

**T.C.  
EGE ÜNİVERSİTESİ  
SOSYAL BİLİMLER ENSTİTÜSÜ  
Temel Bilimler Anabilim Dalı  
Duysal Tasarım Doktora Programı**

**Uygulamalı Matematik Yoluyla,  
Müzikte Kendiliğinden Değişim Üretilmesi**

Doktora Tezi

Hazırlayan:  
Aykut Çağlayan  
92060002692

Tez Danışmanı:  
Prof. Dr. Yetkin Özer

İZMİR  
2010

27/08/2010

Ege Üniversitesi Sosyal Bilimler Enstitüsü Müdürlüğüne sunduğum “Uygulamalı Matematik Yoluyla, Müzikte Kendiliğinden Değişim Üretilmesi”.adlı doktora tezinin tarafımdan bilimsel, ahlak ve normlara uygun bir şekilde hazırlandığını, tezimde yararlandığım kaynakları bibliyografyada ve dipnotlarda gösterdiğimi onurumla doğrularım.

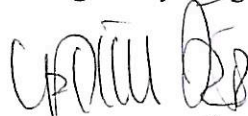
Aykut ÇAĞLAYAN



## TUTANAK

Ege Üniversitesi Sosyal Bilimler Enstitüsü Yönetim Kurulu'nun 04/08/2010 tarih ve 26/171 sayılı kararı ile oluşturulan jüri **Temel Bilimler (Duysal Tasarım)** anabilim dalı doktora öğrencisi **Aykut ÇAĞLAYAN**'ın aşağıda başlığı (Türkçe / İngilizce ) belirtilen tezini incelemiş ve adayı 27/08/2010 günü saat 10.00'da ..... dakika süren tez savunmasına almıştır.

Sınav sonunda adayın tez savunmasını ve jüri üyeleri tarafından tezi ile ilgili kendisine yöneltilen sorulara verdiği cevapları değerlendirerek tezin başarılı/başarısız/düzeltilmesi gerekli olduğuna oybirliğiyle / oyçokluğuyla karar vermiştir.



Prof. Dr. Yetkin ÖZER

Başarılı

Başarısız

Düzeltilme (6 ay süreli)

Prof. Dr. Ahmet YÜRÜR

Başarılı

Başarısız

Düzeltilme (6 ay süreli)

Doç. Dr. Reyhan ALTINAY

Başarılı

Başarısız

Düzeltilme (6 ay süreli)

Doç. Dr. Uzun BORA

Başarılı

Başarısız

Düzeltilme (6 ay süreli)

Yrd. Doç. Dr. Onur NURCAN

Başarılı

Başarısız

Düzeltilme (6 ay süreli)

Tezin Türkçe Başlığı : "Uygulamalı Matematik Yoluyla, Müzikte Kendiliğinden Değişim Üretilmesi"

Tezin İngilizce Başlığı : "The Use of Applied Mathematics in Music to Produce Spontaneous Change"

## Önsöz

Tarih boyunca, sanatçıların eserlerinde doğadan esinlenmesi genel bir fenomendir. Platon ve Aristo gibi filozoflarca da ele alınmış *mimesis* kavramı sanat tarihi boyunca, farklı kültürel bağlamlar içinde karşımıza çıkmaktadır. Kabaca doğayı taklit anlamındaki *mimesis*, kendi içinde farklı anlamlara işaret etmektedir. Bu farklılıklar izlenimsel tasvirde, doğanın temelinde işlediği düşünülen mantığın açılmasına kadar değişik varyantlar sergilemektedir. Algı biçimleri ve epistemoloji değiştikçe doğanın nasıl taklit edildiği, ve doğaya dair olanın eser içinde nasıl temsil edildiği/yeniden üretildiği, farklılıklar sergilemektedir. Her çağın ve kültürel bağlamın doğa tasarımı farklı olduğu için *mimesis* de sanat yapıtlarında katmanlı anlam ve tasarım ağlarının muğlak bir ögesi olarak ortaya çıkabilmektedir. Anlamın katmanlanmasında kültürel örüntüler devreye girmektedir. Müziğe, kültürel bağlamlar içinde sembolik anlamlar yüklenmektedir. Bu sembollerin kültüre yabancı dinleyici tarafından algılanması mümkün değildir. Öte yandan, benim üretmeye çalıştığım müzik, değerini, sembolizmi/semantizmi aşabilmesinden almaktadır. Modern müziğe yönelmemde etkin olan itki, müziğin kültürel konvansiyonları aşabilme potansiyelidir. Kültürel konvansiyonların aşılması, diyalektik olarak gene, kültürün bir uzantısı olan bilimsel/hümanistik çalışmalar ile mümkün olabilmektedir. Ve müzik, sanatçılara, diğer alanlardaki zihinsel üretimleri kendi estetiğine içkinleştirebilme olanağı sunmaktadır.

XX. yy. modernizmi ve XXI. yy'a uzanan dijital çağdan beslenmiş bir sanatçı/araştırmacı olarak benim doğa tasarımı oluşturulan bilimsel kuramlar; parçacık fiziği (*quantum mechanics*), Heisenberg'in belirsizlik ilkesi, kaos ve *complexity* kuramlarıdır. Bu kuramların benim için önemi, doğa algıma denk düşen betimleyici ve modelleyici kapasiteleridir. Bu kapasitelerin başlıcaları, akışkanlar mekaniğinin modellenmesi ya da basit ilişki ağlarının üretebildiği kompleks yapılardır. Estetik olarak da, tercihim, doğa algımın başat ırası olan kaotiklik ve kendiliğinden örgütlenmenin (*self-organisation*) müzikal olarak yansıtılması yönündedir. Doktora tez çalışmamın

ana motivasyonu bu olmuştur.

Müzik öğrenimi görmüş bir kimse olarak, teknik yönü ağır basan bu konulara yönelmemde orta öğrenimde aldığım kuvvetli matematik ve doğa bilimleri derslerinin faydası olmuştur. Üniversite öğrenimim esnasında aldığım bir derste, Lisp programlama dili ile tanışmam, bilgisayar programlamanın sunduğu/sunabileceği imkanlara duyarlık kazanmamda çok etkili olmuştur. Lisp dilinde yetkinleşmemde yararlandığım kaynaklar MIT yayınları oldu (Abelson 1996, Felleisen 2001). Lisp dilinde müzikal çalışmalara başlamadan evvel MIAM'da Reuben de Lator'un verdiği *computer music* derslerinde Max/MSP programlama dilinde çalışmalar yapmaya başlamıştım. Bu çalışmaların odağı tamamen algoritmik besteleme değildi; daha ziyade, bilgisayar ve arabirimlerinin müzikal performansta kullanımına yönelik pratik çalışmalardı. Lisp diline ve algoritmik bestelemeye tam olarak yoğunlaşmamda, Heinrich Taube'nin *Notes from the Metalevel* adlı kitabı çok önemli bir yer tutmaktadır. Kitabını okuyarak başlayan bu süreç, Dr. Taube ile University of Illinois at Urbana-Champaign, School of Music, Composition/Theory Department'da geçirdiğim bir yıl süresince yaptığımız çalışmalarla pekişmiştir. Bu çalışmalarda standart prosedürlerin uygulanması yanında özgün çalışmalar da gerçekleştirilmiştir (*sieve* ve dinamik markov zincirleri gibi). Özgün projeler geliştirebilmek için gerekli bakış açısını kazanmamda etkili olan etkenlerden biri de, Dr. Pınar Yolum'un 2008/2009 döneminde, Boğaziçi Üniversitesi'nde verdiği *Artificial Intelligence* derslerini takip etmem olmuştur.

Bu tez çalışmasında doğadaki komplekslik ve kendiliğinden örgütlenmeye dair modeller olan hücresel otomatlar (*Cellular Automatas*) ve rastantısal Boole ağları (*Random Boolean Networks*) araştırılmıştır. Müzikal zamanın kuruluşu ve mekanik zamanın aşılması perspektifinde bu modellerin müzik kompozisyonu ve ses sentezlemede kullanımına dair teknikler geliştirilmiştir. Bu konularda çalışacak diğer besteci ve araştırmacıların da kullanabileceği çeşitli bilgisayar kodları yazılmıştır.

Tezin sonuçlandırılması ve olgunlaştırılması sürecinde verdiđi destek için danıřmanım Prof. Dr. Yetkin Özer'e; İstanbul ve İzmir'de Duysal Tasarım bölümlerini kurarak, Türkiye'deki akademik müzik çalışmalarına getirdiđi yeni bir soluk için Prof. Dr. Ahmet Yürür'e; üniversite eğitimim boyunca verdiđi desteklerden ötürü Prof. Dr. Alev Gerçeker'e; 2009-2010 döneminde doktora arařtırmamı Amerika'da sürdürebilmeme olanak sađlayan Fulbright Eğitim Komisyonu'na; özel görüşmelerimizde, tez konuma dair netlik kazanmama yardımcı olan Dr. Sever Tipei'ye; bu alanda çalışan arařtırmacı/bestecilere sunduđu *Common Music* sistemi ve algoritmik uygulamalarda deneyim ve pratiklik kazanmamı sađladıđı için Dr. Heinrich 'Rick' Taube'ye teşekkürlerimi sunuyorum.

## İçindekiler

Problem

Zaman Kavramı

Yöntem

- 1 Tarihsel Bağlamı içinde Algoritmik Kompozisyon
  - 1.1 XX Yüzyıl Öncesinde Algoritmik Kompozisyon
  - 1.2 XX. Yüzyılda Algoritmik Kompozisyon
    - 1.2.1 1957'den 1966'ya Urbana Okulu
      - 1.2.1.1 Hiller ve Isaacson
      - 1.2.1.2 James Tenney
      - 1.2.1.3 Herbert Brün
    - 1.2.2 Avrupa Algoritmik Müziği: 1960-1970
      - 1.2.2.1 Gottfried Michael Koenig
  - 1.3 Xenakis
    - 1.3.1 *Sieve*
    - 1.3.2 Markov zincirleri
    - 1.3.3 Hücresel otomatlar
  - 1.4 Biyoloji Esinli Algoritmik Uygulamalar
    - 1.4.1 *Ant Music*
    - 1.4.2 *Zoion*
- 2 Markov Zincirleri
  - 2.1 Markov Zincirlerinin Matematiksel Nitelikleri
  - 2.2 Müzikal Uygulamaları
  - 2.3 Markov Analizi
  - 2.4 Dinamik Markov Zinciri
- 3 Hücresel Otomatlar
  - 3.1 Game of Life (Yaşam Oyunu)
  - 3.2 Hücresel Otomatların Müzikal Uygulamaları
    - 3.2.1 MIDI Hedefli Hücresel Otomat Sistemleri
      - 3.2.1.1 Peter Beyls
      - 3.2.1.2 Dale Millen
      - 3.2.1.3 Eduardo Reck Miranda ve *Camus* Uygulaması
      - 3.2.1.4 Warren Kurt
      - 3.2.1.5 “WolframTones” ve Birebir Haritalama
    - 3.2.2 Ses Sentezlenmesinde Hücresel Otomatların Kullanımı
      - 3.2.2.1 Granüler Sentezleme
      - 3.2.2.2 Karplus-Strong Tipi Sentezleme
  - 3.3 Otomat Nesillerinin İstatistiki ve Esnek Yorumu
    - 3.3.1 Süreç İçinde Hücresel Otomat Kurallarının Değiştirilmesi

- 4 Rastlantısal Boole Ağları
  - 4.1 Boole Ağlarının Dinamik Kuruluşu
  - 4.2 Rastlantısal Boole Ağlarının Türetilmesi
    - 4.2.1 Sistemin Evrimsel Niteliği
  - 4.3 RBA'nın Müzikal Uygulamaları

## 5 Sonuç

Kaynakça

## Problem

*“Son dönemdeki algoritmik kompozisyon ve etkileşimli kompozisyon çalışmaları, tutarlı ancak öngörülemez çıktuların türetimi için, biyolojiden esinlenmiş modellere yönelmiştir”*

(Beyls 2003)

Müzik kompozisyonunda, geleneksel kalıpların aşılabilmesi ve kompleks yapıların organizasyonu, estetik anlamda tarafında durduğum bir konumdur. Kültürel kodlar ve bunlar üzerine inşa edilmiş müzikal kılımlar yerine, doğanın heybeti ve kendiliğindenliğinin geçirilmesi için yeni yöntemler ve müzik estetiği elzemdir. Kendiliğinden değişim, farklı zamansal katmanlarda kurulan müzikal kompozisyonun, organik bir ögesidir. Makro, *meso*, ses nesnesi ve mikro olarak ayrıştırılan zamansal katmanlarda (Roads 2001, 3) kurulmaktadır müzikal form. Ses nesnelere, *meso* zamansal katman içinde sahip olacağı dinamizm için kendiliğinden değişim en etkin yol olarak görünmektedir.(Mikro düzey organizasyonlar için Bkz. 3.2.2.1)

Algoritmik müzik kompozisyonunda, kendiliğinden değişimin sağlanması için biyolojik ve fiziksel modellemelerin kullanımı yaygınlık kazanmıştır. Bu modellemeler, belirli kural kümelerinin çeşitli başlangıç durumlarına uygulanması sayesinde, kompleks yapılar türetebilme özelliklerine sahiptir. Bu modellemelerin en kuvvetlisi Hücreli Otomatlar'dır. Tanımlı lokal ilişkilerin, zamansal süreç içinde kompleks yapılar oluşturabilmesi, hücreli otomat araştırmalarının en önemli motivasyonunu oluşturmaktadır. Bu sistemlerin araştırılmasında kullanılan genel yöntem, datanın iki ya da üç boyutlu planda grafik olarak temsiline dayanmaktadır. Bu türden dinamik sistemlerin müzik kompozisyonu ya da ses sentezlenmesinde kullanımında, sisteme içre kompleks karakterin müzikal olarak yansıtılmadığı fikrindeyim.

Müzikal uygulamalarda kompleks karakterin yansıtılmamasının bir nedeni, sınırlı

boyutlarla temsil edilen datanın birebir müzikal parametrelere haritalanmasından kaynaklanmaktadır. Bu noktada ihmal edilen ve müzikal estetiğe dair hayati önemde olan olgu ise, müziğin bir zaman sanatı olduğudur.

Matematikselsel idealizasyonla, temsili ve modellenmesi sağlanan, kendiliğinden-örgütlenmenin müzikal zamana aktarılmasında alternatifler yollar (yazılımlar) geliştirdiğim bu tez çalışmamda; kendiliğinden-örgütlenmenin doğası, matematikselsel özellikleri ve müzik kompozisyonuna yansımaları üzerine de veriler elde ettim.

### **Zaman kavramı**

Zaman kavramını iki ayrı düzlemde ele almak gerekmektedir;

Form ve forma dair öğelerin planlanması olarak, ve ritim ve temponun matematiği olarak.

Müzikteki zamansal boyutlar, Curtis Roads'ın *Microsound* adlı eserinde 9 katmanda tanımlanmıştır. Form ve formal öğeler bu katmanlara dahildir:

1. *Sonsuz* Matematikselsel sürelerin ideal zaman boyutudur. Örneğin klasik Fourier analizindeki bitevi sinüs dalgaları gibi.
2. *Supra* Bir müzik kompozisyonunu aşan bir süre ölçüsü, aylar ve yıllar gibi.
3. *Makro* Müzik yapıtının baştan sona süresidir. Dakikalar ve saatler sürer.
4. *Meso* Formun alt bölümleridir. Ses nesnelere, hiyerşarşik ya da figüratif biçimlerde kullanıldığı süre ölçüsüdür, dakika ya da saniyelerle ölçülür.
5. *Ses nesnesi* Müzikal yapının en yalın öğesidir. Bir nota ya da evrilen bir tını olabilir. Bir kaç saniye ya da daha kısa sürebilir.
6. *Mikro* İşitme eşiğinin altına kadar inen kısalıkta ses parçacıklarıdır. Milisaniyelerle ifade edilir.
7. *Sample* Dijital ses sisteminin atomik düzeyidir. Standart CD kalitesinde her bir *sample* saniyenin 44100'de biri uzunluğudur.
8. *Subsample* Kayıt edilmesi ve duyumsanması pek mümkün olmayan

kısalıktaki zamansal cereyanlardır. Nanosaniye ile ölçülür.

9. *Sonsuz küçük* Matematiksel olarak tanımlı en kısa zamansal birim. (Roads 2001, 3)

Geleneksel müziklerde, tempo genellikle sabittir (buna karşıt örnekler de verilebilir: Gamelan müziğinde *accelerando* ve *ritardando* çok belirgin bir stilistik öğedir) ve ritimler bu sabit temponun genellikle 2 ve ya 3'ün katlarına bölünmesi ile kurulmaktadır. Ve geleneksel müziklerin formu da, belirli sayıda bir kaç ögenin (diyelim ki A ve B) rasyonel (yani akılda kolay kalıcı) biçimde sıralanması ve zamansal olarak birbirine göre ölçülendirilmesiyle oluşturulmaktadır. Birbirlerine göre ölçülendirme genellikle 4 ve 4'ün katlarında olmaktadır.

Ancak 20. yüzyılda bu türden zaman kuruluşları terkedilmiştir. Elbette müzikteki yeni zamansal kuruluşların fizik bilimindeki paradigmatik değişimlerle doğrudan ilişkisi bulunmaktadır. Geleneksel müziklerdeki doğrusal (*linear*) zaman, modern müzikte doğrusal olmayan (non-linear) zaman önermelerine bırakmıştır yerini. Doğrusal olmayan zamanı ritm ve tempo anlamında netleştirecek olursak:

1. müzikal katmanların zamansal olarak kayması (*phase transition*) (Reich 1967; Taube 2007, 130)
2. katmanların birbirinden farklı periyotlara sahip olması (*polymetric*) (Bkz. G.Ligeti, Piano Concerto [1985-88])
3. vuruş hissini yok etmek için yoğun *tuplet* kullanımı (Bkz. E.Carter ve B. Ferneyhough'un bütün müzikleri)
4. düzenli ritimleri, küçük değerler ekleyerek aksak hale getirme (Messiaen, 1966)

Forma dair 20. yüzyıl müzik estetiğindeki başlıca iki özellik, rastlantısallık ve *process* üzerine müzikal zamanın kurulmasıdır. Rastlantısal formun kullanımında yapı fikri bir kenara bırakılmıştır; tanımlanmış parametreler rastlantısal

prosedürlerle değerler olarak yazıya geçirilmiştir ya da çeşitli grafik/betimsel yazı kullanılarak form sorumluluğu icracıya devredilmiştir. *Process* fikri ile müzik tarihinde, özellikle Beethoven'da çok belirgin olan, bir çok örneğiyle karşılaşılabılır. Ancak 20.yüzyıl estetiğinde *process* tek temalı (*monothematic*) olarak yaygınlık kazanmıştır. Steve Reich'in "*Pendulum Music*" ya da Alvin Lucier'in "*Music On A Long Thin Wire*" adlı eserleri bu estetiğin en billur örnekleridir. Geleneksel müziklerdeki formların zaman kuruluşlarının, soru-cevap, kontrast oluşturma, tonal anlamda beklenti yaratma, gerilim-çözülüm, yön hissi verme gibi işlevleri vardır. Ancak 20.yy'da bu türden işlevsellikler kaybolmuştur. Form öğelerinin sıralanışında, yukarda belirtilen nedensellekler yerini soyut modellere bırakmıştır.

Algoritmik kompozisyonda zamanı kurmanın bir yolu, algoritmik veriyi sağlayan sistemin bir parametresini (artan değer olarak) süreye haritalamak olabilmektedir. Bir başka yöntem de, rastlantısal olarak olasılık kümesinden bir süre değeri seçmektir. Bu iki yöntemde de, müziğin diğer parametreleri ile zamansal kuruluşu birbirinden bağımsız öğeler olarak kurulmaktadır.

Ancak algoritmik sistemlerin kullanımında; hafıza, iç zaman, olasılık sistemi içinde başlangıç ve hedef noktası arasındaki ara değerlerin hesaplanması (*interpolation*) gibi çeşitli önermeler üretildi. Bu tez çalışmasında, Rastlantısal Boole Ağları (*Random Boolean Networks*), (kendi tanımlamamızla) Dinamik Markov Zincirleri ve Hücresel Otomatlar'ın kendine özgün dinamiklerinin yorumlanması ve müziğe aktarılmasında bir dizi yeni yöntem geliştirildi. Boole Ağları ve Hücresel Otomatların kullanımında her bir birim/hücrenin ve boyutun tek bir değere ya da parametreye haritalanması terkedilerek, birimler fonksiyonlar olarak tanımlandı. Bu fonksiyonların başlangıçta aldıkları değerler, bağlı buldukları birimin her tetiklenişinde güncellenerek bir iç zaman oluşturuldu. Yani sistem belirli bir zamandan sonra başlangıç durumuna dönse dahi, çıkan müzikal sonuç ya da sistemin değersel çıktısı farklılık sergilemektedir.

## Yöntem

Fizik ve biyoloji bilimlerinde, doğadaki kompleksliği ve kendinden örgütlemeyi anlamak için çeşitli matematiksel modeller geliştirilmiştir. Bizim çalışmamızda bu modellerden 3 tanesinin (hücrel otomatlar, rastlantısal Boole ağları (RBA), Markov zincirleri) yeni bir müzik estetiğinin yaratılmasında nasıl kullanılabileceği geliştirdiğimiz çeşitli yazılımlarla ortaya konmuştur. Yani kompozisyonel önermeler/tercihler, fizik/biyoloji ve bilgisayar bilimleriyle akrabalığı içinde geliştirilmiştir. İlgili bölümlerde, sistemleri nasıl bir mantıkla kullandığımız, bilgisayar kodlarıyla verilmiştir. Sistemin müzikal çıktısı, partiyon ve ses kaydı olarak ekte verilmiştir.

Hücrel otomatlar için Wolfram'ın '*A New Kind of Science*' adlı çalışması ve *Mathematica* adlı uygulaması birincil kaynak oldu. Wolfram'ın önemi hücrel otomatları çok geniş bir bilimsel perspektif içinde sunabilmesi, hücrel otomatları birbirinden farklı yapılarılarını açıklamış olması ve bunları *Mathematica* adlı programlama dilinde hizmetimize sunmuş olmasından gelmektedir. Bölüm 3.3'de ele aldığımız kompozisyonel uygulamaya temel olan hücrel otomat datası *Mathematica*'da üretildikten sonra *Grace* adlı uygulamada müzikal olarak işlenmiştir. *Mathematica*'da (t)üretilen hücrel otomatları grafik çıktı olarak almak da mümkündür, ancak bizim yöntemimiz hücrel otomatı sıralı listeler olarak almak (yani her bir yeni nesil, bir liste olarak) ve bunları *Grace* (<http://commonmusic.sourceforge.net/>) adlı uygulamayı kullanarak *Scheme* dilinde işlemek biçiminde oldu. *Grace*'de işlenmiş algoritmik datayı MIDI olarak alabileceğimiz gibi *Fonus* (<http://fonus.sourceforge.net/>) adlı arabirimi kullanarak partiyon olarak almak da mümkündür. Bu çalışmadaki partiyonlar *Fonus* kullanılarak yazılmıştır. Kullanılan zamansal stratejinin dışında, farklı karakterlerdeki otomatların seri halinde, tek bir sistem içinde türetilmesi için de *Mathematica* dili kullanıldı.

Biyolog S. Kauffman tarafından geliştirilen RBA sisteminin kompozisyonel olanakları araştırıldı (Bkz. Bölüm 4.2 ve 4.3) . Her bir hücre bir fonksiyona atanarak, aktifleştiginde ilgili fonksiyona atanan değer güncellendi. Dolayısıyla her

bir hücrenin kendi belleği/süreci oluşturuldu. RBA, doğada düzenin kendiliğinden oluşumunu soyutlayan bir sistemdir. Spesifik olarak, gen içi düzenleyici sistemleri modellemekte kullanılmaktadır. Sembolik mantığın kurucusu olan George Boole'nin adından gelmektedir bu sistemin adı. RBA kökenini, Alan Turing ve von Neumann'ın otomat kuramından ve Ross Ashby'nin dinamik sistemler ve sibernetik çalışmalarından almaktadır (Wuensche 1997, 2). “RBA, komşuluk ilişkilerine dayanmayan ve heterojen kurallarıyla biyolojik modellere, hücrel otomatlar ise mütemadi mekan (*continuous space*) ve evrensel özellikleri ile fiziksel modellemelere temel oluşturmaktadır” (Wuensche 1997, 4). RBA'nın hücrel otomatlardan farklı komşuluk ilişkileri yerine rastlantısal olarak belirlenen ağ ilişkilerine dayanmasıdır; sistemin güncellenmesi, her bir hücre için gene rastlantısal olarak belirlenen Boole operatörünün (ve [*and*], ve ya [*or*], değil [*not*]), hücrenin ağda ilişkide bulunduğu diğer hücrelerin Boole değerlerine (doğru [*true*], yanlış [*false*]) uygulanması ile gerçekleştirilir. Eşgüdümsel işlem özelliği, sistem içinde makro bir örgütlenme meydana çıkarmaktadır. RBN sistemini çalıştıran münferit yazılımlar mevcuttur (*BooleanNet, dlab*). Ancak bu yazılımlar çeşitli ağların davranışsal haritasını çıkarmak amacıyla kullanılmaktadır. Müzik besteleme ve ses sentezlemede kullanılmak üzere RBN modelini kullanabilmek için *Scheme* ve *Pure Data* dillerinde bir dizi program yazdık. *Scheme* programının (t)ürettiği data, OSC (*Open Sound Control*) protokolüyle *Pure Data*'ya iletilmektedir. *Pure Data*'da yazdığımız çeşitli fonksiyonlar bu ağın davranışlarıyla tetiklenerek müzikal doku ve zamanın kurulumunu sağlanmaktadır. Hücrel otomatlar yerine RBN'nin çalışmamızda merkezi bir yer işgal etmesinde, lokal olmayan ilişkilerin müzikal olarak daha olumlu sonuçlar doğuracağı varsayımı vardır. Bu varsayım göz ve kulak duyuları arasındaki temel farka da dayanmaktadır: göz tek bir noktaya odaklanırken, duyumda tek bir odak yoktur, geniş bir aralık içinde gerçekleşen sessel olayları aynı anda algılayabiliriz. Dolayısıyla müzik ve ses tasarımında birbirinden farklı katmanlar arasında ağ yaratarak, bütünlüklü bir ses bünye kurmak estetik açıdan daha tatmin edici sonuçlar doğurmaktadır.

Zamansallık ve kendiliğinden değişim amacıyla geliştirdiğimiz üçüncü teknik ise dinamik Markov zincirleri oldu (Bkz. Bölüm 2.4). Markov zincirleri belirli bir olasılık tablosuna dayanır ve sistem bu tablodaki elemanların belli olasılıklarla

biribiri peşi sıra gelişini tayin eder. Markov zincirlerini dinamik hale getirmek için iki ayrı olasılık tablosu kullanıldı. Müzikal zaman ise, bir olasılık tablosunun diğerine tetrici olarak evrimleştirilmesiyle sağlandı. Yani tablodaki elemanların birbirini takip etme olasılığı, artan zamanın bir fonksiyonu olarak değiştirildi.

Ayrı bölümlerde ele aldığımız üç ayrı teknikte esas olan bu sistemlerdeki kendiliğinden değişim kapasiteleri idi. Kendiliğinden değişen bu sistemlerin, matematiksel idealizasyondan müzikal zamana aktarımında aynı kendiliğindenliğin müzikte nasıl temsil edilebileceği araştırılmıştır.

## **1. Bölüm: Tarihsel Bağlamı içinde Algoritmik Kompozisyon**

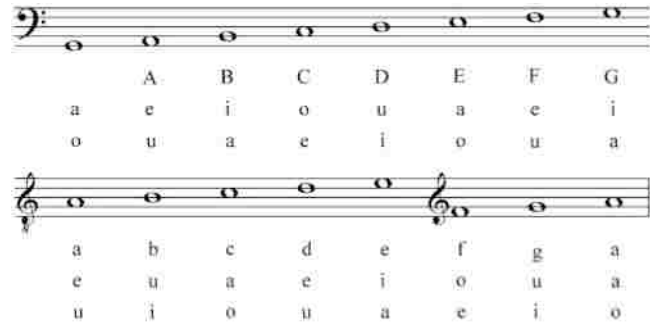
Tez çalışmamızın odağında, bilgisayar uygulamalarıyla olanaklı hale gelebilmiş araştırma yöntemleri ve bunlardan elde edilen bulgular vardır. Ancak araçlar ve bunların doğurduğu sonuçlar değişse de, algoritmik uygulamalar, düşünce ve sanatta, tarih boyunca kendini göstermiştir.

### **1.1 XX Yüzyıl Öncesinde Algoritmik Kompozisyon**

İnsanlık ilk çağlardan bu yana, otomatlara müzik yaptırmaya çalışmıştır. Bu uğraş, teknolojideki gelişmeler ve müziğin formalizasyonu ile bağlantılı olarak farklı çağlarda farklı biçimlerle karşımıza çıkmaktadır. Bilinen en eski müzik otomati Yunan Aeolian Harpıdır. Telleri rüzgar enerjisi ile titreşime giren bu çalgı Hint ve Çin kültüründe de karşımıza çıkmaktadır. Ktesibius'un 3. yüzyılda icat ettiği Hydraulis adlı org, suyun dinamik enerjisini hava basıncına dönüştürerek orgun borularına yönlendiren ilk mekanizma olarak tarihsel öneme sahiptir. Hydraulis, müziğin üretilmesi için kullanıcıya gereksinim duymakta idi. Kendi kendine çalan komplike çalgıların icadı 9. yüzyılda İslam uygarlığında karşımıza çıkmaktadır. El-Cezeri ve Banu Musa kardeşler çeşitli robotik sistemler üreten öncü mühendislerdir. Banu Musa'nın otomatik flütü siberetik ve robotik biliminde bir kilometre taşıdır. Flüt, buharla çalışmaktaydı ve kullanıcı, flütün üreteceği örüntüleri ayarlayabilmekteydi.

Batı müzik tarihine baktığımızda, MS. 1000 yıllarında, sembolik düzeyde, algoritmik bestelemeye dair ilk yaklaşımları görmekteyiz. Guido d'Arezzo (yaklaşık olarak 991 – 1031) müzik yazısının ve solfej pratiğinin gelişmesinde çok önemli katkılara sahiptir. Ortaçağın bu en önemli müzik kuramcısının “Micrologus” adlı eseri tek ve çok sesli şarkı söyleme geleneğine dair yazılmış en eski kaynaklardan biridir. Bu eserin 15. ve 17. bölümlerinde, metin kullanılarak otomatik ezgi üretilmesi betimlenmektedir (Sadie 1995). Buna göre bir dizinin harfleri, heceleri ve

kelimeleri perdeye (*pitch*) ve ezgisel cümlelere haritalanırken, *neume* grupları durgularla (*caesurae*) birbirinden ayrılmaktadır.(Bkz Şekil 1.1). *Neume*, tek hecede gruplanan notalardır. Durgu, *neume* grupları içinde nefes boşluğuna denk düşmektedir ve daha küçük gruplarda es ve uzayan nota işlevini alabilmektedir. Metindeki sesli harfler, şekil 1'de gösterildiği gibi çeşitli perdelere haritalanabilmektedir. Bu örneğin tarihsel önemi, farklı disiplin ve malzemeler arasında kurulan parametrik ilişkiden gelmektedir. Guido d'Arezzo tarafından erken bir tarihte kuramlaştırılan harflerin notalara eşlenme tekniği daha sonraki yüzyıllarda bir çok yapıtta karşımıza çıkmaktadır. Klasik dönemden bu uygulamanın en bilinen örneği Robert Schumann'ın *Abegg* ve *Carnaval* adlı eserleridir. Schumann'ın bu eseri *Abegg* adlı arkadaşına ithafen yazdığı varsayılmaktadır. Bu eserde tematik ekseni A/la, B/Si bemol, E/mi, G/sol notaları oluşturmaktadır. *Carnaval* adlı eserinde ise Schumann birden fazla isim/karakter kullanmıştır.



**Şekil 1.1 Sesli harflerin Guido d'Arezzo tarafından notalara eşlenmesi**  
(Nierhaus 2009, 23)

12. yüzyıldan itibaren batı çoksesli müzik geleneğinde motet en yaygın form haline gelmiştir. Bu formu geliştiren başlıca besteciler arasında Philippe de Vitry (1291 – 1361), Guillaume de Machaut (1300 – 1377), Guillaume Dufay (1400 – 1474), ve Josquin Desprez (1440 – 1521) gibi isimleri sayabiliriz. Motet formu en olgun haline, Giovanni Pierluigi Palestrina (1525 – 1594) ve Orlando di Lassus (1532 – 1594) gibi bestecilerle erişmiştir. 17. yüzyılın başlangıcından itibaren motet formu

eski önemini yitirmeye başlamış olsa da, J.S. Bach, Schütz ve diğer besteciler 18. yüzyıl ortasına kadar bu formda eserler yazmışlardır. Motet formunun algoritmik besteleme bağlamında önemi, *isorhythm* tekniğinin bu eserlerde sıkça kullanılmış olmasından gelmektedir. *Isorhythm* tekniği Philippe de Vitry tarafından yaratılmış olup en olgun haline Guillaume de Machaut'un eserlerinde ulaşmıştır (Nierhaus 2009, s. 23). Bu teknikte önceden belirlenmiş ezgisel (*color*) ve ritmik (*talea*) malzemeler birbirleriyle çeşitli kombinasyonlara sokulmaktadır. Ritm ve ezgi öğelerinin bu şekilde soyutlanması ve kombinasyonel uygulamalarla müzikal yapının oluşturulması 20. yüzyıl diziselciliğinde karşılaştığımız uygulamalarla büyük koşutluklar içermektedir.

Barok çağın önemli müzik kuramcılarında olan Athanasius Kircher *Musurgia Universalis* adlı eserinde *Arca Musarithmica* ve *Organum Mathematicum* isimli algoritmik uygulamaları geliştirmiştir. *Arca Musarithmica* bir kutudur. Bu kutunun içinde, bir dizi, üzerinde tablolar bulunan tahta şeritler bulunur. Tablodaki iletiler, kutunun mekanik olarak nasıl ayarlanacağını belirtir. Bu yönergeler uyarınca, kutudan belirli kollar ve teller çekilir ve sonunda müzik parçası ortaya çıkar.

18. yüzyılda gene kombinasyonel olasılıklara dayanan ancak kullanıcının hesaplamasını/karar vermesini gerektirmiyen bir oyun geniş bir popülerlik kazanmıştır. Bu oyun müzikal bir eğlence aracı olarak müzik eğitimi almamış insanların değişik stillerde parçalar üretmesini olası kılmıştır. Kullanılan yöntemde zamansal birim ölçüdür ve oyunu oynayan kişi zar atarak tablodan ölçüleri seçer ve müzik parçasını oluşturur. Her zamansal konum için (ki bu genellikle ölçü sırasındır), çeşitli sayıda müzikal seçenekler bulunmaktadır. Örneğin 1. ölçü için Tonik akoru notaları ve 2. ölçü için alt-dominant akoru notaları gibi. Bu sayede müzikal tutarlılık sağlanmaktadır. Bu işlem gerekli ölçü adedine ulaşıncaya kadar tekrarlanır. Her ölçü için seçilebilecek aday miktarı genellikle 11'dir. Ki bu, altı yüzlü iki zar kullanılmasından kaynaklanmaktadır. Müzikal zar oyununun oynanması oldukça

basittir, ancak oyunun tasarımında armonik/zamansal tutarlılığın sağlanması ve parti hareketlerinin birbirini tamamlaması göz önüne alınması gereken ince noktalardır. İlk zar oyunu Johann Philipp Kirnberger tarafından 1757 yılında “*Der allezeit fertige Menuetten- und Polonaisenkomponist*” adıyla yayınlanmıştır (Nierhaus 2009, 36). 1812 yılına kadar Kirnberger bu türde en az 20 oyun daha üretmiştir. Ünlü besteciler tarafından üretilen müzikal zar oyunları arasında C. P. E. Bach'a ait 1758 tarihli “*Einfall einen doppelten Contrapunkt in de Octave von sechs Tacten zu machen ohne die Regeln davon zu wissen*” ya da Maximilian Stadler'e ait 1793 tarihli “*Table pour composer des minuets et des Trios `a l'infinie; avec deux dez `a jouer*” sayılabilir. Haydn ve Mozart adıyla da bir dizi müzikal zar oyunu yayınlanmışsa da bu oyunların bizzat bu isimler tarafından üretildiği kanıtlanmamıştır (Glaser 1971).

## **1.2 XX. Yüzyılda Algoritmik Kompozisyon**

XX. yüzyıl başında yaşanan tonal dilden kopuş beraberinde müzik bestelemeye yeni dinamikler getirmiştir. Arnold Schönberg'in geliştirdiği 12 ton sistemi müzik bestelemeye esas etkisini 1950 sonrası müzikte göstermiştir. 1950 sonrası müzik kompozisyonunda, tasarıma dair her bir parametrik öge diziselleştirilmiş, dolayısıyla sayısallaştırılmıştır. Bu genişletilmiş dizisellik, 12 ton sisteminin doğal gelişimi olarak görülebilir. Ruth Crawford Seeger'in 1931 yılında bestelediği yaylı dörtlü genişletilmiş diziselciğin en erken örneklerindedir. Kıta Avrupası'nda genişletilmiş diziselciliğin yaygınlaşması ise Olivier Messiaen'in 1949 yılında yazdığı “*Mode de valeurs et d'intensities*” adlı yapıtıyla olmuştur. Bu yapıtta nota yükseklikleri (pitch), uzunlukları, gürlükleri ve atakları diziselleştirilmiştir. Müzik bestelemeye yaşanan bu paradigmatik kırılmanın, 1950 yıllarda gelişen informatik ve sibernetik disiplinleriyle etkileşmesi fazla zaman almamıştır.

En eski otomatik kompozisyon örneklerinden biri “Tin Pan Alley” ezgileri bestelemek için yazılmış bir programdır; Martin Klein ve Douglas Bolitho tarafından DATATRON bilgisayarı kullanılarak yaratılmıştır. "rastlantısal olarak 1000 ayrı rakam tayin edilir. Bu rakamlar, diyatonik dizideki bir notaya ve izin verilen iki arızalı notaya denk düşmektedir. Daha sonra notalar rastlantısal olarak seçilir ve her biri süreç içinde ezgiye uygunluğuna göre seçilir.” (Anon 1956). Bu programın bir çıktısı aşağıda nota örneğini verdiğimiz "Push Button Bertha" adlı ezgidir. İlk defa 15 haziran 1956'da dinleyici ile buluşmuştur.

**"PUSH BUTTON BERTHA"**

LYRIC BY  
**JACK OWENS**  
1956

MUSIC BY  
**DATATRON**  
MATHEMATICS  
DR. MARTIN KLEIN  
DR. DOUGLAS BOLITHO

MODERN BIGHT BANK

2 2 2 2 2 2 2 2 2 2  
0 0 0 0 7 1 9 2 7  
RANDOM NUMBERS  
USED FOR MUSIC

THE'S PUSH - BUT-TON BER-THA - SWEET MA-CHINE, WHAT A QUEEN  
CAL-CU-LAT-IN' DAL-DI-TA-TIN' CHICK - WITH A  
CHICK - MY PUSH - BUT-TON BER-THA - NOT TOO LAESE, WHAT A CHARGE  
E-LEK-TRON-IC SU-PER-SON-IC FRIEND - THE  
END - SHE SAYS OF-ER - A - TIN' - WATCH HER ROCK AND ROLL -  
BER-THA'S NO- DE-MAND-ING - NEV-ER WANTS YOUR DUGH -  
COOL AND CAL-CU-LAT-IN' - THIS GAL HAS NO HEART OR SOUL - AHER  
AL-WAYS ON - DICE - STAND-ING - JUST FUP A SWITCH AND SHE'LL GO -  
PUSH - BUT-TON BER-THA - AU-TO - MA-TIC DI-VINE  
NOW HEAR THIS - SHE CAN'T KISS  
BY THE LIGHT BULB AND BOURG ROHT SUES MINE - ALL THINE -  
YOU WIGHT ON - MAKES HER LOV-AL DREAM - MA-CHINE -  
CARE BE SUITS COMPANY ON - CARELESS JAILING DE-FOULS SANDERS-KOYIC

Şekil 1.2 “Push Button Bertha”, Dr. Martin Klein, Dr. Douglas Bolitho (Ames 1987, 169)

## 1.2.1 1957'den 1966'ya Urbana Okulu

1955'te Lejaren Hiller, ILLIAC I adlı bilgisayarla University of Illinois at Urbana-Champaign'de bilgisayarla müzik besteleme çalışmalarını başlatmıştır. Sahip oldukları tarihsel önem ve estetik ve teknik alandaki katkıları bu okulu bilgisayar müziğinde bir kilometre taşı haline getirmiştir.

### 1.2.1.1 Hiller ve Isaacson

Klein ve Bolitho'nun rastlatsal datanın çeşitli kriterlere göre elenmesi tekniği, zamanında başarılı bir teknik olarak görülmekte idi. Bunlardan bağımsız olarak aynı yöntem, University of Illinois at Urbana-Champaign'de Lejaren Hiller ve Leonard Isaacson tarafından da geliştirilmiştir. Amerika'daki en eski elektronik müzik stüdyolarından biri bu okulda kurulmuştur ve Robert Baker, James Tenney, Herbert Brün ve John Myhill gibi bir çok araştırmacı/besteciyi burada buluşturmuştur. Hiller ve Isaacson'un çalışmaları, *Experimental Music* adlı kitapta açılmış olduğu biçimiyle dört merhalede özetlenebilir.

*Deney 1:* tek sesli, iki sesli ve dört sesli yazı. *First-species* (bir notaya karşı bir nota) kontrpuan kurallarından seçilen sınırlı sayıda kural, müzikal çıktıyı tayin etmektedir. Kullanılan *cantus firmus*'lar 3 ila 12 nota uzunluğundadır. Kontrpuan biçimi *first-species* olduğu için, tek sesli, iki sesli ve dört sesli yazılar da 3 ila 12 nota uzunluğundadır.

*Deney 2:* Kontrpuan kuralları ile rastlantısal müzik sıralı olarak kurgulanmıştır.

- her elemanın aynı olasılığa sahip olduğu (*white-noise*) rastlantısal müzik
- Bitişik-atlamalı nota hareketi; aynı türden hareket birden fazla kere sıralı olarak tekrarlanamaz

- Do majör akoru; *cantus firmus do*'da başlar ve biter; kadans da *do* tonundadır; *si-fa* artık dörtlü aralığı yalnız VII<sub>6</sub> akorunda ortaya çıkar ve bu aralık *do-mi*'ye çözülür.
- Oktav-sınırı kuralı
- 6/4 akoru haricinde konsonant armoniler
- Disonant ezgisel aralıklar (ikililer, yedililer ve artık dörtlü) yasaktır
- Paralel unison, oktav ve beşli hareketi yapılmaz
- Paralel dörtlüler ve 6/4 akoru yoktur; yüksek seslerde benzer ezgisel oluşumlar tekrarlanmaz

### *Deney 3: Deneysel müzik*

Ritm, dinamikler, yönergeler ve yalın kromatik yazı

- Yalın ritm, dinamikler ve yönergeleri uygulayan kod
- Rastlantısal kromatik müzik
- Rastlantısal kromatik müzik, modifiye edilmiş ritimler, dinamikler ve yönergeleri uygulayan kod ile kombine edilmiştir
- Kromatik müzik, oktav-sınırı kuralı, artık dörtlü çözülme kuralı ve bitişik-atlamalı hareket kuralı ile kontrol edilmiştir
- Denetimli kromatik müzik, modifiye ritimler, dinamikler ve yönergeleri uygulayan kod ile kombine edilmiştir
- Aralık serisi, ton serisi, ve sınırlanmış ton serisi

### *Deney 4: Markov zinciri müziği*

- 0-adımlı armonik olasılık fonksiyonunun çeşitlemeleri: tamamen tonal olarak sınırlanmış seçilimden, ortalama olarak derecelendirilmiş serbest seçilime
- 0-adımlı armonik olasılık fonksiyonunun çeşitlemeleri: tamamen rastlantısal seçilimden, ortalama olarak derecelendirilmiş serbest seçilime
- 0-adımlı, armonik ve yakın olasılıkları seçen bir fonksiyon; bu iki fonksiyon

birlikte kullanılmıştır

- 1-adımlı, armonik ve yakın olasılıkları seçen bir fonksiyon; bu iki fonksiyon birlikte kullanılmıştır
- 0-adımlı, armonik ve yakın olasılıkları seçen fonksiyonun, güçlü ve zayıf zamanlara bölüştürülmesi, iki ayrı kombinasyonla kullanılması
- 1-adımlı, armonik ve yakın olasılıkları seçen fonksiyonun, güçlü ve zayıf zamanlara bölüştürülmesi, iki ayrı kombinasyonla kullanılması
- *i*-adımlı armonik fonksiyonun güçlü zamanda kullanılması, 1-adımlı yakın olasılıkları seçen fonksiyonun zayıf zamanda kullanılması; genişletilmiş kadans; yalın ve kapalı form (Hiller & Isaacson 1959, 155).

Yukarda detaylandırılan deneylerde, Hiller ve Isaacson temel olarak iki yaklaşımı kullanmaktadır:

1. Çeşitli kurallar ile elenen rastlantısal seçimler (*Deney 1-3*); bu yaklaşım Klein ve Bolitho'nun yönteminde de vardır
2. Markov zincirleri kullanılmıştır; gene rastlantısal seçimler sözkonusudur, ancak önceki adımlar olasılık kümesini tayin etmektedir (*Deney 4*).



Şekil 1.3 *Illiad Suite* (1957), Deney 3'ten bir alıntı: 55. - 72. ölçülerde 'yalın ritim, dinamikler ve yönergeleri uygulayan kod'; 73. - 80. ölçülerde 'rastlantsal kromatik müzik'; 81. - 100. ölçülerde 'rastlantsal kromatik müzik, modifiye edilmiş ritimler, dinamikler ve yönergeleri uygulayan kod ile kombine edilmiş' olarak, uygulanmıştır. (Ames 1987, 171)

Hiller'ın daha önceki çalışma arkadaşı Robert Baker MUSICOMP ( *MUSIC Simulator Interpreter for COMpositional Procedures* ) adlı kompozisyon platformunu programlamıştır. MUSICOMP, yeni kompozisyon araçlarının geliştirilmesi için bir alt yapı oluşturmuştur ve farklı bestecilerin kendi amaçları için kullanabileceği bir kod kaynağı oluşturmuştur. Hiller ve Baker MUSICOMP'u kullanarak 1962'de *Computer Cantata* adlı eseri bestelemişlerdir. Bu müzik tamamen, Pierre Boulez'in iki piyano için yazdığı *Structures* adlı eserinde kullandığı dizisel tekniklerle ve *Illiac Suite*'de kullanılan rastlantısal tekniklerle oluşturulmuştur.

### 1.2.1.2 James Tenney

Urbana okulunun bir diğer önemli üyesi de James Tenney'dir. Master tezi olarak yazdığı *Meta-Hodos* adlı eseri XX.yüzyıl müziğinde form konusuna en özgün yaklaşımlardan birisini oluşturmaktadır (Tenney 1988). Tenney Urbana'daki master eğitiminde Hiller ile bilgisayar kompozisyonu üzerine çalışmalar yapmıştır. 1961 yılında *Bell Telephone Laboratories*'e davet edilmiştir. Bu dönemde Max Mathews, sayısal ses sentezleme programını (Music-N serisi) geliştirmekte idi. Tenney, aynı dönemde Xenakis'in geliştirmekte olduğu, "*stochastic music program*"'ından habersiz olarak benzer bir çalışma içindeydi. Ancak gene de etkilendiği kaynak, Xenakis'in kendi betimlemesi ile *Pithoprakta* ve *Achorripsis* adlı müziklerde kullanılan istatistikî tekniklerdi. (Xenakis 1992, 12-42). Tenney'in geliştirdiği yöntem, Xenakis'inkinden şu bakımlardan ayrılmaktadır: 1) bölümlendirilmiş çizgisel graflar kompozisyonun zaman içinde nasıl evrileceğini güdülemekte idi, ve 2) Max Wertheimer'in Gestalt psikolojisinden esinlenilerek hiyerarşik bir prosedür işletilmektedir. Tenney'in kompozisyonlarının çoğu datayı sayılar olarak çıkartmakta ve bu sayılar Mathews'in MUSIC4 programı kullanılarak sayısal sese dönüştürülmekte idi. 1963'te yazılan *Stochastic String Quartet* ise doğrudan görsel partiyon olarak üretilmiştir. Bu eser 3 bölümden oluşmaktadır, bunlar gösterilen tempo yönergeleri doğrultusunda 50, 100 ve 40 saniye sürmektedir. Bölümler,

Tenney'in "*clang*" (Tenney 1988, 87) olarak adlandırdığı daha alt düzey birimlerden oluşmaktadır. *Clang* tınısal bütünlük sergileyen birincil oluşumlardır. Bunlar notalardan oluşmaktadır. Her bölümün karakteri, müzikal özelliklerden oluşan graf ağıyla güdümlenmektedir. Bu müzikal özellikler: *clang* süresi, ataklar arasındaki ortalama zamansal aralıklar, perde yükseklikleri, dinamikler ve vibrato, tremolo, '*spectrum*' (sul tasto, ord., sul pont.), '*envelope*' (pizzicato, arco-staccato, arco-legato, arco-marcato, arco-sforzando) gibi çeşitli tınısal özelliklerdir. Graf ağında, birbirini takip eden her düğüm noktası, bir veya daha fazla rastlantısal sayı üreticine yönlendirilen ortalama bir sayısal değere sahiptir, ortalama değer esas alınarak tayin edilen bu rastlantısal değerler spesifik bir *clang*'ın özelliklerini belirler; ve bunun devamında, program bu *clang* özelliklerini ileriki rastlantısal sayı üreticilerini beslemek için kullanmaktadır, bu sayede bir notanın özellikleri tayin edilmektedir. Aşağıda örnek olarak *Stochastic String Quartet*'in girişi verilmiştir.

Bu eserin bir diğer önemli özelliği de ritmik olarak nasıl rafine edildiği ve zamansal akışkanlık kazandırıldığıdır. Her bir parti için kullanılacak tupletin değeri ve uzunluğu rastlantısal olarak tayin edilir ve bu tupletin içi gene rastlantısal olarak notalandırılır ve ya boş bırakılır (es).

## STOCHASTIC STRING QUARTET

James Tenney  
7/20/63

The image displays a musical score for a string quartet, titled "Stochastic String Quartet" by James Tenney, dated 7/20/63. The score is arranged in four systems, each containing four staves for the instruments: Violin I (Vi. I), Violin II (Vi. II), Viola (Va.), and Cello/Double Bass (Vc.). The tempo is marked as  $\text{♩} = 60$ . The score is written in 4/4 time and features a complex, rhythmic structure with many notes beamed together, creating a dense texture. The notation includes various dynamics such as  $mf$  and  $pp$ , and includes performance instructions like "pp" and "mf" with dashed lines. The score is divided into four systems, each starting with a measure number in a box: 5, 10, 15, and 20. The first system (measures 1-4) shows the initial rhythmic patterns. The second system (measures 5-8) continues the complex rhythmic development. The third system (measures 9-12) shows further rhythmic complexity. The fourth system (measures 13-15) concludes the section with a final rhythmic pattern.

Şekil 1.4 James Tenney, "Stochastic String Quartet" 1963 1.-15. ölçüler. (Ames 1987, 172)

### 1.2.1.3 Herbert Brün

Herbert Brün, 1964 yılında, *Sonoriferous Loops* ve 1966'da *Non-Sequitur VI* adlı eserlerini MUSICOMP sistemini kullanarak yazmıştır. Brün'ün *Sonoriferous Loops* adlı müziği Edgard Varese'in *Deserts* adlı eserindeki modeli temel almaktadır; ancak burada dört, dijital olarak sentezlenmiş ses interlödü ile beş kısa, icra grubunun çaldığı bölümler, birbiri içine girmektedir. İcra grubunda beş çalgı vardır: flüt, trompet, kontrabas, ksilofon/marimba ve perdesiz vurma çalgılar - bunların arasına giren interlödlerde üç sentetik ses vardır. Herbert Brün'ün kendi yazdığı teknik notlara göre program zamansal hareketinde partiyonu 'tarar' ve ilgili partinin bir nota ya da es alıp almadığını sınar. Eğer ilgili partiye bir nota ya da es tayin edilmediyse program aşağıdaki adımları takip eder:

1. *Ritm*. Program rastlantısal olarak *denominator* ve *numerator*'ı 16, 6, 5 ve 4 değerleri içinden seçmektedir; bunlar ortak çarpanı oldukları 240'a uygulanarak süreyi oluştururlar.
2. *Es/Nota Seçimi*. Program devamında es mi vereceğini yoksa nota mı çalacağını tayin eder. Partilerin görece işlerliği 'es/nota olasılıkları'na göre tayin edilmektedir; örneğin açılış ölçüsünde, flüt, zamanın, % 26, trompet % 32, kontrabas % 50, ksilofon/marimba % 68, ve perdesiz vurmaları % 74'ünde susmaktadır.
3. *Perde*. Kromatik bir derece ve oktav tayin edilir. Kromatik derecenin seçiliminde, 12 elemanlı bir listeden bir üye rastlantısal olarak seçilir. Her adımda liste baştan alınır, zira notalar her parti için aynı anda tayin edildiği için bu özdeş elemanların farklı partilerde yinelenmesini engellemektedir. Oktavlar tamamen rastlantısal olarak tayin edilir.

Ritm ve perdeler program tarafından belirlendikten sonra, Brün manuel olarak tempo, dinamikler ve çalgısal teknikleri (pizz., kapalı, stacc., vb) yazmaktadır.

SONORIFEROUS LOOPS  
Herbert Brün  
Opus 32 - 1964

The musical score is presented in two systems. The first system contains measures 0-3, and the second system contains measures 4-7. The instruments are: Flute, Trumpet in Bb, Double Bass, Synthesizer, and Horns. The score is marked with a tempo of  $J = 164$  and includes dynamic markings such as  $ff$  and  $mf$ . The notation includes various rhythmic values, including eighth and sixteenth notes, and rests. The score is a complex, multi-layered composition.

Şekil 1.5 Herbert Brün, *Sonoriferous Loops*, 1964, 0-9. ölçüler (Ames 1987, 173)

1951 yılında John Cage “*Music of Changes*” adlı eserini yayınlamıştır. Bu eserde her bir parametreye atanacak değerlerin belirlenmesi şans operasyonu ile sağlanmaktadır. Parametreler; tempo, dinamikler, tınılar, esler, süreler ve üst üste bindirmelerdir (*superimpositions*). Şans operasyonu ise tamamen *I Ching* adlı antik çin fal kitabına dayanmaktadır.

Aynı yıllarda Joseph Schillinger'de kendi kuramını geliştirmekteydi. John Cage'in eserinde ayrıntılandırılan parametrik süreçlerden farklı olarak, Schillinger soyut matematiksel yapılar üzerine çalışmıştır (Schillinger 1976). Bu matematiksel yapıları armoni, ezgi ve ritmik seçimlere uygulamıştır.

### **1.2.2 Avrupa Algoritmik Müziği: 1960-1970**

Pierre Barbaud, müzik besteleme sürecinde, bilgisayardan faydalanan ilk avrupalı bestecilerdendir. 1960'lı yılların başında, Paris'te, bilgisayar yardımıyla bestelediği algoritmik müzikleri dinleyici ile buluşturmaya başlamıştı. Bu eserlerinde kullandığı teknikleri, *Initiation a la composition musicale automatique* adlı kitabında yazmıştır (Barbaud 1966); bunlar, geleneksel armoniye uygulanan permütasyonlar, dizisel (12 ton) teknikler ve rastlantısal seçim teknikleridir.

Urbana okulu ve Avrupalı algoritmik besteciler arasındaki temel farkı Ames şu şekilde koymaktadır:

Kompozisyonel olarak, Urbana okulu stilistik kurallar biçiminde *local* ilişkileri ayrıntılandırırken, Avrupalılar müzikal pasajların *global* nitelikleri üzerine yoğunlaşmışlardır ve bunları istatistiki yöntemlerle genelleştirmişlerdir. Teknik olarak, bir çok beste üreten programları yayınlayanlar, Iannis Xenakis (Xenakis 1992) ve Gottfried Michael Koenig (Koenig 1969, 1970, 1979) gibi Avrupalılardır, öte yandan Urbana okulunun çalışmaları tek atışlık ve spesifik bir kompozisyon projesine yönelikti. (Ames 1987, 173)

Xenakis'in şu sözleri yukarda zikredilen farka dair şairane bir yorumdur:

Bilgisayar yardımıyla, besteci bir çeşit kaptan haline gelmiştir: düğmelere basmaktadır, koordinatları girmektedir, gemisini ses uzayında, sonik gökadar ve galaksiler boyunca adeta bir düşteymiş gibi yüzdürmektedir. Bu sayede koltuğunda oturarak kolaylıkla keşfe çıkabilmektedir. (Xenakis 1992, 144)

### 1.2.2.1 G. M. Koenig

Algoritmik müzik kompozisyonuna katkısı ihmal edilmeyecek araştırmacılardan biri de Koenig'dir. Kendisi, Utrecht'teki *Institute of Sonology*'de hocalık ve yöneticilik görevlerinde bulunmuştur. Koenig'in algoritmik kompozisyon çalışmaları 1964'te PROJECT1 ile başlamıştır (<http://www.koenigproject.nl/pr1a.htm>). Bu tarihten itibaren bestelediği bir çok eserinde (*Project 1*, 14 çalgı için *Version 1*) bu programı kullanmıştır. Koenig, dizisel kompozisyon geleneğinden gelmektedir ancak, PROJECT1'dan itibaren dizi ve permutasyonlarının yararsızlığına ve sonuçta istatistiki düzenlemenin kompozisyonu belirlediğine kanaat getirmiştir. (Koenig, 1970a, 33)

1969'da geliştirdiği PROJECT2, daha geniş bir palette istatistiki düzenlemeleri kullanma olanağı sunmakta ve bunların değişik kombinasyonlarla birbirlerine bağlanabilmesine olanak sağlamaktadır. PROJECT2'de kullanıcının yönetimindeki başlıca parametreler şunlardır:

*armoni*, ya da kromatik dizinin kullanılma oranı

*register*

*zamansal kayma*, ya da sıralı ataklar arasındaki süre

*süre*

*dinamikler*.

*Armoni* parametresi, programda tanımlandığı biçimiyle dizi (*row*), akor, ve aralık prensiplerine göre belirlenmektedir. Diğer parametreler için kullanıcı ya sabit bir değer girmelidir ya da bir dizi seçenek girmelidir ve alt-işlemler olarak çalışan şu altı süreçten birini seçmelidir (*sequence*, rastlantı (*alea*), diziler, oran, grup ve meyil (*tendency*). Koenig, girilen belirli bir verinin farklı müzikal çıktıları için *variant* terimine vurgu yapmaktadır.

Yukarda sıralanan alt-işlemlerden 'diziler' ve 'rastlantı' arasındaki fark; 'diziler' seçildiğinde, seçilen bir eleman diğer tüm liste elemanları seçilmeden tekrar ortaya çıkamaz, Koenig bunu 12-ton sisteminin bir uzanımı olarak yorumlamaktadır. 'Alea' seçeneğinde ise böyle bir kısıt yoktur.

Koenig, "Composition Process" (Koenig 1978) adlı makalesinde, kompozisyon tekniğini dört maddede özetlemektedir.

*Enterpolasyon*: Yukardan-aşağı bir yaklaşımdır. İki değer arasındaki ara-değerleri ortaya koyar. Zaman eksenini, başlangıç değerinden amaç değerine giden bir süreç olarak kurular.

*Extrapolasyon*: Aşağıdan-yukarı bir yaklaşımdır. Enterpolasyonun tersine, bilinen değerler, yeni bir alana ölçeklendirilir ve bu değerlerden yeni değerler türetilir.

*kronolojik-çağrışımsal*: her müzikal elemanın "zamanda değiştirilemez bir noktaya sahip olması" anlamındadır. "Besteleme süreci, zaman eksenini boyunca açılmaktadır"

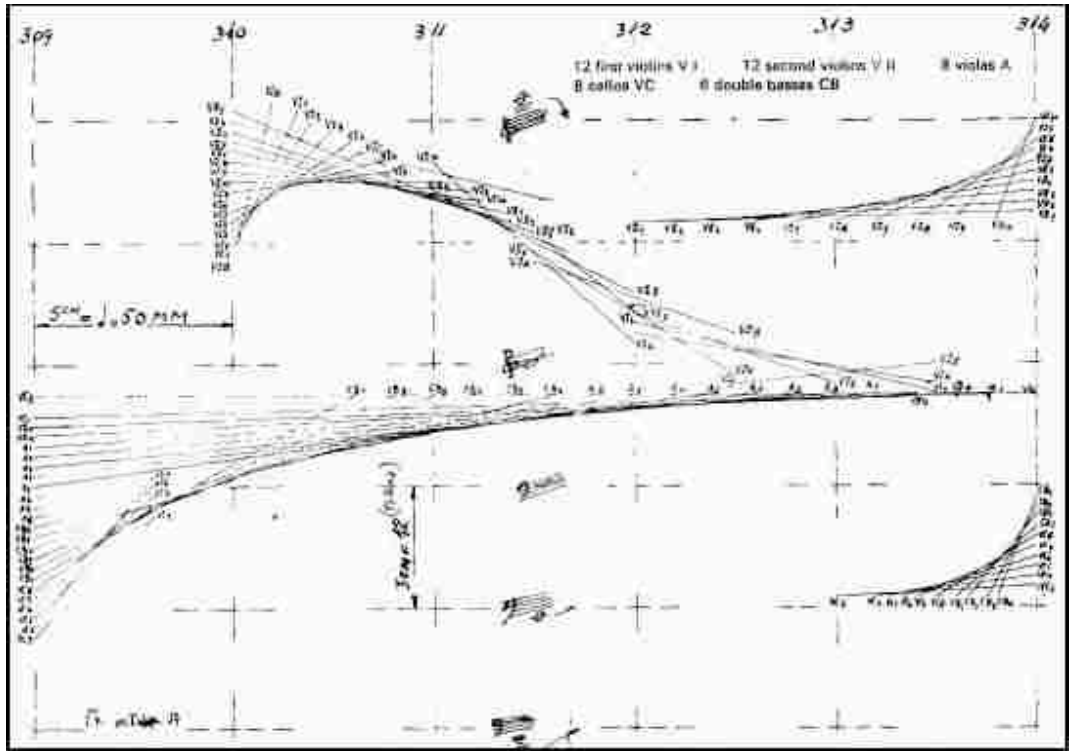
*blokların kompozisyonu*: Dördüncü madde, diğer yöntemlerin birleşimidir. Blokların her biri, birbirinden bağımsız olarak art-süremsel ya da eş-güdümsel olarak tasarlanmıştır.

Çalışmalarına Fransa'da başlayarak, algoritmik müzik kompozisyonunda yetkin ve teorik olarak kapsamlı ürünler veren Iannis Xenakis'i Avrupa algoritmik müziği alt

başlığında ele almak yerine, ayrıca bir başlıkta ele aldım. Xenakis'i Avrupa bağlamının dışında ele almamın esas nedeni, 1967-1972 yılları arasında Indiana Üniversitesi'nde görev almış olması değil, algoritmik müziğe yaptığı emsalsiz katkılarıdır.

### 1.3 Xenakis

20. yüzyılda matematiksel modelleri müzik kompozisyonunda en yetkin biçimde kullanan besteci Iannis Xenakis olmuştur. Xenakis'in algoritmik kompozisyona getirdiği başlıca matematiksel teknikler arasında Olasılık Kuramı, Markov Zincirleri, Grup Kuramı, *Sieve* Kuramını sayabiliriz. Xenakis'in olasılık kuramını kullanarak yazdığı ilk eser *Metastasis*'tir. Bu eserde başat müzikal unsur glissandolardır ve glissandolardırın hızı ve oranı bir dizi olasılık hesabı sonucu üretilmiş ve yaylı çalgı yazısına geçirilmiştir.



Şekil 1.6 Iannis Xenakis, *Metastasis*, 309. ve 314. ölçülerdeki Yaylı Glissandoları. (Xenakis 1992, 3)

### 1.3.1 Sieve

Xenakis, 1965'ten 1992'ye kadar yayınladığı bir çok yazıda *Sieve* kuramını ele almıştır. Bu kuramı sayı dizileri üretici olarak kullanmıştır. Bu modelde diziler, çeşitli başlangıç noktalarından çeşitli modlarla artan sayı dizilerine küme operatörlerinin uygulanmasıyla elde edilmektedir. Örneğin başlangıç noktası 2 ve modu 3 olan bir dizi [2, 5, 8, 11, 14, 17, 20, 23, 26, ∞] ile başlangıç noktası 3 ve modu 2 olan bir başka diziyi [3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, ∞] birleşim kümesi [∪] işlemine sokarsak elde edeceğimiz yeni dizi [2, 3, 5, 7, 8, 9, 11, 13, 14, 15, 17, 19, 20, 21, 23, 25, 26, 27, 29, ∞] olacaktır. Elde ettiğimiz bu birleşim dizisi içerisinde [1, 2, 2, 1] biçiminde bir döngü oluşmuştur. Bu döngü, sonuç dizisi elemanları arasındaki artış algoritmasını vermektedir, ve yani sonuç dizisi 6'lı bir mod üzerine kurulmuştur. Xenakis, bu dizileri ritmik hatları ve ses dizilerini tayin etmek için kullanmıştır. *Jonchaies* adlı orkestra müziğinde kullandığı ses dizisi 17'li bir döngü üzerine kurulmuştur (Harley 2004, 108). (Bkz. Şekil 1.7)

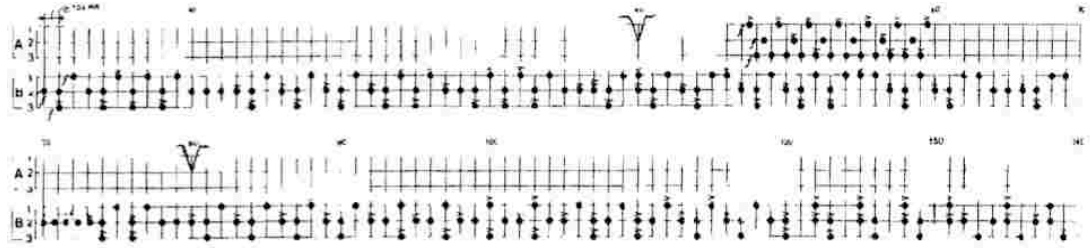


**Şekil 1.7 Iannis Xenakis'in *Jonchaies* adlı eserinin 1. kısmında kullandığı, yaylılardaki nota *sieve* dizisi. (Harley 2004, 109)**

17'li mod üzerine ses dizisi kurmak, kendini her oktavda (yani 12'li mod) yineleyen ses dizileri yerine oktav + tam dörtlü döngüsünde tekrarlayan bir dizi doğurmuştur.

Xenakis, Psappha vurma çalgılar eserinin girişinde 5 ve 8'li modlar üzerine kurulu

ve dolayısıyla 40'lı döngüye sahip bir sieve dizisi kullanmıştır. (Bkz. Şekil 1.8). Verilen şekilde B2 çalgısı sieve noktalarını vurgulamakta, B3 uzun zamansal aralıkları vurgulamakta ve B1 ise B2 ve B3'ün olmadığı noktaları doldurmaktadır (Ellen Rennie Flint'den alan: Harley, 2004, 95).



**Şekil 1.8** Xenakis, *Psappha*'nın açılış kısmındaki ritmik *sieve* dizisi (Harley 2004, 96)

### 1.3.2 Markov Zincirleri

Algorithmik bestelemeye, rastlantısallık ve belirlenmişlik özelliklerini içinde barındıran ve verimli sonuçlar doğuran bir diğer teknik de Markov Zincir'leridir. “Markov zincirleri, gelecekteki olası durumların bir ya da daha fazla geçmiş duruma bağlı olduğu, koşullu olasılık sistemleridir” (Miranda 2001). Örneğin A, B, C ve D olarak dört elemanımız olsun; bu elemanlar için birinden sonra hangi diğer elemanın gelme ihtimali olduğunu ve bu ihtimalin değerini bir Markov tablosu ile gösterelim.

	A	B	C	D
A	0.2	0.1	0.3	0.4
B	0.5	0.1	0.2	0.2
C	0.5	0.2	0.1	0.2
D	0.2	0.2	0.3	0.2

**Tablo 1.1**

Tablodaki deęerler, 0 olmayan ihtimal ve 1 tam ihtimal olarak ölçülandırılmıştır. Yani tabloya göre A'dan sonra A gelme ihtimali 0.2 yani % 20'dir. Rastlantısal süreç çalıştırıldığında, her bir yeni durum bu tablodaki olasılık deęerlerine göre tayin edilir.

Xenakis, 1958 yılında 9 yaylı çalgı için yazdığı *Analogique A* adlı eserinde Markov zincirlerini kullanmıştır. Markov tablosunda, olasılık deęeri alacak durumlar için farklı screen'ler tanımlamıştır. Screen, kompozisyonun zamansal birimidir. Her yeni zamansal birimde, tanımlı Markov tablosuna göre yeni bir screen seçilir. Screen'ler birbirinden, ses frekans aralığı, gürlük ve dinamikler, ve zamansal yoğunluk bakımlarından farklılık gösterir (Xenakis 1992, 98-103).

### 1.3.3 Hücresel Otomatlar

Xenakis'in algoritmik kompozisyona katkısı bağlamında en son ele alacağımız teknik ise Hücresel Otomatlar'dır (*Cellular Automatas*). (Bkz 3. Bölüm). “*Formalized Music*” adlı kitabının 1992 yılında İngilizce yayınlanan versiyonuna yazdığı önsözde şöyle demektedir:

“Tınının gizemine yaklaşım anlamında, geçmiş yıllarda yaptığım kompozisyonlarda kullandığım bir başka teknik ise hücresel otomatlar oldu...hücresel otomatlar armonik yazıda yararlı olabilmektedir ve orkestral çalgılarla yeni ve zengin tınısal bireşimler yapılabilmesini sağlamaktadır. Bunun örnekleri, benim *Ata* ve *Horos* adlı eserlerimde görülebilir.”

Xenakis'in hücresel otomatlar kullanımına dair ilk bulgular Peter Hoffmann tarafından “*Amalgam aus Kunst und Wissenschaft*” adlı çalışmasında verilmiştir:

“Horos'ta Xenakis tek boyutlu bir hücresel otomat kullanmıştır. Bu, akışkanların zamansal süreç içinde dağılımını takip edebilmemizi sağlayan özel bir otomata benzemektedir. Xenakis'in otomatında yer koordinatı nota yükseklikleridir. Otomat her zamansal adımında her bir notanın çalınıp çalınmayacağını, eğer nota çalınacaksa bakır üflemeliler mi, tahta üflemeliler mi yoksa yaylılarda mı çalınacağını belirlemektedir.” (Hoffmann 1994, 147)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	
bars 10	1											1												
	2										1	1	1											
	3									1	4		4	1										
	4								1	2	2	1	2	2	1									
	5								1		2	2	2	2	2		1							
	6							1	1			4	4	4			1	1						
	7					1	4	4	1		1	2	1		1	4	4	1						
	8				1	2	4	4	2	4					4	2	4	4	2	1				
	9			1							4				4						1			
	10		1	1	1															1	1	1		
	11	1	4		4	1												1	4		4	1		
	12	2	2	1	2	2	1										1	2	2	1	2	2	1	
	13		2	2	2	2		1								1	2	2	2	2	2	2	4	
	14	4	4	4	4			1	1						1	1		4	4		4			
	15		1	1	1		1	4	4	1				1	4	4	1		1	1	1			
	16	1	4		4	4	2	4	4	2	1		1	2	4	4	2	4	4		4	1		
	17	2	2	1	1							4						1	1	2	2	1		
18	4	2		4	1												1	4	2		4			
19	4	4	4	2	2	1										1	2	2	4	4	4			
20	1	2		1	2		1								1	2	1	2	1					
21	1						1	1						1	1								1	
22						1	4	4	1			1	4	4	1									
23					1	2	4	4	2	1		1	2	4	4	2	1							
24				1							4							1						
25			1	1	1											1	1	1						
26		1	4		4	1									1	4		4	1					
27	1	2	2	1	2	2	1								1	2	2	1	2	2	1			
28	1	2	2	2	2	2		1						1	2	2	2	2	2	2	2	1		
29	4	4	4	4	4		1	1				1	1		4	4	4	4		4				
30		1	1	2	1		1	4	4	1		1	4	4	1		1	2	1	1				
31	1	4				4	2	4	4	2	4	2	4	4	2	4					4	1		
32	1	2	2			4					1					4					2	2	1	
33	1			4							1	1	1								4		4	

**Şekil 1.9. Xenakis'in *Horos* adlı eserinin 10. ve 14.-15. ölçülerinde kullandığı hücresel otomatın tablosu (Solomos 2005).**

Xenakis bu eserde toplamsal otomat kullanmıştır. Tablo soldan sağa (0'dan 22'ye)

ses dizisinin sırasıyla herbir elemanına eşlenmiştir. Hücre değerlerinin güncellenmesi, bu tabloya göre yukardan aşağı gösterilmiştir. Hücre 0 değeri aldığı anda nota çalmaz, 1 değeri aldığı anda nota bakır üflemeliler tarafından, 2 değerini aldığı anda tahta üflemeliler tarafından, 4 değerini aldığı anda ise yaylılar tarafından çalınır. Bu otomat 3 değerini vermemektedir.

20. yüzyıldaki başlıca algoritmik kompozisyon tekniklerini burda bitiriyoruz. En son olarak hücresel otomatları ele almış olmamız, bu tekniğin 80'li yıllarda derinlemesine incelenmiş olmasından ve güncelliğini hala korumasından kaynaklanmaktadır. Xenakis'in önemi de, burdan gelmektedir. Sanat yaşamı boyunca, bilimsel gelişmeleri takip etmiştir ve hücresel otomatları ilk kullanan besteci de Xenakis'tir.

#### **1.4 Biyoloji Esinli Algoritmik Uygulamalar**

"Müzikte, örüntü (*pattern*), tasarımsal planda birincil kavramdır: tekrar ve ritmik örüntüler, kaos ve düzen, yeni ve tanıdık olan, gürlük ve sessizlik, tınısal ve tonal uyum/uyumsuzluk. Bir kompozisyonu oluşturan, bu öğelerin etkileşimidir. Doğa, besteciler için en önemli esin kaynaklarından biri olmuştur ve böyle olmaya devam edecektir. Bunun nedeni insanların duygusal mekanizmaları ile ilgili olduğu kadar, doğanın örüntüsel karakteri ile de bağlantılıdır. Örüntülerin bilimi olarak matematik, müzik tarihinde önemli bir rol oynamıştır; rönesans çağının müzik besteleme kılıgında, gezegenlerin aksellerinin kullanımından günümüzdeki cebirsel, fraktal denklemler ve bilgisayar kullanımına kadar çeşitli biçimlerde ortaya çıkmıştır bu etki." (Brooks 1996, 27)

Brooks ve Ross, *Ant Music* (Karınca Müziği) adlı kompozisyonlarında, küçük bir karınca kolonisinin kendiliğinden örgütlenme davranışını temel almışlardır. *Zoion* adlı serbest kompozisyonlarında ise kimi hayvan türlerinin çiftleşme ve

evrimleşmesindeki dinamik özelliği modellemeye çalışmışlardır. Bu iki çalışma sunuş kısmında zikrettiğimiz doğanın taklidi ve *mimesis* kavramlarına örnek teşkil etmektedir.

#### 1.4.1 *Ant Music*

*Ant Music*, Toft'un *Leptothorax acevorum* türündeki karıncaların yuva içindeki davranışlarını modelleyen simülasyonuna dayanır (Tofts 1992) ve iki partili vurma çalgı müziğidir. Toft'un simülasyonunda bir yuvadaki karıncalar şu üç halden birine sahiptir: *uykuda*, *uyandırılabilir* veya *uyanık*. Bir karınca uykusunu bitirince *uyandırılabilir* veya *uyanık* konumuna geçer. Uykudaki bir karınca uyandırılmaz. *Uyanık* bir karınca sürekli olarak, *uyandırılabilir* karıncaları *uyanık* konuma geçirmeye çalışır. Ve bunun tersine olarak da, ikisinin iletişime girdiği zamansal adımda; eğer bir karınca *uyandırılabilir* ise, *uyanık* bir karınca onu uyandırabilir, ve ikisi de daha sonra *uykuda* haline geçer. (Toft'un simülasyonunda aynı zamanda, bir karınca uyandırılmadan da *uykuda* haline geçebilir; bu sistemde bu olanak kullanılmamıştır.) Bu türden otomat tarzı davranışları sergileyen bir çok karınca tek bir sistem içinde harekete geçirildiğinde, bireylerin kaotik düzensizliği, toplu bir düzenin oluşmasını sağlar. Ve sistem, düzen ve düzensizlik arasındaki modunu sürekli değiştirir.

Bu otomatik kendiliğinden örgütlenme, karınca ve diğer sosyal hayvanların doğal davranışlarına bir örnek oluşturur. Kendiliğinden örgütlenme fenomeni, karınca kolonilerinin bütüncül davranışlarının idamesini sağlar.

Bu sistemin müzikal kompozisyonda kullanımında, her bir karınca bireyi ayrı bir partiye atanmıştır. Ezgiler ve diğer müzikal özellikler, sistemde karıncaların sahip olduğu hallere göre üretilir; her bir hale farklı olasılıklar atanmıştır; müzikal çıktı MIDI olarak kodifiye edilmektedir. *Uyanık* bir karınca kendisine atanan vurma çalgı tınısı ile özgün ritmik motifini icra eder. Karıncalar etkileşim halinde olduğu için,

kendi özgün ritmik motifleri daha kompleks ve bütüncül birliğe doğru evrimleşir. Uyumlanan bir koloni adeta döngüsel ve organize bir ritme sahip olur.

### 1.4.2 Zoion

*Zoion*'da ise soyut organizmaların kendi kendini idame ettirmeleri müzikal olarak modellenmiştir. Başlangıçta sekiz yaratık vardır. Yaratıklar, nesil (kompozisyon sekizinci nesilde son bulur) ve tür bazında tanımlanırlar. Verili bir zamanda yaratık şu üç halden birinde bulunabilir: *dinlenme*, *çiftleşme* ve *avlanma*. Yaratıklar stokastik olarak tayin edilen bir süre boyunca *dinlenme* halinde bulunurlar. Sonrasında ya *çiftleşme* ya da *avlanma* haline geçerler. Eğer aynı nesilden iki yaratık birbiri ile eşgüdümleirse, 1/16 ihtimalle çiftleşir ve ebeveynlerinin türsel özelliklerinin karışımına sahip olan iki yavru meydana getirirler; bu sayede diğer türlerinin çeşitlemesi olan yeni türler ortaya çıkar. Çiftleşip çoğalan yaratıklar ölürler. Bu süreç sekizinci neslin ortaya çıkışına kadar devam eder ve bu kompozisyonun sonudur.

Türler ve bireyler, mi-minör tonunda, kendilerine özgün akor ve ezgisel motifler kullanırlar; çalınan ezgiler, yaratığın *avlanma* ya da *çiftleşme* haline bağlıdır. Kompozisyon tek bir tonda tanımlanmış olsa da, çıkan tınısal sonuç atonaldir, yani tonal karakter alabildiğine maskelenmiştir. Çeşitli hallere atanan tınılar birbirinden farklıdır.

Bunun bilgisine sahip bir dinleyici, sistemde işleyen doğal süreçleri tınılarla takip edebilir. *Avlanma* sesi çığlıktan pes armonikleri kuvvetli tınılara kadar değişkenlik gösterir. Bunun nedeni farklı türlerin karşılaşması temsil etme ihtiyacıdır. *Çiftleşme* yüksek perdeden uzunca tınlayan bir keman sesiyle temsil edilir. *Çiftleşme* haline geçmemiş bir yaratığın ses şiddeti, zaman ilerledikçe artmaktadır. Yeni evrimleşen bir türde, ebeveynlerin ezgisel özellikleri, yeni bir kombinasyonla içermektedir. Sonuç olarak, her nesil farklı bir çalgı sesine atanmaktadır. (Brooks & Ross 1996)

## 2. Bölüm: Markov Zincirleri

Markov zincirleri adını Rus matematiçi Andrey Markov'dan (1856-1922) almaktadır. Stokastik bir süreç olarak Markov zincirleri, birbiri ile ilişkilendirilen adımların peşpeşe gelebilmesini sağlayan bir sistemdir. Markov zincirlerinde gelecek durumların olasılığı, geçmiş (mevcut) durum(lar)a (*state*) göre belirlenmektedir. Gelecek durumların olasılığında hesaba katılan geçmiş adımların (*order*) sayısına katman diyebiliriz. Tek katmanlı Markov zincirinde, gelecek durumun seçilimi, yalnızca bir önceki duruma göre tayin edilir. Aynı şekilde, iki katmanlı Markov zincirinde ise gelecek durumun seçilimi, son iki duruma göre belirlenmektedir. Markov zincirleri, istenilen sayıda katmandan oluşturulabilir. Geçmiş durumlara göre her bir adımın tayin edilmesi özelliğinden dolayı, Markov zincirleri veri örüntüleri (-nin süreçsel karakterini) içindeki değişimleri değişik derecelerde modelleyebilir ve/ve ya taklit edebilir. Kurulan zincirin katman sayısı arttıkça, sistemin modellediği örüntüye göre tutarlılığı artmaktadır. Markov zincirinin dayandığı örüntü, besteci tarafından tayin edilebileceği gibi, bir verinin istatistiksel analizine de dayandırılabilir.

Markov Zincirleri genellikle durum değişim tablosu (*state transition matrix*) biçiminde gösterilir. Bu tablo, her bir durumun çeşitli mevcut durumlardan sonra vuku bulma ihtimalini gösterir. Aşağıdaki tabloda ardarda yapılan yazı-tura atışları için Markov tablosu verilmiştir. Olasılık değerleri, tam değer 1 olarak verilmektedir.

	gelecek Yazı Tura	
geçmiş Yazı	0.5	0.5
Tura	0.5	0.5

**Tablo 2.1 Yazı/tura atışı için Markov tablosu**

Bu tabloya göre 20 atış sonucu şöyle bir değer dizisi elde edilebilir:  
y t t t y t t y y t t y y t y y t t y t

Yukarıdaki tabloda gerçekleştireceğimiz, olasılık değeri değişiklikleri sonucu, elde edeceğimiz dizinin örüntüsünü değiştirebiliriz. Aşağıdaki tabloya göre her t'den sonra mutlaka y gelmelidir ve y'den sonra y gelme şansı 4'te 3'tür.

geçmiş Yazı	gelecek Yazı Tura	
	0.75	0.25
Tura	1	0

**Tablo 2.2 Yönlendirilmiş yazı/tura Markov tablosu**

Bu durumda ortaya çıkan örüntü değişik olacaktır:  
t y y y y t y t y y t y y y y t y y t y

Aynı tabloyu iki katmanlı Markov zincirine de çevirebiliriz.

geçmiş ş		gelecek Yazı Tura	
	(y y)	0.8	0.2
	(y t)	0.9	0.1
	(y y)	0.2	0.8
	(t y)	0.9	0.1

**Tablo 2.3 İki katmanlı yönlendirilmiş Markov tablosu**

Buna göre elde edebileceğimiz örüntü şu şekilde olabilirdi:  
y y y y y t y y t y y t y y y y t y y t

## 2.1 Markov Zincirlerinin Matematiksel Nitelikleri

Markov Zincirinde, elemalar arası geçişkenliği tanımlamakta fayda vardır. Eğer sonlu sayıda adım sonucunda Y durumundan X'e geçiş olası ise; X, Y'den ulaşılabilir ( *reachable* ) ve şu halde iki durum biribiri ile ilişkilidir ( *communicate* ) denir.

İçsel ilişkiler, bir Markov tablosunda dengeyi oluşturur, çünkü aşağıdaki özellikleri karşılar:

- kendine dönüşlüdür ( *reflexive* ): her durum kendi ile ilişkilidir
- bakışımıdır ( *symmetrical* ): eğer X durumu Y ile ilişkili ise, aynı şekilde Y durumu da X ile ilişkilidir
- geçirgendir ( *transitive* ): Eğer X durumu Y ile ilişkili ise, ve Y durumu da Z ile ilişkili ise, o halde X durumu da Z ile ilişkilidir. (Miranda 2001, 70)

Markov tablosu içinde kimi elemanlar tekrarlı olarak ortaya çıkma eğilimindedir ( *recurrent* ). Yani, belli mevcut durumlar sonucu gelme olasılığı yüksektir. Öte yandan kimi elemanların da seçilim olasılığı çok düşüktür ve bir kez seçildikten sonra tekrar gelme olasılığı çok düşüktür ( *transient* ). Her Markov zinciri en az bir tekrarlı eleman içermelidir ve istenilen sayıda tekrarsız eleman içerebilir.

Yalnız bir tane tekrarlı eleman ve bir çok tekrarsız eleman içeren Markov zincirlerine ergodik denir. Ergodik sistemlerin özelliği, sınırlı bir zaman diliminde sistem içindeki tüm noktalara bağlanabilmeleridir. Klasik armonideki I-IV-V derece akor bağlantıları ergodik bir sistemdir. Bu sayede majör dizi içerisindeki tüm notalar kullanılmış olmaktadır. Müzikal uygulamalar için ergodik özellik gösteren Markov zincirler kurulması daha elverişlidir.

Markov zincirleri, gerçekte vuku bulan olayları modellemekte de

kullanılabilmektedir. Modellemek istediğimiz olayı sıralı değerler olarak kodlayıp, bu diziye istatistiksel analiz uygularız. Bunun sonucu olarak elde ettiğimiz Markov zinciri, modellediği veri ile, istatistiki olarak benzer davranışları sergiler. Markov analizi, veri elemanlarının sıralı olarak sınanmasıyla yapılır. Aynı anda kaç elemanın sınanacağı, kaç katmanlı Markov zinciri kurduğumuza bağlı olarak değişir. Tek katmanlı bir Markov analizinde, aynı anda sıralı iki eleman sınanır: ilk eleman mevcut durumu, ikinci eleman ise bu mevcut durumun ardılını temsil eder. İki katmanlı bir Markov analizinde ise aynı anda sıralı üç eleman birlikte sınanır: ilk iki eleman mevcut durumu, üçüncü eleman ise bu ikili mevcut durumun ardılını temsil eder. Analiz bu şekilde sıralı olarak tüm veri dizisi üzerinde gerçekleştirilir ve her bir özgül mevcut durum için hangi özgül ardılın geldiği sayılır. Örneğin aşağıdaki gibi bir veri dizimiz (diyelim ki, ezgisel cümle) olsun.

a a b c b a

Tek katmanlı bir markov analizi sonucu bu veriden şu (mevcut ve ardılı) çiftleri elde ederiz:

aa, ab, bc, cb, ba, aa

En son çift, verinin baş ve sonunun birleştirilmesi ile oluşturulmuştur, bu sayede verinin son elemanına bir ardıl atanmıştır. Verinin son elemanına bu şekilde bir ardıl tayin edilmesi sistemin sürerliğini garanti etmektedir.

Markov analizindeki son adım ise, elde ettiğimiz çiftlerin mevcut durum ve ardıl sıklığını saymaktır. Yukarıda çiftler bize aşağıdaki tabloyu vermektedir.

	A	B	C
A	2	1	0
B	1	0	1
C	0	1	0

**Tablo 2.4 Markov analiz tablosu**

Bu tablo değerlerinin birimleştirilmiş (*normalization*) hali ise aşağıdaki gibidir.

	A	B	C
A	0.67	0.33	0
B	0.5	0	0.5
C	0	1	0

**Tablo 2.5 Birimleştirilmiş Markov analizi tablosu**

## 2.2 Müzikal Uygulamalar

Markov zinciri örneklerini *Grace* adlı uygulamanın 'make-markov' ve 'markov-analyze' fonksiyonlarını kullanarak türetebiliriz. Aşağıda verdiğimiz Markov tablosu ritm üretmek için kurulmuştur. Bu örnekte, 4/4'lük tartım içinde dörtlük, noktalı sekizlik, sekizlik ve onaltılık ritmler kullanılmıştır.

	♪	♪.	♪	♪
♪	0.2	0.3	0.3	0.2
♪.	0	0	0	1
♪	0	0	0.5	0.5
♪	0	0.4	0.3	0.3

**Tablo 2.6 Ritmik akış için Markov tablosu**

Yukarıdaki tabloyu ritm üretici olarak kullanabilmek için *Grace*'de çalıştırabileceğimiz bir *Scheme* kodu yazalım.

(define (ritmik uzunluk ritm-tablo)

(loop with markov = (make-markov ritm-tablo) ;girilen tablodan markov zinciri oluşturur;

for i below uzunluk ;argüman olarak girilen ritm sayısına  
kadar süreci sürdürür;  
with an = 0

```

for r = (next markov) ;markov zincirinden, sıradaki olası değeri
çağırır;
do
(send "fms:note" :time an :pitch 67 :dur r) ;değerleri Fomus arabirimini
kullanarak partisyona yazar;
(set! an (+ an r))
finally (send "fms:run"))

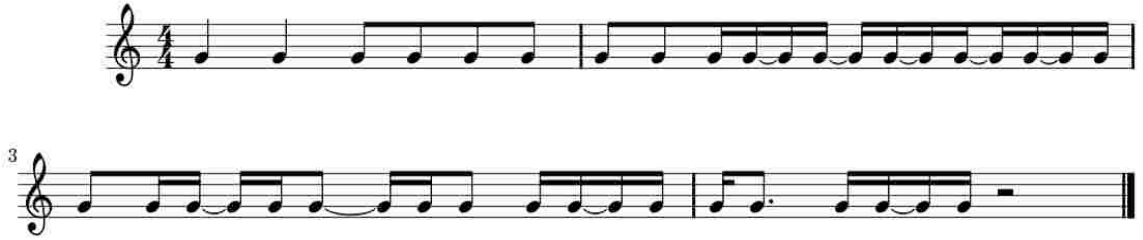
```

```

(let ((ritmler '((1 -> (1 0.2) (0.75 0.3) (0.5 0.3) (0.25 0.2))
(0.75 -> 0.25)
(0.5 -> (0.5 0.5) (0.25 0.5))
(0.25 -> (0.75 0.4) (0.5 0.3) (0.25 0.3))))))
(ritmik 30 ritmler))

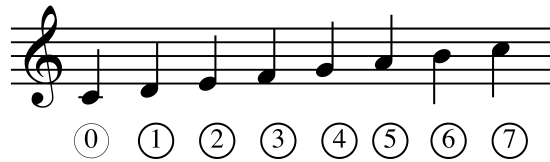
```

Yukarıdaki kodları çalıştırdığımızda elde ettiğimiz çıktı aşağıdaki gibidir.



**Şekil 2.1 Ritmik akış Markov tablosunun nota çıktısı**

Yukarıda kullandığımız ritm üreten algoritmaya ek olarak nota dizilimini belirleyen bir Markov zinciri oluşturalım. Bu nota dizilimi do majör gamında olsun. Her bir notaya bir sıra numarası tayin edelim.



**Şekil 2.2 Numaralandırılmış Do-majör dizisi**

Yukarıdaki notalar kullanılarak oluşturulan, tek katmanlı Markov zinciri tablosu

aşağıdadır. Burda tayin edilen geçişkenlik değerleri tonal bir bağlam oluşturma amacıyla belirlenmiştir.

	0	1	2	3	4	5	6	7
0	0.2	0.2	0.2	0	0.2	0	0	0.2
1	0.33	0	0.33	0	0.33	0	0	0
2	0	0.5	0	0.5	0	0	0	0
3	0.33	0	0.33	0	0.33	0	0	0
4	0.25	0	0	0.25	0.25	0.25	0	0
5	0	0	0	0	0	0	1	0
6	0	0	0	0	0	0	0	1
7	0	0	0	0	0	0.5	0.5	0

**Tablo 2.7 Ezgisel akış için Markov tablosu**

Bu tablo değerlerini, bir önceki ritm üreten kodumuza eklersek:

```
(define (tonal-ritmik uzunluk ritm-tablo nota-tablo)
  (loop with markov-ritm = (make-markov ritm-tablo)
        with markov-perde = (make-markov nota-tablo) ;girilen perdelerden Markov
zinciri oluşturur;
        with gam = '(60 62 64 65 67 69 71 72) ;do majör gamının MIDI protokol
numaraları;
        for i below uzunluk
        with an = 0
        for r = (next markov-ritm)
        for p = (next markov-perde)
        do
        (send "fms:note" :time an :pitch (nth gam p) :dur r :part "oba")
        (set! an (+ an r))
        finally (send "fms:run"))))

(let ((ritmler '((1 -> (1 0.2) (0.75 0.3) (0.5 0.3) (0.25 0.2))
                (0.75 -> 0.25)
                (0.5 -> (0.5 0.5) (0.25 0.5)))
```

(0.25 -> (0.75 0.4) (0.5 0.3) (0.25 0.3))))  
(perdeler '((0 -> (0 0.2) (1 0.2) (2 0.2) (4 0.2) (7 0.2))  
(1 -> (0 0.33) (2 0.33) (4 0.33))  
(2 -> (1 0.5) (3 0.5))  
(3 -> (0 0.33) (2 0.33) (4 0.33))  
(4 -> (0 0.25) (3 0.25) (4 0.25) (5 0.25))  
(5 -> 6)  
(6 -> 7)  
(7 -> (5 0.5) (6 0.5))))))  
(tonal-ritmik 30 ritimler perdeler))

Ritm ve perde geçişkenliği Markov tablolarında tanımlanmış verileri kullanarak ürettiğimiz müzik örneği aşağıdadır.



**Şekil 2.3 Ezgisel akış Markov tablosu nota çıktısı**

## 2.3 Markov Analizi

Şimdiye kadar gördüğümüz örneklerde, Markov zinciri elemanları arasındaki geçişkenliği belirleyen olasılık değerleri tarafımızdan tayin edilmiştir. Ancak Markov sürecini, başka verileri (örneğin ezgileri) modellemekte de kullanabiliriz. Modellemeyi manuel olarak yapabileceğimiz gibi, bilgisayar yardımı ile gerçekleştirmek daha verimli olmaktadır. Markov analizi için bu çalışmada *Grace* uygulamasının 'markov-analyze' fonksiyonu kullanılmıştır. Bu fonksiyona liste olarak girilen verinin, istenilen katmanda Markov zinciri oluşturulmaktadır. Analiz için, kendi kültür dünyamızda çok iyi bilinen aşağıdaki ezgiyi seçtim:



**Şekil 2.4 Kaptanizade Ali Rıza Efendi, *Yıldızların Altında* adlı ezgisi.**

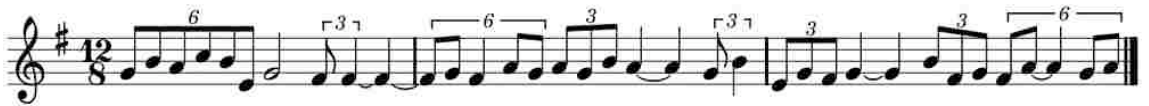
Ezginin tek katmanlı analizi için aşağıdaki *Scheme* kodunu yazdık. Dikkat edileceği

üzre, yeni ezgiler üretmek maksadıyla perde sıralamalarının yanı sıra ritmik değerlerin de Markov analizi yapılmıştır.

```
(define (markov-analiz erim)
  (loop repeat erim
    with an = 0
    with ritm-pat = (make-markov
      (markov-analyze
        '(1/3 1/3 1/3 2/3 1/3 2 1/3 1/3 1/3 2/3 1/3 2 1/3 1/3 1/3 1/3
          1/3 1/3 1 1)
        :order 1 :mode 3))
    with ezgi-pat = (make-markov
      (markov-analyze
        '(64 67 71 69 67 71 64 67 71 69 67 66 66 67 69 69 67
          66 69 72 71 66 67 69 67 66 64)
        :order 1 :mode 3))
    for ritm = (next ritm-pat)
    do
      (send "fms:note" :time an :pitch (next ezgi-pat) :dur ritm)
      (set! an (+ an ritm))
      finally (send "fms:run"))))
```

(markov-analiz 40)

Ve bu kodu çalıştırarak elde edebileceğimiz sonsuz sayıda ezgilerden bir örnek aşağıdadır.



**Şekil 2.5 Yıldızların Altında Markov versiyonu**

Markov Zincirleri, algoritmik kompozisyonda kullanılan en eski tekniklerden birisidir. Lejaren Hiller ve Leonard Isaacson, *Illiatic Suite*'in yazılma sürecinde (*Experiment IV*) çeşitli sayıda katmandan oluşan Markov zincirleri kullanmışlardır (Hiller & Isaacson 1959, 141-148). Daha da önemlisi aynı çalışmada, zamanın bir fonksiyonu olarak Markov zinciri elemanlarının geçişkenlik (olasılık) değerlerini

tedrici olarak deęiřtirmeyi de kullanmıřlardır (Hiller & Isaacson 1959, 139). Verili datanın analizine dayalı Markov zinciri, Hiller ve Robert Baker'ın *Computer Cantata* adlı alıřmalarında kullanılmıřtır (Hiller & Baker 1964). Bu alıřmada Markov analizine temel oluřturan veriler Charles Ives'ın *Three Places in New England* adlı eserinden alınmıřtır.

Markov zincirlerini kullanarak, bir stilde eřitlemeler reten ok sayıda otomat yazılmıřtır. Fred ve Carolyn Attneave batı kovboy řarkılarını analiz etmiřler ve bu analizden hareketle aynı stilde bir ok řarkı tretmiřlerdir (Cohen 1962, 137-163). Harry Olson ve Herbert Belar ise 1950'lerin bařında Princeton laboratuvarlarındaki alıřmalarında *RCA* adlı ses retecisi makinelerine otomatik beste yapıcı ek bir modl eklemiřlerdir. Bu modl bir eřit analog bilgisayar idi ve tek iřlevi řarkı tretmektir. Bu makinenin alıřma prensibi, 19. yzyılda yařamıř Amerikalı řarkıcı Stephen Collins Foster'ın řarkılarının Markov analizine dayanmaktadır. Bu analiz 11 Foster řarkısının ritmik ve ezgisel plandaki istatistięini iermektedir (Holmes 2008, 143-144). Foster řarkılarından elde edilen bu veriler Dodge ve Jerse'nin ortak alıřmalarında tekrar kullanılmıřtır (Dodge & Jerse 1997). Foster řarkılarını yeniden reten bu Markov zincirini, Heinrich Taube *Notes from the Metalevel* kitabına ek olarak *Scheme* dilinde vermiřtir. Bu kodu aynen aktarıyorum.

```
(define (foster len octave chan)
  (let ((p1 (make-markov
    '(b3 d4 -> d4)
    (cs4 d4 -> (d4 .3125) (e4 .3125) (a4 .3125))
    (d4 d4 -> (cs4 .125) (d4 .125) (e4 .5625) (fs4 .125)
    (a4 .0625))
    (e4 d4 -> (b3 .0625) (d4 .0625) (e4 .25) (fs4 .3125)
    (a4 .0625) (cs5 .0625) (d5 .1875))
    (fs4 d4 -> (e4 .75) (fs4 .1875) (g4 .0625))
    (a4 d4 -> (e4 .6875) (fs4 .3125))
    (b4 d4 -> d4)
    (d4 b3 -> d4)
    (d4 cs4 -> d4)
```

(e4 cs4 -> d4)  
 (d4 e4 -> (d4 .1875) (e4 .25) (fs4 .5) (a4 .0625))  
 (e4 e4 -> (cs4 .0625) (d4 .75) (e4 .0625) (fs4 .125))  
 (fs4 e4 -> (cs4 .125) (d4 .4375) (e4 .1875) (fs4 .125)  
 (a4 .0625) (d5 .0625))  
 (d4 fs4 -> (e4 .4375) (fs4 .1875) (g4 .125) (a4 .25))  
 (e4 fs4 -> (d4 .0625) (e4 .1875) (fs4 .3125) (g4 .25)  
 (a4 .0625) (b4 .0625))  
 (fs4 fs4 -> (d4 .1875) (e4 .25) (fs4 .3125) (g4 .125)  
 (a4 .0625))  
 (g4 fs4 -> (e4 .5) (g4 .5))  
 (a4 fs4 -> (d4 .3125) (e4 .25) (fs4 .1875) (g4 .0625)  
 (a4 .125) (b4 .0625))  
 (b4 fs4 -> (e4 .6875) (fs4 .3125))  
 (d4 g4 -> (fs4 .6875) (b4 .3125))  
 (fs4 g4 -> (fs4 .25) (g4 .1875) (a4 .3125) (b4 .1875))  
 (g4 g4 -> (g4 .5) (a4 .5))  
 (a4 g4 -> fs4)  
 (b4 g4 -> b4)  
 (a4 gs4 -> a4)  
 (d4 a4 -> (fs4 .25) (a4 .75))  
 (e4 a4 -> (a4 .8125) (b4 .1875))  
 (fs4 a4 -> (fs4 .125) (a4 .625) (b4 .1875) (d5 .0625))  
 (g4 a4 -> (d4 .125) (a4 .625) (d5 .25))  
 (gs4 a4 -> a4)  
 (a4 a4 -> (fs4 .25) (g4 .0625) (gs4 .0625)  
 (a4 .3125) (b4 .3125))  
 (b4 a4 -> (d4 .0625) (fs4 .5625) (g4 .0625) (a4 .125)  
 (b4 .0625) (d5 .125))  
 (d5 a4 -> (fs4 .875) (a4 .125))  
 (e5 a4 -> a4)  
 (fs4 b4 -> a4)  
 (g4 b4 -> a4)  
 (a4 b4 -> (d4 .0625) (fs4 .0625) (a4 .75) (b4 .0625)  
 (b4 .0625))  
 (b4 b4 -> (fs4 .125) (a4 .75) (d5 .125))  
 (cs5 b4 -> a4)  
 (d5 b4 -> (g4 .0625) (a4 .3125) (b4 .3125) (d5 .25))  
 (d4 cs5 -> d5)  
 (d5 cs5 -> (b4 .75) (d5 .25))  
 (e5 cs5 -> d5)  
 (d4 d5 -> (a4 .125) (b4 .6875) (cs5 .1875) )  
 (e4 d5 -> cs5)

```

(a4 d5 -> (a4 .3125) (b4 .3125) (cs5 .1875) (d5 .125))
(b4 d5 -> (a4 .5625) (b4 .125) (cs5 .3125))
(cs5 d5 -> (b4 .3125) (e5 .625))
(d5 d5 -> b4)
(d5 e5 -> (a4 .3125) (cs5 .6875))))))
(p2 (make-weighting
`((,(make-cycle '(h h)) .375)
,(make-cycle '(q q q q)) .125)
,(make-cycle '(h q q)) .125)
,(make-cycle '(q q h)) .125)
,(make-cycle '(q h q)) .25)
(w .125))))))
(loop repeat len
for k = (next p1)
for d = (rhythm (next p2) 200)
with an = 0
do
(send "mp:midi" :time an :key (key (transpose k octave)) :chan chan :dur d
:amp .5)
(set! an (+ an d))))))

```

## 2.4 Dinamik Markov Zinciri

Yukarıda örnekleri ve tarihsel bağlamı ile ele aldığımız Markov zinciri kullanımlarında sistemin çalışması ve olasılık değerleri statikti. “Dinamik ve olasılığa dayalı, Markov tablo değerlerinin kullanımı pek yaygın olmasa da, vardır” (Ariza 2006, 8). Dinamik olarak olasılık değerlerinin değiştirilmesi örneğini L.Hiller ve L. Isaacson'un *Experimental Music* adlı eserlerinde görmüştük. O örnekte olasılık değerleri müzikal dokunun konzonans-disonans dengesini değiştirmek maksadıyla kullanılmıştı. (Hiller & Isaacson 1959, 139). Bizim geliştirdiğimiz sistemde ise olasılık değerlerinin zamanın bir fonksiyonu olarak değiştirilmesi *melodic morphing*e hizmet etmektedir. Biribirinden tamamen farklı iki ezginin tek katmanlı Markov analizi yapıldı. Ve kurulan sistem, bu iki tablo arasındaki enterpolasyonu gerçekleştiriyor. Yani sistem başlangıcında ilk tablonun elemanları arasından seçimde bulunurken, sisteme girilen zaman argümanı ilerledikçe ikinci tablonun elemanları kendi olasılık değerleriyle yoğunluk kazanmaktadır. Sisteme başlangıç

oluşturan ilk ezgi ve bu ezginin tek katmanlı Markov analizi aşağıdadır. Bu ezgi gong kulelerinde çalınan standart bir kalıptır.



Şekil 2.6 Standart Gong ezgisi

Ezgi A	C <sub>3</sub>	D <sub>3</sub>	E <sub>3</sub>	G <sub>2</sub>
C <sub>3</sub>	0	0.5	0.5	0
D <sub>3</sub>	0.33	0	0.33	0.33
E <sub>3</sub>	0.67	0.33	0	0
G <sub>2</sub>	1	0	0	0

Tablo 2.8 Gong ezgisinin Markov analizi tablosu

Sistemin, sürecin sonunda bağlanacağı hedef ezgi ise Itri'den alınan 'Bayram Tekbiri' ezgisidir. Aşağıda ezgiyi ve bu ezginin tek katmanlı Markov analizini veriyoruz.



B <sub>2</sub>	0	0	1	0	0	0	0	0
B <sub>3</sub>	0	0	0	0	0	1	0	0
C <sub>4</sub>	0	0.07	0.67	0.2	0.07	0	0	0
D <sub>4</sub>	0.17	0	0.67	0	0	0.17	0	0
E <sub>4</sub>	0	0	0	0.5	0.5	0	0	0
E <sub>4</sub>	0	0	0	0.4	0	0.4	0.2	0
F <sub>4</sub>	0	0	0	0	0	0.5	0	0.5
G <sub>4</sub>	0	0	0	0	0	0	1	0

**Tablo 2.9 Bayram Tekbiri Markov analizi tablosu**

Kullanılacak ezgilerin analizi yapıldıktan sonra bu iki Markov zinciri, düğüm noktaları çoktan seçmeli olan ve bu düğüm noktalarındaki olasılık değerleri tedrici olarak değiştirilen, tek bir graf ağı haline getirildi. Örneğin G (sol) notası birinci ezginin analizinde, tek ihtimalle C'ye (do) giderken, ikinci ezginin analizinde gene tek ihtimalle F'ye (fa) bağlanmaktadır. Kurulan graf ağında G notasının bağlanabileceği iki nota da farklı olasılıklarla tanımlandı: sistemin başlangıcında G'nin C'ye bağlanması tam ihtimal (1), F'ye bağlanması ise 0 ihtimaldi. Bu ihtimallerin dinamik olarak değiştirilebilmesi için bunlar değişken (*local variable*) olarak tanımlandı. Ve bu değişkenler her zamansal adımda hedef değere (C'ye bağlanma ihtimali 1'den 0'a, F'ye bağlanma ihtimali ise 0'dan 1'e) yaklaştırıldı. Değerlerin dinamik olarak değiştirilmesinde, zamanın parametreleştirilmesi önkoşuldur. Sistemin her bir adımı, ve yani herbir yeni karar verme anında zaman bir birim artmaktadır. Sisteme argüman olarak girilen toplam adım sayısı birimleştirilmiştir.

Örneğin toplamda 100 adımlı bir süreç istiyorsak,

0'dan başlayıp 100'e kadar giden adım sayısına  $i$  diyelim,

her bir adımda  $i$ 'nin birimleştirilmiş değeri  $(i / 100)^1$  şeklinde hesaplanır.

Yani bu değerlerin alt sınırı 0 üst sınırı 1'dir. Zamansal süreci bu biçimde <sup>1</sup>

<sup>1</sup> Bu ifade Lisp programlama dilindedir; bölme işlemini belirtir.

birimleştirmenin faydası, adım sayısı değiştirilse de sistem içinde ilgili fonksiyonların zamanı hep aynı sınırlar (0-1) içinde yorumlayabilmesidir/oranlayabilmesidir. Dinamik olarak ihtimaller değiştirilirken 0'la 1 arasındaki birimleştirilmiş değerler ilgili değişkenin değerini tayin eder. Bunu hesaplayan, *Common Music* sistemine içre 'rescale' fonksiyonudur. 'rescale' fonksiyonuna beş argüman girilir: (rescale x x1 x2 y1 y2). x, yeni alt ve üst sınırlara göre oranlanacak değer; x1 ve x2 esas sınırlar; y1 ve y2 ise yeni alt ve üst sınırlardır.

Örneğin şu fonkiyonun (rescale .5 0 1 0 100) çıktısı, 50'dir. 0 ve 1 sınırları arasındaki 0.5 değeri, 0 ve 100 sınırları arasında oranlandığında 50 olmaktadır. Yukardaki örneğe geri dönecek olursak; birinci ezgide G tam ihtimalle C'ye bağlanırken, ikinci ezgide G tam ihtimalle F'ye bağlanmaktadır. O halde, artan zamana x dersek G'den sonra C'nin seçilme olasılığı (rescale x 0 1 1 0) [x'le ters orantılı] olarak azalmakta, G'den sonra F'nin seçilme olasılığı (rescale x 0 1 0 1) [x'le doğru orantılı] olarak artmaktadır. Aşağıdaki *Scheme* kodunda, her bir olasılığın *local variable* olarak tanımlandığı okunabilir (a1-t, a2-t. vs).

```
(define (DinamikMarkov erim)
  (let*
    ((a1-t 1) ;;1'den 0'a gider;;
     (a2-t .667) ;;0.667'den 0' gider;;
     (a3-t .333) ;;0.333'den 0'a gider;;
     (a4-t 0) ;;0'dan 0.067'e gider;;
     (a5-t 0) ;;0'dan 0.667'e gider;;
     (a6-t 0) ;;0'dan 0.2'e gider;;
     (a7-t 0) ;;0'dan 0.5'e gider;;
     (a8-t .09) ;; 0.09'dan 0.167'e gider;;
     (a9-t .1) ;;0.1'den 0.429'a gider;;
     (1-a1-t 0) ;;0'dan 1'e gider;;
     (r-.25 0) ;;0'dan 0.5'e gider;;
     (r-.75 .75) ;;0.75'den 0.5'e gider;;
     (r-.75 0) ;;0'dan 0.75'e gider;;
     (r-1 0) ;;0'dan 1'e gider;;
     (r-1.5 .25) ;;0.25'ten 1.5'a gider;;
     (Tal 1)
     (Ev 0)) ;;bu değişken tedrici olarak müziği düzenli tempodan rubato moda sokar;;
    (process with Tala = (make-weighting `
      (,(promise (vary .25 Ev)) ,(promise r-.25))
```

```

        (,(promise (vary .5 Ev)) ,(promise r-.5))
        (,(promise (vary .75 Ev)) ,(promise r-.75) :max 1)
        (,(promise (vary 1 Ev)) ,(promise r-1))
        (,(promise (vary 1.5 Ev)) ,(promise r-1.5) :max 1)
        ("1R" 1 :max 1)))
with Raga = (make-graph `(
"3")      ("g3" ,(make-weighting `(("c4" ,(promise a1-t)) ("g4" ,(promise 1-a1-t))))
          ("c4" ,(make-weighting `(("d4" ,(promise a2-t)) ("e4" ,(promise a3-t))
          ("bf3" ,(promise a4-t)) ("c4" ,(promise a5-t) :max 4) ("df4" ,
(promise a6-t)) ("ef4" ,(promise a4-t)))) "c4")
          ("d4" ,(make-weighting `(("e4" .3) ("g3" .5) ("c4" .3))) "d4")
          ("e4" ,(make-weighting `(("c4" ,(promise a1-t) :max 4) ("f4" ,(promise a8-
t)))) "e4")
          ("bf3" ,(make-weighting `(("ef4" ,(promise 1-a1-t)) ("c4" ,(promise a1-t))))
"bf3")
          ("b3" "c4" "b3")
          ("df4" ,(make-weighting `(("b3" ,(promise a8-t)) ("c4" .667) ("ef4" ,(promise
a8-t)))) "df4")
          ("ef4" ,(make-weighting `(("df4" ,(promise a9-t)) ("ef4" ,(promise a9-t))
("f4" .143) ("d4" ,(promise a3-t)))) "ef4")
          ("f4" ,(make-weighting `(("ef4" .5) ("g4" .5) ("d4" ,(promise a3-t)))) "f4")
          ("g4" "f4" "g4"))
)
for i to erim
for x = (/ i erim)
do
;;aşağıdaki fonksiyonlar tedrici olarak olasılık değerlerini günceller;;
(set! Ev (rescale x 0 1 0 1))
(set! r-.25 (rescale x 0 1 0 .5))
(set! r-.5 (rescale x 0 1 .75 .5))
(set! r-.75 (rescale x 0 1 0 .5))
(set! r-1 (rescale x 0 1 0 .35))
(set! r-1.5 (rescale x 0 1 .25 .5))
(set! Tal (next Tala))
(set! a1-t (rescale x 0 1 1 0))
(set! a2-t (rescale x 0 1 .667 0))
(set! a3-t (rescale x 0 1 .333 0))
(set! a4-t (rescale x 0 1 1 .067))
(set! a5-t (rescale x 0 1 0 .667))
(set! a6-t (rescale x 0 1 0 .2))
(set! a7-t (rescale x 0 1 0 .5))
(set! a8-t (rescale x 0 0.8 .09 .167))
(set! a9-t (rescale x 0 .8 .1 .429))
(set! 1-a1-t (rescale x 0 1 0 1))
;;midi sinyali olarak sistemin çıktısını yollar;;
(send "mp:midi" :key (key (if (equal? Tal "1R") 0 (next Raga))) :dur (if (equal? Tal "1R") 1
Tal))
(wait (/ (if (equal? Tal "1R") 1 Tal 1))))))

```

Bu kod (sprout (m-kette 100)) komutuyla harekete geçirilebilir. Adım sayısı olarak

girilen 100 deęeri deęiřtirilebilir.

Yukarıdaki sistemde ritmlerin de analizi yapılarak, ezgisel planda uygulanan Dinamik Markov teknięi ritmik plana da uygulanmıřtır. Ayrıca iki ezgi arasındaki stil farkına vurgu yapmak bakımından da, müzięi rubato moda sokan ayrı bir zamansal/doęrusal deęiřken/çarpan katılmıřtır koda. Bu deęiřken zamansal birimleri zamanla doęru orantılı olarak artan oranlarda rastlantısal deęerlerle çarpmaktadır. Bu proje Illinois eyaletinin Urbana řhriindeki McFarland otomatik gong kulesinde seslendirilmiřtir (Bkz. Ek-Cd, Parça No.1).

### 3. Bölüm: Hücresel Otomatlar

Çağdaş bilimde, 1950'lerde bilgisayar uygulamalarının gelişmesiyle birlikte araştırma biçimlerinde köklü bir değişim vuku bulmuştur. Maddenin yapısının analiz edilmesi ve bileşenlerinin tanımlanmasından ziyade, ki bu klasik laboratuvar geleneğidir, bileşenlerin farklı fonksiyonlarının tanımlanması ve bunların birbiriyle kurdukları kompleks ilişkilerin ortaya çıkardığı üst düzey örgütlenmeler (*global behaviour*) üzerine yoğunlaşmıştır. Doğa bilimlerinde, görüngüler, bilgisayar ortamında simüle edilerek, işlevsel hareketlilik ve etkileşim, veri örüntüleri olarak incelenmektedir. Yapay yaşam (*artificial life*) çalışmaları bu paradigmatik değişimin bir sonucu olarak ortaya çıkmıştır.(Bora 2003) “Hücresel otomatlar, ayrıntılı matematiksel analiz için yeterince yalındır; aynı zamanda çok çeşitli komplike görüngüleri doğurabilecek kadar da komplekstir. Öte yandan hücresel otomatların sahip olduğu umumiyet, geniş bir yelpazedeki fiziksel, kimyasal, biyolojik ve öteki sistemleri modellemeye de imkan sağlar.” (Wolfram 1983a, 601)

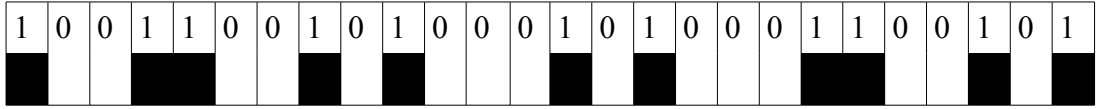
Yapay yaşam çalışmalarında, kimi doğal görüngüleri modellemekte kullanılmaya elverişli sistemlerin başlıcaları hücresel otomatlar (*cellular automata*), genetik algoritmalar ve Lindenmayer sistemleridir (*L-Systems*). Hücresel otomatlar, doğal görüngüleri modellemenin ötesinde kendisine has matematiksel yapısıyla farklı disiplinlerde çalışan bilimciler ve sanatçılar tarafından büyük ilgi görmektedir. Hücresel otomatların dinamik sistemlerin araştırılmasında ve kurulmasında bu kadar ilgi çekici olması, oldukça yalın kurallarla birbiri ile ilişkiye sokulan birimlerin, sistemin çalıştırılmasıyla doğurdukları kompleks örüntülerdir.

Hücresel otomatlar, zaman ve mekanın adımlı (*discrete*) olduğu dinamik sistemlerdir. “Hücresel otomatlar, çoğunlukla doğal sistemleri betimlemek için kullanılan kısmi (*partial*) diferansiyel denklemlerin, adımlı idealizasyonu olarak görülebilir. Adımlı kuruluşu, sayısal bilgisayarlarla aralarındaki önemli bir analogiye

olanak sağlar: hücresel otomatlar, yalın yapısıyla koştur-ışleyen bilgisayarlardır. “  
(Wolfram 1983b)

Çeşitli sayıda boyuta sahip olabilirler, ancak en yaygın kullanılan otomatlar, hücrelerin sıralı değerler olarak (*array*) kurulduğu tek ya da iki yönde gelişen biçimleridir. Hücrelerin güncellenmesini kontrol eden kural kümesi (*algorithm*) bu sıralı değerleri aynı anda günceller. Yani her bir zamansal adımda, o sıradaki hücreler birlikte güncellenir. Her hücre olası değerlerden birini alır (Von Neuman otomatında 0'dan 28'e kadar olan değer kümesi *Game of Life*'da 0 ve 1'dir). Bu değer kümesinin niceliği Wolfram tarafından  $k$  ile ifade edilmektedir (Wolfram 1984a).

$k = 2$  olan bir otomat sırasına aşağıdaki şekil örnek olabilir.

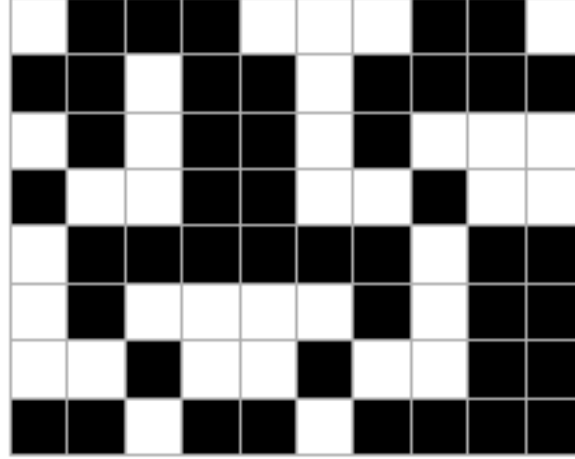


### Şekil 3.1 Bir hücresel otomat nesli örneği

Hücre değerlerinin güncellenmesinde işleyen kurallar, tanımlanan yerel komşuluk (*local neighbourhood*) değerlerini hesaba katar. Komşuların değerlerinin hesaba katılmasından anlaşılması gereken, hücrelerin çevresindeki hangi diğer hücrelere bakılacağıdır. Kaç tane en yakın komşu hücrenin hesaba katılacağı gene Wolfram'ın notasyonunda  $r$  ile ifade edilmiştir. Eğer  $r = 1$  ise, hücrenin kendisi ile bitişiğindeki iki hücrenin oluşturduğu üçlü kombinasyona göre hücre bir sonraki zamansal adımda yeni değerini alır.  $k = 2$  ve  $r = 1$  için 256 ayrı kural tipi vardır. Çünkü yalnız 2 değer alabilen 3 hücrenin sıralı olarak 8 ayrı kombinasyon imkanı vardır, ve bundan hareketle 2'nin 8'inci kuvveti 256'dır. Bu 256 ayrı kuralın numaralandırılması aşağıdaki tabloda gösterilmiştir.

111	110	101	100	011	010	001	000
0	1	0	1	1	0	1	0

Yukardaki kuralın sıra numarasını tayin etmek için, altttaki 0 ve 1 değerlerini 2 tabanlıdan 10 tabanlıya çeviririz;  $01011010_2 = 90_{10}$ . Aşağıdaki grafikte 90 numaralı kuralın 7 adımlı bir zamanda aldığı şekli görebiliriz. Sistemin başlangıç durumu, yani ilk hücre sırası rastlantısal olarak tayin edilmiştir.

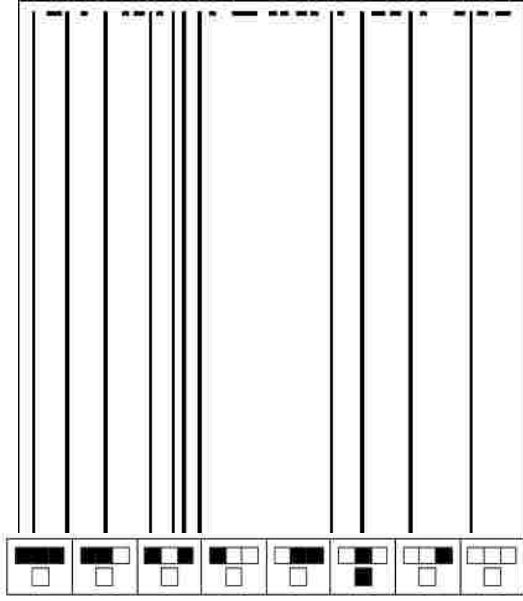


**Şekil 3.2 Kural No:90**

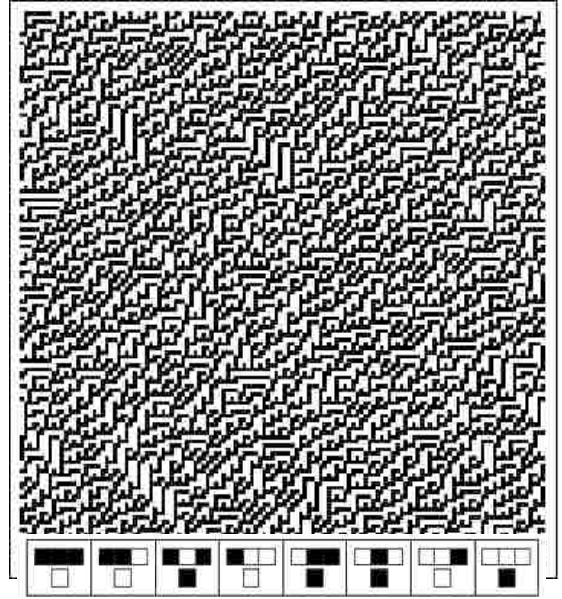
Stephen Wolfram 80'li yıllarda yaptığı yoğun hücresel otomat araştırmaları sonucunda, bu sistemlerin 4 temel davranışını tanımlamıştır (Wolfram 1983b):

1. Sistemin evrimi, homojen bir durumda sabitlenir. Örneğin bütün hücrelerin hep aynı değeri alması gibi;
2. Evrim, bir dizi durağan ve yinelgen yapıya bağlanır;
3. Evrim, kaotik örüntüler türetir;
4. Evrim, kompleks yapılar türetir.

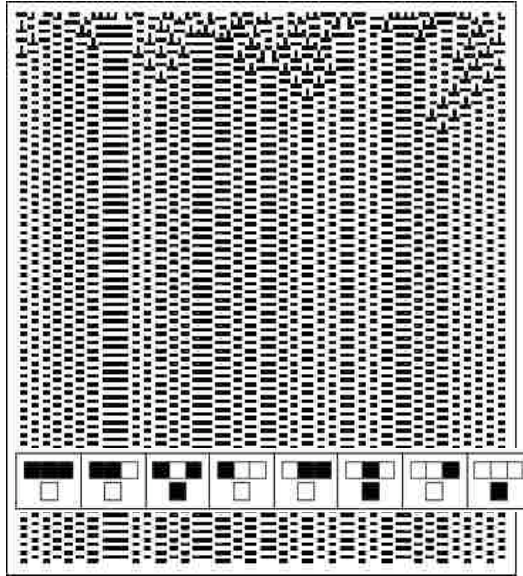
Kompleks yapılar ile kaotik yapılar arasındaki önemli ayrımı vurgulamamız gerekir. Kaotik yapılar, rastlantısal davranış sergiler, ve zamansal süreçte belirgin örüntüler belirmez. Ancak kompleks yapılar, kaotik olana yakın bir yoğunluk içerse de, sistem içinde kimi lokal örüntüler ve farklı alanlar arasında çeşitli bağlantılar oluşur. Aşağıda dört ayrı karakterdeki otomatlara birer örnek verdim.



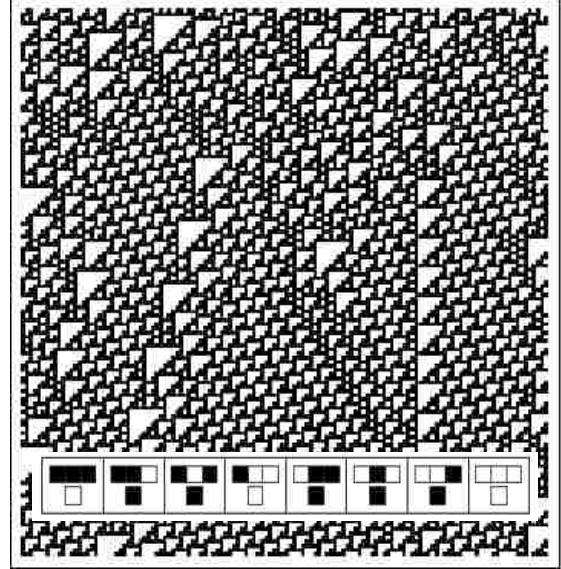
**Kural No. 4 – sabit davranış**



**Kural No. 45 – rastlantısal/kaotik davranış**



**Kural No. 37 – döngüsel davranış**



**Kural No. 100 kompleks davranış**

**Şekil 3.3 Dört temel hücreli otomat karakteri**

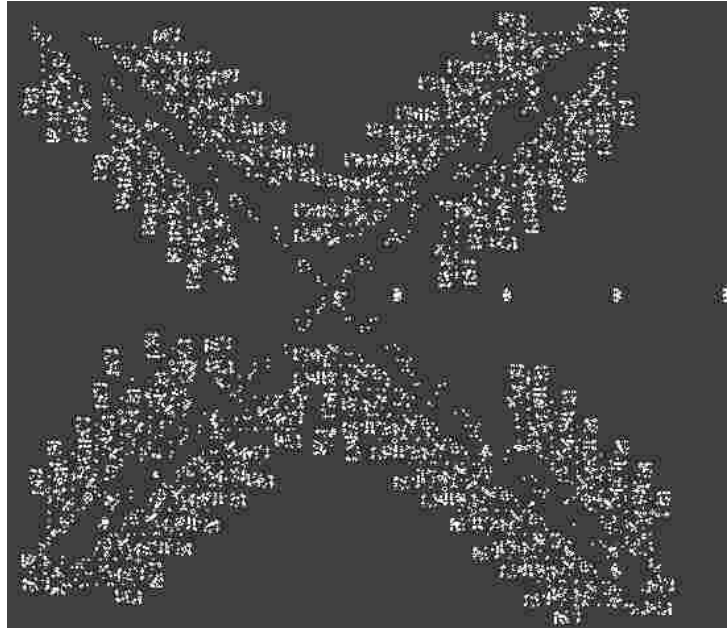
Hücresel otomatlar, 1940'lı yıllarda John von Neumann tarafından, matematikçi Stanislaw Ulam'ın önerisiyle yaratılmıştır. Von Neumann, kendiliğinden örgütlenmenin mantığını makine dilinde formüle etmeye çalışıyordu. Buna bir çözüm olarak hücresel otomatları kullanmıştır. (Mitchell 2009, 149). Von Neumann tarafından kurulan ilk hücresel otomatta, her bir hücre 29 ayrı değer alıyordu (modern hücresel otomat araştırmalarında her bir hücre genellikle 2 değer alır) ve bu sayede başlangıç durumunda hücrelere atanan değerler ve hücrelerin birbirleriyle kurdukları ilişki sonucu, sistem kendini yeniden üretebilmekteydi. Von Neumann'ın orjinal otomatının simülasyonu için *Golly* uygulaması kullanılabilir (<http://golly.sourceforge.net/>).

### 3.1 *Game of Life* (Yaşam Oyunu)

1970'de matematikçi John Conway *Game of Life*'ı (bundan sonra GoL denecek) icat etti. Bu otomat Neumann'inkine göre daha yalındı; hücreler iki değer alıyordu: 1 ve 0 ya da ölü ve canlı. Sistemin ve yani hücre değerlerinin güncellenmesinde, iki boyutlu uzayda bitişik karelerle temsil edilen hücrelerin çevresindeki 8 diğer hücreye bakılır. Sistemin adının yaşam oyunu olması bu bakımdan manidardır, zira Conway bu oyunu yaşamsal süreçlerle tarif etmiştir:

- *doğum*; ölü bir hücrenin eğer 3 komşusu varsa, gelecek adımda bu hücre *canlı* hale gelir.
- *hayatta kalma*; *canlı* bir hücrenin tam olarak 2 ya da 3 *canlı* komşusu varsa, bir sonraki adımda o hayatta kalır.
- *tekbaşınalık*; 2'den az komşusu olan *canlı* bir hücre ölür ve 3'den az komşusu olan *ölü* hücre *ölü* olarak kalır.
- *kalabalık*; *canlı* ve ya *ölü* bir hücrenin 3'ten fazla *canlı* komşusu var ise, o hücre ölür ya da ölü olarak kalır.

Conway bu hücresele otomatı ilginç ve yaşama benzeyen bir kural ararken bulmuştur (Mitchell 2009, 150). Zaman içinde GoL türetebildiği adeta canlı örüntülerle bir çok araştırmacı ve hobistin ilgi odağı olmuştur. Bu türden canlı örüntüler, sonsuz sayıdaki başlangıç durumlarına (canlı hücre popülasyonu ve dağılımı) GoL kurallarının uygulanmasıyla/denenmesiyle keşfedilmiştir. Yöntem, rastlantısal olarak popülasyonun tayin edilmesi ve sistemin evriminde bir periyodiklik gözlemlendiği zaman bunun çözülmesi biçiminde olmaktadır (Niemiec 2003). Ortaya çıkan örüntülerden en bilinenleri *gliders*, *guns*, *breeders*, *spaceships still-lives*'dir. Bunların genel özelliği, çeşitli modüllerden oluşmaları ve bunların oluşturduğu topluluğun tümünün ya da bir kısmının belli bir eksen boyunca hareket etmeleridir (*still-life* hariç). *Golly* adlı program paketi kullanılarak türetilen bir *spaceship-gun* örüntüsü aşağıdaki grafikte görülmektedir. Örüntünün dört kolu sabit durmakta iken, merkezden noktasal gruplar sağa doğru hareket etmektedir.



**Şekil 3.4** *Spaceship-gun* örüntüsü

Tez çalışmamızın sınırlarını aşmakla birlikte, hücresele otomatların işlemsel (*computational*) kullanımları olduğunu da zikretmemiz gerekmektedir. Bilgisayar bilimlerinde üzerinde en çok durulan Turing makinesinin, hücresele otomatlarda simülasyonu Wolfram tarafından gösterilmiştir. (Wolfram 2002, 78). Von Neumann da kendi hücresele otomatlar sisteminin evrensel Turing makinesine eşdeğer olduğunu gösterebilmiştir (Mitchell 2009, 151). John Conway de belli GoL başlangıç durumlarının bir program ve sistemin çalıştırılması sonucu çıkan örüntülerin de programın çıktısı olduğu durumları göstermiştir. Yani Conway, GoL'a dair karakteristik örüntülerin (*gliders, guns, vb*) mantıksal operatörler (**ve, ve ya, değil**) olarak nasıl kullanılabileceğini ispatlamıştır.

“...hücresele otomat fonksiyonlarının evrensel hesaplama kabiliyetine sahip olup olmamasının, artistik olarak faydalanılan, data/değer akışının kullanımıyla, doğrudan bir ilişkisi yoktur. ” (Ariza 2007, 29)

## **3.2 Hücresele Otomatların Kompozisyonel Kullanımı**

Hücresele otomatların müzikal uygulamalarını iki alt başlığa bölmek gerekmektedir. Bunlardan ilki, sembolik iletişim protokolü olan MIDI çıktısı üreten sistemler ve ikincisi ise sayısal ses sentezlemeye yönelik sistemlerdir.

### **3.2.1 MIDI hedefli hücresele otomat sistemleri**

80'li yıllarda kişisel bilgisayarların yaygınlaşmasıyla, sanatçıların algoritmik sistemleri üretimlerinde kullanmaları yaygınlaşmıştır. 2. Bölümde Xenakis'in hücresele otomatları, orkestral çalışmalarında nasıl kullandığını ele almıştık. Ayrıntılı araştırma için şu kaynaklara bakılabilir (Solomos 2005; Hoffmann 1994; Hoffmann

2002). Xenakis örneğinde, hücresele otomatlarla türetilen algoritmik data manuel olarak müzik yazısına aktarılmaktaydı. MIDI protokolünün yaygınlık kazanmasıyla bilgisayarda türetilen datanın ses sentezleyicilerini dolaysız olarak kontrol etmesi yaygınlık kazanmıştır. MIDI sembolik bir protokol olarak, sınırlı parametrik imkanlara sahipti ve dışsal ses üreteçlerine ihtiyaç duymakta idi. Ancak 80'li yılların ikinci yarısından itibaren, kişisel bilgisayarların işlemsel kapasitesinin artması ile dijital ses sentezleme uygulamaları yaygınlık kazanmaya başlamıştır. Bu uygulamalar (*Music V*, *Csound*, *CLM*) Max Mathews'in 50'li yıllardan itibaren Bell Laboratuvarlarında geliştirdiği "Music-N" uygulamalarına dayanmaktadır. Algoritmik datanın müzik yazısına aktarılmasında ise *Common Music* sisteminin bir parçası olan *Common Music Notation*, bu türden uygulamaların en eski örneklerindedir. Algoritmik müzik yazısında sunduğu imkanlarla *Lilypond* ve *FOMUS* kapsamlı özelliklere sahiptir.

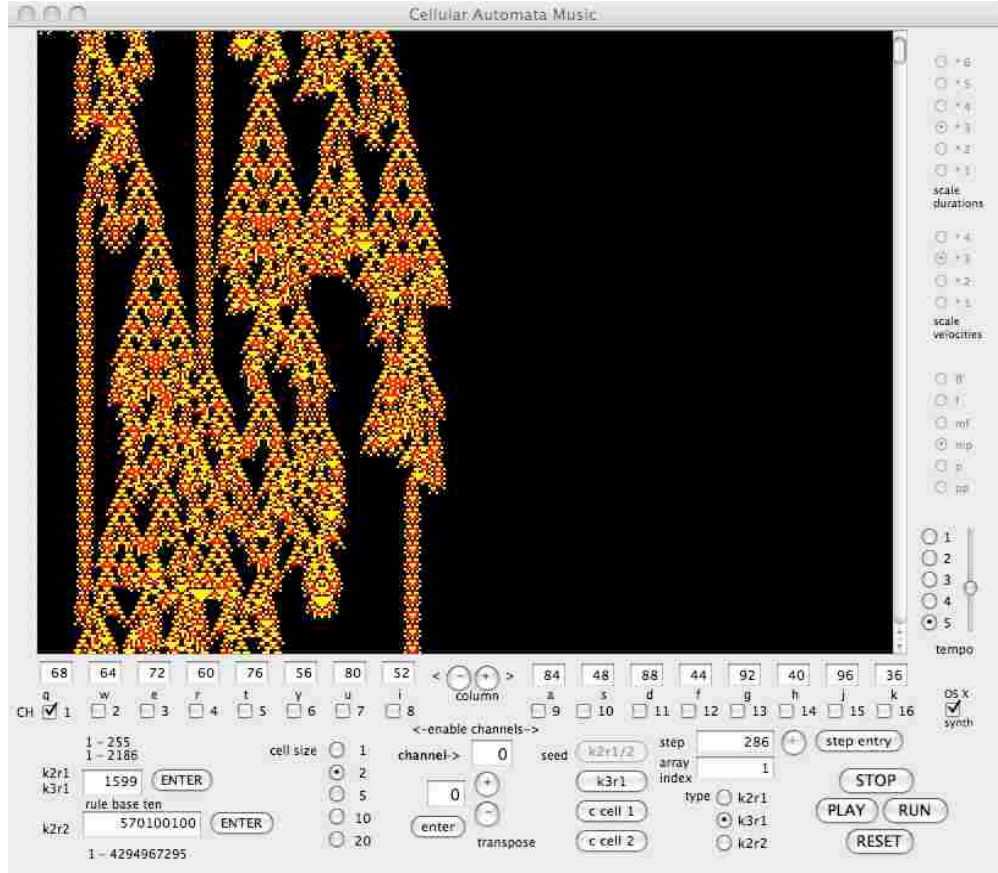
### 3.2.1.1 Peter Beyls

MIDI temelli hücresele otomat uygulamalarının iki öncüsü Peter Beyls (Burraston 2004) ve Dale Millen'dır (Millen 1990; Millen 1992). Beyls, geliştirdiği ilk hücresele otomat sistemlerinde MIDI protokolünü kullanarak Atari ve Yamaha şirketlerinin ticari ürünlerini ses kaynağı olarak kullanıyordu. Geliştirdiği hücresele otomatlar ve bu otomatların müziğe haritalanmasında çok özgün yöntemler geliştirmiştir. Tek ve iki boyutlu otomatlar kullanmıştır. Aynı zamanda, zaman içinde güncellenme kuralları değişen hücresele otomatlar geliştirmiştir. Bir diğer ilginç teknik ise, tek boyutlu otomatların iki boyutlu yorumlanması olmuştur; diyelim ki iki kuralımız var, bunlardan ilki tek boyutlu otomatı günceller, diğeri de tek boyutlu otomatın ilk kuralla aldığı biçimin evvelki ve ilerki durumlarını iki boyutlu olarak okumaktadır/değiştirmektedir. Diğer teknik denemeler ise, komşuluk ilişkilerinin sınanarak kural kümesinin değiştirilmesi ve geçmiş hücre sıralarının taranarak, bunlardan bazılarının hesaba katılması şeklindedir. Beyls, algoritmik datayı esnek

biçimlerde MIDI parametrelerine haritalamak ve hücrel otomatları canlı/etkileşimli olarak kullanmak maksadıyla bu teknikleri geliştirmiştir. Kısacası bu tekniklerde etkileşen öğeler, hücrel otomatın geriye dönük hesaplanması, kullanıcı/icracı tarafından belirlenen başlangıç durumu ve sistemin gerçek zamandaki durumudur. Datanın, çoklu MIDI kanallarına yönlendirilmesinde hücrelerin sıra numarası belirleyici olmaktadır. Eğer hücre sıra numaraları kanal sayısını aşıyorsa, hücre sıra sayısı kanal sayısı tabanına göre yorumlanmaktadır (Burraston 2004). Örneğin 8 sayısının 5 tabanına göre değeri 2 olur.

### **3.2.1.2 Dale Millen**

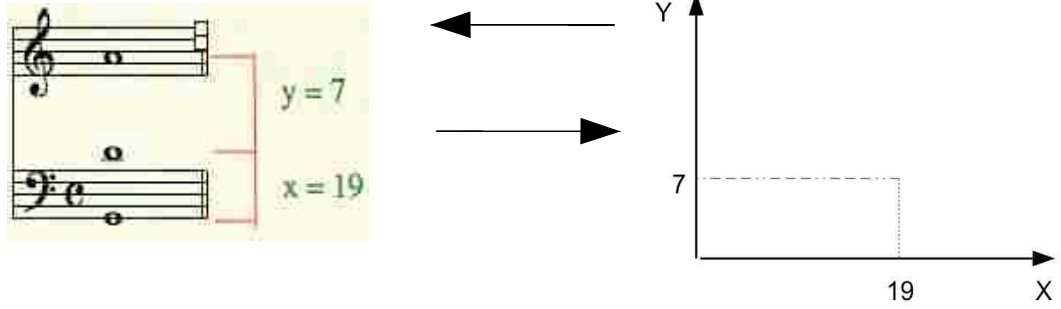
Dale Millen, 1988 yılından itibaren Cellular Automata Music (CAM) uygulamasını geliştirmeye başlamıştır. Mac OS X sisteminde çalıştırılan bu uygulama (**k** hücrelerin alabileceği değer sayısı ve **r** hücrelerin güncellenmesinde hesaba katılacak en yakın komşu sayısı olmak üzere)  $k3r1$ ,  $k2r2$  ve  $k2r1$  özelliğindeki kuralları kullanmaya imkan vermektedir. Kural numarası ve başlangıç durumu, kullanıcı tarafından belirlenir. Aynı zamanda, hücrelerin MIDI parametrelerine haritalanması da kullanıcı tarafından ayarlanabilmektedir. Otomat, bilgisayar ekranından grafik olarak akarken, data MIDI sinyali olarak istenilen modüle yönlendirilebilmektedir (<http://comp.uark.edu/~dmillen/cam.html>).



Şekil 3.5 Cellular Automata Music uygulamasının kullanıcı arabirimi

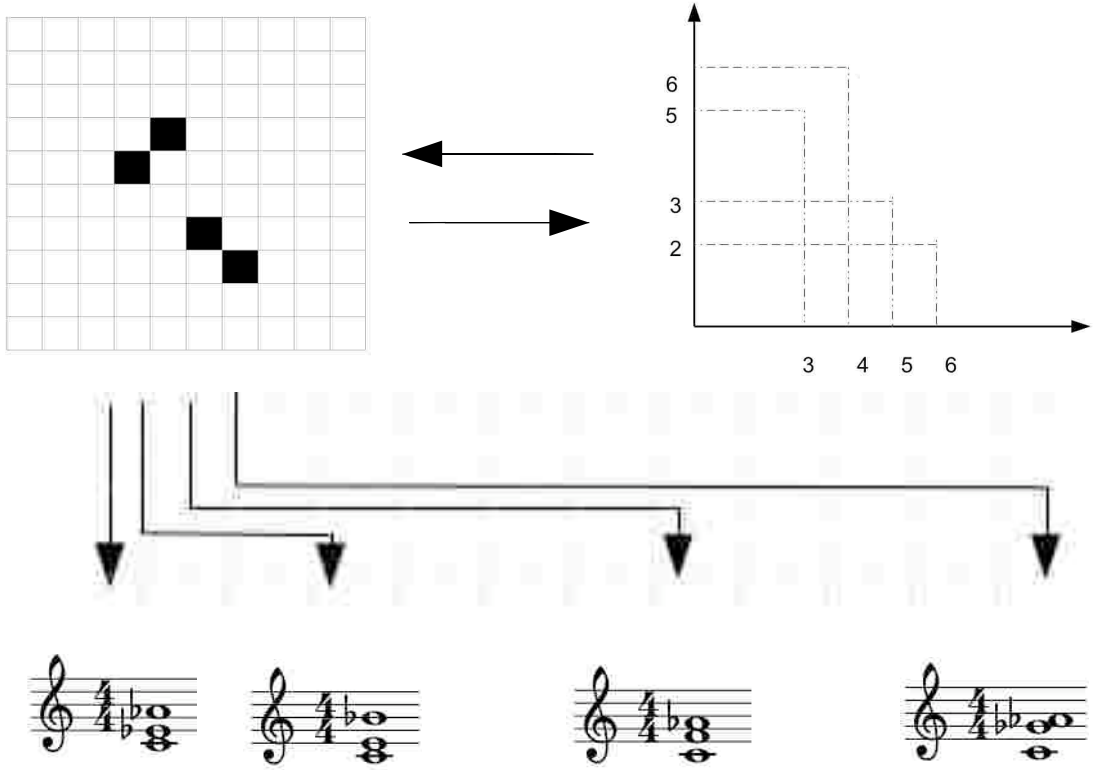
### 3.2.1.3 Eduardo Reck Miranda ve *Camus* Uygulaması

Eduardo Reck Miranda tarafından geliştirilen *Camus* uygulamasında, GoL datası (yani yaşayan hücrelerin koordinatları), kartezyen modellemeyle, 3 ayrı nota tayin eden bir biçimde okunmaktadır. Bu 3 notanın aynı anda mı tınlayacağı, başka bir karar sürecine bağlıdır. Bu üç nota, müzik terminolojisindeki aralık ilişkileriyle tayin edilmektedir. Her bir birim, küçük ikili aralığına denk düşmektedir. Dikey koordinat, ilk aralığı, yatay koordinat ise ikinci aralığı temsil etmektedir.



**Şekil 3.6** *Camus* uygulamasının hücresel otomat datasından akor türetme yöntemi

Hücresel otomat, rastlantısal olarak atanan hücre konfigürasyonu ile başlatılır ve GoL kuralları ile hücre değerleri güncellenir. Sistemin sürerliğinde, yaşayan her hücrenin (yani 1 değerini alan) koordinatından yukarıdaki yöntemle iki aralık çıkarılır ve kullanıcı tarafından belirlenen olası nota kümesinden bir nota seçilerek, bu temel üzerine aralıklar uygulanarak çok sesli bir doku oluşturulur. Temel notanın seçiliminde çeşitli olasılık değerleri girilerek, seçilmesi istenen kimi notalara yüksek ihtimal atanabilir. Aşağıda bir GoL konfigürasyonu ve bunun kartezyen modeli aşağıda verilmiştir.



**Şekil 3.7** Temel olarak seçilen Do notası üzerine *GoL*'dan türetilen akorlar

*Camus* sisteminde, yukarıda ayrıntılandırdığımız *GoL* kullanımının yanında, *Miranda*'nın *Demon Cyclic Space* adını verdiği bir başka hüresel otomat sistemi de kullanılmaktadır. Bu otomatta hücreler 7 ayrı değer alır ve herbir adımda hücre değerini bir derece yükseltir. Rastlantısal olarak atanan hücre değerleriyle sistem başlatılır ve her bir hücre için değerler dögüsel olsa da makro düzeyde kompleks bir dinamizm ortaya çıkar. Bu otomat *GoL*'dan alınan nota yüksekliklerinin hangi çalgısal renklerle çalınacağını belirlemektedir (*Miranda* 2001, 126).

### 3.2.1.4 Warren Kurt

Besteci Warren Kurt 1970'lerin başından bu yana kendi geliştirdiği etkileşimli sistemlerle müzik icra etmektedir. Müziklerinde hücresele otomatları kullanmaktadır.

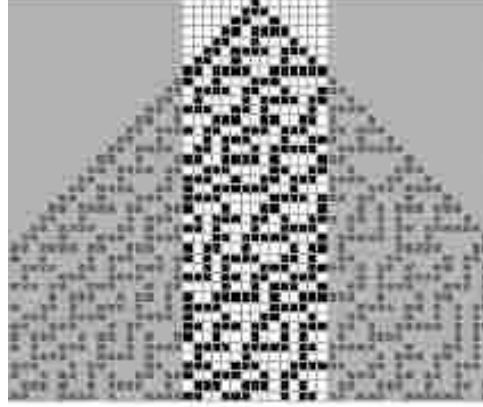
“...*randie* ile çeşitli denklemler kurdum..Bunlardan biri, 1994 yılından etkileşimli bilgisayar müziği olan *Tuning the Furniture of Chaos*'dur; bu kompozisyonda, yaklaşık olarak her 90 saniyelik kısımda şu denklemlerden biri kullanılmıştır: Henon çekeri (*attractor*), hücresele otomatlar, ve birbirine yönlendirilmiş ve farklı limit ve tınılara uygulanan bir dizi sayaç. Hücresele otomatlar, hep, yavaş tempoda piyano tınısına yönlendirilmiştir..” (Burt 1996, 169)

Warren Burt'un betimlediği kendi sisteminde hücresele otomatlar ve diğer algoritmik sistemler öngörülemez ama deterministik sayı akışını sağlamaktadır. Temelde besteci, teknik olarak rastlantısallığı kullanmaktadır, algoritmik sistemler bu bakımdan rastlantısal sayı üreticileri işlevine büründürülmüştür.

### 3.2.1.5 “WolframTones” ve Birebir Haritalama

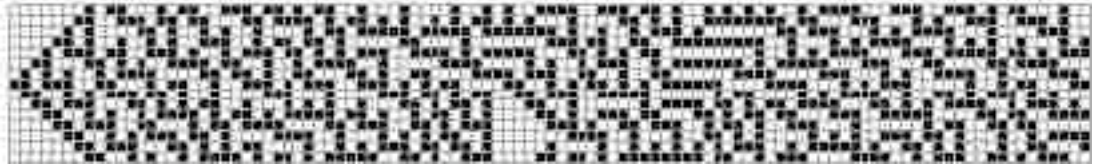
Hücresele otomatların matematiksel özelliklerini araştıran Stephen Wolfram'ın *Mathematica* adlı uygulaması, kendi uygulamalarımızda birincil data üreticimiz olarak kullanıldı. *Mathematica*'nın web tabanlı hücresele müzik üreticisi olan “WolframTones” adlı uygulamayı ve hücre değeri ve boyutlarının müzikal parametrelere haritalanmasını ele alalım.

Wolfram'ın hücresele otomat kullanımında iki boyutlu grafik datanın ortasından boyut boyunca bir kesit alınır.



**Şekil 3.8 Hücresel otomat datasından kesit alınması**

Bu kesit soldan sağa okunacak şekilde döndürülür (90 derece sola).



**Şekil 3.9 Alınan kesitin rotasyona uğratılması**

Bu grafiğin dikey vektörü seçilen ses dizisine haritalanır, yatay vektör ise zamansal adım olarak yorumlanır. Her bir zamansal adımda, dikey vektördeki siyah (1 değerini alan) hücrelerin denk düştüğü nota seslendirilir. Eğer bu grafik, birden fazla çalgıya aktarılıyorsa, birbiriyle (çapraz ya da yatay) temas halindeki siyah hücreler, aynı çalgı tarafından seslendirilir. Web tabanlı bu sistemde,  $r_1$  ve  $r_2$  özellikli hücresel otomat kuralları çalıştırılabilir.

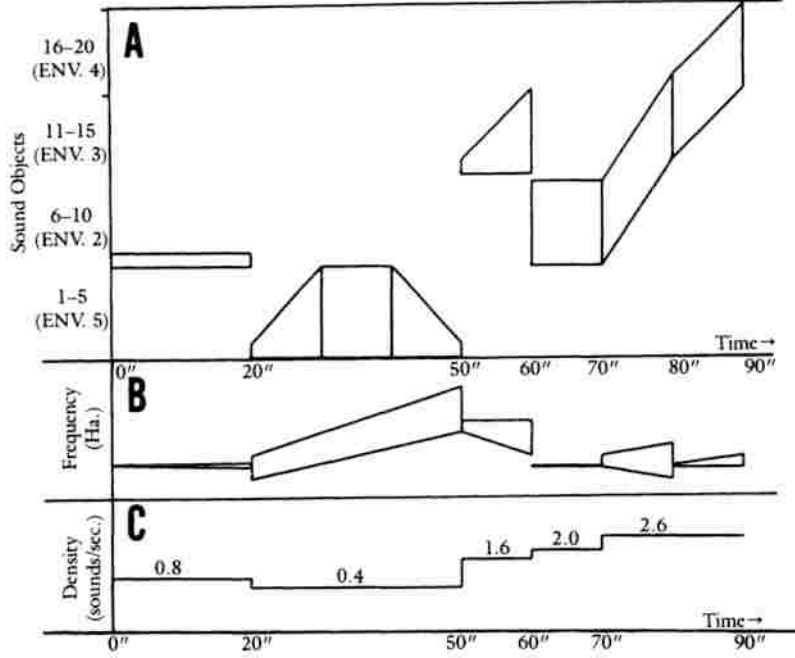
### 3.3 Ses Sentezlenmesinde Hücresel Otomatların Kullanımı

MIDI arabirimli sistemlere göre ses sentezleyen sistemlerin daha sonra geliştirilmeye başlanmış olması, bilgisayarların işlem kapasitesindeki gelişmeyle paralellik göstermektedir. Hücresel otomatların ses sentezlenmesinde kullanımı da bu yüzden 1980'lerin sonlarından itibaren başlamıştır.

#### 3.3.1 Granüler Sentezleme

Hücresel otomatlar, küçük birimlerin oluşturduğu dinamik bir sistem olarak, granüler sentezleme için çok uygun bir model oluşturmaktadır. Tarihte ilk defa fizikçi Denis Gabor ses olgusunu parçacıklar (*grains*) üzerinden açıklayan bir kuram geliştirmiştir (Gabor 1947). Bu kuramı kompozisyonel bir strateji olarak ilk kullanan ise Xenakis'tir (Xenakis 1992). İlk granüler sentezleme uygulamasını Curtis Roads programlamıştır. O tarihte kullandığı ses programlama platformu *Music V* idi (Roads 1978).

Granüler sentezlemede, sonuç, 10-20 milisaniye uzunluğundaki ses birimlerinin sıralanmasıyla elde edilir. Sentezleme için ses bandı kullanılabileceği gibi, sinüs dalgası gibi trigonometrik fonksiyonlara dayanan ses kaynakları da bu teknikle dönüşüme uğratılabilir. Granüler sentezlemede kullanılan başlıca parametreler; parçacığın süresi (ms ile ölçülür) ve parçacıklar arası yoğunluktur (Hz ile ölçülür). Bu parametrelerin dinamik olarak kontrolü sonucu kompleks tınılar elde edilebilmektedir (*tendency masks*). Aşağıdaki örnekte iki ana parametre olan yoğunluk ile sürenin ve ses parçacıklarının alınacağı ayrı ses örneklerinin zaman içindeki parametrik kontrolü verilmiştir (Truax 1990, 122). Grafiğin A kısmı tını seçimini, B kısmı parçacıklar arası yoğunluğu/sıklığı ve C kısmı da parçacıkların süre değerlerini vermektedir.



**Şekil 3.10** Tını seçimi (üst), frekans aralığı (orta), ve zamansal yoğunluk (alt) için olasılıksal eğilim tablosu (*tendency masks*). (Truax 1990, 122)

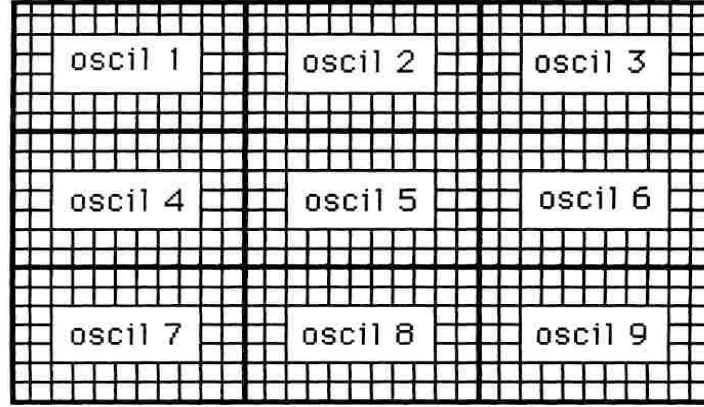
Hüresel otomatlar kullanılarak granüler sentezlemenin yönetilmesinde standart bir teknik, Edinburgh Parallel Computing Centre'da Eduardo Reck Miranda tarafından geliştirilmiştir (Miranda 1995). Adı *ChaOs* olan bu sistem esinini, kimyasal reaksiyonları modellemek üzere geliştirilmiş ve Belousov-Zhabotinskii adıyla bilinen bir otomattan almaktadır (Miranda 2007a, 175). Bu otomat, Miranda tarafından, nöronların elektriksel salınımına benzeyen örüntüler elde etmek amacıyla dönüştürülmüştür. Her bir hücre bir nöron olarak düşünülmüştür. Hücrelerin her bir zamansal adımda alabileceği durumlar şöyledir: (a) pasif, (b)  $n$  ile nicelliği ifade edilen depolarize olma durumlarından birinde, (c) ateşlenmiş. Her bir nöron, aralarında akan elektriksel akım sayesinde komşularıyla etkileşime girmektedir. Bu

etkileşimde, alt ( $V_{min}$ ) ve üst ( $V_{max}$ ) eşik değerleri her bir hücrenin alacağı durumları belirleyen unsurdur. Eğer hücredeki iç gerilim ( $V_i$  [voltage]) alt eşik değerinden ( $V_{min}$ ) küçük ise, o hücre pasiftir.



**Şekil 3.11** *Chaosynth* kullanıcı arabirimi

Yukarıdaki şekilde görülebileceği üzere, kullanıcı hücresel otomatın çeşitli özelliklerini kendisi belirleyerek süreci başlatmaktadır. Hücrelerin alanı, eşit parçalara bölünerek, bölünmüş alanın her biri ayrı bir osilatöre yönlendirilmektedir.



Example grid = 21 x 33 cells  
Each oscillator = 7 x 11 cells

### Şekil 3.12 *Chaosynth*'te hücresel otomat datasının farklı osilatörlere tayin edilme yöntemi

*ChaOs*'un bir diğer özelliği de, granular sentezleme ile elde edilen tınının, klasik analog tekniklerle yeniden işlenebilme imkanı sunmasıdır. Analog tekniklerle kastımız, filtreler, ring modülatörleri, envelope üreteçleri, LFO gibi uygulamalardır.

### 3.3.2 Karplus-Strong tipi Sentezleme

1990 yılında Jacques Chayeron tarafından geliştirilen *LASy* (Lineer Otomat Sentezleme), hücresel otomatları kullanarak Karplus-Strong tipi sentezleme algoritmasını yönlendirmektedir. Bu sistem Milan Üniversitesinde *Intelligent Music Workstation* projesinin bir parçası olarak ortaya çıkmıştır (Miranda 2007a, 179). Karplus-Strong tipi sentezleme, zamansal süreçlerle, çekilerek çalınan telli çalgıların tınısını üretmektedir. *LASy* sisteminde, tek boyutlu hücresel otomatın evrimi sentezlemeyi yönlendirmektedir. Bu sayede *LASy* değişik karakterlerde tınılar üretebilmektedir. Chayeron'a göre bu tını karakterleri kabaca üç başlıkta toplanabilir:

- Basitçe evrilerek, yalın bir duruma geçen tınılar: bu türden tınıları doğuran hücrenel otomatlar, yukarda belirttiğimiz 4 ana tipten sabit ve döngüsel karakterli olanlardır. Bu karakterdeki otomatlar, yeknesakça evrilen tınılar doğurur; ses spektrumundaki değişimler çıkıcı ya da inici bir eğri izler ve sonunda statik bir duruma bağlanır.
- Bir sona bağlanmayan ve basitçe evrilen tınılar: bu tınıları türeten hücrenel otomatların karakteri kompleksdir, ve sürekli birbirine benzeyen ancak tamamen aynı da olmayan eğriler türetirler.
- Sürekli evrilen kompleks tınılar: bu tınıları türeten hücrenel otomat kuralları kaotik/rastlantısal karakterdedir, ve biteviye öngörülemeyen tınısal spektrumlar doğururlar.

### 3.4 Otomat Nesillerinin İstatistiki ve Esnek Yorumu

Wolfram'ın uygulamasında (bkz. Bölüm 3.2.1.5), kulağa çarpan tekdüze zaman kuruluşu ve grafik analojiyi aşabilmek için hücrenel otomatları yorumlayan farklı bir yol geliştirdik. Kendi yöntemimizde, herbir dikey vektördeki 1 değerini alan hücrelere buldukları konuma göre sıra numarası verilir ve bu değer, seçilen dizide bir notaya denk gelir. Ancak bu notalar, yatay vektörün ilgili zamansal adımında aynı anda seslendirilmezler. İlgili adımda 1 değerini alan hücrelerin niceliği, birim zamanın kaç bölüneceğini tayin eder. Bu zamansal birim içinde, sıra numarası almış notalar rastlantısal bir sırayla seslendirilir. Yani, grafik olarak düşünüldüğünde yatay vektördeki her bir adım, süre birimidir; bu birimi 1 değerini alan hücrelerin denk düştüğü notalar doldurur. Dolayısıyla, her bir zamansal adım, farklı kesirlerle bölünmüş birbirine süre olarak eşit zamansal dilimlerdir. Sıralı listeler olarak alınan hücrenel otomat dadasında her bir dikey vektördeki 1 değerini alan hücrelerin numaralanması için aşağıdaki *Scheme* kodu yazıldı.

```
(define (indexer-of-1 liste)
  (loop for A below (length liste)
        collect
        (loop for i below (length (nth liste A))
              when (= (nth (nth liste A) i) 1)
                    collect i)))
```

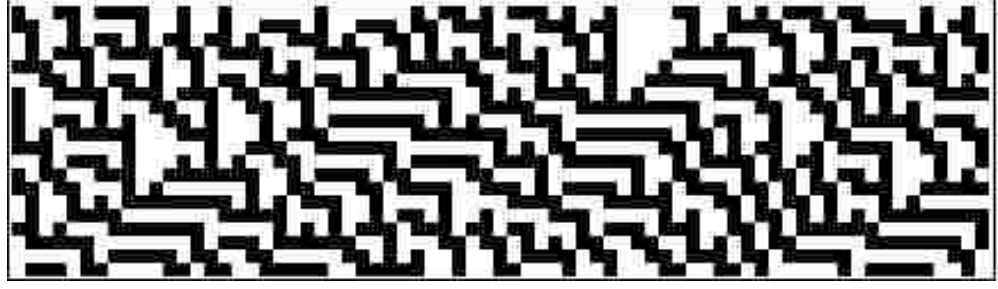
Bu kod sayesinde 0 ve 1 değerlerini taşıyan listelerden özütlenen veride, her bir iç liste yalnızca 1 değerini almış hücrelerin sıra numaraları ile değiştirilmiştir. Örneğin argüman olarak girilen ((0 1 0 1) (1 0 1 1 0 0)) şeklindeki sıralı bir liste, ((1 3) (0 2 3)) olarak çıkmaktadır.

Bu hüresel otomat uygulamasında, data *Mathematica* kullanılarak üretilmektedir. Türetilen data, sıralı listeler olarak *Grace* adlı uygulamada işlenmekte, MIDI dosyası olarak ya da *Fonus* arabirimiyle partiyon olarak çıktısı alınmaktadır. Hüresel otomat üretmek için *Mathematica* bir çok avantaj sunmaktadır:

- Bütün temel komşuluk ilişkilerine dayalı otomatların kuralları standart olarak sistemde içerilmektedir.
- Üretilen data grafik olarak ya da liste olarak alınabilmektedir.

Aşağıdaki kod, Wolfram'ın numaralama sistemine göre 30 no'lu kuralı, dikey eksenindeki 20 hücreye uygular. Başlangıç durumu rastlantısal olarak tayin edilir, ve sistem, hücreleri 70 kez günceller.

```
> ArrayPlot[CellularAutomaton[30, RandomInteger[1, 20], 70]]
```



**Şekil 3.13 Rastlantısal başlangıç durumuyla başlatılan, Kural No:30 otomatı**

Bu datanın sıralı listeler olarak yorumlanması ve partisyona aktarılması için aşağıdaki *Scheme* kodunu yazdık.

```
(define (int-output ca-list bottom top)
  (let ((output (indexer-of-1 ca-list))
        (in-len (length (first ca-list))))
    (loop for i from 0 below (* 2 (length output)) by 2
          for i-r from 0
          for ka = (make-heap (nth output i-r))
          for kl = (length (nth output i-r))
          do
            (loop for ki1 from i below (+ i 2) by (/ 2 kl)
                  for key = (next ka)
                  do
                    (send "fms:note" :time ki1 :pitch (floor (rescale key 0 in-len bottom
top)) :dur (/ 2 kl)
                          :part "pi1")
                    (send "mp:midi" :time ki1 :key (floor (rescale key 0 in-len bottom top))
                          :dur (/ 2 kl)))
                  finally (send "fms:run")))))
```

Bu kodda zamansal birim 2'lik (♩) notadır. Herbir 2'lik nota, o zamansal adımdaki 1 değerini almış notaların niceliğine bölünmektedir. 30 no'lu kural kullanarak üretilen müziğin *Fomus* çıktısını aşağıda veriyorum.

Piano

15 16 17 18

19 20 21 22

23 24 25 26

27 28 29 30

31 32 33 34

System 1: Treble and bass clefs. Treble clef has a dotted line labeled 'Sca' above it. Measures 14, 19, and 20 are indicated below the staff.

System 2: Treble and bass clefs. Measures 11, 19, and 20 are indicated below the staff.

System 3: Treble and bass clefs. Measures 10, 12, and 14 are indicated below the staff.

System 4: Treble and bass clefs. Measures 22, 27, and 28 are indicated below the staff.

System 5: Treble and bass clefs. Measures 9, 17, and 25 are indicated below the staff.

System 6: Treble and bass clefs. Treble clef has a dotted line labeled 'Sca' above it. Measures 21, 27, and 28 are indicated below the staff.

34 *Stra*

15 17 17

This system contains measures 34, 35, and 36. Measure 34 is marked with a *Stra* dynamic. The music features a complex texture with multiple voices in both the treble and bass staves, including some sixteenth-note passages. Measure numbers 15, 17, and 17 are indicated below the bass staff.

37 *Stra*

21 24

This system contains measures 37, 38, and 39. Measure 37 is marked with a *Stra* dynamic. The notation includes various rhythmic patterns and accidentals. Measure numbers 21 and 24 are indicated below the bass staff.

40 *Stra*

13 19

This system contains measures 40, 41, and 42. Measure 40 is marked with a *Stra* dynamic. The music continues with intricate textures and rhythmic complexity. Measure numbers 13 and 19 are indicated below the bass staff.

43 *Stra*

13

This system contains measures 43, 44, and 45. Measure 43 is marked with a *Stra* dynamic. The notation shows dense chordal textures and melodic lines. A measure number 13 is indicated below the bass staff.

46 *Stra*

12 14

This system contains measures 46, 47, and 48. Measure 46 is marked with a *Stra* dynamic. The music features a mix of rhythmic values and complex voicings. Measure numbers 12 and 14 are indicated below the bass staff.

40 *Sforz.* 20

18

This system contains measures 40, 41, and 42. Measure 40 is marked with a forte dynamic (*Sforz.*) and a hairpin crescendo leading to a forte dynamic (**f**) in measure 41. Measure 42 continues the melodic line. The bass line features a steady eighth-note accompaniment.

43 *Sforz.* 21

17

This system contains measures 43, 44, and 45. Measure 43 is marked with a forte dynamic (*Sforz.*) and a hairpin crescendo leading to a forte dynamic (**f**) in measure 44. Measure 45 continues the melodic line. The bass line features a steady eighth-note accompaniment.

46 *Sforz.* 22

17

This system contains measures 46, 47, and 48. Measure 46 is marked with a forte dynamic (*Sforz.*) and a hairpin crescendo leading to a forte dynamic (**f**) in measure 47. Measure 48 continues the melodic line. The bass line features a steady eighth-note accompaniment.

49 23

17

This system contains measures 49, 50, and 51. Measure 49 is marked with a forte dynamic (**f**) and a hairpin crescendo leading to a forte dynamic (**f**) in measure 50. Measure 51 continues the melodic line. The bass line features a steady eighth-note accompaniment.

52 *Sforz.* 24

11

11

This system contains measures 52, 53, and 54. Measure 52 is marked with a forte dynamic (*Sforz.*) and a hairpin crescendo leading to a forte dynamic (**f**) in measure 53. Measure 54 continues the melodic line. The bass line features a steady eighth-note accompaniment.

55 *Sforz.* 25

17

This system contains measures 55, 56, and 57. Measure 55 is marked with a forte dynamic (*Sforz.*) and a hairpin crescendo leading to a forte dynamic (**f**) in measure 56. Measure 57 continues the melodic line. The bass line features a steady eighth-note accompaniment.

60 *Stra* 27 28 *Stra*

This system contains measures 60 to 62. The right hand features a melodic line with a trill in measure 60 and a descending eighth-note scale in measure 61. The left hand provides a steady eighth-note accompaniment. Measure numbers 27 and 28 are indicated below the bass staff.

63 *Stra* 25 26

This system contains measures 63 to 65. The right hand has a melodic line with a trill in measure 63 and a descending eighth-note scale in measure 64. The left hand continues with an eighth-note accompaniment. Measure numbers 25 and 26 are indicated below the bass staff.

66 12 13 14 15 17

This system contains measures 66 to 68. The right hand features a melodic line with a trill in measure 66 and a descending eighth-note scale in measure 67. The left hand has an eighth-note accompaniment. Measure numbers 12, 13, 14, 15, and 17 are indicated below the bass staff.

69 *Stra* 19 20 21 22

This system contains measures 69 to 71. The right hand has a melodic line with a trill in measure 69 and a descending eighth-note scale in measure 70. The left hand continues with an eighth-note accompaniment. Measure numbers 19, 20, 21, and 22 are indicated below the bass staff.

72 *Stra* 17 23 24

This system contains measures 72 to 74. The right hand features a melodic line with a trill in measure 72 and a descending eighth-note scale in measure 73. The left hand has an eighth-note accompaniment. Measure numbers 17, 23, and 24 are indicated below the bass staff.

6

81 *Sea* 13 *Sea* 18

This system contains measures 81 to 83. The right-hand part features a melodic line with a dotted line above it labeled "Sea" that spans measures 81 and 82. The left-hand part provides a harmonic accompaniment. Measure numbers 13 and 18 are indicated below the staff.

84 *Sea* 20 22

This system contains measures 84 to 86. The right-hand part has a melodic line with a dotted line above it labeled "Sea" spanning measures 84 and 85. The left-hand part continues the accompaniment. Measure numbers 20 and 22 are indicated below the staff.

87 14 18 14

This system contains measures 87 to 89. The right-hand part has a melodic line with a dotted line above it labeled "Sea" spanning measures 87 and 88. The left-hand part continues the accompaniment. Measure numbers 14, 18, and 14 are indicated below the staff.

90 17 18

This system contains measures 90 to 92. The right-hand part has a melodic line with a dotted line above it labeled "Sea" spanning measures 90 and 91. The left-hand part continues the accompaniment. Measure numbers 17 and 18 are indicated below the staff.

93 *Sea* 13 12

This system contains measures 93 to 95. The right-hand part has a melodic line with a dotted line above it labeled "Sea" spanning measures 93 and 94. The left-hand part continues the accompaniment. Measure numbers 13 and 12 are indicated below the staff.

System 1: Measures 97-100. Treble clef, bass clef. Measure numbers 97, 98, 99, 100 are indicated. A slur covers measures 97-99. A 'Star' marking is above measure 98. A '20' marking is above measure 97. A '22' marking is above measure 99. A '21' marking is below measure 100.

System 2: Measures 101-104. Treble clef, bass clef. Measure numbers 101, 102, 103, 104 are indicated. A 'Star' marking is above measure 101. A '23' marking is below measure 102. A '24' marking is below measure 103. A '25' marking is below measure 104.

System 3: Measures 105-108. Treble clef, bass clef. Measure numbers 105, 106, 107, 108 are indicated. A 'Star' marking is above measure 105. A '26' marking is below measure 106. A '27' marking is below measure 107. A '28' marking is below measure 108.

System 4: Measures 109-112. Treble clef, bass clef. Measure numbers 109, 110, 111, 112 are indicated. A 'Star' marking is above measure 109. A '29' marking is below measure 110. A '30' marking is below measure 111. A '31' marking is below measure 112.

System 5: Measures 113-116. Treble clef, bass clef. Measure numbers 113, 114, 115, 116 are indicated. A 'Star' marking is above measure 113. A '32' marking is below measure 114. A '33' marking is below measure 115. A '34' marking is below measure 116.

System 6: Measures 117-120. Treble clef, bass clef. Measure numbers 117, 118, 119, 120 are indicated. A 'Star' marking is above measure 117. A '35' marking is below measure 118. A '36' marking is below measure 119. A '37' marking is below measure 120.

The image displays a musical score for a piano piece, identified as Kural No.30. The score is presented in three systems, each with a treble and bass clef. The key signature is two flats (B-flat and E-flat). The first system (measures 114-116) shows a melodic line in the treble and a bass line in the bass. The second system (measures 117-119) continues the piece with similar notation. The third system (measures 120-121) concludes the piece with a double bar line. The score includes various musical notations such as notes, rests, and dynamic markings like 'Sforzando'.

Şekil 3.14 Kural No.30 nota çıktısı (Bkz. Ek-CD, Parça No. 2)

### 3.4.1 Süreç İçinde Hücresel Otomat Kurallarının Değiştirilmesi

Yukarıdaki nota örneğinden anlaşılabilir olduğu üzere, data, baştan sona aynı dokusal karaktere sahiptir. Müzikal zamanın kuruluşunda, değişik dokuların ardardalığı için, farklı karakterdeki hücresel otomat kurallarını kullanmak gerekmektedir. Belirlenen zamansal adımlarda hücrelerin güncellenme kuralları, değiştirilir, ve sistem kaldığı noktadan yeni kurallarla güncellenir. Bu sayede, yukarıda ele aldığımız 4 temel hücresel otomat karakteristiğini, müzikal bestelemeye birbirleriyle bağlantılı olarak temsil etme imkanı doğmaktadır.

Farklı karakterlerdeki hücresel otomatları seri halde ve tek bir evrimsel süreç içinde türetebilmek için aşağıdaki *Mathematica* kodu yazıldı.

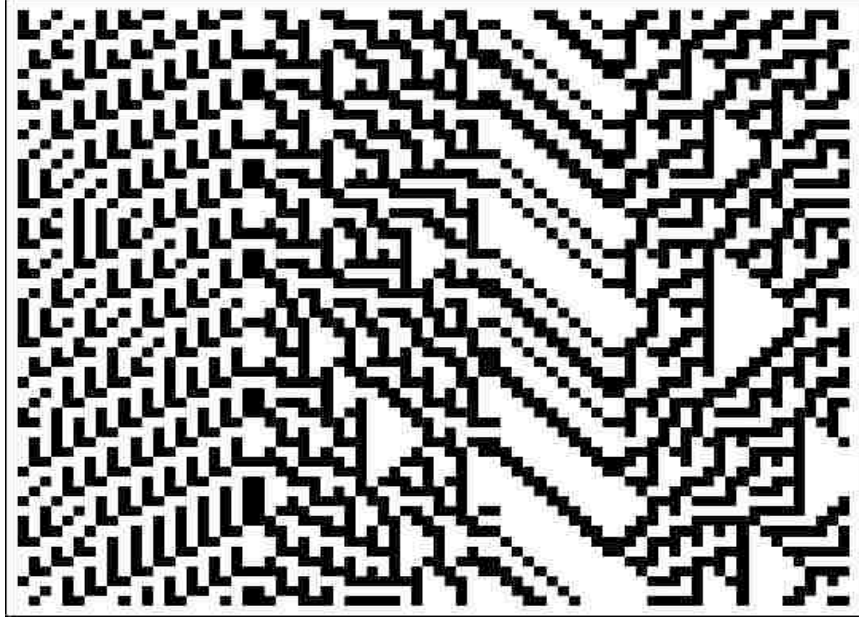
```
> RandInit = RandomInteger[1, 60]
> ArrayPlot[Join
             [CellularAutomaton[25,RandInit,20],
              CellularAutomaton[30,Last[CellularAutomaton[25,RandInit,20]],20],
```

```
CellularAutomaton[10,Last[CellularAutomaton[30,Last[CellularAutomaton[25,
RandInit, 20]], 20]],10],
```

```
CellularAutomaton[86,Last[CellularAutomaton[10,Last[CellularAutomaton[30,Last
[CellularAutomaton[25,RandInit,20]],20]],10]], 20]]]
```

Yukarıdaki kodda, evrim rastlantısal olarak belirlenen bir başlangıç durumundan başlatılmıştır. Sırasıyla, 25 nolu kural (döngüsel davranış), 30 nolu kural (kaotik davranış), 10 nolu kural (sabit davranış) ve en son da 86 nolu kural (kompleks davranış) evrimi yönlendirmektedir. 10 nolu kural hariç bütün kurallar, 20 zamansal adımı kontrol etmektedir. 10 nolu kural ise yalnız 10 adım boyunca evrime hakim

olur. Bu tercih, 10 nolu kuralın tekrarlı karakterde olmasına dayanmaktadır. Yukarıdaki kodun ürettiği datanın grafik çıktısı aşağıdadır.



**Şekil 3.15** 25, 30, 10, ve 86 no'lu kuralların seri halde kullanımı

Piano

Musical score for piano, measures 1-22. The score is written in treble and bass clefs, 2/4 time signature. The key signature has one sharp (F#). The music features a melodic line in the right hand and a supporting bass line in the left hand. Measure numbers 1, 11, and 22 are indicated below the staff.

Musical score for piano, measures 23-32. The score continues from the previous system. Measure numbers 23 and 32 are indicated below the staff.

Musical score for piano, measures 33-42. The score continues from the previous system. Measure numbers 33 and 42 are indicated below the staff.

Musical score for piano, measures 43-52. The score continues from the previous system. Measure numbers 43 and 52 are indicated below the staff.

Musical score for piano, measures 53-62. The score continues from the previous system. Measure numbers 53 and 62 are indicated below the staff.

2

Musical notation for measures 2-21. The system consists of two staves (treble and bass clef). Measure 2 is marked with a '2'. Measure 10 is marked with a '10'. Measure 21 is marked with a '21'. The notation includes various notes, rests, and accidentals.

Musical notation for measures 12-26. The system consists of two staves (treble and bass clef). Measure 12 is marked with a '12'. Measure 26 is marked with a '26'. The notation includes various notes, rests, and accidentals.

Musical notation for measures 14-30. The system consists of two staves (treble and bass clef). Measure 14 is marked with a '14'. Measure 30 is marked with a '30'. The notation includes various notes, rests, and accidentals.

Musical notation for measures 16-27. The system consists of two staves (treble and bass clef). Measure 16 is marked with a '16'. Measure 27 is marked with a '27'. The notation includes various notes, rests, and accidentals.

Musical notation for measures 18-28. The system consists of two staves (treble and bass clef). Measure 18 is marked with a '18'. Measure 28 is marked with a '28'. The notation includes various notes, rests, and accidentals.

3

20

26

26

27

24

29

20

21

26

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

509

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525

526

527

528

529

530

531

532

533

534

535

536

537

538

539

540

541

542

543

544

545

546

547

548

549

550

551

552

553

554

555

556

557

558

559

560

561

562

563

564

565

566

567

568

569

570

571

572

573

574

575

576

577

578

579

580

581

582

583

584

585

586

587

588

589

590

591

592

593

594

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

649

650

651

652

653

654

655

656

657

658

659

660

661

662

663

664

665

666

667

668

669

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

688

689

690

691

692

693

694

695

696

697

698

699

700

701

702

703

704

705

706

707

708

709

710

711

712

713

714

715

716

717

718

719

720

721

722

723

724

725

726

727

728

729

730

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752

753

754

755

756

757

758

759

760

761

762

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

810

811

812

813

814

815

816

817

818

819

820

821

822

823

824

825

826

827

828

829

830

831

832

833

834

835

836

837

838

839

840

841

842

843

844

845

846

847

848

849

850

851

852

853

854

855

856

857

858

859

860

861

862

863

864

865

866

867

868

869

870

871

872

873

874

875

876

877

878

879

880

881

882

883

884

885

886

887

888

889

890

891

892

893

894

895

896

897

898

899

900

901

902

903

904

905

906

907

908

909

910

911

912

913

914

915

916

917

918

919

920

921

922

923

924

925

926

927

928

929

930

931

932

933

934

935

936

937

938

939

940

941

942

943

944

945

946

947

948

949

950

951

952

953

954

955

956

957

958

959

960

961

962

963

964

965

966

967

968

969

970

971

972

973

974

975

976

977

978

979

980

981

982

983

984

985

986

987

988

989

990

991

992

993

994

995

996

997

998

999

1000

Musical notation for measures 27-35. The system consists of a grand staff with a treble clef on the upper staff and a bass clef on the lower staff. Measure numbers 27, 30, and 33 are indicated at the beginning of their respective measures. The music features a melodic line in the treble clef and a supporting bass line in the bass clef.

Musical notation for measures 36-44. The system consists of a grand staff with a treble clef on the upper staff and a bass clef on the lower staff. Measure numbers 36, 39, and 42 are indicated at the beginning of their respective measures. The music continues with a melodic line in the treble clef and a supporting bass line in the bass clef.

Musical notation for measures 45-53. The system consists of a grand staff with a treble clef on the upper staff and a bass clef on the lower staff. Measure numbers 45, 48, and 51 are indicated at the beginning of their respective measures. The music continues with a melodic line in the treble clef and a supporting bass line in the bass clef.

Musical notation for measures 54-62. The system consists of a grand staff with a treble clef on the upper staff and a bass clef on the lower staff. Measure numbers 54, 57, and 60 are indicated at the beginning of their respective measures. The music continues with a melodic line in the treble clef and a supporting bass line in the bass clef.

Musical notation for measures 63-71. The system consists of a grand staff with a treble clef on the upper staff and a bass clef on the lower staff. Measure numbers 63, 66, and 69 are indicated at the beginning of their respective measures. The music continues with a melodic line in the treble clef and a supporting bass line in the bass clef.

Musical notation for measures 27-31. The system consists of a grand staff with a treble clef on the upper staff and a bass clef on the lower staff. The key signature has one flat (B-flat). Measure numbers 27, 28, 29, 30, and 31 are indicated above the staff. The music features a melodic line in the treble clef and a harmonic accompaniment in the bass clef.

Musical notation for measures 32-36. The system consists of a grand staff with a treble clef on the upper staff and a bass clef on the lower staff. The key signature has one flat (B-flat). Measure numbers 32, 33, 34, 35, and 36 are indicated above the staff. The music continues with a melodic line in the treble clef and a harmonic accompaniment in the bass clef.

Musical notation for measures 37-43. The system consists of a grand staff with a treble clef on the upper staff and a bass clef on the lower staff. The key signature has one flat (B-flat). Measure numbers 37, 38, 39, 40, 41, 42, and 43 are indicated above the staff. The music continues with a melodic line in the treble clef and a harmonic accompaniment in the bass clef.

Musical notation for measures 44-50. The system consists of a grand staff with a treble clef on the upper staff and a bass clef on the lower staff. The key signature has one flat (B-flat). Measure numbers 44, 45, 46, 47, 48, 49, and 50 are indicated above the staff. The music continues with a melodic line in the treble clef and a harmonic accompaniment in the bass clef.

Musical notation for measures 51-56. The system consists of a grand staff with a treble clef on the upper staff and a bass clef on the lower staff. The key signature has one flat (B-flat). Measure numbers 51, 52, 53, 54, 55, and 56 are indicated above the staff. The music continues with a melodic line in the treble clef and a harmonic accompaniment in the bass clef.

6

Musical notation for measures 27-28. The system consists of a treble clef staff and a bass clef staff. Measure 27 begins with a treble clef staff containing a whole note chord of G4, B4, and D5, and a bass clef staff containing a whole note chord of G2, B1, and D2. Measure 28 continues with a treble clef staff containing a whole note chord of A4, C5, and E5, and a bass clef staff containing a whole note chord of A2, C3, and E3.

Musical notation for measures 29-30. The system consists of a treble clef staff and a bass clef staff. Measure 29 begins with a treble clef staff containing a whole note chord of B4, D5, and F5, and a bass clef staff containing a whole note chord of B2, D3, and F3. Measure 30 continues with a treble clef staff containing a whole note chord of C5, E5, and G5, and a bass clef staff containing a whole note chord of C3, E3, and G3.

Musical notation for measures 31-32. The system consists of a treble clef staff and a bass clef staff. Measure 31 begins with a treble clef staff containing a whole note chord of D5, F5, and A5, and a bass clef staff containing a whole note chord of D2, F2, and A2. Measure 32 continues with a treble clef staff containing a whole note chord of E5, G5, and B5, and a bass clef staff containing a whole note chord of E3, G3, and B3.

Musical notation for measures 33-34. The system consists of a treble clef staff and a bass clef staff. Measure 33 begins with a treble clef staff containing a whole note chord of F5, A5, and C6, and a bass clef staff containing a whole note chord of F2, A2, and C3. Measure 34 continues with a treble clef staff containing a whole note chord of G5, B5, and D6, and a bass clef staff containing a whole note chord of G2, B2, and D3.

Musical notation for measures 35-36. The system consists of a treble clef staff and a bass clef staff. Measure 35 begins with a treble clef staff containing a whole note chord of A5, C6, and E6, and a bass clef staff containing a whole note chord of A2, C3, and E3. Measure 36 continues with a treble clef staff containing a whole note chord of B5, D6, and F6, and a bass clef staff containing a whole note chord of B2, D3, and F3.



Musical score system 1, measures 43-44. The system consists of two staves (treble and bass clef). Measure 43 shows a complex chordal texture with many accidentals. Measure 44 features a prominent trill in the right hand, with a '7' above it, and a '7' below the bass staff.



Musical score system 2, measures 45-46. Measure 45 contains a trill in the right hand, with a '38' above it. Measure 46 shows a continuation of the texture with various accidentals and a '14' below the bass staff.



Musical score system 3, measures 47-48. Measure 47 features a trill in the right hand, with a '16' above it. Measure 48 shows a continuation of the texture with various accidentals and a '28' below the bass staff.



Musical score system 4, measures 49-50. Measure 49 shows a continuation of the texture with various accidentals and a '16' below the bass staff. Measure 50 features a trill in the right hand, with a '36' above it.



Musical score system 5, measures 51-52. Measure 51 shows a continuation of the texture with various accidentals and a '28' below the bass staff. Measure 52 features a trill in the right hand, with a '28' above it.

Musical score system 16-17. The system consists of two staves, treble and bass clef. It contains two measures of music. Measure 16 starts with a treble clef, a key signature of one flat, and a common time signature. The bass line begins with a low octave. Measure 17 continues the melodic and harmonic progression.

Musical score system 20-21. The system consists of two staves, treble and bass clef. It contains two measures of music. Measure 20 features a more complex texture with multiple voices in the treble clef. Measure 21 continues the piece.

Musical score system 22-23. The system consists of two staves, treble and bass clef. It contains two measures of music. Measure 22 shows a continuation of the melodic lines. Measure 23 concludes the system.

Musical score system 24-25. The system consists of two staves, treble and bass clef. It contains two measures of music. Measure 24 has a simpler texture with fewer notes. Measure 25 continues the piece.

Musical score system 28-29. The system consists of two staves, treble and bass clef. It contains two measures of music. Measure 28 features a melodic line in the treble clef. Measure 29 concludes the system.

System 1: Measures 10-19. The system consists of two staves, Treble and Bass. Measure numbers 10, 14, and 19 are indicated above the staff. The music features a complex harmonic structure with many accidentals and a dense texture.

System 2: Measures 20-29. The system consists of two staves, Treble and Bass. Measure numbers 20 and 29 are indicated above the staff. The music continues with complex harmonies and a dense texture.

System 3: Measures 30-39. The system consists of two staves, Treble and Bass. Measure numbers 30 and 39 are indicated above the staff. The music continues with complex harmonies and a dense texture.

System 4: Measures 40-49. The system consists of two staves, Treble and Bass. Measure numbers 40 and 49 are indicated above the staff. The music continues with complex harmonies and a dense texture.

System 5: Measures 50-59. The system consists of two staves, Treble and Bass. Measure numbers 50 and 59 are indicated above the staff. The music continues with complex harmonies and a dense texture.

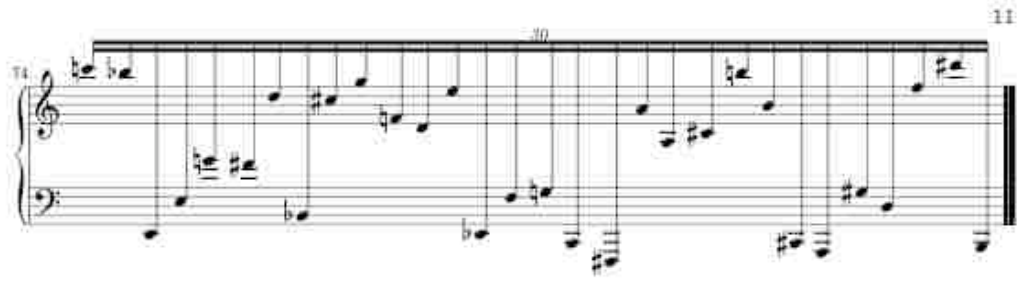
System 1: Measures 22-24. Treble clef, key signature of one sharp (F#). Measure 22 contains a complex chordal texture with sixteenth-note patterns. Measure 23 continues this texture. Measure 24 shows a transition with a different chordal structure.

System 2: Measures 25-29. Treble clef, key signature of one sharp (F#). Measure 25 features a melodic line in the treble and a bass line with chords. Measure 26 continues the melodic and harmonic development. Measure 27 shows a change in the bass line. Measure 28 and 29 conclude the system with sustained chords.

System 3: Measures 30-35. Treble clef, key signature of one sharp (F#). Measure 30 has a melodic line with eighth notes. Measure 31 continues the melody. Measure 32 shows a change in the bass line. Measure 33 and 34 continue the melodic and harmonic progression. Measure 35 ends with a chord.

System 4: Measures 36-40. Treble clef, key signature of one sharp (F#). Measure 36 features a melodic line with eighth notes. Measure 37 continues the melody. Measure 38 shows a change in the bass line. Measure 39 and 40 conclude the system with sustained chords.

System 5: Measures 41-46. Treble clef, key signature of one sharp (F#). Measure 41 has a melodic line with eighth notes. Measure 42 continues the melody. Measure 43 shows a change in the bass line. Measure 44 and 45 continue the melodic and harmonic progression. Measure 46 ends with a chord.



**Şekil 3.16** Seri halde kullanılan farklı karakterdeki otomatların nota çıktısı  
(Bkz. Ek-CD, Parça No. 3)

## 4. Bölüm: Rastlantısal Boole Ağları

Daha önceki bölümde, sistemlerin/maddenin işlevselliğinin incelenmesinde/anlaşılmasında, parçaların birbirleriyle kurdukları ilişkilerin ön plana çıkarılmasını incelemiştik. Hücresel otomatlar bağlamında ele aldığımız bu perspektifte, bünyeyi oluşturan elemanlar/hücreler arasındaki ilişkiler, komşuluk ilişkilerine dayanıyordu. Yani, makro düzeyde kendiliğinden oluşan düzen, lokal düzeyde hücrelerin nasıl etkileşeceğini tayin eden fonksiyonlardan alıyordu kaynağını. Bu bölümde ele alacağımız *network* kuramlarında ise elemanlar arası ilişkiler lokalizasyonlarından bağımsız olarak kurulmaktadır. Müzikal dokuların ve zamanın kuruluşunun lokal ilişkilerden daha soyut bir düzeyde ele alınmasının daha verimli sonuçlar doğuracağını düşünmekteyiz.

*Network* araştırmalarında, ilk çalışma Harvard Üniversitesi psikologlarından Stanley Milgram tarafından başlatılmıştır. Bu çalışmada Milgram'ın cevaplamaya çalıştığı soru, Amerika Birleşik Devletleri'nde birbirini tanımayan iki insan arasında kurulabilecek ilişkinin ortalama olarak kaç bağlantıyla kurulabileceğidir. Bunu sınavabilmek için kullandığı teknik, birbirini tanımayan bir insandan diğerine mektup ulaştırtmak, ve aradaki bağlantıların da bu mektubun ulaştırılmasında kayıtlanması şeklinde olmuştur (Mitchell 2009, 227).

Klasik matematikte, *network* araştırmaları ayrı bir alan olarak ele alınmadan önce graf kuramı altında incelenmekte idi. Markov bölümünde ayrıntılandığımız dinamik markov uygulamasında graf kuruluşlarını kullanmıştık (Bkz Bölüm 2.4).

*Network* kuramının ayrı bir alan olarak belirmesi, 1990'ların sonunda Duncan Watts ve Steven Strogatz tarafından yayınlanan “*Collective Dynamics of 'Small World Networks'*” ve Albert-Laszlo Barabasi ve Reka Albert tarafından yayınlanan “*Emergence of Scaling in Random Networks*” adlı makalelerle başlamıştır. Bu

makaleler *Nature* ve *Science* gibi üst düzey bilimsel mecmualarda yayınlanmış ve kısa zamanda bir çok araştırmacının ilgisine mazhar olmuştur. Bu kuramın gelişmesi, tarihsel olarak internetin, sınırsız veriye ulaşabilme imkanı sağlamış olmasına denk düşmektedir. Öte yandan fizikçiler arasında, soyut fizik çalışmalarına karşı bir doygunluk geliştirilmiş olması, gerçek dünyayı anlamının bu yeni yöntemine karşı ilgiyi yükseltmiştir.

Gen çalışmaları sonucu elde edilen bulgular, ağ kuramının, sistemlerin ve doğadaki kompleksliğin açıklanmasında aldığı merkezi konumu onaylar niteliktedir. İnsan ve hardal bitkisi (*arabidopsis*) genomu eşit sayıda gen taşımaktadır (25000). Bu durum kanıtlamaktadır ki, kompleksliği ortaya çıkartan, genlerin niceliği değil, genler arası kurulan ilişkilerin niteliğidir, başka bir deyişle genlerin *network* olarak nasıl örgütlendikleridir. Genlerin çoğunun işlevi diğerlerini düzenlemektir, yani düzenleyicisi olduğu genin aktif olup olmayacağını kontrol etmektir; bunun dışında kendilerinin sentezledikleri bir protein olmadığı düşünülmektedir.

Genomda belirleyici genlerin (*master genes*) bünyeye temel morfolojik yapısını kazandırdığı düşünülmektedir. Canlıların çoğu büyük oranda ortak genleri paylaşmaktadırlar, ancak belirleyici genler ve genler arası düzenleyici ağlardaki farklar biyolojik çeşitliliğin temeli olarak düşünülmektedir.

Rastlantısal Boole ağları (*Random Boolean Networks* [bundan sonra RBA denecektir]), Stuart Kauffman tarafından geliştirilmiş, gen içi düzenleyici ağları modelleyen bir sistemdir. Rastlantısal hücresel ağlar, hücresel otomatların genişletilmiş halidir. Diğer ağlarda olduğu gibi RBA'da da hücreler ve bunların diğerleriyle kurduğu bağlantılar vardır. Hücresel otomatlarla olan benzerliği de, hücrelerin değerlerini adımlı (*discrete*) zamanda aynı anda güncellemesidir. Her zamansal adımda hücrelerin alabileceği iki değer vardır; bu değerler 1 – 0, açık – kapalı ya da Boole terimleriyle doğru – yanlış olarak ifade edilebilir.

George Boole, 19. yüzyılda matematiksel mantık üzerine ufuk açıcı çalışmalarıyla tanınır. Mantıksal değerler (doğru – yanlış) ve operatörler (ve, ve ya, değil ..vb) Boole mantığının yapı taşlarıdır. RBA sisteminde hücre değerlerinin güncellemesi, Boole değer ve operatörleriyle sağlanmaktadır. Her bir hücrenin değerinin güncellemesinde, o hücreye atanan operatör ve ilişkide bulunduğu diğer hücrelerin değerleri hesaba katılır. Her hücrenin ilişkide bulunduğu hücreler ve bu hücrelere uygulayacakları mantıksal operatör rastlantısal olarak atanır.

18. yüzyılda Daniel Bernoulli tarafından geliştirilen gazların kinetik kuramı, kütleli gaz ve akışkan hareketlerini , moleküllerin belli yönlerdeki hareketleriyle açıklıyordu. Kütleli davranışın, yüksek nicelikteki elemanların hareketlerine dayandırılarak açıklanması için olasılık kuramı kullanılıyordu. RBA, istatistiksel mekanik kuramının biyolojideki yansıması olarak düşünülebilir. Bilgisayar uygulamaları için bir diğer avantajı da, dinamik sistemin adımlı (*discrete*) zamanda yaratılmasıdır. Gaz ve akışkanların kütleli davranışlarında hesaba katılan, moleküllerin hangi yönde hareket ettikleri idi; RBA'da ise kütleli davranışı ortaya çıkartan her bir hücrenin aktif olup olmamasıdır. Her bir hücrenin aktif olup olmaması, diğerleriyle kurduğu mantıksal ilişkiler sonucu belirlenmektedir. Kauffman'a göre RBA'nın, kompleks ve ama düzenli sistemlerin kuruluşunda taşıdığı merkezi önem 7 maddede özetlenmektedir (Kauffman 1993, 182-183):

1. Binlerce ya da milyonlarca değişkenin ilişkilendirildiği, dinamik bir sistem ortaya çıkmaktadır. Bu değişkenler aktif ya da inaktif olabilmektedir ve, gen içi düzenleyici sibernetik ağla, bağımsızlık sistemindeki birbiriyle bağlantılı hücreli ve moleküler yapılarla, *idiotype* ağla (Jerne 1984, Kauffman 1988), polimer sistemlerin içsel ilişkileriyle (Farmer, Kauffman ve Packard 1986; Farmer, Packard ve Perelson 1986; Kauffman 1986b) ya da bir nöral ağdaki etkileşen nöronlarla (McCulloch ve Pitts 1943; Hopfield 1982a, 1982b;

Rummelhart ve McClelland 1986) ilişkilendirilebilirler. Aktif-inaktif devrelerin formalize edilmesi, son derece kompleks sistemleri araştırma imkanı sağlamaktadır. Bu sorunsallara genellikle, *continuous* ve lineer olmayan diferansiyel denklemlerle yaklaşılamamaktadır.

2. Bir çok sistem için, açık-kapalı Boole idealizasyonu ya tamamen tutarlıdır ya da sistemdeki öğelerin lineer olmayan davranışlarının idealizasyonu için en iyi yöntemdir.
3. Boole idealizasyonu göstermektedir ki, bütün olası ağların, açıkca tanımlanmış bir birlikteliği vardır. Buna binaen, açıkca tanımlanmış birlikteliklerin, ortalama olarak, yapısal ve davranışsal nitelikleri çıkarsanabilmektedir. Genel istatistiki mekaniğin, ısı ve basınç dağılımına yaklaşımında olduğu gibi, bu sistemlerin yapısal ve davranışsal olarak öngörülebilirliği, yeni istatistiksel mekanikte de (RBA) makro düzeyde gözlemlenebilmektedir.
4. Karmaşık ve kompleks Boole ağlarının 3 ana davranış biçimi gösterdiği gözlenmiştir: düzenli, kompleks ve kaotik. Oldukça kompleks bu sistemlerin çözümlenmesi, beklenmedik bir yalınlığı gözler önüne sermiştir; bu durumun, biyolojideki gelişim ve evrim konularında önemli açılımlar sağlamıştır.
5. Boole idealizasyonunun kullanımı bize şu soruyu sordurtmuştur ve cevaplamamızı sağlamıştır: Kompleks bir devre (*switching*) ağında bütünlüklü bir düzenin oluşmasının koşulu nedir?
6. Düzenli dinamikliği ve kendiliğinden düzeni sağlayan aynı özellikler, sistemlerin çevresel koşullara uyumlanmasında da etkindir.
7. Karmaşık Boole ağları üç genel davranış özelliği gösterdiği ve ağa karakterini veren parametrelerdeki görece olarak küçük değişikliklerle bir davranış biçiminden diğerine geçebildiği için, doğal seçim, ağları, her davranış biçimine göre ayarlayabilir. Bu sayede, kendiliğinden oluşan düzen, davranışsal komplekslik ve evrimleşme kontrol edilmiş olur.

## 4.1 Boole Ağlarının Dinamik Kuruluşu

Yukarda belirttiğimiz gibi RBA'da hücreler yalnız iki değer alabilmektedir. Bunlar doğru – yanlış (bundan sonra doğru için  $t$ , yanlış için de  $f$  kullanılacaktır) şeklindeki Boole değerleridir. Hücrelere ayrıca *ve* ( $\wedge$ ), *ve ya* ( $\vee$ ) şeklinde Boole operatörleri atanır. Hücrelerin bir sonraki zamansal adımda alacağı değer, ilişkide bulunduğu diğer hücrelerin değerlerine, sahip olduğu Boole operatörünün uygulanması ile belirlenir. Örneğin hücrenin şimdiki değeri  $t$  ve sahip olduğu mantıksal operatör *ve* ( $\wedge$ ) olsun. Eğer ilişkide bulunduğu hücrelerin şimdiki durumdaki değerleri  $t$  ve  $f$  ise, bu hücrenin bir sonraki adımda alacağı değer  $f$  dir [ $t \wedge f = f$ ]. İlişkide bulunduğu hücrelerin bu değerleri ile, ilgili hücrenin bir sonraki adımda aktif olabilmesi için (yani  $t$  olması için) sahip olduğu mantıksal operatörün *ve ya* ( $\vee$ ) olması gerekirdi.

Hücrelerin bağlantıda bulunduğu diğer hücrelerin (bunlara giriş diyelim) niceliğine  $K$  dersek, her bir hücre yalnız iki değer alabildiğinden,  $K$  adet girişin sahip olabileceği kombinasyon sayısı  $2^K$ 'dir. Bütün bu kombinasyonların değeri, belirli bir operatör tarafından belirlenmektedir. Hücreler yalnız iki farklı değer alabildiğinden toplam Boole fonksiyonu  $F$  için  $K$  adet giriş niceliğiyle

$$F = 2^{2^K} \text{dir.}$$

Olası Boole fonksiyonları  $K$ 'nın değeriyle birlikte büyük oranda artış gösterir. En yalın Boole ağı eşzamanlıdır, yani bütün elemanların değerleri aynı anda güncellenir. Aşağıda üç elemanlı bir sistemin elemanları ve olası Boole fonksiyonları verilmiştir. Her elemanın  $K$  değeri 2'dir, demek ki  $2^2 = 4$  ayrı ihtimal vardır her biri için.

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

"ve"

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

"ve ya"

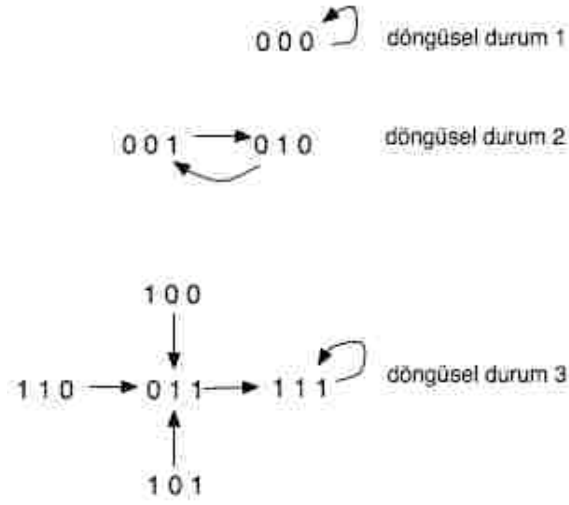
1	3	2
0	0	0
0	1	1
1	0	1
1	1	1

"ve ya"

**Tablo 4.1 3 elemanlı bir ağın alabileceği Boole kurallarına örnek**

Boole ağları sınırlı sayıda değer aldığı için sistem zorunlu olarak daha önce aldığı bir hale geri döner. Sistem deterministik olduğu için, aynı durumdan itibaren aynı rotayı izler; sistem biteviye bir döngüye girer (*state cycle*). Döngüsel durumlar, Boole ağının dinamik çekerleridir (*attractors*) (Ruelle 1998). Bir döngüsel duruma giden ya da onun üzerinde kalan durumların kümesine çeker havuzu (*basin of attraction*) denir. Bir durum döngüsünün uzunluğu en az 1 en çok da  $2^N$ 'dir. Ağların durum döngüleri ve çeker havuzlarının araştırılmasında *ddlab* uygulaması kullanılmaktadır.

Bu türden bir ağ en az bir durum döngüsüne sahiptir, birçoğu daha fazla döngü içerebilir. Yukardaki tablolarda gösterdiğim Boole ağı 3 durum döngüsüne sahiptir. Döngüsel durumlar aşağıdaki gibidir.



**Şekil 4.1 3 elemanlı örnek ağımızın döngüsel durumları (*state cycle*)**

Kauffman, Boole ağlarının aşağıda 6 maddede özetlenen özelliklerini vurgulamaktadır: (Kauffman 1993, 190-191)

1. Bir durum döngüsündeki zamansal adımların niceliği. Bir durum döngüsünün uzunluğu 1 ila  $2^N$  arasında değişmektedir.
2. Bir sistemde kaç değişik durum döngüsü olduğu. Alternatif durum döngülerinin niceliği en az 1'dir. Ancak en fazla  $2^N$  adet durum döngüsü ortaya çıkabilir. Bunlar bütüncül sistemin ilişkili olaylarıdır.
3. Çeker havuzunun büyüklüğü, durum döngüsü çekerlerine bağlıdır; yani ilki ikincisinin alt kümesidir.
4. Çekerler, sistemin işleyişindeki minimal değişimlere karşı statiktirler; minimal değişimden kasıt bir elemanın değerini değiştirmektir.
5. Geçici olarak bir elemanın değeri değiştirildiğinde, dinamik davranışta çeşitli değişiklikler vuku bulmaktadır.
6. Elemanlar arası kurulan bağlantılarda ve Boole kurallarında gerçekleşen

değişiklikler sonucu dinamik çekerlerin ve çeker havuzlarının değişmesi. Bu değişimler çevreye uyumlanmayı ıralamaktadır, ki Boole ağlarında gerçekleşen yapısal ve kurallara dair değişimler bu şekilde gerçekleşmektedir.

## 4.2 Rastlantısal Boole Ağlarının Türetilmesi

Yukarıda kuramsal olarak ele aldığımız Boole ağlarını müzikal uygulamalarda kullanabilmek için bunları türeten ve müzikal/zamansal olarak yorumlayan çeşitli yazılımlar geliştirdik. *Grace* platformunda çalışan *Scheme* kodları Boole uygulamalarının temelini oluşturmaktadır. Kendi yazdığım *Scheme* kodlarında, fonksiyon ve değişkenlere İngilizce isimler verdim. Programlama dilinin verili fonksiyonları İngilizcedir. Kendi tanımladığım fonksiyonlarla verili fonksiyonlar arasında dil tutarlığını sağlayabilmek için bu yolu tuttum. Boole ağlarının ses sentezlemede kullanımında, *Grace*'de ürettiğim datayı OSC (<http://opensoundcontrol.org/>) protokolüyle *Pure Data* (bundan sonra *Pd* denecektir) platformuna aktardım. Zira tekil ses üreten modülleri *Pd* platformunda kurmanın bazı pratik faydaları oldu. Tekil birimlerin kendi hafızalarına/tarihlerine sahip olmaları gerekiyordu. *Pd* deki *abstraction* özelliği buna imkan vermektedir.

Aşağıdaki *Scheme* kodu, argüman olarak girilen uzunlukta Boole değerleri listesi üretir. Bu liste sistemin başlangıç durumunu üretmek için kullanılır.

;Girilen sayı adedinde (amount) Boole değerleri listesi üretir;

```
(define (Gen amount)
  (loop repeat amount
    collect (odds .5)))
```

;10 elemanlı bir liste üretmek için aşağıdaki komut girilir;;

> (Gen 10)

```
(#t #t #t #f #t #f #t #f #f #f)
```

;;girilen eleman listesindeki (NetList) belirtilen eleman (CellPos) için maksimum **K** değeri (max) ya da daha az sayıda bağlantı ağları atar ;;

```
(define (random-networking NetList CellPos max)
  (loop repeat (between 1 (+ max 1))
    with node = (make-heap
      (remove (loop for i from 0 below (length NetList)
        collect i)
        CellPos))
    collect (next node)))
```

;Aşağıdaki komut, 10 elemanlı bir Boole değer listesinde, 4. sıra numarasına sahip eleman için ağ ilişkisi tayin eder.;

> (random-networking (Gen 10) 4 4)

```
(9 6 8 7)
```

Yukardaki fonksiyonda verili dilde tanımlanmamış 'remove' fonksiyonu, elemana atanan ağ oluşumunda, kendi ile ilişki kurmasına mani olmak için yazılmıştır. Yani liste içindeki sıra numarası diyelim ki 2 olan bir elemana atanan girişlerde 2 değeri hariç diğer değerler ihtimal dahilindedir. Yukardaki kodun çalıştırılabilmesi 'remove' fonksiyonunu aşağıda veriyorum.

;;belirtilen elemanı (element) girilen listeden (list) siler;;

```
(define (remove list element)
  (loop for i in list
    with lis = '()
    do
      (when (not (equal? i element))
        (push i lis))
    finally (return (reverse lis))))
```

Aşağıda verdiğim kod yukardaki fonksiyonları kullanarak, Boole fonksiyonu üretmektedir.

```
;;Girilen Boole değerleri listesi (NetList) için Boole fonksiyonları üretir; (max)  
argümanı 'random-networking' fonksiyonu için gerekli bi parametredir;;
```

```
(define (NetUpdateRules NetList max)  
  (loop for i from 0 below (length NetList)  
        with inputs = '()  
        with funs = '()  
        do  
          (set! inputs (push (sort (random-networking NetList i max) <) inputs))  
          (set! funs (push (list (pick '(and or))) funs))  
          finally (return  
                  (loop for p1 in funs  
                        for p2 in inputs  
                        collect (append p1 p2))))))
```

Aşağıdaki komutu girersek,

```
> (NetUpdateRules (Gen 10) 5)
```

```
((and 1 5 7 8) (or 0 2 3 4 9) (or 0 1 4 8) (or 0 2 3 4 7) (or 0 1 2 6 8) (or 0 3 5 6 9)  
(and 0 4 7) (and 0 4 5 6 8) (and 0 3 4 8 9) (or 4 5 9))
```

değerlerini alabiliriz. Çıkan listede, her bir eleman için Boole operatörü ve elemanın değerinin güncellemesinde ilişki kuracağı diğer elemanların sıra numaraları tayin edilmiştir. Dikkat edileceği üzere, ağ kurulumunda beher eleman için maksimum giriş niceliği 5 olarak girilmiştir. Çıkan listede girişlerin niceliği, bu yüzden en fazla 5 ve ya 5'ten azdır.

Aşağıdaki kod, Boole ağını çalıştıran bir mekanizmadır. Bu kombine sistem Boole değer ve fonksiyonlarını da tayin etmektedir.

```
;girilen Boole değerler listesini (NetList), belirlenen defada (rep) günceller. (max)  
argümanı maksimum giriş niceliğini belirler;
```

```

(define (NetUpdate NetList rep max)
  (loop repeat rep
    with rules = (NetUpdateRules NetList max)
    with cell = NetList
    with new-cell = '()
    with cell-list = (list NetList)
    initially (display rules)
    do
      (set! new-cell (loop for im in rules
        for im-count from 0
        collect
        (loop for il in (rest im)
          for ic from 0 below (length (rest im))
          with in = (nth NetList im-count)
          do
            (set! in ((first im) (nth NetList il) in))
            finally (return in))))
        (push new-cell cell-list)
        finally (return (reverse cell-list))))

```

Aşağıdaki komutu çalıştırdığımızda

```
> (NetUpdate (Gen 10) 5 4)
```

aşğıdaki RBA datasını alırız. İlk satırda Boole fonksiyonları verilmiş, diğer satırlar da sistemin her adımında güncellenmiş elemanların Boole değerlerini göstermektedir.

```

((and 2) (or 4) (or 0 1 2 4) (and 5 7) (or 0 1 4 7) (and 2 6) (and 6) (and 0 1 6) (and 5
9) (or 4 5 6))
(#f #t #f #f #t #t #f #f #f #t)
(#f #t #t #f #t #f #f #f #f #t)
(#f #t #t #f #t #f #f #f #f #t)
(#f #t #t #f #t #f #f #f #f #t)
(#f #t #t #f #t #f #f #f #f #t)
(#f #t #t #f #t #f #f #f #f #t)

```

#### 4.2.1 Sistemin Evrimsel Niteliği

Farkedilebileceği üzere, sistem 2. adımdan itibaren döngüsel duruma (*state cycle*) girmiştir, yani hep aynı değeri üretmektedir. Evrilen bir dinamik yapı kurabilmek için, elemanların Boole değerlerinde ya da Boole fonksiyonlarında tedrici değişimler yapmak gerekmektedir. Bunun için geliştirdiğimiz *Scheme* kodunda, sistem sabit bir değeri üretmeye başladığında, Boole kural ve değerleri tedrici olarak değiştirilmektedir. Bu sayede sisteme evrimsel bir potansiyel kazandırılmış olur. Katmanlı bir yapıya sahip aşağıdaki kod, sistemin kural ve değerlerini evrime uğratmaktadır.

```
(define (perturbator Rules&inputs NetList)
  (let* ((samp (between 0 (length Rules&inputs)))
        (inl (nth Rules&inputs samp))
        (change (random (length inl))))
    (if (= change 0) ;;değiştirilecek öğenin Boole fonksiyonu mu, giriş elemanı mı
        olduğunu sınar;
        (if (equal? (nth inl change) and) ;Boole fonksiyonunu değiştirir;
            (set! (nth (nth Rules&inputs samp) change) or)
            (set! (nth (nth Rules&inputs samp) change) and))
        (if (odds .5) ;;elemanın giriş kümesinde değişiklik yapar;
            (set! (nth (nth Rules&inputs samp) change) (first (random-networking-obstacle
NetList samp (concat (rest inl) (nth inl change)) 1))) ;'change' ile sıra numarası
alan giriş kümesindeki elemanı değiştirir;
            (if (= 2 (length inl))
                (set! (nth Rules&inputs samp)
                    (concat (first (nth Rules&inputs samp)) (rest inl) (random-networking-
obstacle NetList samp (rest inl) (between 1 4)))));giriş kümesine yeni bir eleman
ekler;
            (if (odds .5)
                (set! (nth Rules&inputs samp)
                    (concat (first (nth Rules&inputs samp)) (remove-many inl (list (first inl)
(nth inl change)))));giriş kümesinden bir elemanı siler;
                (set! (nth Rules&inputs samp)
                    (concat (first (nth Rules&inputs samp)) (rest inl) (random-networking-
obstacle NetList samp (rest inl) (between 1 3))))));giriş kümesine yeni elemanlar
ekler;
            Rules&inputs))
```

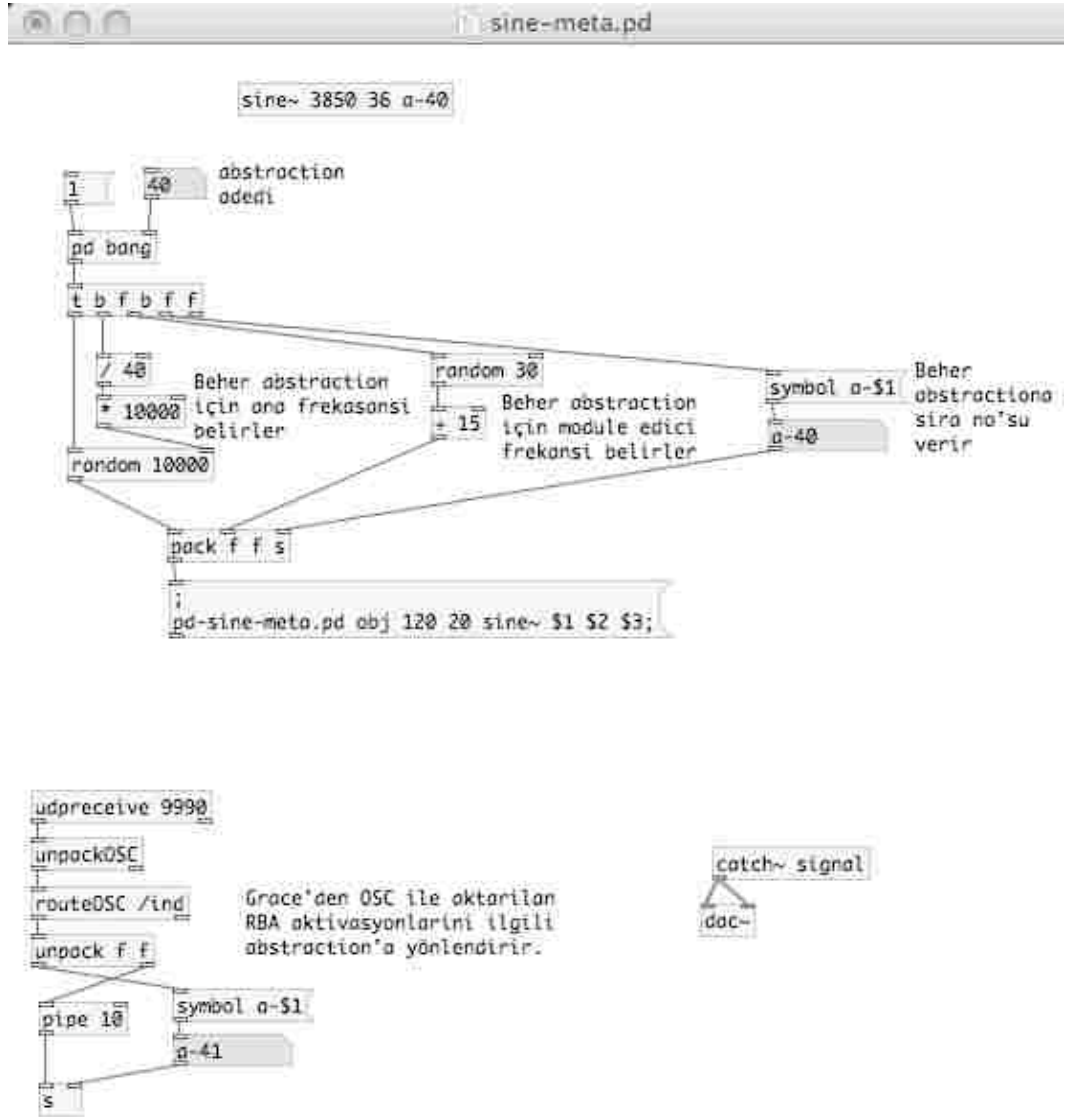
Sistemin sürerliğini garanti altına almak (sisteme evrimsel yetenek kazandırmak) için 'NetUpdate' fonksiyonuna, sistemin birbirini takip eden iki adımında aynı değeri ürettiğini tespit eden ek bir özellik kazandırdım, bu özellik 'perturbator' fonksiyonunu tetiklemektedir. Bu iki özellik 'NetUpdate' fonksiyonuyla bütünleştirildi, bu sayede sistem tıkanıldığında 'perturbator' devreye girerek Boole değer ve fonksiyonlarında küçük değişiklikler yapmaktadır. Aşağıda ilgili kombine kodu veriyoruz.

```
(define (NetUpdate-var NetList rep max)
  (loop repeat rep
    with rules = (NetUpdateRules NetList max)
    with cell = NetList
    with new-cell = '()
    with cell-list = '()
    do
      (set! new-cell (loop for im in rules
        for im-count from 0
        collect
        (loop for il in (rest im)
          for ic from 0 below (length (rest im))
          with in = (nth cell im-count)
          do
            (set! in ((first im) (nth cell il) in))
            finally (return in))))))
    (if (equal? cell new-cell)
      (set! rules (perturbator rules new-cell))
      (begin
        (set! cell new-cell)
        (push new-cell cell-list)))
    finally (return (reverse cell-list))))
```

### 4.3 RBA'nın Müzikal Uygulamaları

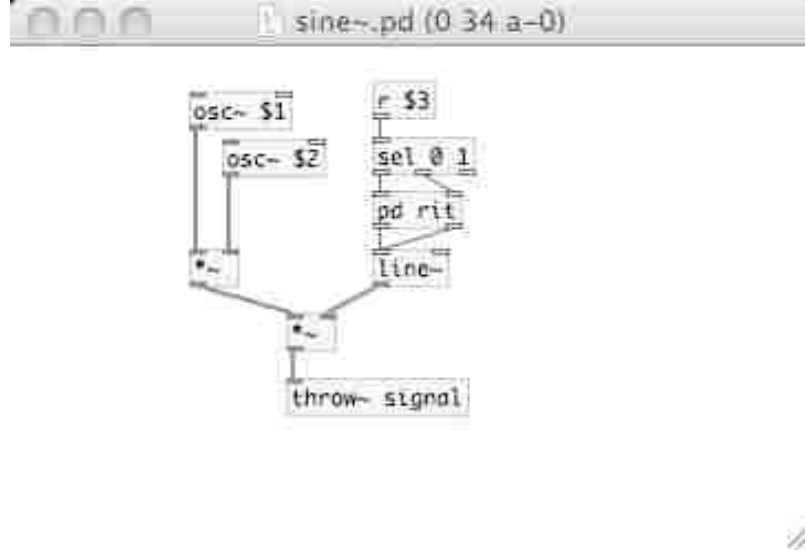
Buraya kadar ayrıntılandığımız sistemlerde RBA datasının üretilmesi ve evrimsel nitelikleri üzerinde durduk. Evrimleşen Boole değerleri, *Pd*'de kurduğumuz ses sentezleme mekanizmalarına *OSC* protokolünü kullanılarak aktarılır. Boole ağındaki her eleman *Pd*'de bir *abstraction*'a atanmıştır. İlgili hücre aktifleştğinde, *Pd*'de denk düştüğü *abstraction* etkinleşmektedir.

Aşağıda verdiğimiz *Pd* patchinde, girilen değer adedinde *abstraction* üretilmektedir. Her bir *abstraction*, RBA'nın bir elemanına atanmıştır. *OSC* ile *Grace*'den gönderilen hücre aktivasyon sinyalleri ilgilie *abstraction*'ı tetikler ya da susturur.



**Şekil 4.2** *abstraction*'ları üreten ve *Grace*'den gelen OSC sinyallerini bu *abstraction*'lara yönlendiren *Pd patchi*

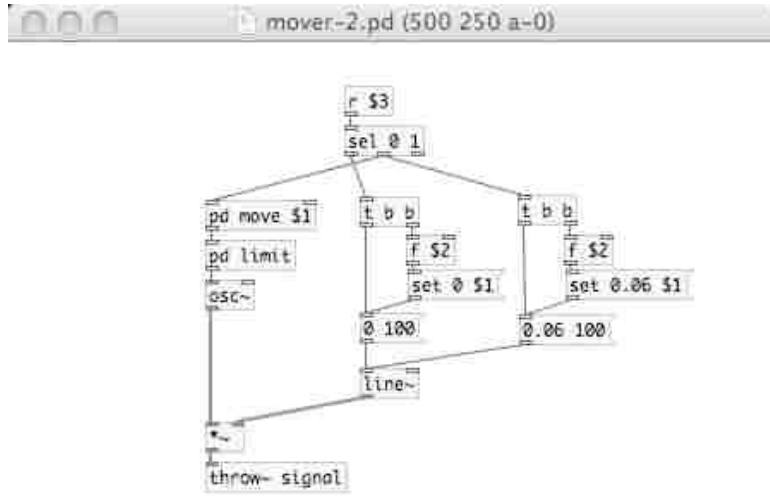
Bu patchde ses üreten 'sine~' adlı *abstraction* modüleli frekans (*Frequency Modulation* [bundan sonra FM denecektir]) tekniğiyle çalıştırılmaktadır. Beher *abstraction*'a iki frekans değeri atanır. Bunlardan ilki ana (*carrier*) frekans, ikincisi ise modüle eden (*modulator*) frekans değeridir (**Bkz. Ek-CD, Parça No.4**).



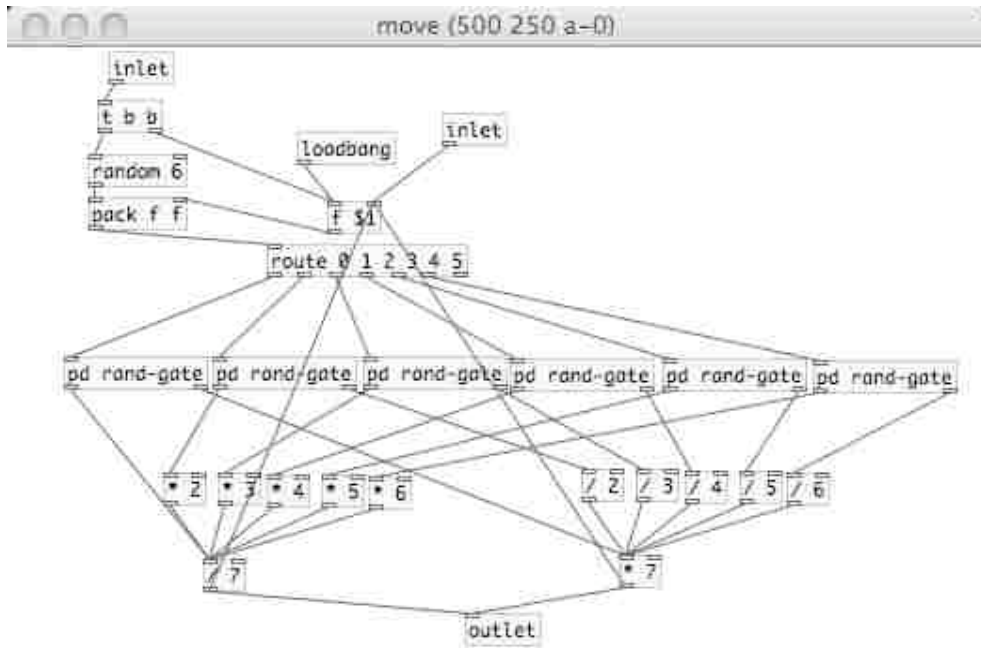
**Şekil 4.3 İki osilatörün birbiri ile modüle edildiği bir *Pd abstraction*'ı**

Yukarıdaki sistemin çalışması birebir ve statik haritalama olarak tanımlanabilir. Ancak FM tekniği ses spektrumunda standart bir komplekslik sağlamaktadır.

Aşağıda verdiğim patchde ise, statik haritalamayı her bir *abstraction*'ı birer fonksiyon olarak tanımlayarak aşmaya çalıştım. Bütün *abstraction*lara aynı frekans değeri atanmıştır. Ancak her bir *abstraction* tetiklendiğinde, kendi değerine, tanımlanmış fonksiyonu uygulayarak, değerini güncellemektedir. Yani beher *abstraction* kendi sürecini oluşturmaktadır. Aşağıdaki örnekte, modüllerde tanımlanan fonksiyon, kendi frekans değerinden hareketle, 7 eşit parçaya bölünmüş oktav alanı içinde yukarı ya da aşağı başka bir frekans değerine gitmek şeklindedir. **(Bkz. Ek-CD, Parça No. 5)**

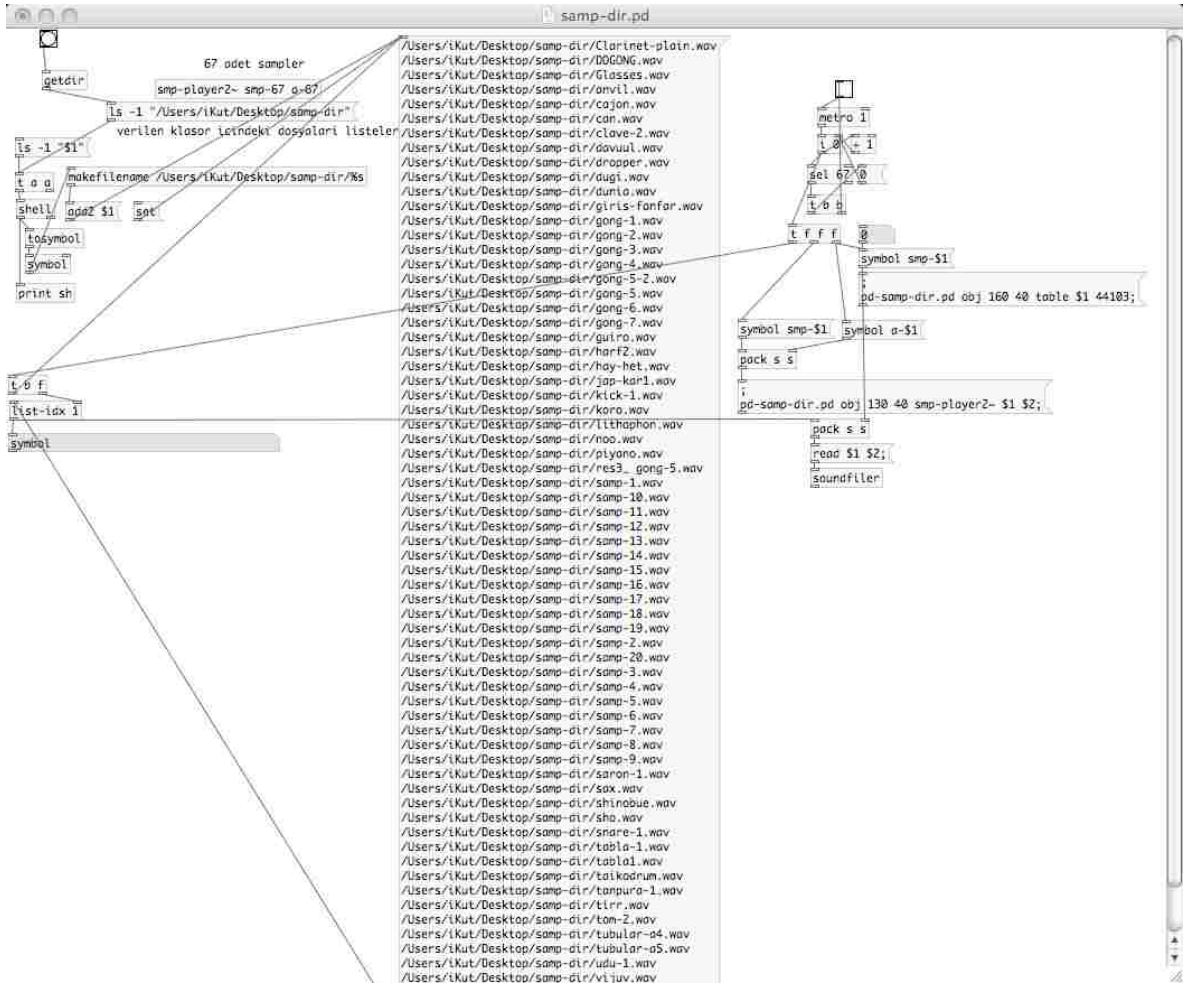


Şekil 4.4 Her tetiklenişinde kendi değerini güncelleyen *Pd patch*'i



Şekil 4.5 Frekansı her adımda bulunduğu noktadan 7 eşit parçaya bölünmüş oktav içinde hareket ettiren *patch*

Rastlantısal Boole ağları ile tetiklenen müzikal mekanizmalarda, evrimsel sürecin izlenebilmesi için ses dosyalarının kullanılması daha uygun bulunmaktadır. Bu maksatla geliştirdiğim bir *Pd* patchini aşağıda veriyorum. Bu patch, belirtilen klasör içindeki her bir ses dosyasını birer *sampler*'a atar ve *Grace*'den gelen OSC sinyalleri ile tetiklenir. RBA datasındaki hücre sayısı, klasördeki ses dosyası sayısına göre türetilir ve her biri bir *sampler*'ı kontrol eder.



**Şekil No. 4.6 Çoklu *sampler*; bir klasör içindeki bütün ses dosyalarını bir *sampler*'a atar. İlgili hücre aktifleştiginde, çalar (Bkz Ek-CD, Parça No.6).**

Buraya kadar vermiş olduğumuz *Pd* patchlerini, *Grace*'de yazdığımız kodları kullanarak kontrol edebilmek için aşağıdaki kodu yazdık. Bu kod RBA datası üretmektedir ve ürettiği datayı girilen zamansal sıklık uyarınca sıra numarası vererek *OSC* protokolüyle yollamaktadır.

```
(define (sender len-in len max rit)
  (let ((kume (NetUpdate-var (Gen len-in) len max)))
    (process for bet in kume
      do
        (if (equal? bet '())
          (sprout (process for pl from 0 below len-in
            do
              (osc:message `("/ind" ,pl 0))
              (wait .005)))
          (sprout (process for bet-in in bet
            for bet-indx from 0
            do
              (set! bet-in (if bet-in 1 0))
              (osc:message `("/ind" ,bet-indx ,bet-in))
              (wait .005))))))
    (wait rit)
    finally (sprout (process for pl from 0 below len-in
      do
        (osc:message `("/ind" ,pl 0))
        (wait .005))))))
> (sprout (sender 39 700 10 .3))
```

## 5. Bölüm: Sonuç

Doğadaki kendiliğinden örgütlenmeyi ve kompleks düzenin oluşmasını modelleyen sistemler olan hücresel otomatlar ve rastlantısal Boole ağlarını, müzik besteleme ve ses sentezleme bağlamında incelediğimiz bu çalışmamızda mekanik zamansal uygulamaları aşmaya yönelik teknikleri ele aldık. Algoritmik müzikte problem olarak koyduğumuz karakteristik, matematiksel idealizasyonun ürettiği datanın, birebir, müzikal parametrelere haritalanması idi. Bu çalışmada, Markov zincirleri, hücresel otomatlar ve rastlantısal Boole ağları kullanan müzikal algoritmeler geliştirildi. Bu algoritmeler, ilgili sistemlerin ürettiği datayı ayrı bir düzlemde müzikal olarak yorumlayan, mekanizmalardır. Yani, datanın doğrudan parametrik alana haritalanmasından önce, değerler, sistem içi bellek, datanın o zamansal adımdaki bütüncül niteliği ya da sürecin zamansal pozisyonu gibi çeşitli kriterler uyarınca, farklı biçimlerde yorumlanmaktadır. Bunun gibi zamansal stratejiler, algoritmik modelin ürettiği datanın niteliğinden bağımsız olarak da, lineer olmayan zamansal kuruluşlara imkan vermektedir. Sistem içi bellek ile kastedilen, fonksiyonların işlem uyguladıkları değeri her işlem sonucunda güncellemeleri ve yani bir sonraki tetiklenişlerinde bu değeri kullanmalarıdır. Datanın o zamansal adımdaki bütüncül niteliğiyle kast edilen, hücresel otomatlar bağlamında, aynı zamansal adımda kaç hücrenin birden aktifleştiği ve hücrelerin alacağı değerlerin, toplam hücre sayısına göre yeniden orantılandırılmasıdır. Sürecin zamansal pozisyon uyarınca farklı değerler üretmesi ise, *envelope* ya da *interpolation* kavramlarına göndermektedir bizi.

Parametrik alan yerine kendi değerlerine işlem uygulayan fonksiyonların konulması; diziselcilikte yeni bir anlayış olarak belirlemektedir. Yani, diziselleştirilen, değerler değil, değerlerin güncellenme biçimleridir. Kompozisyon sanatında bunun üretken uzanımları olacağı kanaatindeyim. RBA bağlamında kurduğumuz sistemde, sınırlı

fonksiyon kümesi, hücre sırası boyunca tekrarlanmaktadır. Örneğin 10 elemanlı bir Boole ağında 5 fonksiyon tanımlandı ise, 0'dan 4'e ve 5'den 9'a sıra numaralarına sahip hücelere aynı fonksiyon serisi atanmıştır. Sınırlı fonksiyon kümesinin, daha fazla elemanlı bir ağda verimli kullanılabilmesi için, bir mod içinde tekrarlanan her bir fonksiyon serisi ayrı bir çalgıya/partiye yönlendirilmektedir. Bu çalışmada gerçekleştiremediğimiz sistem, ağın eleman sayısı kadar fonksiyon tanımlayabilmek oldu. Bunun yerine, az elemanlı bir ağ kullanmaktan kaçındık; çünkü bu durumda sistemin evrimsel kabiliyeti kısır kalmaktadır.

Bu çalışmada geliştirilen bilgisayar kodları, MIDI ve OSC gibi standart protokollerde veri alışverişi sağlamaktadır. Bu sayede, diğer besteci/araştırmacılar bu kodları temel alıp, kendi uygulamalarında kullanabilirler. *Fomus* arabirimi kullanılarak, partiyon çıktısı XML formatında alınabilen müzik örnekleri genel kullanımdaki notasyon programlarında editlenmeye elverişlidir. Müzikal uygulamaların dışında, argümanları kullanıcı tarafından girilebilen bu kodlar, video üretiminde de kullanılabilir potansiyele sahiptir.

## Kaynakça

- Abelson, H., Sussman, G.J., Sussman, J., *Structure and Interpretation of Computer Programs*, MIT Press, 1996
- Adamatzky, A. *Identification of Cellular Automata*, Taylor & Francis. 1994
- Aizawa, Y., Nishikawa, I. & Kaneko, K. "Soliton Turbulence in One-Dimensional Cellular Automata", *Physica D* 45, s. 307-327, 1990
- Albert, J. & Culik, K. "A Simple and Universal Cellular Automaton and its One-Way and Totalistic Version", *Complex Systems* 1, s. 1-16, 1987
- Ames, C., "Automated Composition in Retrospect: 1956-1986", *Leonardo*, Vol. 20, No.2, Special Issue: Visual Art, Sound, Music and Technology, s. 169-185, The MIT Press, 1987
- Ariza, C., "An Open Design for Computer-Aided Algorithmic Music Composition: athenaCL." Ph.D. Dissertation, New York University, 2005.
- Ariza, C., "The Xenakis Sieve as Object: A New Model and a Complete Implementation." *Computer Music Journal* 29(2), s. 40-60, 2005.
- Ariza, C., "Beyond Transition Matrix: A Language-Independent, String-Based Input Notation for Incomplete, Multiple-Order, Static Markov Transition Values." 2006, yayınlanmamış manuskriptö [www.flexatone.net/docs/btmimosmtv.pdf](http://www.flexatone.net/docs/btmimosmtv.pdf) (indirme: 23/07/2009)
- Ariza, C., "Automata Bending: Applications of Dynamic Mutation and Dynamic Rules in Modular One-Dimensional Cellular Automata", *Computer Music Journal*, Vol. 31, No. 1, s. 29-49, 2007.
- Barbaud, P., *Initiation a la composition musicale automatique*, Dunod, Paris, 1966.
- Berg, P., *Using the ACToolbox*. [www.koncon.nl/ACToolbox](http://www.koncon.nl/ACToolbox), 2009.
- Beyls, P., "The Musical Universe of Cellular Automata", In T. Wells & D. Butler, Eds., *Proceedings of the 1989 International Computer Music Conference*. International Computer Music Association s. 34-41, 1989.
- Beyls, P., "Musical Morphologies from Self-organising Systems", *Interface*,

- Journal of New Music Research*, 19(2-3), s. 205-218, 1990.
- Beyls, P., "Self-Organising Control Structures using Multiple Cellular Automata", *Proceedings of the 1991 International Computer Music Conference*. Montreal, Canada: International Computer Music Association s. 254-257, 1991.
- Beyls, P., "Aesthetic Navigation", *Proceedings of the JIM Conference*, Lyon, France, 1997.
- Beyls, P., "Interactive Cellular Automata", *Evolution 2.0 CDRom*, Liverpool Art School and Merseyside On-Line Ltd, 1998.
- Beyls, P., "Synthetic Creatures in Context", *Intersens et Nouvelles Technologies*, MIM (Laboratoire Musique et Informatique de Marseille), 2000.
- Beyls, P., "Selectionist musical automata : Integrating explicit instruction and evolutionary algorithms", *IX Brazilian Symposium on Computer Music*. Brazilian Computing Society, 2003.
- Beyls, P., "Cellular Automata Mapping Procedures", *Proceedings of the 2004 International Computer Music Conference*, 2004.
- Bilotta, E., Pantano, P. & Talarico, V., "Music Generation through Cellular Automata: How to Give Life to Strange Creatures", *Generative Art GA2000*, Milano, Italia, 2000.
- Bilotta, E. & Pantano, P., "Artificial Life Music Tells of Complexity", *Proc. of Artificial Life Models for Musical Applications* (ECAL 2001 Workshop), 2001.
- Bilotta, E. & Pantano, P., "Synthetic Harmonies: Recent results", *Leonardo* 35(2): 35-42, MIT Press, 2002.
- Bilotta, E. Lafusa, A. & Pantano, P., "Searching for Complex CA Rules with GAs", *Complexity* 8(3), s. 56-67, 2003.
- Bilotta, E. Lafusa, A. & Pantano, P., "Life-Like Self-Reproducers", *Complexity* 8(3): 56-67, 2004.
- Bora, U., "Bilgisayar Destekli Kompozisyon Yöntemlerinde Yeni Yönelimler",

- Atatürk Üniversitesi Güzel Sanatlar Fakültesi SANAT Dergisi* , 4, s. 133-145, 2003.
- Bowcott, P., "Cellular Automata as a means of high level control of granular synthesis", In T. Wells & D. Butler. *Proceedings of the 1989 International Computer Music Conference*. San Francisco: ICMA s. 55-57.160, 1989.
- Brooks, S. & Ross, J., R., "Automated Composition from Computer Models of Biological Behavior", *Leonardo Music Journal*, Vol. 6, s. 27-31, 1996.
- Brown, A., "Exploring Rhythmic Automata", *Proceedings of the 3rd European Workshop on Evolutionary Music and Art*, 2005.
- Burks, A. (Ed) *Essays on Cellular Automata*. Univ. of Illinois Press, 1970.
- Burraston, D. Edmonds, E. Livingstone, D & Miranda, E., "Cellular Automata in MIDI based Computer Music", *Proceedings of the 2004 International Computer Music Conference*, 2004.
- Burraston, D. & Edmonds, E., "Global Dynamics Approach to Generative Music Experiments with One Dimensional Cellular Automata", *Proceedings of the 2004 Australasian Computer Music Conference*, 2004.
- Burraston, D., "One Dimensional Cellular Automata Musical Experiments with Max", *Proceedings of the 11th International Conference on Human-Computer Interaction*. HCI International, 2005.
- Burraston, D., "Composition at the Edge of Chaos ", *Proceedings of the 2005 Australasian Computer Music Conference*.
- Burraston, D.,Variety, "Pattern and Isomorphism", *Proceedings of the Third Iteration Conference*. Monash University, Australia, 2005.
- Burraston, D., "Structuring Cellular Automata Rule Space with Attractor Basins", *Proceedings of Generative Arts Practice Symposium 2005*.
- Burraston, D., "*Noyzelab*" *web sitesi*. [www.noyzelab.com](http://www.noyzelab.com).
- Burt, W., "Interactive Improvisations with Electronic Music Systems", *Sounds Australian*, Summer 1991-92.
- Burt, W., "Some parentheses around algorithmic composition", *Organised Sound*

- 1(3), s. 167-172, 1996.
- Busoni, F., *Sketch of a New Esthetic of Music*. In : Three Classics in the Aesthetics of Music, Dover, 1962.
- Cage, J., *Music of Changes*, Edition Peters, New York, 1960.
- Cage, J., *Silence*. Calder & Boyars, London, 1973.
- Candy, L. & Edmonds, E., *Explorations in Art and Technology*. Springer, 2002.
- Castagne, N. & Cadoz, C., "Ten Criteria for Evaluating Physical Modelling Schemes for Music Creation", *Proc. of the 6th Int. Conference on Digital Audio Effects*, 2003.
- Chadabe, J., *Electric Sound : The Past and Promise of Electronic Music*. Prentice Hall, 1997.
- Chareyron, J., "Sound synthesis and Processing by Means of Linear Cellular Automata", *Proceedings of the 1988 International Computer Music Conference*. ICMA, 1988.
- Chareyron, J., "Digital synthesis of self-modifying waveforms by means of linear automata", *Computer Music Journal*, 14(4), s. 25-41, 1990.
- Cohen, J. E., "Information Theory and Music", *Behavioral Science* 7(2), s. 137-163), 1962.
- Cope, D., *New Directions in Music*. Waveland Press (7th Edition), 2000.
- Coveney, P. & Highfield, R., *Frontiers of Complexity : The Search for Order in a Chaotic World*. Faber and Faber, 1995.
- Dewdney, A. K., "A cellular universe of debris, droplets, defects, and demons", *Scientific American*, August, s. 88-91, 1989.
- Dodge, C. ve Jerse, T. A.. *Computer Music: Synthesis, Composition, and Performance*. New York: Shirmer Book, 1997.
- Dorin, A., "Liquiprism : Generating Polyrhythms with Cellular Automata", *Proceedings of the 8th International Conference on Auditory Display*, Nakatsu & Kawahara (eds), Advanced Telecommunications Research International (ATR), Kyoto, Japan, s. 447-451, 2002.

- Dunn, J. *Algorithmic Arts*. www.geneticmusic.com
- Eco, U., *Trattato di semiotica*, Milano, Bompiani.
- Edmonds, E., "Logics for constructing generative art systems", *Digital Creativity* 14(1), s. 23-28, 2003.
- Edmonds, E. & Stubbs, M. Eds., *White Noise*. Australian Centre for the Moving Image, 2005.
- Edmonds, E., Brown, P. & Burraston, D. Eds., *Proceedings of Generative Arts Practice Symposium 2005*. Creativity & Cognition Studio Press.
- Felleisen, M., Fidler, R. B., Flatt, M., Krishnamurthi, S., *How to Design Programs: An Introduction to Programming and Computing*, MIT Press, 2001. (en son indirme: <http://www.htdp.org/>)
- Flake, G. W., *The Computational Beauty of Nature*. MIT Press, 1999.
- Gabor, D., "Acoustical Quanta and the theory of hearing", *Nature* (4044): 591-594, 1947.
- Gardner, M., "The Fantastic Combinations of John Conway's New Solitaire Game 'Life'", *Scientific American* 223(4), s. 120-123, 1970.
- Glaser, A., *History of binary and other nondecimal numeration*. Anton Glaser, Southampton, 1971.
- Gleick, J., *Kaos*, (çev. Üçcan, F.), Tübitak, 1995.
- Grace, <http://commonmusic.sourceforge.net/> (en son 13/08/2010'da bakıldı)
- Griffeath, D. & Moore. C. (ed.), *New Directions in Cellular Automata*. Oxford University Press, 2003.
- Harley, J., *Xenakis: His Life in Music*, Routledge, New York, 2004.
- Hiller, L. & Isaacson, L., *Experimental Music*, McGraw Hill, New York , 1959.
- Hiller, L., Baker, R., "Computer Cantata: A Study in Compositional Method", *Perspectives of New Music*, Vol. 3, No. 1, s. 62-90, 1964.
- Hiller, L., "Music composed with computers – a historical survey", In H. Lincoln, ed. *The Computer and Music*. Ithaca: Cornell University Press. s. 42-96, 1970.

- Hoffmann, P., *Amalgam aus Kunst und Wissenschaft*, PeterLang Verlag, Frankfurt am Main, 1994.
- Hoffmann, P., "Towards an "Automated Art": Algorithmic Processes in Xenakis' Compositions", *Contemporary Music Review* 21 No 2/3: s. 121-131, 2002
- Hofstadter, D. R., *Gödel, Escher, Bach: Bir Ebedi Gökçe Belik*, (Çev.:E. Akça, H. Koyukan), Kabalcı Yay., İstanbul, 2001.
- Holmes, T., *Electronic and experimental music*. Routledge, New York, 2008.
- Hunt, A., Kirk, R. & Orton, R., "Musical Applications of a Cellular Automata Workstation", *Proceedings of the 1991 International Computer Music Conference*. Montreal, Canada: International Computer Music Association. s. 165-168.
- Jaffe, D., "Ten Criteria for Evaluating Synthesis Techniques", *Computer Music Journal*. 19(1), s. 76-87, 1995.
- Jen, E., "Global Properties of Cellular Automata", *Journal of Statistical Physics*. (43), s.219-242, 1986.
- Katrami, A. I., Kirk, R. & Myatt, A., "Manipulation of cellular automata and fractal landscape mappings", *Proceedings of the 1991 International Computer Music Conference*. Montreal, Canada: International Computer Music Association, s. 106-109.
- Kauffman, S., *The Origins of Order, Self-Organisation and Selection in Evolution*. Oxford University Press, 1993.
- Kauffman, S., "The Global Dynamics of Cellular Automata: Review", *Complexity* 5(6), 2000.
- Koenig, G., M., *Performance notes and "Analytical Appendix" to the Übung für Klavier*, Institute of Sonology, Utrecht, 1969.
- Koenig, G., M., "Project 1", *Electronic Music Reports 1*, No. 2, s.32, 1970a.
- Koenig, G., M., "Project 2: A Programme for Musical Composition", *Electronic Music Reports 1*, No. 3 1970b
- Koenig, G., M., "Composition processes" Computer Music Reports on an

- International Project, UNESCO Rapport, 1978.
- Koenig, G., M., "Protocol", *Sonological Reports 4*, Institute of Sonology, Utrecht, 1979
- Kreger, T., "Cellular Automata in the Spectral Domain", *Proceedings of Interface 1997*. University of Auckland, Auckland.
- Kreger, T., "Real-time Cellular Automata Filters Implemented with Max MSP", *Proceedings of the Australasian Computer Music Conference 1999*. Victoria University of Wellington, N.Z.
- Krenek, E., "Extents and Limits of Serial Techniques", In Lang, P. H. Ed. *Problems of Modern Music. The Princeton Seminar in Advanced Musical Studies*. W. W. Norton and Company Inc. New York, 1962.
- Lafusa, A. & Bossomaier, T., "Localisation of critical transition phenomena in Cellular Automata rule-space", *Proceedings of the 2005 Australian Conference on Artificial Life*.
- Landy, L., *What's the Matter with Today's Experimental Music? Organized Sound Too Rarely Heard*. Harwood Academic Publishers, 1991.
- Langton, C. G., "Studying Artificial Life with Cellular Automata", *Physica 22D*, s. 120-149, 1986.
- Langton, C. G., "Life at the Edge of Chaos. In Artificial Life II", *Proceedings Vol. X. SFI Studies in the Sciences of Complexity*, Addison-Wesley, 1991.
- Ligeti, G., *Concerto for Piano and Orchestra*, Schott Music, 1988.
- Martin, A., "Reaction-Diffusion Systems for Algorithmic Composition", *Organised Sound* 1(3), s. 195-201, 1996.
- McCormack, J., "Art and the mirror of nature", *Digital Creativity* 14(1), s. 3-22, 2003.
- McIntosh, H. V., *Linear Cellular Automata*. <http://delta.cs.cinvestav.mx/~mcintosh>, 1990.
- McIntosh, H. V., *What Has and Hasn't Been Done With Cellular Automata*, <http://delta.cs.cinvestav.mx/~mcintosh/newweb/marcowhat.html>, 1990.

- McIntosh, H. V., "Wolfram's Class IV Automata and a Good Life", *Physica D* 25, s. 105-121, 1990.
- Meinhardt, H., *The Algorithmic Beauty of Sea Shells*. Springer, 2003.
- Messiaen, O., *Technique de mon langage musical : Textes avec exemples musicaux*, Alphonse Leduc, Paris, 1966.
- Meyer, L. B., *Emotion and Meaning in Music*. Univ. of Chicago Press, 1956.
- Millen, D., "Cellular Automata Music", *Proceedings of the 1990 International Computer Music Conference*. San Francisco: ICMA, s. 314-316, 1990.
- Millen, D., "Generation of formal patterns for music composition by means of cellular automata", In A. Strange Ed., *Proceedings of the 1992 International Computer Music Conference*. San Francisco: ICMA, s. 398-399, 1992.
- Millen, D., "An Interactive Cellular Automata Music Application in Cocoa", *Proceedings of the 2004 International Computer Music Conference*, 2004.
- Millen, D., [comp.uark.edu/~dmillen/cam.html](http://comp.uark.edu/~dmillen/cam.html), 2005.
- Miranda, E. R., "Granular Synthesis of Sounds by Means of a Cellular Automaton", *Leonardo*, s. 297-300, Vol. 28, No. 4, 1995.
- Miranda, E. R., *Composing Music with Computers*. Focal Press, 2001.
- Miranda, E. R., *Computer Sound Synthesis for the Electronic Musician*. 2nd Edition, Focal Press, 2002.
- Miranda, E. R., "On the evolution of music in a society of self-taught digital creatures", *Digital Creativity* 14(1), s. 29-42, 2003.
- Miranda, E. R., "Cellular Automata Music: From Sound Synthesis to Musical Forms", *Evolutionary Computer Music*, (ed.: Miranda, E. R. & Biles, J. A.), Springer, London, 2007a, s. 171-193
- Miranda, E. R. & Biles, J. A. (ed.), *Evolutionary Computer Music*, Springer, London, 2007b.
- Mitchell, M., *Complexity: A Guided Tour*, Oxford University Press, New York, 2009.
- Niemiec, M. D., "Synthesis of Complex Life Objects from Gliders", *New Directions*

- in Cellular Automata*, (ed.: Griffeath, D. & Moore. C.), 2003, s. 57.
- Nierhaus, G., *Algorithmic Composition*, Springer Verlag, Wien, 2009.
- Nyman, M., *Experimental Music: Cage and Beyond*. Cambridge University Press: New York, 1999.
- Oliveira, G., Oliveira, P. & Omar, N., "Definition and Application of a Five-Parameter Characterization of One-Dimensional Cellular Automata Rule Space", *Artificial Life* 7, s. 277-301, 2001.
- Oliveira, G., Oliveira, P. & Omar, N., "Guidelines for Dynamics-based Parameterization of One-Dimensional Cellular Automata Rule Spaces", *Complexity* 6(2), s. 63-71, 2001.
- Olsen, H. F. & Belar, H., "Aid to Music Composition Employing a Random Probability System", *J. Acoust. Soc. Am.* 33, s. 1163-1170, 1961.
- Orton, R., Hunt, A. & Kirk, R., Graphical Control of Granular Synthesis using a Cellular Automata and the Freehand Program. *Montreal ICMC 1991 Proceedings*, s. 416-418, 1991.
- Özer, Y., *Bilim Perspektifinde Müzik*, Dokuz Eylül Yayınları, 1997.
- Partch, H., *Genesis of a Music*, Da Capo Press, 1974.
- Pure Data*, <http://puredata.info/> (en son 13/08/2010'da bakıldı)
- Reich, S., *Piano Phase*, Universal Edition, 1967.
- Reich, S., *Pendulum Music*, Universal Edition, 1973.
- Reich, S., *Pendulum Music*, <http://www.ubu.com/aspen/aspen8/leadPendulum.html>
- Roads, C., "Automated Granular Synthesis of Sound", *Computer Music Journal*, 2(2) s. 61-62, 1978.
- Roads, C., *The Computer Music Tutorial*. MIT Press, 1996.
- Roads, C., *Microsound*, MIT Press, 2001.
- Rowe, R., *Machine Musicianship*, MIT Press, 2001.
- Ruelle, D., *Rastlantı ve Kaos*, (çev. Yurtören, D.), Tübitak, 1994.
- Sadie, S. (ed) *The New Grove dictionary of music and musicians*, 7. Macmillan Publishers, New York & London, 1995.

- Schillinger, J., *The mathematical basis of the arts*. Kluwer Academic/Plenum Publishers, New York, 1976.
- Schwartz, E., *Electronic Music : A Listeners Guide*. Da Capo Press, 1989.
- Silver, S. A., *Life Lexicon*. [www.argentum.freemove.co.uk/lex\\_home.htm](http://www.argentum.freemove.co.uk/lex_home.htm)
- Sipper, M., *Evolution of Parallel Cellular Machines : The Cellular Programming Approach*. Springer Verlag, 1997.
- Sipper, M., "Fifty years of research on self-replication: An overview", *Artificial Life* 4, s. 237-257, 1998.
- Solomos, M., "Cellular Automata in Xenakis' Music. Theory and Practice", In Georgaki, A. & Solomos, M. Eds. *Proceedings of the International Symposium Iannis Xenakis*. s. 120-137, 2005.
- Taube, H. K., *Notes from the Metalevel*, Taylor&Francis, 2007.
- Tenney, J., *Meta-Hodos*, Frog Peak Music, Oakland, 1988.
- Toffoli, T. & Margolus, N., "Invertible cellular automata: A review", *Physica D* 45, s. 229-253, 1990.
- Tofts, C., "Describing Social Insect Behavior Using Process Algebra," *Transactions of the Society for Computer Simulation* 9 , No. 4, s. 227-283, 1992.
- Truax, B., "Real-Time Granular Synthesis with a Digital Signal Processor", *Computer Music Journal*. 12(2), s. 14-26, 1988.
- Truax, B., "Composing with Real-Time Granular Sound", *Perspectives of New Music*, Vol. 28, No.2, 1990.
- Turing, A., "The Chemical Basis of Morphogenesis", *Philosophical Transactions of the Royal Society*, 1952.
- Vaidhyathan, S., Minai, A. & Helmuth, M., "ca : A System for Granular Processing of sound using Cellular Automata ", *Proceedings of the 2nd COST G-6 Workshop on Digital Audio (DAFx99)* 1999, NTNU, Trondheim, 1999.
- Varga, B. A., *Conversations with Iannis Xenakis*, London : Faber and Faber, 1996.
- Voorhees, B., *Computational Analysis of One-Dimensional Cellular Automata*, World Scientific, 1996.

- Walker, C. C. & Ashby, W. R., "On Temporal Characteristics of Behaviour in Certain Complex Systems", *Kybernetik* 3, s. 100-108, 1966.
- Walker, C. C., "Behaviour of a Class of Complex Systems: the Effect of System Size on Properties of Terminal Cycles", *Journal of Cybernetics* 1(4), s. 55-67, 1971.
- Whitelaw, M., *Metacreation: Art and Artificial Life*. MIT Press, 2004.
- Wilhelm, R., *I Ching ya da Değişimler Kitabı*, (çev. Özşar, L.), Biblos Yayınları, Bursa, 2002.
- Winkler, T., *Composing Interactive Music, Techniques and Ideas Using Max*, MIT Press, 2001.
- Wolfram, S., "Statistical Mechanics of Cellular Automata", *Reviews of Modern Physics*, 55(3), s. 601-644, 1983a.
- Wolfram, S., "Cellular Automata", *Los Alamos Science*, 9, s. 2-21, 1983b.
- Wolfram, S., "Universality and Complexity in Cellular Automata", *Physica D*. 10D, s. 1-35, 1984a.
- Wolfram, S., "Computer Software in Science and Mathematics", *Scientific American*. 251(3), s. 188-203, 1984b.
- Wolfram, S., "Twenty Problems in the Theory of Cellular Automata", *Physica Scripta*, T9, s. 170-183, 1985.
- Wolfram, S., "An Annotated Bibliography of Cellular Automata", In Wolfram, S. Ed. *Theory and Application of Cellular Automata*. World Scientific, 1986.
- Wolfram, S., "Tables of Cellular Automaton Properties", In Wolfram, S. Ed. *Theory and Application of Cellular Automata*. World Scientific, 1986.
- Wolfram, S., "Random Sequence Generation by Cellular Automata", *Advances in Applied Mathematics*, (7), s. 123-169, 1986.
- Wolfram, S., "Complex Systems Theory", In *Emerging Syntheses in Science: Proceedings of the Founding Workshops of the Santa Fe Institute*. Addison-Wesley. s.183-189, 1988.
- Wolfram, S., *A New Kind of Science*. Wolfram Media, 2002.

- Wolfram, S., *Mathematica*, www.wolfram.com.
- Woolf, S. & Thompson, A., "The Sound Gallery – An Interactive A-Life Artwork",  
In Bentley, P. J. & Corne, D. W. Eds. *Creative Evolutionary Systems*. Morgan  
Kaufmann, 2002.
- Wooller, R., Brown, A., Miranda, E., Berry, R. & Diederich, J., "A Framework for  
Comparison of Process in Algorithmic Music Systems", In Edmonds, E.,  
Brown, P. & Burraston, D. Eds. *Proceedings of Generative Arts Practice  
Symposium 2005*. Creativity & Cognition Studio Press. s. 109-124.
- Wuensche, A. & Langton, C., *Complexity in Small Universes*, www.ddlab.com,  
1999
- Wuensche, A. & Lesser, M., *The Global Dynamics of Cellular Automata : An Atlas  
of Basin of Attraction Fields of One-Dimensional Cellular Automata*.  
Addison-Wesley. (www.ddlab.com adresinde PDF formatı mevcuttur),  
1992.
- Wuensche, A., *Complexity in One-D Cellular Automata: Gliders, Basins of  
Attraction and the Z parameter*, Santa Fe Institute working paper 94-04-025,  
1994.
- Wuensche, A., *Attractor Basins of Discrete Networks*. Cognitive Science Research  
Paper 461, Univ. of Sussex, PhD tezi, 1997.
- Wuensche, A., *Classifying Cellular Automata Automatically*, Santa Fe Institute  
working paper 98-02-018, 1998.
- Wuensche, A., "Classifying Cellular Automata Automatically: Finding gliders,  
filtering, and relating space-time patterns, attractor basins, and the Z  
parameter", *Complexity* 4(3), s. 47-66, 1999.
- Wuensche, A., "Glider dynamics in 3-value hexagonal cellular automata: the  
beehive rule", *Journal of Unconventional Computing*, 2005.
- Wuensche, A., *Discrete Dynamics Lab*. Discrete Dynamics Inc. (www.ddlab.com  
adresinde mevcuttur), 2006.
- Xenakis, I., *Formalized Music*. (Revised Edition). Pendragon Press, 1992.

## Özgeçmiş

Aykut ÇAĞLAYAN  
İstanbul/Boğaziçi, 1980

### **Okullar:**

1991-1998 Cağaloğlu Anadolu Lisesi

1998-2001 İstanbul Üniversitesi, Mütercim-Tercümanlık (Almanca) Bölümü [ön lisans].

2001-2005 Yıldız Teknik Üniversitesi, Sanat ve Tasarım Fakültesi, Duysal Tasarım Programı, Kompozisyon dalı [lisans].

2005-2010 Ege Üniversitesi, Devlet Türk Müziği Konservatuvarı, Temel Bilimler (Duysal Tasarım), Kompozisyon ve Etnomüzikoloji dalları [doktora].

### **Yabancı Diller:**

Almanca ve İngilizce, iyi düzeyde.  
Fransızca ve Rusça, başlangıç düzeyinde.

### **İlgi ve Çalışma Alanları:**

Kompozisyon, bilgisayar müziği, müzik teorisi, etnomüzikoloji, piyano, bilgisayar programlama (*Pure Data*, *Lisp*), bilim/felsefe/sanat, holistik tıp.

### **Akademik Faaliyetler:**

2009-2010 döneminde, Fulbright bursuyla gittiğim, University of Illinois at Urbana-Champaign, School of Music, Composition/Theory bölümünde konuk araştırmacı olarak bulundum.

### **Sanatsal Faaliyetler:**

*Dört Halife* (2002), *Emily* (2002), *A bout de soufflé* (2004), *Simbanggok* (2006), *Carillon Tekbir* (2009), *Nuclear Music* (2009) adlı yapıtlarım yurtiçi ve yurtdışında gerçekleşen çeşitli etkinliklerde seslendirilmiştir.

## Özet

Batı müzik tarihi boyunca, algoritmik uygulamalar müzik bestelemenin doğal bir uzanımı olarak görülmektedir. Bilgisayar devrimi ile birlikte, algoritmik yöntemler, besteciler arasında yaygınlık ve derinlik kazanmıştır. Diğer yandan, bilimsel araştırmalarda da, bilgisayarların sağladığı işlemsel hız, dinamik sistemler ve kaotik yapıların araştırılmasında yeni ufuklar açmıştır. Sanatsal ve bilimsel yöntemleri birlikte ele almamın altında yatan gerekçeler; estetik ve epistemolojik alanların birbirleriyle dolaysız ilişki içinde olduğunu düşünmem ve bu tezde, sanatın *mimetik* yönünün öne çıkmış olmasıdır. Bu bağlamda, doğadaki kendiliğinden örgütlenme ve değişim (*self-organisation*) fenomeninin modellendiği yöntemler olan hücresel otomatlar (*Cellular Automata*) ve rastlantısal Boole ağları (*Random Boolean Network*) ile Markov zincirleri uygulaması bu tezde ayrıntıları ve literatürdeki müzikal uygulamaları ile ele alınmıştır. Hücresel otomatlar ve rastlantısal Boole ağları, sistemi oluşturan öğelerin kurallı iç ilişkilerine dayanır ve makro düzeyde kompleks yapılar üretebilen kabiliyetlere sahiplerdir. Bu sistemler deterministik hareket etmelerine rağmen öngörülemez sonuçlar doğurabilmektedirler. Çoğul öğelerin tutarlı iç ilişkiler ile evrimleşmesi algoritmik bestelemede verimli sonuçlar doğurmaktadır. Zira, müzikal estetiğe dair iki temel kavram bu sistemlere içkindir: tutarlılık (*coherence*) ve çeşitlenme (*variation/evolution*). Ancak sözkonusu modeller, zaman/mekansal olarak sınırlı boyutlarda temsil edilmektedir. Bu bakımdan, problem olarak konulan, matematiksel idealizasyonların müzikal zamana aktarılması ve tezin ilgili (2., 3. ve 4.) bölümlerinin son kısımlarında geliştirilen yeni yöntemlerin çıkış noktasını bu problem oluşturmaktadır. Sınırlı boyutlarda temsil edilen matematiksel modellerin tekdüze biçimde zamansal organizasyona aktarılmasının aşılması amacıyla, çizgisel olmayan (*non-linear*) stratejiler/yöntemler geliştirmeye çalıştım. Bu yöntemler, hücresel otomat nesillerinin esnek olarak yorumlanması, iki Markov tablosu arasında *interpolation* kurulması (*melodic morphing*), rastlantısal Boole ağlarında her bir hücrenin bellek sahibi ve kendine dönüşlü bir fonksiyona atanmasıdır. Rastlantısal Boole ağlarının müzik bestelemede kullanımına literatürde rastlanmamıştır. Öğeleri arasında kurulan lokal olmayan ilişki biçimleri, RBA'nın algoritmik bestelemede daha esinleyici olmasını sağlamaktadır. Benim yöntemimde öne çıkan hususiyet, sistemin öğelerinin fonksiyonlara atanmış olmasıdır. Bu yöntemler, çeşitli bilgisayar kodları olarak (*Lisp*, *Pure Data* ve *Mathematica*) metin içinde verilmiş olup, müzikal çıktılarının notaları metin içinde, ses örnekleri ise CD ekinde verilmiştir.

## Abstract

In the course of western classical music, algorithmic procedures have been integral to composition techniques. After the computer revolution, those procedures became prevalent and deepened among composers. Accordingly, the use of computer and its fast computational power opened new dimensions in the research of dynamical systems and modelling self-organisation in the nature. The reason for mentioning artistic and scientific methods side by side is that I think that there is a direct relationship between aesthetical and epistemological fields and the mimetic attribute of art came to prominence in this dissertation. In that sense, the mathematical models of the natural self-organisation; Cellular Automata and random Boolean networks and Markov chains are surveyed in this dissertation. And new techniques by using those systems to produce music and sound have been developed. Cellular Automata and random Boolean networks depends on the interactions of cells, which constitutes the system and are capable of emergent behaviour. Even these systems are deterministic, they are not predictable. The micro interactions produce complex structures in the macro level. The coherent and synchronous evolution of plural constituents are very fruitful in algorithmic composition. Yet, the basic two concepts regarding musical aesthetics are intrinsic to that systems: coherence and variation/evolution. Nevertheless, the mentioned systems are presented in limited time/space dimensions. Henceforth the problem of this dissertation is, transposing the mathematical idealizations into musical time in a natural way rather than mechanical time steps. The last parts of chapters include my original approach to this problem. My approaches are flexible interpretations of Cellular Automata generations, the interpolation between two Markov tables (melodic morphing), assigning each cells to a function, which has a memory and recursive. The use of random Boolean networks in algorithmic composition literature is not available. Non-local interactions between the elements of RBN is very inventive for algorithmic composition. The prominent quality in my approach is assigning each cell to a function. The codes and the resulting scores are given within the text and the sound examples are attached to text within a CD.