x][y]]
        kucuk_kume = kume.ref[g.m[x0][y0]]

# Yon isaretleri tasiniyor
for i in range(4):
    if kume.degme[buyuk_kume][i]:
        kume.degme[kucuk_kume][i] = True
```

```
# Buyuk kume numarasi kucuk olana cevriliyor
kume.ref[kume.ref == buyuk_kume] = kucuk_kume

g.bagYon[x0][y0][git] = True
g.bagYon[x][y][tersYon(git)] = True
glb.bag_say += 1
kume.bag[kucuk_kume] += 1
kume.bag[kucuk_kume] += kume.bag[buyuk_kume]
return kucuk_kume

# Kume ve bos nokta karsilasiyor
if g.m[x0][y0] == 0 and g.m[x][y] != 0:
    g.m[x0][y0] = g.m[x][y]
    g.bagYon[x0][y0][git] = True
    g.bagYon[x][y][tersYon(git)] = True
for i in range(4):
    if g.degme[x0][y0][i]:
        kume.degme[kume.ref[g.m[x][y]]][i] = True
glb.bag_say += 1
kume.bag[kume.ref[g.m[x][y]]] += 1
return kume.ref[g.m[x][y]]

# Kume ve bos nokta karsilasiyor
if g.m[x][y] == 0 and g.m[x0][y0] != 0:
    g.m[x][y] = g.m[x0][y0]
    g.bagYon[x0][y0][git] = g.bagYon[x][y][tersYon(git)] =
True
for i in range(4):
    if g.degme[x][y][i]:
        kume.degme[kume.ref[g.m[x0][y0]]][i] = True
```

```
glb.bag_say += 1
kume.bag[kume.ref[g.m[x0][y0]]] += 1
return kume.ref[g.m[x0][y0]]

#-----
#-----Ana Islemler-----
def main():
    sifirSet()
    pc = 0.
    siz = False
while siz is False:
    i = randAdim()
if i is not 0:
    if kume.degme[i][0] and kume.degme[i][2]:
        siz = 1
        pc = glb.bag_say / glb.toplamBag
    if kume.degme[i][1] and kume.degme[i][3]:
        siz = 1
        pc = glb.bag_say / glb.toplamBag
    print("pc =", pc)
    print("--- %s saniye ---" % (time.time() - start_time))
#-----
#-----Benzetim Kosturma-----
if __name__ == "__main__":
    glb = GlbDeg()
    g = Latis()
    kume = Kumeler()
    start_time = time.time()
    main()
#-----
```

Kaynakça

- Araújo, N., Grassberger, P., Kahng, B., Schrenk, K. J. ve Ziff, R. M. (2014). Recent advances and open challenges in percolation. *European Physical Journal: Special Topics*, C, 223, ss. 2307–2321. Springer Verlag.
- Broadbent, S. R. ve Hammersley, J. M. (1957). Percolation processes. *Mathematical Proceedings of the Cambridge Philosophical Society*, 53(3), 629–641.
- Cook, A., Blom, H. A. P., Lillo, F., Mantegna, R. N., Micciché, S., Rivas, D., Vazquez, R. ve Zanin, M. (2015). Applying complexity science to air traffic management. *Journal of Air Transport Management*, 42, 149–158.
- Feldman, D. (2008). Polymer history. *Designed Monomers and Polymers*, 11(1), 1–15.
- Fernandez-Anez, N., Christensen, K. ve Rein, G. (2017). Two-dimensional model of smouldering combustion using multi-layer cellular automaton: The role of ignition location and direction of airflow. *Fire Safety Journal*, 91, 243–251.
- Fisher, M. E. (1961). Critical Probabilities for Cluster Size and Percolation Problems. *Journal of Mathematical Physics*, 2(4), 620–627.
- Fisher, M. E. ve Essam, J. W. (1961). Some Cluster Size and Percolation Problems. *Journal of Mathematical Physics*, 2(4), 609–619.
- Flory, P. J. (1941a). Molecular Size Distribution in Three Dimensional Polymers. I. Gelation. *Journal of the American Chemical Society*, 63(11), 3083–3090.
- Flory, P. J. (1941b). Molecular Size Distribution in Three Dimensional Polymers. II. Trifunctional Branching Units. *Journal of the American Chemical Society*, 63(11), 3091–3096.
- Flory, P. J. (1941c). Molecular Size Distribution in Three Dimensional Polymers. III. Tetrafunctional Branching Units. *Journal of the American Chemical Society*, 63(11), 3096–3100.
- Hoshen, J. ve Kopelman, R. (1976). Percolation and cluster distribution. I. Cluster multiple labeling technique and critical concentration algorithm. *Physical Review B*, 14(8), 3438–3445.
- Hoshen, J., Kopelman, R. ve Monberg, E. M. (1978). Percolation and cluster distribution. II. layers, variable-range interactions, and exciton cluster model. *Journal of Statistical Physics*, 19(3), 219–242.
- Kaynan, O., Yıldız, A., Bozkurt, Y. E., Yenigun, E. O. ve Cebeci, H. (2020). Electrically Conductive High-Performance Thermoplastic Filaments for Fused Filament Fabrication. *Composite Structures*, 237, 111930.
- Li, J., Ma, P. C., Chow, W. S., To, C. K., Tang, B. Z. ve Kim, J.-K. (2007). Correlations between Percolation Threshold, Dispersion State, and Aspect Ratio of Carbon Nanotubes. *Advanced Functional Materials*, 17(16), 3207–3215.
- Newman, M. E. J. ve Ziff, R. M. (2001). Fast Monte Carlo algorithm for site or bond percolation. *Physical Review E*, 64(1), 016706.
- Parviainen, R. (2007). Estimation of bond percolation thresholds on the Archimedean lattices. *Journal of Physics A: Mathematical and Theoretical*, 40(31), 9253–9258.
- Saberi, A. A. (2015). Recent advances in percolation theory and its applications. *Physics Reports*, 578, 1–32.
- Solomon, S., Weisbuch, G., de Arcangelis, L., Jan, N. ve Stauffer, D. (2000). Social percolation models. *Physica A: Statistical Mechanics and its Applications*, 277(1–2), 239–247.
- Stauffer, D. (1979). Scaling theory of percolation clusters. *Physics Reports*, 54(1), 1–74.
- Stauffer, D. ve Aharony, A. (1985). Introduction to Percolation Theory. *Introduction to Percolation Theory*. Abingdon, UK: Taylor & Francis.
- Stauffer, D., Coniglio, A. ve Adam, M. (1982). Gelation and critical phenomena. *Polymer Networks* (C. 44, ss. 103–158). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Suding, P. N. ve Ziff, R. M. (1999). Site percolation thresholds for Archimedean lattices. *Physical Review E*, 60(1), 275–283.
- Sykes, M. F. ve Essam, J. W. (1963). Some Exact Critical Percolation Probabilities for Bond and Site Problems in Two Dimensions. *Physical Review Letters*, 10(1), 3–4.
- Sykes, M. F. ve Essam, J. W. (1964). Exact Critical Percolation Probabilities for Site and Bond Problems in Two Dimensions. *Journal of Mathematical Physics*, 5(8), 1117–1127.
- Trompeta, A.-F., Koumoulos, E., Stavropoulos, S., Velmachos, T., Psarras, G. ve Charitidis, C. (2019). Assessing the Critical Multifunctionality Threshold for Optimal Electrical, Thermal, and Nanomechanical Properties of Carbon Nanotubes/Epoxy Nanocomposites for Aerospace Applications. *Aerospace*, 6(1), 7.
- Tüzel, E., Özmetin, M. S., Yılmaz, Y. ve Pekcan, Ö. (2000). A new critical point and time dependence of bond formation probability in sol-gel transition: a Monte Carlo study in two dimension. *European Polymer Journal*, 36(4), 727–733.
- Vogel, E. E., Lebrecht, W. ve Valdés, J. F. (2010). Bond percolation for homogeneous two-dimensional lattices. *Physica A: Statistical Mechanics and its Applications*, 389(8), 1512–1520.
- Winterfeld, P. H., Scriven, L. E. ve Davis, H. T. (1981). Percolation and conductivity of random two-dimensional composites. *Journal of Physics C: Solid State Physics*, 14(17), 2361–2376.
- Zeng, G., Li, D., Guo, S., Gao, L., Gao, Z., Stanley, H. E. ve Havlin, S. (2019). Switch between critical percolation modes in city traffic dynamics. *Proceedings of the National Academy of Sciences*, 116(1), 23–28.
- Zerko, S., Polanowski, P. ve Sikorski, A. (2012). Percolation in two-dimensional systems containing cyclic chains. *Soft Matter*, 8(4), 973–979.