

T.C.
İNÖNÜ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

KOMPLEKS KARIŞIMLARIN SPEKTROSKOPİK SİNYALLERİNİN PARALEL
GENETİK ALGORİTMA İLE ANALİZİ VE YORUMLANMASI

FATİH MEHMET AVCU

DOKTORA TEZİ
KİMYA ANABİLİM DALI

MALATYA
Ocak 2012

Tezin Bařlıđı: Kompleks Karıřımların Spektroskopik Sinyallerinin Paralel Genetik Algoritma İle Analizi Ve Yorumlanması

Tezi Hazırlayan: Fatih Mehmet AVCU

Sınav Tarihi: 05.01.2012

Yukarıda adı geen tez jürimizce deđerlendirilerek Kimya Anabilim Dalı'nda Doktora Tezi olarak kabul edilmiştir.

Sınav Jüri Üyeleri

Prof. Dr. Sinan SAYDAM

Fırat Üniversitesi

Yrd. Do. Dr. Mustafa KARAKAPLAN

İnönü Üniversitesi

Prof. Dr. Ali BAYRI

Adıyaman Üniversitesi

Prof. Dr. Bülent ALICI

İnönü Üniversitesi

Prof. Dr. Ergun EKİNCİ

İnönü Üniversitesi

İnönü Üniversitesi Fen Bilimleri Enstitüsü Onayı

Prof. Dr. Asım KÜNKÜL

Enstitü Müdürü

ONUR SÖZÜ

Doktora Tezi olarak sunduđum “Karmaşıık Analitik Sinyallerin Paralel Genetik Algoritma İle Kemometrik Olarak Deđerlendirilmesi” başılıklı bu alıřmanın bilimsel ahlak ve geleneklere aykırı dőşecek bir yardıma bařvurmaksızın tarafımdan yazıldıđını ve yararlandıđım bütőn kaynakların hem metin iinde hem de kaynaklar bۆlőmőnde yۆntemine uygun biimde gۆsterilenlerden olduđunu belirtir, bunu onurumla dođrularım.

Fatih Mehmet AVCU

ÖZET

Doktora Tezi

KOMPLEKS KARIŞIMLARIN SPEKTROSKOPİK SİNYALLERİNİN PARALEL GENETİK ALGORİTMA İLE ANALİZİ VE YORUMLANMASI

Fatih Mehmet AVCU

İnönü Üniversitesi

Fen Bilimleri Enstitüsü

Kimya Anabilim Dalı

117 + xiv sayfa

2012

Tez Danışmanı : **Yrd. Doç. Dr. Mustafa KARAKAPLAN**

Optimizasyon diğer bilim dallarında olduğu gibi analitik kimya araştırmalarında da kullanılan önemli bir hesaplama işlemidir. Yapay zeka kullanımına başlanmadan önce araştırmacılar bir problemdeki parametreleri optimize etmek, aralarındaki ilişkiyi bulmak için deneme yanılma yolunu kullanırlardı. Problemdeki parametre sayısı arttığında çözümsüzlük veya elde edilen çözümden bir sonuç çıkarılamama durumları ortaya çıkmıştır. Yapay zeka tekniklerinden biri olan genetik algoritma son zamanlarda hesaplamalı bilimlerde sıklıkla kullanılır. Bir optimizasyon tekniği olan genetik algoritma, araştırma tekniklerinin alışılmamış bir türü olarak tanımlanmaktadır.

Bu çalışmada amaç kompleks analitik sinyallerin değerlendirilmesinde genetik algoritmanın güçlü bir teknik olduğunu vurgulamak ve özellikle kimya alanında ne derece faydalı bir araç olduğunu göstermektir.

Basitçe bir uzayda doğru sonucu arama işi olarak kullandığımız genetik algoritma, aynı zamanda paralel programlama aracılığı ile birden fazla noktada

alıřtırarak (ada modeli) sonucun bulunma sreside kısaltılmıřtır. Bu tezde farklı ada modelleri kullanıldı ve program en etkili olanı seildi. Geliřtirilen program literatrdeki test fonksiyonları ile denendi, paralel ve paralel olmayan test fonksiyonları karřılařtırılmıřtır.

Bu testlerden sonra geliştirilen yazılım nceki alıřmalarımızdan, İnn niversitesi Merkezi Arařtırma Laboratuvarından ve kimya veri tabanlarından elde edilen veriler kullanılmak suretiyle de test edildi. NMR, UV ve IR verilerinin zmlemesi esnasında bulunan standart hata, ticari programlarla karřılařtırılmıřtır. Geliřtirilen yazılımın ticari yazılımla kısaylandığında daha iyi sonular verdiėi grlr.

Sonu olarak, sonularımız deneysel veriye ok yakın olduėundan geliştirilen yazılımın analitik sinyaller iin gvenli bir řekilde kullanılabileceėi kanaatine varılmıřtır.

Anahtar Kelimeler: Kemometri, Sinyal Deėerlendirme, Genetik Algoritma, Paralel Programlama, Ada Modeli, OpenMPI

ABSTRACT

Ph. D.

**RESOLUTION AND INTERPRETING OF
SPECTROSCOPIC SIGNALS OF COMPLEX MIXTURES
WITH THE PARALLEL GENETIC ALGORITHMS**

Fatih Mehmet AVCU

İnönü University

Graduate School of Natural and Applied Sciences

Department of Chemistry

117 + xiv pages

2012

Thesis Advisor : **Assist. Prof. Dr. Mustafa KARAKAPLAN**

Optimization is also an important calculation method used in analytical chemistry researches, as well as other research disciplines. Researchers used to optimize parameters and find relation between them by-trial-and-error method, before arise of artificial intelligence methods. As the number of parameters increases, it becomes impossible to find a solution and even if there exist a solution it is generally not useful to conclude the problem. Genetic algorithm is one of the artificial intelligence techniques, recently used widely in computational sciences. As an optimization technique, genetic algorithm is described as an unusual example of research techniques.

In this study the aim is to emphasize that genetic algorithm is a powerful technique for evaluating complex analytical signals and to show that it is a very useful tool for chemistry.

Genetic algorithm is simply considered as finding correct answer in space and decreased problem solving duration by running more than one processors

simultaneously(island model) by parallel computation. Different Island models were applied in this thesis and most effective one was selected for program. The developed software was tested with test functions in literature and compared with “parallel and non-parallel” test functions.

After these tests, the developed software was also tested by using the data obtained from our previous studies, Inonu University Central Research Lab. and chemistry databases. Standard error found during solution of NMR, UV and IR data was compared with those from commercial software. It is found that the developed software gives better results compared to commercial software.

As a result, it is concluded that since our results are near to experimental data the developed or written software can be safely used for analytical signals.

Key Words: Chemometrics, Signal Evaluation, Genetic Algorithms, Parallel Programming, Island Model, OpenMPI

Teşekkür

Bu çalışmanın her aşamasında desteğini esirgemedi beni yönlendiren danışman hocam Sayın Yrd. Doç. Dr. Mustafa KARAKAPLAN'a,

Tez çalışması süresince önerileriyle beni yönlendiren Tez izleme Komitesi Üyeleri değerli Hocalarım Sayın Prof. Dr. Ali BAYRİ'ye ve Sayın Prof. Dr. Ergun EKİNCİ'ye,

Tez çalışması süresince bilgi ve desteğini esirgemeyen Sayın Prof. Dr. Bülent ALICI'ya,

Bugünlere gelmemde büyük emeği olan ve desteklerini hiçbir zaman esirgemeyen aileme,

Ayrıca bu çalışmayı (Proje No:2010/14) maddi olarak destekleyen İnönü Üniversitesi Bilimsel Araştırma Projeleri Koordinasyon Birimi'ne teşekkürlerimi sunarım.

Fatih Mehmet AVCU

İçindekiler

Özet	i
Abstract	iii
Teşekkür	v
Şekiller Dizini	ix
Tablolar Dizini	xii
Simgeler ve Kısaltmalar	xiii
1. GİRİŞ	1
1.1 Dayandığı Temeller ve Tezin Amacı	1
1.2 Tezin Genel Tanımı	1
2. KURAMSAL TEMELLER	4
2.1 Kemometri	4
2.2 Optimizasyon	5
2.3 Genetik Algoritma	6
2.3.1 Genel Bilgi	7
2.3.2 Genetik İşlem ve Parametreler	7
2.3.3 Üreme	8
2.3.4 Çaprazlama	8
2.3.5 Mutasyon	9
2.3.6 Yakınsama	10
2.3.7 Genetik Algoritma Durdurma Kriterleri	11
2.3.8 Genetik Algoritmanın Aşamaları, Akış Şeması ve Basit Genetik Algoritma Adımları	12
2.3.8.1 Genetik Algoritmanın Aşamaları	12
2.3.8.2 Genetik Algoritma Akış Şeması	14
2.3.8.3 Genetik Algoritma Adımları	14
2.3.9 Genetik Algoritmanın Diğer Yöntemlerle Kıyaslanması	15
2.4 Analitik Sinyaller	16
2.5 Spektroskopi	17
2.5.1 NMR Spektroskopisi ve Temel Prensipleri	21
2.5.1.1 Manyetik Alanlarda çekirdek	21
2.5.1.2 Salınan Bir Elektromanyetik Alanın Işınlanması	25

2.5.2 UV-Vis Spektroskopisi	28
2.5.3 IR Spektroskopisi	30
2.6 Analitik Kimya ve Genetik Algoritma	32
2.6.1 Eğri Uydurma	33
2.6.2 Parametre Tahmini	34
2.6.3 Fonksiyon Optimizasyonu	34
2.6.4 Dalga Boyu veya Özellik Seçimi	35
2.6.5 Kütüphane Taraması	35
2.7 Hesaplamalı Bilim ve Programlama	36
2.7.1 Seri Programlama	37
2.7.2 Paralel Programlama	37
2.7.3 Paralel Bilgisayar Hafıza Modelleri	39
2.7.3.1 Paylaşılan Hafıza Modeli	39
2.7.3.2 Dağıtılmış Hafıza Modeli	40
2.8 Sinyal İşleme	42
2.8.1 Kimyadaki Sıralı Sinyaller	42
2.8.1.1 Kromotogramlar ve Spektrumlar	42
2.8.2 Sinyal İşleme Temel Bilgileri	42
2.8.2.1 Pik Şekilleri	42
2.8.2.2 Gaussians	44
2.8.2.3 Lorentzians	44
2.8.2.4 Asimetrik Pik Şekilleri	45
2.8.2.5 Tepe Şekli Bilgilerinin Kullanımı	46
2.8.3 Sayısallaştırma	47
2.8.4 Gürültü	47
2.8.4.1 Durağan (Sabit) Gürültü	48
2.8.4.2 Bağlantılı Gürültü	48
2.8.4.3 Sinyalin Gürültüye Oranı	48
2.8.5 Düzgünleştirme/Yumuşatma Fonksiyonları	49
2.8.5.1 Hareketli ortalamalar	49
2.8.5.2 Savitsky-Golay Filtreleri	51
3. MATERYAL ve YÖNTEM	53
3.1 Paralel Genetik Algoritma	53
3.1.1 Global Paralelleştirme	53
3.1.2 Küçük Parçalı Paralel Genetik Algoritmalar	54
3.1.3 Küçük Parçalı Paralel Genetik Algoritmalar-Ada (Island) Modeli	55
3.2 Geliştirilen Yazılımın Akış Şeması	58
3.3 Genetik Algoritma İle Geliştirilen Yazılımın Testi	59
3.3.1 De Jong Fonksiyonu	60

3.3.2	Griewank Fonksiyonu	61
3.3.3	Schwefel Fonksiyonu	63
3.3.4	Test Fonksiyonlarının Paralel ve Paralel Olmayan Ortamda Ada Modeline Uygulanarak Karşılaştırılması	65
4.	ARAŞTIRMA BULGULARI	67
4.1	NMR Verisinin Çözümlemesi ve Gerekli Parametrelerin Optimizasyonu . .	67
4.2	UV-Vis Verilerinin Değerlendirilmesi	78
4.2.1	ZrO_2 'nin UV-VIS Absorbsiyon Spektrumu Değerlendirilmesi	79
4.2.2	$C_{19}H_{16}O$ Spektrumu	81
4.2.3	$C_{45}H_{33}FeO_6$ Spektrumu	82
4.2.4	$C_{36}H_{44}FeN_4O_2$ Spektrumu	83
4.3	IR Verilerinin Değerlendirilmesi	84
4.3.1	$CO_2 + H_2O$ (17K) Spektrumu	85
4.3.2	$(C_{23}H_{30}N_2)_2PdCl_2$, $(C_{27}H_{30}O_4N_2)_2PdBr_2$ ve $(C_{25}H_{26}O_2N_2)_2PdCl_2$ Spektrumları	86
5.	TARTIŞMA ve SONUÇ	89
6.	KAYNAKLAR	92
	EKLER	105
A.	Makale ve Bildiriler	106
B.	Geliştirilen Bilgisayar Programının Kaynak Kodu	107
	Özgeçmiş	117

Şekiller Dizini

2.1	Kemometri Uygulama Alanları	5
2.2	Genetik Algoritma Akış Şeması	14
2.3	Elektromanyetik Tayf	18
2.4	Enerji Seviyeleri Arası Fark	19
2.5	Numuneye gönderilen ve çıkan ışığın şiddeti	20
2.6	Sol: Çekirdek spininin jiroskop modeli. Sağ: $spin\ 1/2$ ve $spin\ 1$ çekirdek için homojen bir manyetik alanda $ I = \sqrt{I(I+1)}$ denklemindeki durumunda olası yönelimler.	21
2.7	Yalpalama Hareketi	23
2.8	Enerji Farkı	24
2.9	Birbirine zıt dönen iki bileşen ile $B_1 \cos(\omega t)$ 'nın gösterimi	25
2.10	Bir kompleks bileşiğin tipik ^1H FID görünüşü.	26
2.11	Kimyasal Kayma	28
2.12	Bir Organik Molekülde Elektronik Geçişler	30
2.13	Moleküldeki Elektronik Geçişler ve UV Spektrumu	30
2.14	Molekül Titreşim Türleri	32
2.15	Seri Hesaplama	37
2.16	İki Matrisin Çarpılması	38
2.17	Paralel Programlama	39
2.18	Paylaşılan Hafıza Modeli	39
2.19	Dağınık Hafıza Modeli	40
2.20	Çok İşlemcili Dağınık Hafıza Modeli	41
2.21	Genel Pik Parametreleri	43
2.22	Gaussian ve Lorentzian Pik Şekilleri	45
2.23	Asimetrik Pikler. (a) Gaussian/Lorentzian (b) Lorentzian/Gaussian	45
2.24	Üç Pikin Kümeleşmesi	46
2.25	Veri Sayısı ve Grafik	47
2.26	Hareketli Ortalama Filtresi	50
2.27	Hareketli ortalama 3,5 ve 7 Noktalı Filtre	51
2.28	Savitsky-Golay 5,7 ve 9 Noktalı Filtre	52
3.1	Global paralelleştirme kavramları: Eşleşmenin rastgele olduğu (Panmiktik) popülasyon solda. Master-Slave yapısı sağda.	53
3.2	İnce Taneli Genetik Algoritmanın Popülasyon Yapısı; Hücresel Modelin Özel Bir Durumu Gösterilmiştir.	55
3.3	Genetik Algoritma-Ada Modeli	55

3.4	Basamak Ada Modeli	56
3.5	Program Aşamasında Denenen Ada Modelleri	58
3.6	Genetik Algoritma Bilgisayar Programı Ara Yüzü	59
3.7	2D De Jong Fonksiyonu, $f(x) = x^2 + y^2$	60
3.8	De Jong (a) 100. Jenerasyon, (b) 200. Jenerasyon, (c) 300. Jenerasyon, (d) 400. Jenerasyon, (e) 500. Jenerasyon	61
3.9	Do Jong Hesaplama	61
3.10	2D-Griewank Fonksiyonu	62
3.11	(a) 1000 Jenerasyon (b) 2000 Jenerasyon (c) 5000 Jenerasyon (d) 10000 Jenerasyon (e) 15000 Jenerasyon	63
3.12	Griewank Hesaplama	63
3.13	2D-Schwefel Fonksiyonu $f(x) = 418,9829n + \sum_{i=1}^n x_i * \sin\sqrt{ x_i }$	64
3.14	Schwefel İterasyon (a)100 Jenerasyon (b)500 Jenerasyon (c)900 Jenerasyon (d)1000 Jenerasyon (e)1300 Jenerasyon	64
3.15	Schwefel Hesaplama	65
4.1	$C_{15}H_{23}N_2Br$ 'nin NMR Spektrumu	67
4.2	Paralel Genetik Algoritmada Araştırma Uzayını Belirleyen Faktörün Etkisi	68
4.3	Adalar Arası Haberleşme Aralığı Sıklığına göre SSR değeri.	69
4.4	Merkezi Ada Modelinde Haberleşme Aralığının Optimizasyona Etkisi	70
4.5	Değişik Ada Modellerinin Optimizasyona Etkisi	71
4.6	Paralel ve Paralel Olmayan Pik Fonksiyonunun Optimizasyona Etkisi	72
4.7	Seçilmiş NMR Spektrum Eğrisi ve İşaretlenmiş Pikler	72
4.8	Deneysel ve Teorik NMR Sonuçlarının Karşılaştırılması	73
4.9	Adil Cevaz 5	74
4.10	Hacı Halil Oğlu	75
4.11	Hacı Kız	76
4.12	Kaba Aşı	77
4.13	ZrO_2 UV-Vis Spektrumu Verisi	79
4.14	Referans [103] için PGA ve Deneysel Veri	80
4.15	Referans [104] için Deneysel ve PGA ile Hesaplanan UV Spektrumu ($C_{19}H_{16}O$)	82
4.16	Referans [105] için Deneysel ve PGA ile Hesaplanan UV-Vis Spektrumu($C_{45}H_{33}FeO_6$)	83
4.17	Referans [106] için Deneysel ve PGA ile Hesaplanan UV Spektrumu ($C_{36}H_{44}FeN_4O_2$)	84
4.18	Referans [107] için Deneysel ve Hesaplanan IR verilerinin Karşılaştırılması	85
4.19	Referans [108] $((C_{23}H_{30}N_2)_2PdCl_2)$ için Deneysel ve Hesaplanan IR verilerinin Karşılaştırılması	86
4.20	Referans [108] $((C_{27}H_{30}O_4N_2)_2PdBr_2)$ için Deneysel ve Hesaplanan IR verilerinin Karşılaştırılması	87

4.21	Referans [108] $((C_{25}H_{26}O_2N_2)_2PdCl_2)$ için Deneysel ve Hesaplanan IR verilerinin Karşılaştırılması	88
------	---	----

Tablolar Dizini

2.1	Kromozomdaki Mutasyon	10
2.2	Önemli İzotopların Rezonans Frekansları	24
3.1	Test Fonksiyonlarının 100 Çalışma Sonrasında Paralel ve Paralel Olmayan Sistemlerde Global Minimum Noktasını Bulma Oranı	66
4.1	Şekil 4.8 için PeakFit ile Pik Sayıları ve SE Değerleri	73
4.2	Adil Cevaz 5 için SE Değerleri	74
4.3	Hacı Halil Oğlu için SE Değerleri	75
4.4	Hacı Kız için SE Değerleri	76
4.5	Kaba Aşı için SE Değerleri	77
4.6	Şekil 4.8 için Bulunan Maksimum Pik Sayısı SE Değerleri	78
4.7	UV Pik Sayısı ve Standart Hata Değerleri	80
4.8	Ticari Yazılı ile Bulunan Standart Hata Değerleri	81
4.9	Referans [104] için SE Değerleri Karşılaştırması ($C_{19}H_{16}O$)	81
4.10	Referans [105] için SE Değerleri Karşılaştırması ($C_{45}H_{33}FeO_6$)	82
4.11	Referans [106] için SE Değerleri Karşılaştırması ($C_{36}H_{44}FeN_4O_2$)	83
4.12	Referans [107] için Ticari Yazılım ve PGA Karşılaştırılması	85
4.13	Referans [108] ($(C_{23}H_{30}N_2)_2PdCl_2$) için Ticari Yazılım ve PGA Karşılaştırılması	87
4.14	Referans [108] ($(C_{27}H_{30}O_4N_2)_2PdBr_2$) için Ticari Yazılım ve PGA Karşılaştırılması	87
4.15	Referans [108] ($(C_{25}H_{26}O_2N_2)_2PdCl_2$) için Ticari Yazılım ve PGA Karşılaştırılması	88

Simgeler ve Kısaltmalar

C	AT&T Bell laboratuvarlarında, Ken Thompson ve Dennis M. Ritchie tarafından UNIX İşletim Sistemi' ni geliştirebilmek amacıyla B dilinden türetilmiş yapısal bir programlama dilidir.
MPI	MPI (Message Passing Interface) bir bilgisayar iletişim protokolüdür. Dağıtık bellekli bir sistemde paralel program koştan düğümlerin arasındaki iletişim için kullanılan filen standart bir protokoldür.
PeakFit	Otomatik olarak bir veri kümesine bir anda 100 kadar tepe arasından zirve belirleme ve yerleştirme yapabilen ve böylece kullanıcılarına tepeleri daha yerinde ve uygun şekilde tanımlama ve verileri için en iyi eşitliği belirleme imkanı veren, spektroskopi ve kromatografide gerçekleştiren doğrusal olmayan tepe ayırma ve analiz yazılım paketidir.
GA	Genetik Algoritma
PGA	Paralel Genetik Algoritma
ADC	Analog Dijital Dönüştürücü
Auto Fit I	En küçük kareler yöntemiyle standart hata hesabı
Auto Fit II	İkinci türev yöntemiyle standart hata hesabı
Auto Fit III	Çıktı parametrelerinden girdiyi tahmin etme (Deconvolution) yöntemiyle standart hata hesabı
SE	Standart Hata
RMSE	Hataların karelerinin toplamının veri sayısına oranının kare kökü
SSR	Fark Kareleri Toplamı

f	Arama Faktörü
PLS	Kısmi En Küçük Kareler Metodu

1. GİRİŞ

1.1 Dayandığı Temeller ve Tezin Amacı

Tezin esas amacı; işlem görmemiş karmaşık analitik sinyallerin (IR, NMR, UV-Vis) değerlendirilmesinde en etkili araç olarak kullanılan genetik algoritmaların, birden fazla işlemcinin kullanıldığı uygun en iyileme yöntemleri önermektir. Daha önce bu konuda yaptığımız çalışmaların paralel hale getirilmesi, yeni yöntemler önerilmesi, önerilen yöntemlerin kompleks karışımların analizinde kullanılabilirliği araştırılacaktır.

1.2 Tezin Genel Tanımı

Veri madenciliğine ve veri işlemeye bilim dallarının bütün alanlarında gereksinim duyulmaktadır. Çeşitli deneylerden elde edilen veriler, veri madenciliği ile değerlendirme sonucunda önceden görülmeyen yeni bilgiler ortaya çıkabilmektedir. Gelişen bilişim teknolojileri ile birlikte artık ön işlemden geçmemiş karışık yapıdaki verilerin analizi de önem kazanmaktadır. Bir çok alanda elde edilmiş veriler yeniden değerlendirilmeye alınmaktadır. Daha önce belirli sistematik yöntemlerle kaydedilmiş veriler yeniden yüksek hızlı işlemcilerle donatılmış bilgisayarlarda anlamlı verilere dönüştürülmektedir. Veri sayısının ve hacminin çokluğu nedeniyle bu verilerin hızlı ve insansız bir şekilde değerlendirilmesi kendi kendine çalışan sistemleri gerektirmekte ve bu yönde yapılan çalışmalar büyük önem taşımaktadır. Ayrıca bilgisayarların hesaplama gücü her geçen gün artmakta ve bu hızın sağladığı olanaklar sayesinde karmaşık sistemlerin çözümüne ilişkin çalışmalar da ön plana çıkmaktadır.

İşlemcilerin hızlarının her geçen gün daha da artmasına rağmen, daha hızlı işlem kapasitelerine ihtiyaç duyulmaktadır. Bu işlemcilerin tek bir sistemde bir arada kullanılması halen teknolojik olarak sınırlıdır. Bunun yanında bilişim

sistemleri arasındaki iletişimin teknolojilerinin de ucuzlaması ile bunların birden fazlasının bir araya getirilerek işlem kapasitesinin doğrusal olarak arttığı sistemler ön plana çıkmıştır. Bu sayede geçmişte ancak büyük araştırma merkezlerinde görebileceğimiz süper bilgisayarlar artık her yerde inşa edilebilir hale gelmiştir. Hesaplama sistemlerinde gözlenen bu gelişme tamamen teknolojik zorlukların aşılması ile olanaklı hale gelmektedir. Diğer yandan kullanılan hesaplama araçlarının hızlarının her geçen gün artmasına rağmen bu araçlarda kullanılan hesaplama algoritmalarının yazılması insan beyin gücüne ve bilgi birikimine bağlı kalmaktadır.

Analitik kimyada kemometrik alanda çalışma yapan araştırmacılarda yukarıda bahsedilen gelişmelerden yararlanmak için son zamanlarda büyük ölçüde çaba harcamaktadırlar. Tezimizde bu teknolojik gelişmelerden aşağıda belirtilen alanlarda yapılan çalışmalara yer verilecektir. Bunlar;

Otomatik Veri Toplama ve Sayısallaştırma: Çok kanallı analog dijital dönüştürücü (ADC) ile aletlerden alınan sinyallerin eş zamanlı olarak insansız değerlendirilmesi ve raporlandırılması yapılacaktır.

DeneySEL Veri Setlerinin Oluşturulması: Çeşitli spektroskopik (UV-VIS, IR, NMR) analiz yöntemleri ile deneySEL veri setleri oluşturulduktan sonra bu deney verileri üzerinde birbiri üzerine binmiş piklerin ayrılması, tanınması, kalitatif ve kantitatif analizleri kendi geliştirdiğimiz paralel genetik algoritma yazılımı ile yapılacaktır. DeneySEL verinin türüne göre ayırma etkinliği ve taban soğurumu da ayrıca değerlendirilecektir.

Gerçek Örneklerle Sistemin Test Edilmesi: Bu aşamada kompleks karışım içeren gerçek örneklerden alınan analitik sinyallerin ayrıştırılması ve değerlendirilmesi yapılacaktır. Analizi yapılacak örneklerin ne olacağına bu aşamada karar verilecektir. Çevresel veya biyolojik örnekler test edileceği gibi, kimyasal proses sırasında deney ortamından alınan örneklerin gerçek zamanlı değerlendirmesinin de yapılabilirliği araştırılacaktır.

Veri Alma ve Değerlendirme Sisteminin Bütünleşik Hale Getirilmesi: Alt aşamaları tamamlanmış ve uygulanabilirliği kanıtlanmış tüm sistemin bütünleşik (yazılım + donanım) hale getirilerek analiz laboratuvarları için uygun hale getirilmesi sağlanacaktır. Amaçlanan bu sistemin analitik cihazlarda insansız analiz sistemlerine bir alternatif getireceğini umuyoruz.

2. KURAMSAL TEMELLER

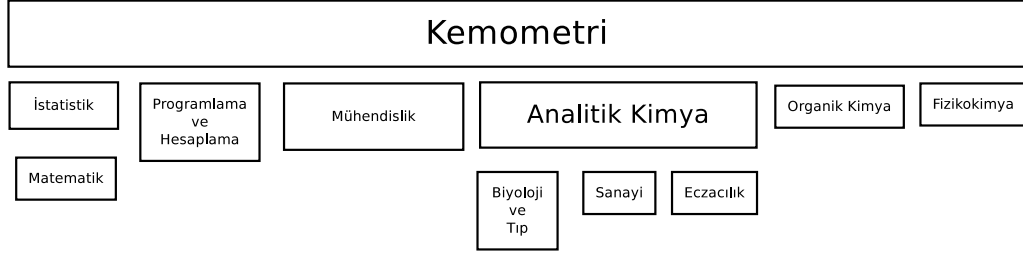
2.1 Kemometri

Temel uygulama alanlarından biri Analitik Kimya olan Kemometri 1970’li yıllarda matematik ve istatistik ile birlikte bilgisayar ve yazılımların kullanıldığı bir bilim adlı olarak söz edilmeye başlanmıştır. İsveçli World ile Amerikalı Kowalski tarafından 1972 yılında ileri sürülmüştür. Sonraki yıllarda uluslar arası kemometri derneği tarafından bu bilim dalının ilk resmi açıklaması yapılmıştır[1].

Kemometrik yöntemler, bilim dallarında özellikle kimya ve analitik kimya alanında kompleks numunelerin analizlerinde hızlı, doğru, kesin ve güvenilir sonuçlara ulaşmak için esnek ve çok yönlü çözümler sunmaya başlamıştır. Yapılan bilimsel makalelerde göstermiştir ki, analitik cihazlarla elde edilen çok boyutlu ve çok değişkenli ölçüm verilerinin işlenmesinde ve değerlendirmesinde kemometrik yöntemlerin en önemli kullanıcısı analitik kimyacılarıdır.

Kemometri içerik olarak, tanımlayıcı ve açıklayıcı istatistik, deney tasarımı, modelleme, sinyal işleme, kalibrasyon, en iyileme (optimizasyon) yapı tanıma, sınıflandırma yapay akıl yöntemleri, resim işleme gibi kavramlardan oluşmaktadır. Biz tezimizde yapay akıl yöntemlerinden biri olan genetik algoritmalar üzerinde duracağız.

Kemometrik yöntemlerin en büyük kullanıcısı analitik kimyacılar olmakla birlikte, laboratuvar ortamında ölçüm yapan komşu branşlarda da kullanıldığı yapılan makalelerde görünmektedir. Kemometrinin diğer disiplinlerle olan bağı şekil 2.1’de gösterilmiştir.



Şekil 2.1: Kemometri Uygulama Alanları

Şekil 2.1 görüldüğü gibi kemometrik çalışmalarda, analitik kimyacıların ve diğer bilim dallarının ihtiyaçları ölçüsünde matematik ve istatistik bilgisine sahip olmaları gerektiği açıktır. Kemometrik çalışmalarda programlama ve hesaplama çok önemlidir. Kemometrik uygulamaların çoğu kompleks uygulamalar içermektedir. Bu hesaplamaları el ile veya basit bir hesap makinesi ile yapmak mümkün değildir. Bu işlemler için özel bilgisayar programlarına ihtiyaç vardır. Bu noktada yol izlenebilir. Bunlar; ya hazır bir paket program satın almak ya da amaca uygun programı kendimizin yazmasıdır. Biz tezimizde ikinci yolu tercih edeceğiz. Dünya üzerinde geçerliliği olan, güçlü ve hızlı bir programlama dili olan C dili ile ayrıca hesaplamaları paralelleştirmek için kullanılan MPI kütüphanesini kullanacağız. Tezimizin ilerleyen bölümlerinde MPI kütüphanesinin açıklaması ve uygulaması bulunmaktadır.

Kemometri ayrıca, adli tıp, biyoloji, gıda kimyası, çevre kimyası, arkeoloji gibi alanlarda da kullanılmaktadır. Sinyal işleme ve çok değişkenli verilerin analizlerinde fizikokimyacıların, reaksiyon basamaklarının oluşturulması ve bunların optimizasyonun da organik kimyacıların, deneysel tasarım ve ilaç tasarımında ve bunların yapı etki araştırmalarında farmasötik kimyacıların kemometriyi bir araç olarak kullandığı görülmektedir.

2.2 Optimizasyon

Belli bir problem hakkında insanlar çalışan insanlar problem hakkında değişik bilgiler toplayarak ve fikirler üreterek en iyi çözüme doğru yol ilerler. Problem hakkında bilgi arttıkça olay daha da iyi anlaşılmaya başlar. Buna benzer olarak, bilim

adamı da uğraştığı konular hakkında bilgisi arttıkça, değişik çözüm yolları bulabilir veya aynı sorunun değişik çözümlerini ortaya çıkarabilir. İşte bu çözümlerden en iyisini en kısa zamanda bulabilmek için geliştirilen bir takım algoritma vardır ki, bunlardan en iyisini tespit edebilmek durum ve şartlara bağlıdır.

Çözüm esnasında en iyi dediğimizde en iyinin ne olduğunu ortaya koymamız gerekmektedir. Genel olarak, en iyi en küçükleme veya en büyükleme şeklinde karşımıza çıkar. Ancak bunların da ötesinde istenilen bir ortalama veya belirli bir hedefe varmak için yapılabilecek en iyilemelerde düşünülebilir. Yeri geldiğinde bir deney tasarımı sırasında kullanılan kimyasalların en aza indirilmesi dahi bir en iyileme olarak düşünülebilir. Böylece, deney için sağlanan para da uygun olarak kullanılmış olur. En iyileme çok belirgin bilgilerle yapılabileceği gibi henüz tam anlamı ile belirlenmemiş, yani eksik veya kesinlik kazanmamış ön bilgilerle de yapılmak zorunda kalınabilir. Genel olarak, basit en iyileme yöntemlerinde belirgin bilgiler kullanılır. Biraz belirsizlik içeren verilerin kesin oldukları kabulü sonucunda en iyileme yöntemleri ile çözüme gidilebilir. Gerçek hayatta hiçbir şey tam kesin değildir. Bu nedenle de belirsizlik içeren en iyileme yöntemleri eskiden ihtimal, istatistik ve stokastik süreçlerle yapılırken bugün daha modern olarak bilgisayarlar yardımıyla geliştirilen algoritmalar yardımıyla yapılmaktadır.

En iyideki 'iyi' kelimesi görecelidir ve uğraşılan sorunun çözümünün bir değil birden fazla seçenekler içerdiğini ima eder. Duruma göre her bir çözümün değeri birbirinden farklıdır. Kabullere, kabul edilen tolerans sınırlarına ve kullanılan yöntemlere göre değişir. Bir bakıma en iyi çözüm sorunun çözülmesinde uğraşanın kişisel görüşüne bile bağlı olabilir.

2.3 Genetik Algoritma

Evrimsel hesaplamanın bir parçası olan genetik algoritmalar Darwin' in evrim teorisinden etkilenecek geliştirilmiştir. Basitçe açıklayacak olursak problem evrimsel süreç kullanılarak bu süreç sonunda en iyi sonucu veren çözüme erişmeye çalışmaktadır. Evrimsel hesaplama bir optimizasyon işlemidir. Amaç hayatta

kalacak olan bireylerin yeteneklerini geliştirerek en iyi bireyi bulmaktır. En iyi çözüm, yapılan çevirimler (iterasyonlar) sonucunda hala hayatta kalabilmeyi başaran çözümdür.

Evrimsel hesaplama ilk olarak I. Rechenberg'in "Evrin Stratejileri" eserinde tanımlanmıştır. Sonraki yıllarda Jonh Holland evrim sürecini bilgisayar ortamında taklit ederek genetik algoritmayı oluşturmuştur. Holland, karmaşık yapıların bir bit dizileri kullanılarak kodlanabileceğini göstermiştir. Hollandın öğrencisi olan D.E. Gollberg genetik algoritmanın çeşitli konularda kullanılabileceğini gösteren bir kitap yayınlamıştır. John Koza ise genetik algoritmayı kullanarak genetik programlamayı geliştirmiştir.

2.3.1 Genel Bilgi

Karmaşık problemleri hızlı ve optimale yakın olarak çözebilen genetik algoritmalar ayrıca, pratik bir yöntem olduğu için geniş bir aralıkta problem çeşidine uygulanabilmektedir. Genetik algoritmanın bir çok problem için çözümü uygulamaları mevcuttur. Genetik algoritma, geleneksel uygulamalarla çözümü çok zor hatta imkansız olarak bilinen bir çok problem üzerinde uygulanabilir. Deneysel çalışmalardan endüstriye kadar geniş bir yelpazede genetik algoritma kullanılmaktadır.

Genetik algoritmalar rassal arama yöntemlerinden olmasına karşın klasik arama yöntemlerinden farklıdırlar. Bu temel farklılıklar; değişkenlerle değil, değişken kümesinin kodlanmış biçimi ile çalışmalarından, tek bir noktadan değil, noktalar kümesinden arama yapmalarından, olasılık içeren kurallar kullanmalarından, türev ve benzeri yardımcı araç kullanmamalarından ve yalnız amaç fonksiyonu bilgisine gerek duymalarından kaynaklanmaktadır.

2.3.2 Genetik İşlem ve Parametreler

Genetik algoritma esasında temel genetik işlem den ibarettir. Bunlar sırasıyla üreme, çaprazlama ve mutasyondur. Genetik işlemler, yeni toplumun oluşturulması, toplumdaki uyumlu kromozomların iyi özelliklerini oluşturan yeni

topluma aktarılması, toplumun bir önceki toplumdaki farklı hale getirilmesi ve çözüm uzayında farklı noktalara erişmek için gerçekleştirilen işlemlerdir. Bu amaçla toplumdaki birey kromozomlarına bu genetik işlemler uygulanarak yeni bireyler elde edilmektedir.

Genetik işlemler toplumdaki bireylere uygulandığından, toplumdaki bireylerin seçilme işi genetik algoritmada önemli bir yer kaplamaktadır. Genetik işlemler üzerinde oldukça etki sahibi olan bu üç genetik parametre aynı zamanda genetik işlemler için bir sınır oluşturma özelliğine de sahiptir.

Ancak şu iyi bilinmelidir ki bir problemin çözümünde kullanılan genetik parametrenin oranları diğer bir problem için ise uygun olmayabilir[8].

2.3.3 Üreme

Genetik algoritma işleyişi için gerekli olan üreme işlemi belli bir seçme kriterine göre bireylerin seçilip yeni kuşağın oluşturulması işlemidir. Bu durum doğal seleksiyondaki gibi en uygunun yaşaması durumuna benzemektedir. Seçme kriterleri uyumluluğu esas alarak birbiriyle uyumlu olan bireyleri seçer. Diğer genetik operatörler çaprazlama ve mutasyon uygulanacak olan bireylerden daha uyumlu yeni bireylerin ortaya çıkması olasılığı dikkate alınarak bireylerin tamamı uygunluk değerine göre seçilebilir veya bir kısmı rastgele seçilerek yeni popülasyona aktarılabilir. Üreme yöntemin amacı, ortalama uygunluğun üzerindeki değerlere çoğalma fırsatı tanımadır[9].

Üreme işleminde dikkat edilecek olan, yeni bireylerin atalarına benzeme olasılığının olmasıdır. Bu olasılığı kaldırmak için çaprazlama ve gerekliyse mutasyon işlemi uygulanmalıdır.

2.3.4 Çaprazlama

Çaprazlama, biyolojik evrim sürecinde olduğu gibi, atadaki genlerin yeni nesle aktarılmasıdır. Çaprazlama işleminde ebeveyn olarak seçilen kromozomlar üzerinde rastgele belirlenen konum veya konumlarda bilgilerin çapraz bir şekilde

yer deęiřtirilmesidir. Bylece ebeveynlerin genleri yeni bireylere aktarılırken kromozomlar arasındaki genetik bilgi srekli deęiřecektir.

aprazlamanın amacı, mevcut iyi kromozomlardan daha iyi kromozomlar elde etmektir. Yeni kromozomlar elde etmek iin aprazlama oranının (p_c) ne olması gerektięine karar vermek gerekir. aprazlama oranı her bir yeni neslin poplasyon byklęnden retilen oęul birey sayısının oranı olarak ifade edilebilir. aprazlamaya bařlamadan nce bu oran belirlenmelidir. Literatr bakıldıęında bu oranın %50-%90 arasında uygulandıęı grlmektedir.

aprazlama oranının yksek tutulması, toplumda deęiřime uęrayacak kromozom sayısının fazla olmasına yani yeni kromozom sayısının artmasına sebep olacaktır. Gereęinden fazla yksek seilmiř bir aprazlama oranında mevcut iyi kromozomların bir sonraki nesle aktarılamaması riski vardır.

Yksek aprazlama oranı, zm uzayını hızlı bir řekilde tararken dięer yandan iyi sonu verebilme ihtimali olan iyi kromozomları da atlayabilir. Yani sistemin global deęilde lokal minimum noktasında takılma ihtimali ortaya ıkabilir. Bu olay genetik algoritmanın performansını dřrebilir.

aprazlama oranının dřk tutulması durumunda ise, deęiřime uęrayacak kromozom sayısı azalacak dolayısı ile hedefe daha ge ulařılacaktır.

Genetik algoritmada en ok kullanılan aprazlama yntemleri ařaęıda anlatılmıřtır. Burada ebeveynler *Kromozom A* ve *Kromozom B*, yavrular ise *Kromozom A'* ve *Kromozom B'* ile temsil edilmektedir.

2.3.5 Mutasyon

Mutasyonun amacı, poplasyonda srdrlebilir genetik eřitlilięi saęlamaktır. Bu iřlem kromozom iindeki genetik materyalin rastgele deęiřimini ierir. Mutasyon genetik algoritmanın bařarısında temel rol oynayan bir faktrdr. aprazlama sonucunda bazen iyi zellikli kromozomlar kaybolabilir yada aprazlama sonucunda artık toplumda farklı trde kromozom oluřmaya bilir. Mutasyon bu kayıp kromozomları bulunmasında yada toplumda yeni trler oluřturmada kullanılır.

Mutasyon; yeni, görülmemiş ve araştırılmamış çözüm elemanlarının bulunmasında kullanılır. Bu operatör aynı zamanda algoritmanın alt bir çözüm noktasında takılmasını engeller ve çalışmanın lokal bir alanda sıkışıp kalmamasını sağlar. Mutasyon doğal seleksiyonda olduğu gibi genetik algoritma işleyişinde de oldukça düşük seçilir.

Mutasyon işleminde en fazla kullanılan tür tersini alma işlemidir. Yani bir kromozomda rastgele seçilmiş bir eleman 1 ise 0, 0 ise 1 ile değiştirilme işlemidir.

Tablo 2.1: Kromozomdaki Mutasyon

1110 1 010	1110 0 010
Çaprazlama ile elde edilen orjinal çocuk	5. değerin mutasyonu sonrası aynı çocuk

Mutasyon işlemlerinde değiştirmeden başka ters çevirme, yer değişikliği ve karşılıklı değişim gibi mutasyon işlemleride vardır.

Mutasyon oranı; popülasyondaki toplam gen sayısının yüzdesel ifadesi olarak kullanılır ve denemeler için yeni genlerin popülasyona girişlerini kontrol eder. Litaretürlere bakıldığında mutasyon oranının genel olarak %0,01 ile %0,001 arasında değiştiği görülmektedir[10].

2.3.6 Yakınsama

Genetik algoritma iterasyon sayısına veya uygun bir çözüme ulaşıncaya kadar devam eder. Bütün kromozomların uygunluk değeri değişmiyorsa genetik algoritma durdurulmalıdır. Çoğu genetik algoritmalar genel ortalama, standart sapma ve minimum uygunluk değerleri gibi istatistiki bilgiler tutarlar. Bu istatistiki bilgiler, yakınsama testi için bir ölçü olabilir. Popülasyonda ortalama uygunluk değeri, algoritmanın yakınsaması hakkında bilgi vermesi bakımından önemli bir kriterdir. Örneğin en büyükleme problemlerinde, farklı iki seçim metodu karşılaştırılıyor ise ortalama uygunluk değeri yüksek olan seçim metodu daha iyi sonuç vermektedir denilebilir.

2.3.7 Genetik Algoritma Durdurma Kriterleri

Yeni nesil; üreme, çaprazlama ve mutasyon işlemlerinden sonra tanımlanmakta ve bir sonraki neslin ebeveynleri olmaktadır. Süreç yeni nesille üreme için belirlenen uygunluk ile devam eder. Bu süreç, önceden belirlenen nesil sayısı kadar veya bir hedefe ulaşıncaya kadar ya da başka bir durdurma kriteri sağlanana kadar devam eder[9].

Bir yerel (lokal) optimum çözüme ulaşıldığında sona eren basit komşuluk arama yöntemlerinden farklı olarak GA stokastik arama yöntemi belirli prensipler altında çalışmaya devam eder. Pratikte, bir sonlandırma kriterine ihtiyaç vardır. Bunlar; istenilen uygunluk fonksiyonu değeri, bilgisayar çalışma zamanı, popülasyon çeşitliğinin geçilmesi olabilir [11].

Fung ve arkadaşları çalışmalarında sonlandırma işleminin, istenen hassasiyet derecesine göre maksimum iterasyon sayısı belirlenebileceğini ve iterasyonun bu sayıya ulaştığında GA döngüsünün sonlandırılabilirliğini belirtmişlerdir. Ayrıca durdurma kriteri olarak iterasyon sayısının kabul edilebileceği gibi hedeflenen uygunluk değeri olarak da kabul edilebileceğini ifade etmişlerdir [12, 13, 14].

Haupt ise genetik algoritmayı durdurmak için seçilecek kriterleri şöyle sıralamıştır[15].

- Seçilen en iyi kromozom, bilinen en iyi çözüme ulaştığında
- Jenerasyonlarda tekrarlı bir şekilde, en iyi kromozom aynı olduğunda
- Belirlenen istatistiksel değerlere erişildiğinde
- Belirlenen jenerasyon sayısı tamamlandığında
- Jenerasyon sonuçları açısından bir gelişme olmadığında
- Optimuma yakın bir değere erişildiğinde

Sonucu önceden belirlenen bir değere yakınsadığında sonlandırılması durumu ise çoğunlukla pratikte uygun olmayan bir durumdur. Ancak genetik algoritmayı

test amacıyla, çözümü önceden bilinen problemlerde kullanılabilir (De Jong, Rastrigin, Schwefel, Griewangk gibi).

Uygulamalarda kullanılan sonlandırma kriterlerinden bir diğeri de, art arda oluşturulan toplumların en iyi amaç fonksiyonu değerleri arasında, bağıl (göreceli) bir farkın kalmaması durumudur. Algoritmada oluşturulan toplumların, daha iyi çözümlere erişip erişemediği bağıl hata değeri α ile ölçülmektedir. Toplumların en iyi amaç fonksiyonu değerleri H_1 ve H_2 arasındaki bağıl hata, aşağıda verilen formül ile hesaplanabilir.

$$\alpha = 100 \frac{|H_1 - H_2|}{H_2} \quad (2.1)$$

Eğer $\alpha < 5$ ise GA işlemlerine son verilir. Aksi durumunda genetik işlemler devam ederek çözüm uzayı aranmaya devam edilir[10]. Algoritma eğer iyi bir çözüme yakınsama yapmıyorsa, GA parametrelerinde değişim yapılabilir. Çaprazlama yöntemi, toplum büyüklüğü, kromozomun kodlama sistemi ya da mutasyon oranı gibi parametrelerin değiştirilmesi ile GA yeniden denenir[15].

2.3.8 Genetik Algoritmanın Aşamaları, Akış Şemesi ve Basit Genetik Algoritma Adımları

2.3.8.1 Genetik Algoritmanın Aşamaları

En basit bir GA aşağıda belirtilen süreçlerden oluşmaktadır[16].

1. **Toplum oluşturulması:** Verilen karmaşık problemlerin aday çözümlerinden (kromozom) oluşan bir toplum oluşturulmasıdır. Kromozomlar, 0 ya da 1 bitlerinden oluşan bir dizi şeklinde ile tanımlanmaktadır.
2. **Uygunluk değerlendirmesi:** Toplumdaki her bir kromozoma sayısal bir değer atayan uygunluk fonksiyonu, kromozom kalitesini, aday çözüm olup olmayacağı şeklinde ölçer.
3. **Yeni toplum oluşturulması:** Yeni toplum oluşturulması için genetik operatörlerin toplumdaki kromozomlara uygulanmasıdır. Bu operatörler;

seçim, çaprazlama ve mutasyondur.

Tipik bir GA aşağıdaki aşamaları gerçekleştirmektedir.

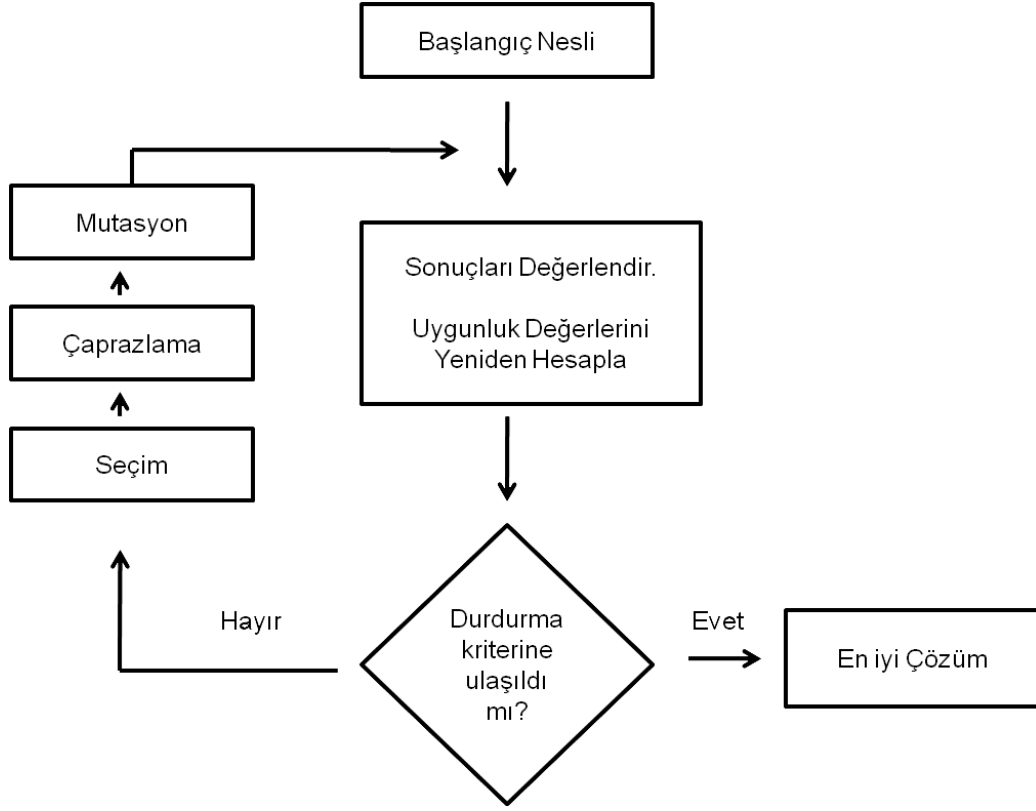
1. **Başlangıç:** n kromozom oluşan rastgele toplum oluşturulur (problemin olası çözümleri).
2. **Uygunluk:** Toplumdaki her x kromozomu için $f(x)$ uygunluk değerini hesapla.
3. **Yeni Toplum:** Aşağıdaki adımlar izlenerek yeni toplum üretilir;
 - (a) **Seçim:** Toplumdan uygunluklarına göre iki ata seçilir (en uygun olanın seçilme şansı daha fazladır).
 - (b) **Çaprazlama:** Çaprazlama olasılığı ile ataları yeni yavru oluşturmak için birbirleriyle eşleştirilir (Eğer çaprazlama yapılmazsa, yavru ataların tıpatıp aynısı olacaktır).
 - (c) **Mutasyon:** Çaprazlama sonucu yeni yavrunun atasına benzeme olasılığı olabilir. Gerekiyorsa mutasyona uğrat.
 - (d) **Kabul:** Yeni yavru, yeni topluma eklenir.
4. **Değiştir:** Yeni toplum algoritmanın tekrar işlenmesinde kullanılır.
5. **Deney:** Eğer bitiş durumu sağlandıysa, durup toplumdaki en iyi çözüm döndürülür.
6. **Döngü:** Adım 2'ye gidilir[6].

Yukarıda da görüldüğü gibi, genetik algoritmanın akışı oldukça kolaydır. İkinci ve altıncı adımlar arasında gerçekleştirilen her bir yineleme, bir jenerasyon ya da yeni toplumu oluşturan bir üreme sürecidir. Her bir üreme sürecinde bir ya da daha fazla uygun kromozom oluşacaktır. Bu oluşum ise, her yeni toplumun bir öncekinden daha iyi olacağını göstermektedir.

Sonuç olarak genetik algoritmalar, belirlenen başlangıç çözüm (kromozom) kümesine genetik işlemlerin uygulanması ile oluşturulan yeni çözüm kümelerinde, amaç fonksiyonunun değişik değerlerini gözden geçirme işleminden ibarettir.

2.3.8.2 Genetik Algoritma Akış Şeması

Akış şeması Şekil 2.2’de gösterildiği gibi işlemler sırasını takip etmektedir.



Şekil 2.2: Genetik Algoritma Akış Şeması

Şekil 2.2’de görüldüğü gibi basit bir genetik algoritmanın başarısı; aday çözümlerin nasıl kodlanacağına, toplum büyüklüğüne, seçim, çaprazlama ve mutasyon operatörlerinin olasılık ve detaylarına ilaveten, maksimum üreme sayısı (iterasyon) gibi detaylara bağlıdır.

2.3.8.3 Genetik Algoritma Adımları

En basit genetik algoritma mekanizması Goldberg[17] tarafından sunulmuştur. Bunun genel yapısı aşağıdaki algoritmada Türkçe olarak kısaca verilmiştir. Bu diyagramda toplumun zamanla değişimini ifade etmek için t sembolü kullanılmıştır. Başlangıç toplumu $T(0)$, t anındaki toplum ise $T(t)$ olarak gösterilmiştir.

Algoritma 2.1 Basit Genetik Algoritma Adımları

```
Başla
    t=0;
    T(t)'yi başlat
    T(t)'yi değerlendir
    Bu sırada
    Başla
        t=t+1
        T(t-1)'den T(t)'yi seç
        T(t) içinde çiftleri yenile
        T(t)'yi değerlendir
    Son
Son
```

2.3.9 Genetik Algoritmanın Diğer Yöntemlerle Kıyaslanması

Son zamanlarda yayımlanan makalelere bakıldığında genetik tabanlı algoritmaya dayanan uygulamalar çok miktarda kullanılmaktadır. Sadece bir bilim disiplini değil birbirinden tamamen farklı bilim disiplinlerinde de kullanıldığı görülmektedir. Buda genetik algoritmanın geleneksel en iyileme yöntemlerinden daha üstün olduğunu göstermektedir.

Konu edilen problem geniş ve karmaşık bir yapı içerdiğinde, eldeki bilgilerin ile sonucun bulunamayacağı düşünüldüğü durumlarda ya da geleneksel arama yöntemleri ile etkin sonuç alınamadığı durumlarda genetik algoritmalarından yararlanılmaktadır. Genetik algoritma ile 10^{30} büyüklüğündeki problemler için bile normal rastgele araştırma metotlarından daha iyi netice verdiği görülmektedir[19].

Diğer araştırma tekniklerinde, türevin var olduğu bilinen doğrusal araştırma ortamlarında kullanılabilirler. Bazı şartlar altında bu tekniklerin performansları oldukça iyidir. Bununla birlikte bu tekniklerde araştırma uzayı çok modlu ise bu durumda tepeye tırmanma ve diğer bu tip algoritmaların en kötü tarafı bölgesel minimum denen noktalara takılma ihtimallerinin çok büyük olmasıdır. Bu durumda global çözümü bulamazlar.

Genetik algoritmanın en iyileme problemlerindeki üstünlükleri göz önüne alındığında, özellikle karmaşık optimizasyon problemlerinin çözümünde neden tercih

edildiği daha iyi anlaşılmaktadır:

- GA'lar parametrenin kendisi ile değil parametre kümesinin kodlanmasıyla çalışmaktadır. $f(x)$ amaç fonksiyonunu en iyileme problemlerinde, x parametresi; sonlu uzunlukta bir dizi olarak kodlanmaktadır. Optimizasyon, bu kod kümesi ile elde edilmeye çalışılmaktadır.
- GA tek bir noktada ya da yönde değil, noktalar kümesi içinde en iyi değeri araştırmaktadır. Algoritmanın bu özelliği, araştırma esnasında her seferinde daha iyi sonuçlara ulaşılmasını garantilemektedir.
- GA'lar amaç fonksiyonu dışında yardımcı bilgilere ihtiyaç duymazlar. Verimli bir araştırma yapılması, her bir dizinin (kromozomun) değerlendirilebileceği bir amaç fonksiyonun tanımlanmasına bağlıdır. Bu da GA'nın performansını artıran önemli bir özelliktir.
- GA'lar kesin bilinen kuralları değil, olasılığa dayalı kuralları kullanmaktadır. Rastgele seçim tekniği kullanılarak, araştırma uzayının başka bir tarafına yönelme yapılmaktadır.

2.4 Analitik Sinyaller

İnsan tarafından anlaşılamayan ve dedekte edilemeyen bir sinyali, insan için anlaşılabilir hale getiren cihazlardan elde edilen (ışın emisyonu, ışın kırılması, elektrik akımı, tepkime hızı gibi) sinyallere analitik sinyal denir. Bu sinyaller kullanılan cihazın detektörü tarafından algılanır ve bizim anlayabileceğimiz bir sinyale dönüştürür.

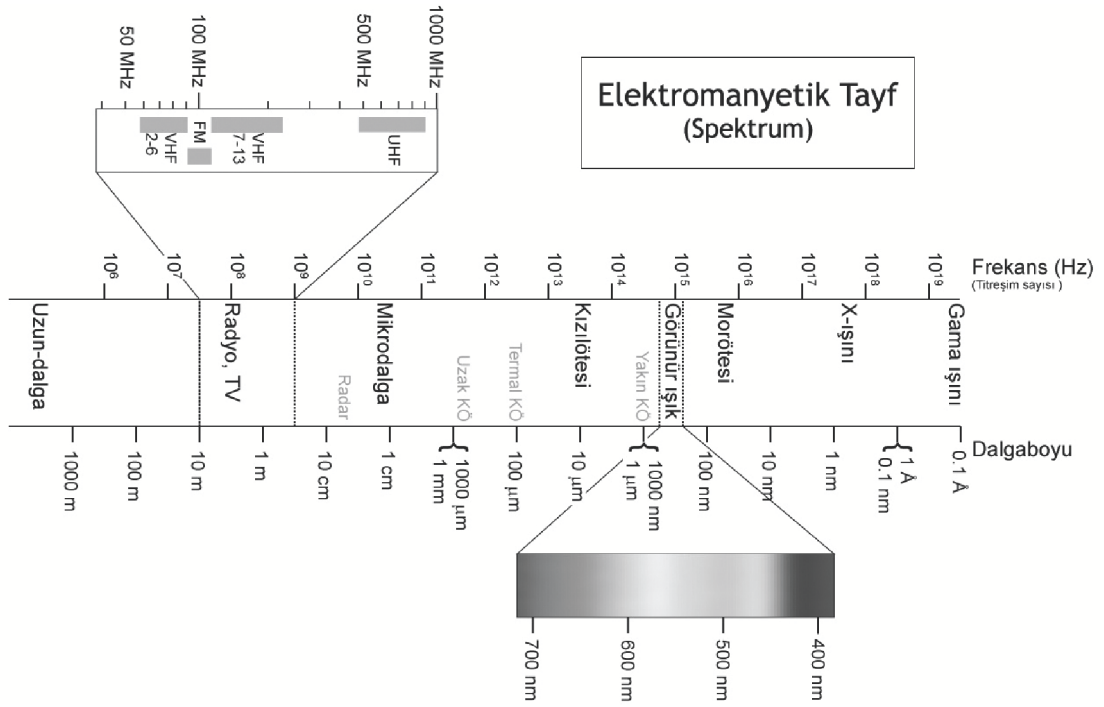
Sinyal kaynağı analitik sinyalin oluşmasını sağlayan, örnek üzerinde fiziksel veya kimyasal değişimlere yol açan bir kaynaktır. Işık, madde, elektrik akımı, ısı gibi sinyaller bir sinyal jeneratörü tarafından oluşturulur. Bu sinyaller örnek üzerine gönderilerek analitik sinyal adı verilen yeni bir sinyal oluşmasını sağlar. Bazı sistemlerde örneğin kendisi doğrudan analitik sinyal verebilir. Örnek, üzerine düşürülen uyarıcı sinyali çeşitli biçimlerde tutar. Üzerine düşürülen bir ışın

ise onu absorplayabilir, yansıtabilir, kırarak içinden geçmesini sağlayabilir. Işığı döndürebilir. Bir elektrik sinyali ise bu elektriğin geçmesine direnç gösterebilir. Kimyasal tepkimeye uğrayabilir. Maddeye ısı verilirse ise ışıma yapabilir, kimyasal tepkimeye uğrayabilir. Bütün bunların sonunda da bir şekilde detektör tarafından algılanabilecek analitik sinyal oluşturur.

Bu sinyal ışık, ısı, madde veya elektrik akımı olabilir. Bu analitik sinyal, sinyal dönüştürücü tarafından elektriksel sinyal haline dönüştürülür. Bu sinyal, sinyal kaydedicide bir dijital değer veya grafik olarak gözlenir. Sinyal dönüştüricide meydana getirilen sinyal, sinyal kaydedicide analizi yapan tarafından algılanamayacak kadar küçük olması halinde bir amplifikatör tarafından önce yükseltgenir. Sinyal kaydedici göz, bir yazıcı, integratör veya bilgisayar olabilir.

2.5 Spektroskopi

Spektroskopi, çeşitli tipte ışınlar kullanılarak madde içi etkileşimleri inceleyen bilim dalı için kullanılan bir terimdir. Önceleri sadece ışın ile madde arasındaki etkileşimleri incelerken, günümüzde madde ve diğer enerji türleri arasındaki etkileşimleri de incelemektedir. Spektroskopik yöntemler, atomik ve moleküler spektroskopi olarak sınıflandırılan geniş bir analitik uygulama alanı olup, günümüzde ışığın ve diğer enerji türlerinin kullanıldığı yöntemler olarak ikiye ayrılmaktadır. Spektroskopi ışığın madde tarafından absorbe edilmesi temeline dayanmaktadır.



Şekil 2.3: Elektromanyetik Tayf

Işık, uzayda sabit hızda doğrusal olarak yayılan ve yayılma eksenini boyunca sinüs salınımları şeklinde birbirine dik elektrik ve manyetik alanlardan oluşmaktadır. Işığın davranışı dalga ve tanecik özellikleriyle açıklanan bir enerji şeklidir.

1) Dalga karakteri: Uzayda dalga hareketleriyle yayılan elektrik ve manyetik vektörlere sahiptir. Spektrofotometride elektrik vektörü kullanılır. Dalga karakterinde ışık, hız ($c = 3.0 \times 10^{10} \text{ cm/sn}$), genlik, frekans (ν) ve dalga boyu (λ) gibi çeşitli parametrelerle tanımlanır.

2) Tanecik karakteri: Bir ışın demeti çok sayıda tanecikten oluşur ve enerjisi aşağıdaki formül ile ifade edilir.

$$E = h\nu = hc/\lambda \quad (2.2)$$

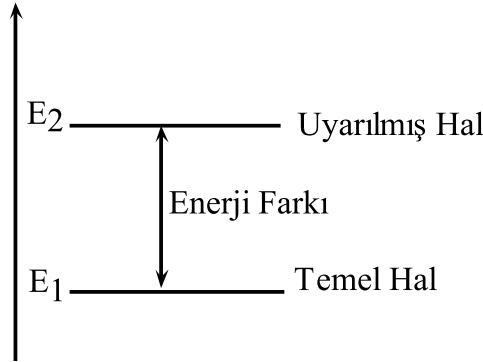
Işık, atom ve moleküller tarafından absorbe edilebilir. UV ve görünür alandaki ışığın absorpsiyonu molekülü oluşturan atomların bağ elektronları ile ilgilidir.

Absorbe edilen enerji, elektronların düşük enerjili orbitallerden (temel hal)

daha yüksek enerjili orbitallere (uyarılmış hal) geçmesine neden olur. Ancak her enerji absorbe edilemez, iki enerji seviyesi arasındaki farka eşit bir enerji absorbe edilebilir. Bu fark;

$$\text{Enerji Farkı} = \Delta E = E_2 - E_1 = h\nu \quad (2.3)$$

şeklinde ifade edilir.



Şekil 2.4: Enerji Seviyeleri Arası Fark

Ultraviyole ve görünür alandaki ışığın absorpsiyonu aynı zamanda bir molekülün vibrasyon ve rotasyon enerji seviyelerinde de değişikliğe neden olur; çünkü bir molekülün toplam enerjisi aşağıdaki formül ile ifade edilmektedir.

$$E_{\text{Toplam}} = E_{\text{Elektronik}} + E_{\text{Titreşim}} + E_{\text{Dönme}} + E_{\text{Öteleme}} \quad (2.4)$$

Aynı zamanda her bir elektronik geçiş enerjisi çok sayıda vibrasyon ve rotasyon enerjilerinin toplamından ibarettir.

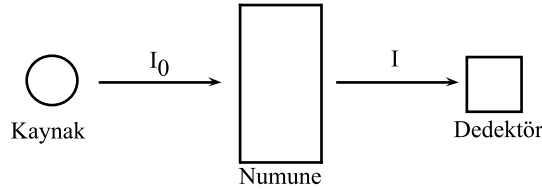
Elektronik enerji diğer enerjilerden büyüktür. Bu nedenle, elektronik uyarılma sırasında vibrasyon ve rotasyon enerji seviyelerinde de değişim olur. Vibrasyon ve rotasyon enerjileri moleküldeki atomların titreşmesi ve molekülün dönmesi ile ilgilidir.

Hangi spektrum bölgesinde olursa olsun belli bir dalga boyundaki ışığın absorpsiyonu, enerjiyi absorbe etme kapasitesine sahip bir yapının varlığının göstergesidir. Bu, en kolay dalga boyunun düzenli olarak değiştirilerek bir

maddenin üzerine düşürülmesinden sonra, ışığın geçmeden önce ve geçtikten sonraki şiddetinin ölçülmesi ile belirlenir. Absorpsiyonun olduğu ya da ışığın geçtiği dalga boyları kaydedilir. Absorpsiyon miktarının dalga boyunun bir fonksiyonu olarak kaydedilmesiyle “absorpsiyon spektrumu” elde edilir.

Atomların elektronik uyarılmaları sonucunda her bir yüksek enerjili atomik orbitale geçişe karşılık fotoğraf plağında bir çizgi gözlenir. Bir atomda da birden çok fazla sayıda atomik orbital olduğu için bu tür uyarılma için çok sayıda çizgiden meydana gelen “çizgi spektrumu” görülür (Şekil 2.13).

Işık absorpsiyonu analitik cihazlarda ölçülür. Bu ölçüm atom, iyon veya molekül üzerine gönderilen ışığın şiddeti (I_0) ile geçen ışığın şiddeti (I) arasındaki farkın ölçümü şeklindedir.



Şekil 2.5: Numuneye gönderilen ve çıkan ışığın şiddeti

Absorbe olan ışık ile örnek konsantrasyonu arasındaki ilişki ilk kez Lambert ve Beer tarafından ortaya konulmuştur.

Beer'e göre aynı derinlikte bir çözeltiden geçen ve çözelti tarafından absorbe edilen monokromatik bir ışın demetinin şiddeti, çözeltinin konsantrasyonu ile logaritmik, üstel veya geometrik olarak azalır. Beer yaklaşımının matematiksel ifadesi,

$$I = I_0 \cdot 10^{-aC} \quad (2.5)$$

şeklindedir. Burada a çözeltinin türüne ve monokromatik ışının dalga boyuna bağlı bir sabit ve C ise konsantrasyonu göstermektedir.

Lambert'e göre ise, bir çözeltiden geçen monokromatik bir ışın demetinin şiddeti, çözelti içinde aldığı yol ile logaritmik, üstel veya geometrik olarak azalır.

Bu ifade aşağıdaki şekilde gösterilebilir.

$$I = I_0 \cdot 10^{-ab} \quad (2.6)$$

Burada b ışığın çözelti içerisinde kat ettiği mesafedir.

Yukarıdaki iki eşitlik kısaca aşağıdaki şekilde verilmektedir.

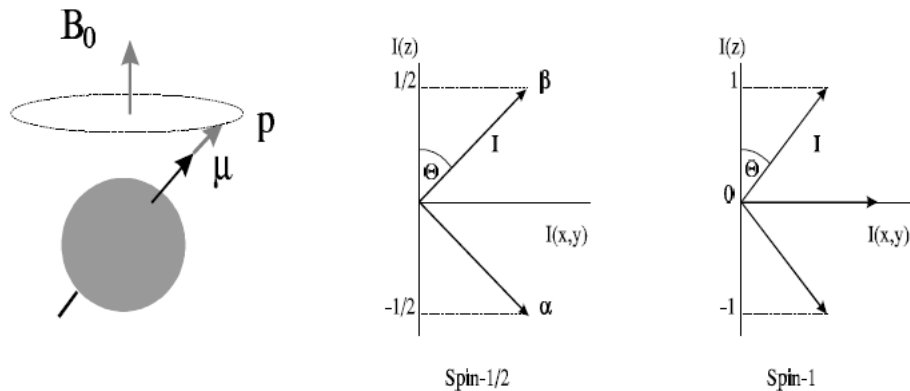
$$A = \epsilon \cdot b \cdot C \quad (2.7)$$

İfadedeki A absorpsiyonu, ϵ molar absorpsiyon katsayısını, b ışık yolunu ve C ise konsantrasyonu göstermektedir.

2.5.1 NMR Spektroskopisi ve Temel Prensipleri

2.5.1.1 Manyetik Alanlarda Çekirdek

Atom çekirdeği nükleonlardan meydana gelir (proton ve nötron). Bu parçacıkların her biri spin adı verilen (açısal momentum gibi davranan) bir özelliğe sahiptir ve bunların toplamı çekirdeğin toplam spinini oluşturur (zıt spinlerin birbirlerini yok etmesinden dolayı sıfır olabilir). Bu spin dış bir manyetik alanla pusula iğnesinin dünyanın manyetik alanıyla etkileşmesine benzer bir şekilde etkileşir (*spin* 1/2 çekirdek için).



Şekil 2.6: Sol: Çekirdek spininin jiroskop modeli. Sağ: *spin* 1/2 ve *spin* 1 çekirdek için homojen bir manyetik alanda $|I| = \sqrt{I(I+1)}$ denklemindeki durumunda olası yönelimler.

- Manyetik alanda hem I hem de I_z kuantize olur.
- Dolayısıyla nükleer spin $2I + 1$ olası şekilde yönelebilir, kuantum sayısı $m_I - I$ ile $+I$ arası değer alır.
- Organik kimyada en önemli çekirdek *spin* 1/2 izotopları olan ^1H , ^{13}C , ^{15}N , ^{19}F ve ^{31}P 'dir.
- *Spin* 1/2 çekirdek gibi manyetik alanda 2 farklı durumları vardır. α ($m_I = -1/2$) ve β ($m_I = +1/2$)

Genellikle statik manyetik alanın yönü z eksenini seçilir ve manyetik kuantum sayısı m_I , m_z olarak adlandırılır. Çünkü spinin z bileşeninin $h/2\pi$ cinsinden büyüklüğünü tanımlar.

$$I_z = m_z h/2\pi \quad (2.8)$$

Bunun sonucunda manyetik moment μ ;

$$\mu = \gamma I \quad (2.9)$$

$$\mu_z = m_z \gamma h/2\pi \quad (2.10)$$

γ , burada izotopa özgü jiromanyetik/magnetojirik sabitidir.

m_z ile tanımlanan bir spin durumu z yönündeki B_0 statik manyetik alanına etkileşme enerjisi şu şekilde tanımlanır.

$$E = -\mu B_0 = \mu B_0 \cos\Theta \quad (2.11)$$

$$E = \mu_z \gamma B_0 h/2\pi \quad (2.12)$$

Spin 1/2 çekirdeğin olası iki spin durumu için ($m_z = \pm 1/2$);

$$E_{1/2} = \frac{0.5\gamma h B_0}{2\pi} \quad (2.13)$$

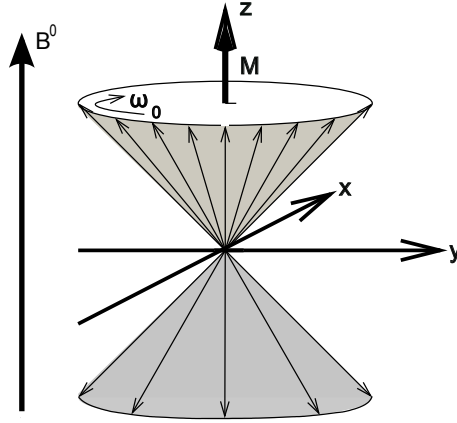
$$E_{-1/2} = \frac{-0.5\gamma h B_0}{2\pi} \quad (2.14)$$

Enerji farkı $\Delta E = E_{1/2} - E_{-1/2} = h\nu = \omega h/2\pi$ larmor frekansı(ω) ile tanımlanan sistem tarafından emilen ya da yayılan enerjiye karşılık gelir.

$$\Delta E = \gamma B_0 h/2\pi \quad (2.15)$$

$$\omega_0 = \gamma B_0 \quad (2.16)$$

Larmor frekansı spinlerin manyetik alan B_0 eksenindeki yalpalama frekansı olarak tanımlanabilir. Larmor frekansının sebebi spinlerin üzerine etki eden ve onları tamamıyla alanla aynı yöne çevirmeye çalışan manyetik kuvvettir.



Şekil 2.7: Yalpalama Hareketi

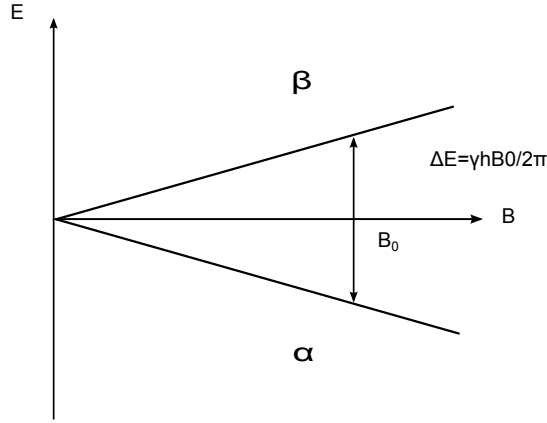
Denklem 2.16'ya göre, bu frekans manyetik alan şiddeti B_0 'a ve spinin jiromanyetik oranı γ 'ya bağlıdır. 11,7 teslalık bir alan şiddeti için en önemli izotopların rezonans frekansları aşağıdaki verilmiştir.

Tablo 2.2: Önemli İzotopların Rezonans Frekansları

İzotop	γ (Göreceli)	11,7 T Alanında Rezonans Frekansı	Doğal Zenginliği	Bağıl Duyarlılık*
^1H	100	500Mhz	%99,98	1
^{13}C	25	125Mhz	%1,1	10^{-5}
^{15}N	-10	50Mhz	%0,37	10^{-7}
^{19}F	94	455Mhz	%100	0.8
^{29}Si	-20	99Mhz	%4,7	10^{-3}
^{31}P	40	203Mhz	%100	0.07

* Tablo verilerinde ayrıca genel band genişlikleri ve gevşeme oranları da dikkate alınmıştır.

Enerji farkı, B_0 , alan şiddeti ile orantılıdır.



Şekil 2.8: Enerji Farkı

Büyük bir spin topluluğunda (NMR gibi) ne kadar enerjinin absorbe olacağı α ve β durumları arasındaki popülasyon farkına bağlıdır (popülasyonlar eşitse rf radyasyonu aynı sayıda spinin enerjisi emmesini ve salmasını sağlayacağından gözlenen net değişim sıfırdır). Boltzmann denkleminde göre;

$$\frac{N(\alpha)}{N(\beta)} = \exp \frac{\Delta E}{kT} = \exp \frac{\gamma h B_0}{2\pi kT} \quad (2.17)$$

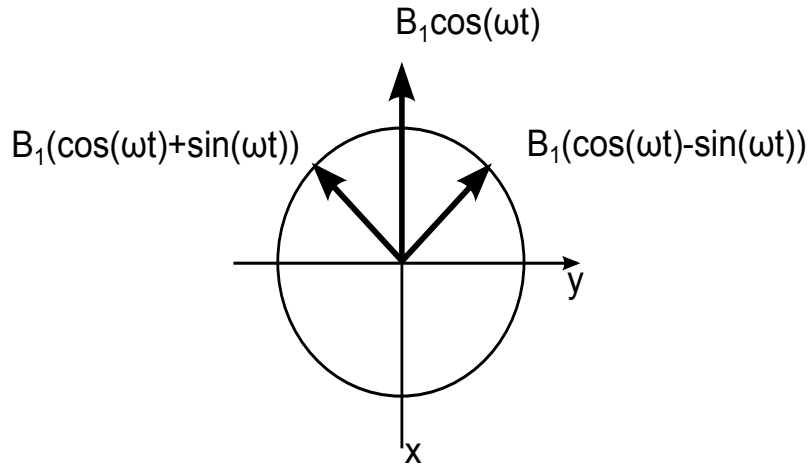
2,35 Tesla (100Mhz) ve 300K için ^1H 'in popülasyon farkı $N(\alpha) - N(\beta) = 8.10^{-6}$ 'dır. Bu da örnekteki toplam spin sayısının %1/1000'inden azdır.

2.5.1.2 Salınan Bir Elektromanyetik Alanın Işınlanması

Absorpsiyon Şartları

Rezonans koşulu rf frekansı Larmor frekansına eş olmalıdır. rf enerjisi α ve β seviyeler arası enerji frekansına eşit olmalıdır.

Doğrusal salınım yapan $B_1 \cos(\omega t)$ alanı birbirine zıt dönen iki bileşenin toplamına eşittir olup bunlar, yalpalama spinleriyle tam olarak rezonans durumundadır.



Şekil 2.9: Birbirine zıt dönen iki bileşen ile $B_1 \cos(\omega t)$ 'nin gösterimi

Dönen Koordinat Sistemi

Laboratuvar koordinat sisteminden, z eksenine etrafında spinlerle rotasyon yapan (rezonans durumundaki) bir koordinat sistemine geçiş herikisinin de statik olmasıyla mümkündür. Genel olarak tüm vektör tanımlamaları rf pulsları v.b. dönen koordinat sistemini kullanır.

Bir rf ışınlamasının makroskopik manyetizasyon üzerindeki etkisi kolaylıkla tanımlanabilir (spinlerin jiroskopik yapıları dikkate alınmalıdır).

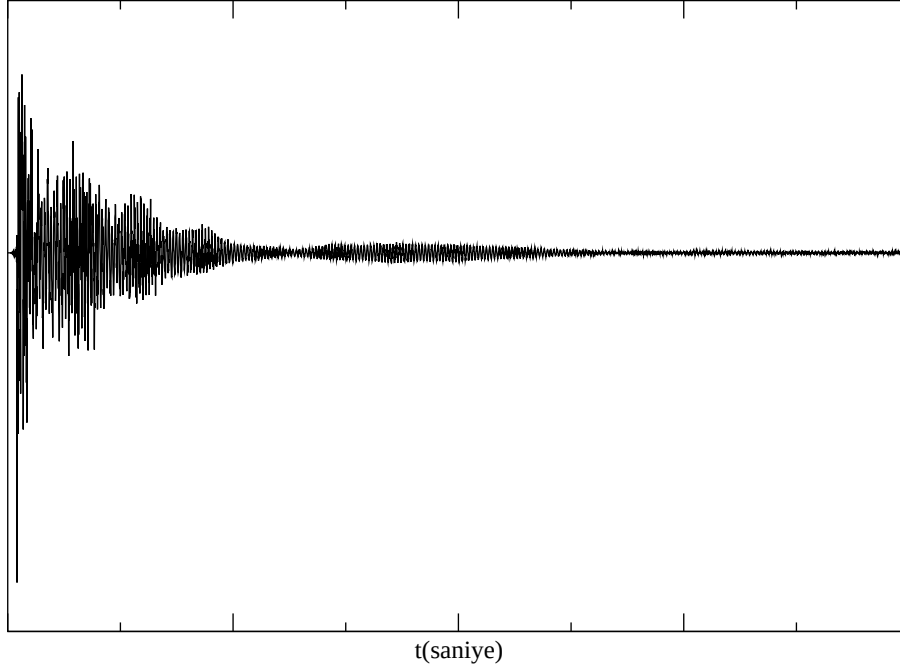
rf pulsunun β döndürme açısı, alan şiddeti B_1 ve geçen zaman τ 'ya bağlıdır.

$$\beta = \gamma B_1 \tau_p \quad (2.18)$$

Tek tek nükleer spinlerden oluşan enine (xy düzleminde) makroskopik

manyetizasyon $M_{x,y}$, z eksenı etrafında B_0 statik alanının etkisinde yalpalama hareketi yapmaya bařlar. Örneęin 90_x^0 pulsdan sonra;

$$M_{-y}(t) = M_{-y}\cos(\omega t) + M_X\sin(\omega t) \quad (2.19)$$



řekil 2.10: Bir kompleks bileřięin tipik ^1H FID grnř.

Denklem 2.19'a gre, FID fazına baęlı olarak sins ya da kosins fonksiyonuyla tanımlanabilir.

Durulma

Uyarılmıř durum Boltzmann dengesine iki mekanizma aracılıęıyla geri dner.

- **Spin-Spin Durulması (Enine Durulma)**

$$dM_{x,y}/dt = -M_{x,y}/T_2 \quad (2.20)$$

Denklem 2.20 faz uyuřmasında bir kayba karřılık gelir. Manyetizasyon x,y dzlemi zerinde dzgn bir řekilde daęılır. B_0 alanı tm spinler iin tam

olarak düzgün olmadığından (diğer spinlerin varlığından kaynaklanan bozulma); T_2 durulması deneysel heterojenlikten dolayı artar.

- **Spin-Yüzey Durulması (Boyuna Durulma)**

$$dM_Z/dt = -\frac{M_Z - M_0}{T_1} \quad (2.21)$$

Boltzman dengesine tekrar ulaşınca kadar, uyarılmış durumun enerjisi yüzeye (örneğin, diğer serbestlik dereceleri moleküler titreşimler, rotasyonlar vb. aracılığıyla) denklem 2.21'deki gibi aktarılır.

Boltzmann dengesinde tüm enine manyetizasyon yok olmalıdır ($T_2 \leq T_1$). Küçük moleküller için $T_1 = T_2$ ' dir. Ancak, T_1 aynı zamanda T_2 'den çok daha uzun olabilir (Bu, taramalar arasındaki durulma gecikmesi açısından önemlidir).

T_1 'in Ölçümü: T_2 durulmasından kaçınmak için sistem Boltzman dengesinden $M_{x,y}$ manyetizasyonu oluşturulmadan çıkarılmalıdır. 180° 'lik bir puls m_z 'yi m_{-z} yönüne çevirir. Daha sonra T_1 durulması tanımlanmış bir τ periyodunda gerçekleşir. Sinyalin tespit edilebilmesi için geriye kalan M_z/M_{-z} bileşeni x, y düzlemine 90° bir pulsle döndürülür ve sinyalin şiddeti ölçülür.

T_2 ve Band genişliği: Fourier transformunun karakteristik özelliklerinden dolayı band genişliği FID'nin bozulma oranına bağlıdır (band genişliğinin yarı yüksekliği için).

$$lw_{1/2} = \frac{1}{\pi T_2} \quad (2.22)$$

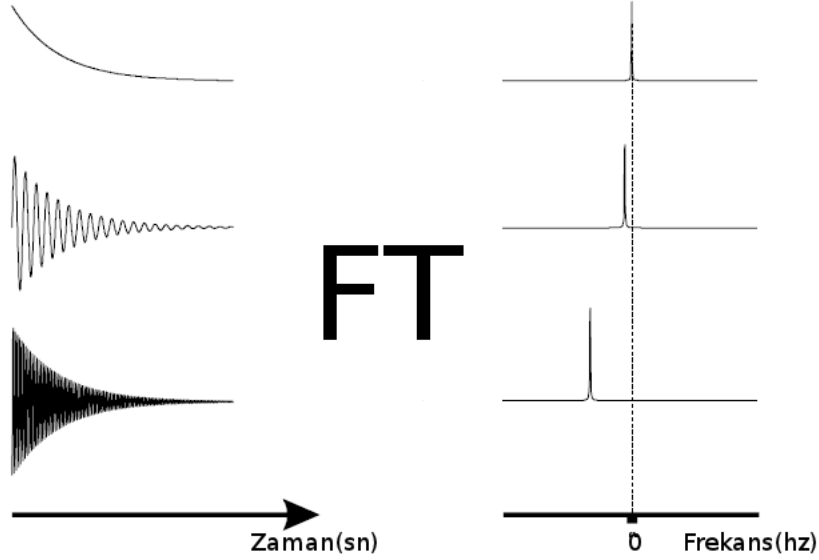
FID üstel olarak azalan sinüs ve kosinüs sinyallerinden oluştuğundan, denklem 2.19, aşağıdaki hale dönüşür.

$$M_{-y}(t) = \{M_{-y}\cos(\omega t) + M_x\sin(\omega t)\} \exp(-t/T_2) \quad (2.23)$$

Kimyasal Kayma

Aynı izotopların farklı moleküler çevrelerde rezonans frekansları bir kaç ppm değişir. 100 Mhz rezonans frekansları mertebesinde bu farklar birkaç bin Hz.

civarındadır. $M_{x,y}$ uyumu oluşturulduktan sonra, her bir spin kendi özel rezonans frekansı ω ile döner ve bu da B_1 verici (ya da alıcı) frekansı ω_0 'dan çok az farklıdır. Döner bir koordinat sisteminde, bu fark $\Omega = \omega - \omega_0$ ofset frekansı farkına eşittir.



Şekil 2.11: Kimyasal Kayma

Hassaslık

Alıcı bobinde indüklenen sinyal;

1. $M_{x,y}$ uyumuna 90 derecelik bir puls ile dönüştürülecek olan M_z 'nin polarizasyon büyüklüğüne (Boltzmann denkleminde göre),
2. Çekirdekte tespit edilen manyetik momente γ_{det} , onun yalpalama frekansına $\omega = \gamma_{det}B_0$ bağlı olan ve alıcı bobinde indüklenen sinyale ,

bağlıdır.

2.5.2 UV-Vis Spektroskopisi

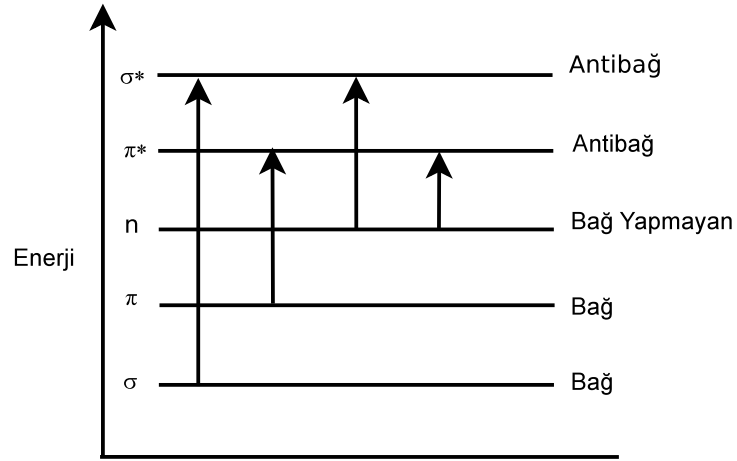
Moleküler absorpsiyon spektroskopisi 160-780 nm dalga boyları arasındaki ışığın, b ışın yoluna sahip bir hücredeki çözeltinin geçirgenliğinin (T) veya absorbansının (A) ölçümüne dayanır. Bu absorpsiyon daha çok moleküllerdeki bağ elektronlarının uyarılmasından kaynaklanır. Bunun sonucu olarak moleküler

absorpsiyon spektroskopisi bir moleküldeki fonksiyonel grupların tanımlanmasında ve aynı zamanda fonksiyonel grupları taşıyan bileşiklerin nicel tayininde kullanılır. UV-Vis spektroskopisi çok sayıda organik ve inorganik bileşiğin analizinde kullanılmaktadır.

Ultraviyole ışımanın enerjisi ise madde üzerinde elektronik geçişler meydana getirmektedir (Şekil 2.12). Aynı zamanda titreşim ve dönme uyarılarına neden olur. Bir organik bileşikte dört tip elektronik geçiş olmaktadır. Bunlar;

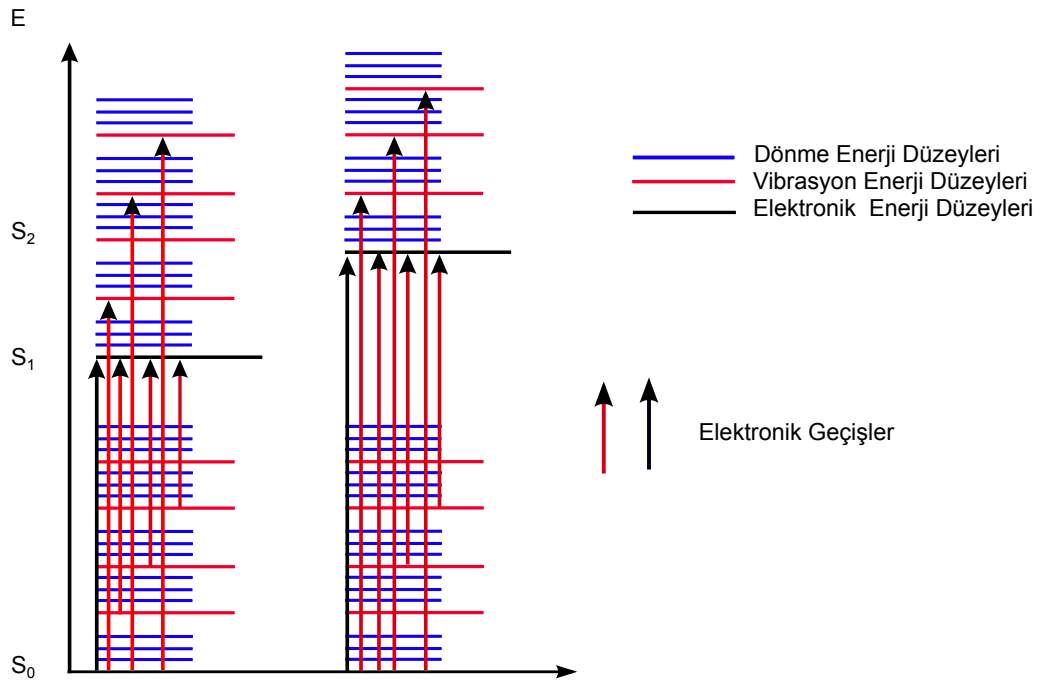
- $\sigma \rightarrow \sigma^*$ Geçişleri: Bir molekülde s bağ orbitalindeki bir elektron, vakum UV bölgesindeki bir ışını absorplayarak antibağ orbitaline uyarılır, bu durumda $\sigma \rightarrow \sigma^*$ geçişi meydana gelmiş olur. Diğer elektronik geçişlere kıyasla $\sigma \rightarrow \sigma^*$ geçişleri için gereken enerji oldukça yüksektir.
- $\pi \rightarrow \sigma^*$ Geçişleri: Bu geçişler ortaklanmamış elektron çiftleri içeren bileşiklerde gözlenir. Genelde bu geçişler $\sigma \rightarrow \sigma^*$ geçişlerinden daha az enerji gerektirir ve absorpsiyon piklerinin çoğu 150-250 nm aralığındaki bölgede yer alır.
- $n \rightarrow \pi^*$ ve $\pi \rightarrow \pi^*$ Geçişleri: 200-700 nm arasındaki spektral bölgede absorpsiyon yaptıklarından UV-Vis spektroskopisinde en çok karşılaşılan geçişlerdir. Bu geçişlerin her ikisi de, π^* orbitallerini içerdiğinden doymamış fonksiyonel grup içeren organik bileşiklerde gözlenir.

Ayrıca, inorganik tuzlarda beşinci bir elektron geçiş türü olan $d \rightarrow d$ geçiş türü inorganik tuzların d orbitallerinde görülür. Bu geçişler görünür bölgedeki zayıf absorblanma ile görülür ve genellikle renklerin değişimine neden olur.



Şekil 2.12: Bir Organik Molekülde Elektronik Geçişler

Moleküllerde ise birden fazla atom içermeleri nedeniyle absorpsiyon sonucunda atomlar için gözlenen çizgilerin bir araya gelmesi söz konusudur ve bunun sonucunda şekil 2.13 gibi spektrum oluşur.



Şekil 2.13: Moleküldeki Elektronik Geçişler ve UV Spektrumu

2.5.3 IR Spektroskopisi

Elektromanyetik spektrumun dalga boyu 0,75 μm ile 1000 μm arasında kalan bölgeye infrared bölgesi adı verilir. İnfrared bölgesi, yakın (0,75 μm -2,5 μm), orta (2,5 μm —15 μm) ve uzak (15 μm —1000 μm) infrared olarak üçe ayrılır.

İnfrared absorpsiyon spektroskopisine titreşim spektroskopisi de denir. Bunun sebebi ise, infrared ışınlarının molekülün titreşim hareketleri tarafından absorplanmasıdır. Çünkü, infrared ışıması UV ve görünür bölge ışıması gibi elektronik geçişleri sağlayacak kadar yüksek enerjili değildir. Ancak, moleküldeki dönme ve titreşim düzeyleri arasındaki geçişleri sağlayabilir.

İnfrared ışının absorplayabilmesi için, bir molekülün titreşim veya dönme hareketi sonucunda, molekülün dipol momentinde net bir değişme meydana gelmelidir. Sadece bu şartlar altında, ışının değişen elektrik alanı ile molekül etkileşebilir ve moleküldeki hareketlerin birinin genliğinde bir değişmeye neden olur. Örneğin, HCl gibi bir molekülün etrafındaki yük dağılımı, klorun hidrojenden daha çok elektron yoğunluğuna sahip olması nedeniyle simetrik değildir. Bu nedenle, HCl'nin belli bir dipol momenti vardır ve bu moleküle polardır. Dipol moment, yük ve yük merkezleri arasındaki uzaklığa bağlıdır. HCl molekülü titreşirken, dipol momentinde bir değişme olur ve ışının elektrik alanı ile etkileşebilecek bir alan meydana gelir. Işının frekansı molekülün doğal titreşim frekansına uyarsa, moleküler titreşimin genliğinde bir değişme meydana getiren net bir enerji alış veriş gerçekleşir; bu da ışının absorpsiyonu demektir. Benzer şekilde, asimetric moleküllerin ağırlık merkezi etrafında dönmesi, ışınla etkileşebilen periyodik bir dipol değişimi meydana getirir.

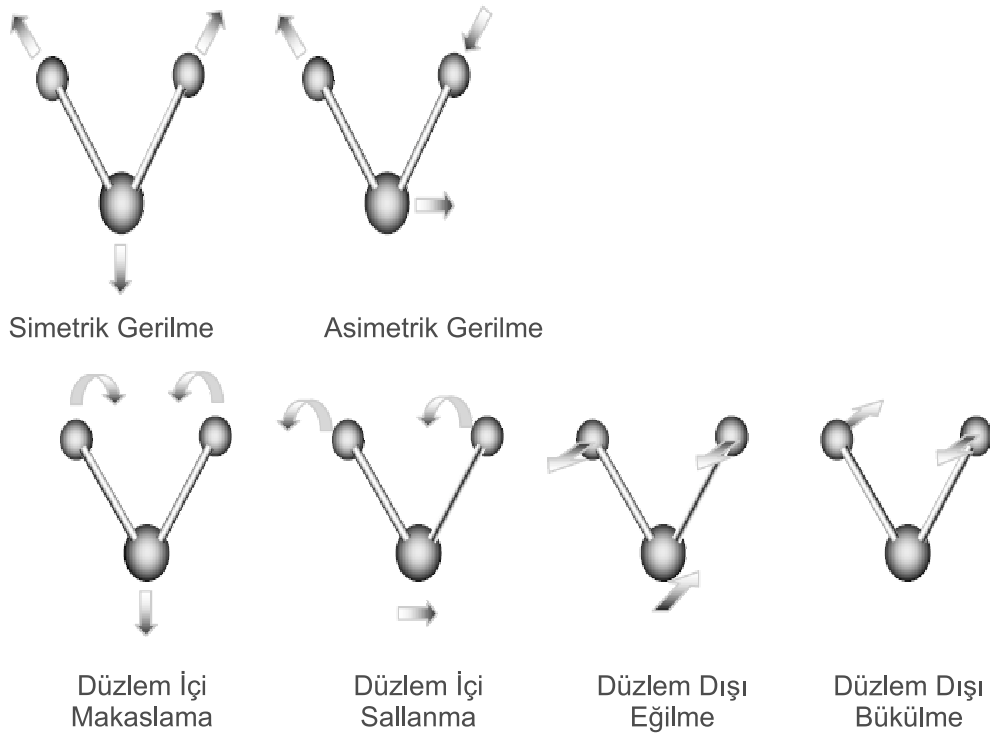
Moleküllerin hareketleri titreşim ve dönme olarak sınıflandırılabilir. Bunun sonucunda spektrumda dönme ve titreşim geçişleri meydana gelmektedir.

Dönme hareketi, asimetric bir molekülde, moleküllerin ağırlık merkezi etrafında dönmesiyle dipol momentte bir değişiklik olur. Dönme enerji seviyesinde bir değişme olabilmesi için gerekli enerji çok küçüktür ($>100 \mu\text{m}$). Dönme seviyeleri kuantlı olduğundan, uzak infrared bölgede gazların absorpsiyonu kesin olarak birbirinden ayrılmış çizgilerle karakterize edilir. Katı veya sıvılarda moleküller arası çarpışma ve etkileşimler, bu çizgilerin genişleyerek sürekli bir spektrum oluşturmaya neden olur.

Titreşim hareketleri ise, gerilme ve eğilme titreşimleri olarak ikiye ayrılır.

Gerilme titreşimleri iki atom arasındaki bağların simetrik veya asimetrik gerilmesi sonucunda (bağ uzaması veya kısalması) ortaya çıkar. Titreşim enerji seviyeleri de kuantlı olup birçok molekül için kuvantum halleri arasındaki enerji farkları orta infrared bölgededir. Her bir titreşim hali birkaç dönme enerji seviyesine sahip olduğundan, gazların infrared spektrumu birbirine yakın çizgi serilerinden ibarettir. Öte yandan, katı ve sıvılarda dönme çok sınırlı olduğundan böyle numunelerde ayrı ayrı titreşim/dönme çizgileri görülmez; onun yerine sadece biraz geniş titreşim pikleri görülür.

N atomlu bir molekül doğrusal ise $3N - 5$ tane, doğrusal değilse $3N - 6$ tane çeşitli türde titreşim kipine sahiptir. Molekülün bunun yanında doğrusal değilse üç tane ötelenme ve üç tane dönme hareketi vardır [37].



Şekil 2.14: Molekül Titreşim Türleri

2.6 Analitik Kimya ve Genetik Algoritma

Sadece kimya dergilerinde bir arama yapıldığında çok sayıda genetik algoritma çalışması görülmektedir. Genetik algoritma literatürünün çoğu sadece yeni uygulamalar üzerine değil aynı zamanda eski uygulamalar üzerinde de

odaklanmıştır. Bu çalışmalarda incelendiğinde analitik kimyada genetik algoritma kullanılabilen çalışmalar beş bölümde incelenebilir. Bunlar;

1. Eğri Uydurma
2. Parametre Tahmini
3. Doğrusal ve doğrusal olmayan fonksiyon optimizasyonu
4. Dalga boyu ve özellik seçiminin sınıflandırılması ve kalibrasyonu
5. Kütüphane Taraması

2.6.1 Eğri Uydurma

Eğri uydurma ile ilgili makalelere bakıldığında genellikle doğrusal olmayan matematiksel fonksiyonların veriye uydurulmasıyla ilgili olduğu görülmektedir. Bu bölümde yapılan çalışmalar, genetik algoritmanın kare regresyonun en küçük ortasının (least median of square regression) non-lineer modelleme ile nasıl genişletilebileceğini incelemektedir[38]. Kare regresyonun en küçük ortası (least median of square regression) ve genetik algoritmanın birleşmesi normal sınırların dışında kalan değerlerin tespitini mümkün kılmaktadır. Genetik algoritmadaki bir uyum fonksiyonuna Markuat'ın metoduna dayanarak non-lineer bir tahmin edicinin yerleştirilmesi çok sayıda birbirinin üstüne geçmiş pikler içeren IR spekturumun çözümünü olanaklı kılmaktadır[39]. Genetik algoritmalar ayrıca Lorenzien çizgi şekillerini Mosbauer spekturumunu uydurmada[40] , NMR[41] da üst üste gelen bandları çözmede[42] ve likit kristal hücrelerdeki kırıcılık verisini uydurmada kullanılırlar. Wise[43] genetik algoritmaların doğrusal olmayan kalibrasyon verilerinden uygun modeli seçmede nöral ağlar kadar etkili olduğunu göstermiştir. Açıkça genetik algoritmaların uygulamalarda iki temel avantajı vardır.

- Daha az kaynaklıdır,
- Hızla yeniden parametrelendirilebilirler.

2.6.2 Parametre Tahmini

Parametre tahmini aslında fiziksel ve kimyasal özelliklerin matematiksel modellemesi ile ilgilidir. Bu çalışmaların çoğu büyük bir çoklu lineer regresyon havuzundan moleküler tanımlayıcılar seçmek için genetik algoritmanın kullanımı içerir. Bu yaklaşım kullanılarak (örneğin genetik regresyon), Jurs[44] lineer regresyon denklemlerinden organik bileşiklerin aktivitesini[44] ve tutuşma sıcaklığını tahmin etmiştir. Genetik algoritmalar ayrıca konsantrasyonun zamana karşı olan verilerinin oran sabitlerini elde etmek amacıyla kinetik bir modele uydurulmasında kullanılırlar. Genetik algoritmanın bu oran sabitlerini belirlemede kullanımı kompleks ve çok modlu bir yüzeyin modellenmesini içerir. En dik azalan Newton-Raftson metoduna hibrid bir genetik algoritmanın eklenmesiyle, Hibert[45] oran sabitlerinin çok modlu tepki yüzeyinden, iyi bir ön tahmin yoksa, elde edilebileceğini göstermiştir. Çeşitli yüzey reaksiyonları için kinetik parametrelerin tahmini[46, 47] bu türden hibrid genetik algoritmalar kullanılarak elde edilmiştir. Hartnett[48] ve Kateman[49] kompleks iyonların oluşum sabitlerinin[48] ve asit ayrışma sabitlerinin[49] uygun bir şekilde yapılandırılmış genetik algoritmanın polarografik ve titrasyon verilerine uygulanmasından elde edilebileceğini göstermişlerdir.

2.6.3 Fonksiyon Optimizasyonu

Bu bölümdeki çalışmalar çok değişken arası ilişkilerin sınıflandırma ve kalibrasyon amacıyla optimizasyonu ile ilgilenir. Parçalı lineer diskriminatların optimizasyonu Small[50, 51] tarafından genetik algoritma tabanlı protokoller kullanılarak araştırılmıştır. Buydens[52] genetik algoritmayı iyon kromatografi veri tabanından sınıflandırma kurallarını öğrenmek amacıyla kullanmıştır. İç ilişkilendirme fonksiyonunun Lineer olmayan PLS'deki (kısmi en küçük kareler metodu) optimizasyonu genetik logaritma ile sağlanmıştır[53]. İyon seçici elektrotlar dizisinden elde edilen tepki verilerinin Nikolskii-Eisenman ifadesine uydurulması genetik algoritma ile sağlanmıştır[54]. Small[55] dijital bir filtre kullanarak gürültülü IR verilerinin düzleştirilmesinin PLS'nin performansını arttırdığını

göstermiştir. Small band geçirici dijital bir filtrenin PLS regresyonu ile olan etkileşmesini yönlendirmek için bir genetik algoritma geliştirmiştir. Small genetik algoritmayı filtrenin konumunu ve genişliğini, regresyon için kullanılan spektrum aralığını ve kalibrasyon modelinin oluşturulmasında kullanılan PLS faktörlerinin sayısını optimizasyonda kullanmıştır. Genetik algoritmalar ayrıca nöral ağı mimarisinin optimizasyonunda[56-63] ve kümeleme algoritmalarının performanslarının arttırılmasında[62, 63] kullanılmıştır.

2.6.4 Dalga Boyu veya Özellik Seçimi

Genetik algoritmaların bu amaçla kullanılması çeşitli guruplar tarafından araştırılmıştır. Bu yaklaşım uygun uyum fonksiyonu seçildiğinde çok etkili olabilir. Leardi[64, 65] ve Brown[66] çok değişkenli kalibrasyonda dalga boyu seçimi konusuyla uğraşmışlar ve bu yaklaşımın geleneksel tüm spektrum metodlarına olan avantajlarını fark etmişlerdir. Bu alandaki araştırmaların çoğu IR üzerinde odaklanmıştır. Kalivas ve arkadaşları[67, 68] genetik algoritmaların spesifik dalga boylarını belirlemede değil IR spektrumunda kalibrasyon için uygun bölgeleri belirlemede kullanılması gerektiğini söylemişlerdir. Small[69] dalga boyu seçimi ve PLS regresyonu için faktör sayısı belirlemek için bir genetik algoritmayı tasarlayarak iyi sonuçlar elde etmiştir. Genetik Regresyon, örneğin değişken seçimi için genetik algoritmayla model geliştirmek için çoklu lineer regresyonun birleştirilmesi, PLS 'ye alternatif olarak bir grup araştırmacı tarafından kullanılmıştır[73-76].

2.6.5 Kütüphane Taraması

Genetik algoritma kimyasal laboratuvaradaki bileşenleri belirlemede kullanılmıştır. Kateman[77] 2D-NMR'da pik belirlemek için bir genetik algoritma rutini geliştirmiştir. Elektro statik potansiyel alanları ile belirlenen 3 boyutlu kimyasal yapı veri tabanlarında benzerlik araştırmaları[78] genetik algoritmayla yapılmıştır. Genetik algoritmalar ayrıca bir kütüphanedeki bir bileşik kümesindeki en yaygın 3 boyutlu yapıyı belirlemede[79] ve bileşiklerin iki boyutlu moleküler grafikler olarak temsil edildiği kütüphane eşleştirme problemlerinden[80] kullanılır.

Yapı tabanlı kütüphane görüntülemesi ilaç tasarımıda önemli bir rol oynar. Bazı kütüphanelerin çok büyük olduğu göz önüne alındığında genetik algoritmanın ilaç keşfinde kütüphane araştırma problemlerinde kullanılması popüler bir alandır ve giderek daha fazla bir sayıda birleşik kütüphanelerin geliştirilmesi ve optimizasyonu ile ilgili genetik algoritma yayını bunun bir kanıtıdır[81-87]. Yapısal farklılıkların ifadesi için uygun metriklerin (birimlerin) geliştirilmesi genetik algoritmanın daha seyrek ve ayrık birleşik laboratuvarlar tasarımıda kullanılması için hayati önem taşımaktadır. Bu yeni önemli bileşiklerin geliştirilmesi için gereken zamanı oldukça kısaltacaktır.

2.7 Hesaplamalı Bilim ve Programlama

Günümüzde bir çok bilim dalındaki insanlar matematiksel modeller oluşturmaktadırlar. Oluşturulan bu modelleri analitik yollarla çözmek modellerin boyutuna göre zorlaşmakta hatta imkansız hal almaktadır. Bilim adamlar kendi modellerini çözmek için bazen analitik yöntemleri kullanmak yerine problemleri bilgisayar ile çeşitli algoritmalar yardımıyla çözerler. İşte herhangi bir bilim disipliniinde analitik yöntemler ile çözülemeyecek kadar büyük olan matematiksel bir modeli, matematik, bilgisayar ve çeşitli algoritmalar ile birleştirerek çözmeye işine hesaplamalı bilim denir.

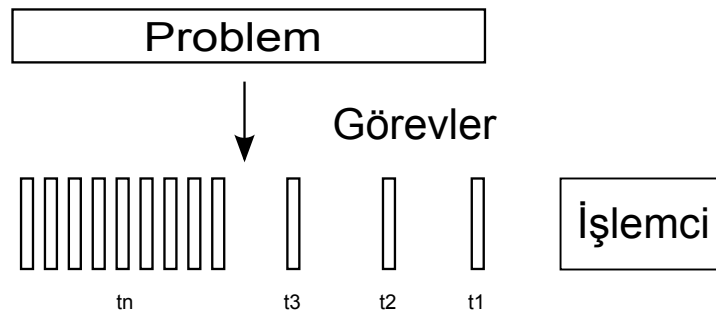
Hesaplamalı bilimde amaç, matematik ve bilgisayar bilimini kendi bilim alanımız ile birleştirip, analitik olarak çözülemeyen probleme istenilen hassasiyete kadar cevap aramaktır. Bu yüzden problemler büyük boyutludur ve problemdeki hata değeri sıfıra yaklaştıkça problem çok daha karmaşık hale gelir. Hatta bazen problem o kadar karmaşık bir hal alır ki, herhangi bir kişisel bilgisayar geliştirilen algoritmayı çözmekte zorlanır. Bazen de çözemez veya sonucu çok geç verir.

Algoritmayı bilgisayar ortamına geçirdiğimizde kullanabileceğimiz iki programlama tekniği vardır. Bunlar seri ve paralel programlamadır. Kimi problemler seri programlamada daha iyi iken kimin problemler de paralel programlamanın üstünlüğü vardır. Bu bağlamda seri ve paralel programlamalara

detaylandırmak ve tezimizin temellerinden biri olan paralel programlamayı genişletmek gerekir.

2.7.1 Seri Programlama

Bilgisayar programları genel olarak sıralı olarak çalışacak şekilde seri algoritma ile yazılırlar. Şekil 2.15'e bakılırsa bilgisayara verilmiş olan görevler sıra ile işlenmekte, sırası ile işlendikten sonra sonraki görev yapılmaktadır. Sıralı komutlar çalıştığında program görevini icra etmiş olur.



Şekil 2.15: Seri Hesaplama

2.7.2 Paralel Programlama

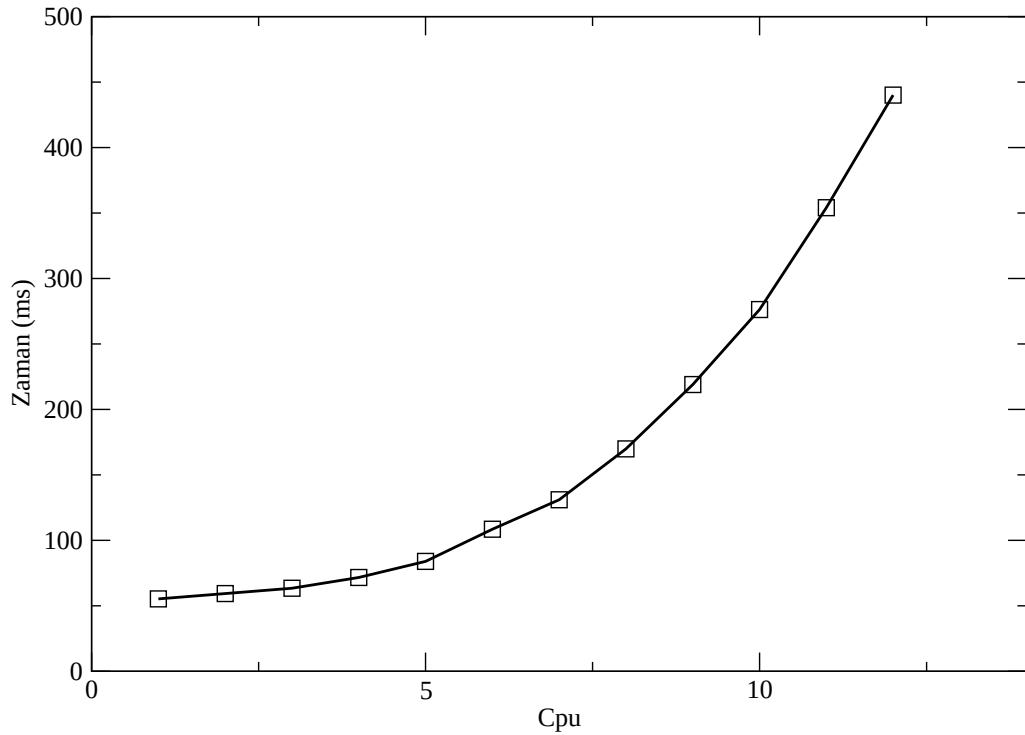
Bilgisayarların işlem gücü fazladır. Fakat günümüz hayat problemlerinde yetersiz kalabilmekte, işlemin çözümü için saatler, aylar, yıllar süren hesaplama zamanına ihtiyaç duyulmaktadır. Örneğin 15-20 atomdan oluşan bir metal kompleksinin enerji optimizasyonu hesabı 2-3 gün sürebilmektedir. Hazırladığımız bilgisayar programında döngüleri, dizileri ve terimleri optimize yoluna gitsek bile bu bize yeteri kadar zaman kazandırmayabilir. Büyük problemi daha kısa sürede çözebilmek için çoklu bilgisayar gücüne ihtiyaç bulunmaktadır. Paralel hesaplamanın birçok yöntemi vardır.

Temelde paralel programlama bir problemin çözülebilmesi için çeşitli bilgisayar kaynaklarının eş zamanlı olarak kullanılmasıdır. Bu yöntem ile işlenecek veri, bölünerek paralel olarak hesaplandığı için hız kazanılmıştır. Paralel programlama, bir bilgisayar programının daha kısa bir sürede çalıştırabilmek için, program komutlarının veya program parçalarının birden fazla işlemci arasında

bölüştürülmesi işlemidir.

Şekil 2.17'ye bakıldığında işlemler eş zamanlı olarak ayrı işlemcilerde (aynı bilgisayar veya farklı bilgisayarlarda) çalıştırılmasıdır. Paralel programlama, çözülmesi uzun zaman gerektiren problemlerin daha kısa sürede çözülebilmesi için bulunmuş bir yöntemdir.

Ancak her problem paralelleştirilmesi bize vakit kazandırır mı sorusu da akıllara gelmektedir. Bir işlemcinin kısa sürede çözebildiği bir problemi paralelleştirmek bize zaman kazandırmaz aksine kaybettirir. Şekil 2.16 incelenirse Canon yöntemi ile çarpılmış 2 matrisin paralelleştirilmesi ile oluşturulmuş, işlemci adedine göre zamanın grafiği görülmektedir. Dikkat edilecek olursa işlemci sayısı

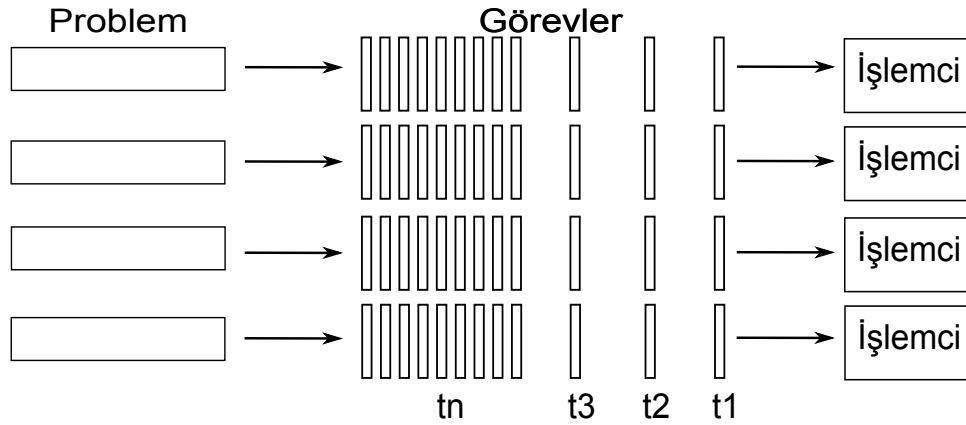


Şekil 2.16: İki Matrisin Çarpılması

Paralel programlama mantığında, ağır hesaplama işlemini aynı anda bir işlemciye yaptırmaktansa, programı parçalara bölüp bu her parçayı başka işlemcilere veya diğer bilgisayardaki işlemcilere yaptırmaktır. Paralel programlama, programın

çalışma hızının, onun eş zamanlı olarak yerel işlemcilerde çalışan parçalara bölünmesi ile artması fikriyle ilgilenir.

Paralel programlama, büyük problemlerin seri algoritmalarla çözümünün uzun zaman alması, maliyetten kazanç, süper bilgisayarlar nazaran maliyeti ucuz hesaplama kaynaklarının varlığı, tek bilgisayarın kısıtlı hafızasının sağladığı dezavantajın üstesinden gelmek için küme bilgisayar hafızalarının kullanılması, eş zamanlılığın sağlanması, aynı zamanda çok uygulamanın yapılabilmesi özelliklerinden dolayı hesaplamalı bilim ile uğraşan bilim insanları için cazip bir hesaplama tekniğidir.

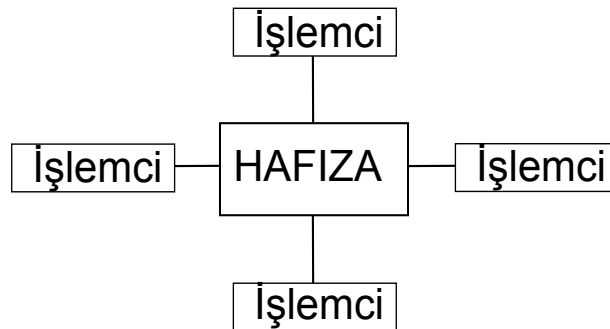


Şekil 2.17: Paralel Programlama

2.7.3 Paralel Bilgisayar Hafıza Modelleri

2.7.3.1 Paylaşılan Hafıza Modeli

Paylaşılan hafızalı paralel bilgisayarlar oldukça yaygın kullanılmaktadır. Bu tür hafıza kullanan işlemlerden her biri hafızanın global adres uzayının tamamını kullanabilirler.



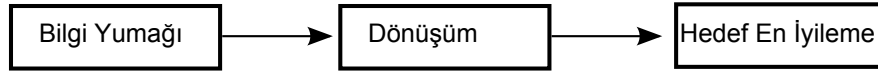
Şekil 2.18: Paylaşılan Hafıza Modeli

İşlemciler bir taraftan bağımsız olarak kendilerine verilen işlemleri yaparken diğer taraftan da ortak hafızadaki bir değişkeni okuyabilir veya değiştirebilir. Böylece bir işlemcinin değişiklik yaptığı hafıza bölgesi diğer işlemciler tarafından da görülebilmekte ve değiştirilebilmektedir.

Programlama açısından kolay bir yöntem olmasına karşın bu modelin bazı dezavantajları vardır. Bunlardan ilk olarak hesaba katılması gereken hafıza ile mikroişlemci arasındaki ölçeklenebilirliktir. Mikroişlemci sayısındaki artış mikroişlemci ile hafıza arasındaki trafiği geometrik olarak arttırmaktadır. Ayrıca bu model komşu bilgisayardaki işlemciye emir veremediğinden sınırlı sayıda işlemci ile işlemlerini yapmaktadır.

2.7.3.2 Dağıtılmış Hafıza Modeli

Paylaşılan hafıza sistemlerinde olduğu gibi dağıtılmış hafıza sistemleri de geniş bir kullanım alanına sahiptir. Dağıtılmış hafıza modelinde komşu bilgisayar işlemcilerin haberleşebilmesi için bir haberleşme ağına ihtiyaç vardır.



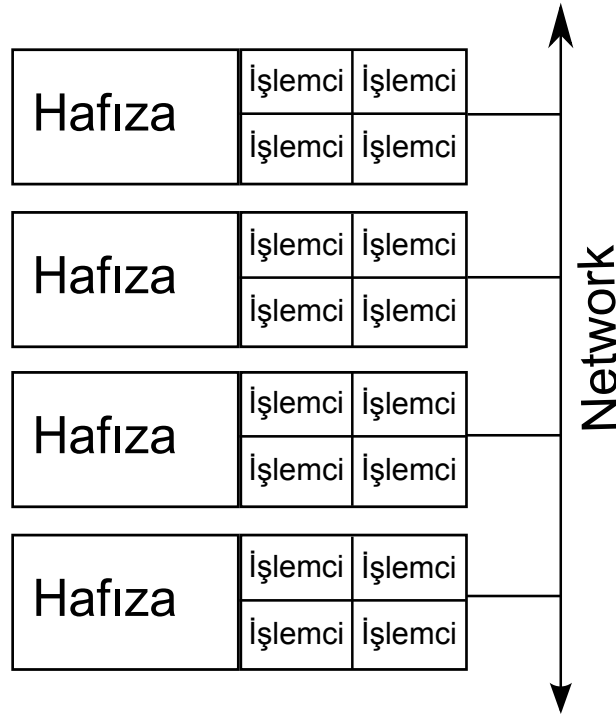
Şekil 2.19: Dağıtık Hafıza Modeli

Şekil 2.19 bakıldığında her işlemcinin kendi lokal hafıza sisteminin olduğu görülmektedir. Bir işlemcinin lokal hafızasına diğer işlemciler tarafından ulaşılamaz. İşlemciler arasında paylaşılan ortak bir hafıza adresi yoktur. Her işlemci kendi lokal hafızasına sahip olduğu için diğer işlemcilerden bağımsız olarak işlem yapabilmektedir. İşlemcilerin kendi lokal hafızalarında yaptıkları değişiklikler diğer işlemcilerin lokal hafızalarını etkilememektedir. İşlemcilerden birisi bir başka işlemcinin üzerinde çalıştığı veriye yada bulduğu sonuca ulaşmak isterse haberleşmenin ne zaman ve ne şekilde olacağının tanımlanması programcının işidir. Veri transferi için genellikle ethernet teknolojisi kullanılmasına rağmen pek çok farklı türden teknoloji de kullanılmaktadır.

İşlemci sayısı artırıldığı oranda hafıza miktarı da orantılı olarak artacaktır.

İşlemciler her hangi bir ara yüz veya hafıza tutarlılık sistemleri kullanmadan kendi lokal hafızalarına ulaştıkları için bu işlem çok hızlı gerçekleşmektedir. Çok işlemcili tek bir bilgisayara göre maliyeti ve verimi daha düşüktür.

Dağıtılmış hafıza sisteminin bir takım dezavantajları da vardır. İşlemciler arasında veri haberleşmesinin tüm detayları tamamen programcının sorumluluğu altındadır. Dağıtılmış hafıza sistemlerinde program geliştirmek ve yönetmek oldukça zahmetli bir iştir ve hafızalar arası veri paylaşımı daha çok zaman almaktadır. Günümüzde dünyanın en gelişmiş ve hızlı bilgisayarları paylaşılmış ve dağıtılmış hafıza mimarilerini bir arada kullanmaktadır. Bu yapı şekil 2.20’de temsil edilmiştir.



Şekil 2.20: Çok İşlemcili Dağıtık Hafıza Modeli

Bu sistemde her bir hafıza kendi işlemcilerine hizmet ettiği gibi aynı zamanda bu hafıza bir network aracılığı ile üzerindeki bilgileri komşu bilgisayarın işlemcilerinin kullanmasına da açmaktadır. Bu sistem hem paylaşılan hafıza hem de dağıtılmış hafızanın bir arada kullanıldığı bir hibrid sistemdir. Böyle bir sistem hem paylaşılan hafıza sisteminin hem de dağıtılmış hafıza sisteminin avantaj ve dezavantajlarına sahiptir. Bu sistemlerin programlanması da bağımsız olarak her iki sistemin programlanmasında daha karmaşıktır.

2.8 Sinyal İşleme

2.8.1 Kimyadaki Sıralı Sinyaller

Sıralı sinyaller şaşırtıcı bir şekilde kimyada yaygındır ve çok fazla sayı içeren analiz metotları için gereklidir. Çoğu veri bilgisayarlı cihazlar aracılığı ile elde edilir (IR, NMR, UV gibi). Ham bilgilerden elde edilen, örneğin pikin yüksekliği, yayılması ve konumu, bilgisayardan bilgi ilk nasıl işlendiğini bağlıdır.

2.8.1.1 Kromotogramlar ve Spektrumlar

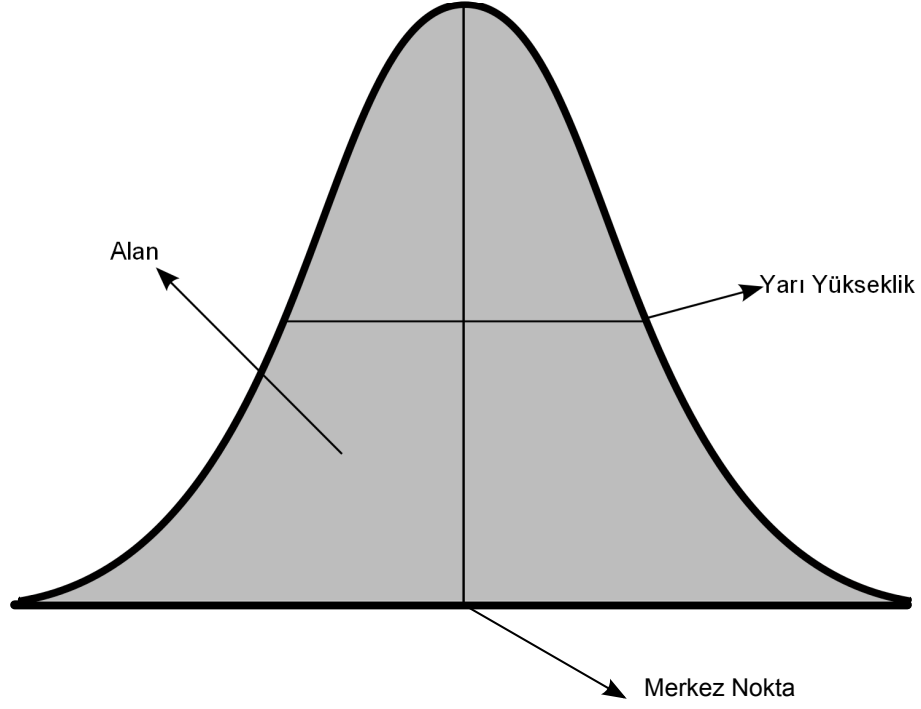
Kimyada kromatografi ve spektroskopide sinyal işleme sinyal toplama ve işleme önem arz etmektedir. Kromatogramın gürültü giderilmesi önemli bir işidir. Kromatogramdaki gürültü giderildiği zaman bazı pikler kaybolmakta ve çözünürlük düşmektedir. Bununla birlikte çözünürlüğün artırılması spektrogramun türev modunda alınması gürültü gidermeye yönelik hızlı hesaplamalı yöntemler gerekmektedir. Bunun dışında eğri uydurma ve Fourier filtrelemeye dayalı başka yaklaşımlara da oldukça sık rastlanmaktadır.

2.8.2 Sinyal İşleme Temel Bilgileri

2.8.2.1 Pik Şekilleri

Kromatogramlar ve spektrumlar normalde üzerine gürültü binmiş bir dizi sayılardan oluşmuştur. Her bir tepe ya karakteristik bir soğurmayı yada bir bileşikten kaynaklanmaktadır. Çoğu zaman altta kalan pik bulanıklık, gürültü veya komşu pik etkisi gibi nedenlerden dolayı bozulmaya maruz kalır. Kemometrik yöntemlerinin temel amacı, altta yatan bozulmamış, bilgi elde etmektir.

Pikleri karakterize etmenin bir çok yolu vardır. Ortak bir yaklaşım Şekil 2.21'de gösterilmiştir. Her pik için Şekil 2.21'de görüldüğü gibi bir pik merkezi, pik yarı yüksekliği, yarı genişliği ve pik alanı vardır.



Şekil 2.21: Genel Pik Parametreleri

Yüksekliğin deneysel ölçümü sıklıkla daha kolay olmasına rağmen; alan ve tepe şekli arasındaki ilişki pik şekline bağlıdır. Eğer tüm pikler aynı şekilde ise yüksekliğin çapı alanların oranı ile orantılıdır. Bununla beraber alan konsantrasyon gibi kimyasal özelliklerin belirlenmesinde daha faydalıdır.

Bazen yarı genişlikten daha ziyade, tepe yüksekliğin farklı yüzdelilerindeki genişlikler kullanılır. Piklerin genel yüksekliğinin, ki pikin toplam genişliği olarak alınır, küçük yüzdeleri halinde gruplandırılmaları veya gürültüyü belirten boyutlarda gruplandırılması bir diğer yaygın ölçüdür.

Spektroskopinin pek çok durumunda pik şekli çok hassas olarak tahmin edilebilir (NMR veya görünür spektroskopi gibi). Diğer durumlarda (kromatografi örneğinde olduğu gibi) pik şekli karmaşık fiziksel işlemlere bağlıdır ve sadece empirik olarak modellenenebilir. Sonraki durumda tam modelin eldesi pratik değildir çok sayıda empirik tahminlerde aynı derecede faydalı bilgiler verebilirler.

Pek çok durum üç pik şekli ile kapsanır.

2.8.2.2 Gaussians

Bu pik şekilleri kromatografi ve spektroskopinin bir çok türünde yaygındır. Basit bir Gauss denklemi Denklem 2.24'de gösterilmiştir.

$$x_i = A \exp[-(x_i - x_0)^2/s^2] \quad (2.24)$$

Burada A pik merkezi yüksekliğini, x_0 merkezin konumunu ve s genişliği ifade etmektedir. Gauss eğrileri normal dağılımı temel alırlar ve x_0 ölçüm serilerinin ortalamasına, $s/\sqrt{2}$ ise standart sapmaya karşılık gelir. Görülebileceği gibi, denklem 2.24 kullanılarak gauss pikinin yarı yüksekliğindeki genişlik $\Delta_{\frac{1}{2}} = 2s\sqrt{\ln 2}$ ve alan πAs olarak verilmiştir. Dikkat edilirse bu hem yüksekliğe hem de genişliğe bağlıdır.

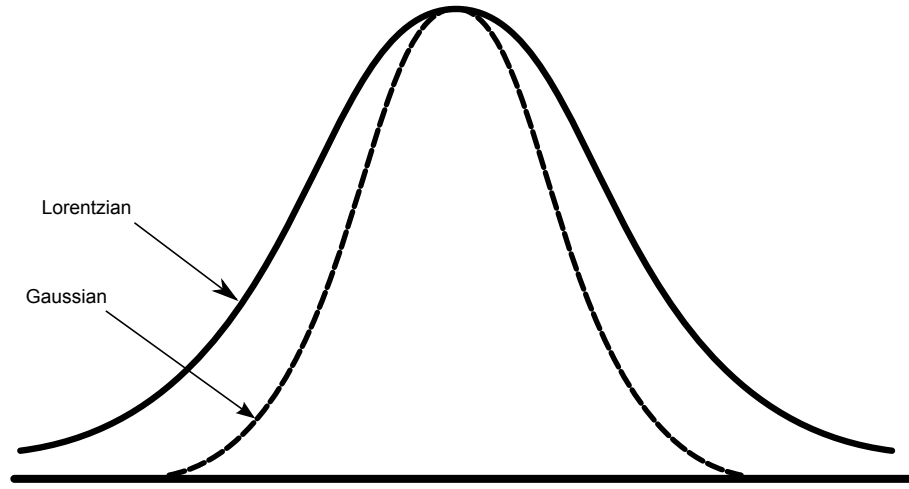
2.8.2.3 Lorentzians

Lorentz pik şekli Cauchy dağılımı denilen bir istatistik işlevine karşılık gelir. Daha az kullanılmasına karşın özellikle NMR spektroskopisinde kullanımı artmaktadır. Basitleştirilmiş Lorentzian Denklem 2.25'de gösterilmiştir.

$$x_i = A/[1 + (x_i - x_0)^2/s^2] \quad (2.25)$$

A merkez yüksekliği, x_0 merkez konumu ve s ise pik genişliğidir.

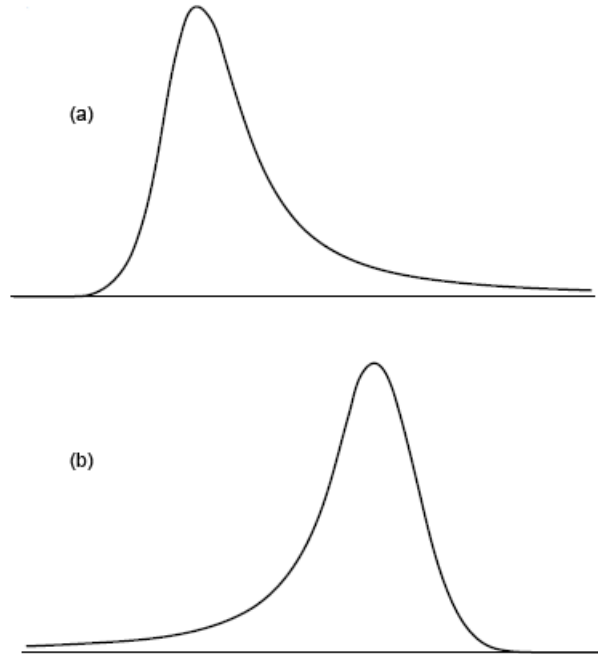
Görülebileceği gibi, Lorentzian pikin yarı yüksekliğindeki genişlik $\Delta_{\frac{1}{2}} = 2s$ ve alan πAs olarak verilmiştir. Dikkat edilirse bu hem yüksekliğe hem de genişliğe bağlıdır. Gauss ve Lorentzian pik şekilleri arasındaki temel fark, Lorentzian pikinin daha kuyruk genişliğinin daha geniş olmasıdır. İki pik şekli üst üste Şekil 2.22'de gösterilmiştir.



Şekil 2.22: Gaussian ve Lorentzian Pik Şekilleri

2.8.2.4 Asimetrik Pik Şekilleri

Kromatografinin bir çok formunda simetrik bir pik elde etmek zordur. Çok sayıda karmaşık model olmasına rağmen, ilk ve basit olarak kullanılan yaklaşım Lorentzian/Gaussian pik şeklidir. Şekil 2.23(a)'da sağ tarafın Lorentzian sol tarafınsa Gaussian ile modellendiği pik kuyrukları gösterilmektedir. Nadir rastlanan tersi durumsa Şekil 2.23(b)'de gösterilmiştir.



Şekil 2.23: Asimetrik Pikler. (a) Gaussian/Lorentzian (b) Lorentzian/Gaussian

2.8.2.5 Tepe Şekli Bilgilerinin Kullanımı

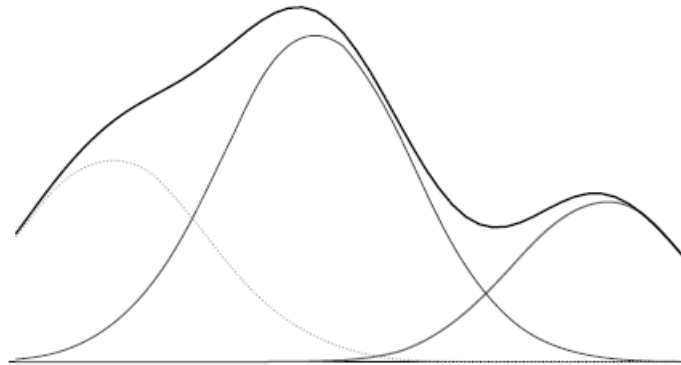
Tepe şekli bilgileri iki ana şekilde kullanılabilir.

1. Genelde kullanılan eğri uydurmadır. Çok sayıda hesaplama algoritması bulunmaktadır, bir tür küçük kareler minimizasyonu pek çoğunda içermektedir.

Eğer Gaussian şeklinde bir kümede üç pik biliniyor veya şüpheleniliyorsa sonrasında dokuz parametrenin bulunması gereklidir. Bunlar üç tepe pozisyonu, tepe genişlikleri ve yükseklikleridir. Herhangi bir eğri uydurmada, özellik ve tepe şekilleri hakkında kesin bilgilerin olup olmadığının belirlenmesi önemlidir (örneğin her bir bileşenin pozisyonları gibi). Pek çok kimyasal veride yeterli kalitenin olmadığı bilinmesi de önemlidir. Kromatografide normalde empirik yaklaşımlar genelde yeterlidir: aşırı modelleme tehlikeli olabilir. Eğri uydurmanın sonuçları sistemi tanımlamak için daha iyi olabilir; örneğin pik alanların bilinmesiyle karışımdaki bileşenlerin göreceli konsantrasyonlarının belirlenmesi mümkündür.

2. Kemometride simülasyonlarda önemli rol oynarlar. Bu tür simülasyonlar sistemi anlamaya çalışmanın bir yolu olabilirler. Eğer kemometrik yöntemin sonuçları gerçek verilere yakın piklerin yeniden belirlenmesi ile sonuçlanırsa yeniden belirlenen pik şekilleri iyi bir tanımlama sağlar.

Tipik bir kromatogram veya spektrum bazen farklı şekil ve yoğunlukta vede pozisyonlarda birkaç pikten oluşur. Şekil 2.24'de toplam yoğunlukları ile beraber üç pikin kümelenmesini göstermektedir.



Şekil 2.24: Üç Pikin Kümelenmesi

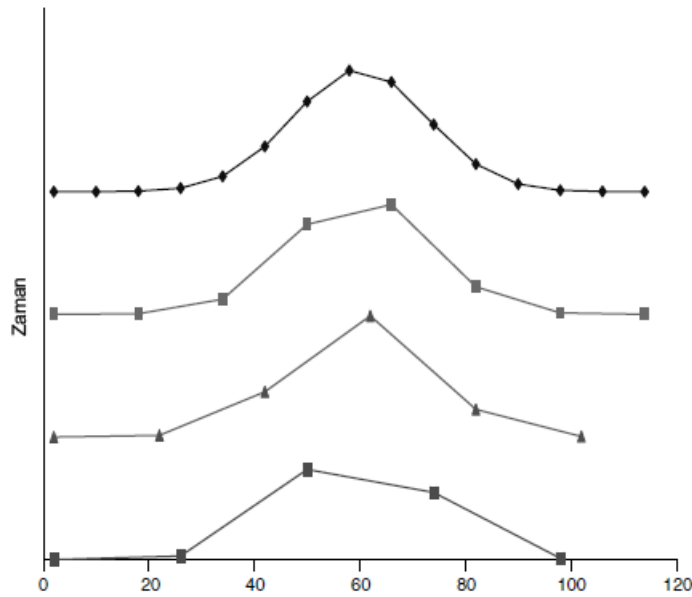
Görüldüğü gibi sağ taraftaki pik çiftinin görsel olarak çözülmesi kolaydır,

sol taraf için aynı şey söylenemez, çeşitli veri analiz yöntemleri kullanmaksızın kümelerin ilk tepe noktalarının yoğunluklarını ve pozisyonlarının belirlenmesi özellikle zordur.

2.8.3 Sayısallaştırma

Neredeyse tüm modern laboratuvarlarda temel veriler analog formdan ziyade bilgisayar yardımı ile toplanmaktadır. Dijital çözünürlüğün tepe çözümü nasıl etkilediğinin anlaşılması önemlidir.

Pek çok bilgi kayıt tekniğinin her bir pik için sadece sınırlı sayıda veri noktaları verebilmektedir. Özellikle iyi çözünürlükte enstrümentasyonla tipik bir NMR piki yarı genişlikte yeterince hertzde 8192 verinin toplanması gerekmektedir. Kötü dijital çözünürlük bilgi elde edilebilirliğini etkiler. Bu hataların ne kadar ciddi olduğunun tespiti faydalıdır. Şekil 2.25'e bakıldığında veri sayısının (çözünürlüğün) ne kadar önemli olduğu daha iyi anlaşılır.



Şekil 2.25: Veri Sayısı ve Grafik

2.8.4 Gürültü

Analitik sinyallerdeki düzensizlikler gürültü olarak tarif edilir. Temel istatistikte gürültünün kaynağı ve doğası genellikle bilinmez ve normal dağılıma

uyduğu kabul edilir. İstatistikte t-testi ve F-testi gibi çoğu istatistik testlerinde gürültünün olmadığı kabul edilir.

Aletli ölçümlerde gözlenen sinyal;

$$x = \tilde{x} + e \quad (2.26)$$

Burada $\tilde{x}$ gerçek yada ideal sinyal, e ise gürültü fonksiyonudur. İki tip ölçüm gürültüsü vardır.

2.8.4.1 Durağan (Sabit) Gürültü

Herhangi bir andaki gürültü değerinin önceki gürültü değeri ile bağlantısının olmadığı durumdur. İki önemli durağan gürültü vardır.

1. Eş değişkenli Gürültü: Gürültünün bir ortalama ve standart sapma ile verilebileceği durumdur. Genelde normal dağılım olarak kabul edilir. Enstürümantel gürültünün çok çeşitli kaynakları vardır. Fakat bunların bileşkesi normal dağılıma uyduğu kabul edilir.

2. Değişken Varyanslı Gürültü: Bu tip gürültü sinyal büyüklüğüne bağlıdır. Normal dağılıma uyabilir ancak dağılımın standart sapması sinyalin büyüklüğüne bağlıdır. Değişken varyanslı gürültü UV/vis veya IR de geçirgenlikten absorbansa logaritmik geçiş sırasında sinyal yüksekliğinden etkilenmeyerek sabit eklenmiş olur.

2.8.4.2 Bağlantılı Gürültü

Herhangi bir adımdaki gürültünün düzeyi bir önceki gürültü seviyesi ile ilintilidir. Örneğin bir tartım sırasında gelen hata ölçüm sırasındaki gürültü düzeyini etkileyecektir.

2.8.4.3 Sinyalin Gürültüye Oranı

Sinyalin gürültüye oranı ölçümlerde kullanılan önemli bir parametredir. Oranın yüksek oluşu sinyaldeki gürültü seviyesinin arttığını gösterir. Bu değer

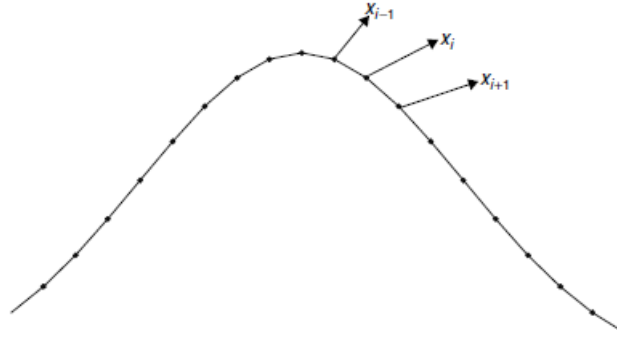
emprik bir deęerdir ve ölçülen en yüksek sinyalin gürültünün karekök ortalama deęerine eşittir. Bu oran ölçüme etki eden gürültü türlerine baęlı olarak deęişir. Çoęu durumda bu oranın %5 düzeyinde kalması istenir.

2.8.5 Düzgünleştirme/Yumuşatma Fonksiyonları

Ana amaç gürültü uzaklaştırıldıktan sonra mümkün olduğunca çok bilgi taşıyan sinyal elde etmektir. Veriler, zaman veya frekans gibi sıralı olarak elde edildiğinde alttaki sinyal tezimizin kısım 2.8.2.1’de anlatıldığı gibi düz ve monotonik fonksiyonların toplamından kaynaklanır. Ancak alttaki gürültü ise ilişkilendirilemeyen fonksiyonlardır. Sinyal eldesinde kullanılan önemli yöntemler verilerin düzleştirme edilmesi esasına dayanmaktadır. Prensip yumuşatma yöntemler kullanılarak sinyal aynı kalırken gürültünün uzaklaştırılmasıdır. Bu yaklaşım bir kaç veri noktasının yarı genişliğine sahip piklere baęlıdır. Dijital çözünürlük çok kötü ise sinyaller sivri uçlar olarak görünür bunlarda gürültü ile karıştırılabilir. Herhangi özel bir uygulama için optimum filtreleme yöntemi seçimi önemlidir. Çok fazla düzleştirme ve sinyalin kendisi yoğunluk ve çözünürlük olarak indirgenir. Çok az düzleştirmede ise gürültü aynı kalmaktadır. Optimum düzleştirme fonksiyonu gürültü karakteristiğine ve tepe genişliğine baęlıdır.

2.8.5.1 Hareketli ortalamalar

Hareketli ortalama kavramsal olarak, en basit yöntemler doğrusal filtrelemelerdir. Burada sonuç düzleştirilmiş veri, ham verinin doğrusal fonksiyonudur. Normalde bu işlem etraftaki noktaların kullanımını içerir. Örnek olarak Şekil 2.26’daki 3 noktanın fonksiyonunu kullanarak i deki deęerin hesaplanması verilebilir.

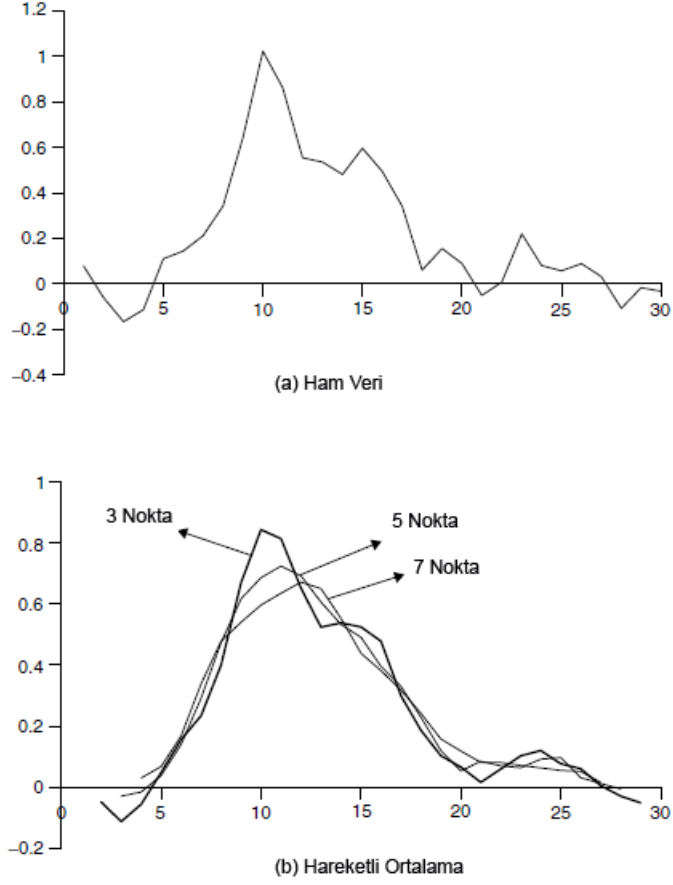


Şekil 2.26: Hareketli Ortalama Filtresi

Matematiksel olarak bu fonksiyonlar şu şekilde açıklanabilir.

$$x_{i,yeni} = \sum_{j=-p}^p c_j x_{i+j} \quad (2.27)$$

En basitlerden birisi üç noktalı hareketli ortalamadır. Her bir noktanın kendisi, kendisi önündeki ve arkasındaki noktaların ortalaması ile yer değiştirir. Yani üstteki eşitlikte her bir nokta için $p = 1$ ve $c_j = 1/3$ olmaktadır. Filtre ($p = 2$, $c_j = 1/5$) beş veya yedi noktaya uzatılabilir. Filtredeki nokta sayısı arttıkça sinyal indirgenmesi artarken sinyali bulanıklaştırma ihtimalide yükselmektedir. Filtredeki noktaların sayısı pencere olarak isimlendirilmektedir. Filtre zaman serileri veya spektrum boyunca ilerler, her bir veri noktası sıralı olarak ilgili filtrelenmiş veri noktası ile yer değiştirir. Optimum filtreleme gürültü dağılımı ve sinyal genişliğine bağlıdır. En iyi yol çok sayıda farklı farklı filtre genişliklerinin denenmesidir.



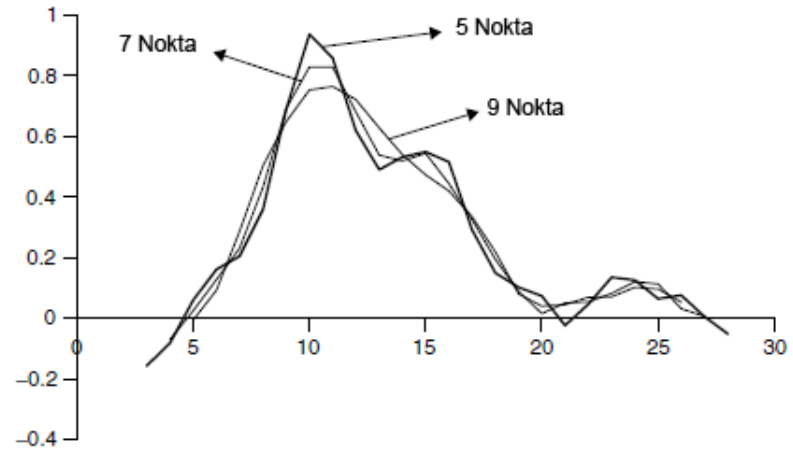
Şekil 2.27: Hareketli ortalama 3,5 ve 7 Noktalı Filtre

2.8.5.2 Savitsky-Golay Filtreleri

Hareketli ortalama filtrelerinin dezavantajı doğrusal yaklaşım kullanmalarıdır. Ancak genelde piklere en iyi yaklaşım polynomial gibi eğrisel yaklaşımlardır. Bu yaklaşım özellikle pik merkezi gibi doğrusal yaklaşımın daha düşük değerler verdiği yerler için daha doğrudur. Hareketli ortalama yaklaşımının prensibi temel olarak kullanılabilir. Örneğin yedi noktalı kübik filtre için kullanılan formül aşağıdadır;

$$\hat{x}_i = b_0 + b_1i + b_2i^2 + b_3i^3 \quad (2.28)$$

Pencerde nokta nokta hareket ederek her bir nokta hesaplanan değerle yer değiştirir. Ancak regresyon hesapsal yoğunluk gerektirir. Yüzlerce veri noktasından oluşan spektrum veya kromotagrom için çok fazla zaman ve emek gerekir.



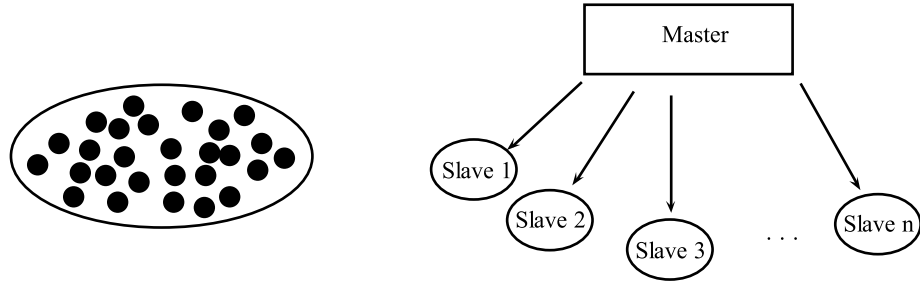
Şekil 2.28: Savitsky-Golay 5,7 ve 9 Noktalı Filtre

3. MATERİYAL ve YÖNTEM

3.1 Paralel Genetik Algoritma

Paralel Genetik Algoritmalar (PGA), paralel ve dağıtılmış programların altında yatan temel fikir bir görevi parçalara ayırmak ve birden fazla işlemci kullanarak eş zamanlı çözmektir. Bu böl ve yönet yaklaşımı değişik şekillerde kullanılabilir ve bazıları genetik algoritmanın davranışını değiştiren bazıları ise değiştirmeyen farklı paralelleştirme metotlarına yol açar. Bazı metotlar (örneğin iyi parçalanmış PGA gibi) paralel bilgisayar mimarisini büyük ölçüde kullanırken diğer metodlar (kötü bölünmüş PGA gibi) daha az sayıda ve güçlü işleme elemanı içeren çoklu bilgisayarlar için uygundur.

Kaba bir sınıflandırmayla genetik algoritma kitaplarında anlatılan paralel genetik algoritma kavramları global paralelleştirme, iri parçalı paralelleştirme ve küçük parçalı paralelleştirme genetik algoritmaları olarak sınıflandırılabilir [101]. Pratik uygulamalar için en popüler model, tezimizde geliştirilen algortmada da kullanılan ada modeli olarak da bilinen iri taneli modeldir.



Şekil 3.1: Global paralelleştirme kavramları: Eşleşmenin rastgele olduğu (Panmiktik) popülasyon solda. Master-Slave yapısı sağda.

3.1.1 Global Paralelleştirme

Ardışık genetik algoritmaya benzer şekilde global paralelleştirmede de sadece bir tek panmiktik popülasyon vardır. Seçim tüm bireyleri ele alarak yapılır.

Algoritmanın davranışı değişmeden kalır ve global genetik algoritma ardışık genetik algoritma ile aynı özelliklere sahiptir. Paralleleştirilmiş en yaygın operasyon, bir bireyin popülasyonun geri kalanından bağımsız olarak uygunluğunun ölçülmesi ve bireylerin buna bağlı olarak değerlendirilmesidir. Bu nedenle bu aşamada gerekli olan tek iletişim, iş yükünün dağıtımı ve toplanmasıdır.

Bir ana düğüm genetik algoritmayı çalıştırır ve uygunluğun değerlendirilmesi çeşitli köle (slave) işlemcilerin arasında paylaştırılır. Popülasyonun parçaları uygun olan işlemcilere atanır. Burada ele aldıkları birey alt kümesinin uygunluk değerlerini hesaplar ve yönetici (master) işlemciye geri gönderir. Merkezi ve hiyerarşik iletişim düzeninden dolayı global paralel genetik algoritmalar tek popülasyonlu yönetici-köle (master-slave) genetik algoritmaları olarak da bilinir.

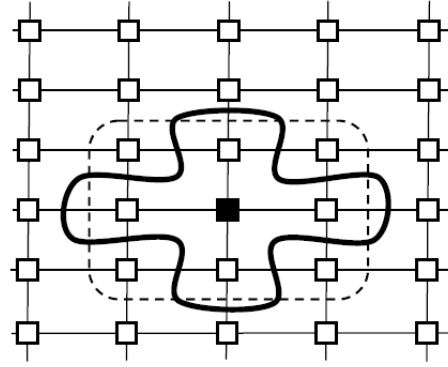
Şekil 3.1, bir yönetici-köle yapıları genetik algoritmanın popülasyon yapısını gösterir. Bu panmiktik genetik algoritmanın tüm bireylerini (siyah nokta ile temsil edilen) aynı popülasyonun içinde bulunur. Yönetici işlemci popülasyonu depolar, genetik algoritma operasyonlarını çalıştırır ve bireyleri köle işlemcilere dağıtır. Köle işlemciler bireylerin uygunluğunu hesaplar. Sonuç olarak global paralelleştirme sadece uyum fonksiyonunun değerlendirilmesi, çalışma süresi tüketimi anlamında bir sorun ise etkilidir.

Global paralelleştirilmiş genetik algoritmalar eğer değerlendirme yönetici işlemci tarafından yürütülen operasyonlar için gereken çabadan daha büyük bir hesaplama çabası gerektiriyorsa uygulama açısından son derece kolay ve etkili bir paralelleştirme yöntemidir.

3.1.2 Küçük Parçalı Paralel Genetik Algoritmalar

İnce taneli model olarak da bilinen bu model çok küçük yalıtılmış grupların büyük sayılarda ele alınması ile oluşur. Şekil 3.2, ince taneli PGA'yı gösterir. Bu PGA sınıfı bir tane uzaysal dağılmış popülasyon içerir. Bu popülasyon paralel bilgisayarlar için son derece uygundur. Ancak diğer süper bilgisayar mimarileri ile de incelenebilir. Tipik bir örnek [102] içsel bir paralel algoritma modeli olan

difüzyon modelidir.

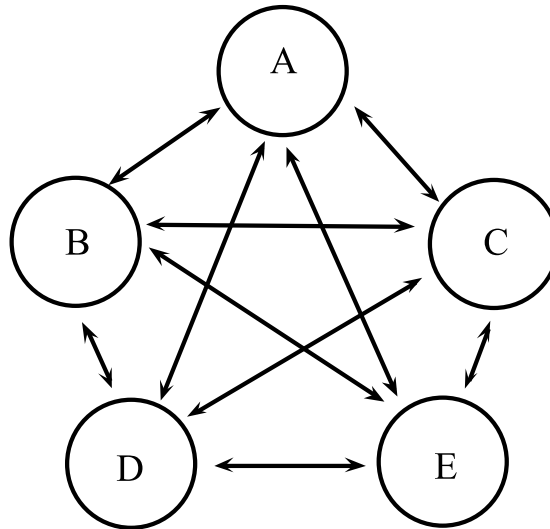


Şekil 3.2: İnce Taneli Genetik Algoritmanın Popülasyon Yapısı; Hücresel Modelin Özel Bir Durumu Gösterilmiştir.

Bu modelin ana fikri bireylerin global popülasyonda difüzyon sürecindeki moleküller gibi dağılmış olmasıdır. Difüzyon modelleri ayrıca hücresel modeller olarak da bilinir. Difüzyon modelinde her bir bireye bir işlemci atanır. Yeniden birleştirme işi ise her bir bireyin yerel komşuluğunda yapılmaya zorlanmasıyla oluşur.

3.1.3 Küçük Parçalı Paralel Genetik Algoritmalar-Ada (Island) Modeli

Aynı anda yaşayan popülasyonlarda genetik işlemler ile elde edilen en iyi bireylerin diğer popülasyonlara göç etme işlemleri ile sürecin ilerleme işlemine genetik algoritmada ada modeli denilmektedir. Ada Modeli problemin paralelleştirilmesine uygun bir algoritmadır.



Şekil 3.3: Genetik Algoritma-Ada Modeli

Ada Modeli Şekil 3.3’de gösterilmiştir. Bu modelde harf ile simgelendirilmiş bölgeler kendi halinde bir popülasyon olup genetik işlemleri yine kendi içinde yapmaktadırlar. Ancak buldukları iyi sonuçları birbiri ile paylaşmaktadırlar. Aslında her bir ada’da çalışan algoritma bir önceki bölümde anlatılan genetik işlemler algoritması ile tamamen aynıdır.

Bu model birden fazla popülasyon kümesi ile ilerlediğinden normal seri bir algoritma ile yazılmış bir koddan daha iyi sonuçlar vermektedir. Ayrıca ada modeli popülasyon çeşitliliğini arttırdığı için erken yakınsama veya yanlış yönde ilerleme gibi sorunların çözümünde de etkilidir.

Genetik işlemler sonucu bir adanın bulduğu en iyi sonucun diğer adalara göç etmesi gerekmektedir. Bunun için bu sonuçların gönderilmesi ve diğer adaların bu sonucu alabilmesi için bir fonksiyon hazırlanması gerekmektedir. Tezimizde bu fonksiyonlar için MPI fonksiyonları kullanılmıştır.

Ada modeli algoritması genetik algoritmaya (Algoritma 2.1) ek olarak aşağıdaki sözde kodların eklenmesi gerekmektedir.

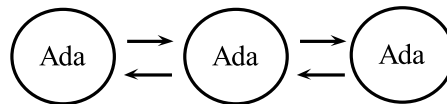
Algoritma 3.1 Ada Modeli Sözde Kod

Eğer(Bu nesilde göç gerekiyorsa)

En uygun bireyleri (T) tüm adalara gönder;
 Diğer adalardan gelen uygun bireyleri (T) al;
 Popülasyonu tekrar değerlendir;

Algoritma 3.1’de adalarda oluşan en iyi bireyler, belirli bir aralıkla ya da belirli bir şart sağlandığında diğer adalara gönderilir. Algoritma tüm adalardan tüm adalara transfer yöntemi ile yapılmaktadır.

Eğer yeni bireyler sadece komşu adalara gönderiliyorsa bu modele basamak modeli denir. Bu modelde ada modeline göre adalar arası haberleşme daha azdır. Ancak sonuca ulaşmak daha fazla vakit alır.



Şekil 3.4: Basamak Ada Modeli

Oluşturduğumuz bu adalarda eğer ki popülasyon büyüklüğü sabit ise diğer adalardan gelen en iyi bireylerin popülasyona nasıl katılacağıdır. Yeni birey popülasyonda rastgele ya da önceden belirlenmiş (uygunlukları en kötü bireyler gibi) bir şekilde kötü bireyin üzerine yerleştirilir. Üzerine yerleştirilen birey kaybolur. İster yeni birey var olanın üzerine yazılsın isterse popülasyon büyüklüğünü arttıracak şekilde popülasyona ekleme yapılarak katılsın, her iki durumda da yeni gelen bireylerin mevcut popülasyona göre uygunluk değerlerinin belirlenmesi için tekrardan bir elemeye tabi tutulmaları gerekmektedir.

Adaların bulduğu en uygun birey her bir nesil çeviriminde göç ettirmek zorunda değildir. Bu işlem için bir parametre algoritmaya eklenir. Her nesilde göç ettirme adalardaki genetik çeşitliliği arttıracığından sonuca ulaşma daha kesin olacaktır. Ancak her birey göçünde tekrar uygunluk değerinin hesaplanması işlem yükü getirecektir.

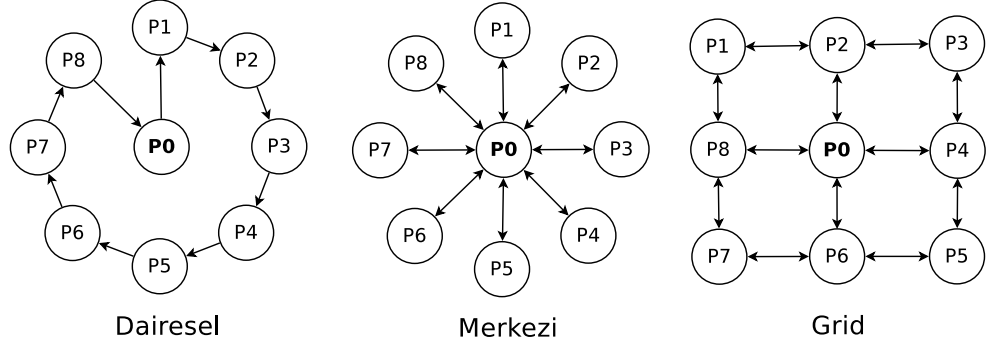
Ada algoritması ile hazırlanan bir programda her bir ada bir işlemciye atanır ve oradaki uygun bireylerin diğer adalara aktarılması bir ağ üzerinden oluşur. Haberleşmenin gecikmemesi için uygun bir band genişliği olan bir ağ yapısı gerekmektedir. Ağ yapısındaki gecikmeler adalardaki yeni neslin oluşumunu da geciktirecektir. Bu da istenmeyen bir durumdur.

Genetik algoritma her bir bireyi bir uygunluk fonksiyonu ile değerlendirir. Seri bir genetik algoritma uygulamasında birey sayısını arttırmak işlem süresini de doğrudan arttıracaktır.

Genetik algoritma belli bir birey sayısı ile belirlenmiş bir popülasyon ile taranacak nokta sayısı her bir nesil için yine birey sayısı kadardır. Birey sayısını sabit tutup işlemci sayısı kadar popülasyonlar oluşturmak, aynı anda taranan nokta sayısını arttıracaktır. Bu, taramada gerçek noktayı bulma olasılığını daha da arttıracaktır.

Ada modelinin kullanılmasının temel amacı;

- Evrenin bir noktasında popülasyon oluşturmak yerine, evrenin değişik bölgelerinde birden fazla popülasyon oluşturarak bir bölgede gerçek değeri



Şekil 3.5: Program Aşamasında Denenen Ada Modelleri

aramak yerine birden fazla bölgede gerçek değeri aramak

- En uygun değere daha hızlı ulaşmayı sağlamak

3.2 Geliştirilen Yazılımın Akış Şeması

Programın çalışma aşamaları genetik algoritma ve aşağıdaki aşamaları içerir.

Adım 1: 10-150 popülasyon arasındaki bir sayıda N tane bireyin rastgele oluşturulması.

Adım 2: Bireylerin pik parametrelerinin $(\sigma_{n_p A}, \sigma_{N_p v}, \sigma_{N_p h}, \sigma_{N_p m})$ her birinin gaussian dağılımının hesaplanması.

Adım 3: Bulunan en iyi sonucun komşu adalara aktarılması.

Adım 4: Gaussian dağılımı kullanılarak herbir pikin parametrelerinin yeni arama uzayının belirlenmesi

$$Min, Max_{N_p A, v, h, m} = Best_{N_p A, v, h, m} \pm \sigma_{N_p A, v, h, m} \times f \quad (3.1)$$

Buradaki f arama uzayı faktörüdür.

Adım 5: Arama uzayında yeni bir rastgele birey oluşturulması.

Adım 6: Yeni birey için RMSE değerinin hesaplanması.

Adım 7: Yeni birey eğer grubun en kötüsünden daha iyi ise bireyi kabul et, en kötüsünü at, yeni bireyin değerlerini istatistiklere ekle kötününkini çıkar ve 2.

adıma git.

Adım 8: Durdurma kriteri sağlanmışsa adım 10'a git.

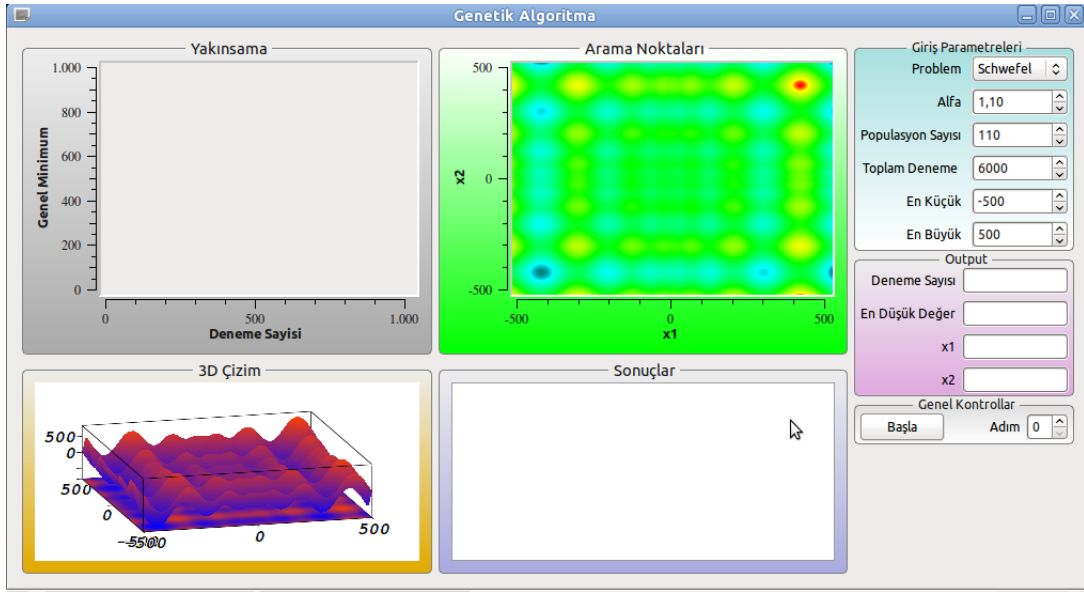
Adım 9: Adım 4'e git.

Adım 10: En iyi popülasyon sonucunu yazdır.

3.3 Genetik Algoritma İle Geliştirilen Yazılımın Testi

Geliştirdiğimiz bilgisayar programının doğru sonuç verip vermediğini anlamak için teste tabi tutmak gerekir. Bu yüzden literatürde sıklıkla kullanılan ve sonucu daha önceden bilinen fonksiyonlar kullandık [95, 96, 97, 98]. Fonksiyonları seçerken tek tepeli/çukurlu (unimodal) ve çok tepeli/çukurlu (multi modal) fonksiyonlar seçildi.

Hazırladığımız program Linux altında hazırlanmış olup kaynak kodu tezimizin ekinde verilmiştir. Programımızın görüntüsü Şekil 3.6'deki gibidir.



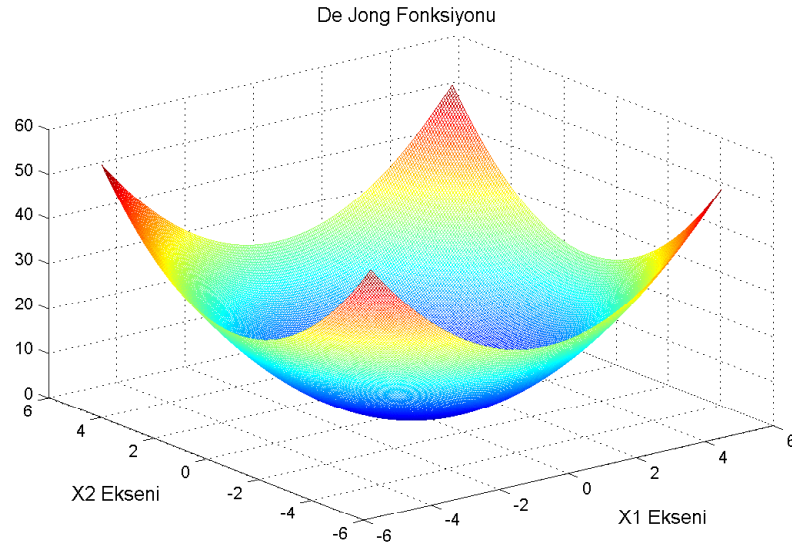
Şekil 3.6: Genetik Algoritma Bilgisayar Programı Ara Yüzü

Genetik işlemler sırasında çaprazlama ve mutasyon oranları programımızda alfa isimli yeni bir terim içinde birlikte kullanılmaktadır.

3.3.1 De Jong Fonksiyonu

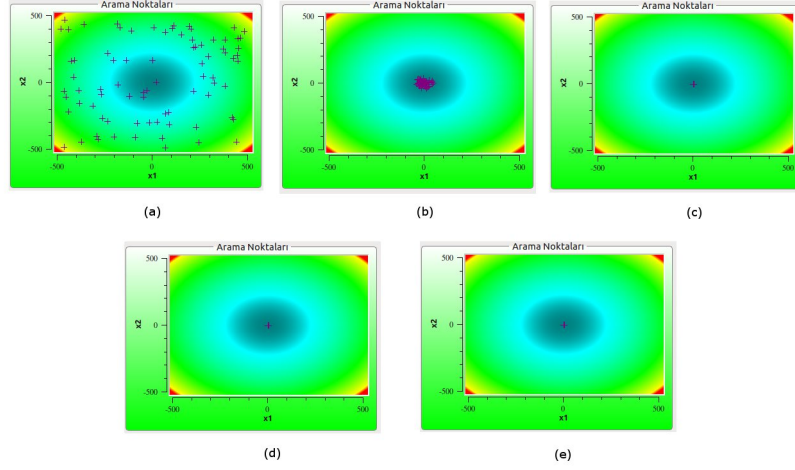
İlk fonksiyonumuz sürekli bir fonksiyon olan tek tepeli/çukurlu, dış bükey ve özellikle performans testlerinde kullanılan De Jong fonksiyonudur. Fonksiyon Denklem 3.2’de gösterilmiştir.

$$f(x) = \sum_{i=1}^n x_i^2 \quad (3.2)$$



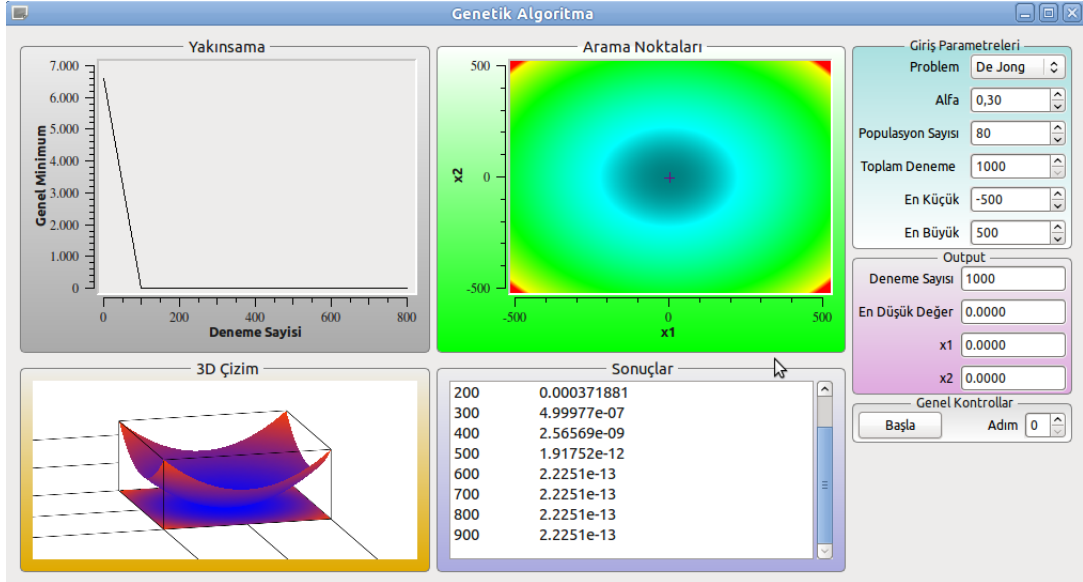
Şekil 3.7: 2D De Jong Fonksiyonu, $f(x) = x^2 + y^2$

De Jong fonksiyonunun tarama uzayı $-5,12 \leq x_i \leq 5,12$ arasındadır. Burada $i = 1, \dots, n$ aralığındadır. Fonksiyonun en iyi noktası $f(x) = 0$ noktasıdır. 2-D De Jong fonksiyonu Şekil 3.7’de görüldüğü gibidir.



Şekil 3.8: De Jong (a) 100. Jenerasyon, (b) 200. Jenerasyon, (c) 300. Jenerasyon, (d) 400. Jenerasyon, (e) 500. Jenerasyon

De Jong fonksiyonu unimodal ve doğrusal bir fonksiyon olduğundan çözümü kolay bir fonksiyondur. Şekil 3.8 bakıldığında 300 iterasyonda minimum $f(x) = 0$ noktasının bulunduğu görülmektedir. Program Şekil 3.9'daki gibi kendine verilen iterasyon sayısı kadar genetik işlemleri devam ettirmektedir.



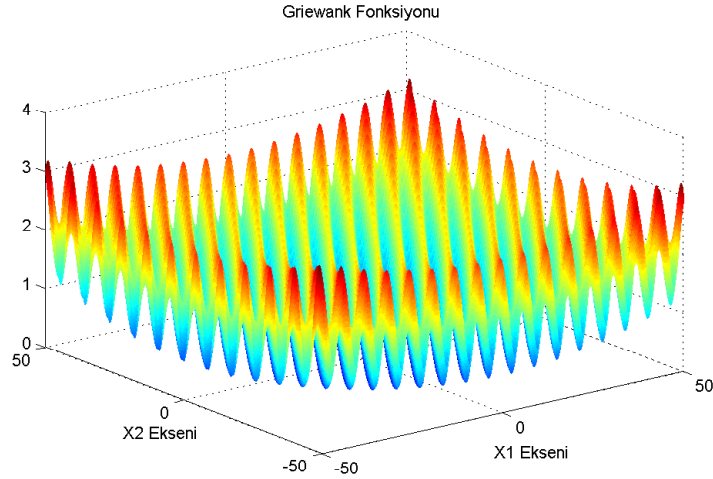
Şekil 3.9: Do Jong Hesaplama

3.3.2 Griewank Fonksiyonu

Griewank fonksiyonu Rastrigin fonksiyonu ile benzerlik göstermektedir. Şekil 3.10'na bakıldığında fonksiyonda çok fazla dağınık lokal minimum noktaları

olduğu görülür. Griewangk fonksiyonu doğrusal olmayan ve oldukça multimodal bir yapıya sahiptir. Griewangk fonksiyonu denklem 3.3'deki gibi tanımlanmıştır.

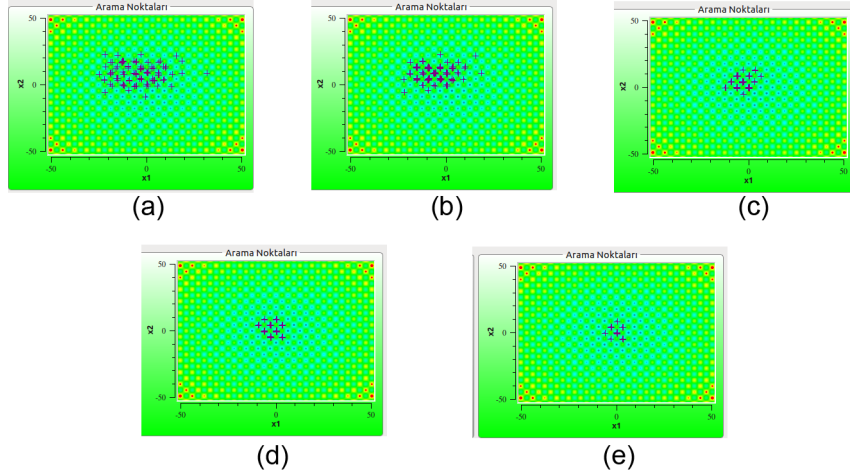
$$f(x) = \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (3.3)$$



Şekil 3.10: 2D-Griewangk Fonksiyonu

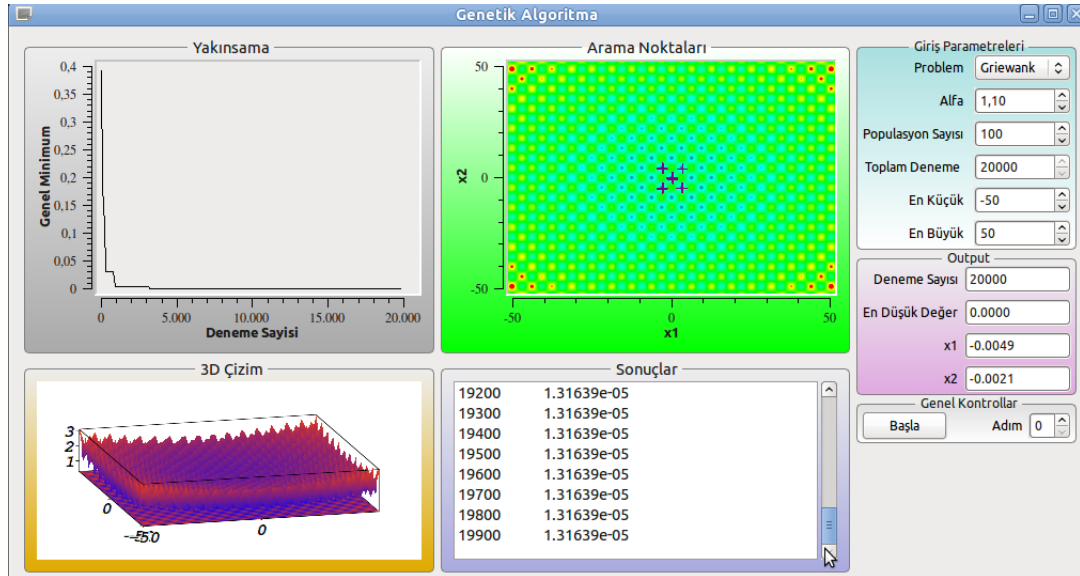
Griewangk fonksiyonu test alanı literatürde farklı değerler göstermektedir. Bizim programımızda $-50 < x_i < 50$ ($i = 1...n$) arasında en iyi noktayı arayacağız. Denklemin global en iyi noktası $f(x) = 0$, $x_i = 0$ ($i = 1...n$) ile tanımlanmaktadır.

Griewangk fonksiyonunda, toplama terimi fonksiyona parabollik özelliği kazandırmaktadır. Bu fonksiyonda yerel minimum derecesi parabollik derecesinden daha üst seviyededir. Çarpım terimi baz alınarak arama uzayının boyutları artırılmakta ve yerel minimumlar azaltılmaktadır.



Şekil 3.11: (a) 1000 Jenerasyon (b) 2000 Jenerasyon (c) 5000 Jenerasyon (d) 10000 Jenerasyon (e) 15000 Jenerasyon

Program, Şekil 3.11’de görüldüğü gibi 15000. jenerasyonda sonuca ulaşabilmiş ve kendine verilen iterasyon sayısına (20000) ulaşınca kadar devam etmiştir.



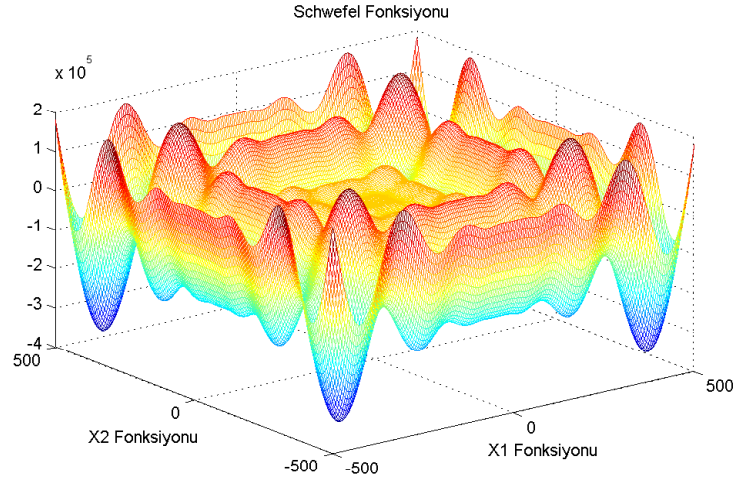
Şekil 3.12: Griewank Hesaplama

3.3.3 Schwefel Fonksiyonu

Schwefel fonksiyonu, Rastrigin ve Griewank fonksiyonları gibi doğrusal olmayan multimodal fonksiyon türlerinin tipik örnektir. Schwefel fonksiyonu Rastrigin fonksiyonundan daha kolay olan ve genel minimumun çok uzağında ikinci

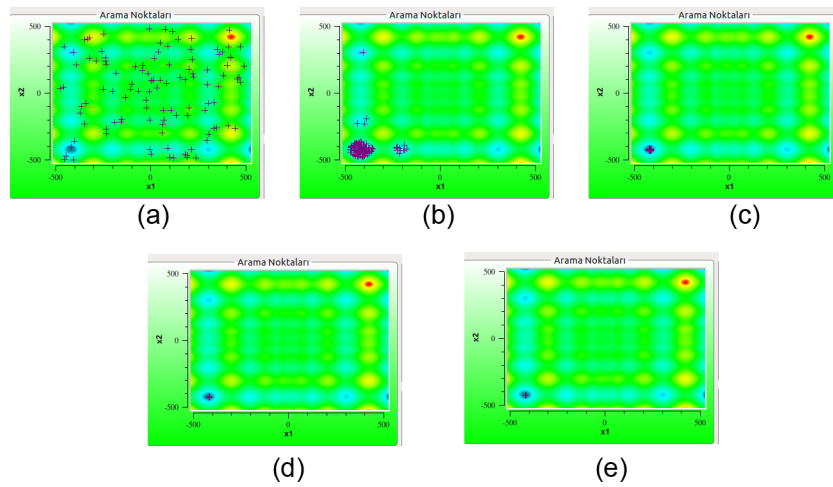
en iyi minimuma sahip olan bir fonksiyondur. Fonksiyon Denklem 3.4'daki gibi tanımlanmıştır.

$$f(x) = 418,9829n + \sum_{i=1}^n x_i * \sin\sqrt{|x_i|} \quad (3.4)$$



Şekil 3.13: 2D-Schwefel Fonksiyonu $f(x) = 418,9829n + \sum_{i=1}^n x_i * \sin\sqrt{|x_i|}$

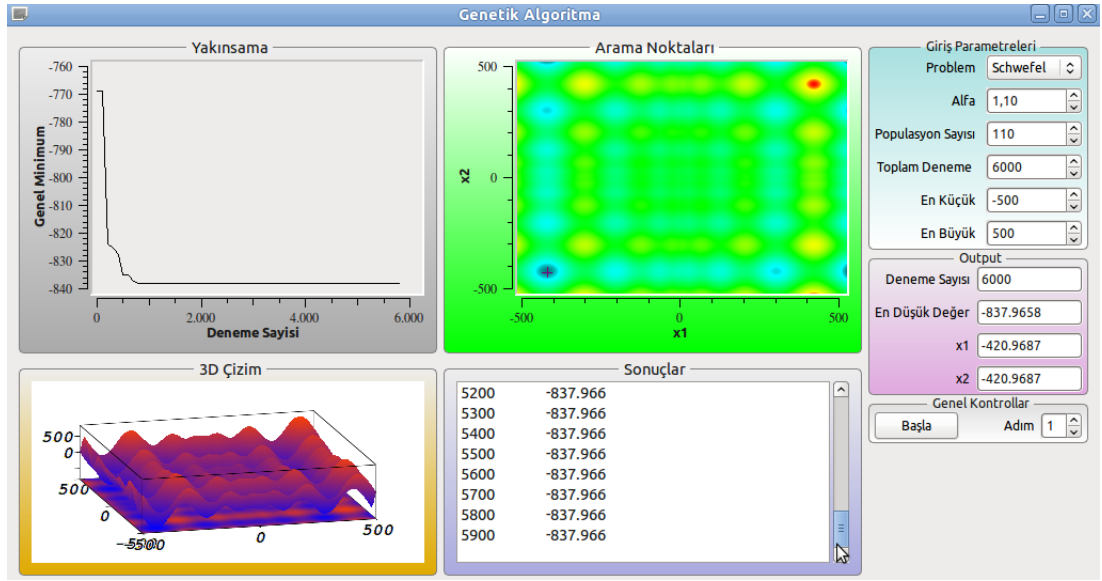
Schwefel fonksiyonu test alanı literatürde farklı değerler göstermektedir. Bizim programımızda hiperkübün $-500 < x_i < 500$ ($i = 1...n$) arasında en iyi noktayı arayacağız. Denklemin global en iyi noktası $f(x) = 0$, $x_i = 420,96870$ ($i = 1...n$) noktası ile tanımlanmaktadır.



Şekil 3.14: Schwefel İterasyon (a)100 Jenerasyon (b)500 Jenerasyon (c)900 Jenerasyon (d)1000 Jenerasyon (e)1300 Jenerasyon

Schwefel fonksiyonu parametre boşluğu olan geometrik bir uzaklığa sahip olduğundan bir sonraki en iyi yerel minimum takılma olasılığı var olduğu için yanıltıcı bir fonksiyondur. Dolayısı ile araştırma algoritmasını yanlış yöne doğru götürmeye potansiyel olarak yatkındır.

Geliştirdiğimiz bilgisayar programında bu yanıltıcılık göz önüne alınmıştır. Şekil 3.14'e bakıldığında yazılım 1300 civarı bir iterasyonda sonucu bulmuş ve bizim girdiğimiz deneme sayısına kadar (Şekil 3.15) devam etmiştir.



Şekil 3.15: Schwefel Hesaplama

3.3.4 Test Fonksiyonlarının Paralel ve Paralel Olmayan Ortamda Ada Modeline Uygulanarak Karşılaştırılması

İşlemciler arası haberleşme 200 işlem basamağı olduğu durumda, 80 popülasyon sayısı ve alfa değeri 0,6 optimum olduğu durumda test fonksiyonlar 100 defa paralel ve paralel olmayan ortamda çalıştırılmıştır. Test fonksiyonlarının global minimum noktaları 10000 iterasyon sonucunda Tablo 3.1 elde edilmiştir.

Tablodan da açıkça görülmektedir ki, paralel ada modelleri paralel olmayan hesaplamalardan çok daha iyidir.

Tablo 3.1: Test Fonksiyonlarının 100 Çalışma Sonrasında Paralel ve Paralel Olmayan Sistemlerde Global Minimum Noktasını Bulma Oranı

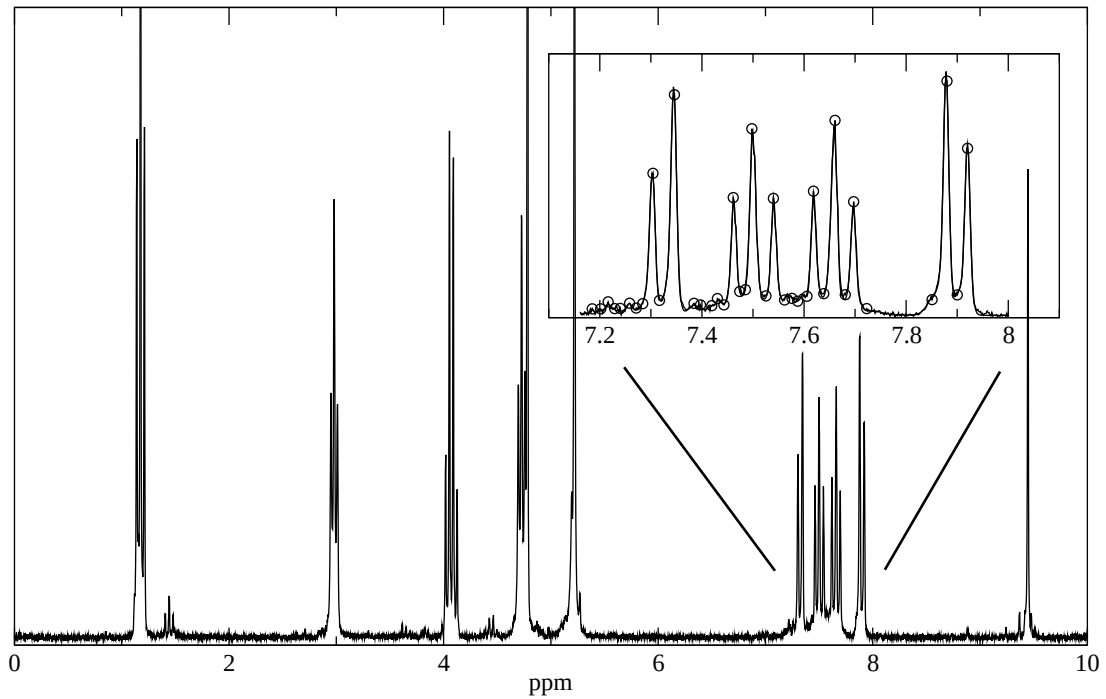
Fonksiyon	Paralel Olmayan	Paralel		
		Dairesel	Merkezlenmiş	Grid
Schwefel	73	100	100	100
Rastrigin	45	100	85	100
Griewank	23	97	100	100

Test fonksiyonlarında hesaplamanın kalitesi global minimumun hızlı ve her defasında aynı noktanın bulunması ile ölçülür. Analitik sinyallerdeki gibi gürültü olmadığından bulunacak sonuç gerçek sonuç olacaktır.

4. ARAŞTIRMA BULGULARI

4.1 NMR Verisinin Çözümlemesi ve Gerekli Parametrelerin Optimizasyonu

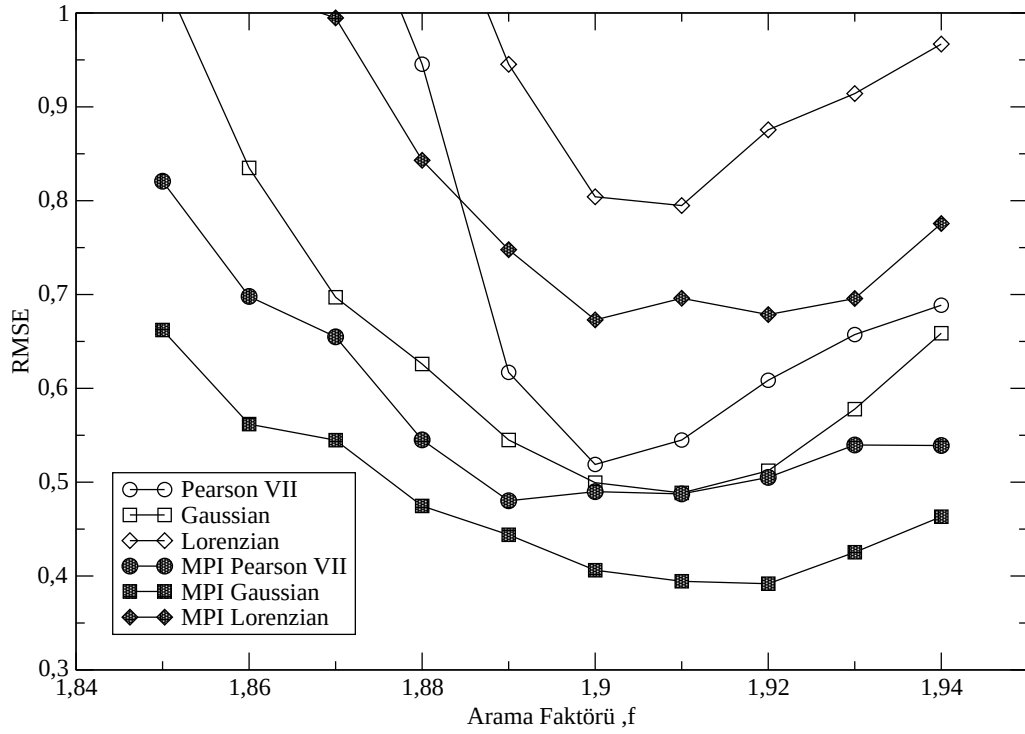
NMR ölçümü alınan örnekler 1:10 oranında döteryum çözgeni ile karıştırıldıktan sonra Bruker UltraShield TM ile ^1H -NMR ölçümü alınmıştır. Alınan bir örnek Şekil 4.1’ de verilmiştir.



Şekil 4.1: $\text{C}_{15}\text{H}_{23}\text{N}_2\text{Br}$ ’nin NMR Spektrumu

Paralel ve paralel olmayan hesaplamalarda arama uzayının genişliğini, stokastik ve gittikçe daralan bir şekilde verebilmek için “ f ” faktöründen yararlanıldı. En uygun f faktörünü bulabilmek için değeri 0,5-2,0 arasında arama yapıldı. Ancak belirli bir aralıkta (1,85 - 1,95) RMSE değeri en düşük düzeye ulaşmaktadır. Bu nedenle f değerleri 1,85-1,95 arasında daha ayrıntılı şekilde araştırılmıştır. Paralel ve paralel olmayan hesaplamalara ait grafik Şekil 4.2 görülmektedir. En iyi sonuçlar paralel ve gaussian fonksiyonu pik denklemi olarak kullanıldığında elde edilmiştir. Şekil

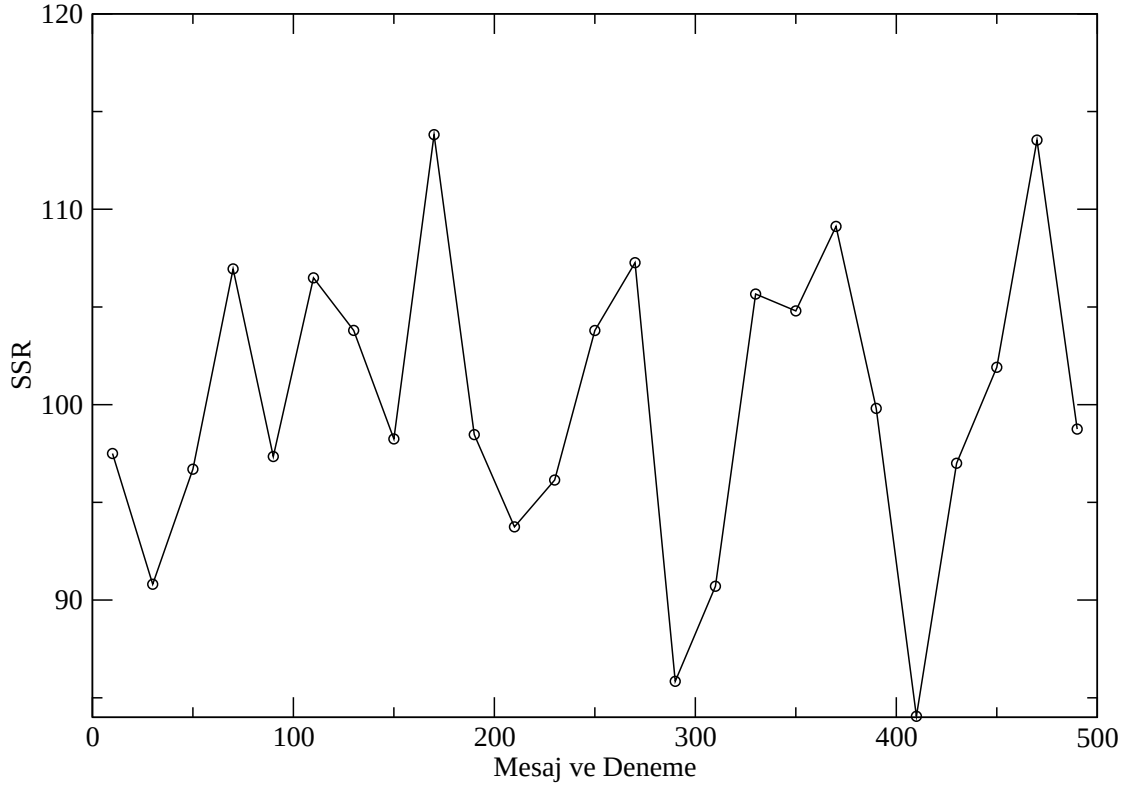
4.2'deki paralel hesaplamalarda merkezi ada modeli kullanılmıştır. Görüleceği gibi paralel hesaplamada %20 daha iyi sonuç vermektedir.



Şekil 4.2: Paralel Genetik Algoritmada Araştırma Uzayını Belirleyen Faktörün Etkisi

Öngörüleceği gibi paralel hesaplamalar en uygun sonucun bulunmasında daha hızlı sonuç vermektedir. Paralel hesaplamanın iki etkisi bulunmaktadır. Bunlardan birincisi aynı sürede daha fazla sayıda paralel hesaplama yapmasıdır. Bilgisayar işlemcilerinin her biri arama uzayında araştırma yapmaktadır. Paralel hesaplamanın diğer etkisi ise genetik farklılığın korunmasını sağlar. Yani bir işlemcili yerel en iyi değere gitmeye çalışırken, paralel diğer bir hesaplamalarda farklı bir minimum noktaya gidebilmektedir. Paralel olmayan hesaplamada en iyi değerin bulunması belirli bir yüzde orana sahiptir. Bu nedenle en iyi değeri bulmak için probleme göre en az 10 kez işlemi çalıştırıp ortalamasını almak gerekir.

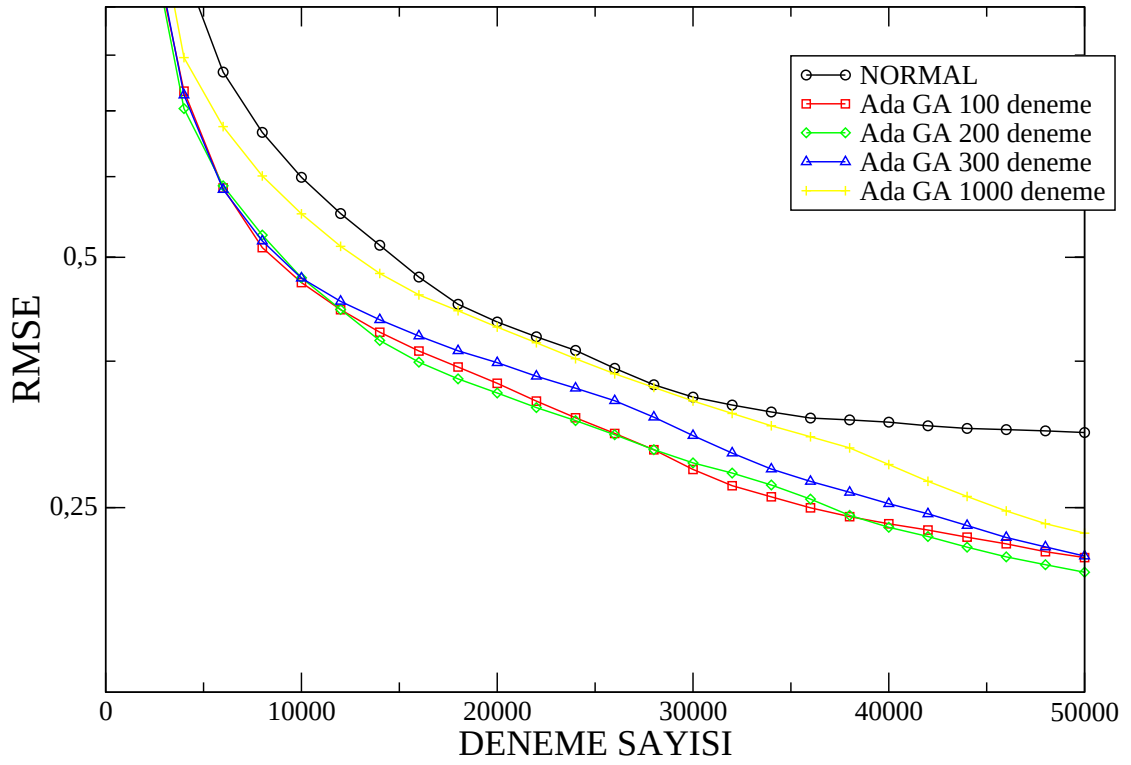
Adalar arası halka tipi haberleşmede en son adanın en ilk ada ile haberleşmesi genetik farklılığın iletilmesi için önemlidir. Bu bağlantı kesildiğinde paralel olmayan hesaplama benzer sonuçlar gözlenmiştir. Ancak grid ve merkezi ada tipi hesaplamalarda ise adalar en yakın komşusu ile karşılıklı haberleşmektedir. Haberleşme aralığı olarak 10-500 hesaplama aralığı ile haberleşme yaptığımızda,



Şekil 4.3: Adalar Arası Haberleşme Aralığı Sıklığına göre SSR değeri.

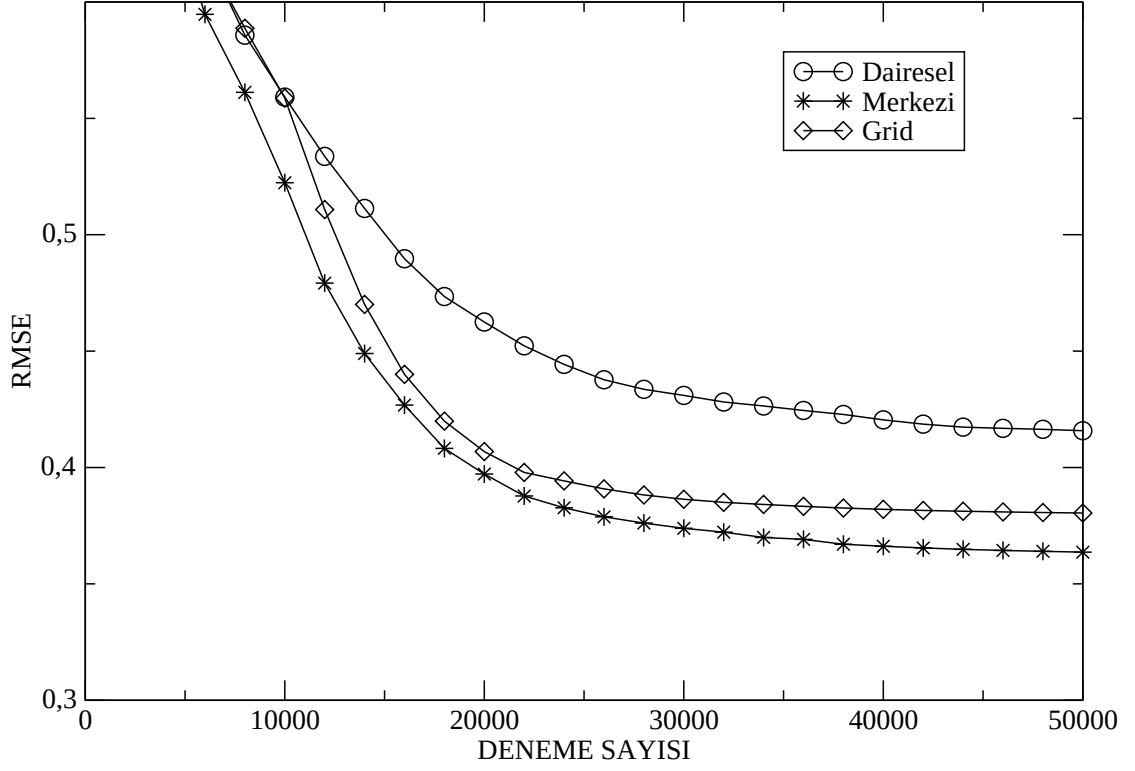
haberleşme aralığının gözle görülür anlamlı bir bağlantı olmadığı düşünülmektedir. Şekil 4.3 verilen grafik ikinci bir çalıştırmada tamamen farklı bir görüntü sergilemektedir.

Şekil 4.3’de mesajlaşma aralığı ve iterasyona karşılık standart hata toplamı verilmiştir. Görüldüğü gibi mesajlaşma aralığının parametreler tahmininde anlamlı bir ilişkisi olmadığı görülmektedir.



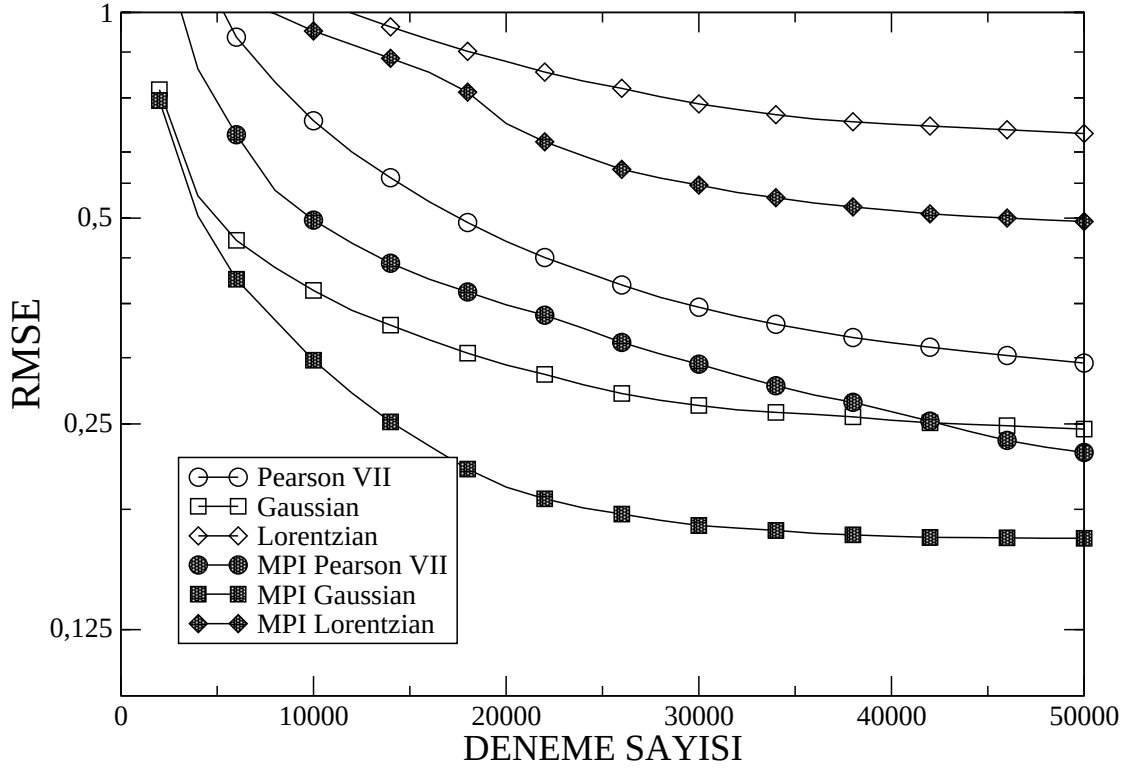
Şekil 4.4: Merkezi Ada Modelinde Haberleşme Aralığının Optimizasyona Etkisi

Çalışmamızda ada tipi hesaplamalarda 200 hesaplamada, adalar arası en iyi sonucu yakın adaya gönderme yöntemi kullanılmıştır. Mesajlaşma aralığının spektrum tipine göre farklı sonuçlar vereceği ve uygun bir değerin bulunup bulunmayacağı konusunda kararsız kalınmıştır.



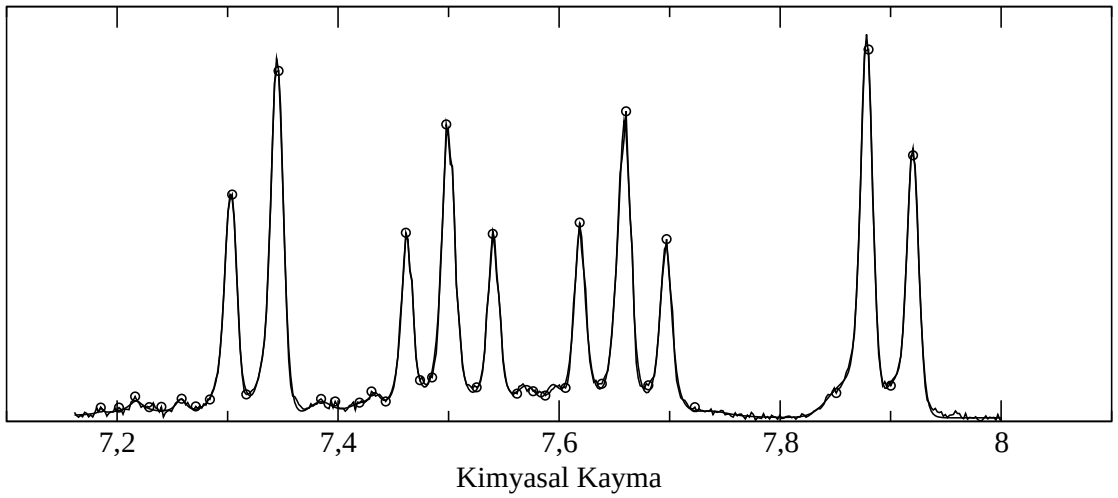
Şekil 4.5: Değişik Ada Modellerinin Optimizasyona Etkisi

Arama uzayı faktörü olarak 1,90, 1,91 alınabileceği şekil 4.2 görülmektedir. $r=1,91$ mesajlaşma aralığı 200 seçilerek her bir hesaplama yönteminin iterasyona bağlı olarak nasıl davrandığı şekil 4.6 verilmiştir. Gaussian denklemi pik modeli olarak kullanıldığında paralel hesaplama (merkezi ada modeline göre) RMSE değeri 5 kat daha iyi sonuç vermektedir. Ayrıca paralel olmayan hesaplamada 50000 iterasyon sonucunda bulunan değere, paralel hesaplama sonucunda 10000 iterasyonda ulaşılmaktadır. 9 işlemci daha fazla sayıda hesaplama yaptığı düşünülse de sonuç daha iyi ve hesaplama zamanı çok daha kısa olmaktadır.

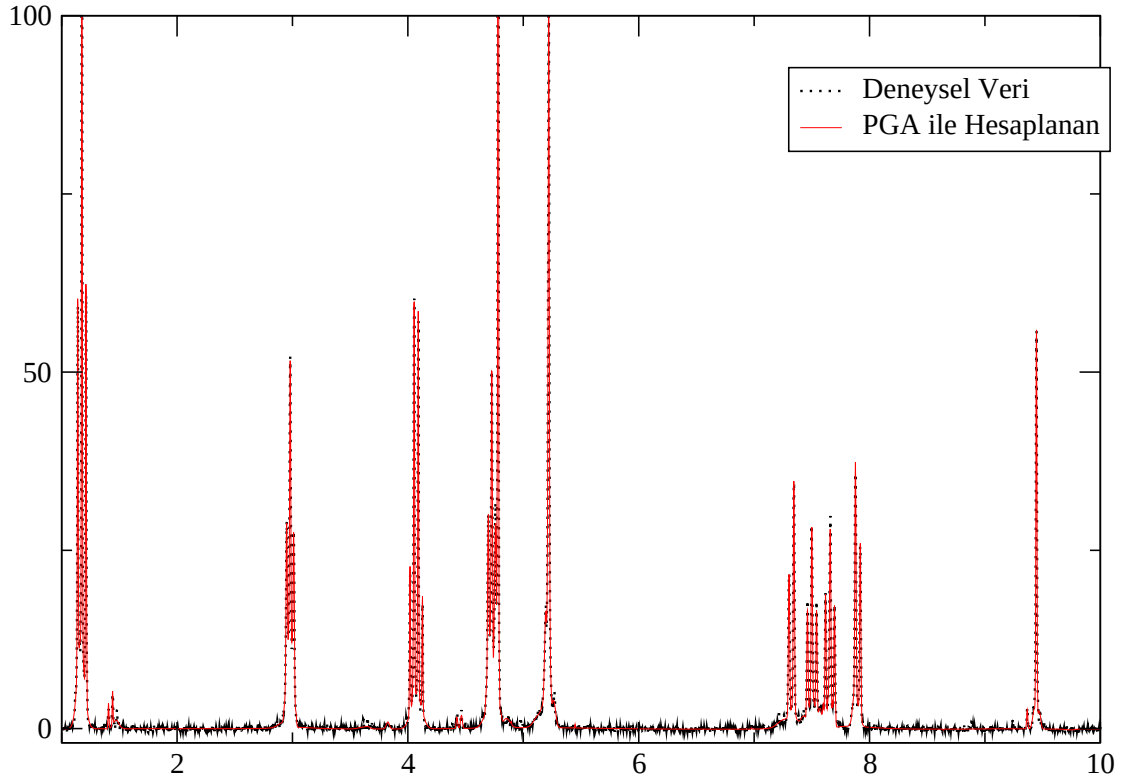


Şekil 4.6: Paralel ve Paralel Olmayan Pik Fonksiyonunun Optimizasyona Etkisi

Yöntemlerin uygun şartlarını bulmak için tüm spektrum yerine spektrumun bir parçası kullanılmıştır. Bu bize hesaplamalarda hız kazandırmaktadır. Şekil 4.7 'de kullanılan orijinal NMR spektrum parçası ve çözülmüş spektrum üst üste verilmiştir. İki spektrum arasında fark az olduğu sinyaller üst üste çıkmıştır. Tüm spektrumun çözülmesi daha uzun süre almaktadır.



Şekil 4.7: Seçilmiş NMR Spektrum Eğrisi ve İşaretlenmiş Pikler



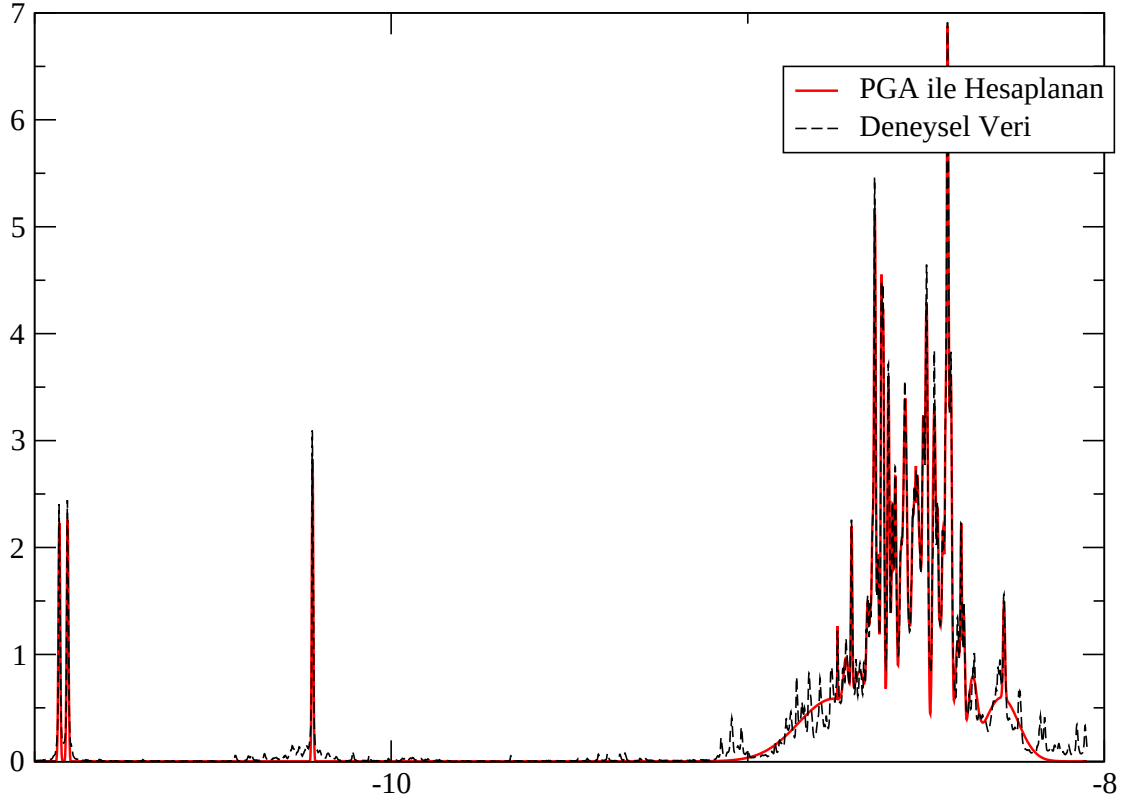
Şekil 4.8: Deneysel ve Teorik NMR Sonuçlarının Karşılaştırılması

Tablo 4.1: Şekil 4.8 için PeakFit ile Pik Sayıları ve SE Değerleri

	AutoFit I	AutoFit II	AutoFit III	PGA ile
Bulunan Pik sayısı	91	11	27	106
SE Değeri	8,86244	7.22486	5,86686	0,6

Tüm spektrumun çözümü Şekil 4.8’de gösterilmektedir. Spektrumun çözümü Linux işletim sistemi ve 9 işlemci yardımıyla yaklaşık 40 dakika sürmektedir.

Geliştirdiğimiz bilgisayar yazılımı ile daha önce ölçümelerini aldığımız kayısı türlerinin NMR örneklerinin çözümlemeleri Şekil 4.9, 4.10, 4.11, 4.12’de görülmektedir. Ticari yazılım ile geliştirdiğimiz program sonuçlarında karşılaştırmalı olarak tablolar da verilmiştir.

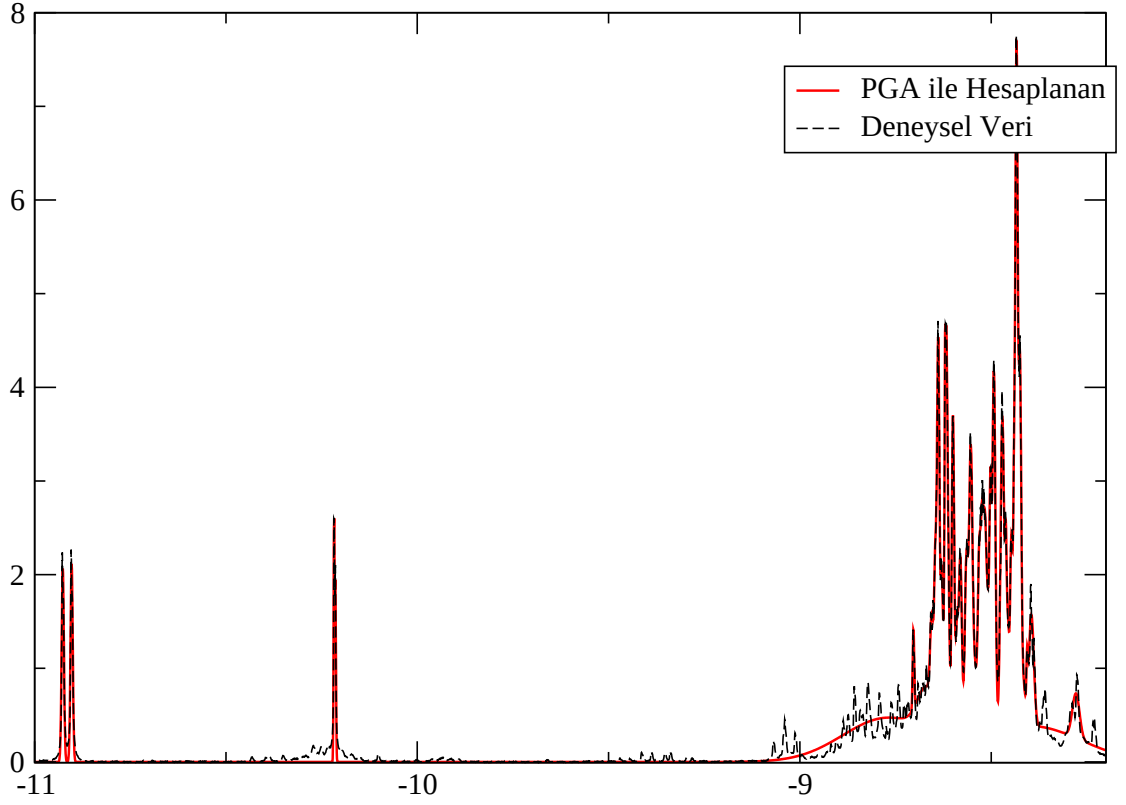


Şekil 4.9: Adil Cevaz 5

Tablo 4.2: Adil Cevaz 5 için SE Değerleri

	AutoFit 1	AutoFit 2	AutoFit 3	PGA
SE	0,3292	0,2698	0,2219	0,0858

Her bir pikin bir fonksiyon olduğu düşünülürse PGA ile geliştirilen yazılım “Adil Cevaz 5” kayısı türü için toplamda 47 pik bulmuştur. 47 pik için bulunan SE değeri 0,0858’dir. Bu değer ticari yazılıma göre oldukça iyidir.

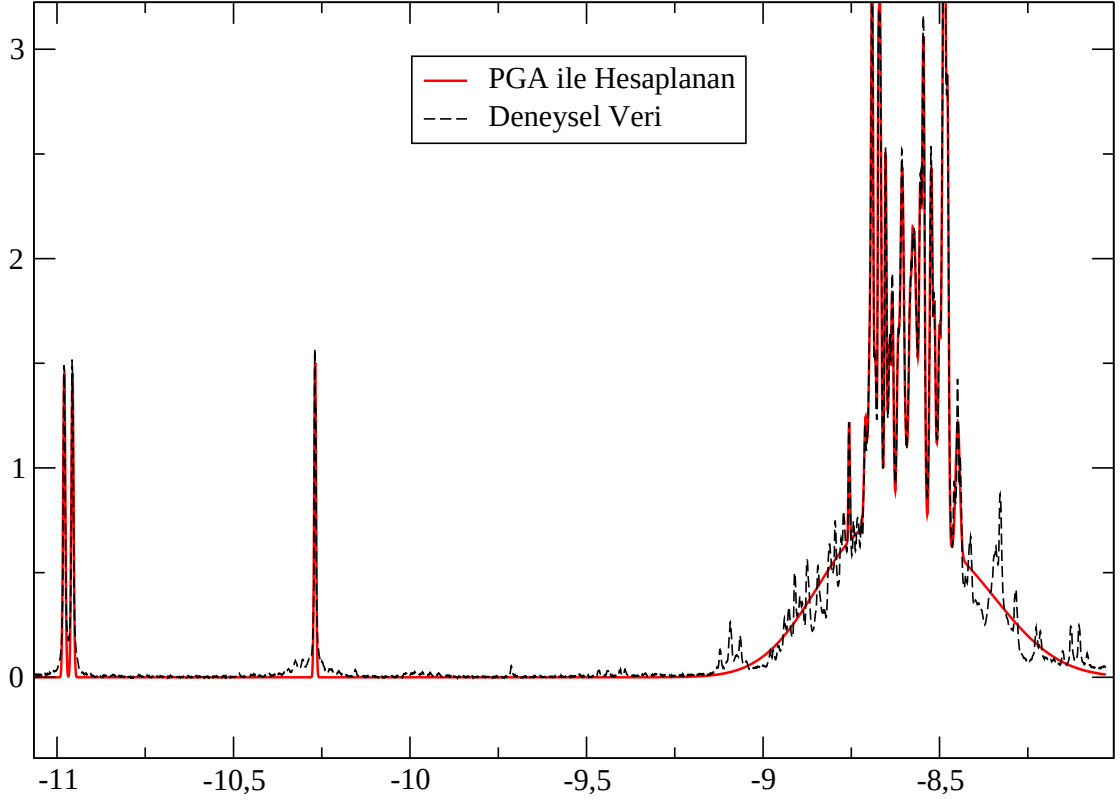


Şekil 4.10: Hacı Halil Oğlu

Tablo 4.3: Hacı Halil Oğlu için SE Değerleri

	AutoFit 1	AutoFit 2	AutoFit 3	PGA
SE	0,2515	0,2203	0,1794	0,0934

Tablo 4.3’de “Hacı Halil Oğlu” kayısı türü için gösterilen PGA değeri 39 pik yardımıyla bulunmuştur. Pik sayısını arttırarak PGA değerini düşürmek mümkündür. Ancak bu hali ile dahi ticari yazılımdan daha anlamlı sonuç verdiği görülmektedir.

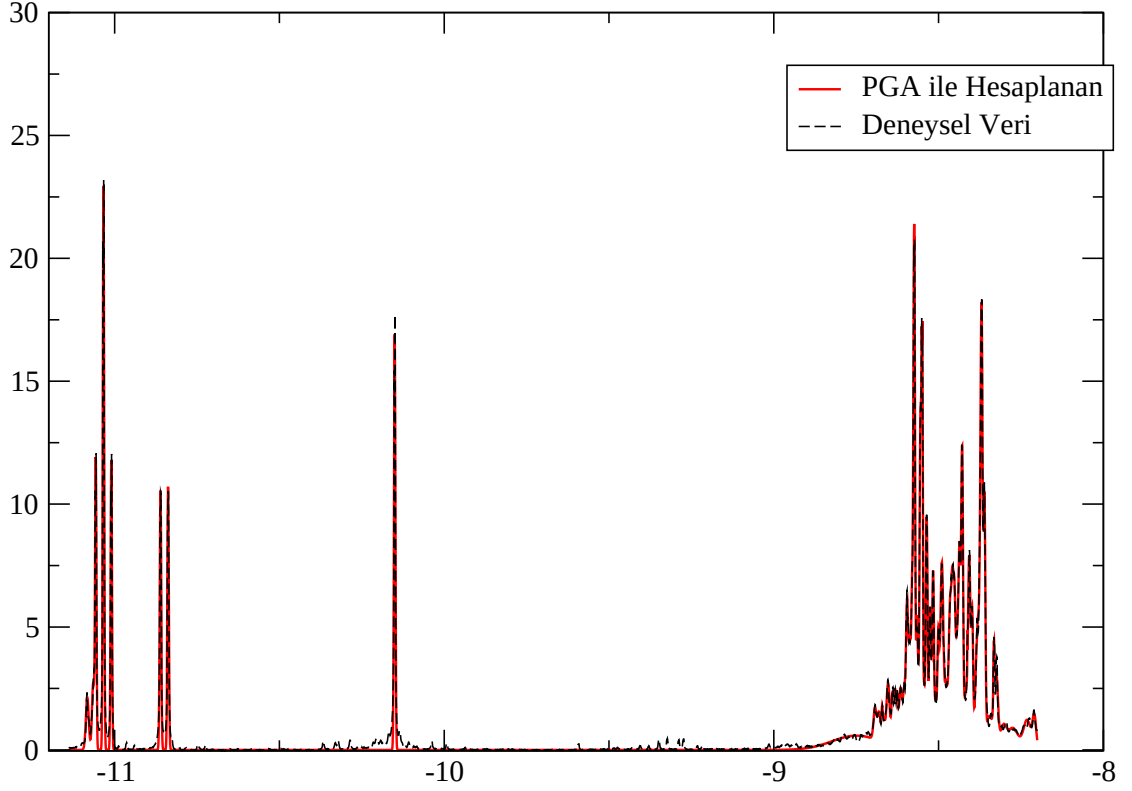


Şekil 4.11: Hacı Kız

Tablo 4.4: Hacı Kız için SE Değerleri

	AutoFit 1	AutoFit 2	AutoFit 3	PGA
SE	0,2089	0,1485	0,1580	0,0771

Tablo 4.4’de “Hacı Kız” kayısı türü için gösterilen PGA değeri 34 pik yardımıyla bulunmuştur. Ticari yazılımdan iki kat iyi sonuç verdiği yine Tablo 4.4’de görülmektedir.



Şekil 4.12: Kaba Aşı

Tablo 4.5: Kaba Aşı için SE Değerleri

	AutoFit 1	AutoFit 2	AutoFit 3	PGA
SE	0,1230	0,1066	0,0772	0,0606

PGA ile geliştirilen yazılım “Kaba Aşı” kayısı türü için sonucu sadece 20 pik kullanarak hesaplamıştır. Ticari yazılım ile karşılaştırması Tablo 4.5’de görülmektedir.

Litaretürde sıklıkla kullanılan ve ticari bir yazılım olan PeakFit¹ adlı yazılımla Şekil 4.8’de gösterilen NMR spektrumu incelendiğinde, program ilk etapta 19 pik bulmakta, taban düzenlemesi ve yumuşatma işlemleri iptal edildiğinde 91 pik bulmaktadır. Bizim geliştirdiğimiz algoritma ile çalışan yazılım ise 106 pik tanımlamaktadır. Çok karışık bir pik içinde parametreleri ayarlayarak bulunan pik sayısını arttırabilmekte ya da azaltabilmekteyiz.

¹<http://www.sigmaplot.com/products/peakfit/peakfit.php>

Tablo 4.1’de de açıkça görülmektedir ki geliştirmiş olduğumuz yazılım ticari yazılımdan daha iyi sonuç vermektedir. Tablo 4.6’de ise ticari yazılım ile taban düzeltme ile yumuşatma fonksiyonları ile oynanarak bulunan maksimum pik sayıları ve SE değerleri verilmiştir.

Tablo 4.6: Şekil 4.8 için Bulunan Maksimum Pik Sayısı SE Değerleri

	AutoFit I	AutoFit II	AutoFit III
Bulunan Pik sayısı	100	100	100
SE Değeri	8,87242	8,87242	8,87242

4.2 UV-Vis Verilerinin Değerlendirilmesi

UV-Vis spektroskopisi organik ve inorganik karışımlarının kalitatif ve kantitatif analizleri için kullanılan ucuz ve hızlı bir tekniktir. Bileşenin türüne ve bileşen sayısına bağlı olarak spektrum karmaşık bir hal alabilir. Üst üste binmiş piklerin yeri, yüksekliği ve genişliği spektral bandlar hakkında bilgi verir. UV-Vis spektrumu diğer spektrumlardan oldukça farklıdır. Daha az sayıda ve daha geniş bandlar içerir. Ayrıca bu absorpsiyon bandları birbirinden farklı fonksiyonlara sahip olabilir.

Çoğunluklu bu spektral bandların çözümlemesinde Lorentzian ve Gaussian fonksiyonuna kullanılmasına karşın UV-Vis spektrumlarının yayvan olmasından dolayı çoğunlukla Gaussian fonksiyonu kullanılmaktadır.

Ultraviyole görünür bölge absorpsiyon spektrumlarının ayrıştırılması için çeşitli yöntemler mevcuttur. SpecFit² ve PeakFit gibi ticari uygulamalar ile bu spektrumlar değerlendirilebilir.

UV-Vis spektrumunun kalitatif değerlendirilmesinde daha çok görsel değerlendirmeden yararlanılır. Pik genişliği ve bölgesi o bileşenin moleküler yapısını gösterir.

UV-Vis bölge spektrumunun kantitatif analizi ise daha çok çoklu bileşen analizi olarak yapılır. Bileşen sayısı kadar dalga boyunda çeşitli derişimlerde çözelti

²<http://www.pha.jhu.edu/~gak/specfit.html>

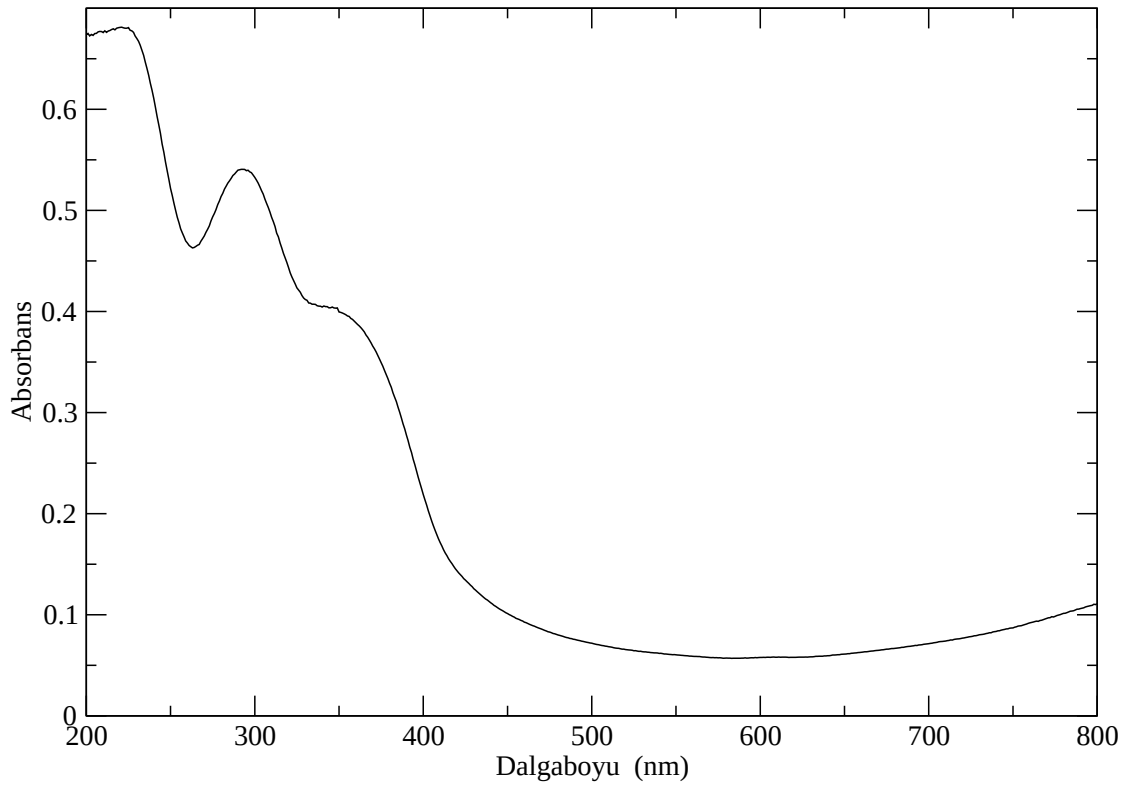
hazırlanarak bir veri matrisi oluştururlur. Bu matrisin çözümü ile kantitatif bilgiye ulaşılır.

Spektrumun piklerinin çözümlemesi ise olası elektronik geçişlere ait bandların teorik incelenmesi için gerekmektedir.

UV pik çözümlemesi için daha önce laboratuvarlarımızda alınmış ölçümler ile UV veritabanından alınan veriler kullanılmıştır.

4.2.1 ZrO_2 'nın UV-VIS Absorbsiyon Spektrumu Değerlendirilmesi

Üniversitemizin nanoteknolojik alanında araştırma yapan araştırmacılarımızın elde ettiği ölçüm [103] şekil 4.13'de verilmiştir. Toplam 4 tane absorbsiyon bandı görünen bu spektrumu 4 tane pik fonksiyonu ile çözmek olanaksızdır. Bu spektrum tablo 4.7'dan görülebileceği gibi 7-21 tane Gaussian



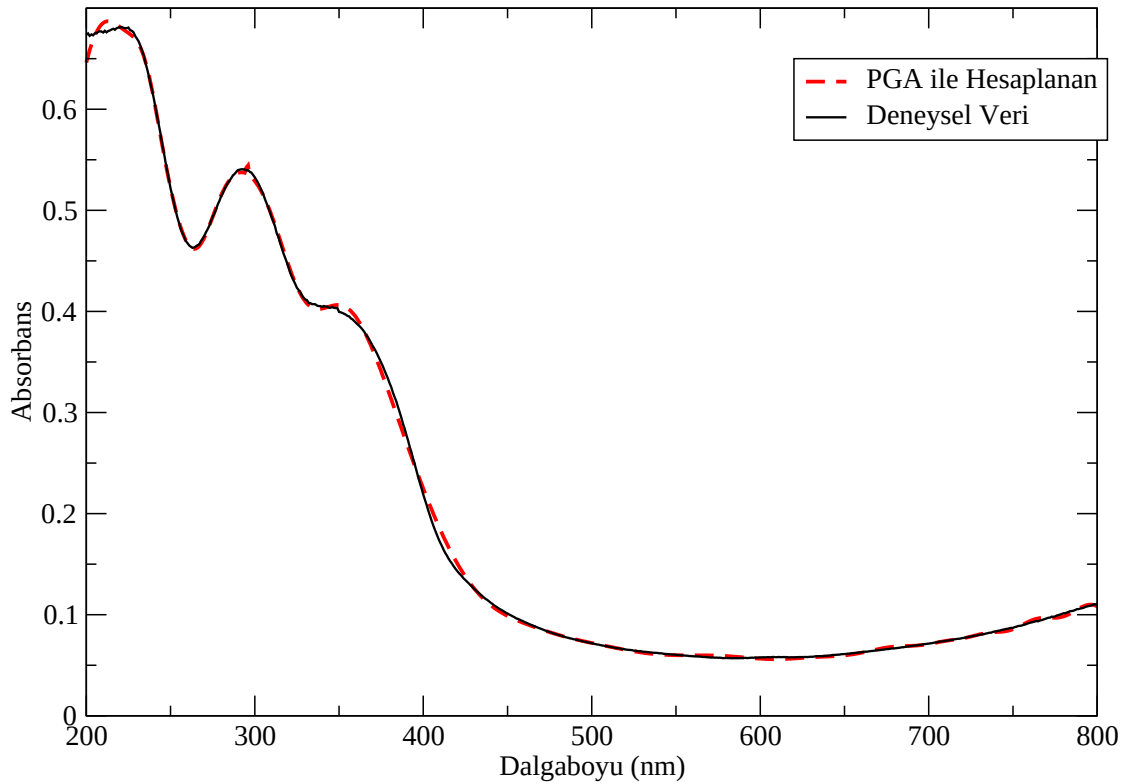
Şekil 4.13: ZrO_2 UV-Vis Spektrumu Verisi

Şekil 4.13'de gösterilen bileşiğin PGA ile pik sayısına göre bulduğumuz

standart hata deęerleri tablo 4.7’de gsterilmiřtir. Tablodan grlebileceęi gibi spektrumu temsil etmek zere toplanan gaussian fonksiyonu sayısı arttıka hata deęerleri byk oranda azalmaktadır. Spektrumdan ka tane gaussian bandının olduęunu kestirmek mmkn deęildir. Ancak UV ve grnr blgedeki geiřlerin sayıları gz nne alınarak kalitatif irdeleme yapılabilir. Kantitatif analiz amalı hesaplama yapabilmek iin eřitli deriřimlerde spektrum alınarak ok bileřenli analiz yapılması gerekir. Bu řekilde bir hesaplama ile deriřime baęlı molekller arası etkileřimlerden gelebilecek standart en aza indirilmiř olacaktır.

Tablo 4.7: UV Pik Sayısı ve Standart Hata Deęerleri

PGA ile Bulunan Pik Sayıları ve RMSE Deęerleri	
Pik Sayısı	RMSE
7	0.0115606
8	0.0097084
11	0.00339492
16	0.00273777
21	0.00118858



řekil 4.14: Referans [103] iin PGA ve Deneysel Veri

Aynı örneğimiz ticari bir yazılım ile değerlendirildiğinde standart hata değerleri aşağıdaki tablodaki gibi görülmektedir.

Tablo 4.8: Ticari Yazılı ile Bulunan Standart Hata Değerleri

SE Değerleri		
AutoFit I	AutoFit II	AutoFit III
0,2225	0,0737	0,1283

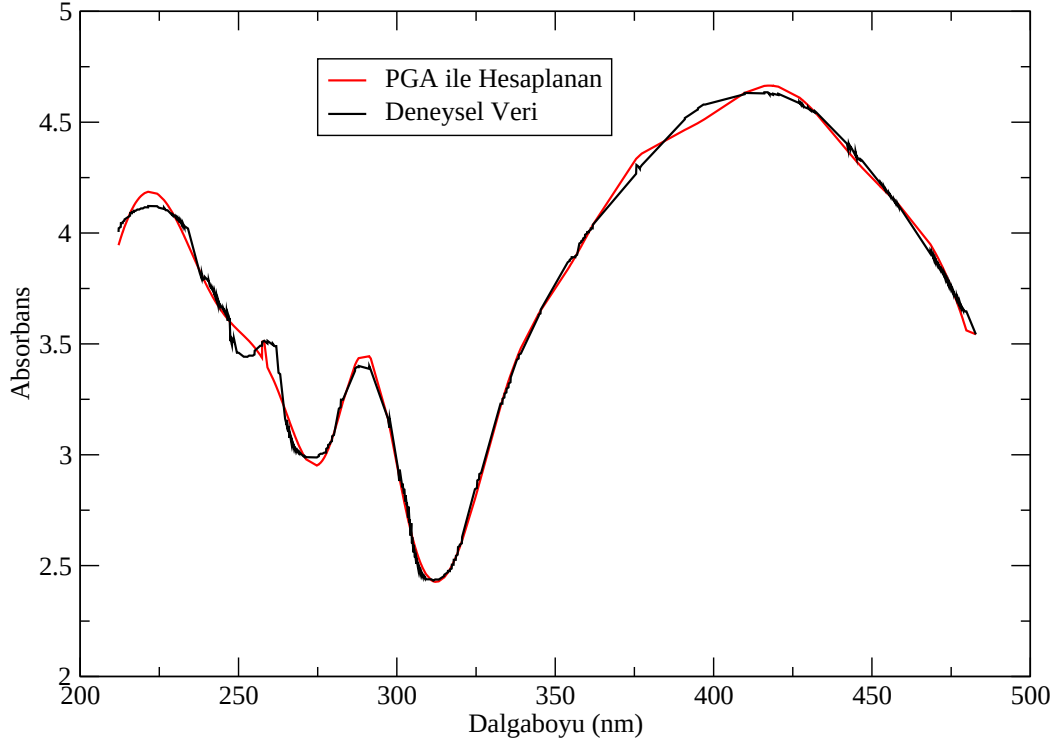
Geliştirdiğimiz program neredeyse Referans [103] için sıfıra yakın bir değer üretmektedir.

4.2.2 $C_{19}H_{16}O$ Spektrumu

Bu spektrum hatalı ölçüm ve az sayıdaki veri noktasından örnek olarak kimya veri tabanlarından alınmıştır[104]. Gürültü içeren geniş ve dar bandlardan oluşan bu spektrum PGA ile değerlendirilmesi yapılmış, şekil 4.15 gösterilmiş ve ticari program ile RMSE kıyaslanması tablo 4.9 verilmiştir. Spektrum verilerine işlem yapıldığında az sayıdaki veri nedeniyle spektrumda veri kaybı meydana gelmektedir. Gürültü olarak değerlendirilebilecek bir sinyal zayıf bir band geçişine karşılık gelmesi olasıdır. PGA ile sinyalin çözümünde toplam 12 pik bulunarak değerlendirme yapılmıştır.

Tablo 4.9: Referans [104] için SE Değerleri Karşılaştırması ($C_{19}H_{16}O$)

AutoFit 1	AutoFit 2	AutoFit 3	PGA
0.8897	0.8849	0.4828	0.0222



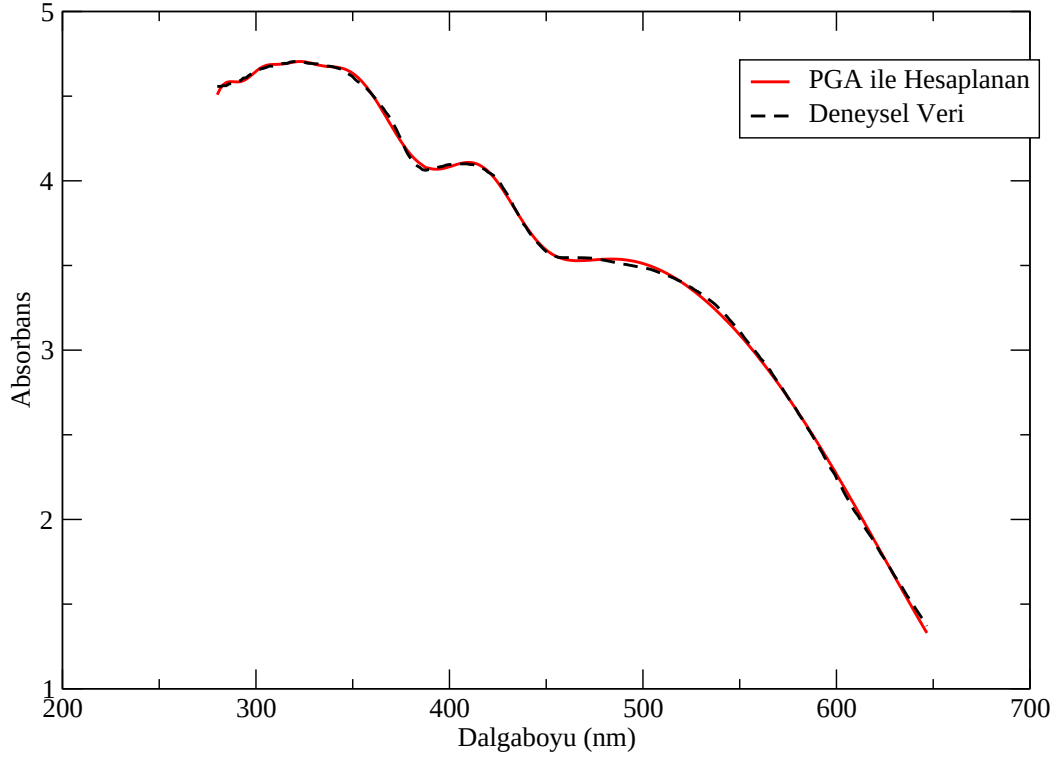
Şekil 4.15: Referans [104] için DeneySEL ve PGA ile Hesaplanan UV Spektrumu ($C_{19}H_{16}O$)

4.2.3 $C_{45}H_{33}FeO_6$ Spektrumu

Kimya veri tabanından alınan bu spektrum verisinde 3 tane absorbsiyon bandı görülmekte ancak taban soğurum piki konusunda herhangi bir bilgi içermemektedir. Hesaplanan ile değerlendirilen veriler arasında yüksek bir korelasyon gözlenmiştir. Toplam 18 pik fonksiyonu ile yapılan değerlendirme Şekil 4.16'de ve karşılaştırmalı değerleri Tablo 4.10'da gösterilmiştir.

Tablo 4.10: Referans [105] için SE Değerleri Karşılaştırması ($C_{45}H_{33}FeO_6$)

AutoFit 1	AutoFit 2	AutoFit 3	PGA
0,9999	0,4395	0,2915	0,0174



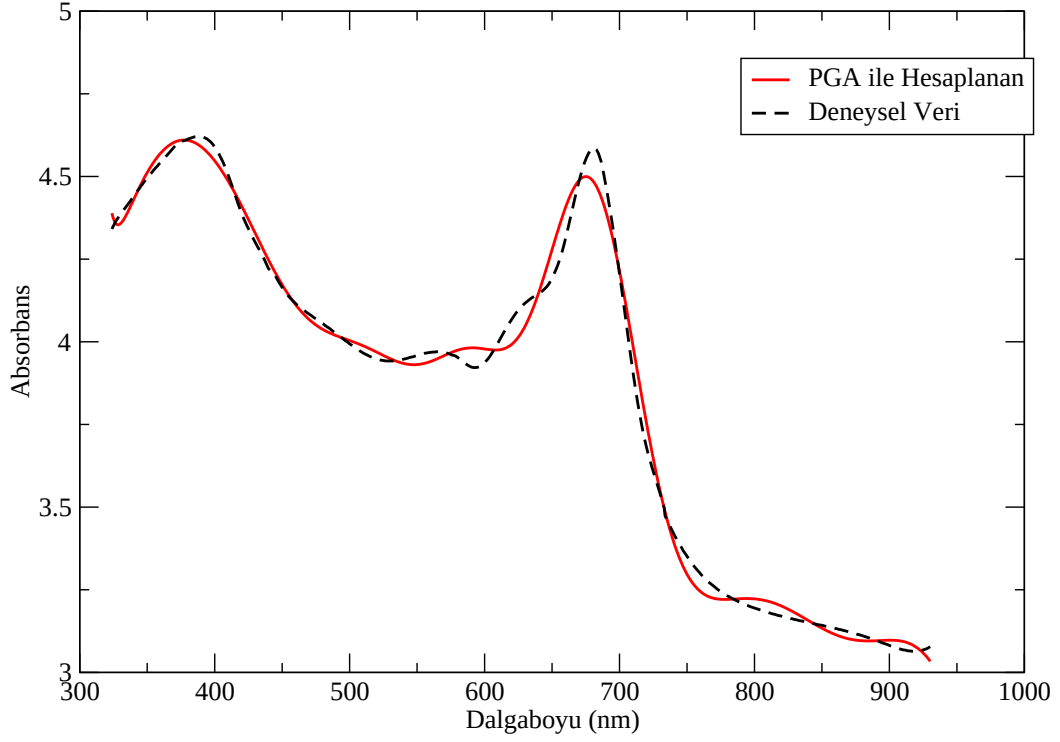
Şekil 4.16: Referans [105] için Deneysel ve PGA ile Hesaplanan UV-Vis Spektrumu($C_{45}H_{33}FeO_6$)

4.2.4 $C_{36}H_{44}FeN_4O_2$ Spektrumu

Taban kayması, geniş ve dar absorpsiyon bandları içeren bu spektrum toplam 12 gaussian fonksiyonu ile hesaplamaya çalışılmıştır. Önceki UV-Vis spektrumunda başarılı bir şekilde hesaplayabilirken bu hesaplamada ticari programa göre daha iyi değer elde edilebilmesine rağmen hesaplamanın yeterli düzeyde olmadığı görülmektedir. Hesaplanan ve deneysel verinin tam çakışmadığı görülmektedir. İterasyon sayısını çok fazla arttırdığımızda çözüme ulaşabilecekken diğer hesaplamalarla aynı sayıda iterasyon içerdiğinde grafik bu şekilde bırakılmıştır. Otomatik pik bulma burada yeterli başarıyı gösterememiştir.

Tablo 4.11: Referans [106] için SE Değerleri Karşılaştırması ($C_{36}H_{44}FeN_4O_2$)

AutoFit 1	AutoFit 2	AutoFit 3	PGA
0.0548	0.7078	0.0692	0,0404



Şekil 4.17: Referans [106] için Deneysel ve PGA ile Hesaplanan UV Spektrumu ($C_{36}H_{44}FeN_4O_2$)

Paralel genetik algoritma ile yaptığımız optimizasyon NMR verilerinin değerlendirilmesinde olduğu gibi UV verilerinin değerlendirilmesinde de iyi sonuç vermiştir.

4.3 IR Verilerinin Değerlendirilmesi

Hesaplama kolaylığı açısından geçirgenlik değerleri denklem 4.1 ile absorpsiyon değerlerine dönüşüm yapılmıştır.

$$Absorbans = -\log\left(\frac{Geçirgenlik}{100}\right) \quad (4.1)$$

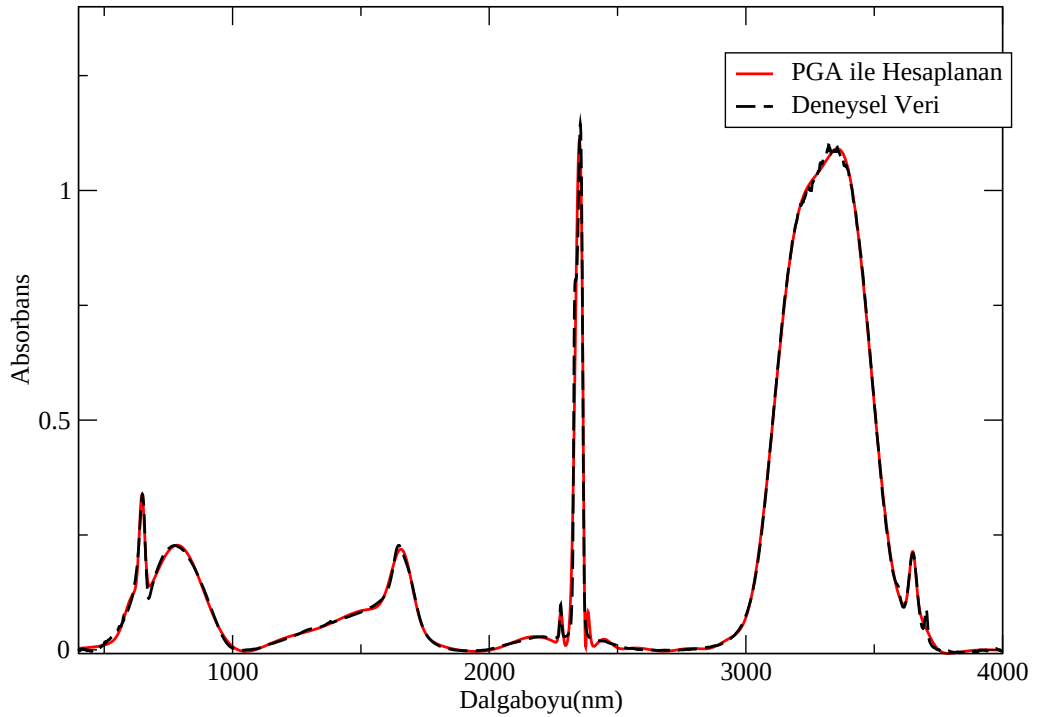
IR spektrumlarının çözümlenmesin de diğer spektrumlara göre daha başarılı sonuçlar elde edilmektedir. Zemin sinyalinin sıfıra yakın olması, zemin kaymasının bulunmaması piklerin simetrik olması nedeniyle hesaplanan ile deneysel veri fark edilmeyecek kadar birbirine uyuşmaktadır.

Genetik algoritma parametreleri Bölüm 4.1'deki gibidir.

Analizlenecek spektrumların seçiminde görsel olarak birbirinden farklı görünen ve farklı pik şekilleri içeren spektrumlar seçilmiştir. Buradaki amaç geliştirdiğimiz algoritmanın başarısını test etmektir.

4.3.1 $CO_2 + H_2O$ (17K) Spektrumu

Kimyasal veri tabanından aldığımız veriler ile piklerin ayrıştırılmasına çalışılmıştır. Şekil 4.18’de görüleceği gibi hesaplanan ile deneysel veri 26 gaussian fonksiyonu kullanılarak hesaplanmıştır. Bu spektrumda Tablo 4.12’ye bakıldığında yaklaşık 10 kat daha iyi sonuç bulunmuştur. İterasyon ve parametreler değiştirilerek daha iyi sonuç elde edilebilmektedir. Bu spektrumun seçilmesindeki temel amaç



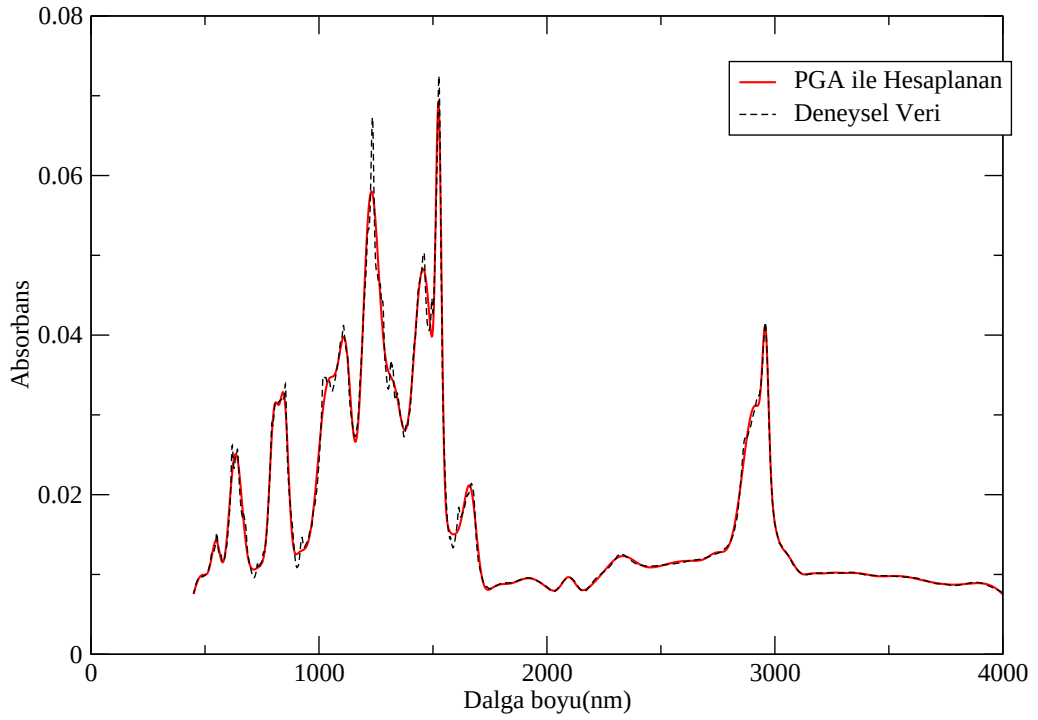
Şekil 4.18: Referans [107] için Deneysel ve Hesaplanan IR verilerinin Karşılaştırılması

Tablo 4.12: Referans [107] için Ticari Yazılım ve PGA Karşılaştırılması

AutoFit 1	AutoFit 2	AutoFit 3	PGA
0,2988	0,0864	0,1677	0,0135

4.3.2 $(C_{23}H_{30}N_2)_2PdCl_2$, $(C_{27}H_{30}O_4N_2)_2PdBr_2$ ve $(C_{25}H_{26}O_2N_2)_2PdCl_2$ Spektrumları

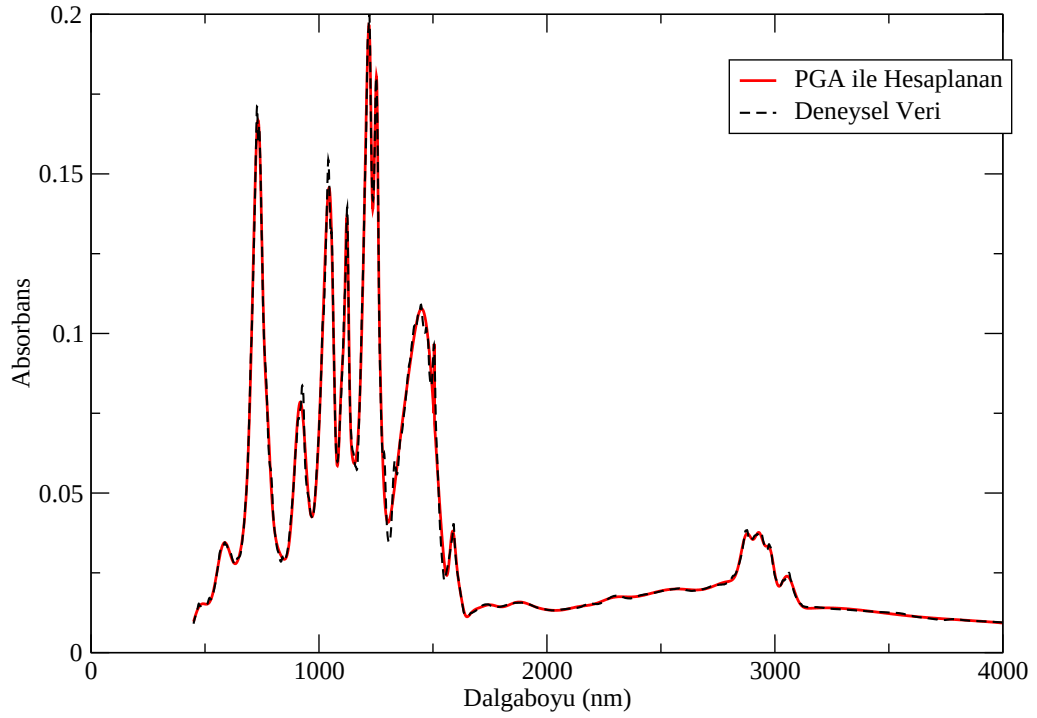
Aynı analizci tarafından alınmış bu 3 spektrumu yazılım toplam 36 gaussian fonksiyonu ile çözümlemiştir. $(C_{23}H_{30}N_2)_2PdCl_2$ ve $(C_{27}H_{30}O_4N_2)_2PdBr_2$ spektrumlarında pozitif bir taban kayması varken $(C_{25}H_{26}O_2N_2)_2PdCl_2$ örneğinin spektrumunda negatif bir taban kayması görülmektedir. Bu spektrumlarda üst üste binmiş dar bandlar ve zemin sinyalinin sıfırdan farklı olduğu bir spektrumlardır. Dar bandlar birbirine çok yakındır. Deneysel ve teorik verinin birbirine tamamen uyması için bazı pik yerlerinin ve sayılarının değişmez şekilde sabitlenmesi gerekir.



Şekil 4.19: Referans [108] $((C_{23}H_{30}N_2)_2PdCl_2)$ için Deneysel ve Hesaplanan IR verilerinin Karşılaştırılması

Tablo 4.13: Referans [108] $((C_{23}H_{30}N_2)_2PdCl_2)$ için Ticari Yazılım ve PGA Karşılaştırılması

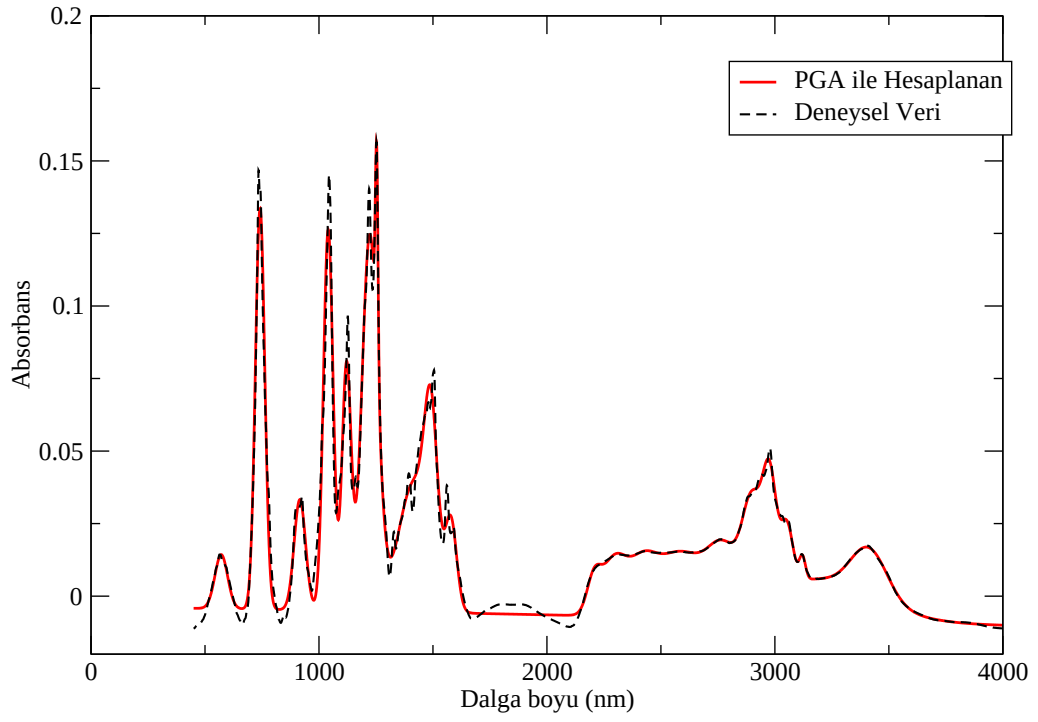
AutoFit 1	AutoFit 2	AutoFit 3	PGA
0,0120	0,0019	0,0032	0,0009



Şekil 4.20: Referans [108] $((C_{27}H_{30}O_4N_2)_2PdBr_2)$ için Deneysel ve Hesaplanan IR verilerinin Karşılaştırılması

Tablo 4.14: Referans [108] $((C_{27}H_{30}O_4N_2)_2PdBr_2)$ için Ticari Yazılım ve PGA Karşılaştırılması

AutoFit 1	AutoFit 2	AutoFit 3	PGA
0,0260	0,0036	0,0131	0,0019



Şekil 4.21: Referans [108] $((C_{25}H_{26}O_2N_2)_2PdCl_2)$ için Deneyisel ve Hesaplanan IR verilerinin Karşılaştırılması

Tablo 4.15: Referans [108] $((C_{25}H_{26}O_2N_2)_2PdCl_2)$ için Ticari Yazılım ve PGA Karşılaştırılması

AutoFit 1	AutoFit 2	AutoFit 3	PGA
0,0135	0,0049	0,0130	0,0038

5. TARTIŞMA ve SONUÇ

Bu çalışmada paralel ve paralel olmayan genetik algoritmik hesaplamalar başarı ile kompleks problemlerin çözümüne uygulanmıştır. Analitik sinyallerin değerlendirilmesi ve ayrıştırılması için uygulanan hesaplamalar ticari programlara göre oldukça yüksek bir düzeyde başarılı sonuçlar vermiştir.

Genetik algoritma yönteminin hesapsal ve yapısal olarak iyileştirilmesi ile bu tür hesaplamalarda kullanılan test amaçlı çok modlu fonksiyonların minimizasyonu her defasında %100'e varan bir doğruluk ile hesaplandığı gözlenmiştir.

Genetik Algoritma hesaplamalarını daha hızlı ve daha etkin yapabilmek için uyguladığımız paralel hesaplama modelleri hem çok modlu fonksiyonların hesaplamasında hem de analitik sinyallerin değerlendirilmesinde gözle görülür bir iyileştirme ve hız sağladığı görülmüştür. Genetik algoritmanın paralelleştirilmesi ve programlanması oldukça zor bir işlemdir. Literatürde paralelleştirilmiş genetik algoritmalar genelde bireylerin hesaplanmasının dağıtılmasında kullanılmaktadır. Bizim geliştirdiğimiz paralel genetik algoritmada doğadaki gibi gerçek evrimsel işlemler benzetimli olarak kullanılmıştır.

Paralelleştirmede hesaplama ve kodlama açısından en uygun ve hızlı olabilecek halka, merkezi ve ağ tipi ada modelleri esnek bir kodlama sağlayacak şekilde problemlere uygulanmıştır. Evrimsel hesaplamalarda ada modelleri kullanılması konusunda yapılan çalışmalar çok yeni olup genelde basit problemlere uygulanmaktadır. Gerçek hayattaki hesaplamalarda yapılan işlemler ise ticari kaygı nedeniyle nasıl yapıldığı konusunda bilgi verilmemektedir. Bu tür hesaplamalar günümüzde teknolojinin ilerlemesinde önemli katkılar sağlamaktadır. Bu çalışmada bulduğumuz sonuçlar ile yüksek hız ve doğruluk gerektiren çalışmalara temel oluşturacağı kanaatindeyiz. Yüksek maliyet gerektiren deneyler bu hesaplamalar ile düşük maliyetle yeterli veri sağlanması olanaklıdır.

Karşılaştırma amacıyla denenen 3 ada modelinden (halka, merkezi ve grid) mesajlaşma yoğunluğu ve yönleri açısından en fazla olan ağ tipi bekleneceği gibi en iyi sonucu vermiştir. Bunun nedeni her bir adanın bulduğu en iyi sonucu diğer adalara ulaştırarak global en iyi noktaya her defasında ulaşması sağlanmıştır. Halka modeli diğer iki modelden kötü sonuç vermektedir. Bunun nedeni de haberleşme sayısının azlığıdır. Aslında haberleşme basamağındaki bilgi alışverişi ada sayısının azlığından dolayı çok fazla zaman alıcı bir işlem değildir. Bununla birlikte bilgisayar kodlaması ve mesajlaşma hiyerarşisinin düzenlenmesi daha basit olduğundan dolayı büyük ölçekli sistemlere kolay uygulanabilirliği nedeniyle tercih edilebilir.

Genetik algoritma hesaplamalarında yeni bireylerin hesaplanması sırasında uyguladığımız iyi sonuç baz alınarak hesaplama uzayının gauss dağılımına bağlı olarak seçilmesi, bu şekilde uyguladığımız yöntem litaretürde rastlanmayan bir hesaplama şeklidir. Diğer önemli bir husus ise litaretürde genelde hesaplamayı hızlandırmak için popülasyon sayısı kadar rastgele yeni birey üretilmesi şeklinde yapılan hesaplamalar yerine çalışmamızda her adımda yeni tek bireyler üreterek evrimsel hesaplama sürekli canlı tutulmuştur. Bu şekilde hesaplama ile daha önce hesaplanmış değerleri yeniden kullanma olanağı da sağlanmıştır.

Hesaplama sayısının diğer yöntemlere göre çok fazla görünmesi bizim yaptığımız hesaplamalarda jenerasyon sayısı yerine hesaplama sayısı olarak almamızdandır. Ayrıca hesaplama sayısında azaltma yapıldığında gerçek değer kaçırılması olasılığı bulunmaktadır.

Genetik algoritma ile sinyaller değerlendirilirken sadece Gaussian eğrisi kontrol edilmemiştir. Bunun yanında Lorentzian ve Pearson VII fonksiyonları da test edilmiştir. Bu fonksiyonlar hem paralel hem de paralel olmayan programlarla kontrol edilmiş en iyi sonucun paralel Gaussian fonksiyonu olduğu görülmüştür.

Fonksiyonlar paralel ortamda test edilirken adalar arası haberleşme aralığı kontrol edilmiştir (10-500). Ancak haberleşme aralığının popülasyon sayısı ve fonksiyon arasında bir anlamlı bağıntı görülememiştir. Paralel hesaplamada sadece ada sayısı ve adalar arası bağlantı sayısının etkili olduğu görülmüştür.

Genetik algoritma ile geliştirilen programa veri girilirken veriler üzerinde taban düzeltmesi veya yumuşatmaya gerek duyulmamaktadır. Gerekli parametreler ile oynayarak spektrum üzerinde bulunan pik sayısı artırılarak hatayı oldukça azaltabilmekteyiz. Rastgele oluşan pikler, taban kaymasından dolayı sinyal düzeyindeki değişimler pik olarak değerlendirildiğinde bulunan standart hata değeri daha da düşmektedir.

İyi bir spektrum grafiğinin çok miktarda veri içermesi gerekmektedir. Veri sayısı fazla olan örneklerde hesaplama daha iyi sonuç vermesine karşı geliştirilen yazılım düşük sayıda veri içeren örneklerde de iyi sonuçlar vermektedir. Bu sorun bütün analitik sinyal değerlendirmelerinde bilinen bir sorundur.

Kompleks bir NMR spekturumunda 100'ün üzerinde pik olduğu varsayılırsa, programın paralel olarak çalışması ve ayrıca hesaplama süresini kısaltması yazılım açısından önemli bir avantajdır.

Geliştirdiğimiz algoritma ve yazılım NMR, UV ve IR verileri üzerinde oldukça başarılı olmuş ve HPLC (Yüksek Performanslı Sıvı Kromatografi), GC (Gaz Kromatografi), CE (Kapiler Elektroforez) gibi ayırma tekniklerinin kromatogramları ile ESR (Elektron Spin Rezonans), XRF (X-Işınları Floresansı) gibi spektroskopik tekniklerin spektrumlarında ayrıca Polarografi, Voltametri tekniklerinin zaman-potansiyel-akım eğrilerinde de başarılı sonuçlar vereceği kanısındayız.

Analitik cihazlar beraberinde verilen ölçme değerlendirme yazılımları genelde yeterli düzeyde değildir. Geliştirdiğimiz algoritma ve yazılım ülkemizde üretilecek herhangi bir analitik cihazın sinyallerini değerlendirmek için verilebilecek yazılımda kullanılabilecek açık kaynak kodlu bir yazılımdır. Ölçme ve değerlendirme işlemlerini grafiksel ara yüz ile yapan yazılımın kaynak kodu çok uzun olduğundan ek olarak verilmemiştir.

Bu konuda gelecekteki çalışmalarımız sinyallerin gerçek zamanlı değerlendirilmesi için analog veya paralel çalışan gömülü sistemler üzerinde olacaktır.

6. KAYNAKLAR

- [1] P. Geladi, B.R. Kowalski, Partial Least Squares Regression (PLS): a Tutorial, *Analytica Chimica Acta*, 185, 1986.
- [2] M. Mori, C. C. Tseng, A genetic algorithm for multi-mode resource constrained project scheduling problem, *European Journal of Operational Research*, V.100, No 1, P.134-141, 1997.
- [3] L. J. Schaffer, R. A. Das, Proceedings of the Third International Conference on Genetic Algorithms, San Mateo, p. 51-60, 1989.
- [4] T. S. Felix, S. Wadhwa, A hybrid genetic algorithm for production and distribution, *Omega-The International Journal of Management Science*, p.345-555, 2005.
- [5] M. Chen, S. Zazala, A genetic approach to motion planning of redundant mobile manipulator systems considering safety and configuration, *Journal of Robotics Systems*, Vol. 14, p. 529-544, 1997.
- [6] Obitko, Introduction to genetic algorithms with Java applets, 1998.
- [7] M. Nakamura, N. Yamashiro, Y. Gong, T. Matsumura, K. Onaga, Iterative Parallel Genetic Algorithms Based on Biased Initial Population, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, V. E88-A, p. 923-929, 2005.
- [8] T. Ikeuchi, Y. Ikkai, D. Araki, T. Ohkawa and N. Komoda, A learning method of scheduling knowledge by genetic algorithm, *Electrical Engineering in Japan*, Vol. 124, No. 2, s43-51, 1998.
- [9] G. Emel, Ç. Taşkın, 2002. Uludağ Üniversitesi İ.İ.B.F. Dergisi, Cilt XXI, Sayı 1, 129-152, 2001.

- [10] Z. Şen, Genetik algoritma ve En İyileme Yöntemleri, İstanbul, Su Vakfı, 2005.
- [11] C. Reeves, Genetic Algorithms. In: Handbook of Metaheuristics. (Glover, F.,-ed.), pp.55-83, Kluwer Academic Publishers, NJ, ABD., 2002.
- [12] R.Y.K. Fung, J. Tang and D. Wang, Extension of A Hybrid Genetic Algorithm For Non-linear Programming Problems with Equality and Inequality Constraints. Computers & Operations Research, Vol. 29, Is. 3, s. 261-274, 2001.
- [13] Kedar S. Naphade, S. David Wu, Robert H. Storer, Problem space search algorithms for resource constrained project scheduling, Annals of Operations Research, 70 ,ss.307-326, 1997.
- [14] P. Brucker, A. Drexl, R. Möhring, K. Neumann and E. Pesch, Resource-constrained project scheduling: Notation, classification, models and methods, European Journal of Operational Research, Vol. 112, No. 1, s.3-41, 1999.
- [15] R.L. Haupt and S. E. Haupt, Practical Genetic Algorithms, New Jersey: Second edition, Jhon Wiley & Sons Inc., 2004.
- [16] M. Mitchell, and C.E. Taylor, Evolutionary computation: An Overview, Annual Review of Ecology ad Systematics, 30: s.593-616, 1999.
- [17] D. E.Goldberg, Genetic Algorithms in Search Optimization and Machine Learning. Addison Wesley. s. 41, 1989.
- [18] B. Bolat, E.C. İmsak, Mühendislik ve Fen Bilimleri Dergisi, Sigma 2004/4, ss.264-271, 2004.
- [19] N. Karaboğa, Sayısal Filtre Katsayılarının Genetik Algoritma ile Yuvarlatılması, Doktora Tezi, Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Kayseri, 1994.

- [20] A. Bağış, Genetik Algoritma Kullanılarak Ders Programının Optimum Şekilde Düzenlenmesi, Yüksek Lisans Tezi, Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Kayseri, 1996.
- [21] Y. Arslan, Mühendislikte Tersine Problem Uygulamaları İçin Genetik Programlama Yaklaşımı, Yüksek Lisans Tezi, Zonguldak Karaelmas Üniversitesi Fen Bilimleri Enstitüsü, 2005.
- [22] G. G. Emel, Ç. Taşkın, Genetik Algoritmalar ve Uygulama Alanları, Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi Cilt XXI, Sayı 1, s. 129-152, 2002.
- [23] O. Engin, A. Fırlalı, Genetik Algoritmalarla Akış Tipi Çizelgelemede Üreme Yöntemi Optimizasyonu, İstanbul Teknik Üniversitesi Dergisi/d Mühendislik Cilt:1 Sayı:1, 2002.
- [24] M. E. Jackson, Optimal Desing of Discrete-Time Delta Sigma Modulators, Yüksek Lisans Tezi, Graduate College University of Nevada, Las Vegas, 2009.
- [25] S. D. Bungay, Optimization of Transition State Structures Using Genetic Algorithms, Yüksek Lisans Tezi, Departments of Chemistry, Mathematics and Statistics and Computer Science Memorial University of Newfoundland, 2000.
- [26] T. Emre KALAYCI, Yapay Zeka Kullanan Üç Boyutlu Grafik yazılımları İçin “Extensible 3D” (X3D) ile Bir Altyapı Oluşturulması ve Gerçekleştirimi, Yüksek Lisans Tezi, Ege Üniversitesi Fen Bilimleri Enstitüsü, 2006.
- [27] S. Handschuh, M. Wagner and J. Gasteiger, J. Superposition of Three-Dimensional Chemical Structures Allowing for Conformational Flexibility by a Hybrid Method, Chem. Inf. Comput. Sci., 38(2), 220-232, 1998.

- [28] R.S. Judson, E.P. Jaeger, A.M. Treasurywala and M.L. Peterson, Conformational searching methods for small molecules. II. Genetic algorithm approach, *J. Comput. Chem.*, 14(11), 1407-1414, 1993.
- [29] M. Karakaplan, Fitting Lorentzian peaks with evolutionary genetic algorithm based on stochastic search procedure, *Analytica Chimica Acta*, 587:2, 235-239, 2007.
- [30] A. M. Thompson, B. I. Dunlap, Optimization of analytic density functionals by parallel genetic algorithm, *Chemical Physics Letters*, Vol. 463, Is. 1-3, Pages 278-282, 2008.
- [31] N. Sim, D. Konovalov, D. Coomans, High-Performance GRID Computing in Chemoinformatics, *Comprehensive Chemometrics*, Chapter 4.15, Pages 507-539, 2009.
- [32] I. Kuzmanovski, M. Novic, M. Trpkovska, Automatic adjustment of the relative importance of different input variables for optimization of counter-propagation artificial neural networks, *Analytica Chimica Acta*, Vol. 642, Is. 1-2, Pages 142-147, 2009.
- [33] J. W. Allwood, C. Deborde, Y. Xu, M. Maucourt, B. Beauvoit, W. B. Dunn, D. Jacob, R. Goodacre, D. Rolin and A. Moing, 1H NMR, GC–EI-TOFMS, and Data Set Correlation for Fruit Metabolomics: Application to Spatial Metabolite Analysis in Melon Benoit Biais, *Anal. Chem.*, 81 (8), pp 2884–2894, 2009.
- [34] R. Leardi, Genetic algorithms in chemistry, *Journal of Chromatography*, 1158, 226–233, 2008.
- [35] J. Workman, M. Koch, B. Lavine and R. Chrisman, Process Analytical Chemistry, *Anal. Chem.*, 81 (12), pp 4623–4643, 2009.
- [36] W. Rivera, Scalable Parallel Genetic Algorithms, *Artificial Intelligence Review*, 16: 153–168, 2001.

- [37] L.A. Smith, *Applied Infrared Spectroscopy*, John Wiley & Sons., 322s. USA, 1979.
- [38] P. Vankeerberghen, J. Smeyers-Verbeke, R. Leardi, C.L. Karr and D.L. Massart, Robust regression and outlier detection for non-linear models using genetic algorithms, *Chem. Intell. Lab. Syst.*, 28(1), 73-87, 1995.
- [39] A.P. de Weijer, L. Buydens, G. Kateman and H.M. Heuvel, Spectral curve fitting of infrared spectra obtained from semi-crystalline polyester yarns, *Chemom. Intell. Lab. Syst.*, 28(1), 149-164, 1995.
- [40] H. Ahonen, P. A. de Souza and VK. Garg, *Nuclear Instruments and Methods in Physics Research Section A.*, Sect. B, 124(4), 633-638, 1997.
- [41] D. A. Pearlman, A new genetic algorithm-based method for fitting NMR data, *J. Biomol. NMR*, 8(1), 49-66, 1996.
- [42] D. J. Mikulin, D.A. Coley and J.R. Sambles, Fitting reflectivity data from liquid crystal cells using genetic algorithms, *Liquid Crystals*, 22(3), 301-307, 1997.
- [43] B. M. Wise, B. R. Holt, N. B. Gallagher and S. Lee, A comparison of neural networks, non-linear biased regression and a genetic algorithm for dynamic model identification, *Chemom. Intell. Lab. Syst.*, 30(1), 81-89, 1995.
- [44] B. E. Mitchell and P.C. Jurs, Prediction of Infinite Dilution Activity Coefficients of Organic Compounds in Aqueous Solution from Molecular Structure, *J. Chem. Inf. Comput. Sci.*, 38(2), 200-209, 1998.
- [45] D. Brynn Hibbert, A hybrid genetic algorithm for the estimation of kinetic parameters, *Chemom. Intell. Lab. Syst.*, 19(3), 319-329, 1993.

- [46] R. Moros, H. Kalies, H. G. rex and St. Schaffarczyk, A genetic algorithm for generating initial parameter estimations for kinetic models of catalytic processes, *Comput. Chem. Eng.*, 20(10), 1257-1270, 1996.
- [47] D. Wolf and R. Moros, Estimating rate constants of heterogeneous catalytic reactions without supposition of rate determining surface steps - An application of a genetic algorithm, *Chem. Eng. Sci.*, 52(7), 1189-1199, 1997.
- [48] M. K. Hartnett, M. Bos, W. E. van der Linden, and D. Diamond, Determination of stability constants using genetic algorithms, *Analytica Chimica Acta*, 316(3), 347-362, 1995.
- [49] A. Parczewski, B. C. Lucasius and G. Kateman, Evolutionary determination of physico-chemical parameters and concentrations of analytes from titration data, *J. Anal. Chem.*, 348(10), 626-632, 1994.
- [50] R. E. Shaffer and G. W. Small, Comparison of optimization algorithms for piecewise linear discriminant analysis: application to Fourier transform infrared remote sensing measurements, *Analytica Chimica Acta*, 331(3), 157-175, 1996.
- [51] R. E. Shaffer and G. W. Small, Genetic algorithms for the optimization of piecewise linear discriminants, *Chemom. Intell. Lab. Syst.*, 35(1), 87-104, 1996.
- [52] A.H.C. van Kampen, Z. Ramadan, M. Mulholland, D.B. Hibbert and L.M.C. Buydens, Learning classification rules from an ion chromatography database using a genetic based classifier system, *Analytica Chimica Acta*, 344(1-2), 1-16, 1997.
- [53] H. Yoshida and K. Funatsu, Optimization of the Inner Relation Function of QPLS Using Genetic Algorithm, *J. Chem. Inf. Comput. Sci.*, 37(6), 1115-1121, 1997.

- [54] M. Hartnett and D. Diamond, Potentiometric Non-linear Multivariate Calibration with Genetic Algorithm and Simplex Optimization, *Analytica Chimica Acta*, 69(10), 1909-1918, 1997.
- [55] R.E. Shaffer, G.W. Small and M.A. Arnold, Genetic Algorithm-Based Protocol for Coupling Digital Filtering and Partial Least-Squares Regression: An Application to the Near-Infrared Analysis of Glucose in Biological Matrices, *Analytica Chimica Acta*, 68(15), 2663-2675, 1996.
- [56] A.P. de Weijer, C.B. Lucasius, L. Buydens, G. Kateman and H.M. Heuvel, Using genetic algorithms for an artificial neural network model inversion, *Chem. Intell. Lab. Syst.*, 20(1), 45-65 , 1993.
- [57] S.S. So and M. Karplus, Three-dimensional quantitative structure-activity relationships from molecular similarity matrices and genetic neural networks. 1. Method and validations, *J. Med. Chem.*, 40(26), 4347-4359, 1997.
- [58] J.H. Jiang, J.H. Wang, X.H. Song and R.Q. Yu, Network training and architecture optimization by a recursive approach and a modified genetic algorithm, *J. Chemom.*, 10(3), 253-267, 1996.
- [59] S.S. So and M. Karplus, Evolutionary optimization in quantitative structure-activity relationship: An application of genetic neural network, *J. Med. Chem.*, 39(7), 1521-1530, 1996.
- [60] F.R. Burden, B.S. Rosewame and D.A. Winkler, Predicting maximum bioactivity by effective inversion of neural networks using genetic algorithms, *Chemom. Intell. Lab. Syst.*, 38(2), 127-137, 1997.
- [61] A. Broudiscou, R. Leardi and R. Phan-Tan-Luu, Genetic algorithm as a tool for selection of D-optimal design, *Chemom. Intell. Lab. Syst.*, 35(1), 105-116, 1996.

- [62] C.B. Lucasius, A.D. Dane and G. Kateman, On k-medoid clustering of large data sets with the aid of a genetic algorithm: background, feasibility and comparison, *Analytica Chimica Acta*, 282(3), 647- 669, 1993.
- [63] J.H. Jiang, J.H. Wang, X. Chu and R.-Q. Yu, Clustering data using a modified integer genetic algorithm (IGA), *Analytica Chimica Acta*, 354(1-3), 263-274 , 1997.
- [64] R. Leardo, Application of a genetic algorithm to feature selection under full validation conditions and to outlier detection, *J. Chemom.*, 8(1), 65-79, 1994.
- [65] R. Leardi, R. Boggia and M. Terrile, Genetic algorithms as a strategy for feature selection, *J. Chemom.*, 6(5), 267-281, 1992.
- [66] L.G. Weyer and S.D. Brown, Applications of new variable selection techniques to near infrared spectroscopy, *Journal of Near Infrared Spectroscopy*, 4(1-4), 163-174, 1996.
- [67] U. Hoerchner, J.H. Kalivas, Further investigation on a comparative study of simulated annealing and genetic algorithm for wavelength selection, *Analytica Chimica Acta*, 311(1), 1-13, 1995.
- [68] J.M. Benchley, U. Horchner, and J.H. Kalivas, Wavelength Selection Characterization for NIR Spectra, *Applied Spectroscopy*, 51(5), 689-699, 1997.
- [69] A.S. Bangalore, R.E. Shaffer, G.W. Small and M.A. Arnold, Genetic algorithm-based method for selecting wavelengths and model size for use with partial least-squares regression: Application to near-infrared spectroscopy, *Anal. Chem.*, 68(23), 4200-4212, 1996.
- [70] M. Acros, M. Ortiz, B. Villahoz and L.A. Sarabia, Genetic-algorithm-based wavelength selection in multicomponent

- spectrometric determinations by PLS : application on indomethacin and acemethacin mixture, *Analytica Chimica Acta*, 339(1- 2), 63-77, 1997.
- [71] H. Kubinyi, Variable Selection in QSAR Studies. I. An Evolutionary Algorithm, *Quant. Struct. Act. Relat.*, 13(3): 285–294, 1994.
- [72] K. Hasegawa, Y. Miyashita and K. Funatsu, GA Strategy for Variable Selection in QSAR Studies: GA-Based PLS Analysis of Calcium Channel Antagonists, *J. Chem. Inf. Comput. Sc.*, 37(2), 306-310, 1997.
- [73] D. Jouan-Rimbaud, D.L. Massart, R. Leardi and O.E. Noord, Genetic Algorithms as a Tool for Wavelength Selection in Multivariate Calibration, *Analytica Chimica Acta*, 7(23), 4295-4301, 1995.
- [74] R.P. Paradkar R.R. Williams, Genetic Regression as a Calibration Technique for Solid-Phase Extraction of Dithizone-Metal Chelates, *Applied Spectroscopy*, 50(6), 753-758, 1996.
- [75] D. Broadhurst, R. Goodacre, A. Jones, J.J. Rowland, and D.B. Kell, Genetic algorithms as a method for variable selection in multiple linear regression and partial least squares regression, with applications to pyrolysis mass spectrometry, *Analytica Chimica Acta*, 348(1-3), 71-86, 1997.
- [76] T. Aishima, N. Togari, and R. Leardi, Differentiation of Soy Sauce by Pattern Recognition Analysis of Mid- and Near-IR Spectra, *Food Sc. Technol. Int.*, 2(2), 124-126, 1996.
- [77] R. Wehrens, C. Lucasius, L. Buydens and G. Kateman, HIPS, a hybrid self-adapting expert system for nuclear magnetic resonance spectrum interpretation using genetic algorithms, *Analytica Chimica Acta*, 277(2), 313-324, 1993.

- [78] D.J. Wild and P. Willet, Similarity Searching in Files of Three-Dimensional Chemical Structures. Alignment of Molecular Electrostatic Potential Fields with a Genetic Algorithm, *J. Chem. Inf. Comput. Sci.*, 36(2), 159-167, 1996.
- [79] M. Wagener and J. Gasteiger, Determination of maximum common 3D substructures using a genetic algorithm, *AIP Conf. Proc.*, 330, 532-537, 1995.
- [80] R.D. Brown, G. Jones, P. Willet and R.C. Glen, Matching two-dimensional chemical graphs using genetic algorithms, *J. Chem. Inf. Comput. Sci.*, 34(1), 63-70, 1994.
- [81] J. Antel, I. Reuter and D. Schomburg, *Comput-Assisted Lead Find. Optim.*, H. Van de Watebeemd, Eds., Gerd Publishers, Verlag Helvetica Chimica Acta, Basel, Switz, 1997.
- [82] L. Weber, S. Wallbaum, C. Broger and K. Gubemator, Optimization of the Biological Activity of Combinatorial Compound Libraries by a Genetic Algorithm, *Angew. Chem.*, 34(20), 2280-2282, 1995.
- [83] D. Tomandl, A. Schober and A. Schwienhorst, Optimizing doped libraries by using genetic algorithms, *J. Comput.-AidedMol. 'Des.*, 11(1), 29-38, 1997.
- [84] R.D. Brown and Y.C. Martin, Designing Combinatorial Library Mixtures Using a Genetic Algorithm, *J. Med Chem.*, Vol. 40, Is. 15, 2304-2313, 1997.
- [85] D. Liu, H. Jiang, K. Chen and R.Y. Ji, A New Approach to Design Virtual Combinatorial Library with Genetic Algorithm Based on 3D Grid Property, *J. Chem. Inf. Comput. Sci.*, 38(2), 233- 242, 1998.
- [86] S.J. Cho, W. Zheng and A. Tropsha, Rational Combinatorial Library Design. 2. Rational Design of Targeted Combinatorial Peptide Libraries

- Using Chemical Similarity Probe and the Inverse QSAR Approaches, J. Chem. Inf. Comput. Sci., 38(2), 259- 268, 1998.
- [87] R.P. Sheridan and S.K. Kearsley, Using a Genetic Algorithm To Suggest Combinatorial Libraries, J. Chem. Inf. Comput. Sci., 35(2), 310-320, 1995.
- [88] Zhang et al., Optimal choice of parameters for particle swarm optimization, Journal of Zhejiang University, SC1, 6,4(6):528-534, 2005.
- [89] Gallica - Fourier, Jean-Baptiste-Joseph (1768-1830). Oeuvres de Fourier. 1888.
- [90] A. Graps, An Introduction to Wavelets, IEEE Computational Science & Engineering, Vol.2, Is.2, 50-61, 1995.
- [91] I. M. Tibuelac and I.E. Herrin, An Automatic Method for Determination of Lg Arrival Times Using Wavelet Transform, Seism. Res. Lett. Vol. 70, p. 577-595, 1999.
- [92] M. Vetterli, C. Herley, Wavelets and Filter Banks: Theory and Design, IEEE Transactions on Signal Processing. Vol. 40, No. 9, 1992.
- [93] C.S. Burrus, R.A. Gopinath, H. GUC, Introduction to Wavelets and Wavelet Transforms, Texas, Prentice Hall, 1998.
- [94] I. Daubechies, The Wavelet Transform-Frequency Localization and Signal Analysis, IEEE Transactions on Information Theory, 36, 5, 961-1005, 1990.
- [95] A. Byrski and M. Kisiel-Dorohinicki, Agent-Based Evolutionary and Immunological Optimization, Computational Science – ICCS 2007 Lecture Notes in Computer Science, 2007, Vol. 4488, 928-935, 2007.
- [96] O. Montiel, O. Castillo, P. Melin, A. R. Diaz, R. Sepulveda, Human evolutionary model: A new approach to optimization, Information Sciences, Vol. 177, Is. 10, Pages 2075-2098, 2007.

- [97] J. Yisu, J. Knowles, L. Hongmei, L. Yizeng, D. B. Kell, The landscape adaptive particle swarm optimizer, *Applied Soft Computing*, Vol. 8, Is. 1, Pages 295-304, 2008.
- [98] K.E. Parsopoulos, M.N. Vrahatis, Parameter selection and adaptation in Unified Particle Swarm Optimization, *Mathematical and Computer Modelling*, Vol. 46, Is. 1-2, 2007.
- [99] E. Alba and J. M. Troya, A survey of parallel distributed genetic algorithms. *Complexity (USA)*, 4(4):31–52, 1999.
- [100] E. Alba, *Parallel Metaheuristics: A New Class of Algorithms*. Wiley Interscience, 2005
- [101] D. Dumitrescu, B. Lazzerini, L. C. Jain, and A. Dumitrescu. *Evolutionary Computation. The CRC Press International Series on Computational Intelligence*. CRC Press, 2000.
- [102] H. Mühlenbein. Parallel genetic algorithms, population genetics and combinatorial optimization. *Proceedings of the 3rd International Conference on Genetic Algorithms*, pages 416– 421, 1989.
- [103] M. Asiltürk, F. Sayilkan, E. Burunkaya, E. Arpac, “Hydrothermal Synthesis and Characterization of Nanocrystalline ZrO_2 and Surface Modification with 2-acetoacetoxyethyl methacrylate” *J Sol-Gel Sci. Techn.*, 51(2), 182-189, 2009.
- [104] <http://webbook.nist.gov/cgi/cbook.cgi?ID=C76846&Units=SI&Mask=400>, 2011.
- [105] <http://webbook.nist.gov/cgi/cbook.cgi?ID=C14405493&Units=SI&Mask=400>, 2011.
- [106] <http://webbook.nist.gov/cgi/cbook.cgi?ID=C54000656&Units=SI&Mask=400>, 2011.

- [107] <http://science.gsfc.nasa.gov/691/cosmicice/spectra.html>, 2011.
- [108] M. Kaloğlu, N-Heterosiklik Karben Katalizli Aminasyon, Yüksek Lisans Tezi, İnönü Üniversitesi, Fen Bilimleri Enstitüsü, 2011.

EKLER

A. Makale ve Bildiriler

- M. Karakaplan, F. M. Avcu, A parallel and non-parallel genetic algorithm to resolve peaks in NMR spectra, Submitted to *Analytica Chimica Acta*, 2012
- M. Karakaplan, F. M. Avcu, Paralel Genetik Algoritma İle Çok Merkezli Fonksiyonların Optimizasyonu, 3. Ulusal Yüksek Başarımlı Hesaplama Konferansı'na sunulmak üzere gönderildi.

B. Geliştirilen Bilgisayar Programının Kaynak Kodu

```
//ga.cpp dosyası

#include <QtCore/QCoreApplication>

#include "ga.h" #include <sqlite3.h> #include <stdio.h> #include
<stdlib.h>

#include <time.h> #include <math.h> #include <float.h> #include
<mpi.h>

#include <iostream>

using namespace std; void dataRead()

{ cout <<"Data Reading...."<< endl; int i;int resp; float xrv,yrv; FILE
*fp_sporig;

    if((fp=fopen("kabaasi.dat","r+"))==NULL) { cout <<"Dosya Açılamadı"
<< endl;

    exit(1); }

    else { double toplam=0.0;i=0; while (!feof(fp)) {

    resp=fscanf(fp,"%f",&xrv); resp=fscanf(fp,"%f",&yrv); xdata[i]=(double)
xrv;

    ydata[i]=(double) yrv/1.0e6; toplam=toplam+(double) yrv/1.0e6;

    i++; } NPOINT=i; cout << NPOINT << endl; double
eski_egim,egim,alan,yamuk,pikmy;

    int eski_indis;

    int piksay=0; alan=0.0; pikmy=-9999999.0;double piklx=0.0;double
eski_pik=0.0;

    eski_egim=0.0; eski_indis=0;
```

```

fp_sporig=fopen("giris_dosyas1.dat","w+"); for (i=0;i<NPOINT;i++)
{
    fprintf(fp_sporig,"%f\t%f\n",    xdata[i],    ydata[i]);    pikmy=ydata[i];
piklx=xdata[i];

    egim=(ydata[i+1]-ydata[i])/(xdata[i+1]-xdata[i]);

    yamuk=fabs((xdata[i+1]-xdata[i])*ydata[i])+fabs(((xdata[i+1]-xdata[i])*(ydata[i+1]-ydata[i])));

    alan=alan+yamuk; if (sign2(egim) != sign2(eski_egim) && (i-eski_indis)>1
&& pikmy>1)

    { pikx[piksay]=piklx;piky[piksay]=pikmy;pikh[piksay]=alan/50.0;

    cout << piksay << ".    " << "\tx:" << piklx << "\ty:" << pikmy <<
"\tegim:" << egim << endl;;

    piksay++;alan=0.0;pikmy=-9999999.0;eski_pik=piklx;eski_indis=i;}eski_egim=egim;}

fclose(fp_sporig);NUM_OF_PEAK=piksay;}

cout <<"Data Read Ok\t" << NUM_OF_PEAK <<" peaks are counted"
<< endl;}

void    calculateSpectrum(){int    i,n;double    yy;for(i=0;i<NPOINT;i++)
{yy=0.0;

    for (n=0;n<NUM_OF_PEAK;n++){

yy=yy+pops->a[n]/pow(((1.0+4.0*pow(((xdata[i]-pops->v[n])/pops->h[n]),2.0)

*(pow(2,(1/pops->m[n]))-1)),pops->m[n]);}yteoric[i]=yy;}

void PrintSpectrum(){FILE *fp_spec, *fp_pparams;int i,n;double yy;

fp_spec=fopen("spectrum.dat","w+");for(i=0;i<NPOINT;i++) {yy=0.0;

    for (n=0;n<NUM_OF_PEAK;n++){if (type==1)

yy=yy+pop->a[n]/pow(((1.0+4.0*pow(((xdata[i]-pop->v[n])/pop->h[n]),2.0)*

(pow(2,(1/pop->m[n]))-1)),pop->m[n]);

//2 Gaussian

```

```

if (type==2) yy=yy+pop->a[n]*exp(-pow((xdata[i]-pop->v[n]),2.0)/(pop->h[n]*pop->h[n]
//3 Lorentzian

if (type==3)

yy=yy+((pop->a[n]*pop->h[n]*pop->h[n])/(pow(pop->h[n],2.0)+4.0*
pow((xdata[i]-pop->v[n]),2.0)));}

yteoric[i]=yy;fprintf(fp_spec,"%f\t%f\n", xdata[i], yteoric[i]);}

fclose(fp_spec);fp_pparams=fopen("bulunanpikler","w+");

for (n=0;n<NUM_OF_PEAK;n++){

if (type==1) fprintf(fp_pparams,"%d\t%f\t%f\t%f\t%f\n", n,
pops->a[n],

pops->v[n], pops->h[n], pops->m[n]);

if (type==2) fprintf(fp_pparams,"%d\t%f\t%f\t%f\n", n, pops->a[n],
pops->v[n], pops->h[n]);

if (type==3) fprintf(fp_pparams,"%d\t%f\t%f\t%f\n", n, pops->a[n],
pops->v[n], pops->h[n]);}fclose(fp_pparams);}

void gaReset(){oldr=9e300;pops=&population[0];pope=&population[NPOPUL-1];

popc=&population[NPOPUL];pp=&peaks[0];MPI_Get_address(
&population[0], &a1 );

MPI_Get_address( &population[1], &a2 );for
(i=0;i<NUM_OF_PEAK;i++){

pp->amed=0.0;pp->astd=0.0;pp->ast=0.0;pp->atot=0.0;

pp->a=piky[i]; pp->amin=piky[i]-0.001;pp->amax=piky[i]+0.001;

pp->vmed=0.0;pp->vstd=0.0;pp->vst=0.0;pp->vtot=0.0;

pp->v=pikx[i];pp->vmin=pikx[i]-0.002;pp->vmax=pikx[i]+0.002;

pp->hmed=0.0;pp->hstd=0.0;pp->hst=0.0;pp->htot=0.0;

pp->h=drand(0.001,0.01);pp->hmin=0.001;pp->hmax=0.005;

```

```

pp->mmed=0.0;pp->mstd=0.0;pp->mst=0.0;pp->mtot=0.0;

pp->m=drand(0.5,1.5);pp->mmin=pp->m-0.5;pp->mmax=pp->m+0.5;

pp++;}first_population();qsort(pops,NPOPUL,sizeof(generation),
compare_r);

calc_population_statistics();}double drand(double a, double b){double sayi;
sayi=(drand48()*(b-a))+a;return sayi;}

void calc_generation(int np){int n;double
yy,se;pop=&population[np];pp=&peaks[0];

for(n=0;n<NUM_OF_PEAK;n++){pop->a[n]=drand(pp->amin,pp->amax);
pop->v[n]=drand(pp->vmin,pp->vmax);pop->h[n]=drand(pp->hmin,pp->hmax);
pop->m[n]=drand(pp->mmin,pp->mmax);pp++;}

spp=&spectrump[0];pop->r=0.0;se=0.0;for (i=0;i<NPOINT;i++){yy=0.0;
pp=&peaks[0];for (n=0;n<NUM_OF_PEAK;n++){

//1 pearson IV

if (type==1)

yy=yy+pop->a[n]/pow((1.0+4.0*pow(((xdata[i]-pop->v[n])/pop->h[n]),2.0)*
(pow(2,(1/pop->m[n]))-1)),pop->m[n]);

//2 Gaussian

if (type==2) yy=yy+pop->a[n]*exp(-pow((xdata[i]-pop->v[n]),2.0)/
(pop->h[n]*pop->h[n]));

//3 Lorentzian

if (type==3)

yy=yy+((pop->a[n]*pop->h[n]*pop->h[n])/(pow(pop->h[n],2.0)+4.0*
pow((xdata[i]-pop->v[n]),2.0)));

pp++;}spp->yi=yy;se=se+((ydata[i] - spp->yi)*(ydata[i] -
spp->yi));spp++;}

```

```

pop->r=sqrt(se/(double)NPOINT);}

void first_population(void){int n;for (n=0;n<NPOPUL;n++){

calc_generation(n);pp=&peaks[0];for (np=0;np<NUM_OF_PEAK;np++){

pp->atot=pp->atot+pop->a[np];pp->ast=pp->ast+(pop->a[np]*pop->a[np]);

pp->vtot=pp->vtot+pop->v[np];pp->vst=pp->vst+(pop->v[np]*pop->v[np]);

pp->htot=pp->htot+pop->h[np];pp->hst=pp->hst+(pop->h[np]*pop->h[np]);

pp->mtot=pp->mtot+pop->m[np];pp->mst=pp->mst+(pop->m[np]*pop->m[np]);

pp++;}}

void statistics_add (){int n;pp=&peaks[0];for (n=0;n<NUM_OF_PEAK;

n++){

pp->atot          =          pp->atot+popc->a[n];pp->ast          =

pp->ast+popc->a[n]*popc->a[n];

pp->vtot          =          pp->vtot+popc->v[n];pp->vst          =

pp->vst+popc->v[n]*popc->v[n];

pp->htot          =          pp->htot+popc->h[n];pp->hst          =

pp->hst+popc->h[n]*popc->h[n];

pp->mtot          =          pp->mtot+popc->m[n];pp->mst          =

pp->mst+popc->m[n]*popc->m[n];

pp++;}}

void statistics_extract (){int n;pp=&peaks[0];for

(n=0;n<NUM_OF_PEAK;

n++){

pp->atot          =          pp->atot-popc->a[n];pp->ast          =

pp->ast-popc->a[n]*popc->a[n];

pp->vtot          =          pp->vtot-popc->v[n];pp->vst          =

pp->vst-popc->v[n]*popc->v[n];

```

```

        pp->htot          =          pp->htot-popc->h[n];pp->hst          =
pp->hst-popc->h[n]*popc->h[n];

        pp->mtot          =          pp->mtot-popc->m[n];pp->mst          =
pp->mst-popc->m[n]*popc->m[n];

        pp->amed = pp->atot/NPOPUL;pp->vmed = pp->vtot/NPOPUL;

        pp->hmed = pp->htot/NPOPUL;pp->mmed = pp->mtot/NPOPUL;

        pp->astd=sqrt( (pp->ast-((pp->atot*pp->atot)/NPOPUL))/NPOPUL);

        pp->vstd=sqrt( (pp->vst-((pp->vtot*pp->vtot)/NPOPUL))/NPOPUL);

        pp->hstd=sqrt( (pp->hst-((pp->htot*pp->htot)/NPOPUL))/NPOPUL);

        pp->mstd=sqrt( (pp->mst-((pp->mtot*pp->mtot)/NPOPUL))/NPOPUL);

        pp++;}}

        void          calc_population_statistics(){int          n;pp=&peaks[0];for
(n=0;n<NUM_OF_PEAK;

        n++){

        pp->astd=sqrt( (pp->ast-((pp->atot*pp->atot)/NPOPUL))/NPOPUL);

        pp->amed=pp->atot/NPOPUL;

        pp->vstd=sqrt( (pp->vst-((pp->vtot*pp->vtot)/NPOPUL))/NPOPUL);

        pp->vmed=pp->vtot/NPOPUL;

        pp->hstd=sqrt( (pp->hst-((pp->htot*pp->htot)/NPOPUL))/NPOPUL);

        pp->hmed=pp->htot/NPOPUL;

        pp->mstd=sqrt( (pp->mst-((pp->mtot*pp->mtot)/NPOPUL))/NPOPUL);

        pp->mmed=pp->mtot/NPOPUL;

        pp++;}}

        void          change_width_best(double          w){int          n;pp=&peaks[0];for
(n=0;n<NUM_OF_PEAK;

        n++)

```

```

    {pp->amin=pp->amed - w*pp->astd;    pp->amax=pp->amed +
w*pp->astd;

    pp->vmin=pp->vmed - w*pp->vstd; pp->vmax=pp->vmed + w*pp->vstd;

    pp->hmin=pp->hmed - w*pp->hstd;    pp->hmax=pp->hmed +
w*pp->hstd;

    pp->mmin=pp->mmed - w*pp->mstd;    pp->mmax=pp->mmed +
w*pp->mstd;

    if (pp->mmin<0.0) pp->mmin=0.0;pp++;}}int compare_r(const void *a,
const void *b){

    struct generation *ia = (struct generation *)a;

    struct generation *ib = (struct generation *)b;

    if (ia->r > ib->r) return(1);else return(-1);}

    void compare_new_generation(){ if (popc->r < pope->r){statistics_add();

    qsort(pops,NPOPUL+1,sizeof(generation),compare_r);

    statistics_extract();if (pops->r < oldr){oldr=pops->r;}

    change_width_best(alfa);}}

    void Hesapla(){while(iter<NITER){if( iter % INTERVAL ==0 && iter >
5){

        if(island==1) {

            if (nid<8) MPI_Send(pops,    a2-a1,    MPI_CHAR,    nid+1,    99,
MPI_COMM_WORLD);

            if (nid==8) MPI_Send(pops,    a2-a1,    MPI_CHAR,    0,    99,
MPI_COMM_WORLD);

            if (nid>1) MPI_Recv(&population[NPOPUL], a2-a1,

            MPI_CHAR, nid-1,99,MPI_COMM_WORLD,&status);

            if (nid==0) MPI_Recv(&population[NPOPUL],

```

```

a2-a1, MPI_CHAR, 8,99,MPI_COMM_WORLD,&status);

compare_new_generation();} }

MPI_Barrier(MPI_COMM_WORLD); }

else {calc_generation(NPOPUL);compare_new_generation(); }

if( iter % 200 ==0 && iter > 100){

    cout << "alt:" << INTERVAL << "\trun:" << run << "\ttype:" << type
<< "\tisle:" << island << "\tproc:"<< nid << "\titer #" << iter << "\talfa:"
<< alfa << "\tpopsize:"<< NPOPUL << "\tpops:" << pops->r << endl;

    }iter++;}}

void print_statistics ()

{int n;pp=&peaks[0];

for (n=0;n<NUM_OF_PEAK;n++){

piksy[n]=pops->a[n];piksx[n]=pops->v[n];pp++;}}

int main(int argc, char *argv[])

{QCoreApplication a(argc, argv);

randomize();MPI_Init(&argc, &argv);

MPI_Comm_size(MPI_COMM_WORLD, &noprocs);

MPI_Comm_rank(MPI_COMM_WORLD, &nid);

cout << "Processor ID: " << nid;

cout << " Size of generation: " << sizeof(generation) << " Size of Double:
" << sizeof(double);

dataRead();nan_number=0;NITER=200001; NPOPUL=20;

type=2;INTERVAL=200;alfa=1.90;island=1;iter=0;totiter=0;

gaReset();Hesapla();

MPI_Barrier(MPI_COMM_WORLD);

if (nid==0) PrintSpectrum();

```

```

MPI_Finalize();

}

//ga.cpp dosya sonu

//ga.h dosyası

#ifndef GA_H

#define GA_H

#include <stdlib.h>#include <stdio.h>#include <mpi.h>#include
<sqlite3.h>

#define NRUN 10 #define WIDTH 1.5 #define NPEAK 110

#define NPOPULM 20 #define REPN 20

#ifndef M_PI #define M_PI 3.14159265354

#endif

#define noise(per) ((drand48()*4.0)-2.0)*per

#define randomize() srand48((unsigned)time(NULL))

#define max(a,b) (((a) > (b)) ? (a) : (b))

#define min(a,b) (((a) < (b)) ? (a) : (b))

#define sign2(x) (( x > 0 ) - ( x < 0 ))

FILE *fp;

int NPOINT; int NPOPUL;int NITER; int INTERVAL;

int island;int noprocs, nid;MPI_Status status;MPI_Aint a1, a2;

double xdata[10000];double ydata[10000];double yteoric[10000];

double yavg[10000];double pikx[110];double piky[110];

double pikh[110];double piksx[110];double piksy[110];

int rank, size;int nan_number;struct point {double x,y,yi;};

struct point spectrump[10000];struct point *spp;

```

```

struct pparam {double a,amin,amax,ac,amed,astd,ast,atot,
v,vmin,vmax,vc,vmed,vstd,vst,vtot,
h,hmin,hmax,hc,hmed,hstd,hst,htot,
m,mmin,mmax,mc,mmed,mstd,mst,mtot;};

struct pparam peaks[NPEAK]; struct pparam *pp;

struct generation {double r,a[NPEAK],v[NPEAK],h[NPEAK],m[NPEAK];};

struct generation population [NPOPULM+1];

struct generation *pop,*pops,*pope,*popc,*p1,*p2,*p3;

double results[NRUN][REPN];double *rp;

double drand(double a, double b);

void first_population(void);void compare_new_generation(void);

void calc_population_statistics(void);void calc_generation(int gen);

void statistics_add (void);void statistics_extract (void);

void change_width_best(double alfa);

static int compare_r(const void *, const void *);

void parametersSet(void);void dataRead(void);

void calculateSpectrum(void);void print_statistics (void);

int i,p;int iter,totiter,run,nhit, bhit;double w,drift_level;

int cm,np,aiter;double oldr;double xx,yy,yyo,maxr,minr;

double alfa;double minx,maxx,minz;int type;

double cminx,cmaxx, cminy,cmaxy, best_resp;int DRIFT;

sqlite3 *db;int rc;char *db_err;char sorgu[300];

void gaReset();void Hesapla();

#endif

```

Özgeçmiş

Fatih Mehmet AVCU, 1979 yılında Malatya’da doğmuş, İnönü İlkokulunu, Hasan Varol Orta Okulunu ve Malatya Lisesini bitirdikten sonra 2001 yılında İnönü Üniversitesi, Eğitim Fakültesi, Kimya Öğretmenliği Bölümünden mezun olmuştur.

2001 yılından beri İnönü Üniversitesinde Okutman olarak göreve yapmakta olan AVCU, 2005 yılında Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Kimya Anabilim Dalında Yüksek Lisans öğrenimini tamamlamıştır.

İnönü Üniversitesinde yapılan bir çok kongre ve çalıştayda yerel komisyonda bulunmuş, Üniversitemiz BAP birimince desteklenen 2 projede araştırmacı olarak görev yapmıştır. Kendi adına bir kongre CD’si, SCI kapsamında 2 adet makalesi ve ulusal konferanslarda sunulmuş 1 adet bildirisi bulunmaktadır.

E-Posta Adresi: **fatihavcu@gmail.com**

Bu tez L^AT_EX[‡] ile Linux[§] işletim sisteminde hazırlanmıştır.

[‡]L^AT_EX bilimsel rapor, tez, makale hazırlama programıdır.

[§]Linux İnternet üzerinden geliştirilen ve serbestçe dağıtılan bir işletim sistemidir.