

**YAPAY SİNİR AĞLARININ MİMARİ SEÇİMİ İÇİN TABU  
ARAMA ALGORİTMASI**

**USING TABU SEARCH ALGORITHM IN THE SELECTION  
OF ARCHITECTURE FOR ARTIFICIAL NEURAL  
NETWORKS**

**ÇAĞDAŞ HAKAN ALADAĞ**

Hacettepe Üniversitesi

Lisansüstü Eğitim – Öğretim ve Sınav Yönetmeliğinin

İSTATİSTİK Anabilim Dalı İçin Öngördüğü

DOKTORA TEZİ

olarak hazırlanmıştır.

2009

Fen Bilimleri Enstitüsü Müdürlüğü'ne,

Bu çalışma jürimiz tarafından **İSTATİSTİK ANABİLİM DALI 'nda DOKTORA TEZİ** olarak kabul edilmiştir.

Başkan :.....  
Prof. Dr. Süleyman Günay

Üye (Danışman) :.....  
Prof. Dr. Gülsüm Hocaoğlu

Üye :.....  
Prof. Dr. Hasan Bal

Üye :.....  
Doç. Dr. Cem Kadılar

Üye :.....  
Doç. Dr. Özlem Müge Aydın

ONAY

Bu tez ...../...../..... tarihinde Enstitü Yönetim Kurulunca kabul edilmiştir.

Prof.Dr. Erdem Yazgan  
Fen Bilimleri Enstitüsü Müdürü

*Sevgili anneannem Fevziye Aladađ*

*ve dedem Abdurrezzak Aladađ için,*

# YAPAY SİNİR AĞLARININ MİMARİ SEÇİMİ İÇİN TABU ARAMA ALGORİTMASI

Çağdaş Hakan Aladağ

## ÖZ

Yapay sinir ağları yöntemiyle zaman serilerinde öngörü elde edilirken, en iyi yapay sinir ağı mimarisinin nasıl seçileceği önemli bir problemdir. Bu çalışmada, belirtilen problemi çözmek için tabu arama algoritmasına dayalı bir mimari seçim yöntemi geliştirilmiştir.

Birinci bölümde, öngörü işleminin önemi vurgulanmış ve yapay sinir ağları yöntemi ile öngörü yapılırken karşılaşılan problem ortaya konmuştur. Tabu arama algoritması ve temel kavramları ikinci bölümde verilmiştir. Üçüncü bölümde, yapay sinir ağları yöntemi ve bu yöntemin zaman serilerinin çözümlenmesinde nasıl kullanıldığı açıklanmıştır. Dördüncü bölümde, mimari seçimi probleminin çözümü için geliştirilen tabu arama algoritmasına dayalı yaklaşım verilmiştir. Beşinci bölümde, yapılan uygulama anlatılmış ve son bölümde elde edilen sonuçlar tartışılmıştır.

**Anahtar Kelimeler:** Mimari seçimi problemi, öngörü, tabu arama, yapay sinir ağları, zaman serileri.

Danışman: Prof. Dr. Gülsüm Hocaoğlu, Hacettepe Üniversitesi, İstatistik Bölümü, Yönelem Araştırması Anabilim Dalı.

# USING TABU SEARCH ALGORITHM IN THE SELECTION OF ARCHITECTURE FOR ARTIFICIAL NEURAL NETWORKS

Çağdaş Hakan Aladağ

## ABSTRACT

The determination of the best architecture is an important problem when time series are forecasted by artificial neural networks. In this study, an architecture selection method based on tabu search algorithm is proposed to solve this problem.

In Section 1, the importance of forecasting procedure is emphasized and the problem of the artificial neural networks method in this procedure is revealed. Tabu search algorithm and its basic concepts are given in Section 2. In Section 3, artificial neural networks method and its usage for analyzing time series are presented. The proposed approach based on tabu search algorithm is introduced in Section 4. Implementation is given in Section 5 and the obtained results are discussed in the last section.

**Keywords:** Architecture selection problem, artificial neural networks, forecasting, tabu search, time series.

Advisor: Prof.Dr. Gülsüm Hocaoğlu, Hacettepe Üniversitesi, Department of Statistics, Operation Research Section.

## TEŞEKKÜR

Danışmanım Prof. Dr. Gülsüm Hoccoğlu'na, önerileri ile tez çalışmamı yönlendirdiği için, sayın Prof. Dr. Hasan Bal, sayın Doç. Dr. Cem Kadılar ve sayın Doç. Dr. Özlem Müge Aydın'a, yaptıkları olumlu katkıları için,

Değerli hocam Prof. Dr. Süleyman Günay'a, karşılaştığım güçlüklerde bana yol göstererek ilerleyebilmemi sağladığı ve verdiği destek için,

Aralarında olmaktan büyük gurur ve mutluluk duyduğum tüm değerli bölüm hocalarıma ve arkadaşlarıma, sağladıkları manevi destek için,

Değerli arkadaşlarım Yrd. Doç. Dr. M. Alper Başaran, A. Alparlan Başaran, Burak Kasımoğlu ve Umut Arıcan'a , dostluklarıyla sağladıkları destek için,

Değerli çalışma arkadaşım ve dostum Yrd. Doç. Dr. Erol Eğrioğlu'na, güler yüzlü desteğiyle her zaman yanımda olduğu için,

Ve hayattaki en büyük şansım, sevgili ailem dayım Şinasi Aladağ, kuzenim Arda Aladağ, teyzem Mukaddes Aladağ, kardeşim Fevziye Gökçen Aladağ, annem Rezzan Aladağ, babam Hikmet Aladağ, anneannem Fevziye Aladağ, ve dedem Abdurrezzak Aladağ'a, sağladıkları eşsiz destek ve bulunduğu yere gelmemi sağlayan değerli emekleri için,

teşekkürlerimi sunarım.

# İÇİNDEKİLER DİZİNİ

	<u>Sayfa</u>
ÖZ .....	i
ABSTRACT .....	ii
TEŞEKKÜR .....	iii
İÇİNDEKİLER DİZİNİ .....	iv
ŞEKİLLER DİZİNİ .....	vi
ÇİZELGELER DİZİNİ .....	vii
1. GİRİŞ .....	1
2. TABU ARAMA VE TEMEL KAVRAMLARI .....	6
2.1. Tabu Arama .....	8
2.1.1. Tabu aramanın temel kavramları .....	8
2.1.2. Tabu aramanın genel işleyiş biçimi .....	15
2.2. Tabu Arama Algoritmasında Hafıza Kullanımı .....	17
2.2.1. Kısa dönem hafıza bileşeni .....	17
2.2.2. Orta dönem hafıza bileşeni .....	20
2.2.3. Uzun dönem hafıza bileşeni .....	21
2.3. Frekans Temelli Hafıza .....	23
2.3.1. Seçim kurallarının değiştirilmesi .....	25
2.4. Yeniden Başlama Stratejisi .....	25
2.5. Kritik Olay Hafızası .....	26
2.6. Stratejik Salınım .....	28
3. YAPAY SİNİR AĞLARI İLE ZAMAN SERİLERİNDE ÖNGÖRÜ .....	29
3.1. Yapay Sinir Ağları ve Bileşenleri .....	30
3.1.1. Mimari yapı .....	33
3.1.1.1. YSA yönteminde mimari seçimi problemi .....	36
3.1.2. Aktivasyon fonksiyonu .....	40
3.1.3. Öğrenme algoritması .....	42
3.2. Zaman Serilerinde Öngörü için YSA ile Modelleme .....	42
3.2.1. Deneme yanılma yöntemi .....	48

4. GELİŞTİRİLEN TABU ARAMA ALGORİTMASI .....	50
4.1. Başlangıç Seçimleri .....	50
4.1.1. YSA modellerinde başlangıç ağırlıklarının belirlenmesi .....	50
4.1.2. Öğrenme algoritması .....	52
4.1.3. Başlangıç havuzu çözümlemesi .....	52
4.2. Geliştirilen Tabu Arama Algoritmasının Elemanları .....	57
4.3. Geliştirilen Tabu Arama Algoritmasının İşleyişi .....	63
5. UYGULAMA .....	66
5.1. Uygulamada Kullanılan Zaman Serileri .....	66
5.2. Çözümlemede Kullanılan Mimari Seçim Yöntemleri .....	69
5.3. Uygulamada Elde Edilen Sonuçlar .....	70
6. SONUÇ VE TARTIŞMA .....	73
KAYNAKLAR .....	75



## ŞEKİLLER DİZİNİ

	<u>Sayfa</u>
Şekil 2.1. Tabu arama algoritmasının arama adımları .....	16
Şekil 2.2. Kısa dönemli tabu aramanın genel akış şeması .....	19
Şekil 2.3. Sadece yenilik temelli hafıza kullanıldığında, bir iterasyon adımı .....	20
Şekil 2.4. Stratejik salınım .....	28
Şekil 3.1. McCulloch-Pitts birim modeli.....	33
Şekil 3.2. Genel yapay birim modeli .....	34
Şekil 3.3. Çok tabakalı ileri beslemeli yapay sinir ağı .....	35
Şekil 3.4. Doğrudan bağlantı içeren ileri beslemeli yapay sinir ağı .....	36
Şekil 3.5. Eğitim ve test kümesi örneği.....	46
Şekil 3.6. Örnek yapay sinir ağı mimarisi.....	47
Şekil 4.1. Hareket türleri örnekleri .....	58
Şekil 5.1. Hasta sayıları zaman serisi.....	66
Şekil 5.2. Türkiye'ye gelen turist sayıları zaman serisi .....	67
Şekil 5.3. Avusturya bira tüketimi zaman serisi .....	67
Şekil 5.4. Kükürt dioksit miktarı zaman serisi .....	68
Şekil 5.5. Türkiye toplam elektrik tüketimi zaman serisi.....	68

## ÇİZELGELER DİZİNİ

	<u>Sayfa</u>
Çizelge 3.1. İstatistik terminolojisinde kullanılan terimlerin yapay sınır ağları terminolojisindeki karşılıkları.....	32
Çizelge 4.1. Örnek başlangıç havuzu çözümleri.....	53
Çizelge 4.2. Başlangıç havuzundaki mimariler için elde edilen HKOK değerleri .....	53
Çizelge 4.3. Hatalar matrisinde satırlardaki en küçük değerler.....	54
Çizelge 4.4. $x=[x_1 \ x_2]$ çözümünden elde edilen komşu çözümler .....	58
Çizelge 4.5. $x=[4 \ 11]$ çözümünden elde edilen komşu çözümler .....	59
Çizelge 4.6. $x'=[4 \ 12]$ çözümünün komşu çözümleri .....	59
Çizelge 5.1. Test kümeleri üzerinden elde edilen ortalama HKOK değerleri .....	71
Çizelge 5.2. Seçilen en iyi ve en kötü mimariler .....	72

## 1. GİRİŞ

Gelecekte ne olabileceğini tahmin etmek, çok uzun zamandır insanlığı düşündüren, merak uyandıran bir konudur. “Yarın yağmur yağıp yağmayacağı”, “gelecekte ne zaman büyük bir deprem meydana gelebileceği” ya da “ekonomide ne zaman bir kriz yaşanabileceği” gibi soruların cevapları daima merak uyandırmış ve insanlığı bu ve bunun gibi soruların cevaplarını aramaya yöneltmiştir. Tarihte krallar, padişahlar, devlet adamları, komutanlar ve bilim adamları başta olmak üzere birçok insan geleceği tahmin etmeye çalışmıştır. Bazen bilimsel bir bakış açısıyla, bazen de farklı inançlara göre doğaüstü güçlerle, gelecekte ne olacağı sorusunun yanıtı aranmıştır hep. Hangi yöntemle olursa olsun yapılan gelecek tahminlerine göre, bazen bir ülkenin kaderini belirleyecek ekonomik bir hamle, bazen çok büyük bir savaşın sonucunu şekillendirecek stratejik bir müdahale yapılmıştır. İnsanlık tarihi yapılan gelecek tahminleri doğrultusunda verilen büyük veya küçük kararlarla doludur.

Günümüzde, artan teknolojik olanaklar doğrultusunda giderek zorlaşan rekabet koşulları, bireylerin, işletmelerin, kurumların ya da ülkelerin “Gelecekte ne olabilir?” sorusunun cevabını daha büyük bir iştahla araştırmalarına neden olmaktadır. Günümüz rekabet koşullarında, en iyi stratejileri belirleyebilmek için geleceğe yönelik tahminler elde edebilmek amacıyla birçok yöntem geliştirilmiş ve geliştirilmektedir. Gelecek olayları ya da koşulları tahmin etmeye öngörü denir (Kadılar, 2005). Farklı disiplinlerden araştırmacılar daha güvenilir öngörüler elde edebilmek için çeşitli yöntemler üretmekte ya da mevcut yöntemleri daha da iyileştirmek için çalışmaktadırlar.

Günümüzde kamu ve özel kuruluşlar, işletmeler ya da şirketler hatta şahıslar geleceğe yönelik araştırmalar yaparak, gelecekte beklenen koşullara göre bugünden önlemlerini alabilmektedir. Başka bir deyişle, günlük kararların alınabilmesi için bile gelecek hakkında bilgi sahibi olunması gerekmektedir (Kadılar, 2005).

Bilimsel bakış açısından geleceği tahmin etmek için yararlanılabilecek iki yöntem mevcuttur. Bu yöntemlerden birincisi parapsikolojik yeteneklerin kullanımudur. İnsanlardaki parapsikolojik yetenekleri inceleyen bilim dalı parapsikolojidir.

Parapsikoloji'nin çıkışı 1932 yılında Amerika Birleşik Devletleri'ndeki Duke Üniversitesi'ne gelen Profesör Dr. Joseph Rhine ile başlamıştır. Doktor Zenner ile birlikte binlerce üniversite öğrencisi üzerinde, meşhur 25 kart deneyi ile telepatik yetenekleri olan öğrencileri araştırmışlardır. Doktor Rhine 1942 yılında ilk sonuçları açıkladığında bilim dünyası ayağa kalkmış ve onun çalışmalarını reddetmiştir. Çalışmalarının kabul edilmesi için aradan yıllar geçmesi gerekmiştir. 1960 yılına gelindiğinde Amerika'daki birçok üniversite parapsikoloji laboratuvarları kurarak, telepatik yetenekleri güçlü, geleceği önceden görebilen insanlar aramaya başlamıştır. Bu çalışmalar günümüzde de sürüp gitmektedir. Amerika Birleşik Devletleri'nde CIA ve FBI gibi kuruluşlar psişik yeteneklere sahip insanlar çalıştırmaktadır. Günümüzde dünyanın Rusya da dâhil olmak üzere 30'dan fazla üniversitesinde parapsikoloji deneyleri sürüp gitmektedir (Bektan, 2007).

Gelecek hakkında tahminler elde etmek için yararlanılabilecek ikinci bilimsel yol istatistiksel modellerin kullanılmasıdır. Günümüz dünyasında üretim, planlama, finans, yönetim ve bunun gibi birçok alanda, karar vericiler güvenilir öngörüler elde edebilmek için çeşitli istatistiksel modeller kullanmaktadır.

İlgilenilen alan ne olursa olsun, gelecek tahmini yapıldığında elde edilen öngörülerin güvenilir olması yani, yapılan tahminlerin gerçekleşecek durumları en iyi şekilde yansıtması amaçlanır. İstatistik Bilimi'nin bakış açısıyla, tahmin edilen olayla gerçekleşen olay arasındaki farkın, bir başka ifadeyle hatanın en küçük olması istenir. Kullanılan istatistiksel bir modelle elde edilen öngörü ve gerçekleşen değer arasındaki fark ne kadar az ise o kadar başarılı bir model kurulmuştur ve elde edilen öngörüler de o kadar güvenilirdir.

Gelecek tahmini elde etmek amacıyla üzerinde en çok çalışılan veri türlerinden biri zaman serileridir. Olayların zaman içinde gözlemlenmesi ile elde edilen ölçümlerin oluşturduğu serilere zaman serileri denir (Günay vd., 2007). Belli bir zaman aralığında, havadaki kirlilik oranı, su tüketimi, elektrik tüketimi, bir ülkeye gelen turist sayısı, bir hastaneye gelen hasta sayısı, kişi başına düşen milli gelir, işsizlik oranı, rüzgâr şiddeti, hava sıcaklığı, borsa değerleri ve bunun gibi farklı zaman serileri için öngörüler elde etmek amacıyla çeşitli matematiksel modeller kullanılmaktadır. Zaman serileri analizinde kullanılan bu yöntemlerden en çok tercih edilenleri, incelenen verinin yapısını açıklayabilen istatistiksel modeller,

yapay sinir ağırları (YSA) modelleri, bulanık zaman serisi öngörü modelleri ve çeşitli öngörü yöntemlerinin birlikte kullanılmasıyla oluşturulan melez modellerdir (Aladağ vd., 2009b). Zaman serileri öngörü yöntemlerinin temel dayanak noktası, geçmişte gözlemlenen değerler göz önünde bulundurularak oluşturulan modelle gelecek tahminleri yapmaktır. Öncelikle gözlemlenmiş zaman serisi değerlerini en iyi temsil eden istatistiksel model oluşturulur. Daha sonra oluşturulan bu model kullanılarak gelecek için tahminler elde edilir.

Daha önce de açıklandığı gibi, gerçekleşen değerler ile öngörüler arasındaki fark, bir başka ifadeyle hata ne kadar küçükse kullanılan öngörü modeli o derece iyi olarak kabul edilir. Geçmişte gözlemlenen değerleri en iyi temsil eden model bulunurken çeşitli hata ölçütlerinden yararlanır. Örneğin, bu ölçütler arasında literatürde en çok kullanılanlardan bir tanesi (1.1) eşitliğinde verilen hata kareler ortalaması karekök (HKOK) değeridir.

$$HKOK = \left( \frac{\sum_{i=1}^T (y_i - \hat{y}_i)^2}{T} \right)^{1/2} \quad (1.1)$$

Burada,  $T$  gözlem sayısını,  $y_i$  gözlemlenmiş değerleri,  $\hat{y}_i$  elde edilen tahminleri göstermektedir. En küçük HKOK değerine sahip olan model en iyi model olacaktır.

Zaman serilerinde öngörü elde etme amacıyla son yıllarda YSA modelleri birçok çalışmada başarıyla kullanılmaktadır (Eğrioğlu vd., 2008). YSA yönteminin zaman serilerini modellemesindeki başarısının altında yatan temel neden, verideki hem doğrusal hem de eğrisel yapıyı öğrenebilme yeteneğidir (Aladağ ve Eğrioğlu, 2005). Değişik alanlarda ve farklı türdeki zaman serilerinin öngörülmesinde YSA yöntemi kullanımının çok iyi sonuçlar verdiği görülmüştür (Qi ve Zhang, 2001; Buhamra vd., 2003). Kullanım kolaylığı, öngörü elde etmedeki başarısı ve herhangi bir varsayım kısıtı gerektirmemesi nedeniyle, YSA yönteminin zaman serilerinin öngörüsünde kullanımı günden güne artmaktadır.

YSA yönteminin bileşenleri genel olarak mimari yapı, öğrenme algoritması ve aktivasyon fonksiyonudur (Aladağ vd., 2009a). YSA yöntemi kullanılırken, yöntemin bileşenlerinin doğru belirlenmesi elde edilecek öngörülerin güvenilirliği

açısından çok önemlidir (Roy ve Mukhopadhyay, 1997). Bileşenlerin yanlış seçimi bazı durumlarda çok yanıltıcı sonuçlar doğurabilir (Aladağ vd., 2007). Zaman serileri öngörü probleminde YSA yönteminin sağladığı avantajlarına karşın, en iyi öngörüler üretecek YSA mimarisinin nasıl belirleneceği sorusunun kabul görmüş genel bir yanıtı yoktur. Zaman serilerinde öngörü elde etmek amacıyla YSA yöntemi kullanıldığında en önemli problem, *optimal* öngörüyü elde etmeyi sağlayacak *en iyi* ağı seçilmesidir (Bodyanskiy ve Popov, 2006). En iyi ağı belirlenmesinde en önemli konu tabakalarda kullanılacak uygun birim sayılarının belirlenmesi; bir başka deyişle en iyi mimarinin seçilmesidir. Buna karşın, belirtilen seçimin yapılabilmesini sağlayan genel bir teori bulunmamaktadır (Zou vd., 2007a; 2007b).

Literatürde, öngörü probleminde kullanılacak YSA modelinin mimari yapısının belirlenmesi yönünde yapılan çeşitli çalışmalar olmasına karşın, bu çalışmalar sadece özel durumlar için yapıldığından, günümüzde YSA öngörü uygulamalarının çoğunda halen mimari seçiminde deneme yanılma yöntemi en çok tercih edilen yöntemdir (Gareta vd., 2006; Gonzalez-Romera vd., 2007; Azadeh vd., 2007; Zou vd., 2007a). Buna karşın, deneme yanılma yönteminin kullanımında önemli bir sorun, sistematik bir yaklaşım olmaması ve hangi aralıktaki mimarilerin deneneceğinin belirsizliğidir. Bu nedenle, zaman serilerinde en iyi öngörüler elde etmek amacıyla, en iyi mimarinin belirlenmesinde kullanılacak yeni yöntemler üzerinde halen çalışılmaktadır. Yapılan tez çalışmasında, zaman serilerinde öngörü elde edilirken YSA yöntemi kullanıldığında, en iyi öngörüler veren mimarinin belirlenebilmesi için sistematik alternatif bir yöntem önerilmiş ve güvenilir öngörüler elde etmek amacıyla, en iyi YSA mimarisini belirlemek için bir tabu arama algoritması geliştirilmiştir.

En iyi öngörü değerlerini veren mimarinin bulunması problemi bir optimizasyon problemi olarak düşünülebilir. Bu optimizasyon probleminde, her bir mimari ayrı bir çözümlü, her bir mimariye karşılık gelen öngörü hata değeri amaç fonksiyonu değerini gösterir. Buna göre, tez çalışmasında, hatayı (YSA performans ölçütünü) en küçük yapan, en iyi mimari yapıyı bulabilmek için etkin bir tabu arama algoritması geliştirilmiştir. Geliştirilen algoritmada, tabu aramanın ileri düzey stratejileri kullanılarak, çözüm uzayının etkili bir şekilde aranması ve en iyi öngörü

değerlerini veren mimarinin belirlenmesi amaçlanmıştır. Geliştirilen yöntem ve literatürde en çok kullanılan yöntem olan deneme yanılma yöntemi farklı gerçek zaman serileri üzerinde uygulanarak, elde edilen sonuçlar karşılaştırılmıştır. Yapılan karşılaştırma sonucunda, önerilen yöntemin çok daha iyi sonuçlar, bir başka ifadeyle çok daha güvenilir öngörüler ürettiği görülmüştür.

Tezin ikinci bölümü tabu arama algoritmasının genel kavramlarını özetlemektedir. Üçüncü bölümde, zaman serilerinde öngörü elde edilmesinde, genel hatlarıyla YSA yönteminin nasıl kullanıldığı açıklanmıştır. Mimari seçimi için geliştirilen tabu arama algoritması dördüncü bölümde verilmiştir. Beşinci bölümde önerilen yöntemin ve deneme yanılma yönteminin gerçek zaman serilerine uygulanmasından elde edilen sonuçlar verilmektedir. Sonuç bölümünde ise uygulamadan elde edilen sonuçlar yorumlanmış ve tartışılmıştır.

## 2. TABU ARAMA VE TEMEL KAVRAMLARI

Günümüz optimizasyon problemlerinin çözümünde, en etkili sezgisel yöntemlerin başında gelen tabu arama algoritmasının tanımını vermeden önce sezgisel ve tam yöntem tanımlarını yapmak gerekir. Sezgisel kelimesi, eski Yunancada “problemleri çözmek için yeni yöntemler geliştirme” ya da “problem çözme sanatı” anlamına gelen “heuriskein” kelimesinden gelmektedir (Aladağ, 2004). Yöneylem araştırmasında “sezgisel” terimi, tam çözümü bulunması zor hatta imkansız olan problemler için makul çözümler bulmada kullanılan yöntemler için kullanılır. Özel olarak optimizasyonda bir sezgisel yöntem, optimale yakın ya da yaklaşık olarak optimal bir çözüme götüren (fakat bunu garanti etmeyen) stratejilere dayanan, hızlı ve pratik bir yöntem anlamına gelir. Buna göre, bu sezgisel yöntemlerden söz edildiğinde, “çözmek” fiili, “optimale, tatmin edici bir yakınlık bulmak” yan anlamına sahiptir. İyi sezgisel yöntemler, prensipte, izin verilen zamanda elde edilebilecek en iyi çözümü belirleyebilmelerine rağmen optimal bir çözüm bulmayı garanti etmezler (Murty, 1995).

Bir optimizasyon probleminde, eğer varsa, makul bir zaman aralığında optimal çözümü bulmayı garanti eden yöntemlere tam yöntemler ya da tam algoritmalar adı verilir. 1960 ve 1970’li yıllarda tam algoritmalar, doğrusal programlama, karesel programlama ve doğrusal olmayan programlama problemleri gibi belli türdeki optimizasyon problemleri için geliştirilen karmaşık matematiksel ifadelere dayanıyorlardı (Aladağ, 2004). Yukarıda verilen bu tür problemlerin, optimal çözümleri için etkili nitelendirmeler sağlayan optimalite koşulları bilinmektedir ve bu nedenle iyi problemler olarak da adlandırılmaktadırlar. İyi problemlerin çözümleri için geliştirilen tam algoritmalar da bu bilinen optimalite koşullarına dayanmaktadır. 1980’li yıllar ve sonrasında, karmaşık tam algoritmaların kullanımı, zamanın bilgisayar sistemleri sayesinde yaygınlaşmıştır (Murty, 1995).

İyi problemleri çözmek için kullanılan tam algoritmaların gelişimi, önemli bir araştırma başarısıydı. Buna karşın, yapılan bu çalışmalar sıfır-bir ve tam sayılı programlama problemleri ve kombinatoriyal optimizasyon problemleri gibi karmaşık problemler için makul zamanda optimal çözümü bulan herhangi bir tam algoritma üretememiştir. Birerleme adı verilen ve tüm olası çözümleri birer birer inceleyen yaklaşım kullanılarak, bilgisayar desteğiyle karmaşık problemlerin orta büyüklükte



olanlarının çözülebilmeye karşı, genelde birerleme yaklaşımının zaman gereksinimi, çözülen problemin büyüklüğüyle üstel olarak büyüdüğünden, gerçek hayatta karşılaşılan problemlerin çözümünde etkisiz kalmaktadır. Bu nedenle büyük boyutlu, karmaşık problemlerin çözümünde, araştırmacılar sezgisel algoritmaların kullanımına yönelmek zorunda kalmıştır (Murty, 1995).

Karmaşık problemler optimal çözümün bulunması için çok sayıda çözümün değerlendirilmesini gerektirir. Sezgisel yöntemler, kabul edilebilir zaman içerisinde çözüm ve değerlendirmelerin sayısını azaltma bakımından bu tür problemler için etkili olmaktadır. Karmaşık optimizasyon problemleri için tam çözüm yöntemleri bilinse dahi, aşırı hesaplama gerekliliği büyük bir engel olarak ortaya çıkmaktadır. Kullanışlı bir algoritma geliştirmek ve mevcut kaynaklarla bu engellemeleri ortadan kaldırmak için sezgisel yöntemlerin katkısı gerekmektedir (Aladağ, 2004).

Sezgisel yöntemleri kullanmanın getirdiği diğer bazı avantajlar aşağıdaki gibi özetlenebilir (Aladağ, 2004):

- Sezgisel yöntemler karar verici mekanizma için sadeleştirici olabilir.
- Sezgisel yöntem, herhangi bir tam yöntemin parçası olarak öğrenme amacıyla kullanılabilir.
- Gerçek hayatta karşılaşılan problemler için her zaman matematiksel formülasyon kurmak kolay olmayabilir ve problemi basitleştirmek gerekebilir. Bu basitleştirme sonucu oluşan hata, bir sezgisel yöntemin sağladığı optimale yakın çözümün sahip olduğu hatadan daha büyük olabilir (Kalınlı, 1996).

Yapılan açıklamalar ışığında, bir optimizasyon problemi için optimal çözümü bulmanın çok zor hatta imkansız olabildiği görülmektedir. Buna karşı, sezgisel algoritmaların kullanımıyla optimale yakın çözümler elde etmek mümkündür. Tam algoritmalar, matematiksel olarak tanımlanamayan problemlerin çözümünde etkili olamazlarken, sezgisel yöntemlerin bu tür problemler için bile başarıyla uygulanabildikleri kanıtlanmıştır (Osman, 1991).

Sezgisel yöntem tanımını ve günümüz optimizasyon problemlerinin çözümlenmesinde kullanıma nedenlerini açıkladıktan sonra, sezgisel algoritmaların en önemli sınıfını oluşturan meta sezgisel algoritmalara da kısaca değinilmelidir. Meta sezgisel terimi öncelikle, Glover'ın tabu arama yaklaşımından ilk kez bahsettiği makalede (Glover, 1986) geçmiştir ve daha sonra literatürde yaygın olarak kullanılmaya başlanmıştır (Glover, 1989b; Laporte ve Osman, 1995; Gendreau vd., 1995; Charon ve Hurdy, 1996; Osman ve Kelly, 1996).

Meta sezgisel algoritmalar kısaca, bazı yerel arama yöntemlerini ve ileri düzey farklı stratejileri kullanan yapılara sahip algoritmalar olarak tanımlanabilir (Reeves, 1993). Tabu arama, tavlama benzetimi, genetik algoritmalar ve YSA, meta sezgisel algoritmaların en bilinen örnekleri arasında yer almaktadır.

## **2.1. Tabu Arama**

Günümüzde farklı alanlardaki, çeşitli karmaşık optimizasyon problemlerinin çözümünde kullanılan en etkili sezgisel algoritmalarından biri tabu arama yöntemidir (Aladağ ve Hocoğlu, 2007). Tabu arama ilk olarak Glover tarafından öne sürülmüştür (Glover, 1989b; 1990b). "Tabu" kelimesi sözlük anlamı olarak, tehlikeli doğaüstü güçler tarafından korunan veya kutsal bir şeye saygısızlık edici davranışı ya da teması yasaklayıcı anlamına gelmektedir (Aladağ, 2004). Yöneylem araştırması alanında, tabu aramanın doğaüstü güçler ya da ahlaki konularla hiç ilgisi yoktur. Bunun yerine tabu arama, çözüm uzayında bazı bakımlardan güç bölgelerin geçilmesinde arama işlemine rehberlik etmek için birtakım kısıtlamaların oluşturulmasıyla ilgilidir. Bu kısıtlamalar, yasaklanmış olarak sınıflandırılan arama seçeneklerinin hariç tutulması ve değiştirilmiş değerlendirmelere ve seçim olasılıklarına dönüşüm gibi birkaç şekilde işler. Kısıtlamalar, hafıza yapılarından yararlanılarak oluşturulur ya da yaratılır (Glover ve Laguna, 1997).

### **2.1.1. Tabu aramanın temel kavramları**

Tabu arama, değişik türden karar problemlerindeki sözel ya da simgesel ifadelerle, bu ifadeleri matematiksel formülasyonlara dönüştürmeye gerek kalmadan, doğrudan doğruya uygulanabilir (Glover ve Laguna, 1997). Buna karşın, tabu aramanın temel kavramlarının ve belirli özelliklerinin tanımlanmasında bir temel teşkil etmesi amacıyla, matematiksel bir gösterim oluşturmanın yararlı olacağı

düşünülmüştür. Buna göre, belirli kısıtlar altında bir  $f(x)$  amaç fonksiyonunun optimal (minimum ya da maksimum) değeri araştırılacaktır. Tanımlamalar yapılırken ilgilenilen problem,  $x \in X$  kısıtı altında  $f(x)$  amaç fonksiyonunun minimum yapılması olarak düşünölmüştür.

Optimal değeri araştırılan  $f(x)$  amaç fonksiyonu doğrusal olabilir ya da olmayabilir.  $x \in X$  koşulu,  $x$  vektörü üzerindeki kısıtları göstermektedir. Bu kısıtlar doğrusal ya da doğrusal olmayan eşitsizlikleri içerebilir ve  $x$  vektörü öğelerinin tümü ya da bazıları kesikli değerler alabilir.

Yapılan tanımlar doğrultusunda, tabu arama yönteminin temel kavramları aşağıda verilmiştir:

#### *Hareket*

$X$  çözüm uzayında herhangi bir  $x$  çözümünden, hareket adı verilen bir işlemle bir  $x' \in X$  çözümü yaratılır. Problemin yapısına göre belirlenen farklı hareket türleri, tüm sezgisel algoritmalarda olduğu gibi, tabu arama yönteminde de aramanın performansını etkileyen en önemli bileşenlerden biridir (Aladağ ve Hocoğlu, 2007).

#### *Komşu çözüm*

Bir hareketin sonucu olarak  $x$  mevcut çözümünden yaratılan bir çözüme komşu çözüm denilir ve  $x'$  ile gösterilebilir. Komşu çözümlere deneme çözümleri de denilir. Deneme çözümlerine götüren hareketler deneme hareketleri olarak adlandırılır. Her bir  $x \in X$  çözümü, ilgili bir  $N(x) \subseteq X$  komşular kümesine sahiptir. Tanımlanan hareket türü ile  $x$  çözümünden  $x' \in N(x)$  komşu çözüme ulaşılır. Normalde komşuluk ilişkisinin simetrik olduğu varsayılır. Yani  $x'$ ,  $x$  çözümünün bir komşusu ise  $x$  de  $x'$  çözümünün bir komşusudur (Aladağ, 2004). Tanımlanan komşuluk yapısı, tabu arama algoritmasının işleyişini etkileyen en önemli faktörlerin başında gelir (Aladağ vd., 2009c).

### *Bir hareketin niteliği*

$x$  den  $x'$  çözümüne bir hareket sonucu çözümün değişen herhangi bir özelliği ya da bileşeni bu hareketin bir niteliği olabilir. Tek bir hareket birden fazla niteliğe de sahip olabilir (Sait ve Youssef, 1999).

Eğer çözümlerin ikili kodlandığı varsayılırsa  $x$  den  $x'$  çözümüne bir hareketin bazı mümkün nitelikleri aşağıda verilmiştir (Reeves, 1993):

- (a) Seçilen bir  $x_j$  değişkeninin 0 değerinden 1 değerine değişmesi,
- (b) Seçilen bir  $x_k$  değişkeninin 1 değerinden 0 değerine değişmesi,
- (c) (a) ve (b) deki değişkenlerin birlikte değişmesi,
- (d)  $f(x)$  den  $f(x')$  ne olan değişim,
- (e)  $g(x)$  den  $g(x')$  ne olan değişim ( $g$  doğal olarak problem formülasyonunda oluşan ya da stratejik olarak yaratılmış bir fonksiyon olabilir),
- (f)  $g(x') - g(x)$  fark değeri tarafından gösterilen değişim,
- (g) Eş zamanlı olarak ilgilenilen birden fazla  $g$  fonksiyonu için (e) ve (f) nin birlikte değişimleri.

Gerçekleştirilen tek bir hareket birden fazla niteliğin ortaya çıkmasına da neden olabilir. Örneğin, iki değişkenin değerini eşzamanlı olarak değiştiren bir hareket (a), (b) ve (c) niteliklerinin üçünün birden meydana gelmesine neden olabilir. (e), (f), (g) niteliklerinin dayandığı  $g$  fonksiyonu  $f$  amaç fonksiyonundan tamamen farklı olan ve stratejik olarak seçilmiş bir fonksiyon olabilir. Örneğin  $g$  fonksiyonu, verilen herhangi bir çözüm ile o ana kadar bulunan en iyi çözüm veya son ziyaret edilen yerel optimal gibi bir referans çözüm arasındaki farkın (ya da uzaklığın) bir ölçütü olabilir. Buna göre (f), bir deneme çözümünün, referans noktasına yakın mı olduğunu ya da aramayı referans noktasından daha ileriye mi götürdüğünü gösterir (Reeves, 1993).

### *Tabu kısıtlaması*

Hareketlerin seçilmiş niteliklerini *tabu* yaparak (yasaklayarak) önceden ziyaret edilmiş çözümlere geri dönmekten kaçınmak için yaratılan yasaklamalardır. Aramanın yerel optimal içeren bir bölgede sıkışıp kalmasını engelleyerek, böyle bölgelerin aşılmasına olanak sağlar (Aladağ, 2004).

### *Aspirasyon kriteri*

Mümkün olduğunda bir hareketin tabu statüsünü geçersiz kılmak için kullanılan bir araçtır. Bir başka ifadeyle, aspirasyon kriteri tabu olan bir hareketin yasak olmasına rağmen yapılabileceği durumları belirtmek için kullanılır. Uygun aspirasyon kriterinin kullanılması, tabu arama performansı üzerinde oldukça etkilidir (Pham ve Karaboğa, 2000).

Aspirasyon kriterinin basit bir türü, herhangi bir hareket o ana kadar görülmüş en iyi çözümden daha iyi bir çözüm üretiyorsa yani  $f(x') < f(x^{eniyi})$  ise bu hareketin tabu statüsünün geçersiz kılınmasıdır. Bu şekilde tanımlanan kritere, *Amaca Göre Tümel Aspirasyon* (Global Aspiration by Objective) adı verilir ve en yaygın kullanılan kriterdir. Bu yöntemin bir çeşidi de, belli bir bölgedeki en iyi çözümün amaç değerine göre tabu statüsünün iptal edilebildiği *Amaca Göre Yerel Aspirasyondur* (Regional Aspiration by Objective) (Woodruff ve Zemel, 1993).

### *Aday liste*

Gerçekleştirilen hareketlerin oluşturdukları komşulukların alt kümelerini içeren listedir. Kombinatoriyal optimizasyon problemlerinin çözümünde karşılaşılan ortak bir nokta, herhangi bir  $x$  çözümü için oluşturulabilecek  $N(x)$  komşuluğunun oldukça büyük olmasıdır. Özellikle problem boyutu büyük olduğunda  $N(x)$  komşuluğundaki tüm seçeneklerin incelenmesi hesaplama zorluğu nedeniyle çok pahalı hatta imkânsız olabilir. Böyle durumlarda, incelenen çözümlerin sayısını aday liste stratejileri kullanımıyla kısıtlamak mümkündür (Glover, 1989a; Glover, 1995; Hübscher ve Glover, 1994).

Bir aday liste stratejisinin etkinliği, sadece hesaplama zorluğunu azaltması açısından düşünülmemelidir. Bunun yanında, belli bir bilgisayar zamanı içinde

bulunan çözümün kalitesi de, kullanılan aday liste stratejisi için bir etkinlik ölçütü olmalıdır (Alabaş, 1999).

### *Tabu süresi*

Belirli bir tabu niteliği barındıran bir hareketin ya da çözümün yasaklı kalma süresidir. Bir tabu kısıtlamasını tanımlayan niteliklerin farklı türlerinin tabu sürelerinin belirlenmesinde farklı değerler kullanılmasına izin vermek genellikle akıllı bir tercihtir. Örneğin bazı nitelikler bir tabu kısıtlamasına diğerlerinden daha güçlü katkıda bulunabilirler. Bu durumda, kısıtın çok fazla katı olmasından kaçınmak için sözü edilen niteliklere daha kısa bir tabu süresi verilmelidir (Sait ve Youssef, 1999).

### *Tabu listesi*

Yasaklı çözümlerin ya da hareket niteliklerinin saklandığı listedir. Tabu listesinin belirlenmesinde birçok farklı yaklaşım kullanılmıştır. Tabu listesi oluşturmada önceleri her denenen çözüm tabu listesine kaydedilmekteydi. Çözümlerin kaydedilmesinde Ters Eleme Yöntemi (Dammayer ve Voss, 1993) ve Hashing Yöntemi (Battiti ve Tecchioli, 1992) sıklıkla kullanılmıştır. Hiçbir ek bilgi gerektirmemesi bu listenin tek avantajıdır, fakat bu yöntemlerde iterasyon arttıkça tabu listesinin boyutu büyümekte, liste üzerinde arama zorlaşmakta ve bellekte çok büyük yer kaplamaktadır.

Tam tabu listesindeki liste uzunluğunun çok fazla olması dezavantajından kaçınmak için liste uzunluğunun sabit tutulduğu sınırlı tabu listesi geliştirilmiştir. LIFO (Last In First Out - Son Giren İlk Çıkar) veya FIFO (First In First Out - İlk Giren İlk Çıkar) kuralları ile oluşturulan listeler kullanılarak en son elde edilen çözümlerin yeniden denenmesine kısıtlama getirilmiştir (Rangaswamy vd., 1998).

Tabu arama algoritmasının parametrelerinden biri tabu listesinin büyüklüğüdür. Gereken hafıza uzayının büyüklüğü, nitelik ve komşuluk büyüklüğüne bağlıdır fakat tabu listesi uzunluğu bağlı değildir. Tabu listesinin uzunluğunun doğru saptanması çok önemlidir. Liste boyunun küçük tutulması, algoritmanın önceden denenmiş çözümlere tekrar geri dönmesine, dolayısıyla kısır döngüye girmesine,

liste boyunun yeterinden büyük seçilmesi ise algoritmanın hiçbir hareket yapmadan kısa sürede tıkanmasına neden olur (Aladağ, 2004).

### *Hareket değerlendiricisi*

Hareket değerlendiricisi amaç fonksiyonunun ve arama tarihinin birleşik (composite) bir fonksiyonudur. Bir hareketin “iyiliğini” (goodness) belirlemek için kullanılır (Aladağ, 2004).

### *Hareket değeri*

Bir hareketin değeri amaç fonksiyonundaki azalıştır ya da daha genel olarak hareket değerlendirici fonksiyondaki azalıştır (Aladağ, 2004).

### *Elit çözüm*

Yüksek kaliteli yerel optimal çözüm, elit çözüm olarak adlandırılır (Glover ve Laguna, 1997).

### *Hafıza*

Arama sırasında elde edilen çeşitli türde bilginin saklandığı yapıya hafıza adı verilir. Tabu aramada hafıza yapıları dört boyuta dayanarak işlerler. Bunlar, *yenilik* (recency), *frekans* (frequency), *kalite* (quality) ve *etki* (influence) boyutlarıdır (Aladağ, 2004). Yenilik ve frekans temelli hafıza birbirini tamamlayan yapılardır ve ilerleyen bölümlerde ayrıntılı olarak açıklanacaktır.

### *Kalite*

Hafıza yapılarında kalite boyutu arama sırasında uğranmış çözümlerin derecelerini ayırma yeteneğine karşılık gelir. İşlemsel olarak kalite, iyi çözümlere götüren hareketleri desteklemek için ödüllerin, zayıf çözümlere götüren hareketleri engellemek için cezaların sağlandığı, yoğunlaşma stratejisine dayalı öğrenme için bir temel oluşturur (Aladağ, 2004).

### *Etki*

Etki boyutu sadece kalite üzerinde değil, aynı zamanda arama sırasında yapılan seçimlerin etkileri üzerinde de durur. Bir açıdan kalite etkinin özel bir şeklidir

(Glover ve Laguna, 1997). Kalite boyutunda ilgilenilen nokta, çözümlerin derecelendirilmesiyle, etki boyutunda önemli olan, çözümler arasındaki farklılığın büyüklüğüdür.

#### *Açık (explicit) hafıza*

Açık hafıza çözümlerin tam olarak saklandığı hafıza biçimidir. Açık hafızada genellikle arama sırasında karşılaşılan elit çözümler saklanır. Bu hafızanın daha gelişmiş elit çözümlerin oldukça çekici (attractive) ancak incelenmemiş komşularını kaydeder. Hafızaya alınmış elit çözümler yerel aramayı genişletmek için kullanılır (Aladağ, 2004).

#### *Niteliksel (attributive) hafıza*

Niteliksel hafıza bir çözümden diğerine yapılan harekette değişen çözüm nitelikleri hakkındaki bilgiyi kaydeder (Glover, 1995). Açık hafıza yapısında çözüm tam olarak saklandığından, hem hafıza kullanımı artar, hem de algoritma hızı düşer. Bu nedenle tabu aramada alternatif olarak bazı durumlarda niteliksel hafıza kullanılabilir.

#### *Yoğunlaşma stratejisi*

Yoğunlaşma stratejileri geçmişte bulunan iyi çözüm özelliklerini ve hareket kombinasyonlarını teşvik etmek için, değiştirilmiş seçim kurallarını kullanır. Aynı zamanda aramayı daha metodik ve dikkatli yapabilmek için çekici bölgelere bir geri dönüş başlatır. Anlık komşularının gözden geçirilmesi için elit çözümlerin kaydedilmesi gerektiğinden, açık hafıza, yoğunlaşma stratejilerinin uygulanmasında önemli bir rol oynar (Reeves, 1993). Yoğunlaşma stratejisinde bir çözümün komşuları yaratılırken, iyi çözümlerin bileşenleri birbirleri ile birleştirilir ya da böyle bileşenlerin ilk bilgisini geçerli bir çözüme dönüştüren değiştirilmiş değerlendirme stratejileri kullanılabilir (Aladağ, 2004).

#### *Çeşitleme stratejisi*

Çeşitleme stratejileri, ziyaret edilmemiş bölgeleri gözden geçirmek ve daha önce karşılaşılan çözümlerden birçok yönden farklı çözümler yaratmak için arama sürecini teşvik eder. Çeşitleme yaklaşımı da değiştirilmiş değerlendirme



stratejilerini kullanabilir. Buna örnek olarak ceza fonksiyonlarının kullanımı verilebilir (Glover, 1995).

### *Uyarlamalı hafıza programlaması*

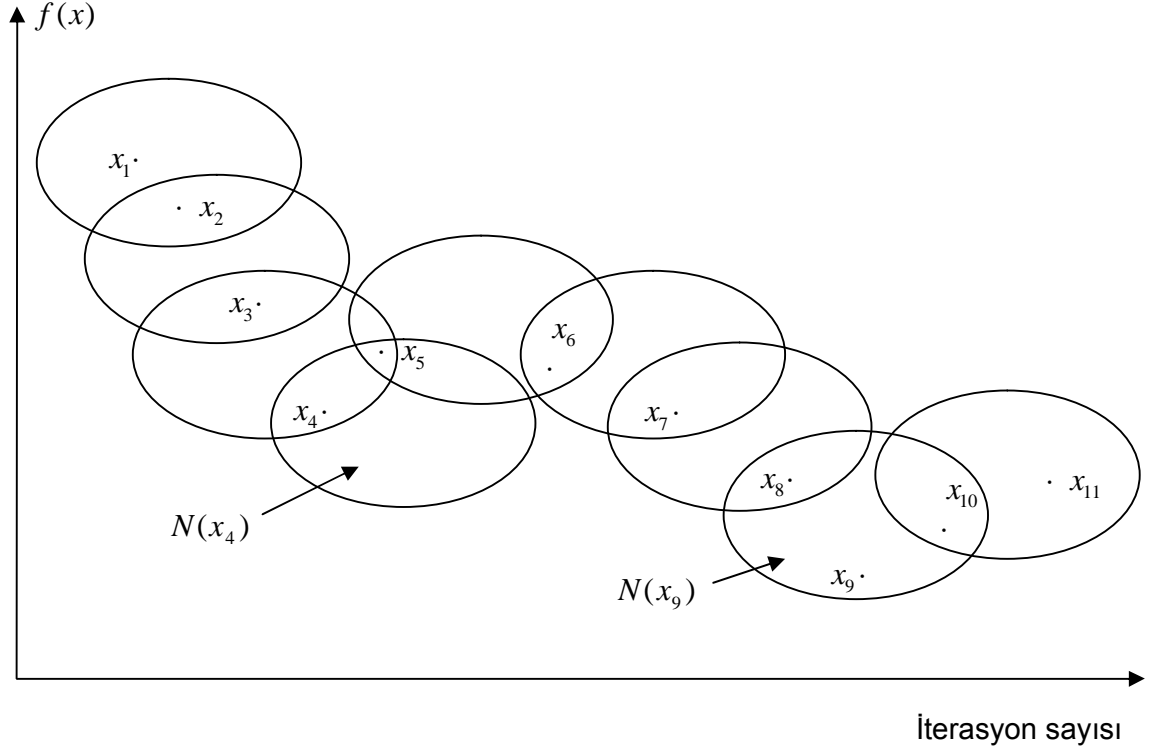
Optimizasyon problemlerinin çözümünde, tabu aramanın iki farklı türü kullanılır. Bunlardan biri tabu aramanın daha basit, diğeri daha ileri düzey olan iki farklı şeklidir. Basit olan yöntem, tabu arama modelinin kısıtlanmış bir parçasını içermekte ve bileşenlerinin sınırlı alt kümelerinin performanslarını test etmek için kullanılan başlangıç analizlerinde kullanılmaktadır (genellikle sadece kısa dönemli hafıza ile ilgilidir). Daha ileri düzey olan yöntem, daha geniş bir yapı şekillendirir. Bu yapı, yoğunlaşma ve çeşitleme stratejileri ile ilgili daha uzun dönemli hafızayı içerir. İkinci yaklaşım, stratejik hafıza bileşenlerinin bir derlemine kullanma üzerine odaklanmasına bağlı olarak, uyarlamalı hafıza programlaması adını da alır (Glover, 1996).

#### **2.1.2. Tabu aramanın genel işleyiş biçimi**

Aladağ, tabu arama algoritmasının genel olarak işleyişini aşağıdaki gibi açıklamıştır (Aladağ, 2004):

Şekil 2.1’de görüldüğü gibi aramaya  $x_1$  çözümü ile başlanır. Bu çözüm ya rastgele ya problemle ilgili mevcut bilgiye dayalı olarak ya da başka bir yöntem kullanılarak bulunabilir.  $f(x_1)$ , bu çözüm için amaç fonksiyonu değeridir.  $N(x_1)$ , bir hareketle  $x_1$  çözümünden ulaşılabilecek komşu çözümlerin kümesidir.  $x_1$  çözümünün bir  $x_2 \in N(x_1)$  komşusu, uygun bir komşuluk mekanizmasıyla  $\Delta = f(x_2) - f(x_1)$  fark değeri kullanılarak üretilir.  $\Delta \leq 0$  olduğunda yerel arama yöntemleri (örneğin iniş yöntemi gibi) ve tabu arama, üretilmiş bu komşuyu yeni çözüm olarak kabul eder.  $\Delta > 0$  ise iniş yönteminde,  $x_1$  mevcut çözüm olarak saklanmakta ve diğer komşuların bulunması için aramaya devam edilmektedir. Herhangi bir  $i$  için  $x_i$ ,  $N(x_i)$  komşuluğunda en iyi ise o zaman  $x_i$ , yerel optimal çözüm olarak bulunmuş olur. Tabu arama algoritması ise  $\Delta > 0$  olduğu zaman farklı davranarak, kötüleşmiş olan bu çözümü (amaç fonksiyonu değerinde artış olan çözüm) uyguladığı belirli kriterler sonucu yeni mevcut çözüm olarak kabul edebilmektedir. Tabu aramanın belirli kriterleri sağlayan kötüleşmiş çözümleri de kabul

etmesindeki amaç, yerel optimalin sınırlarını aşarak daha uzaklarda tümel (global) optimal için arama yapabilmektir (Kalınlı, 1996).



Şekil 2.1. Tabu arama algoritmasının arama adımları (Aladağ, 2004).

Şekil 2.1'deki  $x_4$  bir yerel optimal çözümü göstermektedir. Tabu arama  $f(x_5) > f(x_4)$  olmasına rağmen  $x_5$  çözümüne götüren hareketi kabul edebilir.  $x_5$  çözümünün mevcut çözüm olarak kabul edildiğini varsayalım. Eğer  $x_4$  mevcut çözümü  $N(x_4)$  komşuluğunda en iyi amaç fonksiyonu değerine sahipse, tabu arama en yüksek değerlendirme kriterini kullanır. Bu stratejide, komşulukta amaç fonksiyonu değerini en fazla düşüren çözüm en iyi çözüm olarak kabul edilir. Bu strateji ile böyle hareketlerin kabul edilmesi, aynı yerel optimal çözüme dönülmesi olasılığını oldukça artırır ve bu, aramanın döngüye girmesine neden olabilir. Döngüden kaçınmak için tabu arama algoritması, bazı nitelikleri ortaya çıkaracak hareketleri yasaklayarak (tabulaştırarak), geri kalan çözümler arasından en yüksek değerlendirme kriterli çözümün seçebilmesine olanak sağlamak amacıyla tabu listesini kullanır. Kabul edilen hareketlerin karakteristikleri tabu listesinde tutularak, daha sonraki belli bir iterasyon sayısı boyunca, önceden ziyaret edilen çözümlere geri dönüş engellenir. Başka bir ifadeyle, tabu listesi mevcut çözümden bir

hareketle ulařılabilecek çözümleri belirler. Tabu arama algoritması yeni bölgelerin araştırılmasını sađlayan ve döngüyü engellemek için tabu kısıtlamalarını kullanan bir yerel arama yöntemi gibi görülebilir. Tabu arama yönteminde arama, süre ya da amaç fonksiyonunun kabul edilebilir bir deęeri ya da başka bir şekilde belirlenebilecek bir durdurma kriteri tarafından sonlandırılır (Aladađ,2004).

## 2.2. Tabu Arama Algoritmasında Hafıza Kullanımı

Tabu arama yönteminin temel dayanak noktası, ařađıda verilen üç hafıza bileşeninin etkili biçimde kullanımındadır (Glover, 1990a; Sun ve McKeown, 1993):

- 1) Kısa dönem hafıza bileşeni: Tabu arama yönteminin çekirdeğini oluşturur.
- 2) Orta dönem hafıza bileşeni: Aramayı bölgesel olarak yoğunlařtırmak için kullanılan hafıza bileşenidir.
- 3) Uzun dönem hafıza bileşeni: Aramada tümel olarak çeşitleme etkisi yaratmak için kullanılan hafıza bileşenidir.

### 2.2.1. Kısa dönem hafıza bileşeni

Kısa dönem hafıza bileşeninin ana fikri, belirli arama yönlerini tabu olarak sınıflandırmak ve böylece yerel optimal tuzaklarından aramayı uzak tutmaktır. En çok kullanılan kısa dönemli hafıza, yakın geçmişte deęişen çözüm niteliklerini ya da çözümlerin kendisini saklar ve yenilik temelli hafıza olarak adlandırılır (Reeves, 1993). Yenilik temelli hafıza bu çözüm bilgisini saklamak için tabu listesini kullanılır. Tabu listesi kullanılarak önceden ziyaret edilmiş çözümlere geri dönmekten kaçınılmış olunur. Böylece arama yeni ziyaret edilmiş çözümlerden uzak tutulur (Aladađ, 2004). Aspirasyon kriteri sađlandığında ise çözümlerin ya da niteliklerin tabu durumu iptal edilir.

Kısa dönem hafızanın kullanımında, karşılaşılan durumların bir ayırt edici  $H$  tarihi, arama sürecini yönetmek için saklanır.  $N(x)$  komşu kümesi yerine  $H$  tarihinin bir fonksiyonu olan ve  $N(H,x)$  ile gösterilen bir deęiřtirilmiş komşuluk kümesi kullanılır.  $x$  çözümünden bir hareketle hangi çözüme ulařılabileceğini tarih belirler çünkü  $x'$  yeni çözümü,  $N(H,x)$  komşuluđundan seçilir (Reeves, 1993).

Sait ve Youssef tarafından kısa dönemli tabu aramanın yapay kodu aşağıdaki gibi verilmiştir (Sait ve Youssef, 1999):

Adım-1

Uygun bir  $x \in X$  başlangıç çözümüyle başla  
Tabu listesini ve aspirasyon düzeyini başlangıç durumuna getir

Adım-2

Sabit iterasyon sayısı için,

$V' \subset N(H, x)$  komşu çözümleri yarat

En iyi  $x' \in V'$  çözümünü bul

Eğer  $x$  den  $x'$  çözümüne olan hareket tabu listesinde ise,

Hareketi kabul et ve en iyi çözümü güncelle

Tabu listesini ve aspirasyon düzeyini güncelle

İterasyon sayısını arttır

Değilse,

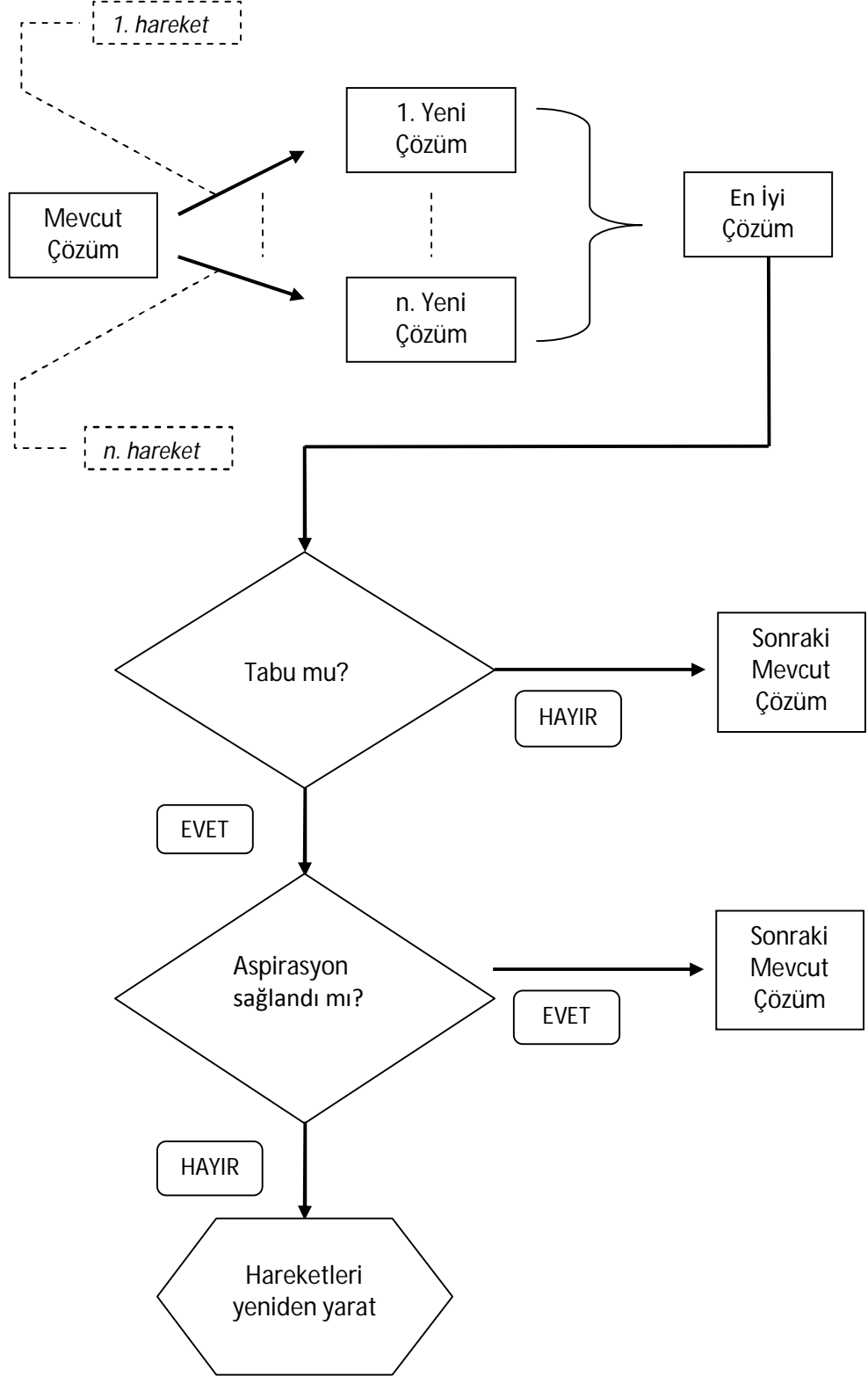
Eğer  $f(x') < \text{aspirasyon\_düzeyi}$  ise,

Hareketi kabul et ve en iyi çözümü güncelle

Tabu listesini ve aspirasyon düzeyini güncelle

İterasyon sayısını arttır.

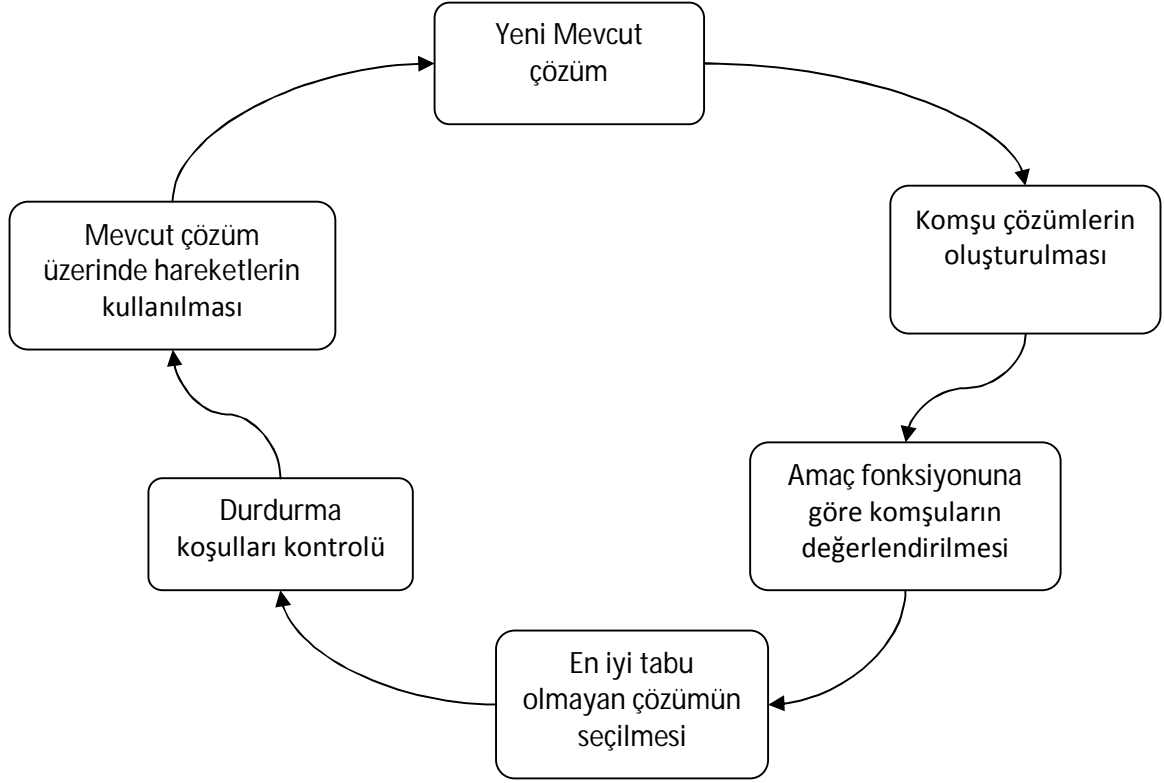
Kısa dönemli tabu arama,  $X$  çözüm uzayındaki bir  $x$  başlangıç uygun çözümden (*mevcut çözüm*) başlar. Her bir  $x$  için bir  $N(H, x)$  komşuluğu tanımlanır. Komşu çözümlerin bir  $V' \subset N(H, x)$  örnekleme yaratılır.  $V'$  kümesi aday listesini gösterir. Uç bir durum tüm komşuluğu yaratmak yani  $V' = N(H, x)$  almaktır. Bu genel olarak çok fazla işlem yükü gerektirdiği için elverişsiz olduğundan, komşuların küçük bir  $V' \subset N(H, x)$  ( $|V'| < |N(H, x)|$ ) örnekleme yaratılır. Aday listedeki komşu çözümler arasından en iyi  $x' \in V'$  çözümü seçilerek, yeni çözüm olarak düşünülür.  $x'$  çözümüne götüren hareket,  $x'$ ,  $x$  çözümünden kötü yani  $f(x') > f(x)$  olsa bile incelenir. Gerekli koşullar sağlandığında  $x$  den  $x'$  çözümüne bir hareket gerçekleştirilir (Aladağ, 2004).



Şekil 2.2. Kısa dönemli tabu aramanın genel akış şeması (Sait ve Youssef, 1999).

Kısa dönemli tabu arama algoritmasının genel akış şeması Şekil 2.2'de verilmiştir. Sadece tabu arama yönteminin çekirdeğini oluşturan kısa dönem hafıza bileşenini,

yani yenilik temelli hafızayı kullanan tabu arama algoritmasının işleyişinde, bir iterasyonda yapılan işlemler Şekil 2.3.'de verildiği gibi özetlenebilir. Şekil 2.3.'de verilen iterasyon adımında, önceden belirlenen durdurma koşulu ya da koşulları sağlandığında arama sonlandırılır.



Şekil 2.3. Sadece yenilik temelli hafıza kullanıldığında, bir iterasyon adımı.

### 2.2.2. Orta dönem hafıza bileşeni

Tabu arama algoritmasında, orta dönem hafıza bileşeninin temel rolü, umut vaat eden bölgelerde aramanın yoğunlaşmasını sağlamaktır. Yoğunlaşma stratejisi tabu arama yönteminin daha iyi sonuçlara ulaşmasını sağlayan önemli bileşenlerinden biridir. Yoğunlaşma stratejisi kullanımıyla arama daha agresifleşir. Yoğunlaşma stratejileri seçim kurallarını değiştirerek, geçmişte iyi olarak bulunan çözüm özelliklerini ve hareket kombinasyonlarını ödüllendirirler. Ayrıca bu stratejiler çekici bölgeleri daha iyi inceleyebilmek için bir geri dönüş başlatırlar. Tabu aramada genel olarak yoğunlaşma stratejisinin işleyişi Glover ve Laguna tarafından aşağıda verildiği gibi tanımlanmıştır (Glover ve Laguna, 1997).

Adım-1 Kısa dönem hafızalı tabu aramayı çalıştır.

Adım-2 Elit seçim stratejisini uygula.

Adım-3

- (A) En iyi deneme çözümlerinden birini seç.
- (B) Seçilen çözümü başlangıç çözümü olarak kısa dönemli tabu aramayı yeniden çalıştır.
- (C) Yeni bulunan uygun ve iyi çözümleri elit listesine ekle.
- (D) İterasyon sayısı önceden belirlenmiş bir sınırı aşarsa ya da elit listesi boşsa DUR, aksi halde Adım\_3 (A) dan tekrar devam et.

Orta dönem hafıza bileşeninde, aramanın belirli bir periyodu esnasında yaratılan en iyi deneme çözümlerinden seçilmiş  $n$  tanesi kaydedilir. Örneğin, seçilecek bu çözümler, ardışık çözümlerden  $n$  tane en iyisi ya da arama esnasında ulaşılan  $n$  tane yerel optimal çözüm olabilir (Sait ve Youssef, 1999). İyi olarak nitelendirilen bu seçilmiş çözümler ya da çözümlerin belli niteliklerini içeren yeni çözümler üzerinde yoğunlaşarak, daha iyi bir amaç değerine ulaşılmaya çalışılır.

### 2.2.3. Uzun dönem hafıza bileşeni

Tabu aramada uzun dönem hafıza bileşenin amacı aramanın farklı bölgelere yönlendirilmesiyle çeşitleme etkisinin yaratılabilmesidir. Tabu aramanın yerel optimal tuzagından kurtulmasını sağlayan bileşenlerinden biri çeşitleme stratejisidir. Bu nedenle, çeşitleme etkisinin yeterli ölçüde yaratılabilmesi algoritmanın optimal yakını çözümlere ulaşabilmesinde büyük önem taşır.

Burada bağlı kalınan ilkeler, orta dönem hafıza fonksiyonu tarafından kullanılan ilkelerle ters düşer. Orta dönem hafıza bileşeninde önceden bulunmuş iyi çözümleri içeren bölgelere aramanın daha yoğun olarak odaklanması sağlanır. Uzun dönem hafıza bileşeninde ise arama, şimdiye kadar incelenen bölgelerden farklı yeni bölgelere yönlendirilir (Reeves, 1993).

Aramanın çözüm uzayında henüz ulaşamadığı yeni bölgeleri keşfetmek amacıyla çeşitleme stratejileri kullanılır. Sezgisel arama yöntemleri genellikle, bir kombinatoriyal optimizasyon problemi tarafından tanımlanan çözüm uzayını keşifte etkilerini arttırabilmek için, çeşitleme stratejilerine güvenirlir. Bu stratejilerden

bazıları döngüden kaçınmak, yani arama sürecinde durmadan aynı hareket dizilerinin yapılmasını ya da daha genel olarak devamlı aynı çözüm kümelerinin yeniden ziyaret edilmesini engellemek amacı ile geliştirilmişlerdir. Diğerleri ise aramaya ek sağlamlık (robustness) vermek için ileri sürülmüş stratejilerdir. Çeşitleme stratejisi kullanılmadığında, arama çözüm uzayının küçük bir alanında yerleşir ve tümel optimal bir çözüm bulma olasılığı çok zayıflar (Sait ve Youssef, 1999).

Arama sırasında nadiren kullanılan ya da hiç kullanılmayan çözümleri ya da bu çözümlerin niteliklerini kullanabilmek için seçim kurallarının değiştirilmesi, tabu arama çeşitleme stratejilerinin temelini oluşturur. Tabu aramada çeşitleme stratejileri bu az kullanılan ya da kullanılmayan niteliklerin alt kümelerini aday çözümlere ekleyen yöntemleri tekrarlı olarak kullanabilir ya da çözüm sürecini kısmi veya tam olarak yeniden başlatabilirler. Tabu aramada çeşitleme stratejilerinin özellikle etkin oldukları nokta, daha iyi çözümlere ulaşmak için çözüm uzayındaki tümseklerin ya da engellerin aşılmasını sağlamalarıdır (Glover, 1995).

Arama yapılan çözüm uzayının zorluğuna bağlı olarak, bazen optimal yakını bir çözüme ulaşmak çok fazla işlem zamanı ya da yükü gerektirebilir. Böyle bir durumda, sağladığı diğer avantajlara ek olarak, uzun dönemli hafıza kullanımı uzun çözüm işlemleri gerçekleştirilmeden sonuca ulaşmaya olanak verir. Uzun dönemli hafıza kullanılmadığı durumda, harcanan zamandan daha az sürede çözüme ulaşmak mümkün olduğundan, uzun dönem hafıza kullanımıyla ekonomik bir zaman aralığında oldukça yüksek kalitede çözümler bulunabilir. Örnek olarak, bazı yol ve çizelgeleme problem türleri için en hızlı yöntemler, uzun dönemli tabu arama hafızasını içermektedirler. Ek olarak, optimal çözüme henüz ulaşılmadığı bir durumda, arama yapılan süre gittikçe artarken, daha iyi bir çözüm bulma şansının, sadece kısa dönemli hafıza kullanımına oranla uzun dönemli hafıza kullanımıyla arttığı da gösterilmiştir (Dammayer ve Voss, 1993).

Çeşitleme stratejilerinden bahsedilirken, bilinmesi gereken önemli bir nokta, rastgeleleştirme ile çeşitleme terimleri arasındaki farklılıktır. Çözümlerin çeşitlendirilmiş bir derlemine aramak, çözümlerin rastgeleleştirilmiş bir derlemine aramaktan tamamen farklıdır. Tabu aramada, genel olarak sadece çeşitlenmiş bir derlem ile ilgilenilmez, aynı zamanda çeşitlenmiş bir dizi (sequence) ile de ilgilenilir



çünkü genellikle incelenen elemanların sırası tabu arama yöntemi için önem taşır. Örneğin yeni (daha önce görülmemiş) çözümlerin bir dizisi tanımlanırken, her bir ardışık çözümün önceden yaratılmış tüm çözümlerle ilişkili olarak maksimum çeşitli (maximally diverse) yani önceki tüm çözümlerden farklı olması istenir (Pham ve Karaboğa, 2000).

### **2.3. Frekans Temelli Hafıza**

Tabu aramada uzun dönem hafızayı kullanan çeşitleme stratejileri, aramayı keşfedilmemiş bölgelerdeki yeni başlangıç noktalarına götürmek için bir değerlendirici (evaluator) kullanır (Glover, 1989b). Kullanılan değerlendirici ile seçim kuralları değiştirilerek çeşitleme etkisi yaratılır. Örneğin, gezgin satıcı probleminde, uzun dönem hafızanın basit bir şekli arama sırasında incelenen turlarda, her bir kenarın görünme sayılarını saklamaktır. Sonra bu bilgiye göre, çok fazla kullanılan kenarlar cezalandırılarak, aramanın daha sonraki adımlarında tekrar fazlasıyla ortaya çıkmalarını önlemek için bir değerlendirici kullanılır. Böylece geçmişte çok kullanılmış kenarları içermekten kaçınılarak, diğer umut verici turların yaratılması tercih edilmiş olur. Bir başka ifadeyle, aramanın belli kenarları içeren çözümleri barındıran bir bölgeye takılı kalması engellenmiş olur. Yenilik temelli tabu kriterinin tersine, bu çeşit yaklaşımda kullanılan kriter frekans temellidir. Bu şekilde işleyen bir uzun dönem hafıza, uzun dönemli bir tabu listesini ya da herhangi bir diğer uygun veri yapısını kullanabilir (Sait ve Youssef, 1999).

Tercih edilen hareketlerin seçimi aşamasında yararlanılan değerlendirme esasları, frekans temelli hafıza kullanımıyla genişletilmiş olur. Yenilik temelli hafızadan elde edilen bilgiyi tamamlayıcı türde bir bilgi elde edilmiş olur. Yenilik temelli yaklaşımda olduğu gibi, çözüm kalitesi ve hareket etkisi hesaba katılır ve buna göre frekanslar çoğunlukla ağırlıklandırılır ya da alt sınıflara ayrıştırılır (Glover ve Laguna, 1997). Elde edilen frekans bilgisi genellikle çeşitleme etkisi yaratmak için kullanılmasına karşın, iyi çözümlerde çok görülen nitelikleri de işaret ettiğinden, yoğunlaşma etkisi yaratmak için de kullanılabilir. İyi çözümlerde sıkça görülen niteliklerin frekansları yüksek olacağından, algoritmayı iyi çözümlerin bulunabileceği bölgelerde yoğunlaştırmak için bu niteliklerden yararlanılabilir.

Frekanslar oranlardan oluşuyor olarak düşünülebilir. 'geçicilik' (transition) ve 'kalıcılık' (residence) adı verilen iki farklı ölçüt, bahsedilen oranların pay değerlerini

açıklar. Geçicilik ölçütü, bir niteliğin çözüme girerek veya çözümden çıkarak, belli bir yörüngede (trajectory) ziyaret edilmiş çözümleri değiştirdiği iterasyonların sayısını göstermektedir. Kalıcılık ölçütü ise, bir niteliğin, belli bir yörüngede ziyaret edilmiş çözümlere ait olduğu toplam iterasyon sayısını ya da bir niteliğin özel bir alt kümedeki çözümlere kaç kere ait olduğunun toplam sayısını göstermektedir (Aladağ, 2004). Paydalar ise genellikle

- (1) Paylar tarafından gösterilen tüm olayların toplam meydana geliş sayısı (ilgili iterasyonların toplam sayısı gibi),
- (2) Payların toplamı,
- (3) Payların ortalaması,
- (4) En büyük pay değeri

değerlerinden birini gösterir (Reeves, 1993). Payların ağırlıklandırılmış miktarları gösterdikleri durumlarda (bazen negatif değer alabilirler), (2) paydası mutlak değerlerin toplamı, (3) mutlak değerlerin ortalaması ve (4) mutlak değer olarak ifade edilir. Bu oranlar, niteliklerin ne sıklıkta değiştiğini belirten geçicilik frekanslarını ve niteliklerin ne sıklıkta yaratılan çözümlerin üyesi olduğunu belirten kalıcılık frekanslarını üretir (Reeves, 1993).

Yüksek kalitede çözümler içeren bir bölgede, elde edilen büyük bir kalıcılık frekansı değeri çok çekici bir niteliği işaret edebilir ya da eğer ilgilenilen bölge düşük kalitede çözümler içeriyorsa bunun tersini işaret edebilir. Diğer taraftan, eğer ilgilenilen bölge hem yüksek, hem de düşük kalitede çözümler içeriyorsa, büyük ya da küçük bir kalıcılık frekansı aramanın kısıtlanmasına neden olan yerleşik (ya da hariç) bir niteliği işaret edebilir. Buna göre, çeşitleme sağlamak ve aramanın bu bölgede sıkışıp kalmasını engellemek amacıyla, sözü edilen yerleşik niteliğin kullanılmamasına (ya da kullanılmasına) karar verilmesinde, elde edilen bu kalıcılık frekansı kullanılır. Yüksek bir geçicilik frekansı ise amaç fonksiyonunun daha iyi değerler alabilmesi için çözümlere dahil olan ya da çözümlerden çıkan bir niteliğin varlığının belirlenmesinde yardımcı olabilir (Glover, 1995).

Uzun dönemli hafızada geçicilik ve kalıcılık frekansları, aramanın yönünü belirlemede kullanılırlar. Bu amaçla, arama boyunca elde edilen frekans ölçütleri,

hangi hareketlerin seçileceğini belirlemek için ödül ve ceza bilgisine dönüştürülürler (Alabaş, 1999).

### 2.3.1. Seçim kurallarının değiştirilmesi

Tabu arama algoritmasında seçim kuralları şu şekilde değiştirilir: Elde edilen geçmiş bilgi doğrultusunda bir hareketin veya o hareketin üreteceği bir çözümün değeri,

$$\text{ÇözümDeğeri}^* = \text{ÇözümDeğeri} + d \times \text{Ceza} \quad (2.1)$$

şeklinde verilen ceza fonksiyonu kullanılarak değiştirilir. Burada *ÇözümDeğeri*, örneğin ilgilenilen hareketin ürettiği çözümün amaç fonksiyonu değerini gösterebilir. *Ceza* değerinin genellikle frekans ölçütlerinin bir fonksiyonunu ve *d* değerinin bir çeşitleme parametresini gösterdiği bu ceza yaklaşımı tabu aramada yaygın biçimde kullanılır. Daha fazla çeşitleme etkisi yaratmak için daha büyük *d* değerleri kullanılır. Örneğin, minimum yayılan ağaç probleminde çözümü daha sık değiştiren düğümler, diğer düğümleri içeren hareketlerin seçimini teşvik amacıyla daha ağır cezalandırılabilir. Negatif cezalar (ödülleri) düşük frekanslı nitelikleri teşvik amacıyla kullanılabilirler. Ceza, hareketlerin niteliklerine uygulandığı gibi hareketlerin sınıflarına da uygulanabilir (Glover, 1990c).

### 2.4. Yeniden Başlama Stratejisi

Tabu arama yönteminde yeniden başlama işlemi, çeşitleme etkisi yaratmak için kullanılan bir stratejidir. Algoritmanın işleyişi periyodik olarak kesilir ve yeni bir çözüm dizisi üretmek için tasarlanmış bir süreçle yeniden başlatılır. Yeniden başlama stratejisinin kullanılma amacı, aramayı çözüm uzayında daha önce ziyaret edilmemiş yeni bölgelere yönlendirmektir (Glover, 1995).

Rastgeleliğe dayanan klasik yeniden başlama stratejileriyle de çeşitleme etkisi yaratmak mümkündür. Buna karşın, daha stratejik yeniden başlatma işlemleri kullanılması, oluşturulacak tabu arama algoritmasını daha etkin kılacaktır. Çeşitlemenin genellikle bir şekilde rastgeleleştirmeye aynı olduğu konusunda gizli bir varsayım vardır. Birçok arama işleminde, çeşitleme etkisi yaratmak için rastgele bir eleman tanımlanması yaygın bir işlemdir, hatta tavlama benzetimi ve genetik algoritmalar temelde böyle bir işleme dayanır. Rastgeleleştirme ve çeşitleme

kavramları soyut bir açıdan birbirlerine eşit olarak da görülebilir. Buna karşın, tabu aramada çeşitlilik, belli bir kümenin (örneğin yeni başlangıç çözümlerinin ya da çözüm uzayında belli bir bölgedeki çözümlerin oluşturduğu bir küme) elemanları arasındaki farklılıklar anlamına gelir ve böyle farklılıkların tespit edilmesi ancak etkili bir arama stratejisiyle mümkündür. Bu nedenle, diğer arama yöntemlerindeki rastgeleleştirmenin popüler kullanımı tabu aramada etkisizdir. Tabu aramada genellikle farklılığın sağlanabilmesi için stratejik bir seçim en iyidir ve rastgele bir seçim en kötüdür (Glover ve Laguna, 1997).

Rastgeleleştirme terimi arama yöntemleri üzerinde çalışan araştırmacılar arasında popüler bir kavramdır ve arama yöntemlerinin çoğunda kullanılmaktadır. Einstein'ın "Tanrı'nın zar atmadığına eminim" sözü popülerliğini yitirmiştir. Araştırmacıların çoğu kör amaçsızlığın yarattığı mucizevî sonuçlara hayranlık duymaktadır. Tabu arama yönteminin işleyişiyle, bahsedilen bu anlayış birbiriyle ters düşer. Rastgeleliği kullanan olasılıksal tabu arama çeşidinde bile, rastgele etkilere bağlı olan elemanlar oldukça sınırlandırılır ve iyi belirlenmiş olasılıklar tarafından rastgeleleştirme kullanımı hafifletilir. Bir başka ifadeyle, rastgeleliğin yer aldığı durumlar dikkatli bir biçimde şekillendirilir. Tabu arama yönteminde zar atılmış olsa bile, atılan zarlar ağırlıklandırılmıştır. Örneğin, rastgeleleştirme yeniden başlama stratejisinin bir parçası olarak kullanıldığında, frekans bilgisi, yeni başlangıç çözümünün üretim sürecini etkileyen olasılık dağılımını tahmin etmek için kullanılabilir. Arama tarihi tarafından bu şekilde yönlendirilen rastgelelik artık bir kör mekanizma olmaktan çıkar (Glover ve Laguna, 1997).

## **2.5. Kritik Olay Hafızası**

Yeniden başlama işleminin amacı aramanın daha önce gidilmemiş bölgelere ulaşmasını sağlamaktır. Yeniden başlama işlemi belli periyotlarla tekrar tekrar uygulanırken, eğer daha önce aramanın yeniden başlatıldığı bir çözümden arama tekrar başlatılırsa, daha önce incelenmiş olan bölge bir daha incelenmiş olacak yani, kullanılan yeniden başlama stratejisi başarısız olacaktır. Bu nedenle, yeniden başlama stratejisinde tabu aramanın yeniden başlatıldığı yeni başlangıç çözümünün, önceki başlangıç çözümlerinden veya arama boyunca incelenmiş tüm diğer çözümlerden farklı olması istenebilir. Tabu aramada bu çeşitlemeyi

gerçekleřtirmek amacıyla kullanılan yöntemlerden biri Kritik Olay Hafızasının (Critical Event Memory) kullanımınıdır (Glover, 1996).

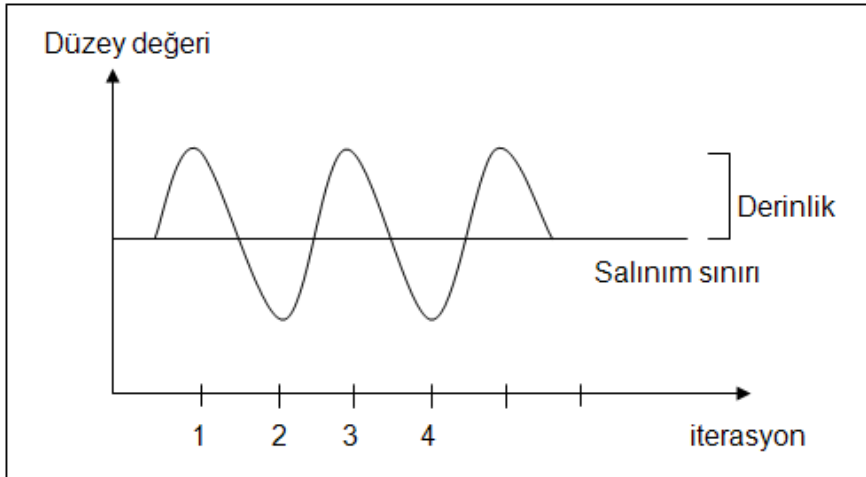
Yeniden başlama stratejisinin başarılı olabilmesi için yeniden başlangıç için seçilen çözüm noktaları arasında farklılık olması gerekmektedir. Bu nedenle, arama sırasında belirli kritik olayların oluşması izlenir ve bu olaylara ait bilgi kritik olay hafızasında saklanır. Daha sonra, elde edilen bu bilgi tabu aramanın yeniden başlatılacağı çözümlerin yaratılmasında kullanılır. Örneğin, yeni bir başlangıç çözümü aranırken önceki başlangıç çözümleri kritik olayların oluşturulmasında rol oynayabilir. Yeniden başlama işlemi çok kez uygulanırsa, tüm önceki başlangıç çözümleri kritik olay olur ve kritik olay hafızasında saklanır. Daha sonra yeni başlangıç çözümleri oluşturulurken bu kritik olaylar dikkate alınarak, önceki başlangıç çözümlerinden farklı yeni başlangıç çözümleri üretilir. Amaç farklı yeni başlangıç noktalarına bağlı olarak farklı çözüm dizileri elde etmek ve böylece, çözüm uzayının daha önce incelenmemiş bölgelerine ulaşmaktır (Glover, 1996).

Kritik olaylara karşılık gelen çözümlere kritik çözümler denilir ve bu çözümler farklı şekillerde tanımlanabilir. Arama sırasında incelenmiş çözümlerin her birini bir kritik olay olarak tanımlamak bu yollardan biridir, fakat çoğu kez daha küçük bir çözümler kümesini izole etmek de yeterlidir. Örneğin, kritik olay hafızası arama sırasında bulunan yerel optimal, yani amaç fonksiyonu değeri kendinden hemen önceki ve sonraki çözümden daha iyi olan ya da daha kötü olmayan, çözümlerden yararlanılarak da oluşturulabilir (Glover ve Laguna, 1997).

Yeniden başlama stratejisinin başarıyla uygulanabilmesi için yararlanılan kritik olay hafızası, tabu arama algoritması işleyişi üzerinde genel olarak çeşitleme etkisi yaratan bir bileşendir. Buna karşın, çeşitleme etkisi yaratma amaçlı kullanılan bu hafızayı yoğunlaşma etkisi yaratmak için de kullanmak mümkündür. Arama sırasında karşılaşılan iyi çözümleri içeren umut vaat eden bölgelere yeniden dönmek amacıyla da kritik olay hafızası kullanılabilir. Bahsedilen iyi çözümler kritik olaylar olarak tanımlanır ve gerekli bilgi hafızada saklanır. Daha sonra bu iyi çözümlere yeniden dönülerek, buldukları bölge daha iyi bir çözüm elde etmek amacıyla yeniden incelenir. Bu şekilde, umut vaat eden bölgeler üzerinde aramanın yoğunlaşması sağlanmış olur.

## 2.6. Stratejik Salınım

Tabu aramada stratejik salınım yaklaşımı, orta ve uzun dönem üzerinden, yoğunlaşma ve çeşitleme arasında anlamlı bir karşılıklı etkileşim sağlar. Stratejik salınım yaklaşımının kullanılmasıyla amaçlanan, aramayı önce seçilmiş bir uygunluk sınırına ve sonra bu sınırdan uzağa doğru yönlendirmektir. Belirtilen sınır, aramanın normalde daha ileri geçmeyeceği bir noktayı gösterir ve kritik düzey olarak da adlandırılır (Sait ve Youssef, 1999). Normalde dışına çıkılmasına izin verilmeyen bu sınırdan algoritmanın durmasını engellemek yerine, komşuluk tanımı genişletilerek ya da seçim kurallarında bazı değişiklikler yaratılarak, algoritmanın bu kritik düzeyin ötesine geçmesine olanak verilir. Buna göre, arama sınırının ötesinde önceden belirlenmiş bir uzaklıkta ilerler ve geri döner. Daha sonra, sınıra tekrar yaklaşılır ve geçilir, bu defa zıt yönden yeni bir dönüş noktasına ilerlenir. Tekrar tekrar farklı yönlerden sınıra yakınsama ve geçme süreci içinde arama belirtilen sınır etrafında bir salınım gerçekleştirilmektedir. Bu nedenle, Şekil 2.4.'de gösterilen bu yöntem stratejik salınım ismi verilmiştir. Dolaşılan bölgeye ve arama yönüne bağlı olarak değiştirilmiş hareket kuralları ve değerlendirmeleri yaratılarak bu salınım üzerinde kontrol sağlanır. Önceki bir yörüngenin tekrar izlenmesinden, standart tabu mekanizmaları (kısa dönemli hafıza fonksiyonları) kullanılarak kaçınılır (Reeves, 1993).



Şekil 2.4. Stratejik salınım (Aladağ, 2004).

Grafik ayırma, çizelgeleme ve diğer zor optimizasyon problemlerinin çözümünde stratejik salınımın başarıyla uygulanmıştır (Rolland ve Pirkul, 1991; Kelly vd., 1993; Mooney ve Rardin, 1993).

### 3. YAPAY SİNİR AĞLARI İLE ZAMAN SERİLERİNDE ÖNGÖRÜ

1920'lerden önce, zaman serilerinde öngörü değerleri basit ekstrapolasyon (dış değer bulma) işlemi ile elde ediliyordu (Pino vd., 2008). Daha sonra, Yule 1927 yılında güneş ışığının yıllık öngörü değerlerini elde etmek için otoregresif yöntemleri önerdi (Yule, 1927). Yule'un modeli öngörü değerlerini önceki gözlemlerin ağırlıklı toplamları olarak hesaplıyordu. Önerilen bu doğrusal yöntemden başarı elde edebilmek için gürültü (noise) olarak adlandırılan dış bir etkenin göz ardı edilmemesi gerekiyordu. Box ve Jenkins (1970) tarafından ARIMA (autoregressive integrated moving average) yöntemi önerilene kadar, gürültü etkenini içeren bu doğrusal sistem 50 yıl boyunca yaygın bir biçimde kullanıldı.

Bu noktadan itibaren, güçlü teorik çalışmalar durağan olmayan (non-stationary) ve doğrusal olmayan (non-linear) zaman serileri üzerinde yoğunlaştı. Örnek olarak çift doğrusal (bilinear), çift spektral (bi-spectral) ya da eşik (threshold) modelleri verilebilir (Tong, 1983, 1990; Priestley, 1988; Tsay, 1991; Rao, 1992). 1980'lerde zaman serileri araştırmalarını değiştiren iki önemli gelişme olmuştur. Birincisi, bilgisayarların özellikleri ve kapasitelerinin gelişmesi ile birlikte daha uzun ve karmaşık zaman serileri üzerinde çalışılabilmiş ve daha gelişmiş algoritmalar kullanılabilmiştir. Bu birinci gelişme ikinci gelişmeyi doğurmuş ve YSA gibi makine öğrenme (machine learning) özelliğine sahip yöntemler geliştirilmiştir (Pino vd., 2008). Zhang ve arkadaşları 1998 yılında YSA'nın zaman serileri çözümlemesinde kullanım şekilleri hakkında ayrıntılı bir çalışma yapmıştır (Zhang vd., 1998).

Günümüzde halen kullanılmakta olan Box ve Jenkins modelleri doğrusal modellerdir. Buna karşın, Granger ve Teravista gerçek hayatta karşılaşılan zaman serilerinin genellikle doğrusal olmayan bir yapıya sahip olduğunu belirtmişlerdir (Granger ve Teravista, 1993). Bu nedenle, doğrusal olmayan yapıdaki zaman serilerini çözümleyebilmek için literatürde çeşitli yaklaşımlar önerilmiştir. Önerilen yaklaşımlar arasında, çift doğrusal (Granger ve Anderson, 1978), eşik otoregresif (Tong ve Lim 1980), düzleştirme geçiş (smoothing transition) otoregresif (Bollerslev, 1986), otoregresif koşullu heterosekdestik (ARCH) (Engle, 1982) ve genelleştirilmiş otoregresif koşullu heterosekdestik (GARCH) modeller (Bollerslev, 1986) en çok bilinenlerdir. Belirtilen bu modeller, belirli doğrusal olmayan yapılar

için geliştirildiklerinden, zaman serilerinde görülen diğer doğrusal olmayan yapıları modelleyememektedir (Zou, 2007a). YSA yöntemi ise zaman serilerindeki hem eğrisel hem de doğrusal yapıyı öğrenebilme yeteneğine sahip olduğundan, diğer zaman serisi modelleme yöntemlerine göre daha iyi öngörü sonuçları verebilmektedir (Aladağ ve Eğrioğlu, 2005).

YSA yönteminin diğer klasik zaman serisi çözümleme yöntemlerine göre sağladığı başka bir avantaj, az sayıda gözlem içeren serilerle de çalışılabilmesine olanak sağlamasıdır. Ek olarak, YSA yöntemi klasik yöntemler gibi çeşitli model varsayımları gerektirmez. Model varsayımlarının sağlanmadığı ve doğrusal olmayan zaman serilerinin söz konusu olduğu durumlarda, Hill ve arkadaşları klasik zaman serisi yöntemlerinin yanlış sonuçlara götürebileceğini göstermişlerdir (Hill vd., 1996). Belirtilen bu durumlarda, doğru öngörü sonuçları elde edebilmek için YSA yönteminin kullanılması en iyi seçim olacaktır (Gutierrez vd., 2008).

Zaman serileri analizinde, YSA yönteminin neden klasik yöntemlere tercih edildiği sorusunun cevabı aşağıda verildiği gibi özetlenebilir (Aladağ vd., 2007).

- Zaman serisinin eğrisel mi yoksa doğrusal mı olup olmadığı test edilmeksizin çözümleme yapılabilir.
- Geleneksel zaman serisi yöntemlerinden daha iyi öngörü sonuçları elde edilebilmektedir.
- Geleneksel eğrisel zaman serisi modelleri sadece belli eğrisel yapılar için kullanıldığından genel olarak yeterli esnekliğe sahip değildir (Zhang vd., 1998). Buna karşın, YSA yöntemiyle çözümleme yapıldığında, zaman serisinin eğrisel yapısının nasıl olduğu önemli değildir.
- YSA yöntemi, geleneksel zaman serisi tahmin yöntemleri gibi karmaşık bir teoriye sahip değildir, anlaşılması ve uygulanması daha kolaydır.

### **3.1. Yapay Sinir Ağları ve Bileşenleri**

“Yapay sinir ağı nedir?” sorusu ilk yanıtlanması gereken sorudur. Picton bu soruyu, “Neden yapay sinir ağı olarak adlandırılmaktadır?” ve “Yapay sinir ağı ne yapar?” olmak üzere iki parçaya bölerek cevaplamıştır (Picton, 1994). Yapay sinir ağı olarak adlandırılmasının nedeni, birbirine bağlı birimlerden oluşan bir ağ



yapısında olmasıdır. Bu birimler biyolojik sinir sistemi çalışmalarından ilham alınarak tanımlanmıştır. Bir başka ifadeyle, YSA insan beynine benzer şekilde işleyen bir makine gibi tasarlanmıştır. Bu makinenin bileşenleri biyolojik nöronlar taklit edilerek oluşturulmuştur. “Yapay sinir ağı ne yapar?” cevaplanması gereken ikinci sorudur. Yapay sinir ağına bir girdi deseni verildiğinde buna göre bir çıktı deseni üretir. Ağa girdi olarak sunulan örneklerden öğrenilen bilgiye dayalı olarak bir çıktı değeri hesaplanır. Günümüzde YSA, kullanım kolaylığı ve ürettiği iyi sonuçlar nedeniyle, matematik, istatistik, fizik, mühendislik ve bilgisayar bilimleri uygulama alanlarında başarıyla kullanılmaktadır (Günay vd., 2007).

Mühendislerin, matematikçilerin, istatistikçilerin ve fizikçilerin yeni fikirler elde etmek için biyolojik bilimlere olan ilgisi uzun bir geçmişe sahiptir. YSA yönteminin oluşturulmasında biyolojik sinir ağlarından esinlenilmiş olmasına karşın, hala biyolojik sinir sistemleri ile YSA arasında hem mimarileri hem de yetenekleri yönünden büyük farklılıklar bulunmaktadır. Gerçek beyin fonksiyonları hakkındaki bilgi oldukça kısıtlıdır ancak bunu taklit etmek için yeterli küçük bir miktar bilgi bulunmaktadır. Hiçbir model insan beyninin performansını aynen taklit etmede başarılı olamamıştır. Bu nedenle, geliştirilen çeşitli yapay sinir ağı yapılarında insan beyni sadece bir mecaz olarak kalmıştır (Günay vd., 2007).

Öngörü elde etmede YSA, biyolojik sinir ağlarını taklit eden matematiksel modellerdir (Aladağ vd., 2009b). Matematiksel bir algoritma olarak tanımlanan yapay sinir ağları, örneklerden öğrenebilen ve öğrendiğini genelleştirebilen algoritmadır. Ağ gösterimi, matematiksel algoritmanın grafiksel ifadesidir (Günay vd., 2007).

Matematiksel bakış açısıyla, YSA fonksiyon yaklaşırıcı olarak da düşünülebilir (Hornik vd., 1989; Cybenko, 1989). Bir başka ifadeyle, YSA herhangi bir veri kümesini temsil eden en iyi fonksiyonu belirleyebilir. Özellikle fonksiyonlar karmaşık olduğunda bu çok önemli bir özelliktir. Ek olarak YSA, yapısı gereği doğrusal olmayan bir sistemdir (Rumelhart ve McClelland, 1987). Bu nedenle, yalnızca doğrusal fonksiyonları tahmin etmekle kalmaz aynı zamanda doğrusal olmayan fonksiyonları da belirleyebilir (White, 1992).

Birçok yapay sinir ağı modeli, iyi bilinen istatistiksel modellerle benzer ya da özdeş olmasına karşın, YSA literatüründe kullanılan terminoloji ile istatistik literatüründe kullanılan terminoloji arasında farklılık vardır. Literatürde YSA ile bazı istatistiksel modeller arasındaki ilişkiyi gösteren çalışmalar da bulunmaktadır (Sarle, 1994). Çizelge 3.1’de YSA ve istatistik terminolojilerinde aynı kavramları belirten bazı terimler gösterilmektedir (Nelles, 2001; Günay vd., 2007).

Çizelge 3.1: İstatistik terminolojisinde kullanılan terimlerin yapay sinir ağı terminolojisindeki karşılıkları.

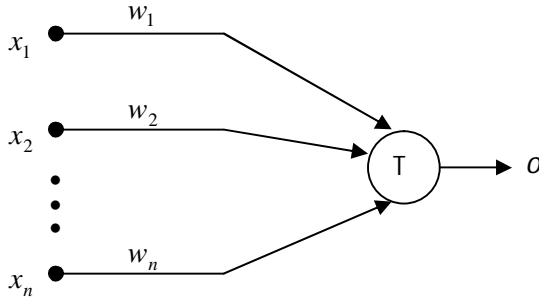
İstatistik terminolojisi	Yapay sinir ağı terminolojisi
Model	Yapay sinir ağı
Parametre	Ağırlık
Bağımsız değişken	Girdi
Tahmin değeri	Çıktı
Bağımlı değişken	Hedef
Artık	Hata
Güven aralığı	Hata çizgisi
Temel fonksiyon	Nöron
Bağımsız değişkenler kümesi	Girdi tabakası
Temel fonksiyonlar kümesi	Gizli tabaka
Tahmin değerleri kümesi	Çıktı tabakası
Gözlem	Örüntü
Kestirim ya da optimizasyon	Eğitim, Öğrenme ya da Adaptasyon
Örneklem adaptasyonu	Çevrimiçi öğrenme
Grup adaptasyonu	Çevrimdışı öğrenme
Kestirim ölçütü	Hata, Maliyet ya da Lyapunov fonksiyonu
Dönüşüm	Fonksiyonel bağlantı
Diskriminant analizi	Sınıflama
Regresyon	Eşleme, Yaklaşım ya da Denetimli öğrenme
Veri indirgeme	Denetimsiz öğrenme ya da şifreleme
Genelleştirme	İnterpolasyon ya da Ekstrapolasyon

Tabakalarda kullanılacak birim sayılarını gösteren “mimari yapı”, birimlerde kullanılan “aktivasyon fonksiyonu” ve ağın eğitim aşamasında yararlanılan “öğrenme algoritması”, YSA yöntemini oluşturan üç temel bileşendir (Aladağ vd., 2008). Belirtilen bileşenlerin doğru olarak seçilmesi, YSA yönteminin performansını belirleyen en önemli nedenlerden biridir (Aladağ vd., 2007). Bu bölümde, sözü edilen bileşenler kısaca tanıtılacaktır.

### 3.1.1. Mimari yapı

YSA birbirlerine bağılı yapay birimleri (düğümleri, nöronları) içeren bir ağ yapısına sahiptir. Yapay sinir ağı mimarisinin temel ögesi olan her bir birim için, birime gelen girdi değeri ve bu birimin diğere birim veya birimlere ilettiği bir çıktı değeri vardır. Birime gelen girdi değeri tek bir birimden gelen çıktı değeri olabileceği gibi birden fazla birimden gelen çıktı değerlerinin toplamı da olabilir. Birimin aldığı bu girdi değeri bir fonksiyon yardımıyla çıktı değerine dönüştürülür ve diğere birim veya birimlere girdi değeri olarak iletilir. Bir birimin çıktı değerinin diğere bir birime girdi değeri olarak iletilmesi işleme sinyal akışı adı verilir (Günay vd., 2007).

Bir yapay birim modelinin ilk resmi tanımı 1943 yılında McCulloch ve Pitts tarafından verilmiştir (Zurada, 1992). McCulloch-Pitts birim modeli Şekil 3.1'de görülmektedir.



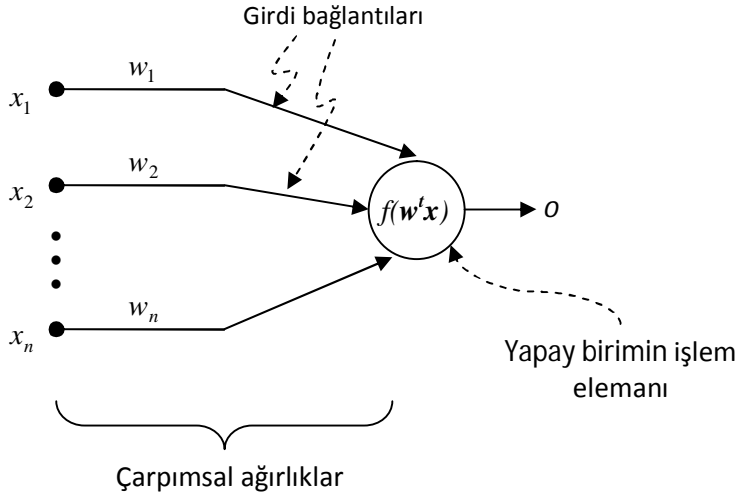
Şekil 3.1. McCulloch-Pitts birim modeli.

Burada  $x_i$  girdileri ( $i = 1, 2, \dots, n$ ),  $k$  anındaki girdi değerinin yokluğuna ya da varlığına bağılı olarak sırasıyla 0 ya da 1 değerini alır. Birimin çıktı değeri  $o$  ile gösterilmiştir. Bu model için  $k+1$  anında elde edilecek çıktı değeri

$$o^{k+1} = \begin{cases} 1, & \sum_{i=1}^n w_i x_i^k \geq T \\ 0, & \sum_{i=1}^n w_i x_i^k < T \end{cases}$$

şeklinde hesaplanır. Burada, diğere nöronların çıktısı olan her bir  $x_i$  girdi değeri, karşılık gelen  $w_i$  ağırlığıyla ( $w_i = \pm 1, i = 1, 2, \dots, n$ ) çarpılarak toplanır ve elde edilen değer eğer  $T$  gibi bir eşik değerinden büyükse, nöronun  $k+1$  anındaki  $o^{k+1}$  çıktısı 1 değerini alır, diğere durumda nöronun çıktı değeri 0 olacaktır (Günay vd., 2007).

Zaman serilerinin öngörüsünde kullanılacak birim modelinin tanımı için McCulloch-Pitts birim modeli bir başlangıç noktası oluşturur. Tanımlanacak yapay birim modelinde, ağdaki tüm birimlerin işlemlerinin eş zamanlı olduğu varsayılır. Çıktı değerleri [0,1] aralığında herhangi bir değer alabilir. Ağırlıklar model içinde düzeltilir ve sinyal akışı dışında ağın birimleri arasında hiçbir etkileşim olmaz. Buna göre,  $t$  üst indisi devrikliği göstermek üzere, kullanılacak genel yapay birim modeli Şekil 3.2.'de gösterilmiştir (Günay vd., 2007).



Şekil 3.2. Genel yapay birim modeli.

Mimari yapı içinde her birim, girdi bağlantılarına ve tek çıktıya sahiptir. Birimin çıktı sinyal akışı gibi,  $x_i$  birim girdilerinin sinyal akışı da tek yönlü olarak düşünülür. Eğer ağdaki tüm sinyal akışları ileri yönlü ise, bu ağa *ileri beslemeli yapay sinir ağı* adı verilir. Şekil 3.2'de görülen genel sembolik gösterim, yapay birimin işlem elemanını ve ağırlıkların bir kümesini gösterir.  $f(.)$  kullanılan aktivasyon fonksiyonunu göstermek üzere, birimin çıktı değeri

$$o = f(\mathbf{w}^t \mathbf{x})$$

ya da

$$o = f\left(\sum_{i=1}^n w_i x_i\right)$$

şeklinde verilir.  $\mathbf{w}$ , ağırlık vektörü

$$\mathbf{w} = [w_1 \quad w_2 \quad \dots \quad w_n]^t$$

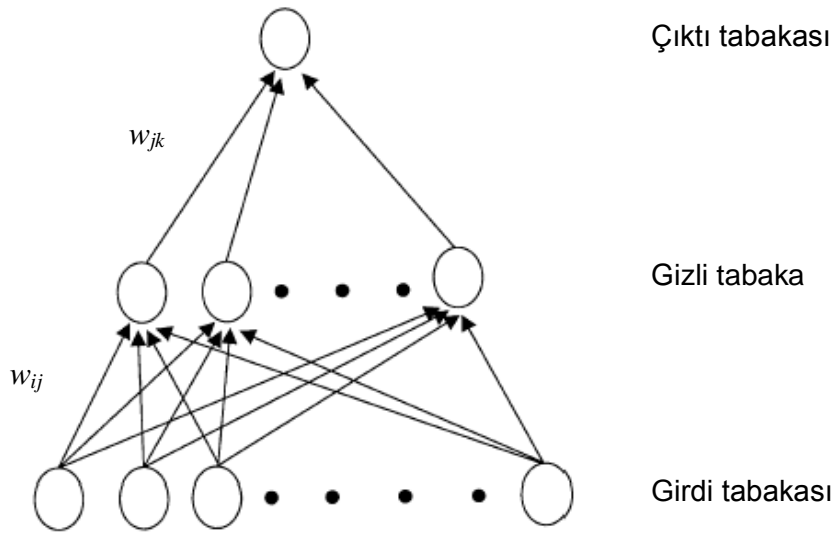
ve  $\mathbf{x}$  girdi vektörü

$$\mathbf{x} = [x_1 \quad x_2 \quad \dots \quad x_n]^t$$

şeklinde ifade edilir (Günay vd., 2007).

Girdi tabakası, gizli tabaka ve çıktı tabakası adı verilen üç tabakadan oluşan ağlara *çok tabakalı yapay sinir ağları* denilir. En basit hali ile çok tabakalı ileri beslemeli bir yapay sinir ağı mimari yapısı Şekil 3.3'de de verilmiştir. Mimarinin tam olarak belirlenmesi tabakalardaki birim sayısına karar verilmesi ile gerçekleşir. YSA da birimler birbirlerine ağırlıklar ile bağlıdır. Daha öncede ifade edildiği gibi, ileri beslemeli ağlarda bu bağlantılar tek yönlü ve ileri doğrudur. Aynı tabakanın birimleri arasında bağlantı yoktur (Aladağ vd., 2007).

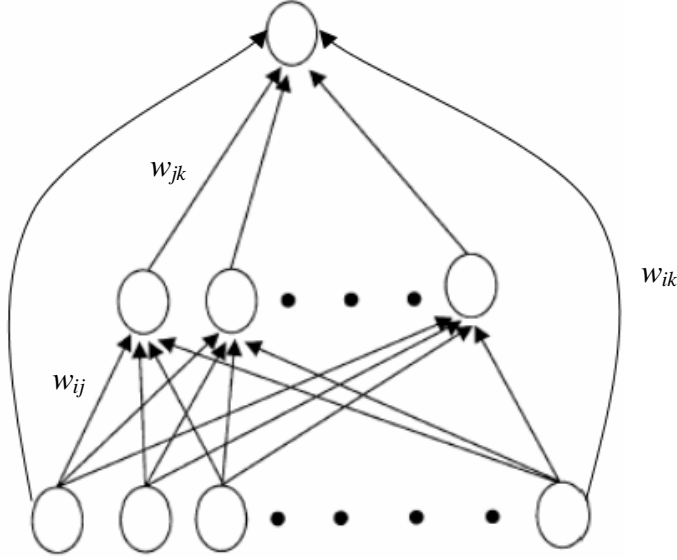
YSA mimarileri tek girdi ve tek çıktı tabakasından oluşur. Bunun yanında, mimariler birden çok gizli tabaka içerebilirler.



Şekil 3.3. Çok tabakalı ileri beslemeli yapay sinir ağı.

Şekil 3.3'de çıktı tabakasında yalnızca bir birim vardır. Çıktı tabakasında, isteğe göre birden çok birim kullanılabilir. Burada  $w_{ij}$ , girdi tabakasındaki  $i$  birimi ile gizli tabakadaki  $j$  birimi arasındaki ağırlığı ve  $w_{jk}$ , gizli tabakadaki  $j$  birimi ile çıktı tabakasındaki  $k$  birimi arasındaki ağırlığı gösterir. Şekilde çıktı tabakasında,

yalnızca bir birim olduğu için,  $k$  indisi bu tek birimi gösterir. Bir başka mimari çeşidinde, Şekil 3.4'de gösterildiği gibi, girdi tabakasındaki birimler ile çıktı tabakasındaki birim (ya da birimler) arasında doğrudan bağlantı da kurulabilir (Günay vd., 2007).



Şekil 3.4. Doğrudan bağlantı içeren ileri beslemeli yapay sinir ağı.

Yukarıda verilen şekilde,  $w_{ik}$ , girdi tabakasındaki  $i$  birimi ile çıkı tabakasındaki  $k$  birimi arasındaki ağırlığı göstermektedir. Zaman serilerinin öngörüsünde sağladığı başarıdan dolayı, en çok tercih edilen YSA modellerinden biri ileri beslemeli yapay sinir ağlarıdır (Zhang vd., 2001; Eğrioğlu vd., 2008). Bu nedenle, yapılan çalışmada da ileri beslemeli YSA mimarisi üzerinde yoğunlaşmıştır.

### 3.1.1.1. YSA yönteminde mimari seçimi problemi

YSA yöntemi kullanılırken, yöntemin bileşenlerinin doğru belirlenmesi elde edilecek öngörülerin iyiliği açısından çok önemlidir (Roy ve Mukhopadhyay, 1997). Doğru bileşenler seçilmediğinde, YSA yöntemi yanıltıcı sonuçlara götürebilir (Aladağ vd., 2007). Doğru bileşenlerin seçimiyle en uygun YSA modeli belirlenmeye çalışılır. En iyi bileşenin nasıl seçileceği konusunda genel bir kural henüz belirlenememiştir. Eldeki veriye göre en iyi olarak belirlenen bir bileşen, başka bir veri söz konusu olduğunda çok kötü sonuçlar verebilir (Bodyanskiy ve Popov, 2006).

Belirtilen bu üç bileşen arasında, belirlenme aşamasında en çok seçenek sunan kuşkusuz mimari yapıdır. Genelde kullanılan aktivasyon fonksiyonlarının sayısı altıyı geçmezken (Günay vd., 2007), MATLAB 2007 paket programında bir yapay sinir ağını eğitmede kullanılabilecek öğrenme algoritması sayısı on altıdır. Buna karşın, tabakalarda kullanılacak birim sayılarına göre yüzlerce hatta binlerce mimari seçeneği oluşabilir. Bu kadar çok seçenek arasından, en iyi sonuçlara götüreceği yapay sinir ağı mimarisinin belirlenmesi önemli bir problemdir.

YSA yönteminde mimari seçimi problemi için literatürde önerilen yöntemler, gözlemlere/tecrübelere dayanan ve sistematik olan yaklaşımlar olarak iki grupta toplanabilir.

#### **A) Gözlemlere/tecrübelere dayalı yaklaşımlar**

Literatürde, gözlem ya da tecrübeye dayalı olarak öne sürülmüş yaklaşımlar, tabakalara göre Günay ve arkadaşları tarafından aşağıdaki gibi verilmiştir (Günay vd., 2007):

##### *Girdi birimlerinin sayısının belirlenmesi:*

Girdi tabakasında, çok az ya da çok fazla girdi biriminin kullanılması yapay sinir ağının öngörü ve öğrenme yeteneğini etkileyecektir. Tang ve Fishwick (1993) girdi birimlerinin sayısının AR (autoregressive) terimlerinin sayısına bağlı olacağını düşünmüştür ancak Zhang bunun doğru olmadığını savunmuştur. Çünkü MA (moving average) modeli AR terimi içermediğinden ancak yapay sinir ağı ile tahmin edilebilir. Sharda ve Patil (1992) ve Tang ve Fishwick (1993) sezgisel olarak aylık veriler için 12 girdi birimini ve çeyreklik veriler için 4 girdi birimini kullanmışlardır. Buna karşın, Aladağ ve arkadaşları yaptıkları çalışmada, her mevsimsel zaman serisi için girdi birimlerinin sayısının serinin periyoduna eşit alınmasının iyi sonuç vermediğini göstermişlerdir (Aladağ vd., 2008). Lachtermacher ve Fuller (1995), bir adım ileri gelecek tahmini için fazla girdi biriminin kötü sonuç verdiğini ancak çok adım ileri gelecek tahmini için iyi etkileri olduğunu söylemişlerdir. Zhang ve arkadaşları ise parametre değerlerinin belirlenmesinin, eğrisel regresyondaki teorik çalışmalar ile yapılması gerektiğini savunmuştur (Zhang vd., 1998).

### *Gizli tabakaların ve birimlerinin sayılarının belirlenmesi:*

Gizli tabakadaki birimlerin sayısı zaman serisinin özelliklerini belirlemeye, verideki düzeni yakalamaya ve girdi çıktı birimleri arasındaki eğrisel eşlemeye imkan vermektedir. Öngörü probleminde, teorik çalışmalar, yapay sinir ağına tek gizli tabaka kullanılmasının yeterli olacağını ortaya koymuştur. Diğer yandan, tek gizli tabaka kullanımı gizli tabakadaki birimlerin çok sayıda alınmasını gerektirebilir. Bu da ağıın genelleştirme yeteneğini kötüleştirir. Lippmann (1987), Cybenko (1988), Lapedes ve Faber (1988) çalışmalarında, iki gizli tabakadan daha fazlasının daha iyi tahmin sonuçları vermeyeceği sonucuna ulaşılmıştır.

Genelleştirme yeteneğini artırdığından, gizli tabakalarda az birim kullanılması genellikle tercih edilmiştir. Gizli birimlerin sayısını belirlemede kullanılan en yaygın yöntem deneme yanılma yöntemidir. En az on girdi birimi almak gibi kaba bir kural da literatürde kullanılmıştır. Bir gizli tabaka kullanımı için bazı kılavuz yöntemler vardır. Örneğin Lippmann (1987) ve Hecth (1990), girdi birimi sayısı  $n$  olmak üzere, gizli tabakadaki birim sayısı olarak  $2n+1$  değerini kullanmıştır, Wong (1991)  $2n$  değerini, Tang ve Fishwick (1993)  $n$  değerini, Kang (1991) ise  $n/2$  değerini kullanmıştır. Ek olarak, Tang ve Fishwick 1993 yılında, gizli birimlerin sayısının öngörü performansı üzerindeki etkisini araştırmış ve gizli birimlerin sayısının öngörü performansı üzerinde çok etkili olmadığını öne sürmüşlerdir.

### *Çıktı birimlerinin sayısının belirlenmesi:*

Yapay sinir ağları yönteminde, çok adım ilerisi için öngörü yapmak amacıyla iki değişik yaklaşım kullanılmıştır. Bunlardan birincisi, Box-Jenkins yönteminde de kullanılan iteratif öngörü, yani ilk öngörünün, veri olarak kullanılıp, bir sonraki öngörünün elde edilmesi yöntemidir. Bu durumda yalnızca bir çıktı birimi kullanmak yeterli olmaktadır. İkinci yöntem ise, doğrudan öngörü yöntemidir. Doğrudan öngörü yönteminde birden çok çıktı birimine ihtiyaç duyulmaktadır. Hangi öngörü yönteminin daha iyi sonuç vereceği konusunda genel bir bulgu yoktur.



## B) Sistematik yaklaşımlar

YSA yönteminde mimari seçimi probleminin çözümü için literatürde kullanılan bazı sistematik yaklaşımlar

- Yapıcı/budama (constructive/pruning) algoritmaları (Siestema ve Dow, 1988; Fahlman ve Lebiere, 1990; Reed, 1993; Lahnajarvi vd., 2002),
- Polinomial zaman algoritması (Roy vd., 1993),
- Ağ bilgi kriteri yöntemi (Murata vd., 1994),
- Kanonik ayırma yöntemi (Wang vd., 1994),
- İteratif yapılandırma algoritması (Rathbun vd., 1997),
- Box-Jenkins yöntemine dayanan mimari seçimi yöntemi (Buhamra vd, 2003),
- Bilgi entropisine dayalı yöntem (Yuan vd., 2003),
- Genetik algoritmalar (Arifovic ve Gençay, 2001; Niska vd., 2004; Dam ve Saraf, 2006),
- Temel bileşenler analizi (Zeng vd., 2007),
- Ağırlıklı bilgi kriteri yöntemi (Eğrioğlu vd., 2008),
- Silme/değiştirme/ekleme algoritması (Durbin vd., 2008),
- Mevsimsel zaman serileri için mimari seçim yöntemi (Aladağ vd., 2008)
- Deney tasarımı (Balestrassi vd., 2009),

olarak sıralanabilir.

Zaman serilerinde YSA yöntemiyle öngörü elde edilirken en önemli problem, en doğru öngörülerini elde etmeyi sağlayacak en iyi mimarinin belirlenebilmesidir (Bodyanskiy and Popov, 2006). Buna karşın, en iyi mimarinin belirlenmesi için genel bir teori bulunmamaktadır (Zou vd., 2007a; 2007b) ve kapsamlı bir

simülasyon çalışması yapılmamıştır (Eğrioğlu vd., 2008). Yukarıda belirtilen, mimari seçiminde kullanılacak yöntemlerin bazıları zaman serileri öngörü problemine uygun değilken, bazıları da belli özel durumlar için geliştirilmiştir. Ek olarak, bu yöntemlerin hiç biri optimal çözümü garanti etmez. Bu nedenle, günümüzde YSA öngörü uygulamalarının çoğunda halen mimari seçimi için deneme yanılma yöntemi kullanılmaktadır (Gareta vd., 2006; Gonzalez-Romera vd., 2007; Azadeh vd., 2007; Zou vd., 2007a). Deneme yanılma yöntemi ise sistematik bir yaklaşım değildir ve hangi aralıktaki mimarilerin deneneceği sorusunun cevabı yoktur. Bu nedenle zaman serileri analizinde, en iyi öngörüler üretecek mimari yapının belirlenmesi problemi üzerinde çalışmalar halen devam etmektedir.

### 3.1.2. Aktivasyon fonksiyonu

Yapay sinir ağı birimlerinde kullanılacak aktivasyon fonksiyonu, oluşturulan modelin performansını belirleyecek önemli bir bileşendir ve kullanıldığı birimin girdileri ve çıktısı arasındaki ilişkiyi belirler. Bu nedenle kullanılan YSA modelinde, sunulan veri kümesiyle, elde edilecek çıktı arasındaki ilişki türü seçilecek aktivasyon fonksiyonu ile belirlenir. En büyük avantajı, eldeki verinin hem doğrusal hem de doğrusal olmayan yapısını öğrenebilme yeteneği olan yapay sinir ağının, doğrusal olmama derecesini gösteren bileşeni aktivasyon fonksiyonudur. Herhangi bir yapay sinir ağı, verilen girdi ve elde edilen çıktı değerleri arasında, seçilen aktivasyon fonksiyonunun sağladığı doğrusal olmayan eşleşmeyi gerçekleştirir.

Bir önceki kısımda, birim çıktı değerinin hesaplanmasında aktivasyon fonksiyonu  $f(.)$  ile temsil edilmiş ve buna göre,  $w$ , ağırlık ve  $x$  girdi vektörünü göstermek üzere, Şekil 3.2'de verilen birimin  $o$  çıktı değerinin

$$o = f(wx)$$

şeklinde hesaplandığı gösterilmiştir.

Bir yapay sinir ağı modelinde tüm birimlerde aynı aktivasyon fonksiyonu kullanılabileceği gibi, aynı ağ içinde farklı birimlerde değişik aktivasyon fonksiyonları kullanılabilir. Birimlerde genelde tek kutuplu, çift kutuplu ya da doğrusal aktivasyon fonksiyonu kullanımı tercih edilir. Tek kutuplu aktivasyon

fonksiyonunun çıktı değeri 0 ile 1 arasındayken, çift kutuplu aktivasyon fonksiyonu -1 ile 1 arasında bir değer üretir. En çok kullanılan aktivasyon fonksiyonları aşağıda verilmiştir (Günay vd., 2007):

a) Doğrusal aktivasyon fonksiyonu:

$$f(x) = x \quad (3.1)$$

b) Adımsal aktivasyon fonksiyonu:

$$f(x) = \begin{cases} 0, & x > 0 \\ 1, & x < 0 \end{cases} \quad (3.2)$$

ya da

$$f(x) = \begin{cases} -1, & x > 0 \\ 1, & x < 0 \end{cases} \quad (3.3)$$

c) Lojistik aktivasyon fonksiyonu:  $\lambda$  eğim parametresini göstermek üzere lojistik aktivasyon fonksiyonu

$$f(x) = \frac{1}{1 + \exp(-\lambda x)} \quad (3.4)$$

ya da

$$f(x) = \frac{1}{1 + \exp(-\lambda x)} - 1 \quad (3.5)$$

şeklindedir.

d) Radyal Temelli aktivasyon fonksiyonu:

$$f(x) = \exp(-\lambda x^2) \quad (3.6)$$

e) Tanjant hiperbolik aktivasyon fonksiyonu:

$$f(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)} \quad (3.7)$$

f) Sinüs, Kosinüs aktivasyon fonksiyonu:

$$f(x) = \text{Sin}(x) \text{ ve } f(x) = \text{Cos}(x) \quad (3.8)$$

(3.2) ile (3.4) eşitliklerinde verilen aktivasyon fonksiyonları tek kutuplu ve (3.3) ile (3.5) eşitliklerinde verilen fonksiyonlar ise çift kutuplu aktivasyon fonksiyonlarıdır.

### 3.1.3. Öğrenme algoritması

Yapay sinir ağı modelinin parametreleri ağıın ağırlık deęerleridir. Aęa girdi olarak sunulan verinin yapısının öğrenilmesi işlemi, eldeki veriyi en iyi temsil edecek ağırlık deęerlerinin bulunmasıdır. En iyi ağırlık deęerlerinin belirlenmesi sürecine ağıın eğitimi adı verilir. Yapay sinir ağıının eğitimi kullanılan öğrenme algoritması ile gerçekleştirilir. Bir başka ifadeyle, kullanılan öğrenme algoritmasıyla, eldeki veriyi en iyi temsil edecek model parametreleri belirlenir. Literatürde, YSA'nın eğitimi için önerilmiş çeşitli öğrenme algoritmaları bulunmaktadır. Bu algoritmalar arasında en çok tercih edileni Geri Yayılım öğrenme algoritmasıdır. Geri Yayılım algoritması hakkında ayrıntılı bilgi Cichocki ve Unbehauen (1993)'de bulunabilir.

YSA yönteminde en iyi ağırlıkların bulunması bir optimizasyon problemidir. Buna göre, öğrenme işlemi, çok boyutlu parametre uzayında, önceden belirlenmiş bir amaç fonksiyonunu optimum yapan çözümü arama işlemidir. Tanımlanan optimizasyon probleminin çözümü için geliştirilmiş çeşitli öğrenme algoritmaları vardır. Son yıllarda, yapay sinir ağlarının eğitiminde kullanılan yöntemler arasında sezgisel algoritmalar da bulunmaktadır. Genetik algoritmalar, tavlama benzetimi ve tabu arama algoritması, eğitim için en çok tercih edilen sezgisel algoritmalarıdır (Günay vd., 2007).

### 3.2. Zaman serilerinde öngörü için YSA ile modelleme

Yapay sinir ağıına girdi olarak sunulacak zaman serisinin öngörüsü için oluşturulacak modelin belirlenmesi

- Yapay sinir ağıındaki girdi birim sayısı, gizli tabaka sayısı, gizli tabakadaki birim sayısı ve çıktı tabakasındaki birim sayısı gibi deęişkenlerden oluşan mimari yapının,
- Tabakalardaki birimlerde kullanılacak aktivasyon fonksiyonununun,

- En iyi ağırlık değerlerine ulaşmada kullanılacak öğrenme algoritmasının ve bu algoritmanın parametrelerinin,
- Zaman serisinin eğitim ve test kümelerinin büyüklüklerinin,
- En iyi modelin belirlenmesinde kullanılacak performans ölçütünün,

ne olacağına karar verilmesidir. Yapılacak bu seçimler öngörü probleminde elde edilecek sonuçları önemli ölçüde etkileyebilen faktörlerdir (Günay vd., 2007).

YSA modeli oluşturulurken, yukarıda belirtilen kararlar arasında en zor olanı mimari yapının belirlenmesi olduğundan daha önce bahsedilmiştir. Bunun nedeni, mimari seçiminin en çok seçeneğin incelenmesi gereken durum olmasıdır. Diğer durumlar en fazla on beş ile yirmi arası seçenek sunarken, tabakalarda kullanılacak birim sayılarına bağlı olarak yüzlerce hatta binlerce farklı mimari yapı seçeneği bulunabilir.

YSA yönteminin üç temel bileşeni olan mimari yapı aktivasyon fonksiyonu ve öğrenme algoritması önceki kısımlarda tanımlanmıştır. Burada kısaca eğitim ve test kümeleri ile performans ölçütü tanımları açıklanacaktır.

#### *Eğitim ve test kümesi*

Zaman serileri analizinde YSA yöntemi kullanılırken, eldeki seri genellikle eğitim kümesi ve test kümesi olarak ikiye bölünür. Eğitim kümesi üzerinden en iyi ağırlık değerleri bulunurken, test kümesi üzerinden hesaplanan öngörü performans ölçütüne göre ağın genelleştirme yeteneği ölçülür. Bir başka ifadeyle, eğitim kümesi yapay sinir ağı modelinin geliştirilmesi için, test kümesi ise geliştirilen modelin öngörü yeteneğinin ölçülmesi amacıyla kullanılır. İncelenen zaman serisinin eğitim kümesindeki kısmı için en iyi tahmin değerlerini veren ağırlıklar, kullanılan öğrenme algoritması ile belirlenir. Daha sonra, serinin test kümesinde yer alan değerleri, hesaplanan en iyi ağırlık değerlerini kullanan model ile tahmin edilir. Son aşamada, gerçek değerler ile tahminler arasındaki farka dayanan belirli bir performans ölçütü tüm seçenek sinir ağı modelleri için hesaplanır ve en iyi değeri veren model seçilir. Literatürde, eğitim & test kümesi ayrımı için genellikle %90 & %10 veya %80 & %20'lik yapılar tercih edilmiştir. Örneğin, %90 & %10'luk

yapı kullanılması, zaman serisinin %90'lık kısmının eğitim, %10'luk kısmının test verisi olarak kullanılması anlamına gelmektedir (Günay vd., 2007).

Literatürde gerçekleştirilen bazı uygulamalarda, veri kümesi üçe de bölünebilmektedir. Buna karşın, incelenilen zaman serisi üçe bölündüğünde, ağıın eğitiminde kullanılabilir veri sayısı azalacağından, iyi öngörü sonuçları elde edilemeyeceği yapılan çalışmalarda görülmüştür. Daha büyük eğitim kümesi kullanıldığında daha güvenilir öngörüler elde edilebileceği Zhang ve arkadaşları tarafından gösterilmiştir (Zhang vd., 1998). Yine yapılan bir başka çalışmada, Nam ve Schaefer eğitim kümesinin büyüklüğü arttıkça yapay sinir ağı performansının daha iyi olacağını söylemişlerdir (Nam ve Schaefer, 1995).

### *Performans ölçütü*

Kullanılan yapay sinir ağı modelinin, incelenen zaman serisinin içerdiği deseni ne kadar iyi öğrenebildiğini belirlemek amacıyla kullanılan çeşitli performans ölçütleri bulunmaktadır. Öngörü problemde kullanılan ölçütlerin dayanak noktası, gerçek değer ile ağıın bu değer için verdiği tahmin değeri arasındaki farktır. Kullanılan değişik performans ölçütleri, sözü edilen bu farkı değişik şekilde ölçer.

En çok tercih edilen performans ölçütleri,  $z_p$ , yapay sinir ağı çıktı değerini,  $d_p$ , hedef çıktı değerini ve  $T$ , test kümesinin büyüklüğünü göstermek üzere

$$HKO = \frac{\sum_{p=1}^T (d_p - z_p)^2}{T} \quad (3.9)$$

$$HKOK = \sqrt{HKO} \quad (3.10)$$

$$NHKO = \frac{\sum_{p=1}^T (d_p - z_p)^2}{\sum_{p=1}^T (d_p - \bar{d}_p)^2} \quad (3.11)$$

$$MHO = \frac{\sum_{p=1}^T |d_p - z_p|}{T} \quad (3.12)$$

$$OMHY = \frac{100}{T} \sum_{p=1}^T \left| \frac{d_p - z_p}{d_p} \right| \quad (3.13)$$

şeklindedir. Burada, *HKO*, hata kareler ortalaması, *HKOK*, hata kareler ortalaması karekök değeri, *NHKO*, normalleştirilmiş hata kareler ortalaması, *MHO*, mutlak hata ortalaması ve *OMHY*, ortalama mutlak hata yüzdesini göstermektedir (Günay vd., 2007).

Smith, YSA yöntemi kullanılarak zaman serilerinde öngörü elde etmeyi, aşağıda verildiği gibi yedi adımda özetlemiştir (Smith, 2002).

### 1. Adım: Verinin ön işlenmesi

Birinci adımda, zaman serisinin gözlemleri küçük bir aralığa dönüştürülür. Örneğin, birimlerde kullanılacak aktivasyon fonksiyonu çift kutuplu ise, ağın girdi değerleri  $[-1,1]$  aralığına

$$x'_i = \frac{2x_i - (Maks(x_i) + Min(x_i))}{Maks(x_i) - Min(x_i)} \quad (3.14)$$

şeklinde dönüştürülebilir. Burada,  $x_i$ , girdi değerlerini,  $Maks(x_i)$  ve  $Min(x_i)$  sırasıyla girdi değerleri arasındaki en büyük ve en küçük değerleri göstermektedir. Birimlerde tek kutuplu bir aktivasyon fonksiyonu tercih edilecekse,  $[0,1]$  aralığına dönüşüm

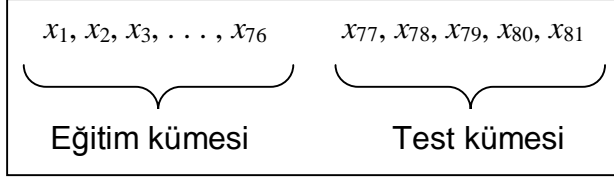
$$x'_i = \frac{x_i - Min(x_i)}{Maks(x_i) - Min(x_i)} \quad (3.15)$$

şeklinde gerçekleştirilebilir.

### 2. Adım: Test ve eğitim kümelerinin belirlenmesi

Zaman serisinin yüzde kaçının ağın eğitiminde ve yüzde kaçının en iyi YSA modelinin belirlenmesinde kullanılacağına bu adımda karar verilir. Bir başka ifadeyle, kullanılacak eğitim ve test kümelerinin büyüklüğü belirlenir.

Örneğin, 81 gözlem içeren bir zaman serisi incelenirken, eğitim kümesi için ilk 76 ve test kümesi için son 5 gözlem kullanıldığında,  $x_t$ ,  $t$  zamanında gözlenen değeri göstermek üzere, seri üzerinde yapılan seçim Şekil 3.5'te gösterilmiştir.



Şekil 3.5. Eğitim ve test kümesi örneği.

### 3. Adım: Modelleme

Modelleme adımında,

- Girdi birim sayısı, gizli tabaka sayısı, gizli tabakalardaki birim sayıları ve çıktı tabakasındaki birim sayısı, kısaca mimari yapı,
- Birimlerde kullanılacak aktivasyon fonksiyonu türü,
- Ağın eğitiminde kullanılacak öğrenme algoritması ve kullanılan bu algoritmanın parametreleri,
- Test kümesi üzerinden hesaplanacak performans ölçütü

seçimleri yapılarak, kullanılacak yapay sinir ağı modeli oluşturulur.

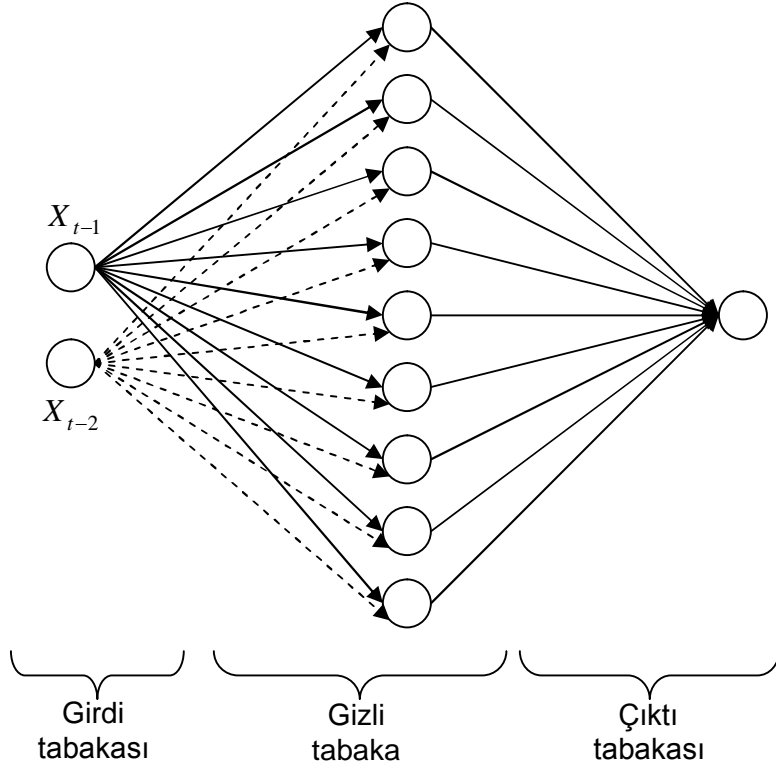
### 4. Adım: Girdi değerlerinin belirlenmesi

Kullanılan yapay sinir ağının girdi değerlerini, incelenen zaman serisinden elde edilen gecikmeli zaman serileri oluşturur. Örneğin, bir önceki adımda  $n$  tane girdi birimi kullanılmasına karar verilmişse, eldeki  $X_t$ , zaman serisi için girdi değerleri,

$$X_{t-1}, X_{t-2}, \dots, X_{t-n}$$

şeklinde verilen, gecikmeli zaman serileridir. Burada oluşturulan gecikmeli zaman serisi sayısının yine  $n$  tane olduğuna dikkat edilmelidir. Örneğin, iki birim içeren girdi tabakasına, 9 birim içeren tek gizli tabakaya sahip ve tek çıktı biriminden oluşan, ileri beslemeli bir yapay sinir ağı mimarisi Şekil 3.6'da verilmiştir.





Şekil 3.6. Örnek yapay sinir ağı mimarisi.

Şekil 3.6'da verilen mimari 2-9-1 mimarisi olarak adlandırılabilir.

### 5. Adım: Ağın eğitimi

Seçilen öğrenme algoritması, eğitim kümesi değerleri üzerinde kullanılarak, belirlenen yapay sinir ağı mimarisi için en iyi ağırlık değerleri hesaplanır. Elde edilen bu en iyi ağırlık değerleri kullanılarak, kurulan yapay sinir ağı modelinin çıktı değerleri eğitim kümesi için elde edilir. Bir başka ifadeyle, uyum iyiliği tahminleri hesaplanır.

### 6. Adım: Performans ölçütünün hesaplanması

Öncelikle oluşturulan yapay sinir ağının, bulunan en iyi ağırlık değerleri kullanılarak test kümesi için tahminler elde edilir. Tahminler elde edilirken, Bölüm 3.1.1.1.'de açıklanan iteratif öngörü ya da doğrudan öngörü yöntemlerinden biri kullanılır. Daha sonra, 5. adımda elde edilen çıktı değerlerine ve bu adımda elde edilen tahmin değerlerine, 1. adımda uygulanan dönüşüm işleminin tersi uygulanır. Bu dönüşüm işlemi sonucu elde edilen değerler sırasıyla, eğitim kümesinin

tahminlerini ve test kümesinin tahminlerini oluşturur. Son olarak, test kümesindeki gerçek gözlem değerleri ile test kümesi için elde edilen tahmin değerleri arasındaki farka dayalı olarak, seçilen performans ölçütü hesaplanır.

## **7. Adım: Öngörü**

Son adımda, öğrenme algoritmasının kullanımıyla 5. adımda bulunan en iyi ağırlık değerlerinden yararlanılarak, test kümesinde belirtilenden daha ileri zamanlar için yani, gelecek için öngörü değerleri, yine iteratif öngörü ya da doğrudan öngörü yöntemlerinden biri kullanılarak hesaplanır.

### **3.2.1. Deneme yanılma yöntemi**

Tek bir yapay sinir ağı kullanımıyla, zaman serileri için nasıl öngörü elde edileceği kısaca bir önceki kısımda açıklandı. Bir başka ifadeyle, YSA yöntemi bileşenlerine karar verildikten sonra nasıl çözümlene yapıp, öngörü değerlerinin nasıl elde edildiği özetlendi. Zaman serilerinde öngörü elde edilirken, YSA yönteminin kullanılması için beş etkenin belirlenmesi gerektiğinden bahsedilmişti. Bu seçimlere göre oluşturulacak yapay sinir ağıyla öngörü elde etmek, günümüzde bilgisayar programlarının yardımıyla çok kolaydır. Buna karşın, YSA modeli belirlenirken, YSA bileşenlerinin hangisinin seçiminin daha iyi öngörü sonuçlarına götüreceği sorusunun cevabını vermek, bu kadar kolay bir süreç değildir ve bu seçim kararları YSA yöntemi kullanımının en önemli noktasıdır. Yanlış bir bileşenin seçilmesi ya da eğitim test kümesi oranının kötü bir seçimle belirlenmesi hatalı öngörü sonuçlarına götürebilir. Bu durumda, YSA yönteminin sağladığı hiçbir avantajdan bahsedilemez. Güvenilir öngörü sonuçlarına ulaşabilmek için YSA modeli kurulurken yapılan seçimler bu nedenle çok büyük önem taşımaktadır.

Belirtilen etkenleri doğru bir şekilde seçmek ya da bir başka ifadeyle en iyi YSA modelini belirlemek, güvenilir öngörülere ulaşılmasında çözülmesi gereken bir problemdir. En iyi modeli belirleme probleminde, seçilecek etkenler için çok büyük, hatta bazı durumlarda sonsuz büyüklükte bir seçim aralığı bulunmaktadır. Bu nedenle, tüm etkenleri seçim aralıklarında değiştirerek en iyi modeli aramak neredeyse imkansızdır. Bunun yerine, en iyi model aranırken, etkenler için tecrübeye dayalı olarak seçim aralıklarını küçültmek ve bazı etkenleri değiştirirken, bazılarını sabit tutmak daha akıllı bir yaklaşımdır. Özet olarak, gerçek

hayatta, verilen bir zaman serisi için öngörü elde etmede, en iyi yapay sinir ağı modeli aranırken, tüm etkenlerin incelenmesi yerine, bazı etkenler sabit tutulurken, diğer etkenler için belli bir seçim aralığı incelenir (Günay vd., 2007).

Her bir etken için belirlenen seçim aralıklarına göre, farklı YSA model kombinasyonları teker teker denenir. Denenen her bir modelin test kümesi üzerinden performans ölçütü hesaplanır ve en küçük hatayı veren model en iyi model olarak seçilir. Seçilen bu en iyi model kullanılarak hesaplanan öngörü değerleri gelecek tahminleri olarak verilir. En iyi modelin bulunması sürecinde gerçekleştirilen bu işlem *deneme yanılma yöntemi* olarak adlandırılır. Örneğin, diğer etkenler sabitken, sadece bir etken için de belirlenen seçim aralığını deneme yanılma yöntemiyle araştırmak mümkündür.

Daha önce de belirtildiği gibi, bu etkenler arasında en çok seçeneği sunan mimari yapıdır. Diğer etkenlerin sunduğu seçenek sayıları, mimari yapının sunduğu seçenek sayısına göre çok küçük olduğundan ve diğer etkenler için literatürde sabitleşmiş bazı seçimler zaten bulunduğu için, incelenmesi gereken en önemli YSA bileşeni mimari yapıdır. Bu nedenle, literatürde yapılan birçok çalışmada, diğer etkenler önceden belirlenmiş değerlerde sabit tutulurken, en iyi öngörülere ulaşılmasını sağlayacak mimari yapının belirlenmesinde, deneme yanılma yöntemi kullanılmaktadır.

## 4. GELİŞTİRİLEN TABU ARAMA ALGORİTMASI

Zaman serileri öngörü probleminde YSA kullanıldığında, seçilecek mimari elde edilecek öngörülerin iyiliğini doğrudan etkileyen önemli bir faktördür. Yapılan çalışmada, tek gizli tabaka içeren ileri beslemeli YSA modelleri kullanıldığında, öngörü için tercih edilecek mimaride, girdi tabakasında ve gizli tabakada kaç birim olacağının belirlenmesi problemi ele alınmıştır. Tanımlanan mimari seçimi problemini çözmek amacıyla geliştirilen tabu arama algoritması bu bölümde açıklanmıştır.

### 4.1. Başlangıç Seçimleri

Tüm sezgisel algoritmalarda olduğu gibi, tabu arama algoritmasında da başlangıç koşullarının belirlenmesi, algoritmanın performansını belirleyen en önemli etkenlerin başında gelir. Bu nedenle, tabu arama algoritması çalıştırılmadan önce, çeşitli başlangıç seçimlerinin akıllıca yapılması gerekmektedir. Geliştirilen tabu arama algoritması kullanılmadan önce, aramanın daha etkin bir şekilde gerçekleştirilebilmesi için düşünülen seçim stratejileri bu kısımda tanıtılmıştır.

#### 4.1.1. YSA modellerinde başlangıç ağırlıklarının belirlenmesi

YSA yöntemiyle zaman serilerinde çözümlene yapılıırken, en iyi öngörü değerlerini üreten ağırlık değerlerinin bulunmaya çalışıldığı belirtilmişti. Kullanılan öğrenme algoritmasıyla en iyi ağırlık değerleri hesaplanmaya çalışılırken, bir optimizasyon süreci işletilir. Bu optimizasyon sürecinde aramaya başlanılacak ilk noktanın yani, ağırlıkların başlangıç değerlerinin ne olacağı, güvenilir öngörülere ulaşılmasında önemli bir noktadır. Başlangıç ağırlıklarına belli bir kurala göre sabit değerler vermek mümkünken, uygulamada genellikle rastgele oluşturulan başlangıç ağırlıkları kullanılarak çözümlene yapılır. Rastgele başlangıç ağırlıkları kullanıldığında, aynı yapay sinir ağı mimarisiyle bile, birden fazla çözümlene yapıldığında elde edilen sonuçlar birbirinden çok farklı olabilir. Bunun nedeni, YSA yönteminde kullanılan öğrenme algoritmalarının yerel optimal tuzağına takılmasıdır.

Önerilen yöntemde, çözümlene yapılacak mimarinin en iyi ağırlıkları belirlenirken, yerel optimalden kurtulabilmek amacıyla basit ama etkili bir algoritma kullanımı düşünülmüştür. Ağın eğitimi, farklı rastgele başlangıç ağırlıkları için belli bir tekrar

sayısı kadar (tkr) gerçekleştirilir ve en iyi sonucu veren ağırlıklar kullanılır. İlgili algoritmanın Matlab kodu (Algoritma 1) aşağıda verilmiştir. Verilen algoritma ile 'tkr' sayısı kadar çözümlene yapilir ve her çözümlenmede aynı mimari için farklı rastgele ağırlıklarla çözümlenmeye başlanılır. Aynı girdi ve gizli tabaka birim sayısı için, (mi:girdi sayısı, mh:gizli birim sayısı) en iyi sonucu veren ağırlıklar bulunur. Daha sonra, mi ve mh değerlerinden oluşan mimari için, bulunan bu en iyi ağırlıklara (Enlyi\_Net) göre, test kümesi üzerinden HKOK (Enlyi\_HKOK) değeri ve test kümesi tahminleri (Enlyi\_Ongoruler) hesaplanır.

### Algoritma 1.

```
EnlyiHKOK=1000000000000;  
for ind=1:tkr  
    [HKOK,Ongoruler,net_t]=NNFF(Veri,mi,mh,ntest);  
    if HKOK < EnlyiHKOK  
        Enlyi_HKOK      =HKOK;  
        Enlyi_Ongoruler =Ongoruler;  
        Enlyi_Net       =net_t;  
    end  
end
```

Algoritma 1'de NNFF fonksiyonu, çözümlenecek zaman serisi (Veri), girdi birim sayısı (mi), gizli tabaka birim sayısı (mh) ve test kümesi büyüklüğü (ntest) değerleri girildiğinde, mi-mh-1 ileri beslemeli yapay sinir ağı mimarisiyle çözümlene yapan ve test kümesi üzerinden HKOK değerini, test kümesi tahminlerini (Ongoruler) ve bulunan en iyi ağırlıkları içeren sinir ağını (net\_t) çıktı olarak veren fonksiyonu göstermektedir. NNFF fonksiyonu her çağırıldığında, farklı rastgele başlangıç ağırlıkları ile işlemektedir.

Özetlemek gerekirse, belli bir mimari ile çözümlene yapilirken, belirli bir tekrar sayısı kadar aynı mimariyle fakat her seferinde farklı rastgele başlangıç ağırlıkları kullanılarak, test kümesi öngörülere elde edilir. Her tekrar için test kümesi üzerinden elde edilen HKOK değerleri içinde en küçük hatayı veren ağırlıkları içeren sinir ağı kullanılır.

#### 4.1.2. Öğrenme algoritması

Tabu arama algoritması ile en iyi mimari aranırken, denenecek YSA modellerinde kullanılacak öğrenme algoritmasının seçimi yine elde edilecek sonuçları etkileyecek önemli bir başlangıç tercihidir. Matlab 7.4 programında sinir ağları ile çözümlene yapılıırken, kullanılabilir 16 öğrenme algoritması mevcuttur. Bu algoritmaların program içindeki fonksiyon isimleri `trainlm`, `trainbfg`, `trainrp`, `traingd`, `trainr`, `trainc`, `trainb`, `traingdm`, `traingd`, `trainscg`, `trainoss`, `trainbfgc`, `traincgp`, `traincgf`, `traincgb` ve `trainbr` şeklinde belirtilmiştir. Kullanılabilir öğrenme algoritmaları hakkında ayrıntılı bilgi, Matlab programı içindeki yardım klasöründen elde edilebilir. Matlab paket programında Levenberg Marquardt Backpropagation (`trainlm`) algoritması, en hızlı öğrenme algoritması olduğundan varsayılan (default) öğrenme algoritmasıdır. Bu nedenle, kodlanan programda da öğrenme algoritması olarak 'trainlm' algoritması tercih edilmiştir.

#### 4.1.3. Başlangıç havuzu çözümlemesi

Kullanılacak tabu arama algoritmasının hangi başlangıç çözümünden başlayacağı ya da kullanılacak hafıza türlerinin başlangıç değerlerinin ne olacağı, arama sonucunda ulaşılabilecek çözümün ne derecede iyi olacağını belirleyecek başlangıç seçimlerinden bazılarıdır. Başlangıç çözümünün oluşturulmasında kullanılan yöntemlerden biri bu çözümün rastgele üretilmesidir ancak başlangıç çözümünün belirli bir strateji ile akıllıca seçilmesi, algoritmanın iyi sonuçlara daha çabuk ulaşmasına ya da çok daha iyi sonuçlar üretmesine yardımcı olur. Bu nedenle, başlangıç çözümünün oluşturulmasında ve kullanılan hafıza yapılarının başlangıç değerlerinin belirlenmesinde, *Başlangıç Havuzu* adı verilen bir strateji önerilmiş ve kullanılmıştır.

Önerilen başlangıç havuzu stratejisi, çözüm uzayında belli küçük bir bölgenin incelenmesine ve incelenen bu bölgeden elde edilen bilgi kullanılarak başlangıç koşullarının oluşturulmasına dayanmaktadır. Öncelikle, kullanıcının belirlediği bir aralıktaki çözümler incelenir. Örneğin, kullanıcı aralığı girdi birimleri için 1 ve 4, gizli birimleri için 1 ve 3 olarak belirlediğinde, başlangıç havuzunda incelenecek 12 mimari Çizelge 4.1'de gösterilmiştir. Çizelgede verilen tüm mimarilerin kullanılması sonucu, test kümesi üzerinden hatalar hesaplanarak, başlangıç havuzu incelenir. Kullanıcı tarafından belirlenen mimariler incelenirken, çözümlere ait sıklık bilgisi

kaydedilir. En sonunda, kaydedilen bu bilgi kullanılarak başlangıç çözümü oluşturulur ve kullanılacak hafıza türlerinin başlangıç değerleri belirlenir.

Çizelge 4.1: Örnek başlangıç havuzu çözümleri.

Çözüm No	Girdi birim sayısı	Gizli birim sayısı
1	1	1
2	2	1
3	3	1
4	4	1
5	1	2
6	2	2
7	3	2
8	4	2
9	1	3
10	2	3
11	3	3
12	4	3

Önerilen başlangıç havuzu stratejisinin nasıl işlediğini daha iyi açıklamak amacıyla bir örnek üzerinde stratejinin işleyişi gösterilmiştir. Herhangi bir zaman serisi için girdi birimlerinin 1 ve 6, gizli birim sayısının 1 ve 7 arasında değiştiği başlangıç havuzunun incelenmesi sonucunda, örnek olarak Çizelge 4.2'de verilen HKOK değerleri elde edilmiş olsun. Verilen çizelgede örneğin, 3. satır 2. sütundaki 25108,48, 3-2-1 mimarisiyle çözümlene yapıldığında test kümesi üzerinden hesaplanan HKOK değerini göstermektedir.

Çizelge 4.2: Başlangıç havuzundaki mimariler için elde edilen HKOK değerleri.

		Gizli birim sayısı						
		1	2	3	4	5	6	7
Girdi birim sayısı	1	16107,84	24964,93	26811,32	27226,45	27658,33	5648,86	228419,37
	2	16585,55	80218,73	24519,05	21338,38	13258,15	5499,92	12483,65
	3	16824,26	25108,48	25366,48	74307,22	25434,97	68023,79	17283,55
	4	17331,10	26812,58	76035,23	23624,26	26441,36	16624,67	12938,61
	5	17043,86	14710,91	25434,97	27243,95	23510,44	18477,80	27943,15
	6	17104,69	16177,90	27474,60	25434,97	25434,97	47740,97	38416,96

Girdi birim sayısı sabit tutulduğunda, gizli birim sayısı için her satırda en küçük değeri veren gizli birim sayısı önem teşkil eder. Örneğin, 4 girdi birimi

kullanıldığında en iyi sonuç 7 gizli birimi için elde edilmiştir. Öncelikle, Çizelge 4.2'de verilen hatalar matrisinde, satırlardaki en küçük değerler belirlenir. Çizelge 4.3'de her bir girdi birimi için en küçük hatalar koyu yazılmıştır.

Çizelge 4.3: Hatalar matrisinde satırlardaki en küçük değerler.

		Gizli birim sayısı						
		1	2	3	4	5	6	7
Girdi birim sayısı	1	16107,84	24964,93	26811,32	27226,45	27658,33	<b>5648,86</b>	228419,37
	2	16585,55	80218,73	24519,05	21338,38	13258,15	<b>5499,92</b>	12483,65
	3	<b>16824,26</b>	25108,48	25366,48	74307,22	25434,97	68023,79	17283,55
	4	17331,10	26812,58	76035,23	23624,26	26441,36	16624,67	<b>12938,61</b>
	5	17043,86	<b>14710,91</b>	25434,97	27243,95	23510,44	18477,80	27943,15
	6	17104,69	<b>16177,90</b>	27474,60	25434,97	25434,97	47740,97	38416,96

Satırlardaki en küçük değerler belirlendikten sonra, her satırda en küçük değer bulunduğu hücre değerine 1, aynı satırdaki diğer hücelere 0 değeri verilerek yeni bir matris oluşturulur. Buna göre, 0 ve 1 değerlerinden oluşan 6x7 boyutlu

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

matrisi elde edilir.

0 ve 1 değerlerinden oluşan bu matris, denenen her bir girdi birimi için en iyi sonucu veren gizli birim sayısını göstermektedir. Oluşturulan matrisin sütun değerleri toplamları alınarak

$$[ 1 \ 2 \ 0 \ 0 \ 0 \ 2 \ 1 ]$$

vektörü bulunur. Hesaplanan bu dizi, en iyi sonuç veren gizli birim sayılarının sıklık değerleri bilgisini taşımaktadır. Elde edilen bilgiyi geliştirilen tabu arama algoritmasında kullanabilmek için sonuçların seçim kurallarına yansıtılması gerekir. Oluşturulan sıklık bilgisi seçim kurallarına dönüştürüldüğünde, başlangıç çözümünün üretilmesinde kullanılabilir. Sadece başlangıç koşullarının belirlenmesinde değil, aynı zamanda algoritmanın diğer adımlarında da, örneğin



komşu çözümlerin belirlenmesinde ya da seçiminde, sıklık bilgisi seçim kurallarına yansıtılarak kullanılabilir. Bunun için, hesaplanan sıklıklar olasılıklara dönüştürülmelidir. Yukarıda verilen sıklıklar öncelikle, gizli birim sayılarını gösteren indis numaralarıyla büyükten küçüğe doğru sıralanır. Verilen örnek için sıklıklar indis numaralarıyla birlikte

İndis	1	2	3	4	5	6	7
Sıklık	1	2	0	0	0	2	1

şeklindedir. Sıklıklar büyükten küçüğe sıralandığında

İndis	2	6	1	7	3	4	5
Sıklık	2	2	1	1	0	0	0

şeklinde verilen diziler elde edilir.

Elde edilen bu değerlere, olasılık değerleri atamak için geliştirilen algoritmanın Matlab kodu (Algoritma 2) aşağıda verilmiştir. Verilen algortmada, Dizi, yukarıdaki sıklık dizisini, OlasilikD, olasılıkların yaratıldığı diziyi, KOD, kümülatif toplamı olasılık dizisini ve ss, dizinin eleman sayısını göstermektedir.

### Algoritma 2.

```

Dtop=0;
for j=1:ss
    if Dizi(j)>0
        OlasilikD(j)=5*Dizi(j);
    end
    if Dizi(j)==0
        OlasilikD(j)=1;
    end
    Dtop=Dtop+OlasilikD(j);
end

for j=1:ss
    OlasilikD(j)=(OlasilikD(j)/Dtop)*100;
end

KOD(1)=OlasilikD(1);
for j=2:ss
    KOD(j)=KOD(j-1)+OlasilikD(j);
end

```

Algoritma 2 kullanıldığında, KOD dizisi ve karşılık gelen gizli birim sayıları

İndis	2	6	1	7	3	4	5
KOD	30,30	60,61	75,76	90,91	93,94	96,97	100,00

şeklinde hesaplanır. Elde edilen KOD dizisi kullanılarak, tabu arama algoritmasındaki seçim kurallarını değiştirmek artık mümkündür. Öncelikle, 1 ile 100 arasında rastgele bir sayı üretilir. Üretilen sayının, KOD dizisindeki daha küçük olduğu elemanlardan en küçüğüne karşılık gelen indis, seçilecek gizli birim sayısını gösterir. Buna göre, örneğin gizli birim sayısının 2 seçilmesi olasılığı %30,30 veya 7 seçilmesi %15,15 ( $90,91-75,76=15,15$ ) dir.

Benzer işlemler girdi birim sayılarına göre tekrarlanarak, girdi birim sayıları için de seçim kuralları oluşturmak mümkündür. Başlangıç havuzu çözümlemesinden elde edilen bilgiye göre oluşturulan seçim kurallarıyla, girdi ve gizli tabaka birim sayıları belirlenerek, tabu arama algoritmasının başlayacağı başlangıç çözümü yaratılabilir. Buna karşın, eğer başlangıç mimarisinin hem girdi birim hem de gizli birim sayısı bu şekilde belirlenirse, yine başlangıç havuzu içinde incelenmiş çözümlerden birinden başlanılmış olacaktır. Tabu arama algoritması eğer böyle bir başlangıç çözümünden başlatılırsa, başlangıç havuzunda incelenen bölgede tekrar arama yapılmış olacaktır. Zaten incelenmiş bölgenin yeniden incelenmesi ise sadece zaman kaybına neden olur. Zaman kaybını önlemek için, başlangıç çözümü oluşturulurken, sadece gizli birim sayısı açıklandığı şekilde sıklık bilgisinden elde edilen olasılıklara göre seçilir. Girdi birim sayısı ise kullanıcının girdi birimleri için belirlediği aralığın üst sınırının bir fazlası olarak alınır. Böylece, tabu arama algoritmasının başlangıç havuzunda zaten incelenen bölgede tekrar arama yapması engellenerek gereksiz zaman kaybı önlenmiş olur.

Başlangıç havuzunda, girdi ve gizli tabakalarda kullanılan birim sayılarından elde edilen bilgiye göre, tabu arama algoritmasında kullanılacak frekans temelli hafızanın ve tabu listesinin başlangıç durumları da belirlenir. Başlangıç havuzunda incelenen mimariler tabu listesine aktarılır ve elde edilen sıklık bilgisi de frekans temelli hafızaya kaydedilir. Böylece, kullanılacak bu iki hafıza yapısının başlangıç değerleri yaratılmış olur.

## 4.2. Geliştirilen Tabu Arama Algoritmasının Elemanları

Bu bölümde, geliştirilen algoritmada kullanılan tabu arama elemanlarının açıklaması verilmiştir.

### A) $x$ çözümü

Girdi tabakasındaki birim sayısı ( $x_1$ ) ve gizli tabakadaki birim sayısından ( $x_2$ ) oluşan  $[x_1 \ x_2]$  vektörü her bir  $x$  çözümünü göstermektedir. Birçok uygulamada olduğu gibi, kullanılan yapay sinir ağı mimarilerinde çıktı birim sayısı hep 1 alındığından, çözüm vektöründe gösterilmesine gerek yoktur. Buna göre, her bir  $x$  çözümü bir yapay sinir ağı mimarisini temsil etmektedir.

### B) $X$ çözüm uzayı

$MiU$  ve  $MhU$  sırasıyla, incelenecek girdi birim sayısı ve gizli tabaka birim sayısı için üst sınırları temsil etsin. Buna göre,  $MiU$ ,  $MhU$  ve  $tkr$  sayılarının çarpımı kadar ( $MiU \times MhU \times tkr$ ) mimariden oluşan bir çözüm uzayında arama yapılacaktır. Kullanıcı tarafından belirlenecek bu parametreler geliştirilen tabu arama algoritmasına yön veren parametreler arasındadır.

### C) $f(x)$ amaç fonksiyonu

Her bir  $x$  mimarisinin değerlendirilmesinde

$$f(x) = HKOK$$

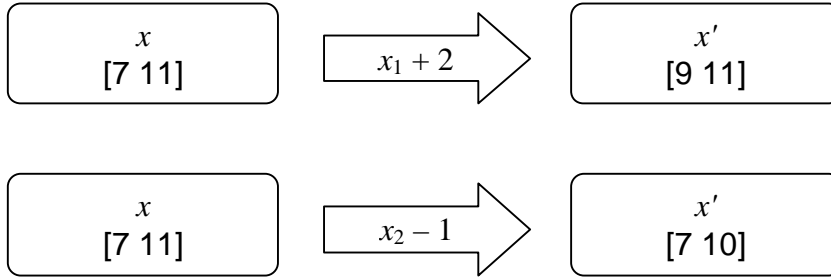
şeklinde verilen amaç fonksiyonu kullanılmıştır.  $T$ , test kümesinin büyüklüğünü,  $y_i$ ,  $i$  zamanı için gözlem değerini ve  $\hat{y}_i$ ,  $i$  zamanı için elde edilen tahmin değerini göstermek üzere,  $HKOK$  değeri

$$HKOK = \left( \frac{\sum_{i=1}^T (y_i - \hat{y}_i)^2}{T} \right)^{1/2}$$

şeklinde hesaplanabilir. Buna göre, geliştirilen tabu arama algoritması, test kümesi üzerinden hatayı minimum yapan, yani en güvenilir öngörü değerlerini üreten yapay sinir ağı mimarisini bulmaya çalışacaktır.

#### D) $N(x)$ komşuluğu ve hareket türleri

Ekleme ve çıkarma hareketleri kullanılarak bir  $x$  çözümünün  $x'$  komşusuna geçmek mümkündür. Örneğin, Şekil 4.1'de gösterildiği gibi,  $x=[7 \ 11]$  çözümünde girdi birim sayısı 2 artırılarak  $(x_1+2)$ ,  $x'=[9 \ 11]$  komşu çözümü elde edilir ya da çıkartma hareketi kullanımıyla, gizli tabakadaki birim sayısı 1 azaltılarak  $(x_2-1)$ ,  $x'=[7 \ 10]$  komşu çözümü elde edilebilir. Geliştirilen algorithmada, bir  $x$  çözümünün  $N(x)$  komşuluğu oluşturulmasında, ekleme ve çıkarma hareketleri kullanılmıştır.



Şekil 4.1. Hareket türleri örnekleri.

#### E) Aday liste stratejisi

Herhangi bir  $x$  çözümü için oluşturulabilecek tüm  $N(x)$  komşuluğunu incelemek, hesaplama maliyeti olarak çok fazla ve anlamsızdır. Bunun yerine, bir aday liste stratejisi kullanmak akıllı bir tercih olacaktır. '+', 1 birim artırmayı, '-' 1 birim azaltmayı ve '0' değişiklik yapılmamasını göstermek üzere,  $x=[x_1 \ x_2]$  çözümünden 6 komşu çözüm Çizelge 4.4'de verildiği gibi elde edilir.

Çizelge 4.4:  $x=[x_1 \ x_2]$  çözümünden elde edilen komşu çözümler.

	$x_1$	$x_2$	Komşu çözüm
1	+	0	$[x_1+1 \ x_2]$
2	0	+	$[x_1 \ x_2+1]$
3	+	+	$[x_1+1 \ x_2+1]$
4	-	0	$[x_1-1 \ x_2]$
5	0	-	$[x_1 \ x_2-1]$
6	-	-	$[x_1-1 \ x_2-1]$

Örnek olarak,  $x=[4 \ 11]$  çözümünün komşuları oluşturulduğunda, Çizelge 4.5'de verilen komşu çözümler elde edilecektir.

Çizelge 4.5:  $x=[4 \ 11]$  çözümünden elde edilen komşu çözümler.

	$x_1$	$x_2$
1	5	11
2	4	12
3	5	12
4	3	11
5	4	10
6	3	10

Tanımlanan bu aday liste stratejisi kullanılarak, tüm  $N(x)$  komşuluğunu incelemek yerine, mevcut çözümün 6 komşusu incelenecektir.

Elde edilen komşu  $x'$  çözümünün komşuluğunda da,  $x$  çözümünün bir önceki adımda incelenen komşuları bulunacaktır. Örneğin,  $x=[4 \ 11]$  komşularından  $x'=[4 \ 12]$  bir sonraki adım için mevcut çözüm olarak seçilirse, komşuları Çizelge 4.6'da verildiği gibi bulunur. Dikkat edilirse, "\*" işaretli çözümler, Çizelge 4.5'de gösterilen bir önceki mevcut çözümün de komşuları arasındadır.

Çizelge 4.6:  $x'=[4 \ 12]$  çözümünün komşu çözümleri.

	$x_1$	$x_2$
1*	5	12
2	4	13
3	5	13
4	3	12
5	4	11
6*	3	11

Yeni mevcut çözümün bazı komşularının bir önceki mevcut çözümün de incelenmiş komşularıyla aynı olması durumunda, bu çözümler tekrar incelenecektir. Böylece, ilgilenilen bölgede bir yoğunlaşma etkisi yaratılması amaçlanmıştır.

#### F) Tabu listesi

Tabu listesi için açık (explicit) hafıza kullanılmıştır. Tabu listesinde çözümlerin girdi ve gizli tabakalarındaki birim sayıları yani  $[x_1 \ x_2]$  vektörleri saklanmıştır. Böylece, aspirasyon kriteri sağlanmadığı sürece, önceden incelenmiş bir çözümün yeniden mevcut çözüm olarak seçilmesi, kullanıcı tarafından belirlenen tabu süresi boyunca engellenir.

### G) Aspirasyon kriteri

Amaca göre tümel aspirasyon kriteri uygulanmıştır. Kullanılan aspirasyon kriterine göre, bir çözüm tabu olmasına karşın, o ana dek bulunan en iyi amaç değerinden daha iyi bir amaç değerine sahipse, tabu durumu iptal edilir.

### H) Hareketlerin seçimi

Mevcut çözümden üretilen komşu çözümler arasında seçim yapılırken, en yüksek değerlendirme kriteri kullanılmıştır. Komşular arasında en iyi amaç değerine sahip, tabu olmayan komşu bir sonraki adımda mevcut çözüm olarak seçilir. Eğer en iyi amaç değerine sahip çözüm tabuysa ve aspirasyon kriterini sağlıyorsa, tabu durumuna rağmen yeni mevcut çözüm olarak seçilir. Diğer yandan, aspirasyon kriterini sağlamıyorsa, bir sonraki en iyi amaç değerine sahip komşu, aynı şekilde incelenerek devam edilir.

### I) Frekans (sıklık) temelli hafıza

Girdi ve gizli tabaka birim sayılarının, mevcut çözümden görünme sayıları ayrı ayrı frekans temelli hafızada saklanır. Frekans temelli hafızada tutulan ve her bir iterasyonda güncellenen bilginin çeşitleme stratejisinde kullanılmasıyla, çözüm uzayının keşfedilmemiş bölgelerine aramanın yönlendirilmesi sağlanarak, çeşitleme etkisi yaratılmaktadır. Böylece, aramanın belli bölgelere sıkışıp kalması ve yerel optimal tuzaklara yakalanması engellenmiş olur.

Örneğin, tabu arama algoritmasının herhangi bir iterasyonunda girdi değerleri için frekans değerleri

Girdi sayısı	1	2	3	4	5	6	7	8	9	10
Sıklıklar	0	10	6	2	2	3	2	5	7	5

şeklinde olsun. Öncelikle, verilen değerler

Girdi sayısı	1	4	5	7	6	8	10	3	9	2
Sıklıklar	0	2	2	2	3	5	5	6	7	10

şeklinde küçükten büyüğe sıralanır. Daha sonra, elde edilen bu değerlere olasılık değerleri atamak için oluşturulan algoritma aşağıda verilmiştir. Verilen algoritmada,

Dizi, yukarıdaki sıklık dizisini, OlasilikD, olasılıkların yaratıldığı diziyi, KOD, kümülatif toplamı olasılık dizisini ve ss, dizinin eleman sayısını göstermektedir.

### Algoritma 3.

Dtop=0;

```

for j=1:ss
  if Dizi(j)>0
    OlasilikD(j)=3.5*Dizi(j);
  end
  if Dizi(j)==0
    OlasilikD(j)=1;
  end
  Dtop=Dtop+OlasilikD(j);
end

```

```

for j=1:ss
  OlasilikD(j)=(1-(OlasilikD(j)/Dtop))*100;
end

```

```

for j=1:ss
  OlasilikD(j)=OlasilikD(j)/(100-OlasilikD(j));
end

```

KOD(1)=OlasilikD(1);

```

for j=2:ss
  KOD(j)=KOD(j-1)+OlasilikD(j);
end

```

```

for j=1:ss
  KOD(j)=(KOD(j)/KOD(ss))*100;
end

```

Algoritma 3 kullanıldığında, KOD dizisi ve karşılık gelen girdi birim sayıları

Girdi sayısı	1	4	5	7	6	8	10	3	9	2
KOD	58,86	66,92	74,99	83,05	88,30	91,28	94,27	96,69	98,71	100,00

şeklinde hesaplanır. Elde edilen bu KOD dizisi kullanılarak, yeni seçim kuralları oluşturmak mümkündür. 1 ile 100 arasında rastgele bir sayı üretilir ve sayının KOD dizisinde, daha küçük olduğu elemanlardan en küçüğüne karşılık gelen indis, seçilecek girdi birim sayısını gösterir. Buna göre, örneğin algoritmada o ana dek hiç kullanılmamış, 1 girdi birim sayısının seçilmesi olasılığı %58,86 veya 2 kez kullanılmış olan 7 birimin seçilmesi olasılığı %8,06 ( $83,05 - 74,99 = 8,06$ ) dır.

Benzer işlemler gizli birim sayılarına göre tekrarlanarak, gizli birim sayıları için de seçim kuralları oluşturulur. Bu şekilde, frekans temelli hafıza kullanılarak oluşturulan seçim kurallarıyla, girdi ve gizli tabaka birim sayılarını seçmek mümkündür. Herhangi bir birim sayısı, seçilen mevcut çözümlerde ne kadar çok görülmüşse, seçilme şansı o kadar azdır ya da bir başka ifadeyle, seçilen mevcut çözümlerde en az kullanılan birim sayısının seçilme şansı en fazladır.

#### **K) Çeşitleme stratejisi**

Çeşitleme stratejisi, iki farklı durumda frekans temelli hafızanın kullanılmasıyla uygulanmıştır. Çeşitleme etkisinin yaratıldığı durumlar aşağıda verilmiştir.

*i)* Herhangi bir mevcut çözümün komşularından hiçbiri yeni mevcut çözüm olarak seçilemediğinde, frekans temelli hafıza kullanılarak yeni mevcut çözüm belirlenir. Yüksek sıklığa sahip nitelikler cezalandırılarak, az kullanılan birim sayılarının seçim şansları artırılır. Böylece, yeni oluşturulacak mevcut çözümün o ana dek incelenmemiş çözümler arasından seçilmesi sağlanarak, arama çözüm uzayında henüz keşfedilmemiş yeni bir bölgeye yönlendirilir.

*ii)* Belli bir iterasyon boyunca tümel amaç fonksiyonu değerinde iyileşme sağlanamadığında, bir tür yeniden başlama stratejisi ile yeni bir mevcut çözüme geçilir. Buradaki iterasyon sınırı çeşitleme parametresi olarak adlandırılmıştır. Yeni mevcut çözüm belirlenirken, bir önceki durumda anlatıldığı şekilde frekans temelli hafıza kullanılır.

#### **L) Yoğunlaşma stratejisi**

Düşünülen yoğunlaşma stratejisi için kritik olay hafızadan yararlanılmıştır. Tümel amaç fonksiyonu değerinde iyileşme sağlayan mevcut çözümler kritik olaylar olarak düşünülmüş ve bu çözümler kritik olay hafızasında saklanmıştır. Algoritma iterasyonları tamamlandıktan sonra, kritik olay hafızada en iyi amaç değerine sahip ilk üç çözüm tekrar çözümlenmiştir. Bu çözümlerin belirlenmiş ağırlık değerlerinin etrafında bir tür stratejik salınım uygulanarak, daha düşük hataya götüren ağırlık değerleri araştırılmıştır.



Yoğunlaşılacak çözümün ağırlık değerleri  $w_i$  olsun.  $w_i$  ağırlıklarının  $[(w_i - \delta) (w_i + \delta)]$  aralığındaki komşulukları rastgele incelenerek, daha iyi amaç değerine götüreceği yeni değerleri araştırılır. Burada,  $\delta$  değeri ilgili birimden çıkan tüm ağırlıkların en küçük ve en büyük değerlerinden oluşan aralıktan rastgele seçilen bir uzaklığı göstermektedir. Yeni ağırlıkların yaratılmasında

**function** [IW,HW] = **Yogunlasma\_AgirlikYarat**(Winput,Whidden,mi,mh)

şeklinde verilen fonksiyon kodlanarak kullanılmıştır. Burada, IW ve HW sırasıyla, yeni elde edilen girdi ve gizli tabaka ağırlıklarını tutan matrisleri, Winput ve Whidden sırasıyla, girdi ve gizli tabaka ağırlıklarının ilk değerlerini tutan matrisleri, mi ve mh sırasıyla incelenen çözümün girdi birim ve gizli tabaka birim sayılarını göstermektedir.

“Yogunlasma\_AgirlikYarat” fonksiyonu her çağırıldığında, girdi ve gizli tabakalardaki ağırlık değerlerinin en büyük ve en küçüğü arasındaki farka dayalı olarak, rastgele üretilen bir  $\delta$  değerine göre yeni ağırlık değerleri oluşturulur. Böylece, elit çözümlerin ağırlıklarının etrafında yeni bir arama yapılarak, aynı mimari için daha iyi ağırlık değerleri araştırılmış olur. Kaç farklı  $\delta$  değeri kullanılacağı, yani en iyi üç mimari için kaç tane komşu çözümün inceleneceği, yine kullanıcı tarafından belirlenecek bir algoritma parametresidir. Bu parametre yoğunlaşma parametresi olarak adlandırılabilir.

#### **M) Durdurma koşulları**

Tabu arama algoritması kullanıcının belirlediği toplam iterasyon sınırına ulaşıldığında durdurulur.

### **4.3. Geliştirilen Tabu Arama Algoritmasının İşleyişi**

Buraya kadar verilen tüm tanımlar doğrultusunda, zaman serilerinde öngörü elde edilirken en iyi mimariyi belirlemek amacıyla geliştirilen tabu arama algoritmasının işleyişi aşağıda verilmiştir.

#### **A) Başlangıç havuzunu incele.**

**B)** Başlangıç havuzunda bulunan en iyi çözüm değerlerini (BH\_HKOK, BH\_Net), tümel en iyi çözüm değerleri (Enlyi\_HKOK, Enlyi\_Net) olarak belirle.

$$\text{Enlyi\_HKOK} = \text{BH\_HKOK};$$

$$\text{Enlyi\_Net} = \text{BH\_Net};$$

**C)** Kullanılacak hafızaların başlangıç durumlarını oluştur.

i) Başlangıç havuzunda oluşan frekans bilgisini frekans temelli hafızaya aktar.

ii) Başlangıç havuzunda denenen çözümleri tabu listesine aktar.

**D)** Tabu arama algoritmasını işlet.

**Adım 1.** Birinci iterasyon için mevcut çözüm olarak, başlangıç havuzundan elde edilen frekans bilgisine göre seçilen çözümü al.

**Adım 2.** Komşuluk üretme işlemini kullanarak mevcut çözümün komşularını üret.

**Adım 3.** Üretilen komşular içinden bir sonraki mevcut çözümü seç.

i) Mevcut komşu tabu değilse, bir sonraki mevcut çözüm olarak ata ve Adım 4'e geç. Eğer tabuysa, Adım 3 ii)'ye geç.

ii) Mevcut komşu tabu olmasına rağmen aspirasyon kriterini sağlıyorsa, mevcut komşuyu bir sonraki mevcut çözüm olarak ata ve Adım 4'e geç. Aspirasyon kriteri sağlanmıyorsa, Adım 3 iii)'ye geç.

iii) Üretilen komşular içinde en iyi amaç değerine sahip bir sonraki komşuyu, mevcut komşu olarak belirle ve Adım 3 i)'ye geri dön. Eğer tüm komşuların incelenmesi bittiyse Adım 4'e geç.

**Adım 4.** Bir önceki adımda incelenen komşulardan hiçbiri bir sonraki mevcut çözüm olarak seçilemediyse, frekans temelli hafızayı devreye sok ve bir sonraki mevcut çözümü üretip Adım 5'e geç. Diğer durumda, doğrudan Adım 5'e geç.

**Adım 5.** Seçilen mevcut çözüme göre hafızaları güncelle.

i) Mevcut çözümlü tabu listesine ekle.

ii) Mevcut çözümlde kullanılan birim sayılarını frekans temelli hafızaya ekle.

**Adım 6.** Seçilen mevcut çözümlün amaç değeri o ana kadar bulunan en iyi amaç değeriinden daha iyiyse, yeni en iyi çözüml olarak belirle, çeşitleme sayaç değeriini sıfırla, mevcut çözümlü kritik olay hafızaya ekle ve Adım 7'ye geç.

EnIyi\_HKOK = MevcutCozum\_HKOK;

EnIyi\_Net = MevcutCozum\_Net;

CesitlemeSayac = 0;

Diğer durumda, çeşitleme sayaç değeriini bir artır ve Adım 7'ye geç.

**Adım 7.** Çeşitleme sayaç değeri belirlenen çeşitleme parametresinden küçükse, Adım 8 e geç. Diğer durumda, frekans temelli hafızayı devreye sokarak bir sonraki mevcut çözümlü üret ve Adım 8'e geç.

**Adım 8.** Belirlenen iterasyon sayısına ulaşıldıysa, Adım 9'a geç. Diğer durumda, Adım 2'ye geri dön.

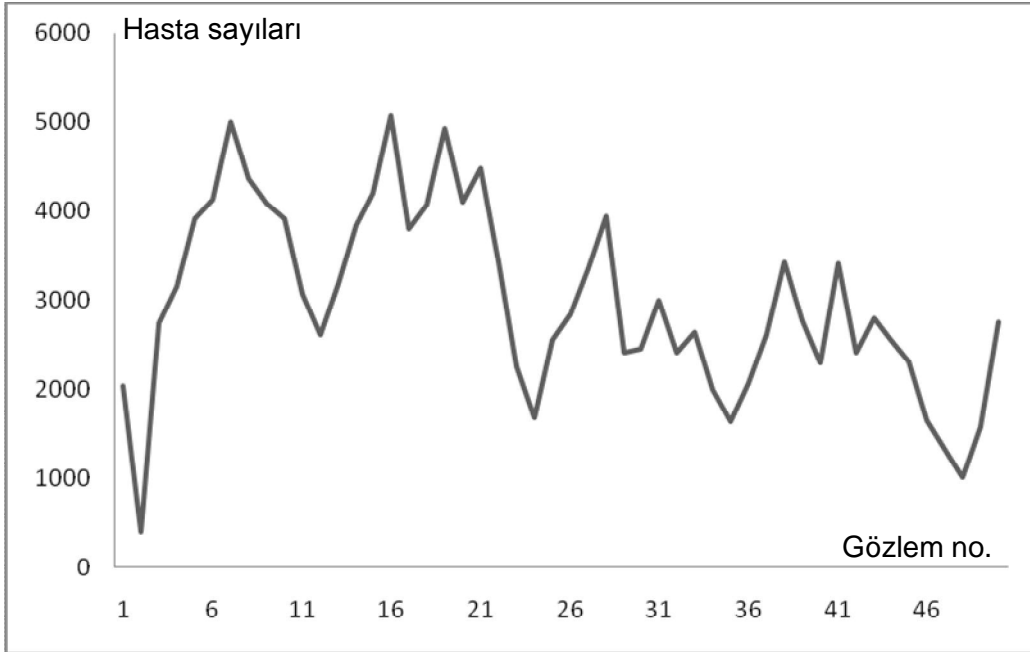
**Adım 9.** Kritik olay hafızadaki en iyi amaç değeriine sahip ilk üç çözüml için yoğunlaşma stratejisini belirlenen yoğunlaşma parametresine göre uygula. Yoğunlaşma stratejisinden sonra en iyi amaç değeriine sahip mimariyi seç.

## 5. UYGULAMA

Zaman serilerinde öngörü elde edilirken, en iyi yapay sinir ağının belirlenmesi probleminin çözümü için geliştirilen tabu arama algoritmasının etkinliğini görebilmek amacıyla, önerilen algoritma çeşitli gerçek zaman serilerine uygulanmıştır. Uygulamada kullanılan zaman serileri, literatürde en çok kullanılan deneme yanılma yöntemiyle de analiz edilmiştir. Analizlerde kullanılan iki yöntemden elde edilen öngörü sonuçları karşılaştırılarak, önerilen tabu arama algoritmasının daha güvenilir öngörüler üreten mimariler seçtiği gösterilmiştir.

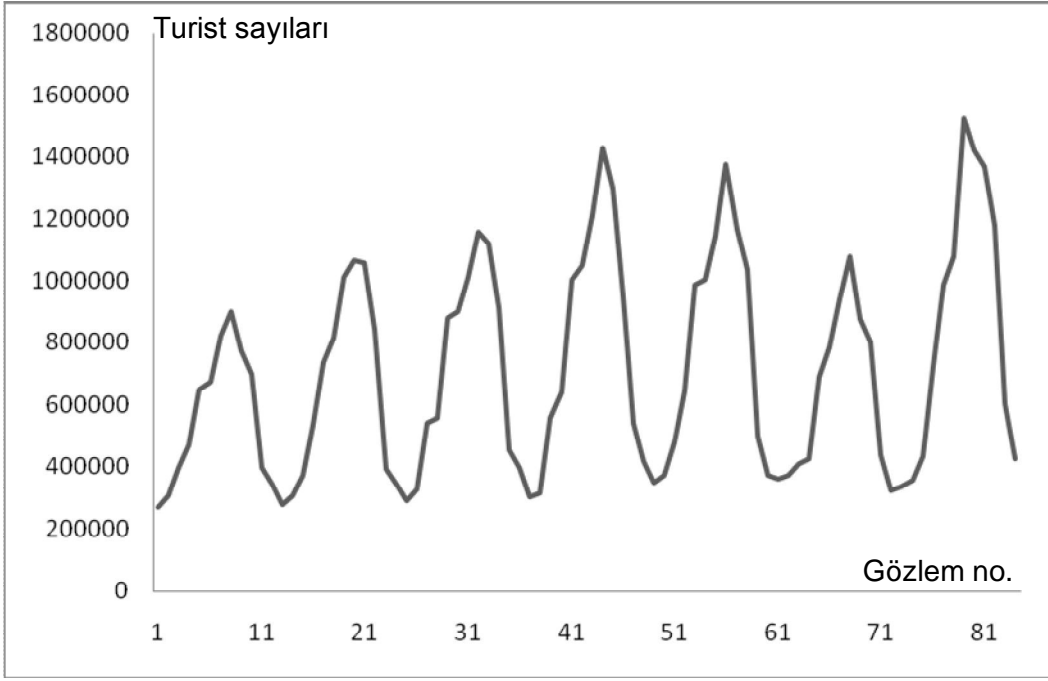
### 5.1. Uygulamada Kullanılan Zaman Serileri

Uygulamada 5 farklı zaman serisi kullanılmıştır. Kullanılan zaman serilerinden birincisi, 2004 Eylül ayı ile 2008 Ekim ayı arasında Hacettepe Hastanesi'ne gelen hasta sayılarıdır. Aylık gözlemlerden oluşan zaman serisi 50 gözlem içermektedir ve grafiği Şekil 5.1'de verilmiştir.



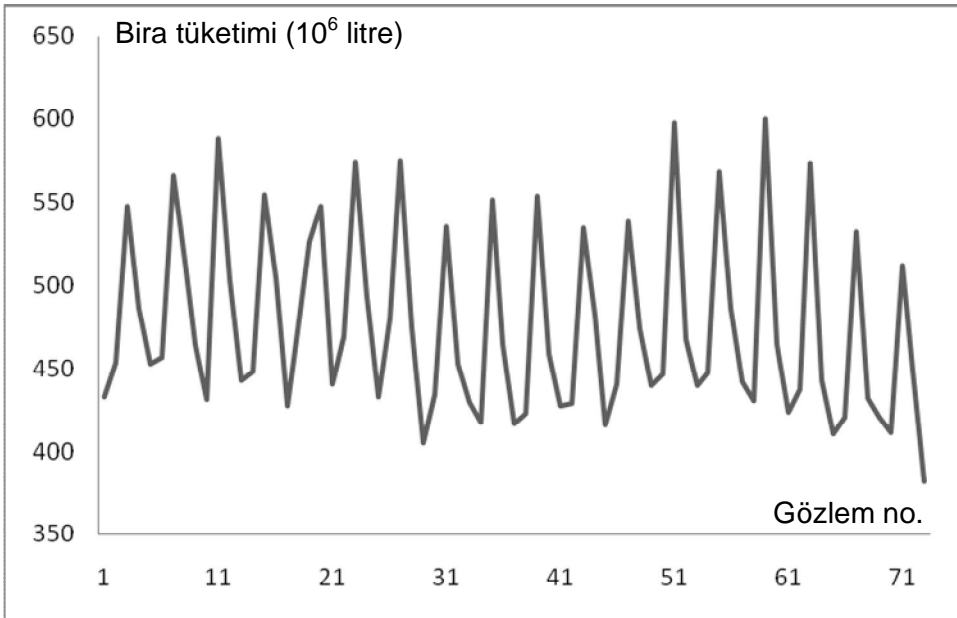
Şekil 5.1. Hasta sayıları zaman serisi.

Kullanılan bir diğer zaman serisi, 1995 yılı Ocak ayı ile 2001 yılı Aralık ayı arasında Türkiye'ye gelen turist sayılarıdır. Aylık olarak gözlemlenen turist sayılarından oluşan zaman serisi 84 gözlem içermektedir ve grafiği Şekil 5.2'de verilmiştir.



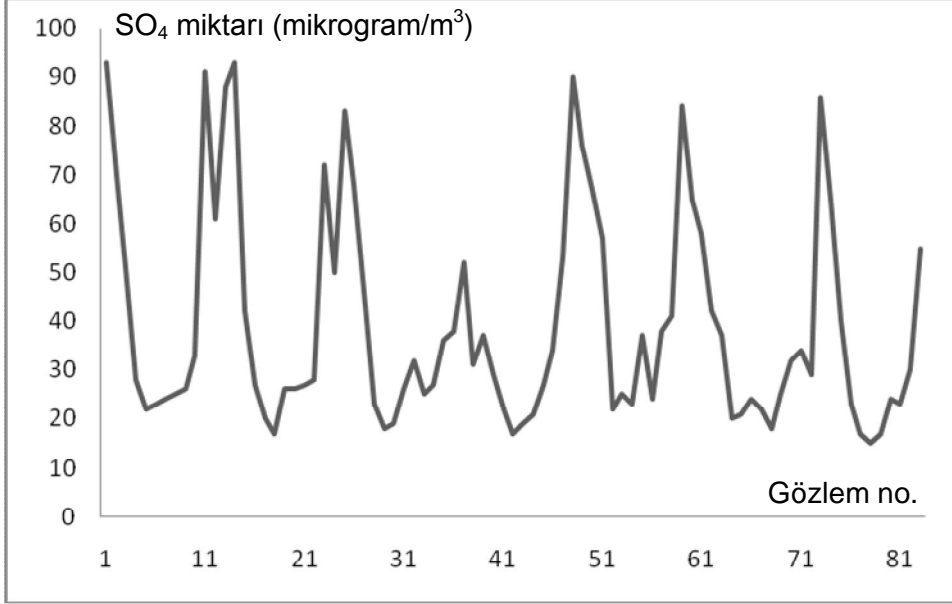
Şekil 5.2. Türkiye'ye gelen turist sayıları zaman serisi.

Kullanılan üçüncü zaman serisi çeyreklik gözlemlenmiş, literatürde zaman serileri uygulamalarında sıkça kullanılan Avusturya bira tüketimi serisidir. 1976 yılının birinci çeyreği ile 1994 yılının birinci çeyreği arasında gözlenen 73 değerden oluşan zaman serisinin grafiği Şekil 5.3'de verilmiştir.

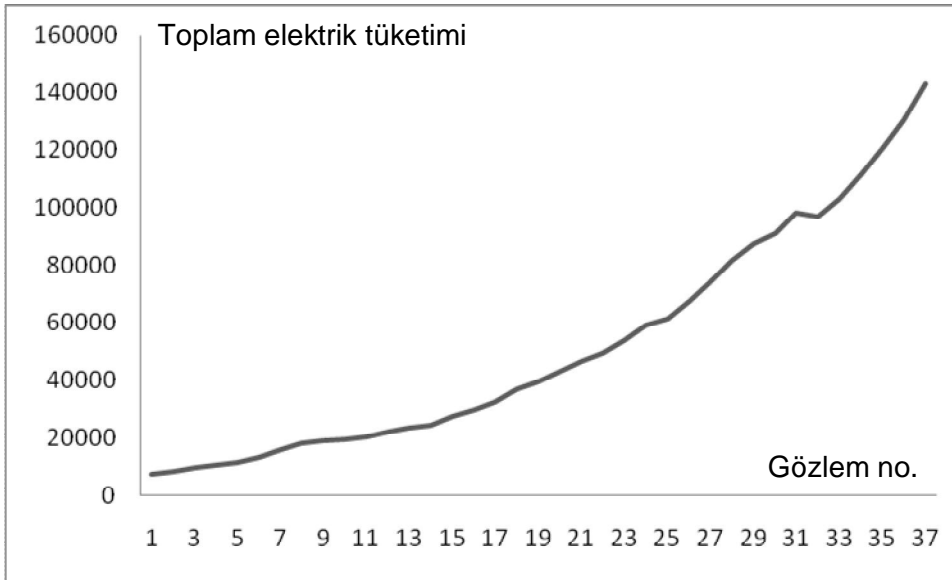


Şekil 5.3. Avusturya bira tüketimi zaman serisi.

Hava kirliliği derecesinin belirlenmesinde havadaki kükürt dioksit ( $SO_4$ ) miktarı ölçülmektedir. Ankara şehri için yapılan metreküpteki kükürt dioksit miktarı (mikrogram) ölçümleri, uygulamada kullanılan dördüncü zaman serisidir. Aylık kaydedilen ölçümler, 1996 yılının Ocak ayı ile 2002 yılının Kasım ayı arasındaki 83 gözlemden oluşmaktadır. Ankara şehri havadaki kükürt dioksit miktarı zaman serisinin grafiği Şekil 5.4'de görülmektedir.



Şekil 5.4. Kükürt dioksit miktarı zaman serisi.



Şekil 5.5. Türkiye toplam elektrik tüketimi zaman serisi.

1970 ile 2006 yılları arasındaki yıllık gözlemlerden oluşan, Türkiye toplam elektrik tüketimi ( $10^6 kWh$ ) zaman serisi uygulamada kullanılan beşinci zaman serisidir. 37 gözlemden oluşan serinin grafiği Şekil 5.5’de verilmiştir.

## 5.2. Çözümlemede Kullanılan Mimari Seçim Yöntemleri

Bir önceki bölümde belirtilen zaman serileri, tek gizli tabakanın kullanıldığı, çıktı tabaksında tek birimin bulunduğu ileri beslemeli YSA modelleri ile analiz edilmiştir. Tüm YSA modellerindeki birimlerde, tek kutuplu lojistik aktivasyon fonksiyonu tercih edilmiş ve öğrenme algoritması olarak Levenberg-Marquardt Geri Yayılım algoritması kullanılmıştır. Öğrenme algoritması olarak Levenberg-Marquardt algoritmasının seçilme nedeni Matlab programında en hızlı yakınsayan öğrenme algoritması olmasıdır.

En güvenilir öngörüler üretecek, en iyi mimarinin belirlenmesi probleminde, literatürde en yaygın kullanıma sahip deneme yanılma yöntemi ve önerilen tabu arama algoritmasına dayalı yöntem kullanılmıştır. Kullanılan yöntemler aşağıda özetlenmiştir.

### i) Deneme yanılma yöntemi

YSA yöntemiyle zaman serilerinde öngörü elde edilirken, mimari seçiminde en çok tercih edilen yöntem deneme yanılma yöntemidir. Önce, denenecek mimariler için bir aralık belirlenir. Daha sonra, belirlenen bu aralıktaki tüm mimariler teker teker denenerek, test kümesi üzerinde en iyi performans ölçütü değerine sahip mimari seçilir. Uygulamada girdi ve gizli tabakalarındaki birimler 1 ile 18 arasında değiştirilerek toplam 324 mimari incelenmiştir.

### ii) Geliştirilen tabu arama algoritması

Geliştirilen tabu arama algoritması Bölüm 4’de ayrıntılı biçimde açıklanmıştır. Mimari seçiminde, önerilen tabu arama algoritması kullanılırken, algoritma parametreleri aşağıdaki gibi alınmıştır.

Maksimum girdi tabakası birimi sayısı .....: 18

Maksimum gizli tabaka birim sayısı .....: 18

Başlangıç havuzu girdi tabakası birim sayısı .....: 3

Başlangıç havuzu gizli tabaka birim sayısı .....: 3

Her bir çözüm için tekrar sayısı .....: 8  
Maksimum iterasyon sayısı .....: 50  
Tabu süresi.....: 20  
Yoğunlaşma parametresi.....: 40  
Çeşitleme iterasyon parametresi .....: 8

Belirlenen bu parametrelere göre, girdi birim sayısının ve gizli tabakadaki birim sayısının 1 ile 18 arasında değiştiği toplam 324 farklı mimariyi içeren bir aralık üzerinde arama yapılacaktır. Her bir çözüm için tekrar sayısı 8 alındığından, çözüm uzayı büyüklüğü  $324 \times 8 = 2592$  olacaktır. Başlangıç havuzunda incelenecek mimariler için girdi birim sayısı ve gizli tabaka birim sayısı 1 ile 3 arasında değişeceğinden, toplam 9 çözüm incelenecektir.

Maksimum iterasyon sayısı 50 olarak belirlendiğinden, arama 50 iterasyon sonra sonlandırılacaktır. İncelenen mimarilerin tabu listesinde kalma süreleri 20 iterasyondur, yani bir çözümün tabu durumu 20 iterasyon sürmektedir. Yoğunlaşma parametresi 40 olarak alındığından, algoritma 50 iterasyon sonunda sonlandırıldıktan sonra, bulunan en iyi üç mimarinin ağırlıkları için 40 farklı kombinasyon daha denenerek, daha iyi ağırlık değerlerine ulaşılmaya çalışılır. Çeşitleme iterasyon parametresi 8 olarak seçilmiştir. Buna göre, algoritma ardı ardına 8 iterasyon boyunca tümel olarak bulunan en iyi çözümden daha iyi bir çözüme ulaşamamışsa, frekans temelli hafıza kullanılarak yeni bir başlangıç çözümünden arama yeniden başlatılır.

### **5.3. Uygulamada Elde Edilen Sonuçlar**

Uygulamada kullanılan performans ölçütü test kümesi üzerinden hesaplanan HKOK değerleridir. Çözömlenen tüm zaman serilerinde, son 5 gözlem test kümesi için kullanılırken, geri kalan gözlemler eğitim kümesi için kullanılmıştır. Hem deneme yanılma hem de önerilen tabu arama algoritmasına dayalı mimari seçim yöntemleri için Matlab programında kodlanan fonksiyonlar, her bir zaman serisi için 50 ve 100 kez çalıştırılmıştır. Belirtilen iki yöntemin kullanılmasıyla, test kümeleri üzerinden elde edilen HKOK değerlerinin ortalama değerleri Çizelge 5.1'de verilmiştir. DY, deneme yanılma ve ÖTA önerilen tabu arama algoritmasını göstermektedir.



Çizelge 5.1: Test kümeleri üzerinden elde edilen ortalama HKOK değerleri.

Zaman serisi	Çalışma sayısı	DY	ÖTA	İyileşme (%)
Hasta sayıları	50	385,66	183,92	52,31
	100	364,69	188,15	48,41
Turist sayıları	50	49593,68	31067,23	37,36
	100	48894,23	30538,23	37,54
Bira tüketimi	50	10,41	5,18	50,23
	100	10,20	5,11	49,90
Kükürt dioksit miktarı	50	3,20	2,05	36,03
	100	3,25	1,96	39,58
Elektrik tüketimi	50	4347,92	2424,19	44,24
	100	4496,20	2354,07	47,64

Kodlanan fonksiyonlar 50 ya da 100 kez çalıştırıldığında çok yaklaşık sonuçlar üretildiği görülmektedir. Buna göre, tekrar sayısının sonuçları yorumlamak için yeterli olduğunu söylemek mümkündür.

Çizelge 5.1 incelendiğinde, çözümlenen her zaman serisi için, mimari seçiminde geliştirilen tabu arama algoritmasının kullanımıyla elde edilen öngörülerin çok daha iyi olduğu açıkça görülmektedir. Önerilen yöntem kullanıldığında, deneme yanılma yöntemine göre çok daha az hataya sahip, bir başka deyişle çok daha güvenilir öngörüler elde edilmiştir. Önerilen algoritmanın ne denli etkili olduğunu göstermek için, hatanın deneme yanılma yöntemine göre ne kadar azaltıldığı Çizelgenin “İyileşme (%)” başlıklı son son sütununda hesaplanan değerlerle gösterilmiştir. Örneğin, hasta sayıları zaman serisi deneme yanılma yöntemi kullanılarak 100 kez çözümlendiğinde, elde edilen HKOK değerlerinin ortalaması 364,69 dur. Buna karşın, önerilen yöntem ile mimari seçimi yapılarak seri 100 kez çözümlendiğinde, elde edilen ortalama HKOK değeri 188,15 dir. Bu durumda son sütunda görüldüğü üzere, deneme yanılma yönteminin ürettiği hata, önerilen yöntemin kullanılmasıyla %48,41 azalmıştır. Benzer şekilde, tüm zaman serileri için sonuçlar incelendiğinde, önerilen yöntemin çok daha doğru öngörüler üreten mimariler seçtiği açıkça görülmektedir.

Her iki yöntemden yararlanılarak, öngörü için kullanılacak mimarinin seçimi her bir zaman serisi için toplamda 150 kez tekrarlanmıştır. Her bir zaman serisi için belirtilen yöntemlerin kullanılmasıyla seçilen, en iyi ve en kötü mimariler ve bu mimarilere ait HKOK değerleri Çizelge 5.2'de verilmiştir.

Çizelge 5.2: Seçilen en iyi ve en kötü mimariler.

Zaman serisi		DY		ÖTA		İyileşme (%)
		HKOK	Mimari	HKOK	Mimari	
Hasta sayıları	En iyi	216,71	8-10-1	118,41	8-12-1	45,36
	En Kötü	620,19	17-10-1	281,35	6-6-1	54,63
Turist sayıları	En iyi	28343,83	6-3-1	20313,21	14-12-1	28,33
	En Kötü	53899,01	13-1-1	43025,50	17-16-1	20,17
Bira tüketimi	En iyi	4,69	9-15-1	1,09	15-13-1	76,76
	En Kötü	13,71	14-16-1	8,46	10-10-1	38,29
Kükürt dioksit miktarı	En iyi	1,95	18-11-1	1,19	7-5-1	38,97
	En Kötü	4,46	1-15-1	2,73	17-15-1	38,79
Elektrik tüketimi	En iyi	2145,50	13-18-1	1109,55	16-4-1	48,28
	En Kötü	7700,15	18-9-1	3266,09	7-8-1	57,58

Çizelge 5.2 incelendiğinde, örneğin elektrik tüketimi zaman serisi için deneme yanılma yöntemi kullanılarak 150 kez seçim yapıldığında, 2145,50 ile en düşük HKOK değerine sahip en iyi mimari 13-18-1 mimarisidir. Önerilen yöntem kullanılarak 150 kez seçim yapıldığında ise en iyi mimari 16-4-1'dir ve sahip olduğu HKOK değeri 1109,55'dir. Deneme yanılma yöntemi kullanıldığında ulaşılabilen en düşük hata değeri 2145,50 iken, önerilen tabu arama algoritması ile mimari seçimi yapıldığında ulaşılabilecek en küçük hata değeri 1109,55'dir. Bir başka deyişle, önerilen yöntem kullanıldığında, ulaşılabilecek en düşük hata değeri %48,28 azalmaktadır. Önerilen yöntemin kullanımıyla hata neredeyse yarı yarıya azaltılmıştır. Benzer şekilde, elektrik tüketimi zaman serisi için 150 tekrarda en kötü seçim durumları incelendiğinde, yine önerilen yöntemin kullanımıyla en kötü durumda bile hatanın %57,58 daha az olduğu görülmektedir. Tüm zaman serileri için elde edilen sonuçlar incelendiğinde, önerilen yöntemin kullanılmasıyla, çok daha güvenilir öngörüler üreten mimarilere ulaşılabildiği açıkça görülmektedir.

## 6. SONUÇ VE TARTIŞMA

Zaman serilerinde öngörü problemi literatürde çok çalışılan bir konudur. Gelecek hakkında daha güvenilir öngörüler elde etmek amacıyla çeşitli yöntemler geliştirilmiştir. Son yıllarda zaman serilerinin modellenmesinde ve geleceğe yönelik tahminlerin hesaplanmasında en çok tercih edilen yöntem YSA yöntemidir. Kullanım kolaylığı, varsayım kısıtlamalarına bağlı kalmadan çalışabilmesi ve uygulamalarda gösterdiği başarıdan dolayı, öngörü elde etmede YSA yöntemi kullanımı gün geçtikçe artmaktadır.

Klasik yaklaşımlara göre en büyük avantajı, çözümlenen zaman serisindeki hem doğrusal hem eğrisel yapıyı modelleyebilme yeteneği olan YSA yöntemini daha da geliştirmek ve daha sağlam öngörüler elde edebilmek amacıyla çalışmalar sürmektedir. Yapılan çalışmalar YSA yöntemini oluşturan bileşenler üzerinde yoğunlaşmaktadır. Öngörü için kurulacak YSA modelinde bileşenlerin doğru seçilmesi elde edilecek öngörülerin iyiliğini doğrudan etkilemektedir. Belirlenecek YSA modelinde en çok seçeneği sunan bileşen mimari yapıdır. Bu nedenle, birçok seçenek arasından en iyi öngörüler verecek yapay sinir ağı mimarisinin doğru seçilmesi önemli bir problemdir. Literatürde en iyi mimarinin belirlenmesi için çeşitli yaklaşımlar geliştirilmiştir.

Literatürde mimari seçimi problemi için önerilen yaklaşımlar tecrübeye dayalı ve sistematik yaklaşımlar olarak iki grup altında toplanabilir. Tecrübeye dayanan yaklaşımlar, veriye dayalı bir yaklaşım olan YSA yönteminde, her zaman serisi için geçerli değildir. Önerilen sistematik yaklaşımlar ise belli özel durumlar için geliştirilmişlerdir ve kullanımları oldukça zordur. Bu nedenlerden dolayı, günümüzde halen mimari seçimi için birçok çalışmada deneme yanılma yöntemi tercih edilmektedir.

Yapılan bu çalışmada, YSA yönteminde en iyi öngörülerin elde edilebilmesi için en uygun mimarinin belirlenmesinde, tabu arama algoritmasına dayalı bir yaklaşım önerilmiştir. En küçük öngörü hatasını verecek mimarinin seçilmesi bir optimizasyon problemidir. Her bir mimari ayrı bir çözümü ve her bir mimariye karşılık gelen öngörü hata değeri amaç fonksiyonu değerini gösterir. Buna göre, hatayı en küçük yaparak, en iyi mimariyi belirleyen bir tabu arama algoritması

geliştirilmiştir. Tabu arama algoritmasına dayalı önerilen yaklaşım sistematik bir yaklaşımdır.

Geliştirilen tabu arama algoritmasının ne derece etkili olduğunu göstermek için gerçek zaman serileri hem önerilen yöntem hem de deneme yanılma yöntemi kullanımıyla çözümlenmiştir. Uygulamada, Hacettepe Hastanesi'ne gelen hasta sayısı, Türkiye'ye gelen turist sayısı, Avusturya'da gerçekleşen bira tüketimi ( $10^6$  litre), Ankara şehri havası için yapılan metre küpteki kükürt dioksit miktarı (mikrogram) ölçümleri ve Türkiye toplam elektrik tüketimi ( $10^6$  kWh) zaman serileri kullanılmıştır. İncelenen zaman serilerinin öngörülmesinde kullanılacak en iyi mimari yapı, hem önerilen yaklaşım hem de deneme yanılma yöntemiyle belirlenerek, test kümeleri üzerinden elde edilen öngörü hata değerleri birbiriyle karşılaştırılmıştır. Yapılan karşılaştırma sonucunda, önerilen tabu arama algoritmasının tüm zaman serileri için öngörü hatalarını büyük ölçüde azalttığı, bir başka ifadeyle çok daha iyi öngörü sonuçları veren mimarileri seçtiği açıkça görülmüştür.

Uygulamada başarısı kanıtlanan tabu arama algoritmasına dayalı yöntemin taşıdığı diğer avantajlar ise aşağıda verildiği gibi özetlenebilir:

- Önerilen yaklaşım sistematik bir yaklaşımdır. Herhangi bir ön bilgi ya da tecrübe gerektirmez.
- Önerilen yaklaşım, çözümlenen zaman serisine göre mimari uzayını incelediğinden, her zaman serisi için kullanılabilir. Sadece belli bir özel durum için geliştirilmemiştir.
- Kullanım kolaylığına sahiptir. Temel bileşenler analizi, deney tasarımı ya da Box-Jenkins yöntemleri gibi ek bir yöntem bilgisi gerektirmemektedir.
- Önerilen yaklaşım çözümler olarak mimarileri, amaç fonksiyonu değeri olarak hatayı kullandığından, hem zaman serileri öngörü probleminde hem de sınıflama problemlerinde kullanılabilir.

## KAYNAKLAR

- Alabaş, Ç., 1999, Tabu arama ve tavlama benzetimi algoritmalarıyla bilgisayar şebekelerinin topolojik optimizasyonu, Yüksek Lisans Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara.
- Aladağ, Ç.H., 2004, Tabu arama algoritması ile bir ders zaman çizelgeleme probleminin çözümü, Yüksek Lisans Tezi, Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, Ankara.
- Aladağ, Ç.H. ve Eğrioğlu, E., 2005, Yapay sinir ağları ile zaman serilerinde öngörü üzerine bir uygulama, 14. İstatistik Araştırma Sempozyumu Bildiriler Kitabı, 441-446.
- Aladağ, Ç.H., Eğrioğlu, E. ve Günay, S., 2007, Doğrusal ve eğrisel aktivasyon fonksiyonlarının yapay sinir ağlarında öngörü üzerindeki etkisi, 5. İstatistik Kongresi 2007 Bildiriler Kitabı, 770-777.
- Aladağ, Ç.H. ve Hocaoğlu, G., 2007, A tabu search algorithm to solve course timetabling problem, Hacettepe Journal of Mathematics and Statistics, 36(1), 53-64.
- Aladağ, Ç.H., Eğrioğlu, E. ve Günay, S., 2008, A new architecture selection strategy in solving seasonal autoregressive time series by artificial neural networks, Hacettepe Journal of Mathematics and Statistics, 37(2), 185-200.
- Aladağ, Ç.H., Başaran, M.A., Eğrioğlu, E., Yolcu, U., Uslu, V.R., 2009a, Forecasting in high order fuzzy times series by using neural networks to define fuzzy relations, Expert Systems with Applications 36, 4228–4231.
- Aladağ, Ç.H., Eğrioğlu, E., Kadılar, C., 2009b, Forecasting nonlinear time series with a hybrid methodology, Applied Mathematics Letters 22, 1467-1470.
- Aladağ, Ç.H., Hocaoğlu, G., Başaran, M.A., 2009c, The effect of neighborhood structures on tabu search algorithm in solving course timetabling problem, Expert Systems with Applications, article in press. (doi:10.1016/j.eswa.2009.04.051)
- Arifovic, J., Gençay, R., 2001. Using genetic algorithms to select architecture of a feedforward artificial neural network, Physica A 289, 574-594.
- Azadeh, A., Ghaderi, S.F., Sohrabkhani, S., 2007, Forecasting electrical consumption by integration of Neural Network, time series and ANOVA, Applied Mathematics and Computation, 186, 1753–1761.
- Balestrassi, P.P., Popova, E., Paiva, A.P., Marangon Lima, J.W., 2009, Design of experiments on nn training for nonlinear time series forecasting, Neurocomputing, Volume 72, Issues 4-6, 1160-1178.

- Battiti, R. ve Tecchiolli, G., 1992, Parallel biased search for combinatorial optimization: Genetic algorithms and Tabu, *Microprocessors and Microsystems*, 16, 351-367.
- Bektan, A., 2007, Atatürk'ün kehanetleri, Truva Yayınları, Yayın no: 209, İstanbul, 314s.
- Bodyanskiy, Y. ve Popov, S., 2006, Neural network approach to forecasting of quasiperiodic financial time series, *European Journal of Operational Research*, 175, 1357-1366.
- Bollerslev, T., 1986, Generalised autoregressive conditional heteroscedasticity, *J. Econ.*, 31, 307-327.
- Box, G.E.P. ve Jenkins, G.M., 1970, *Time series analysis: Forecasting and control*, San Francisco: Holden-Day.
- Buhamra, S., Smaoui, N. ve Gabr M., 2003, The box-jenkins analysis and neural networks: prediction and time series modelling, *Applied Mathematical Modeling*, 27, 805-815.
- Charon, I. ve Hurdy, O., 1996, Mixing different components of metaheuristics, *Meta-Heuristics: Theory and applications*, Osman, I.H. and Kelly, J.P. (eds.), Kluwer Academic Publishers, 589-604.
- Cichocki, A., Unbehauen, R., 1993, *Neural networks for optimization and signal processing*, John Willey & Sons, New York.
- Cybenko, G., 1988, continuous valued neural networks with two hidden layers are sufficient, Technical report, Tuft University.
- Cybenko, G., 1989, Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2, 303-314.
- Dam, M. ve Saraf, D.N., 2006, Design of neural networks using genetic algorithm for on-line property estimation of crude fractionator products, *Computers and Chemical Engineering*, 30, 722-729.
- Dammayer, F. ve Voss, S., 1993, Dynamic tabu list management using the reverse elimination method, *Annals of Operations Research*, 41, 31-46.
- Durbin, B., Dudoit, S., van der Laan, M.J, 2008, A deletion/substitution/addition algorithm for classification neural networks, with applications to biomedical data, *Journal of Statistical Planning and Inference*, 138, 464 - 488.
- Eğrioğlu, E., Aladağ, Ç.H. ve Günay, S., 2008, A new model selection strategy in artificial neural network, *Applied Mathematics and Computation*, 195, 591-597.
- Engle, R.F., 1982, Autoregressive conditional heteroscedasticity with estimates of the variance of UK inflation, *Econometrica*, 50, 987-1008.

- Fahlman, S.E. ve Lebiere, C., 1990, The cascade correlation learning architecture, Technical Report CMU-CS-90-100, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.
- Gareta, R., Romeo, L.M., Gil, A., 2006, Forecasting of electricity prices with neural networks, *Energy Conversion and Management*, 47, 1770–1778.
- Gendreau, M., Laporte, G., Potvin, J.Y., 1995, Metaheuristic for the vehicle routing problem, *Local Search Algorithms*, John Wiley & Sons, Chichester.
- Glover, F. ve Laguna, M., 1997, *Tabu Search*, Kluwer, MA.
- Glover, F., 1996, Tabu search and adaptive memory programming – advances, applications and challenges. Technical Report, College of Business, University of Colorado, Boulder.
- Glover, F., 1995, Tabu search fundamentals and uses. Technical Report, Graduate School of Business Administration, University of Colorado, Boulder.
- Glover, F., 1990a, Artificial intelligence, heuristic frameworks and tabu search. *Managerial and Decision Economics*. 11, 365-375.
- Glover, F., 1990b, Tabu search-Part II, *ORSA Journal on Computing*, (2)1, 4-32.
- Glover, F., 1990c, Tabu search: A tutorial, Technical Report, University of Colorado, Boulder.
- Glover, F., 1989a, Candidate list strategies and tabu search, Technical Report, Graduate School of Business Administration, University of Colorado, Boulder.
- Glover, F., 1989b, Tabu search-Part I, *ORSA Journal on Computing*, 1(3), 190-206.
- Glover, F., 1986, Future paths for integer programming and links to artificial intelligence, *Computers & Operations Research*, 13(5), 533-549.
- Gonzalez-Romera, E., Jaramillo-Moran, M.A., Carmona-Fernandez, D., 2007, Forecasting of the electric energy demand trend and monthly fluctuation with neural networks, *Computers & Industrial Engineering*, 52, 336–343.
- Granger, C.W.J. ve Anderson A.P., 1978, An introduction to bilinear time series models, Vandenhoeck and Ruprecht, Gottingen.
- Granger, C.W.J. ve Terasvirta, T., 1993, *Modelling nonlinear economic relationships*, Oxford University Press, Oxford.
- Gutierrez, R.S, Solis, A.O., Mukhopadhyay S., 2008, Lumpy demand forecasting using neural networks, *International Journal of Production Economics*, 111, 409-420.

- Günay, S., Eğrioğlu, E. ve Aladağ, Ç.H., 2007, Tek değişkenli zaman serileri analizine giriş, Hacettepe Üniversitesi Yayınları, ISBN 978-975-491-242-5, Ankara, 230s.
- Hecht-Nielsen, R., 1990, Neurocomputing, Addison-Wesley, Menlo Park.
- Hill, T., O'Connor, M., Remus, W., 1996, Neural network models for time series forecasts, Management Science, 42, 1082–1092.
- Hornik, K., Stinchcombe, M., White, H., 1989, Multilayer feedforward networks are universal approximators, Neural Networks, 2, 359–366.
- Hübscher, R. ve Glover, F., 1994, Applying tabu search with influential diversification to multiprocessor scheduling, Computers & Operations Research, 21(8), 877-884.
- Kadılar, C., 2005, SPSS uygulamalı zaman serileri analizine giriş, ISBN 975-8201-95-6, Ankara, 299s.
- Kalınlı, A., 1996, Geri beslemeli yapay sinir ağlarının genetik operatörlere dayalı tabu araştırma algoritması kullanılarak eğitilmesi, Doktora Tezi, Erciyes Üniversitesi Fen Bilimleri Enstitüsü. Kayseri.
- Kang, S., 1991, An investigation of the use of feedforward neural networks for forecasting, Ph.D. Thesis, Kent State University.
- Kelly, J.P., Golden, B.L., Assad, A.A., 1993, Large scale controlled rounding using tabu search with strategic oscillation, Annals of Operations Research, 41, 69-84.
- Lachtermacher, G. ve Fuller, J.D., 1995, Backpropagation in time-series forecasting, Journal of Forecasting, 14, 381-393.
- Lahnajarvi, J.J.T., Lehtokangas, M.I., Saarinen, J.P.P., 2002, Evaluation of constructive neural networks with cascaded architectures, Neurocomputing, 48, 573–607.
- Lapedes, A. ve Farber, R., 1988, How neural nets work, in: anderson, d.z., (ed.), neural information processing systems, American Institute of Physics, New York, 442-456.
- Laporte, G. ve Osman, I.H., 1995, Metaheuristics in combinatorial optimization, Annals of Operations Research, Vol. 60, J. C. Baltzer Science Publishers, Basel, Switzerland.
- Lippmann, R.P., 1987, An Introduction to Computing with Neural Nets, IEEE ASSP Magazine, April, 4-22.
- Mooney, E.L. ve Rardin, R.L., 1993, Tabu search for a class scheduling problems, Annals of Operations Research. 41, 253-278.



- Murata N., Yoshizawa S., Amari S., 1994, Network information criterion-determining the number of hidden units for an artificial neural network model, *IEEE Transacion on Neural Networks*, 5, 865-872.
- Murty, K.G., 1995, *Operations Research: Deterministic optimization models*, Prentice Hall, New Jersey, 581s.
- Nam, K. ve Schafer, T., 1995, Forecasting International Airline Passenger Traffic Using Neural Network, *Logistics and Transportation*, 31 (3), 239-251.
- Nelles, O., 2001, *Nonlinear system identification*, Springer, ISBN 3-540-67369-5, Germany, 785s.
- Niska, H., Hiltunen, T., Karppinen, A., Ruuskanen, J., Kolehmainen, M., 2004, Evolving the neural network model for forecasting air pollution time series, *Engineering Applications of Artificial Intelligence*, 17, 159–167.
- Osman, I.H., 1991, *Meta strategy simulated annealing and tabu search algorithms for combinatorial optimization problems*, PhD. Thesis, Imperial College, University of London.
- Osman, I.H. ve Kelly, J.P., 1996, *Meta-Heuristics: Theory and Applications*, Kluwer Acedemic Publishers, Norwell, MA.
- Qi, M. ve Zhang, G., 2001, An investigation of model selection criteria for neural network time series forecasting, *European journal of operational research*, 132, 666-680.
- Pham, D. T. ve Karaboğa, D., 2000, *Intelligent optimization techniques*, Springer-Verlag London Limited, Great Britain.
- Picton, P.D., 1994, *Introduction to neural networks*, Macmillan Press Ltd., London, 168s.
- Pino, R, Parreno, J., Gomez, A. ve Priore, P., 2008, Forecasting next-day price of electricity in the Spanish energy market using artificial neural networks, *Engeneering Applications of Artificial Intelegence*, 21(1), 53-62.
- Priestley, M.B., 1988, *Non-linear and non-stationary time series analysis*, Academic Press, London.
- Rangaswamy, B., Jain, A.S., Glover, F., 1998, Tabu search candidate list strategies in scheduling, University of Colorado at Boulder CO, University of Dundee, Scotland, UK.
- Rao, T.S., 1992, Analysis of nonlinear time series (and chaos) by bispectral methods. In: Casdagli, M., Eubank, S. (Eds.), *Nonlinear modeling and forecasting*, Santa Fe Institute Studies in the Sciences of Complexity Proceedings, vol. XII, pp. 199–226.
- Rathbun, T.F., Rogers, S.K., DeSimo, M.P., Oxley, M.E., 1997, MLP iterative construction algorithm, *Neurocomputing* 17, 195-216.

- Reed, R., 1993, Pruning algorithms a survey, *IEEE Transactions on Neural Networks*, 4, 740-747.
- Reeves, C.R., 1993, *Modern Heuristic Techniques for Combinatorial Problems*, Halsted Press, John Wiley & Sons. Inc., New York, 303s.
- Rolland, E. ve Pirkul H., 1991, Heuristic search for graph partitioning, 31<sup>st</sup> Joint National TIMS/ORSA Meeting, Nashville.
- Roy, A., Kim, L.S., Mukhopadhyay, S., 1993, A Polynomial time algorithm for the construction and training of a class of multilayer perceptrons, *Neural Networks*, 6, 535-545.
- Roy, A. ve Mukhopadhyay, S., 1997, Interactive generation of higher-order nets in polynomial time using linear programming, *IEEE Transactions on Neural Networks*, 8, 402–412.
- Rumelhart, D.E., McClelland, J., 1987, *Parallel Distributed Processing*, MIT Press, ISBN 978-0-262-68053-0, Cambridge.
- Sait, S.M. ve Youssef, H., 1999, *Iterative computer algorithms with applications in engineering*, Computer Society, California.
- Sarle, W.S., 1994, Neural networks and statistical models, *Proceedings of the Nineteenth Annual SAS Users Group International Conference*, April.
- Sharda, R. ve Patil, R.B., 1992, Connectionist approach to time series prediction: an empirical test, *Journal of Intelligent Manufacturing*, 3, 317-323.
- Siestema, J. ve Dow, R., 1988, Neural net pruning: why and how?, In: *Proceedings of the IEEE International Conference on Neural Networks*, 1, 325-333.
- Smith, K.A., 2002, *Neural networks in business: techniques and applications*, Idea Group Publishing, USA.
- Sun, M. ve McKeown, P. G., 1993, Tabu search applied to the general fixed charge problem, *Annals of Operations Research*, 41, 405-420.
- Tang, Z. ve Fishwick, P.A., 1993, Feedforward neural nets as models for time series forecasting, *Operations Research Society of America*, 5: 4, 374-385.
- Tong, H. ve Lim, K.S. 1980, Threshold autoregression, limit cycles and cyclical data, *J. R. Stat. Soc.*, B 42, 245–292.
- Tong, H., 1983, *Threshold models in non-linear time series analysis*, Springer, New York.
- Tong, H., 1990, *Non-linear time series: a dynamic system approach*, Oxford University Press, Oxford.

- Tsay, R.S., 1991, Non-linear time series analysis: diagnostics and modelling, *Statistica Sinica*, 1, 432–451.
- Wang, Z., Massimo, C.D., Tham, M.T., Morris A.J., 1994, A procedure for determining the topology of multilayer feedforward neural networks, *Neural Networks*, 7, 291-300.
- White, H., 1992, Learning and statistics. In: White, H. (Ed.), *Artificial Neural Networks: Approximation and Learning Theory*. Blackwell, Oxford, UK, s. 79.
- Wong, F.S., 1991, Time series forecasting using backpropagation neural networks, *Neurocomputing*, 2, 147-159.
- Woodruff, D.L. ve Zemel, E., 1993, Hashing vectors for tabu search, *Annals of Operations Research*, 41, 123-137.
- Yuan, H.C., Xiong, F.L., Huai, X.Y., 2003, A method for estimating the number of hidden neurons in feed-forward neural networks based on information entropy, *Computers and Electronics in Agriculture*, 40, 57-64.
- Yule, G., 1927, On a method of investigating periodicity in disturbed series with special reference to Wolfer's sunspot numbers. *Philosophical Transactions of the Royal Society of London*, 226, 267–298.
- Zeng, J., Guo, H., Hu, Y., 2007, Artificial neural network model for identifying taxi gross emitter from remote sensing data of vehicle emission, *Journal of Environmental Sciences*, 19, 427–431.
- Zhang, G., Patuwo, B.E., Hu, Y.M., 1998, Forecasting with artificial neural networks: the state of the art, *International Journal of Forecasting*, 14, 35-62.
- Zhang, G.P., Patuwo, B.E., Hu, Y.M., 2001, A simulation study of artificial neural networks for nonlinear time-series forecasting, *Computers and Operations Research*, 28, 381-396.
- Zou, H.F., Xia, G.P., Yang, F.T., Wang, H.Y., 2007a, An investigation and comparison of artificial neural network and time series models for Chinese food grain price forecasting, *Neurocomputing*, 70, 2913-2923.
- Zou, H.F., Xia, G.P., Yang, F.T., Han, Y., 2007b, A neural network model based on the multi-stage optimization approach for short-term food price forecasting in China, *Expert Systems with Applications* 33, 347–356.
- Zurada, J.M., 1992, *Introduction of artificial neural systems*, West Publishing Company, New York, 683s.

## ÖZGEÇMİŞ

**Adı Soyadı** : Çağdaş Hakan ALADAĞ

**Doğum Yeri** : Ankara

**Doğum Yılı** : 1978

**Medeni Hali** : Bekar

### Eğitim ve Akademik Durumu

**Lise** : 1992 – 1995 Bahçelievler Deneme Lisesi

**Lisans** : 1996 – 2001 Hacettepe Üniversitesi Fen Fakültesi İstatistik Bölümü

**Yüksek Lisans** : 2001 – 2004 Hacettepe Üniversitesi Fen Fakültesi İstatistik Bölümü

**Yabancı Dil** : İngilizce

### İş Tecrübesi

2001 - Hacettepe Üniversitesi Fen Fakültesi İstatistik Bölümü  
Araştırma Görevlisi