

İSTANBUL TEKNİK ÜNİVERSİTESİ

FEN EDEBİYAT FAKÜLTESİ

BİTİRME ÇALIŞMASI



BOOLCU AĞLARDA MOTİF İSTATİSTİĞİ İÇİN BİR MODEL

Mehmet Ali Anıl

*Bölümü: Fizik Mühendisliği Bölümü*

Güz 2011

İSTANBUL TEKNİK ÜNİVERSİTESİ

FEN EDEBİYAT FAKÜLTESİ

BİTİRME ÇALIŞMASI



BOOLCU AĞLARDA MOTİF İSTATİSTİĞİ İÇİN BİR MODEL

Mehmet Ali Anıl

*Bölümü: Fizik Mühendisliği Bölümü*

Güz 2011

# Özet

Bu çalışmada, bir Boolcu ağ yapısı önerilmiş, bu ağların dinamikleri incelenmiş, önerilen ağların gen regülasyon ağlarının bazı özelliklerini betimleyebilir olma ihtimali göz önünde bulundurularak, bu ağlardan oluşan bir topluluk genetik algoritmaya tabi tutulmuştur. Bu genetik algoritma sonucunda topluluğun maruz bırakıldığı başarımların Boolcu ağların topolojilerinde nasıl bir etkide bulunduğu, bu ağların zaman içerisindeki motif frekanslarındaki değişim ile incelenmiştir.

# İçindekiler

Özet	ii
Kısaltmalar	v
Şekil Listesi	vi
1 Giriş	1
2 Boolcu Ağlar	3
2.1 Boolcu Ağlar	3
2.1.1 Maske Dinamiği	6
2.1.2 Boolcu ağların durum uzayı	7
2.2 Genetik Algoritma ile Boolcu ağların evrimi	8
2.2.1 Genetik Algoritma	8
3 Boolcu Ağların Evrimi	12
3.1 Üretilen Boolcu Ağların Özellikleri	12
3.2 Genetik Algoritmanın Öğeleri	15
3.2.1 Dereceyi Koruyan Mutasyon	16
4 Sonuçlar ve Tartışma	19
4.1 Rastgele Yaratılan Ağların Motif Frekansları	19
4.2 Derece Koruyan Mutasyonun Topluluktaki Motif Sıklığına Etkisi	20
A Tanımlar	30
A.1 Çizgeler	30
A.1.1 Çizge	30
A.1.2 Eşitlik	31
A.1.3 Komşuluk Matrisi	31
A.1.4 Yönelimli Çizge	32
A.1.5 Çizge Eşyapılılığı	32
A.1.6 Derece	33
B Algoritmalar	35
B.1 Çizge Eşyapılılığı Algoritması	35
B.2 Çizgenin tek parçalılığının kontrolü	36
B.3 Motif Frekansları Çıkaran Algoritma	36
B.3.1 Tüm Olası Motiflerin Çıkarılması	36
B.3.2 Motif Frekanslarını Çıkaran Algoritma	37

B.4	Çalışmada Kullanılan Kod . . . . .	37
<b>Index</b>		<b>40</b>

# Semboller ve Kısaltmalar

$\mathbb{B}^n$	n boyutlu Boole uzayı
$\mathbb{Z}$	tamsayılar kümesi
$\phi$	başarı fonksiyonu
$\Sigma$	bir ağın durumu
$\Sigma^{star}$	bir ağın elemanları elenmiş durumu
$\sigma_i$	i. düğümün durumu
$\mathbf{k}^{star}_i$	i. düğüme atanan elemanları elenmiş anahtar
$\mathbf{k}_i$	i. düğüme atanan anahtar
$A$	bir çizgenin komşuluk matrisi
$A \cong B$	A ile B eşyapılıdır. ( <i>isomorphism</i> )
$A \rightarrow B$	A kümesinden B kümesine yapılan bir atama
$a \rightarrow b$	a elemanını b elemanına atayan bir atama
$D$	köşegeninde düğüm derecelerinin bulunduğu matris
$E$	bir çizgenin kenar kümesi
$f_i$	i. düğümün Boolcu fonksiyonu
$G$	Tanım 3.1.1’da verilen başarı fonksiyonu
$L$	Laplasyan matrisi
$M$	mutasyon operatörü
$N$	bir topluluğun içerisinde eleman sayısı
$n$	bir ağın içerisindeki düğüm sayısı
$p$	iki düğümü bağlantılandırma olasılığı
$U(\cdot)$	Heaviside fonksiyonu
$V$	Bir çizgenin kenar kümesi

# Şekil Listesi

1.1	Gen Regülasyonunun Şematik Gösterimi . . . . .	1
2.1	Boolcu Ağın Gösterimi . . . . .	4
2.2	LUT dinamiğine sahip bir Boolcu ağ . . . . .	4
2.3	Boolcu ağın durumunun zamanda evrimi . . . . .	7
2.4	Boolcu ağın temsili bir faz uzayı . . . . .	7
2.5	Başarım fonksiyonu, topluluktaki bireyler, ve genetik algoritmanın başarım eşiği değeri . . . . .	9
2.6	Genetik algoritmanın kopyalama ve mutasyon safhası . . . . .	10
2.7	Genetik algoritmanın eleme safhası . . . . .	10
2.8	Boolcu ağın temsili bir faz uzayı . . . . .	11
3.1	Çeşitli $p$ bağlandırma olasılıkları için ağlardaki derece dağılımı . . . . .	12
3.2	Bir ağın durum başlangıç durumuna göre . . . . .	13
3.3	Bağlantılandırma olasılığına göre ağların dinamiği . . . . .	14
3.4	Bağlama olasılığı $p \in [0, 1]$ a göre $G$ ve $\sigma_G$ standard sapma değerleri. Burada noktalar ağların aldığı $G$ değerleri, düz çizgi ise bu değerlerin standard sapmasıdır. Bazı ağlar aynı $G$ değerini almaktadır, dolayısı ile her nokta birden fazla ağa denk düşebilmektedir. . . . .	15
3.5	Dereceyi Koruyan Mutasyon'a hazırlanan çizge . . . . .	16
3.6	Dereceyi koruyan mutasyonun ilk safhası . . . . .	17
3.7	Dereceyi koruyan mutasyonun ikinci safhası . . . . .	17
3.8	Dereceyi koruyan mutasyonun üçüncü safhası . . . . .	17
3.9	Dereceyi koruyan mutasyon uygulandıktan sonra . . . . .	18
4.1	Üçlü motiflerin frekansları . . . . .	20
4.2	Üçlü motiflerden 1, 2, 3, 4, 5 ve 13 numaralı motifler. . . . .	20
4.3	$\tau = 0$ anında $M_k$ mutasyonunun uygulandığı genetik algoritmadaki Motif sıklıkları . . . . .	21
4.4	$M_k$ ile Genetik Algoritmaya tabii tutulan topluluğun $\langle k \rangle$ tüm düğümler üzerinden derecelerinin ortalamasının zamana göre değişimi . . . . .	22
4.5	$M_k$ ile Genetik Algoritmaya tabii tutulan topluluktaki $\langle G \rangle$ ortalama başarımının zamana göre değişimi. GA'da düşük değerler tercih edilmiştir. . . . .	22
4.6	$\tau = 40$ anında $M_k$ mutasyonunun uygulandığı genetik algoritmadaki Motif sıklıkları . . . . .	23
4.7	$\tau = 180$ anında $M_k$ mutasyonunun uygulandığı genetik algoritmadaki Motif sıklıkları . . . . .	23
4.8	$\tau = 320$ anında $M_k$ mutasyonunun uygulandığı genetik algoritmadaki Motif sıklıkları . . . . .	24

4.9	$\tau = 480$ anında $M_k$ mutasyonunun uygulandığı genetik algoritmadaki Motif sıklıkları . . . . .	24
4.10	$\tau = 0$ anında $M_k$ mutasyonunun uygulandığı genetik algoritmadaki Motif sıklıkları . . . . .	25
4.11	$M_k$ ile Genetik Algoritmaya tabii tutulan topluluğun tüm düğümler üzerinden derecelerinin ortalamasının zamana göre değişimi . . . .	26
4.12	$M_k$ ile Genetik Algoritmaya tabii tutulan başarımının zamana göre değişimi. GA'da düşük değerler tercih edilmiştir. . . . .	26
4.13	$\tau = 40$ anında $M_k$ mutasyonunun uygulandığı genetik algoritmadaki Motif sıklıkları . . . . .	27
4.14	$\tau = 180$ anında $M_k$ mutasyonunun uygulandığı genetik algoritmadaki Motif sıklıkları . . . . .	27
4.15	$\tau = 320$ anında $M_k$ mutasyonunun uygulandığı genetik algoritmadaki Motif sıklıkları . . . . .	28
4.16	$\tau = 480$ anında $M_k$ mutasyonunun uygulandığı genetik algoritmadaki Motif sıklıkları . . . . .	28
A.1	Çizgenin Temsili . . . . .	30
A.2	Çizgelerde Eşyapılılık (Isomorphism) . . . . .	32
A.3	Çizgelerde Eşyapılılık (Isomorphism) . . . . .	33



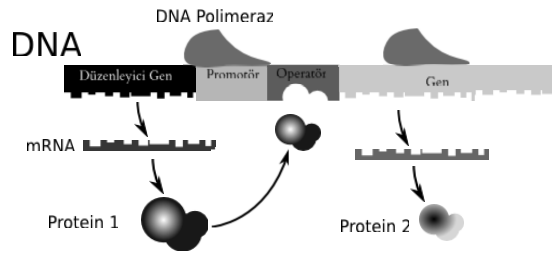


# Bölüm 1

## Giriş

Biyolojik sistemler, genetik kodlarından üretilen proteinler ile yaşamsal faaliyetlerini sürdürebilir. Bu sistemler ile ilgili bir önyargı, genin sadece okunan, sadece bilgiyi taşıyan bir obje olduğu, sahip olduğu kodun işlevselleşebilmesi için RNA polimeraz enziminin yardımı ile bir RNA'nın, dolayısı ile sonunda bu RNA dan bir proteinin oluşması ve ancak bu proteinin hücre içindeki dinamiğe katkıda bulunacağıdır. Yirminci yüzyılın başında, biyolojik sistemlerin kimyasal analizleri daha detaylı yapılabilmesinin getirisi olarak, genetik materyalin de hücre içerisindeki kontrol mekanizmalarında birincil dereceden etkin olabildiği gözlenir. Bu gelişmeler genin, o dokunulmaz, optimize bir kod olduğu fikrini, dinamik, adaptasyonda ve işlevsellikle elçi olarak bir proteine gereksinim duymayan bir obje olduğu fikrine bırakır. [5]

DNA'nın kendi içerisinde transkripsiyonunu düzenleyen, bir yapısı vardır. DNA, transkripsiyon sonucunda üretilen proteinlerin bağlanabildiği, başka bir yerdeki genetik kodu etkinleştiren ya da susturulmasında rol oynayan bölgelere sahiptir. [2]



Şekil 1.1:

Şekil 1.1 de görüldüğü gibi, Protein 2 nin üretimini sağlayan gene bu genden mRNA oluşturabilmek için DNA nın ikili sarmalını açan bir enzimin, RNA Polimeraz'ın promotör alanına (*promoter*) tutunup, geni başından sonuna kat etmesi gerekmektedir. Bu sürecin gerçekleşmesi bu örnekte, operatör alanına (*operator*) ancak protein 1 bağlı olmadığı zaman söz konusu olsun. Burada protein 1, baskılayıcı pro-

tein olmaktadır. Protein 1 ise genin başka bir kısmında, ki burada düzenleyici gen yazarak belirtilmiş yerdeki kod okunarak üretilmektedir. Burada düzenleyici genin diğer gene olan kontrolü net bir biçimde görünmektedir, eğer düzenleyici gen aktif ise, protein 2 yi üreten gen baskılanacak , düzenleyici gen bir nedenden dolayı çalışmazsa da protein 2 sentezlenmeye başlayacaktır.

DNA, bu tip baskılayıcı ya da tetikleyici bir etkileşmeler bütünüdür, ve aracı olarak da sentezlenen proteinler kullanılmaktadır. Bu proteinler, dış etkenlere hassas olduklarından, gen, bu yapı aracılığı ile dış ortama da uyum sağlayabilir. Mesela, *E. Coli* bakterisinin laktoz'u parçalayan proteinleri salgılamasının koşulu, laktoz moleküllerinin baskılayıcı proteinlere bağlanması, onların operatör alanına bağlanamamalarına neden olmasıdır. Sonuç olarak laktozu sindirecek proteinlerin transkripsiyon ile üretimi, laktozun varlığı ile ayarlanmaktadır. [2]

DNA, bu şekilde bakıldığında çok büyük bir kontrol ağı ile modellenenebilir. Bu tip ağlara **gen regülasyon ağları** denilmektedir, ve bu ağlar biyolojik sistemlerin işlevselliğini açıklamada temel bir rol oynar. Gen regülasyon ağlarının karmaşık açma - kapama (etkinleştirme - baskılama) mekanizmasını rastgele Boolcu ağların dinamiği ile açıklama fikri 1969'da Kauffman tarafından sunulur. [8] [7] Bu bakış açısı, biyolojik sistemlerin görünürdeki işlevselliğini niceliksel bir analiz ile inceleme imkanı doğurmuştur. Artık belli proteinleri sisteme tanıtır ya da belli proteinleri yok edip neden sonuç ilişkilerini bir nevi amaç ya da tasarımı fikri, yahut ilüzyonu altında incelemektense, rastgele kurulmuş yapılarda daha önceden amaçlandığını varsaydığımız sonuçların çıkabileceğinin fikri oldukça heyecan vericidir. Bu çalışmada, modifiye edilmiş Kauffman ağlarından oluşan bir topluluk, genetik programlama içeren bir simülasyon altında ele alınmaktadır. Sonuçta S.A. Kauffman'ın bahsini ettiği gib proto-canlılar basit Boolcu ağlar ile modellenenebilir sistemler idiyseler, belli bir başarı kriteri ile bu ağların evrimi incelenebilir olmalıdır. Bu bitirme çalışmasında bu yönetime sahip bir çaba açıklanacak ve gözlemler paylaşılacaktır.

## Bölüm 2

# Boolcu Ağlar

### 2.1 Boolcu Ağlar

Bu bölümde Boolcu ağların özelliklerine değinilecektir. Boolcu ağlar, düğümleri 0 ya da 1 Boolcu değerlerini alabilen, ve her  $t + 1$  zamanı için olan durumları birbirlerinin  $t$  anındaki değerlerinin durumlarının bir fonksiyonu olan çizgelerdir. Ağ yapılarını ilgilendiren tanımlar Ek A da, kullanılan algoritmalar Ek B de verilmiştir.

#### Tanım 2.1.1 (Boolcu Ağlar)

[6] Boolcu ağlar, düğümleri  $i$ . düğümü  $\sigma_i \in \{0, 1\}$  durumunu alan, ve kendi durumu  $\Sigma = (\sigma_1, \sigma_2, \dots, \sigma_n)$  ile betimlenebilen çizgelerdir.

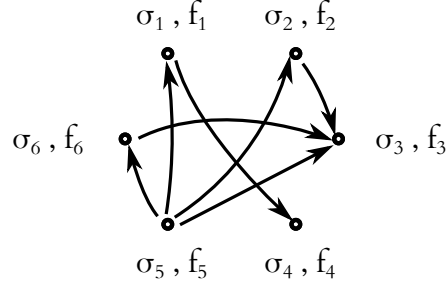
$$\sigma_i \in \{0, 1\} = \mathbb{B}$$

$$\Sigma = (\sigma_1, \sigma_2, \dots, \sigma_n)$$

Boolcu dinamik ağlarda, her  $\sigma_i$  düğümünün  $t \in \mathbb{Z}$  anındaki durumunu  $t + 1$  deki durumuna taşıyan Boolcu bir  $f_i : \mathbb{B}^n \rightarrow \mathbb{B}$  fonksiyonu bulunur. Bu fonksiyon ağın  $t$  anındaki  $(\sigma_1(t), \sigma_2(t), \dots, \sigma_n(t))$  durumundan,  $i$ . düğümün  $t + 1$  anındaki durumu belirler.

Dolayısı ile bir Boolcu ağdan bahsedildiğinde, bu ağı temsil eden  $\sigma_i$  düğümleri ve bu düğümlere atanan  $f_i$  fonksiyonlarından bahsedilmektedir. Bu tanımda önemli olan bir nokta, ağın çizgesinin bağlantılandırılışından hiç bahsedilmemesidir. Bunun nedeni, aslında Boolcu dinamik bir ağın, çizgesinin kenar kümesi  $V$  (Tanım A.1.1) deki bilginin  $f_i$  Boolcu fonksiyonlarının içerisinde bulunuyor olmasıdır. Eğer bir  $\sigma_i$  durumunun değeri  $f_j$  fonksiyonunun çıktısını  $\sigma_i$  hariç tüm düğümlerin durumundan bağımsız olarak değiştirmiyor ise,  $i$  den  $j$  ye bir kenar yok denilebilir. Yani, kenarın olmadığı durum ile  $f_j$  nin  $\sigma_i$ 'den “bağımsız” durum çizgenin dinamiği ile ayırd edilemez.

Bu çalışmada, çizgelerin topolojileri ve dinamik davranışları arasındaki istatistiksel ilişki incelendiğinden bahsi geçen Boolcu ağlar,  $\sigma_i$  düğüm durumları,  $f_i$  fonksiyonları ve  $A$  komşuluk matrisi ile betimlenecektir.

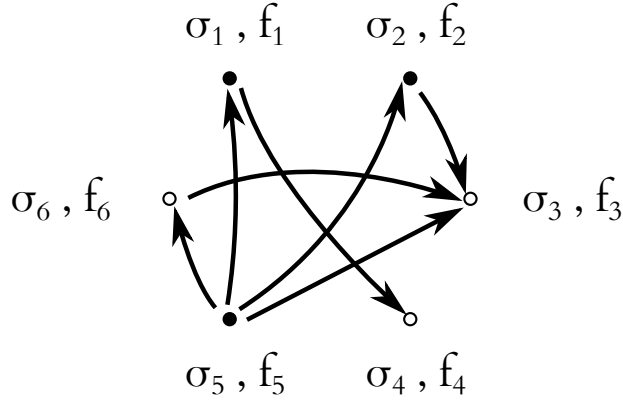


Şekil 2.1: Bir Boolcu ağ, düğümlerinin durumları  $\sigma_i$ , Boolcu fonksiyonları  $f_i$ , ve bu düğümleri bağlayan kenarları ile tanımlanır.

Bir örnek ile  $f_i$  fonksiyonunun dinamiği açıklanabilir.

**Örnek 2.1.1 (Başvuru Çizelgesi (Lookup Table) kullanılan Boolcu ağ)**

Şekil 2.1 deki ağ ele alınsın. Bu ağın  $t$  anındaki durumu  $\Sigma = (1, 1, 0, 0, 1, 0)$  olsun ve aşağıdaki gibi gösterilsin.



$f_3$	$\sigma_2$	$\sigma_5$	$\sigma_6$	$\sigma_3(t+1)$
	○	○	○	○
	○	○	●	●
	○	●	○	○
	○	●	●	●
	●	○	○	●
	●	○	●	○
	●	●	○	●
	●	●	●	○

Şekil 2.2: Başvuru çizelgesi (LUT) dinamiğine sahip bir Boolcu ağ. Burada  $\sigma_i = 0$  ve  $\sigma_i = 1$  durumları sırası ile içi boş ve içi dolu çemberler ile gösterilmiştir. Ağın altında ise,  $\sigma_3$  düğümünün  $(\sigma_2, \sigma_5, \sigma_6)$  nın  $t$  anında alabileceği tüm olası durumlara göre  $\sigma_3$ 'ün  $t + 1$  anında hangi değeri alacağı bir tablo yardımı ile gösterilmektedir.

Bu dinamik sistemde,  $\sigma_3(t+1)$  i bulmak için,  $\sigma_3$  'ün bağlı olduğu  $\sigma_2$ ,  $\sigma_5$  ve  $\sigma_6$  nın durumlarının alabileceği her konfigürasyon için  $\sigma_3$ ün bir sonraki iterasyonda alacağı değer önceden belirlenmiştir. Dolayısı ile verilen örnekte bu düğümler sırası ile 1, 1, 0 değerlerine sahip olduğundan  $\sigma_3(t+1) = 1$  olacaktır. Sonuç olarak, her bir düğüm kendi başvuru çizelgesine sahip olacak ve tüm ağıın durumu bu çizelgedeki değerlere göre zaman içerisinde değişecektir.

Başvuru çizelgeleri tüm olası durum kombinasyonlarını kapsayan çizelgeler olduğundan, olası tüm Boolcu fonksiyonları betimleyebilir. Mesela sadece bir düğümün değerini kullanan, diğerlerine “hassas” olmayan bir “kanallayıcı” (*canalizing*) bir fonksiyonu gerçekleyebilir. [4] Ya da bir çizelge başta her durum için rastgele bir ikili değer seçilerek yaratılabilir.

Lakin, bazı koşullarda başvuru çizelgeleri çok kullanışlı olmayabilir. Bu çizelgeler, Örnek 2.1.1 teki gibi bir düğüme bağlı olan düğüm sayısına göre hazırlanmış ise, o zaman çizgedeki herhangi bir değişim bu çizelgeyi kullanılamaz kılacaktır. Eğer çizgelerdeki  $f_i$  fonksiyonlarının olası bir topolojik değişiklikte kullanılabilir kalması isteniyorsa, her bir düğümün  $2^n$  tane konfigürasyona sahip bir çizelgeye sahip olması gerekecektir, bu da hesaplama kaynaklarında duruma göre büyük bir bedel olabilir.

Fakat bu tip kullanışsızlıklardan ziyade, bu durum biraz da modellenen sisteme bağlı bir seçimdir. Moleküler boyuttaki biyolojik sistemlerin, mesela bir protein sentezini gerçekleyen tüm sistemin, ona bağlı olan diğer sistemlerin durumuna olan hassaslığı, etrafında onu etkileyen bir çok faktöre bağlıdır. Diyelim ki bir  $A$  biyolojik objesinin (bu protein olabilir), bir başka sistemi tetiklemesi için gereken tek koşul, bir  $B$  objesinin üzerindeki kenetlenilebilecek  $\eta$  sayıdaki noktalardan birine kenetlenmesi olsun. Aynı zamanda bir  $C$  objesi de bulunsun, ve aynı noktaya kenetlenebilmesine rağmen aynı tetiklemeyi yaratmasın. Buradaki sistemde belli ki  $C_1, C_2, \dots, C_\eta$  proteinlerinin varlığı ancak ve ancak hepsi sistemde mevcut olduğunda, bu mevcudiyete 1 dersek, hepsi 1 olduğunda etkili olacaktır. Diğer tüm durumlarda dinamik,  $B$  proteininin durumuna bağlıdır.

Bir başka bir sistemde ise bir  $E$  proteinin sentezi  $A, B, C, D$  proteinlerinin sentezlenip sentezlenmemesinin karmaşık bir fonksiyonu olabilir. Zira, proteinlerin şekilsel karmaşıklığı bu tip bir sistemin varlığını da mümkün kılmaktadır. Proteinin birden fazla aktif bölgesi olabilir, bir aktif bölgesinin kullanılıyor ya da engelleniyor olması proteinin şeklini ve dolayısı ile işlevini değiştirir.

Bir diğer sistemde ise, bir sistemin cevabı toplamsal olabilir. Bu sefer örnek mekanik bir sistem olsun ve sistemdeki giriş işaretleri bir katı objenin üstten mi alttan mı belli bir kuvvetle itildiğini simgelesin. Bu objenin en sonunda hengî tarafa doğru

ivmeleneyeceği, tüm kuvvetlerin karmaşık bir Boolcu fonksiyonu olmaktan ziyade kuvvetlerin toplamına bağlıdır, yani 1 ler çoğunlukta ise 1, 0 lar çoğunlukta ise bu sistem 0 dönecektir.

Bu özel durumlar biyolojik sistemleri betimleyen Boolcu ağlar ile kabaca eşleştirilebilir. İlk verilen örnekte, aslında kanallayıcı bir dinamiğin bulunduğu söylenebilir, çünkü bir protinin sentezi, büyük oranda bir başka proteinin o ortamdaki varlığına ya yokluğuna bağlıdır. İkinci verilen örnekte, dinamik, bir başvuru çizelgesi ile betimlenebilir, çünkü ilgili protein etrafındaki konfigürasyona göre henüz nedenleri fiziken çözülmemiş bir davranış sergiliyordur. Üçüncü durumda ise, bir biyolojik süreci körükleyen ve onu engelleyen iki ayrı süreçten bahsediliyor olabilir. Bu süreci betimlemek için ise, körükleyen elemanlar ile engelleyen elemanların çoğunluğuna göre 1 ya da 0 durumunu alan bir düğüm bir model olarak sunulabilir.

### 2.1.1 Maske Dinamiği

Tüm bu sistemler ve kısıtlar göz önünde bulunarak niyete mahsus bir  $f_i : \mathbb{B}^n \rightarrow \mathbb{B}$  fonksiyonu taslağı düşünülmüştür.

#### Tanım 2.1.2 (Maske Fonksiyonu)

Durumju  $\Sigma = (\sigma_1, \dots, \sigma_n)$ , fonksiyonları  $f_i$  ve kenarları  $A$  komşuluk matrisi ile tanımlı olan bir Boolcu dinamik ağ ele alınsın.

Her  $\sigma_i$  düğümüne bu ağda bir  $k_i \in \mathbb{B}^n$  **anahtarı** ve basit bir Boolcu  $\star$  **işlemi** atansın. Maske fonksiyonunda,  $k_i$  anahtarını sadece  $i$ . düğüme bağlı olan düğümlerin simgelerinden oluşan bir  $\mathbf{k}_i^*$  anahtarına indirger, sadece  $i$ . düğüme bağlı olan düğümlerin durumlarına sahip bir  $\Sigma^*$  vektörü ile  $\mathbf{k}^*$  vektörüne eleman eleman  $\star$  işlemini uygular. Sonuçta  $i$ . düğüm için bir  $\mathbf{r}_i^*$  vektörü elde edilir.  $k_{ij}^*$ ,  $\mathbf{k}_i^*$  nin,  $r_{ij}^*$ ,  $\mathbf{r}_i^*$  nin,  $\Sigma_j^*$ ,  $\Sigma^*$  ın elemanları olmak üzere

$$r_{ij}^* = \Sigma_j^* \star k_{ij}^*$$

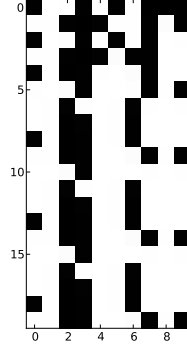
eşitliğinin yarattığı  $\mathbf{r}_i^*$  vektöründe sayıca fazla olan simge  $\sigma_i$  nin yeni değeri olur. Bu durum  $U(\cdot)$  Heaviside fonksiyonu,  $n_i^*$ ,  $r_i^*$  ın eleman sayısı olmak üzere,

$$\sigma_i(t+1) = U \left( \sum_j^{n_i^*} r_{ij}^* - n_i^* \right)$$

Bu dinamiğin bir önceki bölümde bahsedilen farklı durumlara özgü açıklanan özellikleri kabaca birleştirmesi, ama biyolojik sistemin bir özelliğini vurgulamaya yetenmemesi amaçlanmıştır. Maske fonksiyonu adını verdiğimiz bu fonksiyon,  $\Sigma(t)$

durumunu bir anahtar ile eleman eleman  $\star$  işleminden geçirdiği için  $\Sigma(t)$  durumuna bağıllığı değiştirilebilir, ama aynı zamanda bu fonksiyonun çıktısındaki çoğunluğu temsil eden simgeyi aldığı için de “toplamsal” olacaktır. Bu çalışmada  $\star$  işlemi  $XOR$  olarak seçilmiştir, çünkü giren simgeler aynı olasılıkta üretiliyor ise  $XOR$ , çıkan simgeleri de eş frekansta kılan bir işlemdir.

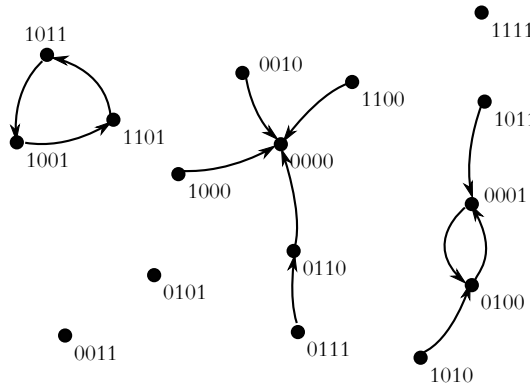
Bu tip bir fonksiyona sahip bir ağın durum uzayındaki ilerleyişi Şekil 2.3 de verilmiştir.



Şekil 2.3: Boolcu ağın durumunun zamanda evrimi. İlk satır,  $\Sigma(0)$  ilk durumunu, onu takip eden tüm satırlar ise, durumun  $t$  ye göre nasıl değiştiğini göstermektedir. Bu şekildeki ağ,  $n = 10$  düğümlü ve 5 li bir yörüngeye takılmıştır.

### 2.1.2 Boolcu ağların durum uzayı

Boolcu  $n$  düğümlü bir ağ,  $\sigma \in \{0, 1\}$  olduğundan  $(0, 0, \dots, 0)$  tan  $(1, 1, \dots, 1)$  e kadar tam  $2^n$  ayrı konfigürasyona sahip olabilir. Dolayısı ile  $\Sigma(0)$  konfigürasyonundan başlatılan bir ağ,  $n$  boyutlu bir hiperküpün köşelerinde dolaşarak  $\Sigma(0)$  dan başlayan bir yörünge çizer.



Şekil 2.4: Boolcu ağın faz uzayının bir temsili.

Şekil 2.4 de dört düğümlü bir ağın temsili durum uzayı çizilmiştir. Boolcu fonksiyonlar deterministik ve belleksiz olduğundan burada her bir durum, zaman içerisinde



aynı şekilde evrilmektedir.  $\Sigma(0) = (0111)$  ile başlayan bir ağ sırası ile  $\Sigma(1) = (0110)$  ve  $\Sigma(2) = (0000)$  durumlarını alır ve bundan sonra durum değiştirmez. Burada  $(0000)$  durumu bir kararlı denge noktasıdır.  $(0000)$  denge noktası,  $\{(0000), (0111), (0110), (1000)\}$  kümesindeki durumlar ile başlayan tüm durumlarda varılan son durum olduğundan, bu küme  $(0000)$  denge noktasının **çekici havzası** adını alır.

Aynı ağda, başka bir çekici havza ise, sonunda bir iki durumlu salınım varmaktadır. Buna benzer bir başka havzada ise üç durumlu bir salınım gözlenmektedir. Diğer tüm başlangıç durumları için ağ kendi durumunu korumaktadır.

Bir Boolcu Ağın durum uzayı, onun işlevinin basit bir özetini sunar. Denge durumları, ilgili sistemin belli bir iç dinamiğin sonucunda süregelen kıldığı, durumdaki değişimlere daha bağışık, ve zamanının çoğunu geçirdiği durum olacaktır. Ayrık olmayan bir sistem ile analogi kurulur ise, bu sistemin durumunu, bir bilyenin kasesinin çukurunda bulunduğu duruma benzetilebilir.

Bu çalışmada Boolcu dinamik sistemlerin faz uzayı, bir karmaşıklık ölçütü olarak kullanılacaktır. Bir gen regülasyon ağının dinamiğinin  $\Sigma(0)$  başlangıç durumuna bağlı farklı çekicilere sahip olması, işlevselliğini ilk koşulun ne olduğuna bağlı kılacaktır. Bu da birbiri ile tamamen aynı iki sistemin başlangıç koşulları farklı olduğundan farklı son durumlara sahip olmalarını, mesela farklı proteinler salgılamalarına neden olacaktır. Dolayısı ile bir sistemin sürdürülebilir bir işlevselliğe sahip olabilmesi için açıklanan tipte dinamiğe sahip olan ağların olabildiğince az sayıda olması gerektiği varsayımında bulunulmuştur.

## 2.2 Genetik Algoritma ile Boolcu ağların evrimi

### 2.2.1 Genetik Algoritma

Boolcu Ağların belli bir başarıım kriterine göre evrilmesi sağlanıp, bu başarıım kriterine göre başarılı saydığımız ağların topolojik özellikleri incelenecektir. Bunun için öncelikle Genetik Algoritmanın ne olduğuna değinmek doğru olur.

#### Tanım 2.2.1 (Genetik Algoritma)

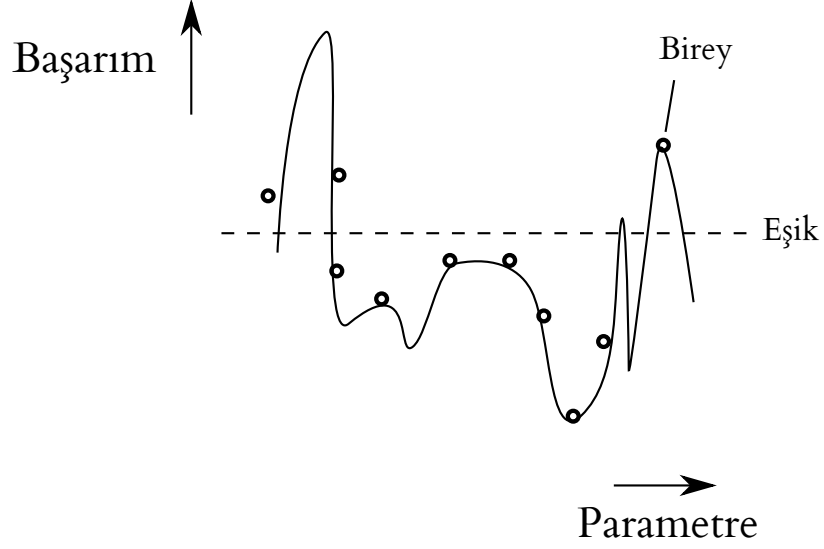
Genetik Algoritma (GA) , üzerinde bir  $M$  **mutasyon operasyonu** ve bir  $\phi$  **başarıım fonksiyonu** tanımlanmış objelerden oluşan bir topluluğa uygulanan, başarıım kriterine göre optimizasyon yapan bir algoritmadır. [1]

Bu algoritmada, bu objelerden  $N$  tanesinin oluşturduğu bir topluluk yaratılır, bu toplulukta belli bir başarıımdan fazlasını sergileyen  $m$  tane eleman kopyalanır ve kopyaları mutasyon operasyonuna tabii tutulur. Yeni yaratılan objeler topluluğa yeniden kazandırılır, ve nüfusu  $N + m$ 'e çıkan topluluktan rastgele bireyler eksilir.

Genetik Algoritma, tahmin edilebileceği gibi, doğadaki genetik kodun evrimini taklit eden bir buluşsal (*heuristic*) bir optimizasyon yöntemidir.

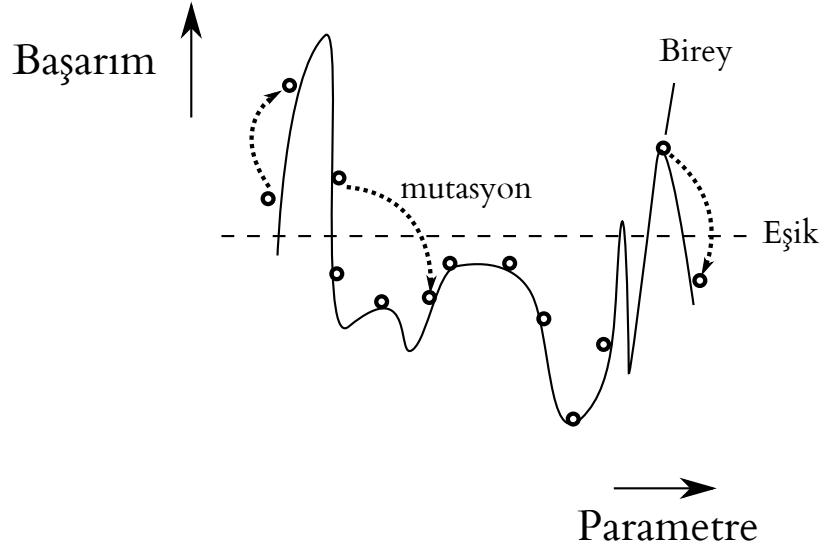
### Örnek 2.2.1 (Genetik Algoritma)

Bu yöntemi netleştirmek adına bir örnek verilebilir. Genetik algoritmada, bu örnekte bir boyutlu bir parametreler uzayı olsun. Bu parametrenin farklı değerleri bireyleri temsil etmekte, ve bu bireylerin başarımı da parametreye bağlı bir fonksiyon oluşturur. Şekil 2.5 de bu durum temsil edilmektedir.



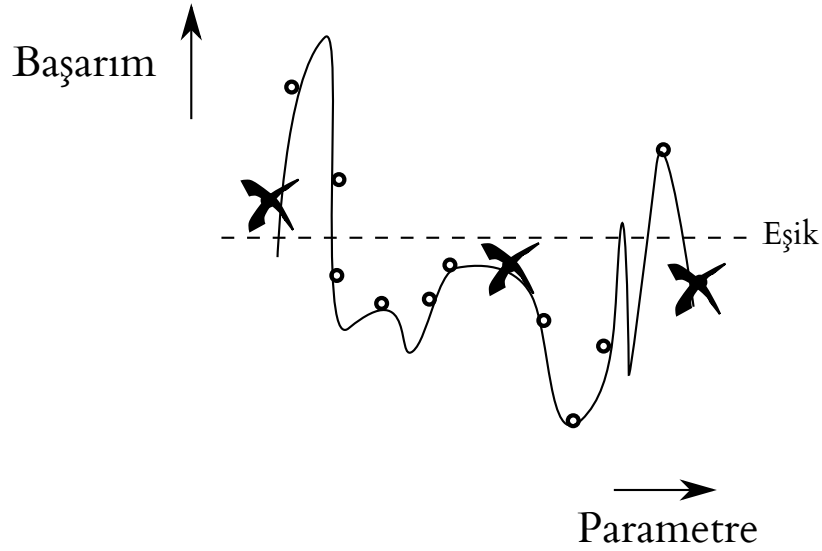
Şekil 2.5: Başarım fonksiyonu, topluluktaki bireyler, ve genetik algoritmanın başarımlar eşik değeri

Burada topluluktaki bireyler noktalar ile temsil edilmekte ve her birey x eksenindeki parametrelere ve y eksenindeki başarımlara sahiptirler. Dolayısı ile bu eksenlere sahip bir  $y = f(x)$  başarı fonksiyonundan bahsedilebilir. Bu fonksiyonun tüm noktadaki değeri bilinmemektedir, zaten aranılan da optimum başarıma sahip birey olduğundan, fonksiyon biliniyor olsaydı, çözüm de biliniyor olurdu, ama burada anlatım rahatlığı uğruna bu fonksiyon çizilmiştir. Şekil 2.5 de görüldüğü gibi, topluluktaki bireyler belli bir eşik değerinin altında ve üstünde başarımlara sahiptir, fakat hiçbirisi en iyi başarımları göstermemektedir.



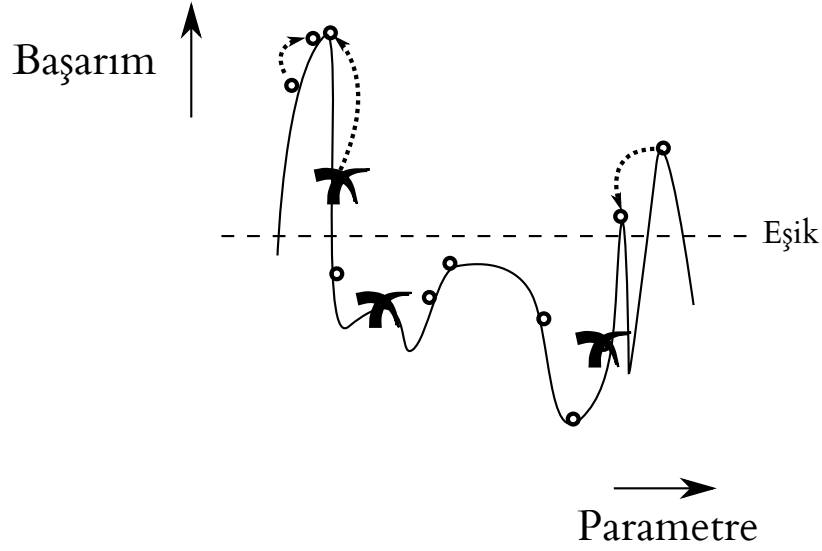
Şekil 2.6: Genetik algoritmanın kopyalama ve mutasyon safhası

Şekil 2.6 de gösterilmiş olan genetik algoritmanın ikinci safhasında, eşik değerinin üzerinde kalan bireyler başarılı sayıldığından kopyalanıp mutasyona uğratılmıştır. Mutasyona uğramış kopyalar ebeveynlerinden daha iyi ya da daha kötü başarı sergileyebilirler.



Şekil 2.7: Genetik algoritmanın eleme safhası

Bir diğer önemli safha da topluluğun bir elemeye tabi tutulduğu safhadır. Şekil 2.7 te görüldüğü gibi, bu safhada bireyler rastgele yok edilirler. Görüldüğü gibi kıstasa göre başarılı sayılan bireyler de yok olabilir, lakin bu safhanın rastgele olması sonradan global maksimuma evrilebilecek başarısız elemanlar ile lokal maksimuma takılmış görece başarılı elemanları ayırmadığından eniyilemeyi daha etraflı yapacaktır.



Şekil 2.8: Boolcu ağın faz uzayının bir temsili.

Şekil 2.8 de tüm safhalar birlikte gösterilmiştir, görüldüğü gibi, genetik algoritmanın sonucunda bazı bireyler en iyi başarıma oldukça yaklaşmıştır. Popülasyonun bu döngüdeki makroskopik parametreleri, mesela ortalama başarım, belli bir dengeye oturduktan sonra popülasyondaki başarılı bireyler incelenebilir.

GA'da topluluktaki elemanlar, mutasyonun tanımlı olduğu belli bir parametre seti ile tanımlı olan bir parametre uzayında, bir nokta olarak temsil edilebilirler. Genetik algoritmanın uygulanması söz konusu ise cevaplanması gereken sorulardan bir tanesi bu parametrelerin ne olacağıdır. Doğadaki canlılarda bu parametreler, DNA diziliminin kendisidir, Boolcu ağlarda da çalışmada odak noktası olan bir özelliğin seçilmesi gerekmektedir.

Boolcu ağlarda ağların bağlantılandırılışı  $A$  komşuluk matrisi ile,  $f_i$  Boolcu fonksiyonu  $k_i$  anahtarları ile değişime tutulabilir objelerdir. İlk safhada bağlantıların mutasyona uğratılması tercih edilmiştir, zira Boolcu ağın dinamiğinin çizgenin topolojisine bağlılığı ön plandadır.

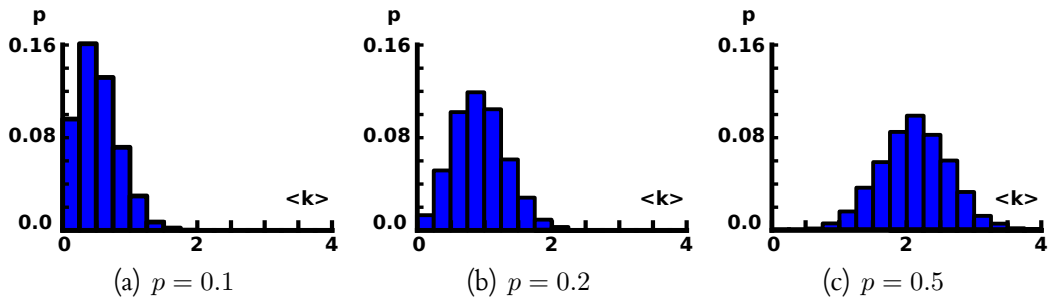
## Bölüm 3

### Boolcu Ağların Evrimi

Bu bölümde, anlatılan ağların Genetik Algoritmaya tabi tutuluşlarının detayları ele alınacaktır. Gen regülasyonunu temsil edecek ağlardan oluşan bir topluluk yaratılacak ve bu topluluğun evrimi üzerinden istatistiksel çıkarımlarda bulunulacaktır. Çalışmada ağlar, her bir düğüm çifti  $p$  olasılıkla bağlanarak, ve anahtarındaki her bir değeri  $p_k = 0.5$  olasılıkla 1 yapan bir "yaratıcı fonksiyon" ile yaratılmaktadır.

#### 3.1 Üretilen Boolcu Ağların Özellikleri

Üretilen ağlar ile ilgili bir fikir edinmek için, yaratılan topluluklar incelenmiştir. Öncelikle yukarıdaki yaratıcı fonksiyon ile yaratılan topluluğu kontrol etmek amacıyla farklı  $p$  değerleri için  $n = 4$  olan ağların derece dağılımına bakıldı.



Şekil 3.1: Çeşitli  $p$  bağlandırma olasılıkları için ağlardaki derece dağılımı

Şekil 3.1 de kontrol amaçlı  $N = 10000$  tane  $n = 4$  lük ağlara sahip bir topluluk yaratılmış, ve bu topluluktaki tüm ağların tüm düğümleri üzerinden düğüm derecelerinin beklenen değeri hesaplatılmıştır. Şekilde de görüldüğü gibi, topluluklarda derece dağılımları  $n = 4$  için olan  $p = 0.1$ ,  $p = 0.2$ ,  $p = 0.5$  binom dağılımlarını vermektedir. Buradan rastgele sayı yaratıcısından alınan olasılıklar ile yaratılan ağların bağlantılarının istatistiksel varsayım ile çelişmeyecek şekilde üretildiği görülmektedir.

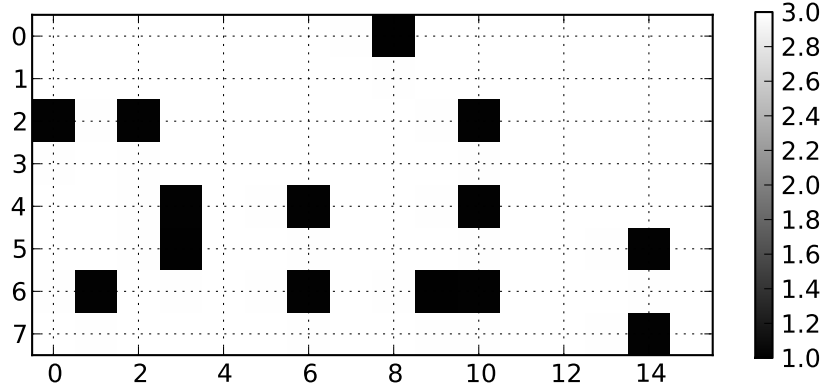
Topluluktaki bireylerin ilgilenilen bağıntılarından bir tanesi, ortalama derecesi  $\langle k \rangle$  nın onun dinamiğine etkisidir. Bunu makroskopik olarak incelemek için bir fonksiyon tanımlanmıştır.

### Tanım 3.1.1

Bir Boolcu ağın tüm olası  $\Sigma(0)$  başlangıç durumlarında denge koşuluna ulaştığı anki yörünge uzunluklarını (eğer kararlı denge noktası ise 1, iki durum arasında gidip geliyor ise 2 gibi) ele alalım.  $G(\Sigma_i)$ ,  $\Sigma_i$ 'den başlayan çözümün sonunda ulaştığı denge durumunun yörünge uzunluğu olsun. Uzunlukların tüm  $\Sigma$  değerleri üzerinden toplamını aşağıdaki gibi tanımlayabiliriz.

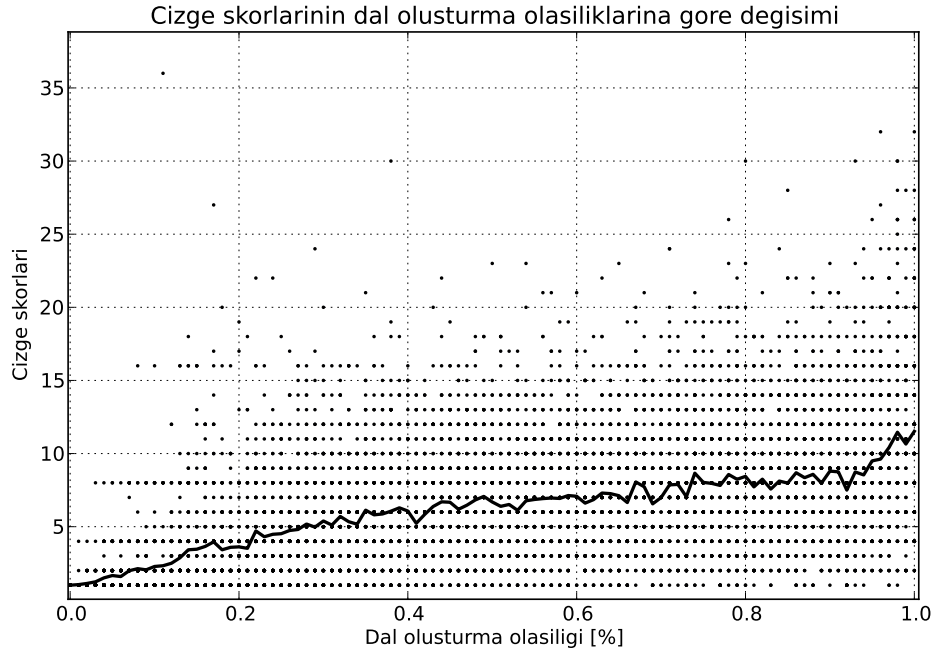
$$G = \sum_{\Sigma} g(\Sigma_i) , \Sigma_i \in \mathbb{B}^n$$

Burada dikkate değer bir nokta, bu değer yörünge uzunluklarına göre artması, ama aynı zamanda iki ayrı denge noktasına sahip dinamiklerde de çekim havzalarının büyüklüğüne de orantılı olmasıdır. Yani bu değer bir genetik algorithma bir kıstas olarak kullanılırsa, genetik algoritma hem yörünge uzunluklarına hem de ağın durum uzayında hangi çekicinin baskın olduğuna hassas olacaktır.



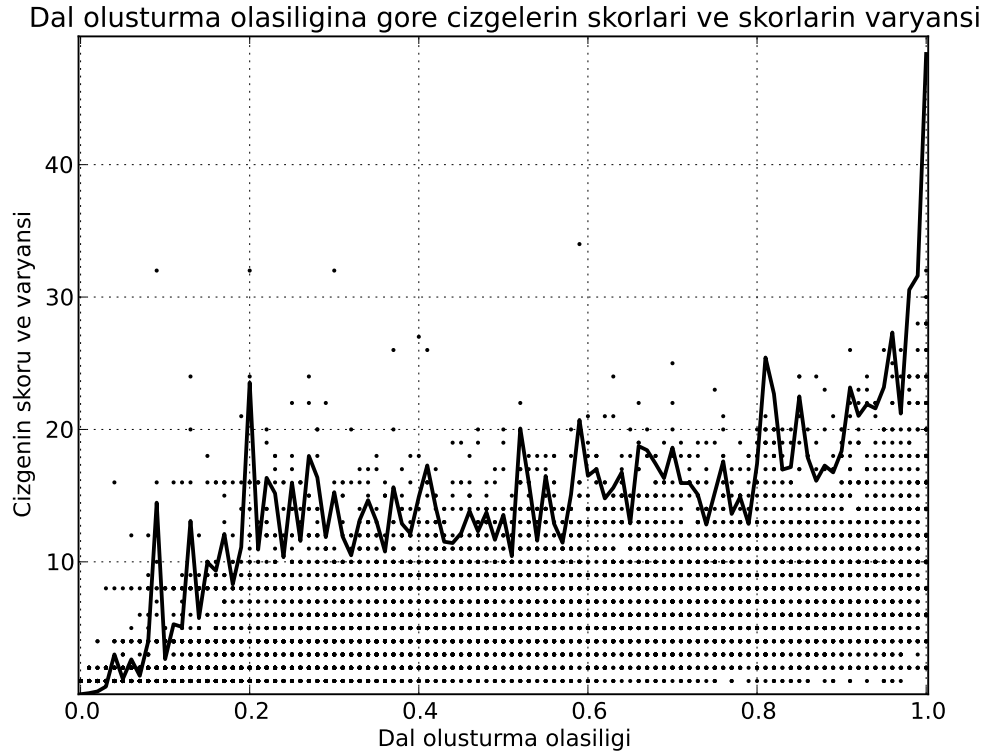
Şekil 3.2: Bir ağın durum başlangıç durumuna göre çözümün kararlı durumdaki yörüngesinin uzunluğu. Buradaki grafikte ilk durumlar sol üst köşede  $\Sigma(0) = (0000000)$  den başlayarak sağ alt köşede  $\Sigma(0) = (1111111)$  ile bitecek şekilde konumlandırılmıştır. Siyah noktalar sonunda bir kararlı çekiciye giden, beyaz noktalar ise sonunda bir üçlü yörüngede dengeye varan ilk durumlardır.

Bu tanımlı şimdilik yaratılan ağın dinamiğinin karmaşıklığının nicel bir göstergesi olarak kullanılacaktır. Zira durum uzayında uzun dengeli yörüngeleri bulunan bir ağ kaotik, daha az sayıda ve daha kısa dengeli yörüngelere ya da çekici noktalara sahip ağları da dengeli olarak ele alacağız.



Şekil 3.3: Bağlama olasılığı  $p \in [0, 1]$  a göre  $G$  ve  $\langle G \rangle$  değerleri. Burada noktalar ağların aldığı  $G$  değerleri, düz çizgi ise bu değerlerin ortalamasıdır. Bazı ağlar aynı  $G$  değerini almaktadır, dolayısı ile her nokta birden fazla ağa denk düşebilmektedir.

Elde edilen toplulukların  $p$  iki düğümü bağlantılandırma olasılığı ile dinamiğin karmaşıklığı tanımlanan  $G$  fonksiyonu ile irdelenebilir. Bu bağıntıyı gözlemlemek için  $p \in [0, 1]$  aralığında 100 tane eşit aralıklı bağlantılandırma olasılığı değerinin her biri için ayrı,  $N = 50$  bireyli  $n = 7$  düğümlü topluluklar yaratılmıştır. Bu toplulukların her birinde  $i$ . bireyin  $G_i$  fonksiyonu ve o topluluk içindeki  $\langle G \rangle$  beklenen değeri hesaplanmıştır. Sonuçlar Şekil 3.3 ve Şekil 3.4 de gösterilmiştir. Şekil 3.3 de çizge skoru olarak gösterilen  $G$  fonksiyonu değerleri ve  $\langle G \rangle$  beklenen değerlerinde  $p$  olasılığına göre bir artış gözlenmektedir. Bu da daha sık bağlı düğümlere sahip ağların daha kaotik davranışlar sergileyeceği hakkında bazı ipuçları vermektedir. Aynı zamanda,  $p$  arttıkça her ne kadar düşük  $G$  değerli ağların sayısında bir azalma gözlenirse de, topluluğun  $p$  arttıkça daha geniş bir aralıkta  $G$  değerlerine sahip olduğu da Şekil 3.4 de gözlenebilir. Düşük başarımlı bireyler  $p = 1$  de bile görülmektedir, bunun nedeninin rastgele anahtarlara sahip bireylerde tüm düğümler bağlı olsa da bir düğümün diğerine olan bağımlılığının yine de anahtar tarafından dikte edildiği, dolayısı ile düğümlerin bazı olası  $f_i$  fonksiyonları için yine de birbirinden yeterince bağımsız olması olduğu söylenebilir.



Şekil 3.4: Bağlama olasılığı  $p \in [0, 1]$  a göre  $G$  ve  $\sigma_G$  standard sapma değerleri. Burada noktalar ağların aldığı  $G$  değerleri, düz çizgi ise bu değerlerin standard sapmasıdır. Bazı ağlar aynı  $G$  değerini almaktadır, dolayısı ile her nokta birden fazla ağa denk düşebilmektedir.

### 3.2 Genetik Algoritmanın Öğeleri

Şu ana kadar gen regülasyon ağlarının temsil ettiği belirtilen Boolcu ağlar tanıtılmıştır. Burada bu Boolcu ağları tabii tutulduğu genetik algoritmanın iç mekanizması, ve bu mekanizmadaki yapılmış seçimler açıklanacaktır. Ağların biyolojik işlevselliğini kendi durumu ( $\Sigma$ ) ne olursa olsun, ve bu işlevini dış etkenlerdeki küçük değişiklikler ile kaybetmemesi başarı fonksiyonunun tanımında esas alınmıştır. Bu fonksiyonlar durum uzayının basitliğini ödüllendirecek şekilde seçilmiştir. Tanım 3.1.1 da tanımlanan  $G$  fonksiyonu, bir başarı fonksiyonu olarak seçilmiş ve genetik algoritma,  $G$  fonksiyonu belli bir değerden düşük olan ağları üretecek şekilde hazırlanmıştır. Dolayısı ile sadece bir çekicisi olan, ve çekim havzası tüm durum uzayı olan bir ağ olabilecek en başarılı ağ, tüm durum uzayını dolaşan bir yörüngeye sahip bir ağ ise en başarısız ağ olmuştur. Bu tip bir başarı kriterine sahip bir genetik algoritma durum uzayında sadece kararlı denge noktalarına sahip ağları tercih edecek, ikili, üçlü yörüngelere sahip ağları azaltmaya çalışacaktır.

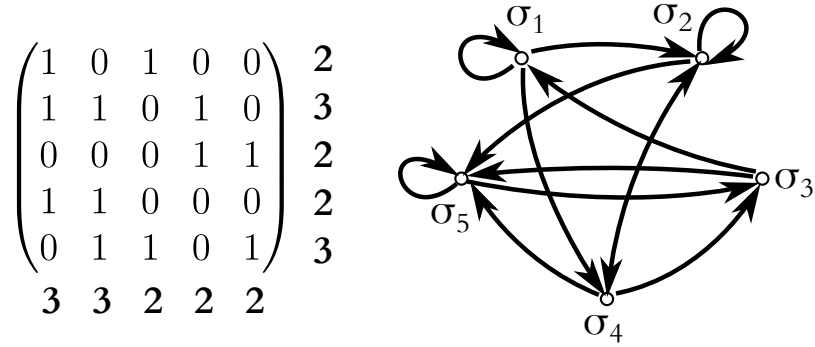
Bu tip bir başarı seçmek, yanında bazı aşık elementlerin popülasyonda ön plana çıkmasına neden olacaktır. Bu aşık çözümleri elemek için genetik algoritmada



uygulamaya özgü seçimler yapılmıştır. Mesela, 7 tane düğümü olan, ama hiçbir bağlantısı olmayan ağ, başlangıç koşulu ne olursa olsun o durumunu koruyacaktır, zira durumunu değiştirecek fonksiyona eski durumun bilgisi gitmemektedir. Dolayısı ile hiçbir bağlantısı olmayan ağlar popülasyonda ciddi bir kesimi oluşturmaya başlayabilir. Şekil 3.3 de görülebileceği gibi bu çizgeler  $p = 0$  durumuna denk düşmektedir ve tüm olası  $p$  değerleri arasında en düşük  $G$  değerine sahip grubu (tüm ağlar için  $G = 1$  dir.) oluşturmaktadır. Bunun yanı sıra, bağlantıları bulunan, ama aslında iki ayrı alt çizgeye ayrılabilen ağlar da ilgimizin dışında kalmaktadır. Çünkü bu tip ağlar birbirinden bağımsız farklı sistemler oluşturmakta, bir alt çizgedeki durum diğerini etkilememektedir. Dolayısı ile yaratıcı fonksiyonu sadece tamamiyle bağlı çizgeler oluşturacak şekilde seçilmiştir. Bu, mutasyonlar sonucu ile çizgenin koparılamayacağı, ya da bağlantısız bir çizge haline getirilemeyeceği anlamına gelmez.  $G$  fonksiyonunu en azlamaya çalışan bir algoritma, Şekil 3.3’de bu fonksiyonu en azlayacak, dolaylı olarak  $p$  parametresi daha düşük olan ağları tercih edecek, ve sonuç olarak bireylerden  $\langle k \rangle$  ortalama derecesi en az olanlarının yaşamasına olanak tanıyacaktır. Bu durum da ağların kopmasına neden olacaktır.

### 3.2.1 Dereceyi Koruyan Mutasyon

Bu istenmeyen fenotiplerden arınmış bir topluluk üzerinde çalışmak tercih edildiğinden dereceyi koruyan bir mutasyon üzerinde çalışıldı. Bir çizgenin derecelerinin korunması için  $A$  komşuluk matrisinin satırlarının ve sütunlarının toplamlarından oluşan vektörün mutasyon öncesinde ve sonrasında aynı kalması gerekmektedir.



Şekil 3.5: Dereceyi Koruyan Mutasyon'a hazırlanan çizge

Komşuluk matrisinde yukarıda bahsi geçen koşulu sağlayan bir mutasyon tanımlanmıştır. Burada öncelikle Şekil 3.5 de gösterilen çizgeye Dereceyi koruyan mutasyon uygulansın. Öncelikle Şekil 3.6 deki gibi rastgele bir matris elemanı seçilir.

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{pmatrix} \begin{matrix} 2 \\ 3 \\ 2 \\ 2 \\ 3 \end{matrix}$$

$$\begin{matrix} 3 & 3 & 2 & 2 & 2 \end{matrix}$$

Şekil 3.6: Dereceyi koruyan mutasyonun ilk safhası

Bu eleman seçildikten sonra, seçilen elemanın satırındaki ve sütunundaki seçilen eleman olmayan elemanlar saptanır. Bu saptanan elemanlardan birer tanesi Şekil 3.7deki gibi rastgele seçilir.

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{pmatrix} \begin{matrix} 2 \\ 3 \\ 2 \\ 2 \\ 3 \end{matrix}$$

$$\begin{matrix} 3 & 3 & 2 & 2 & 2 \end{matrix}$$

Şekil 3.7: Dereceyi koruyan mutasyonun ikinci safhası

Bundan sonra seçilen üç tane matris elemanı sırası ile  $a_{ij}$ ,  $a_{ik}$ ,  $a_{lj}$  ise,  $a_{lk}$  elemanına bakılır. Bu eleman  $a_{ij}$  ile aynı değeri alıyor ise seçilen satır ve sütunların aşağıdaki iki matrisi oluşturacak şekilde seçilmişlerdir.

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{pmatrix} \begin{matrix} 2 \\ 3 \\ 2 \\ 2 \\ 3 \end{matrix}$$

$$\begin{matrix} 3 & 3 & 2 & 2 & 2 \end{matrix}$$

Şekil 3.8: Dereceyi koruyan mutasyonun üçüncü safhası

dolayısı ile bu elemanların hepsi  $1 \mapsto 0$  ve  $0 \mapsto 1$  şeklinde değiştirilirse ağ derecesi değişmeden mutasyona uğratılmış olur. Çünkü bu operatör,  $A$  matrisinin satırları ya da sütunları toplandığı zaman oluşturulan vektörlerdeki her bir elemanı sabit tutmaktadır. Sonucu Şekil 3.9 te görünmektedir.

$$\begin{pmatrix}
 1 & 0 & 1 & 0 & 0 \\
 0 & 1 & 1 & 1 & 0 \\
 0 & 0 & 0 & 1 & 1 \\
 1 & 1 & 0 & 0 & 0 \\
 1 & 1 & 0 & 0 & 1
 \end{pmatrix}
 \begin{matrix}
 2 \\
 3 \\
 2 \\
 2 \\
 3
 \end{matrix}$$

$\begin{matrix} 3 & 3 & 2 & 2 & 2 \end{matrix}$

Şekil 3.9: Dereceyi koruyan mutasyon uygulandıktan sonra

Eğer yapılan her mutasyon, Bölüm B.2 deki tek parçalılığı kontrol eden algoritmaya göre kabul edilir ya da edilmez ise, mutasyon hem dereceyi hem tekparçalılığı koruyacaktır. Dereceyi koruyan mutasyona  $M_k$ , Hem dereceyi hem tek parçalılığı koruyan mutasyona ise  $M_{kc}$  denecektir.

## Bölüm 4

### Sonuçlar ve Tartışma

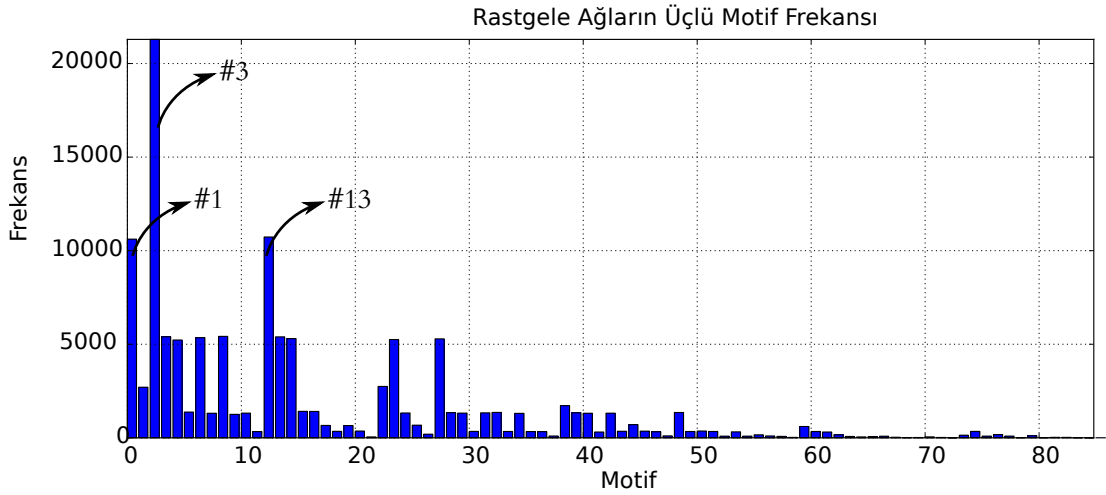
Önceki bölümlerde açıklanan yöntemler ve motivasyon ile, gen regülayon ağlarını betimleyecek  $n = 7$  düğümlü,  $N = 200$  tane rastgele Boolcu ağ bağlantılandırma olasılığı  $p = 0.2$  olacak şekilde yaratılmıştır. Tüm ağlar Tanım 2.1.2 de açıklanan Maske Boolcu fonksiyonu ile yaratılmış,  $k_i$  anahtarları da rastgele 1 ve 0 lar ile yaratılmıştır. Bu Boolcu ağların başarı kriteri Tanım 3.1.1 de açıklanan  $G^1$  fonksiyonudur. Bu yaratılan topluluk, iki ayrı mutasyon ile genetik algoritmaya tabii tutulmuş ve sonuçlar elde edilmiştir. Burada bahsedilen mutasyonlar Bölüm 3.2.1 de açıklanan  $M_k$  ve  $M_{kc}$  derece koruyan ile derece ve tek parçalılığı koruyan mutasyonlardır.

#### 4.1 Rastgele Yaratılan Ağların Motif Frekansları

Eldeki topluluğun  $G$  başarı fonksiyonuna göre adapte olurken geçirdiği evrimi karakterize edebilmek adına bir referans noktası elde etmek için rastgele yaratılan ve hiç evrim geçirmemiş bir topluluktaki frekansları elde edelim.

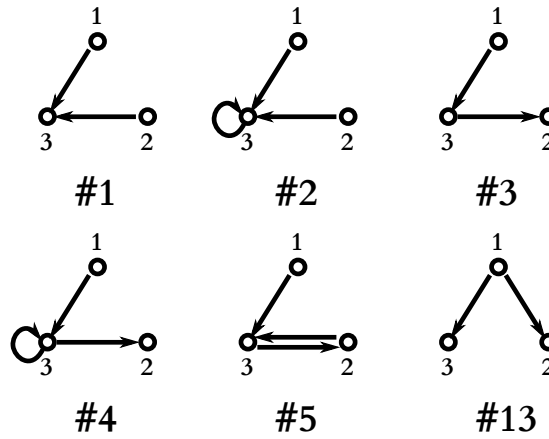
---

<sup>1</sup>Tüm olası başlangıç durumuna göre ağın sonunda ulaştığı yörüngenin uzunluklarını toplayan fonksiyon



Şekil 4.1: Üçlü motiflerin frekansları

yukarıda motifler numaralandırılmıştır. Bu listenin nasıl yaratıldığı Bölüm B.3.1 de anlatılmıştır. Bu yöntem ile bulunan motiflerden en ön plana çıkanları aşağıdakilerdir.



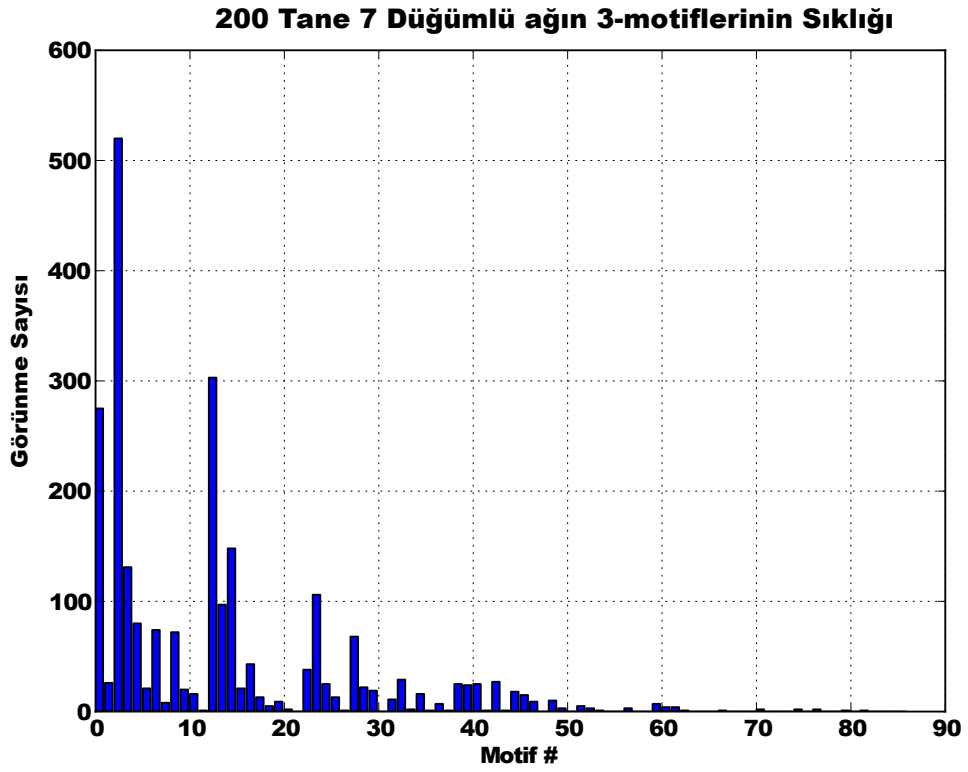
Şekil 4.2: Üçlü motiflerden 1, 2, 3, 4, 5 ve 13 numaralı motifler.

Şekil 4.2 de görüldüğü gibi dal sayısı arttıkça rastgele ağlarda bir motifi bulmak gittikçe zorlaşmaktadır. Bu her bir dalın oluşturulma olasılığının sabit oluşunun doğal bir sonucudur.

## 4.2 Derece Koruyan Mutasyonun Topluluktaki Motif Sıklığına Etkisi

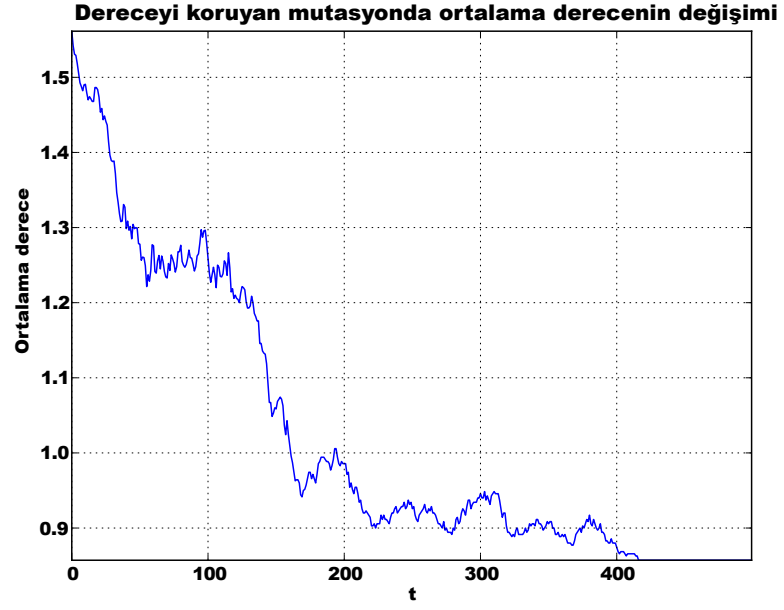
200 nüfuslu topluluk genetik algoritmaya tabi tutulmadığında motif sıklığı Şekil ??eki gibidir ve Şekil 4.1 deki rastgele yaratılmış ağların sıklıklarına benzemektedir, tek farkı daha az sayıda eleman ile elde edilmiş bir verinin söz konusu olmasıdır.

Burada  $t$  Boolcu sistemlerin dinamiğinde kullanıldığından, Genetik Algoritmanın her bir iterasyonu  $\tau$  değişkeni bir artırılacak gösterilecektir.

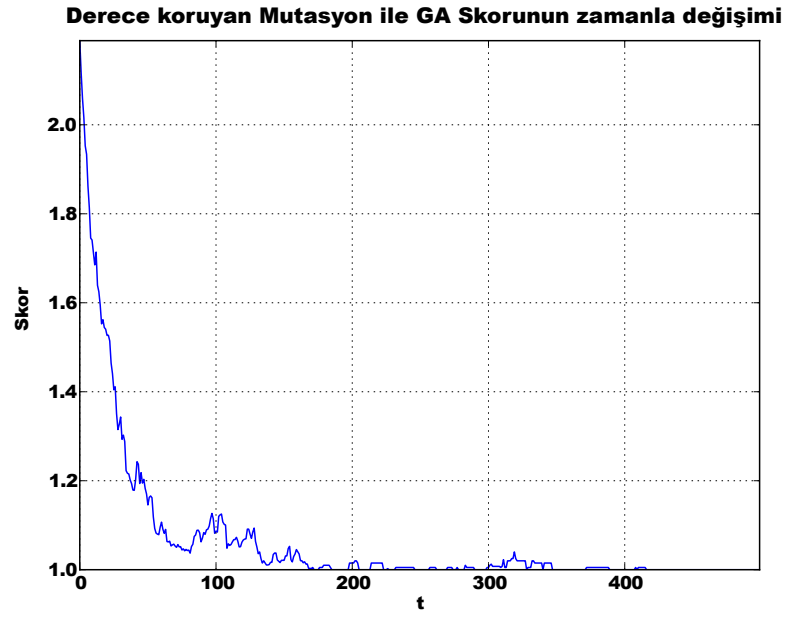


Şekil 4.3:  $\tau = 0$  anında  $M_k$  mutasyonunun uygulandığı genetik algoritmadaki Motif sıklıkları

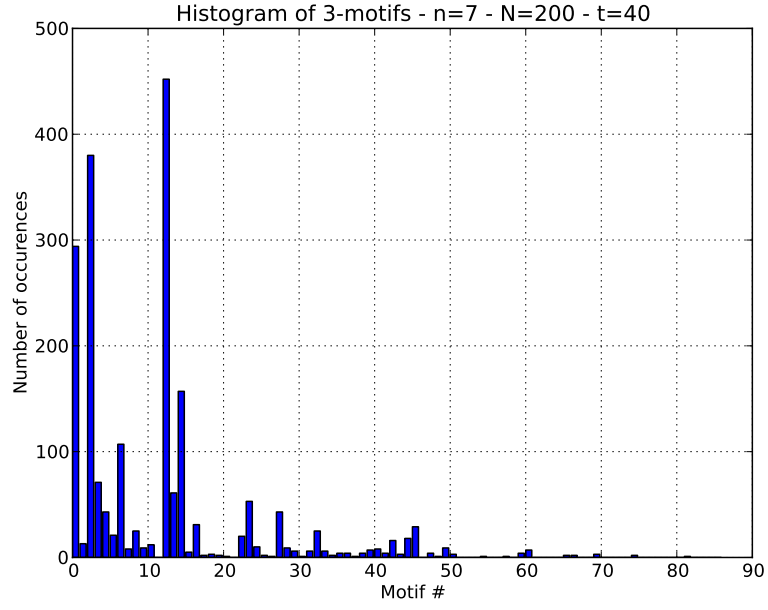
Genetik algoritma koşulduğunda topluluktaki istatistiksel parametrelerin değişimi aşağıdaki gibidir.



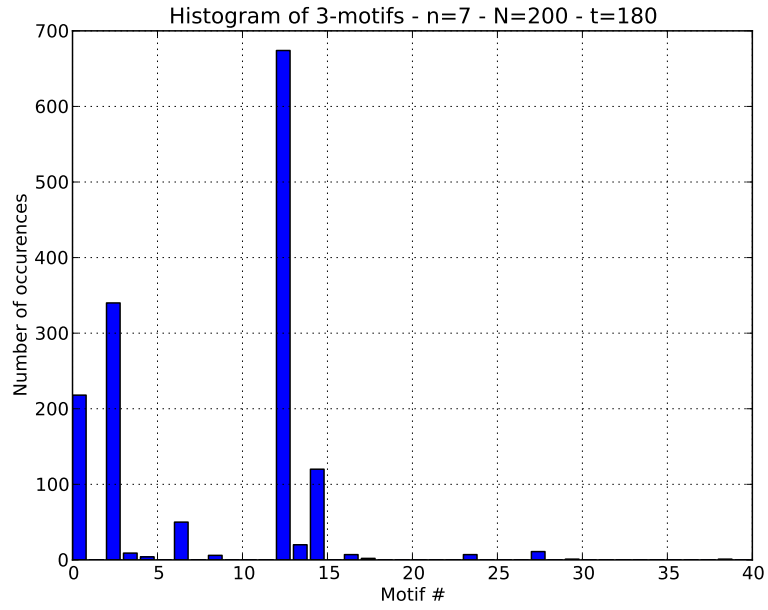
Şekil 4.4:  $M_k$  ile Genetik Algoritmaya tabii tutulan topluluğun  $\langle k \rangle$  tüm düğümler üzerinden derecelerinin ortalamasının zamana göre değişimi



Şekil 4.5:  $M_k$  ile Genetik Algoritmaya tabii tutulan topluluktaki  $\langle G \rangle$  ortalama başarımının zamana göre değişimi. GA'da düşük değerler tercih edilmiştir.

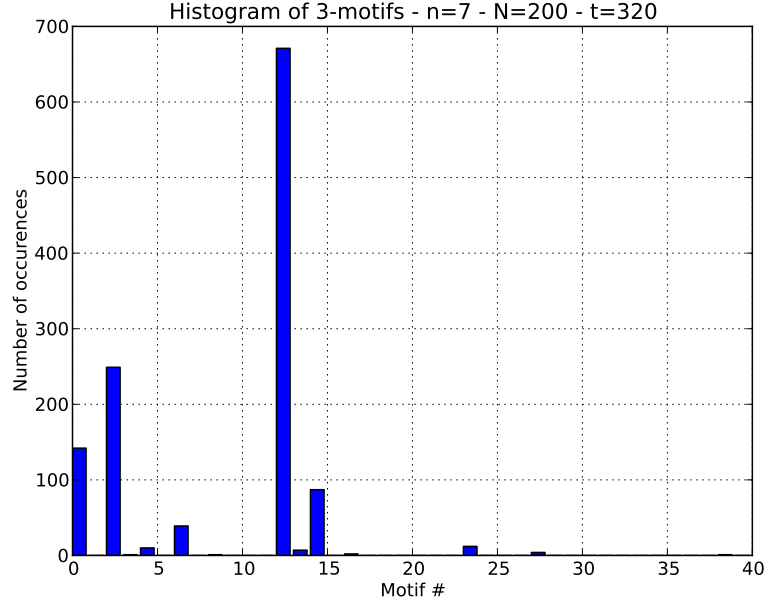


Şekil 4.6:  $\tau = 40$  anında  $M_k$  mutasyonunun uygulandığı genetik alitmadaki Motif sıklıkları

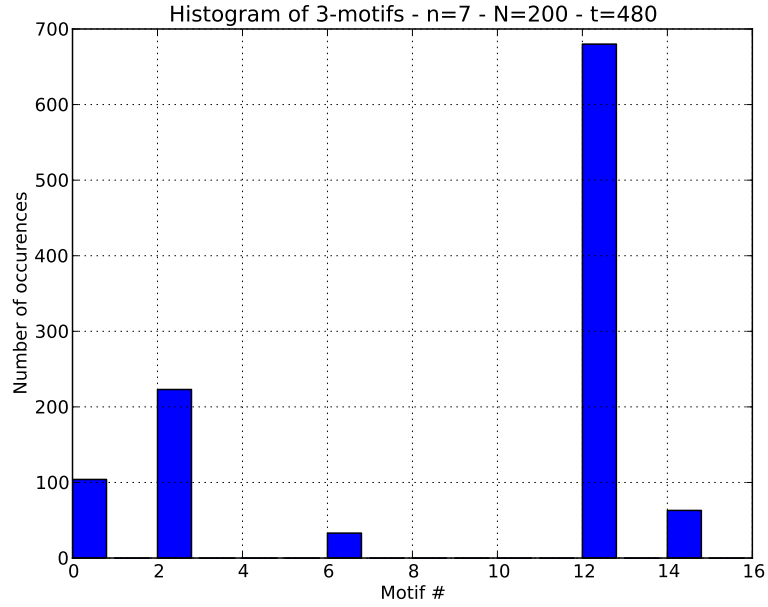


Şekil 4.7:  $\tau = 180$  anında  $M_k$  mutasyonunun uygulandığı genetik alitmadaki Motif sıklıkları





Şekil 4.8:  $\tau = 320$  anında  $M_k$  mutasyonunun uygulandığı genetik algorithmadaki Motif sıklıkları

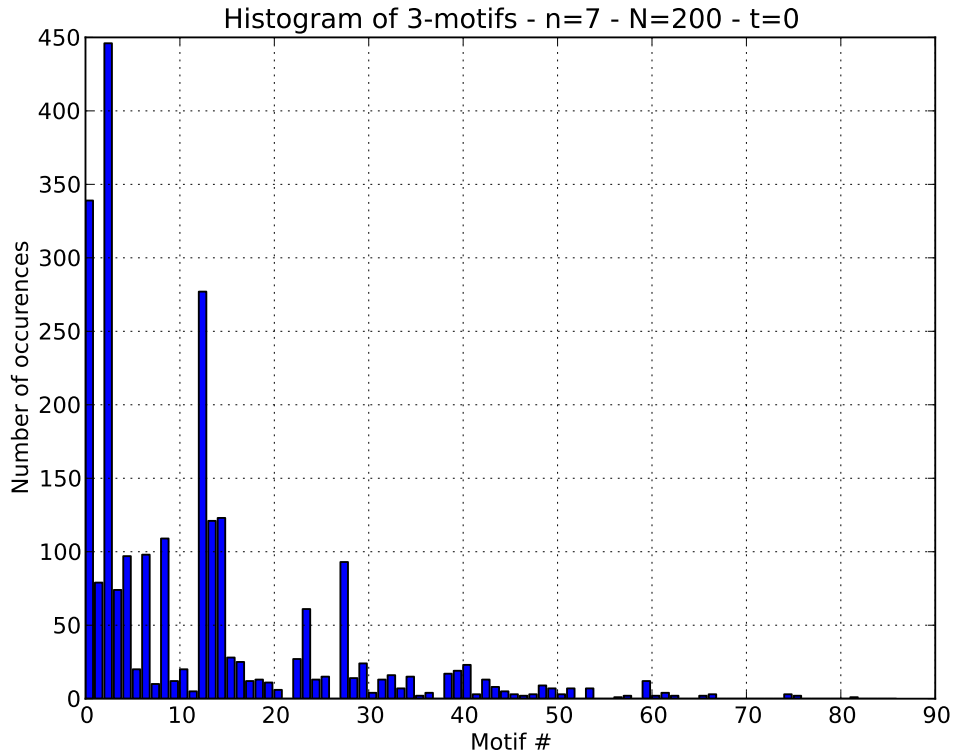


Şekil 4.9:  $\tau = 480$  anında  $M_k$  mutasyonunun uygulandığı genetik algorithmadaki Motif sıklıkları

Elde edilen veriden yapılabilecek çıkarımlardan ilki, Genetik Algorithma  $G$  fonksiyonunun değeri düştükçe,  $\langle k \rangle$  için olan değer de ona paralellik göstererek düşmektedir. Aynı zamanda histogramlardan da görülebileceği gibi sadece bazı motifler ön plana çıkmakta, bunlar da en düşük bağlantı sayısına sahip olanlar, yani 1., 3. ve 13. motifler olmuştur.

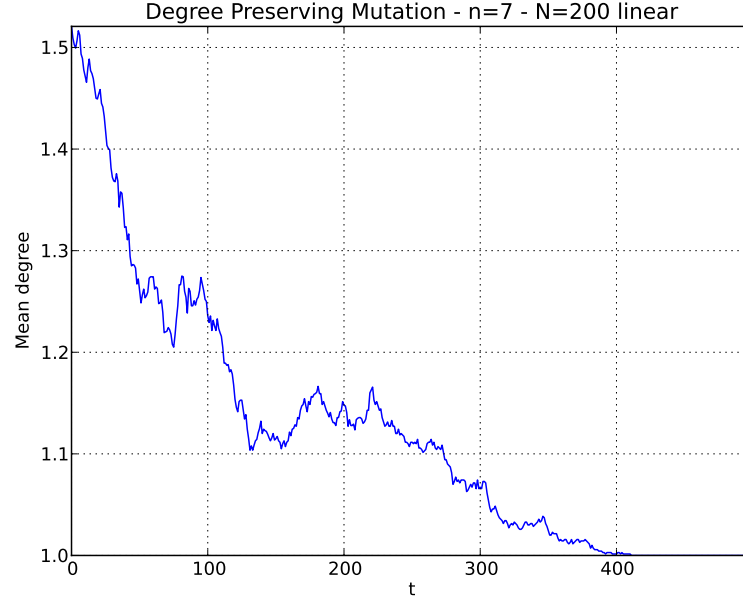
Öncelikle yaratılan ilk topluluktaki bağlantı sayısı, ya da düğümlerinin derecesinin ortalaması en düşük olan bireyler ön plana çıkmıştır, aynı zamanda bu bireyleri  $M_k$  mutasyonuna uğrattırırken derecelerinin artmayacağı da kesinleştirilmiştir. Dolayısı ile zamanla ( $\tau$ ) ilk topluluktaki en iyi başarıyı gösteren bir grubun ya da bir bireyin  $\langle k \rangle$  ına yaklaşmış ve hiç değişmemiştir. Burada, bir süre sonra olası tüm  $M_k$  operasyonları  $\langle k \rangle$  değerleri ve  $G$  değerleri aynı genotipleri döndürmektedir. Yani  $M_k$  operatörü, herhangi bir elemanı parametre uzayında belli bir noktalar kümesi içerisinde tutmaktadır.

Bunun yanı sıra,  $M_k$  operatörü her ne kadar her bir düğümün derecesini korusa da tümüyle bağlı bir Boolcu ağı iki bağımsız ağa, yani iki bağımsız dinamik sisteme dönüştürebilir. Bu Bölüm 3.2nde açıklanan nedenlerden kaçındığımız bir durumdur. Dolayısı ile aynı simülasyon, hem dereceleri hem de tek parçalılığı koruyan  $M_{kc}$  operatörü ile gerçekleştirilmiştir.

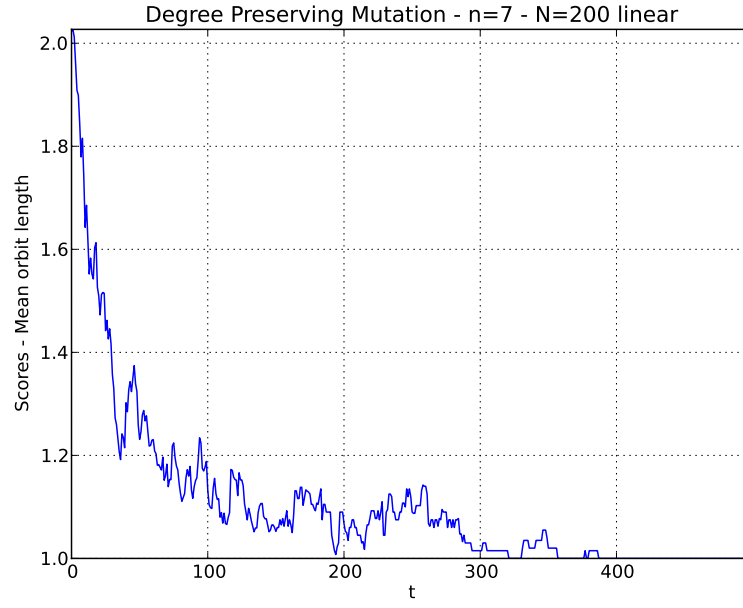


Şekil 4.10:  $\tau = 0$  anında  $M_k$  mutasyonunun uygulandığı genetik algoritmadaki Motif sıklıkları

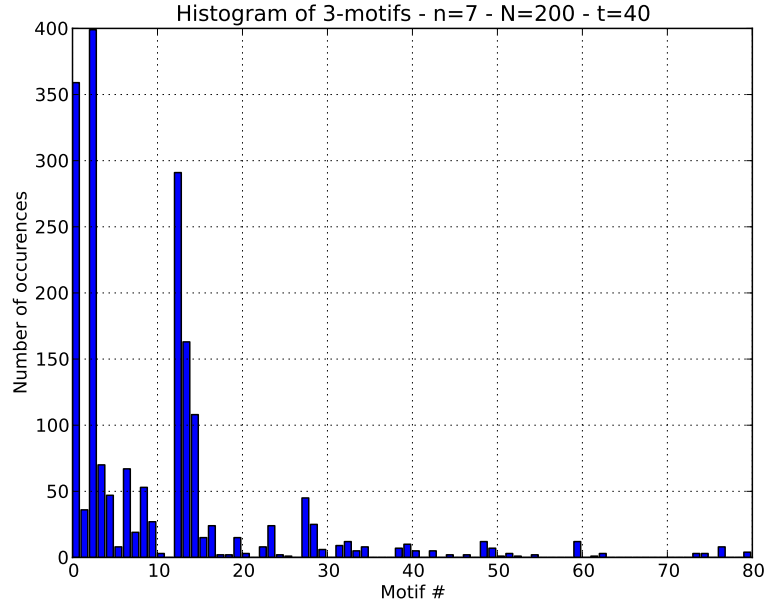
Genetik algoritma koşulduğunda topluluktaki istatistiksel parametrelerin değişimi aşağıdaki gibidir.



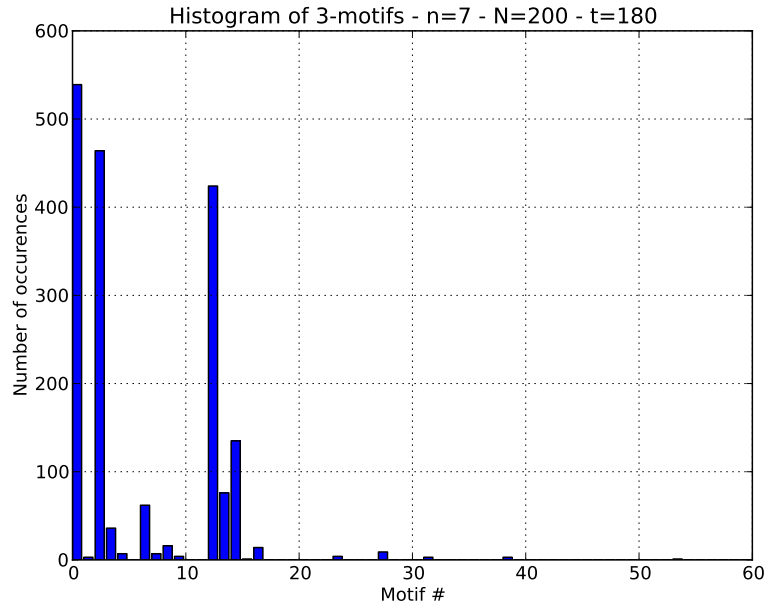
Şekil 4.11:  $M_k$  ile Genetik Algoritmaya tabii tutulan topluluğun tüm düğümler üzerinden derecelerinin ortalamasının zamana göre değişimi



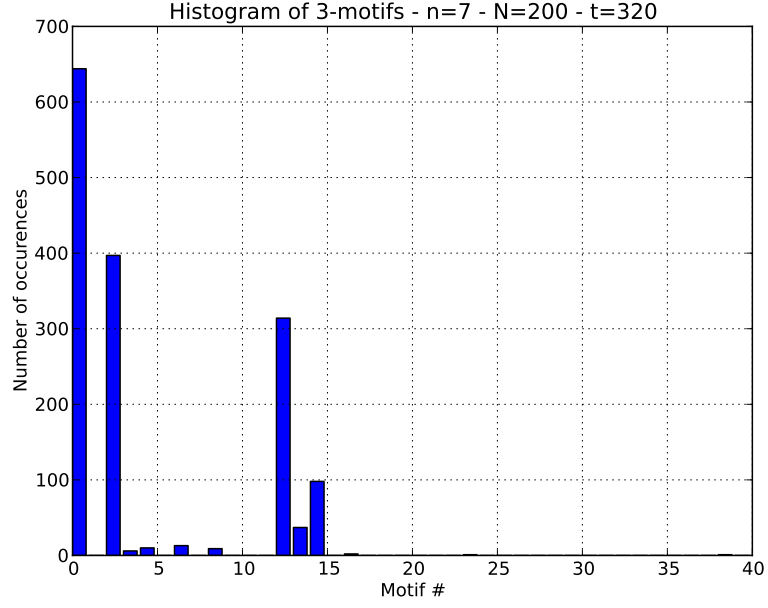
Şekil 4.12:  $M_k$  ile Genetik Algoritmaya tabii tutulan başarımının zamana göre değişimi. GA'da düşük değerler tercih edilmiştir.



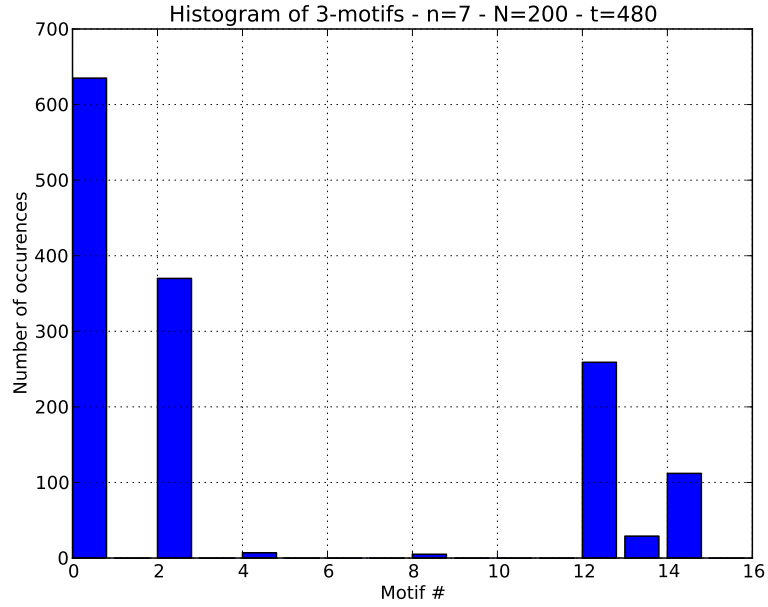
Şekil 4.13:  $\tau = 40$  anında  $M_k$  mutasyonunun uygulandığı genetik alitmadaki Motif sıklıkları



Şekil 4.14:  $\tau = 180$  anında  $M_k$  mutasyonunun uygulandığı genetik alitmadaki Motif sıklıkları



Şekil 4.15:  $\tau = 320$  anında  $M_k$  mutasyonunun uygulandığı genetik algoritmadaki Motif sıklıkları



Şekil 4.16:  $\tau = 480$  anında  $M_k$  mutasyonunun uygulandığı genetik algoritmadaki Motif sıklıkları

$M_k$  yerine  $M_{kc}$  operatörü kullanıldığında genel olarak çok büyük bir değişim görülmemektedir.  $\langle k \rangle$  ve  $\langle G \rangle$  düşmekte, ve daha önce bahsedilen motiflerin frekansı yüksek çıkmaktadır.

Burada ilginç bir nokta, bir ağ rastgele bağlantılandırıldığında 13. ve 1. motiflerden iki kat daha sık çıkan 3 numaralı motif, Genetik algoritmanın ilerlemesinden sonra

13 ve 1 numaralı daha sık görülmeye başlanmıştır. Tüm topluluktaki 3 numaralı motifin sayısının değişmemesi, fakat 1 ve 13 numaralı motiflerin hızla artması gözlemlenen duruma neden olmuştur. İlginçtir ki sistemler, durum uzaylarındaki karışıklığı ( $G$  fonksiyonunu) azaltmak için 3 numaralı motifi seyrekletmeye gitmişlerdir. Bunun genel geçerliğini test etmek için bunun gibi bir kaç simülasyon yapmak gerekecektir.

# Ek A

## Tanımlar

### A.1 Çizgeler

#### A.1.1 Çizge

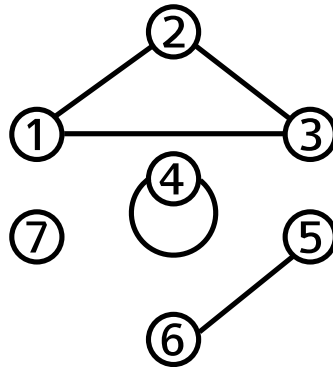
##### Tanım A.1.1 (Çizge)

Bir çizge, düğüm kümesi  $V$  ve kenar kümesi  $E$  sıralı ikisi  $(V, E)$  ile tanımlanan bir objedir. <sup>1</sup>

Çizgelerde, düğüm kümesi boş olmayan herhangi bir küme,  $v_i \in V$ ,  $E$  kümesi ise boş değil ise, elemanları  $V$  kümesinin iki elemanlı alt kümelerinden  $e_k = \{v_i, v_j\} \in E$  oluşturulan bir kümedir. [9]

##### Örnek A.1.1

Bir örnek vermek gerekir ise, bir  $G = (V, E)$  çizgesi  $V = \{1, 2, 3, 4, 5, 6\}$  ve  $E = \{\{1, 2\}, \{1, 3\}, \{3, 2\}, \{4, 4\}, \{5, 6\}\}$  ikilisi ile oluşturulabilir. Bu grafin temsili genellikle nokta ve çizgiler ile yapılır.



Şekil A.1: Örnekteki  $G = V, E$  çizgesinin temsili

<sup>1</sup>Bu bölümde çizgelerin ve Boolcu ağların sadece bitirme çalışmasını ilgilendiren özellikleri ve tanımlar üzerinde durulacaktır. Daha fazla bilgi edinmek için [9], [8] kaynaklarına göz atabilirsiniz.

Temsilin de yardımı ile, kenar kümesindeki bir elemanın, aslında iki düğüm arasındaki bir ilintiyi betimlediği kestirilebilir. Bu kurulan ilinti, "bir  $e_k = v_i, v_j$  kenar kümesi elemanı (yahut kısacası  $e_k$  kenarı)  $v_i$  ile  $v_j$  düğüm kümesi elemanlarını (düğümlerini) bağlıyor" şeklinde açıklanır.

Çizgelerin tanımlanmış olması, çizgilerarası ilişkilerin de tanımlanmasını gerektirecektir. Bu tanımları ele alalım.

### A.1.2 Eşitlik

**Tanım A.1.2 (Çizge eşitliği)**

[9]  $G_1 = (V_1, E_1)$  ve  $G_2 = (V_2, E_2)$  çizgeleri birbirine  $V_1 = V_2$  ve  $E_1 = E_2$  küme eşitlikleri sağlandığında söz konusudur.

### A.1.3 Komşuluk Matrisi

Çizgelerin tanımı daha temel objeler olan kümeler ile yapılmıştır, fakat çizgelerin nasıl bağlandığı ile ilgili bilgiyi farklı objeler ile de temsil etmek mümkündür. Zira kümeler, her ne kadar bir çizgenin en basit objeler ile betimini sağlasa da, daha karmaşık objeler olan matrisler, bazı yönlerden kolaylıklar sunabilir.  $n$  düğüm sayısına sahip bir çizgenin temsili  $n \times n$  büyüklüğünde bir matris ile gerçekleştirilebilir. Çizgedeki düğüm kümesi  $V = \{v_0, v_1, \dots, v_{n-1}\}$  olsun, bunun sonucunda  $E$  kümesinde  $(v_i, v_j)$  ikilileri bulunacaktır. Olası tüm  $(v_i, v_j)$  kombinasyonlarını içeren  $E$  kümesinin her hangi bir alt kümesi, ilgili  $A_{ij}$  elemanı 1 olan ve diğer elemanları 0 olan bir  $A$  matrisi ile betimlenebilir.

**Tanım A.1.3**

Bir  $G = (V, E)$  çizgesi olsun varsayalım. Bu  $G$  çizgesinin düğüm sayısı  $n$ ,  $i$ . düğümü temsil eden  $V$  kümesi elemanı da  $v_i$  olsun. Bu  $G$  grafını betimleyen  $A$  komşuluk matrisi aşağıdaki gibi belirlenir.

$$A_{ij} = \begin{cases} 1 & ; (v_i, v_j) \in E \\ 0 & ; (v_i, v_j) \notin E \end{cases} \quad (\text{A.1})$$

Bundan sonra bir çizge betimlerken, iki gösterim de birbirleri ile dönüşümlü olarak kullanılacaktır. Komşuluk matrisi ile biricik olarak betimlenebilen çizgeler, olası tüm çizgelerin bir alt kümesidir, lakin bu çalışmada düğüm kümesinin elemanlarının ne olduğu önem teşkil etmediğinden <sup>2</sup>,  $A$  matrisi esas alınacaktır.

<sup>2</sup> $G_1$  ve  $G_2$  çizgeleri  $(V_1 = \{0, 1, 2\}, E_1 = \{\{1, 2\}\})$  ve  $(V_2 = \{a, b, c\}, E_2 = \{\{b, c\}\})$  kümeleri ile gösterilebilir, bu iki çizge komşuluk matrisleri olan  $A_1$   $A_2$  ile ayırd edilemez, lakin burada bu tip bir ayırımı ihtiyacımız bulunmamaktadır.



#### A.1.4 Yönelimli Çizge

$A$  matrisinin tanıtılması aynı zamanda başka bir belirsizliği ortadan kaldırmaktadır. Her hangi bir  $G$  grafinin kenar kümesi  $E$  nin  $\{e_i, e_j\}$  elemanı ile  $\{e_j, e_i\}$  elemanı birbirini ile ayırd edilmemektedir, bu da kenarların yönelimi bilgisinin Tanım A.1.1 deki yapıda taşınmadığını gösterir. Fakat bazı durumlarda bir çizge ile betimlenen olguda yönelim önem teşkil eder. Bunun için yeni bir tanıma ihtiyaç doğmaktadır, Tanım A.1.1 aşağıdaki gibi değiştirilir ise , **yönelimli çizgenin** tanımı yapılabilir.

##### Tanım A.1.4 (Yönelimli Çizge (Digraph))

$G$  yönelimli çizgesi, düğüm kümesi  $V$  ve kenar kümesi  $E$  sıralı ikisi  $(V, E)$  ile tanımlanan bir objedir.  $V$ , düğüm kümesi boş olmayan herhangi bir küme,  $v_i \in V$ ,  $E$  kümesi ise boş değil ise, elemanları  $V$  kümesinin elemanlarından oluşan  $e_k = (v_i, v_j)$  **sıralı ikililerinden** oluşan bir kümedir.[3]

Bu tanıma göre bir  $G = (V, E)$  çizgesindeki  $E$  nin  $(e_i, e_j)$  elemanı ile  $(e_j, e_i)$  artık ayırt edilebilir. Bir  $A$  matrisinin  $A_{ij}$  ve  $A_{ji}$  elemanları da ayırd edilebilir olduğundan ,  $A$  matrisi bir yönelimli çizgeyi de betimleyebilir. Çalışmada belirtilen çizgeler aksi belirtilmedikçe yönelimli çizgelerdir.

#### A.1.5 Çizge Eşyapılılığı

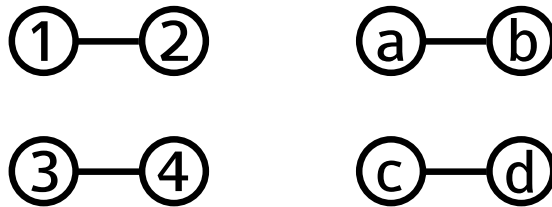
##### Örnek A.1.2

$G_1 = (V_1, E_1)$  ve  $G_2 = (V_2, E_2)$  çizgelerini ele alalım ve ilgili kümeler aşağıdaki elemanlara sahip olsunlar.

$$V_1 = \{1, 2, 3, 4\} , E_1 = \{\{1, 2\}, \{3, 4\}\}$$

$$V_2 = \{a, b, c, d\} , E_2 = \{\{a, b\}, \{c, d\}\}$$

Bu çizgeler Tanım A.1.2'e göre eşit değildir, fakat bu grafları çizseydik aşağıdaki gibi yapıları benzer olacaktı.



Şekil A.2: Örnekteki  $G_1$  ve  $G_2$  çizgelerinin temsilleri

Bu iki çizge arasındaki yapı benzerliğini bir tanım ile pekiştirilmesi gerekmektedir.

**Tanım A.1.5 (Çizge eşyapılılığı (isomorphism))**

$G_1 = (V_1, E_1)$  ve  $G_2 = (V_2, E_2)$  çizgeleri olsun.  $v_{1i} \in V_1$  ve  $v_{2j} \in V_2$  olmak üzere,  $\phi : v_{1i} \mapsto v_{2i}$  ataması tanımlansın.  $\phi$  atamasından türetilen bir başka  $\Phi : (v_{1i}, v_{1j}) \mapsto (\phi \cdot v_{1i}, \phi \cdot v_{1j})$  ataması olsun.

Bir  $\phi$  ve  $\phi$  den türetilen biricik  $\Phi$  için,  $(\phi \cdot V_1, \Phi \cdot E_1) = (V_2, E_2)$  geçerli ise,  $G_1$  ve  $G_2$  eşyapılıdır, ve bu ilişki  $G_1 \cong G_2$  şeklinde gösterilir.

Dolayısı ile, Örnek A.1.2 deki  $G_1$  çizgesi,  $G_2$  çizgesi ile eşyapılıdır, zira  $\phi : 1 \mapsto a, 2 \mapsto b, 3 \mapsto c, 4 \mapsto d$  dönüşümü,  $V_1 = V_2$  ve aynı anda  $E_1 = E_2$  yapar. Bu örnekte eşyapılılık tasvirleri birbirinin tıpkısı olan iki çizge üzerinde gösterildi, fakat eşyapılılık bu kadar aşık olmayaabilir.

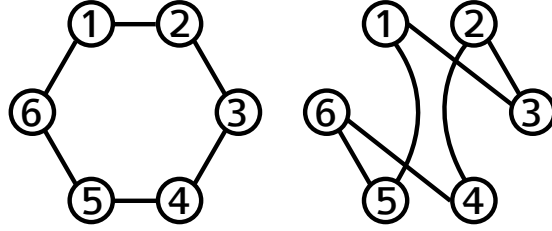
**Örnek A.1.3**

$G_1 = (V_1, E_1)$  ve  $G_2 = (V_2, E_2)$  çizgeleri aşağıdaki gibi tanımlansın.

$$V_1 = \{1, 2, 3, 4, 5, 6\}, E_1 = \{\{1, 2\}, \{2, 3\}, \{3, 4\}, \{4, 5\}, \{5, 6\}, \{6, 1\}\}$$

$$V_2 = \{1, 3, 2, 4, 6, 5\}, E_2 = \{\{1, 3\}, \{3, 2\}, \{2, 4\}, \{4, 6\}, \{6, 5\}, \{5, 1\}\}$$

Bu çizgeleri çizdiğimizde eşyapılı oldukları Örnek A.1.2 deki gibi net değil.



Şekil A.3: Örnek A.1.3 de tanımlanmış  $G_1$  ve  $G_2$  çizgelerinin temsilleri

Fakat,  $\phi : (1, 2, 3, 4, 5, 6) \mapsto (1, 3, 2, 4, 6, 5)$  permütasyonu  $V_1$  ve  $E_1$  e uygulandığında,  $V_1 = V_2$  ve  $E_1 = E_2$  eşitlikleri sağlanacaktır, dolayısı ile  $G_1 \cong G_2$  dir.

Çizge eşyapılılığı algoritması Bölüm B.1'de açıklanmıştır.

**A.1.6 Derece****Tanım A.1.6 (Derece)**

Bir  $G$  çizgesindeki  $v_i$  düğümünün derecesi,  $v_i$  ye bağlı olan düğümlerin sayısıdır ve  $k_i$  ile gösterilecektir.

Yönelimli çizgelerde derece iki şekilde, bir düğüme giren ve o düğümden çıkan bağlantıların sayısı olarak tanımlanabilir.

**Tanım A.1.7 (Derece)**

Bir  $G$  çizgesindeki  $v_i$  düğümünün iç-derecesi o düğüme giren, dış-derecesi o düğümden çıkan bağlantıların sayısıdır. Sırası ile iç-derece ve dış-derece komşuluk matrisinden

$$k_j^- = \sum_{i=0}^{n-1} A_{ij}$$

$$k_i^+ = \sum_{j=0}^{n-1} A_{ij}$$

şeklinde elde edilir.

## Ek B

# Algoritmalar

### B.1 Çizge Eşyapılılığı Algoritması

Boolcu ağlar ile çalışırken gerekli olan fonksiyonlardan birisi, iki çizgenin bir-biri ile eşyapılı olup olmadığını kontrol eden bir fonksiyon olmuştur. Bir çizge, eğer bir fiziksel sistemi betimliyor ise, ve bu sistemde düğümler arasındaki fark gözetilmeyecek ise, ya da ağın sadece topolojik özellikleri ile ilgileniliyor ise, iki çizgenin eşyapılılığı iki sistemin de aynı olduğunu, ya da topolojik olarak ikisinin arasında fark gözetilemeyeceği anlamına gelir. Dolayısı ile çalışmada iki çizgenin eşyapılı olup olmadığını kontrol eden bir fonksiyon yazılmıştır.

Eşlenikliği test edilecek  $n$  düğümlü yönelimli çizgeler,  $G_1$  ve  $G_2$  olsun. <sup>1</sup> Bu çizgeler,  $A_1$  ve  $A_2$  komşuluk matrisleri ile temsil edilsinler. Eğer  $G_1$  çizgesinin düğümlerinin bir permütasyonu sonucunda oluşacak bir  $G_1^*$  çizgesi,  $G_2$  çizgesi ile eşit ise,  $G_1$  ile  $G_2$  çizgesi eşyapılıdır. Dolayısı ile eşyapılılığı kontrol eden bir algoritma, aşağıdaki gibi kodlanabilir.

Öncelikle, eğer  $G_1$  çizgesindeki düğümlerin dereceleri ile  $G_2$  çizgesindeki düğümlerin dereceleri arasında birinde olup da öbüründe olmayan bir değer var ise, düğümlerin yerlerini değiştirmek dereceleri değiştirmeyeceğinden,  $G_1 \not\cong G_2$  olduğu söylenebilir. Eğer bu durum söz konusu değil ise düğümlerin yerlerini değiştirmemiz gerekmektedir.

$n$  tane düğüm,  $n$  uzunluklu bir  $(1, \dots, n)$  dizisindeki elemanlar ile temsil edilsin. Bu dizinin  $n!$  farklı permütasyonunu barındıran küme de  $P$  kümesi olsun.

$P$  kümesindeki her bir  $p_k$  permütasyonu  $G_1$  çizgesindeki düğümleri yani yaratılan  $G'$  çizgesindeki yeni sıralarına atamaktadır. Bu atama sonucunda,  $i$ . düğüm, yeni çizgede  $i'$ . düğüm olsun.  $G_1$  çizgesinde  $i$ . düğüme  $j$ . düğüm bağlı ise,  $p_k$  permütasyonu özelinde  $i'$ . düğüme  $j'$ . düğüm bağlı olacaktır. Sonuçta,  $A_{ij}$  nin değeri ne ise,  $G'$  grafinin komşuluk matrisi olan  $A'$  nün  $A'_{ij}$  o değeri alacaktır. Dolayısı ile

---

<sup>1</sup>İki çizgenin düğüm sayısı eşit değil ise zaten eşyapılı olamazlar.

her  $p_k \in P$  permütasyonu için  $A'$  yaratılır. Bunlardan herhangi birisi  $A_2$  ye eşit ise,  $G_1 \cong G_2$  dir.  $G_1 \not\cong G_2$  durumu ise ancak  $P$  kümesindeki tüm permütasyonlar için  $A' \neq A_2$  ise söylenebilir.

## B.2 Çizgenin tek parçalılığının kontrolü

Üzerinde çalışacağımız çizgelerin tek parça halinde kalması, birbiri arasında bağlantısı olmayan iki ayrı çizgeye ayrılmaması bizim için önem teşkil etmektedir Dolayısı ile bir çizgenin tek parça olduğunu kontrol eden algoritma açıklanmalıdır.

Bunun için bir çizgenin Laplaceyan matrisinin bir özelliği kullanılacak. Bir çizgenin Laplaceyan matrisi, o çizgenin düğümlerinin derecelerinin köşegeninde bulunduğu  $D$  matrisi ile komşuluk matrisi  $A$  farkı ile oluşturulur.

$$L = D - A \quad (\text{B.1})$$

$L$  matrisinin özelliklerinden birisi, 0 olan özdeğerlerinin sayısının o çizgenin kaç tane parçadan oluştuğunu göstermesidir. Dolayısı ile çizgenin üzerinde “emekleyerek” çizgenin tümünün bağlı olup olmadığını kontrol eden rutinlerden ziyade, sadece bir matrisin özdeğerlerini bulup sayan bir rutin yeterli olacaktır.

Dolayısı ile çizgenin tek parçadan oluşup oluşmadığını kontrol eden algoritma,  $L + \mathbb{I}$  matrisini oluşturarak her çizgede çıkan sıfır özdeğeri yok edip sıfır özdeğeri varsa olumsuz, sıfır özdeğeri yoksa olumlu cevap dönmektedir.

## B.3 Motif Frekansları Çıkaran Algoritma

Bir çizgedeki  $n$  düğümlü motiflerin bulunması, çizgedeki tüm olası  $n$  düğümlü alt çizgeleri çıkarılması, ve her bir alt çizge için verilen bir listede eşyapılı çizgesinin sayısının düzeltilmesi işlemidir. Bunun için öncelikle motif listesinin nasıl çıkarıldığına bakmak gerekir.

### B.3.1 Tüm Olası Motiflerin Çıkarılması

Olası tüm  $n$  düğümlü çizgeler,  $n \times n$  büyüklüğündeki bir komşuluk matrisinin alabileceği tüm durumlardan oluşturulabilir, bunun sonucunda  $2^n$  tane çizge oluşturulur. Lakin bu çizgelerin bir kısmı kopuk kısımlara sahip olabilir, ya da bu listedeki iki çizge birbiri ile eşyapılı çıkabilir, dolayısı ile algoritma aşağıdaki gibi oluşturulmuştur.

1. Öncelikle 0 dan  $2^n$  e kadar tüm tamsayıların bir listesi oluşturulur ve listedeki sayılar ikili (*Binary*) sayılara dönüştürülürler. Her bir liste elemanı, 0 ve 1 den

oluşan  $n^2$  uzunluğundaki bir listedir.

2. Listedeki her bir vektör  $n \times n$  lik bir matrise dönüştürülür, ve bu matrislerden çizgeler yaratılır. Bu listede tüm olası  $n$  düğümlü çizgeler bulunur.
3. Listedeki bir çizge alınır.
  - (a) Bu çizgenin her bir düğümü birbiri ile bağlı mı kontrol edilir. Bunun için Bölüm B.2 deki algoritma kullanılır. Eğer bu çizge tek parçalı ise devam edilir, değil ise listeden bir sonraki çizge alınır.
  - (b) Çizgenin oluşturulan havuzda bir eşleniğinin bulunup bulunmadığına bakılır. Eğer bir eşyapılı yoksa, çizge havuza eklenir, eğer bir eşyapılı varsa listeden bir sonraki çizge alınır.
4. Listedeki son çizge de algorithmadan geçtiğinde havuzda birbiri ile eşlenik olmayan, tek parçalı motiflerin listesi oluşmuştur.

### B.3.2 Motif Frekanlarını Çıkaran Algoritma

$m$  düğümlü bir ağın  $n$  düğümlü motiflerinin frekansları çıkarılması için:

1.  $(1, \dots, m)$  dizisinin  $\binom{m}{n}$  adet  $n$  li kombinasyonu bir liste halinde çıkarılır. Her bir kombinasyon  $n$  uzunluğunda bir vektör olacaktır.
2. Her bir kombinasyon için:
  - (a) Bu kombinasyondaki düğümlere sahip olan alt çizge çıkarılır. Mesela  $(1, 4, 5)$  dizisi için sadece 1., 4. ve 5. düğümler ve aralarındaki bağlantılara sahip çizge çıkarılır.
  - (b) Bu çizgenin eşleniği Bölüm B.3.1 te oluşturulmuş listede bulunur ve listede bu çizge için olan sayı bir artırılır.
3. Tüm kombinasyonlar bitirildiğinde tüm motifleri içeren listedeki sayılar çizgede ilgili motifin kaç kere bulunduğunu gösterir.

## B.4 Çalışmada Kullanılan Kod

Çalışma sırasında Boolcu ağların dinamiğinin incelenmesini, bu ağlardan istatistiksel verinin elde edilmesini, Genetik algoritmanın koşulmasını sağlayan bir büyük modül Python dilinde yazılmıştır. Bu bitirmedeki tüm algoritmalar ve daha fazlası bu modülden incelenebilir. Modüle *Kreveik* adı verilmiştir ve bir Git kütüphanesi olarak <https://github.com/mehmetalialil/Kreveik> adresinden ulaşılabilir, kaynak kodu buradan incelenebilir.

Aynı zamanda koda katkıda bulunmak, kullanmak isteyenler Git programı yükledikten sonra kendi kopyalarını (*fork*) aşağıdaki komutlarla edinebilirler.

```
1|git clone git://github.com/mehmetalianil/Kreveik.git
```

Bu komut, çalışmada kullandığım **kaynak kodunun** son halini bulduğunuz klasöre yerştirecektir. Programın yeni versiyonu çıktıkça kaynak kodu

```
1|git pull origin
```

komutu ile yenilenebilir. Kaynak kodunda değişiklik yapmışsanız bu yenileme işlemi gerçekleşmez, bu durumda yaptığınız modifikasyonun yazılıma katmayı konuşmak için [mehmet.ali.anil@ieee.org](mailto:mehmet.ali.anil@ieee.org) adresine email atabilirsiniz.

# Kaynakça

- [1] P.J. Bentley and D.W. Corne, editors. *Creative Evolutionary Systems*. Morgan Kaufmann, San Francisco, 2002.
- [2] Neil A. Campbell, Jane B. Reece, Lawrence G. Mitchell, and Martha R. Taylor. *Biology: Concepts and Connections (4th Edition)*. Benjamin Cummings, 4 edition, 7 2002.
- [3] Edgar G. Goodaire and Michael M. Parmenter. *Discrete Mathematics with Graph Theory (3rd Edition)*. Prentice Hall, 3 edition, 7 2005.
- [4] Stephen E. Harris, Bruce K. Sawhill, Andrew Wuensche, and Stuart Kauffman. A model of transcriptional regulatory networks based on biases in the observed regulation rules. *Complexity*, 7(4):23–40, 2002.
- [5] François Jacob and Jacques Monod. Genetic regulatory mechanisms in the synthesis of proteins. *Journal of Molecular Biology*, 3(3):318 – 356, 1961.
- [6] L. Kadanoff, S. Coppersmith, and M. Aldana. Boolean Dynamics with Random Couplings. *eprint arXiv:nlin/0204062*, April 2002.
- [7] S. A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.*, 22:437–467, Mar 1969.
- [8] Stuart A. Kauffman. *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, USA, 1 edition, 6 1993.
- [9] Richard J. Trudeau and Mathematics. *Introduction to Graph Theory (Dover Books on Mathematics)*. Dover Publications, 2nd edition, 2 1994.



# Dizin

Boolcu Ağlar, 3

yönelimli çizge  
tanım , 32

çizge  
eşitlik, 31  
eşyapılılık, 33  
tanım, 30  
yönelimli, 32