

**Eleman Sayısı Kısıtlı Portföy Optimizasyonu Problemi için
Etkin Algoritmaların Tasarlanması**

Program Kodu: 3501

Proje No: 214M224

Proje Yürütücüsü:

Yrd. Doç. Dr. Can Berk KALAYCI

Araştırmacı(lar):

Yrd. Doç. Dr. Olcay POLAT

Danışman(lar):

Prof. Dr. Aşkİner Güngör

Bursiyer(ler):

Ökkeş Ertenlice

Hasan Akyer

Leyla Polat

NİSAN 2017

DENİZLİ

Önsöz

Bu projede, literatürde hesaplama karmaşıklığının yüksek olduğu ispatlanmış ve NP-Zor sınıfında bir problem olarak yer alan eleman sayısı kısıtlı portföy optimizasyonu problemi ele alınmıştır. Bu problemin çözümü için metasezgisel tekniklerden yararlanılarak etkin algoritmaların geliştirilmesi öncelikli olarak amaçlanmıştır. Bilhassa, finans, matematik ve bilgisayar bilimi alanlarındaki birçok araştırmacı tarafından ilgi gören bu problemin çözümü için proje ekibi, iki yıllık proje süresi boyunca yoğun bir şekilde çalışarak, proje önerisinde öngörülen hedeflere ulaşmayı başarmıştır. Problemi hızlı ve etkin biçimde çözebilen algoritmaların, yapılan analizler ile ticari yazılıma dönüştürülebileceği ispatlanmış olması sayesinde, finans alanında yatırımcılara tavsiye niteliğinde öneriler getirebilecek, yeni bir şirket kurulumunun önünü açmıştır. Bu bulgular, projenin son aşamasında yapılan bir çalıştay da tartışılarak destek bulmuştur. Ayrıca, dünyadaki gelişmiş ülke endekslerin performans testlerine ek olarak, gelişmekte olan Türkiye piyasalarındaki BIST30 ve BIST100 verilerine de uygulanarak SCI-Expanded kapsamında taranan saygın bir dergide yayınlanması sayesinde Türkiye'deki endekslere ait veri setleri, Pamukkale Üniversitesi web alanında da yayınlanarak, uluslararası literatüre kazandırılmıştır. Projedeki bilgi ve tecrübelerin paylaşılması kapsamında, Pamukkale Üniversitesi lisans müfredatına IENG 473 kodlu Portföy Optimizasyonu isimli bir seçmeli ders eklenmiş ve bizzat proje yürütücüsü tarafından yönetilmiştir. Potansiyel genç araştırmacıların ilgisi çekilmiş, yüksek lisans ve doktora öğrencileri proje kapsamında yetiştirilmiştir. Son olarak, proje yürütücüsünün kariyerini geliştirmesi amacını da taşıyan bu proje, proje çıktılarıyla birlikte, proje yürütücüsünün ÜAK tarafından belirlenen doçentlik kriterlerini sağlmasına yardımcı olmuştur.

214M224 numaralı bu proje, 3501 - Kariyer Geliştirme Programı kapsamında TÜBİTAK tarafından desteklenmiştir.

İçindekiler

Önsöz	ii
İçindekiler	iii
Şekiller Dizini.....	v
Tablolar Dizini.....	vii
Özet.....	viii
Abstract	ix
1. GİRİŞ.....	1
2. LİTERATÜR ÖZETİ.....	2
2.1. Kesin çözüm algoritmaları.....	4
2.2. Yakınsama teknikleri.....	5
2.2.1. Metasezgiseller.....	5
2.2.1.1. Popülasyon temelli algoritmalar.....	5
2.2.1.1.1. Evrimsel algoritmalar.....	5
2.2.1.1.2. Sürü temelli algoritmalar.....	8
2.2.1.1. Tek çözüm temelli algoritmalar.....	11
2.3. Makine öğrenimi algoritmaları.....	13
3. GEREÇ VE YÖNTEM	15
3.1. Veri setlerinin toplanması.....	15
3.2. Matematiksel model, deneme çözümleri ve kesin çözüm algoritması	16
3.3. Literatürdeki mevcut yöntemlerin kodlanması.....	25
3.3.1. Algoritmalarda kullanılan ortak prosedürler	25
3.3.2. Genetik Algoritma (GA).....	27
3.3.3. Parçacık Sürü Optimizasyonu (PSO)	30
3.4. Yapay Arı Kolonisi (YAK) tekniğinin probleme uyarlanması	32
3.4.1. Tamir Prosedürü – II	34
3.4.2. Uyumsuz Çözüm Kontrol Prosedürü	34
3.4.3. Uyumsuzluğa İzin Veren Seçim Prosedürü	35

3.4.4. İyileştirilmiş YAK Algoritması (YAK-II)	36
3.5. Karınca Kolonisi Optimizasyonu (KKO) tekniğinin probleme uyarlanması.....	38
3.6. Değişken Komşuluk Arama (DKA) tekniğinin probleme uyarlanması	41
3.6.1 Yerel Arama Prosedürleri	41
3.6.1.1. Türev Bilgisi Kullanan Yerel Arama Prosedürleri.....	41
3.6.1.1.1. Steepest-Descent Algoritması	41
3.6.1.1.2. Fletcher-Reeves Algoritması	42
3.6.1.2. Türev Bilgisi Kullanmayan Yerel Arama Prosedürleri	43
3.6.1.2.1. Hooke-Jeeves Algoritması	43
3.6.1.2.2. Nelder-Mead Algoritması.....	46
3.6.2 DKA Algoritması	47
4. BULGULAR	50
4.1. YAK-II algoritmasının deneysel sonuçlar	50
4.1.1. Parametre Analizi	50
4.1.2. YAK-I ve YAK-II algoritmaları için deneysel sonuçlar	52
4.2. KKO algoritmasının deneysel sonuçlar	57
4.2.1. Parametre Analizi	57
4.2.2. KKO Algoritması İçin Deneysel Sonuçlar	59
4.3. DKA algoritmasının deneysel sonuçlar	64
4.3.1. Parametre Analizi	64
Nelder-Mead Algoritması	64
4.3.2. DKA Algoritması İçin Deneysel Sonuçlar.....	65
4.4. Karşılaştırmalı sonuçlar.....	71
4.4.1. Proje kapsamında geliştirilen algoritmaların hata performansı karşılaştırılması....	71
4.4.2. Literatürdeki diğer yöntemlerle hata performansı karşılaştırması	72
4.5. Borsa İstanbul Getiri Analizi.....	75
5. SONUÇ VE TARTIŞMA	90
Referanslar.....	95

Şekiller Dizini

Şekil 1: Yıllara göre ortalama-varyans modelli PO problemini çözmek için yapılan yayınların dağılımları.....	2
Şekil 2: Ortalama-Varyans temelli PO problemine uygulanan algoritmaların sınıflandırılması	3
Şekil 3: PO problemi çözümü için uygulanan tekniklerin dağılımı	14
Şekil 4: Hang Seng veri seti için kısıtsız etkin sınır	18
Şekil 5: Temsili bir hata hesaplama.....	20
Şekil 6: Hang Seng veri setinde $K = 10$ değeri için GAMS ile bulunan etkin sınır	23
Şekil 7: Hang Seng veri setinde farklı K değerleri için GAMS ile bulunan etkin sınırlar	24
Şekil 8: Uygunluk Hesaplama Prosedürü	26
Şekil 9: Uygunluk Hesaplama Prosedürü	27
Şekil 10: Genetik Algoritmalar'ın ana adımları	28
Şekil 11: Çaprazlama Prosedürü.....	28
Şekil 12: $BLX-\alpha$ çaprazlama operatörü	28
Şekil 13: Mutasyon Prosedürü	29
Şekil 14:Rulet tekeri	29
Şekil 15: PO problemi için önerilen Genetik Algoritma	30
Şekil 16: Parçacık Sürü Optimizasyonu algoritmasının ana adımları.....	31
Şekil 17: Parçacık Sürü Optimizasyonu Algoritması	32
Şekil 18:Tamir-II prosedürü	34
Şekil 19: Uyumsuz çözüm kontrol prosedürü	35
Şekil 20: Uyumsuzluğa izin veren seçim prosedürü	36
Şekil 21: YAK-II algoritması.....	37
Şekil 22: KKO algoritması ana adımları (Socha ve Dorigo 2008)	38
Şekil 23: KKO algoritmasında çözüm arşivinin yapısı (Liao vd. 2014)	38
Şekil 24: KKO algoritması	40
Şekil 25:Steepest-Descent algoritması	42
Şekil 26:Fletcher-Reeves Algoritması	43
Şekil 27: Hooke-Jeeves Algoritması.....	45
Şekil 28: Nelder-Mead Algoritması.....	47
Şekil 29: DKA algoritması.....	49
Şekil 30: Hata ölçütü için ana etki grafikleri.....	51
Şekil 31: Maksimum iterasyon sayısı için ana etki grafikleri.....	51
Şekil 32: Hang Seng veri seti için YAK-I ve YAK-II algoritmaları ile çizilen etkin sınırlar	52
Şekil 33: DAX 100 veri seti için YAK-I ve YAK-II algoritmaları ile çizilen etkin sınırlar	53

Şekil 34: FTSE 100 veri seti için YAK-I ve YAK-II algoritmaları ile çizilen etkin sınırlar	53
Şekil 35: S&P 100 veri seti için YAK-I ve YAK-II algoritmaları ile çizilen etkin sınırlar	54
Şekil 36: Nikkei veri seti için YAK-I ve YAK-II algoritmaları ile çizilen etkin sınırlar	54
Şekil 37: BİST 30 veri seti için YAK-I ve YAK-II algoritmaları ile çizilen etkin sınırlar	55
Şekil 38: BİST 100 veri seti için YAK-I ve YAK-II algoritmaları ile çizilen etkin sınırlar	55
Şekil 39: as, ilimit ve rR parametreleri için yapılan üç aşamalı parametre analizi	58
Şekil 40: Hang Seng veri seti için KKO algoritması ile çizilen etkin sınır	59
Şekil 41: DAX 100 veri seti için KKO algoritması ile çizilen etkin sınır	60
Şekil 42: FTSE 100 veri seti için KKO algoritması ile çizilen etkin sınır	60
Şekil 43: S&P 100 veri seti için KKO algoritması ile çizilen etkin sınır	61
Şekil 44: Nikkei veri seti için KKO algoritması ile çizilen etkin sınır	61
Şekil 45: BİST 30 veri seti için KKO algoritması ile çizilen etkin sınır	62
Şekil 46: BİST 100 veri seti için KKO algoritması ile çizilen etkin sınır	62
Şekil 47: Hang Seng veri seti için DKA algoritması ile çizilen etkin sınır	66
Şekil 48: DAX 100 veri seti için DKA algoritması ile çizilen etkin sınır	66
Şekil 49: FTSE 100 veri seti için DKA algoritması ile çizilen etkin sınır	67
Şekil 50: S&P 100 veri seti için DKA algoritması ile çizilen etkin sınır	67
Şekil 51: Nikkei veri seti için DKA algoritması ile çizilen etkin sınır	68
Şekil 52: BİST 30 veri seti için DKA algoritması ile çizilen etkin sınır	68
Şekil 53: BİST 100 veri seti için DKA algoritması ile çizilen etkin sınır	69
Şekil 54: Beş günlük geçmiş veriyle BİST 30 endeksi için anapara değişim grafiği	77
Şekil 55: On beş günlük geçmiş veriyle BİST 30 endeksi için anapara değişim grafiği	78
Şekil 56: Otuz günlük geçmiş veriyle BİST 30 endeksi için anapara değişim grafiği	79
Şekil 57: Kırk beş günlük geçmiş veriyle BİST 30 endeksi için anapara değişim grafiği	80
Şekil 58: Altmış günlük geçmiş veriyle BİST 30 endeksi için anapara değişim grafiği	81
Şekil 59: Beş günlük geçmiş veriyle BİST 100 endeksi için anapara değişim grafiği	82
Şekil 60: On beş günlük geçmiş veriyle BİST 100 endeksi için anapara değişim grafiği	83
Şekil 61: Otuz günlük geçmiş veriyle BİST 100 endeksi için anapara değişim grafiği	84
Şekil 62: Kırk beş günlük geçmiş veriyle BİST 100 endeksi için anapara değişim grafiği	85
Şekil 63: Altmış günlük geçmiş veriyle BİST 100 endeksi için anapara değişim grafiği	86
Şekil 64: BIST 30 endeksi için ana etki grafikleri: ortalama değerler	87
Şekil 65: BIST 100 endeksi için ana etki grafikleri: ortalama değerler	87
Şekil 66: BIST 30 endeksi için ana etki grafikleri: son, minimum, maksimum ve standart sapma değerleri	88
Şekil 67: BIST 100 endeksi için ana etki grafikleri: son, minimum, maksimum ve standart sapma değerleri	89

Tablolar Dizini

Tablo 1: OVPO problemi için uygulanan kesin çözüm algoritmaları	4
Tablo 2: OVPO problemine uygulanan evrimsel algoritmalar	6
Tablo 3: PO problemine uygulanan evrimsel algoritmalar.....	8
Tablo 4: PO problemine uygulanan sürü temelli algoritmalar.....	9
Tablo 5: PO problemi için uygulanan sürü temelli algoritmalar	11
Tablo 6: PO problemine uygulanan tek çözüm temelli algoritmalar	12
Tablo 7: PO probleminin çözümü için tek çözüm temelli algoritma uygulayan yayınlar	12
Tablo 8: PO problemine uygulanan makine öğrenimi algoritmaları.....	13
Tablo 9: PO problemi için makine öğrenimi algoritması uygulayan yayınlar	13
Tablo 10: Eleman sayısı kısıtlı portföy optimizasyonu problemi için geliştirilen GAMS kaynak kodları.....	22
Tablo 11: Farklı çözümler için Hang Seng veri seti ile GAMS çözümleri.....	23
Tablo 12: Hang Seng veri setinde farklı K değerleri için GAMS ile bulunan sonuçların performans analizi.....	24
Tablo 13: Deneysel tasarımda kullanılan seviyeler	50
Tablo 14: YAK-I ve YAK-II algoritmalarının performans karşılaştırması	56
Tablo 15: mr , $qBest$ ve ξ parametrelerinin başlangıç ve bitiş değerleri.....	57
Tablo 16: KKO algoritması için performans sonuçları	63
Tablo 17: Yerel Arama Prosedürlerinde kullanılan parametreler ve yapılan değişiklikler	64
Tablo 18: DKA algoritması için performans sonuçları	70
Tablo 19: Proje kapsamında geliştirilen algoritmaların hata performansı karşılaştırılması	72
Tablo 20: YHO, YHMED, YHMİN, YHMAKS hataları için performans karşılaştırması	73
Tablo 21: GHV-I, OGH-I hata performansı karşılaştırması	74
Tablo 22: OÖU, GHV-II, OGH-II hata performansı karşılaştırması	74
Tablo 23: Başarı ölçütleri gerçekleşme tablosu.....	90

Özet

Portföy seçimi problemi, farklı kısıtlar altında finansal riski minimize ederek mümkün olan en iyi getiriyi elde etmek için portföyde bulundurulacak olan varlıkların oranlarını belirleme işlemidir. Matematiksel ifadeyle, ikinci dereceden amaç fonksiyonu ile lineer kısıtlar altında gerçek değerli değişkenlerin optimizasyonu problemidir. Sermayenin varlıklara dağıtılması üzerine portföydeki varlık sayısı sınırlanması ve her varlığa yatırılan sermayenin oranlarına alt ve üst limitler koyulması gibi birçok kısıt uygulanabilir. Bu kısıtlar ele alındığında, literatürde probleme eleman sayısı kısıtlı portföy optimizasyonu adı verilmiş ve bu problemin NP-Zor olduğu ispatlanmıştır. Bu proje önerisinde, portföy optimizasyonu problemine uygulanmamış olan tekniklerden, karınca kolonisi optimizasyonu, yapay arı kolonisi optimizasyonu ve değişken komşuluk arama teknikleri bu projede ilk defa portföy optimizasyonu problemini çözmek için uyarlanmıştır. Daha etkin algoritmaların geliştirilmesine ihtiyaç duyan bu problem için özel olarak geliştirilen alternatif çözüm yaklaşımları finans alanına öncelikle metodolojik olarak katkı sağlanmıştır. Ayrıca, literatürdeki önerilen mevcut yöntemler, şu ana kadar sadece gelişmiş ülkelerdeki varlık endekslerine uygulanmıştır. Ancak, bu projede hem mevcut yöntemler hem de geliştirilen yöntemler Türkiye'deki BIST 30 ve BIST 100 endekslerine de uygulanmış ve uyarlanan metotların karşılaştırmalı analizleri sunulmuştur. Bu sayede, Türkiye piyasalarındaki endekslerine ait veri setleri de uluslararası literatüre kazandırılmıştır. Bu proje sayesinde, teorik altyapısı başarıyla tamamlanarak geliştirilen metodolojiler, yakın gelecekte finans alanındaki uygulama çalışmalarında bir yer bulabilecektir. Ayrıca, bu projede geliştirilen algoritmaların, proje ekinde de detayları verilen, proje kapsamında düzenlenen portföy optimizasyonu çalıştayında, ilerideki aşamalarda kazançlı ticari bir yazılıma dönüştürülerek yatırımcılara bir karar destek sistemi sağlayabileceği ispatlanmıştır. Nitekim, böyle bir yazılım, değişken piyasa şartlarında yatırımcıya güvenilir bir yön gösterebilir. Böylece, finans alanında faaliyet gösterebilecek olan yeni bir şirket için zemin hazırlanmıştır. Şu andan itibaren, projenin yürütüldüğü üniversitenin finans sektöründe aktif rol alması ve daha fazla işbirliği sağlaması mümkün olabilmektedir. Son olarak, elde edilen proje çıktıları sayesinde, proje yürütücüsü kariyerini bir seviye daha yükselterek yetiştirdiği mezun öğrencilerin gururuyla doçentlik unvanına bir adım daha yaklaşmıştır.

Anahtar Kelimeler:

Portföy optimizasyonu, karınca kolonisi, değişken komşuluk arama, yapay arı kolonisi

Abstract

Portfolio selection problem is the determining the ratios of assets that are kept in a portfolio in order to obtain the best possible return by minimizing the financial risk under different constraints. Mathematically, it can be formulated as an optimization problem over real valued variables with a quadratic objective function under linear constraints. Several constraints such as a limit on number of assets as well as lower and upper limits for the ratio of assets in the portfolio can be applied on the distribution of capital to assets. When these constraints are taken into account, the problem is called as cardinality constraint portfolio optimization problem in the literature and is proved to be NP-Complete. Techniques that has not yet been applied to portfolio optimization problems in the literature, such as ant colony optimization, artificial bee colony and variable neighborhood search are adapted solve portfolio optimization problem for the first time in this project. Alternative solution approaches specifically developed for this problem that is in need of efficient algorithms primarily contribute to the field of finance in a methodological way. Moreover, existing methods proposed in the literature has only been applied to asset indexes of developed countries so far. However, both existing methods and newly proposed algorithms are also applied to solve BIST (Istanbul Stock Market) indexes and comparative analysis of all adapted methodologies are presented. By doing so, new data sets of Turkey indexes are introduced to the international literature. Thus, after theoretical groundwork has been successfully completed, developed methodologies by means of this project, will find a place in practical studies in the field of finance in the near future. Moreover, it has been proved in portfolio optimization workshop organized in the scope in this project that the efficient algorithms developed in this project can provide a decision support system by being transformed into a lucrative commercial software in future stages. As a matter of fact, such a software can provide a reliable guidance than its counterparts for the investors in volatile market conditions. Thus, the stage has been set for a new company that may operate in financial field. Hence, making the university play an active role in the finance sector and providing an increased cooperation is now possible. Finally, by means of the outcomes of this project, the project manager has achieved a step up in his career and is now one step closer to associate professor position with an honor of raised graduate students.

Keywords:

Portfolio optimization, any colony, variable neighborhood search, artificial bee colony

1. GİRİŞ

Günümüz rekabet koşulları altında, hem bireysel hem de kurumsal yatırımcı açısından ekonomik faaliyetlerdeki karar verme süreçleri geçmişte olduğundan daha fazla önem kazanmıştır. Bu kararlardan bir tanesi portföy seçimi ve yönetimidir. Optimal yatırım portföylerinin seçimi niceliksel finans alanında önemli bir problemdir ve bilimsel alanda çok ilgi görmüştür. Amerikan ekonomist Harry Markowitz'in 1952 yılında önermiş olduğu modern portföy teorisi, menkul kıymetler arasındaki ilişkiyi matematiksel olarak ifade ederek kıyaslanamayan ve çakışan iki farklı kriterin eş zamanlı olarak optimizasyonu üzerinedir. Bu kriterler; portföyün getirisi ve finansal kayıplarla ilgili risktir. Getiri, belirli bir dönem içinde yapılan bir yatırıma karşılık elde edilen geliri ifade etmektedir. Risk ise, genel anlamda gelecekte beklenmeyen sonuçlarla karşılaşma olasılığı olarak tanımlanmakta, beklenen durumlardan sapmayı ifade etmektedir. Bu nedenle, en iyi portföyü elde etmek için finansal karar verici, önceliklerine, tercihlerine, yargılarına veya sezgilerine göre bu kriterler arasında ödün vermek zorundadır. Portföy yöneticileri, yatırımcıları için risk ile getiri arasındaki en iyi dengeyi veren daha güvenilir stratejiler geliştirmeye çalışmaktadırlar.

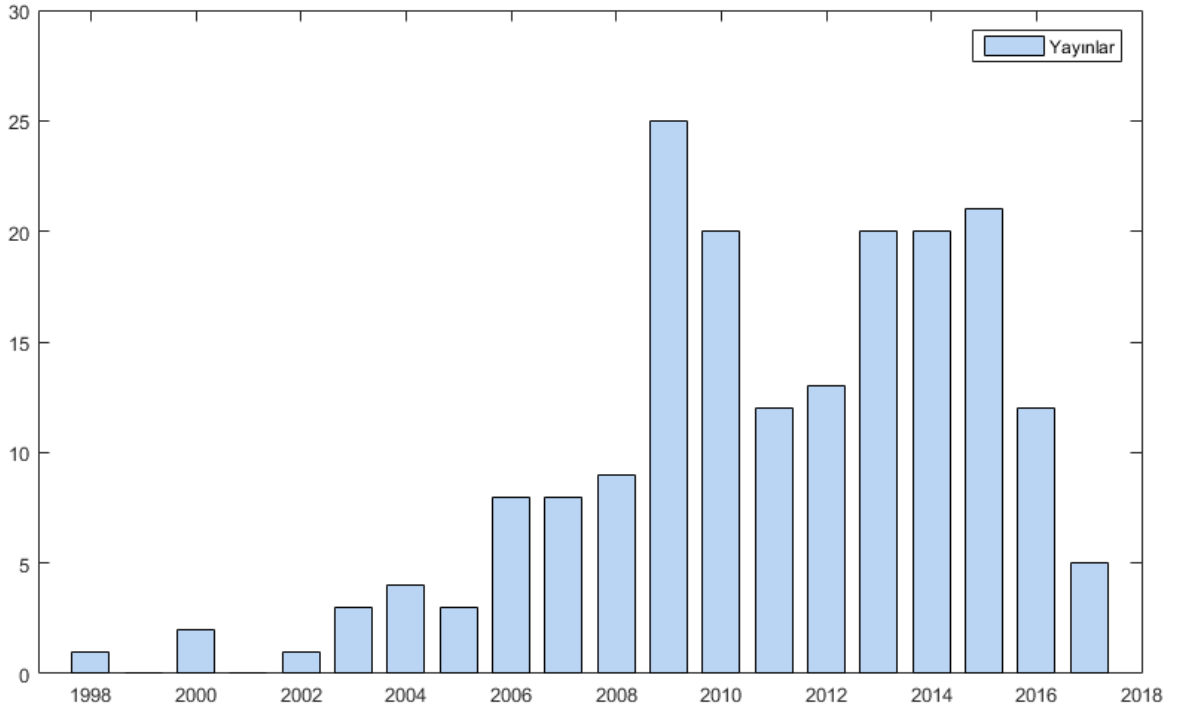
1952 yılında Harry Markowitz portföy teorisi için ortalama- varyans (O-V) modeli ile (Markowitz 1952) devrimsel bir yaklaşım getirmiş ve modern portföy çağını başlatmıştır. Varyansın bir risk ölçütü olarak kullanılması bu devrimin altında yatan en önemli fikir olup niceliksel finansa giden yoldaki ilk kıvılcım yakılmıştır. Harry Markowitz 1991 yılında finans teorilerindeki öncü çalışmaları sayesinde Nobel ekonomi ödülüne layık görülmüş ve portföy optimizasyonu (PO) problemi akademik dünyanın da ilgisini çekmeye başlamıştır.

PO problemi ikinci dereceden bir amaç fonksiyonuna ve lineer kısıtlara sahiptir. NP-zor olduğu kanıtlanan bu problem için literatürde kısıtlı sayıda kesin çözüm üreten algoritma önerilmiştir. Araştırmacılar bu problemin çözümü için büyük çoğunlukla sezgisel yöntemlere başvurmuşlardır. Genetik algoritma (GA) ve parçacık sürü optimizasyonu (PSO) PO problemi için en çok tercih edilen tekniklerdir. Bu proje kapsamında, literatürde çok kullanılan bu iki tekniğin kodlanmasının yanı sıra üç teknik daha kullanılarak PO probleminin çözümü için etkin algoritmalar tasarlanmıştır. Bu teknikler, yapay arı kolonisi (YAK), karınca kolonisi optimizasyonu (KKO) ve değişken komşuluk arama (DKA) teknikleridir.

Bölüm 2'de literatürde PO problemi için yapılan çalışmaların özeti verilmiş, Bölüm 3'de ise geliştirilen algoritmaların testlerinde kullanılan veri setleri (Bölüm 3.1), O-V modeli (Bölüm 3.2) ve GA ve PSO algoritmaları (Bölüm 3.3) verilmiştir. PO probleminin çözümü içine geliştirilen üç algoritma: YAK, KKO ve DKA sırası ile Bölüm 3.4, Bölüm 3.5 ve Bölüm 3.6'da anlatılmıştır. Bölüm 4'te YAK, KKO ve DKA algoritmalarının bulguları ve karşılaştırmalı sonuçlar verilmiştir. Bölüm 5'te ise geliştirilen algoritmalar tartışılmıştır.

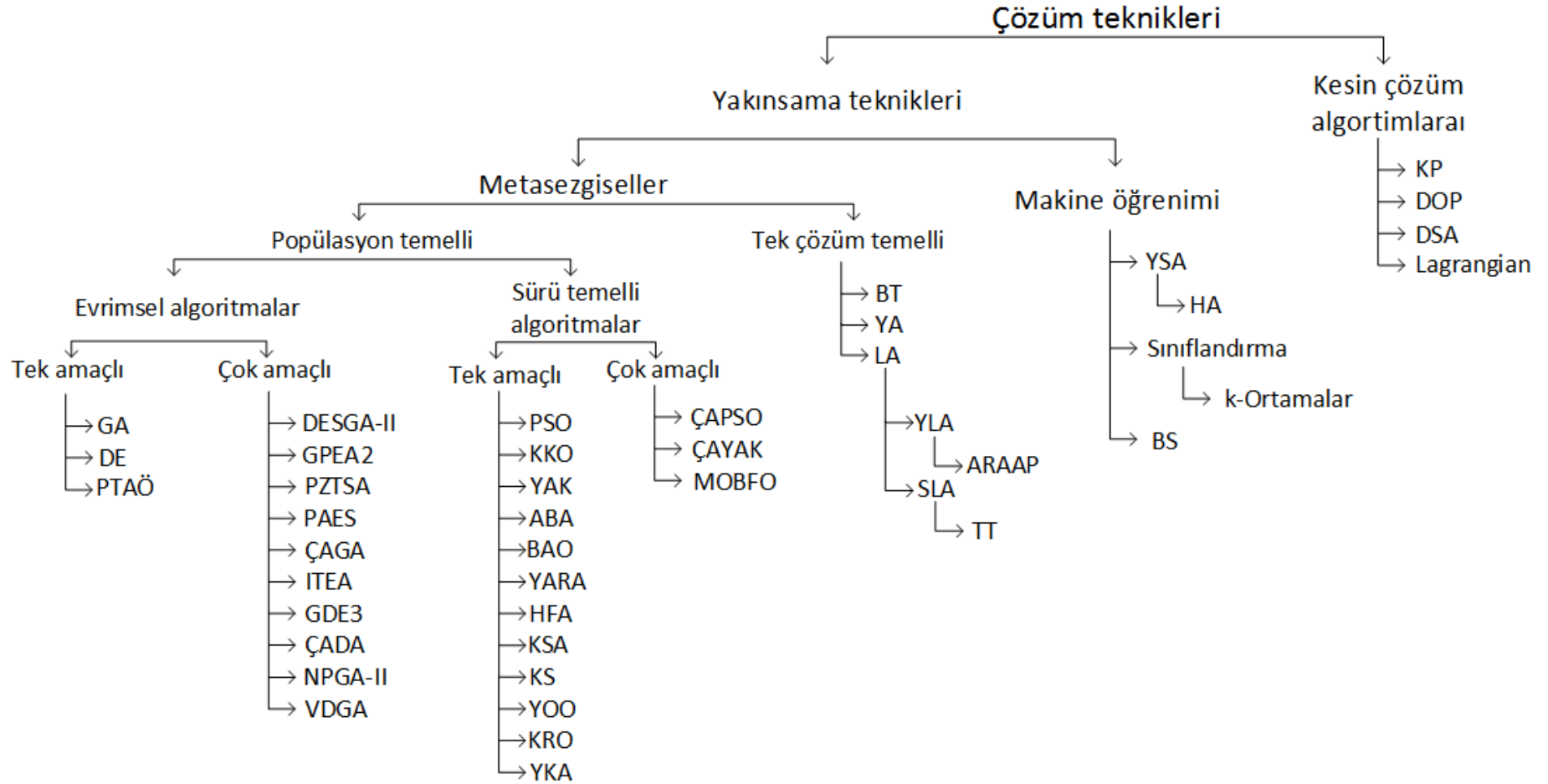
2. LİTERATÜR ÖZETİ

Markowitz (1952), modern portföy teorisinin kapısını açtıktan sonra, insanlar yatırımlarını geleneksel yöntemler yerine niceliksel yöntemleri tercih ederek yapmışlardır. PO problemi akademik platformda da giderek artan bir ilgi görmektedir (Şekil 1). Araştırmacıların Markowitz (1952) tarafından önerilen O-V modelini etkin bir şekilde çözmek için algoritma geliştirme çabaları, literatürde dev bir arşiv oluşturmuştur. O-V modeli ikinci dereceden bir yapıdadır ve dolayısıyla kesin çözüm veren algoritmalar O-V modelli PO problemini çözebilir niteliktedir. Ancak, ek kısıtlar eklendiğinde ya da problem boyutu arttığında, kesin çözüm algoritmaları PO problemini çözmekte yetersiz kalmaktadırlar. Bu nedenle, pek çok araştırmacı NP-zor olduğu kanıtlanan (Moral-Escudero vd. 2006) PO problemini çözmek için yakınsama tekniklerini kullanmışlardır. PO probleminin çözümü için uygulanan algoritmalar Şekil 2'de sınıflandırılmıştır.



Şekil 1: Yıllara göre ortalama-varyans modelli PO problemini çözmek için yapılan yayınların dağılımları

PO probleminin çözümü için geliştirilen teknikler: Kesin çözüm algoritmaları, Metasezgiseller, Makine öğrenimi algoritmaları sırası ile Bölüm 2.1, Bölüm 2.2 ve Bölüm 2.3'de özetlenmiş ve bu teknikleri kullanan yayınlar tablolarda gösterilmiştir.



Şekil 2: Ortalama-Varyans temelli PO problemine uygulanan algoritmaların sınıflandırılması

YAK: yapay arı kolonisi, KKO: karınca kolonisi optimizasyonu, YKA: yapay kooperatif arama, YARA: yarasa algoritması, BAO: bakteriyel avlanma optimizasyonu, DSA: dal sınır algoritması, KRO: kimyasal reaksiyon optimizasyonu, KSA: kedi sürüsü algoritması, DE: diferansiyel evrim, ABA: ateş böceği algoritması, HFA: havai fişek algoritması, GA: genetik algoritma, GDE3: genelleştirilmiş DE3, ARAAP: açgözlü rasgele adaptif arama prosedürü, TT: tepe tırmanışı, HA: Hopfield ağı, ITEA: indikatör temelli EA, YOO: yayılan ot optimizasyonu, YLA: yinelemeli lokal arama, KS: Krill sürüsü, LA: lokal arama, ÇAYAK: çok amaçlı YAK, ÇAGA: çok amaçlı GA, ÇAPSO: çok amaçlı PSO, ÇABAA: çok amaçlı BAA, ÇADA: çok amaçlı dağınık arama, DOP: doğrusal olmayan programlama, YSA: yapay sinir ağları, NPGA-II: Niched Pareto GA 2, DEGA-II: domine edilmemiş GA2, PAES: pareto arşivlenmiş evrim stratejisi, PTAÖ: popülasyon temelli artan öğrenme, PZTSA: pareto zarf tabanlı seçim algoritması, PSO: parçacık sürü optimizasyonu, KP: kuadratik programlama, BT: benzetilmiş tavlama, SLA: stokastik LA, GPEA2: güç paretosu EA 2, YA: yasaklı arama, VDGA: vektör değerlendirilmiş GA

2.1. Kesin çözüm algoritmaları

PO problemi kesin çözüm algoritmaları ile çözülebilir. En çok uygulanan kesin çözüm algoritması ikinci dereceden (kuadratik) programlamadır (KP). [Aouni vd. \(2005\)](#) ve [Mansour vd. \(2007\)](#) PO problemi için hedef programlama modelleri geliştirmişler ve geliştirilen modellerin çözümünü için KP kullanmışlardır. [Hu ve Zhangy \(2010\)](#) farklı PO modellerinin performanslarını kıyaslamışlardır. Bu modeller: O-V, ortalama mutlak sapma (MAD), koşullu riskteki değer (CVaR) ve minimaks model (MM), geliştirilen modeller KP ile çözülmüştür. [Peng vd. \(2011\)](#) kuadratik formda işlem maliyetli bir O-V modeli önermiştir ve önerilen modeli KP kullanarak çözmüştür. [Brito ve Vicente \(2014\)](#) ilk amacı portföydeki hisse sayısını minimize eden ikinci amacı ise getiri/risk maksimizasyonu olan iki amaçlı bir model geliştirmiştir ve geliştirilen modeli KP ile çözerek sonuçları raporlamışlardır. O-V modeli kuadratik yapıda olmasına rağmen, lineer olmayan kısıtlar eklendiğinde kuadratik yapısı bozulur. Bu nedenden böyle durumlarda KP problemi çözemeyen ancak doğrusal olmayan programlama (DOP) çözebilir. [Kocadağlı ve Keskin \(2015\)](#) bulanık hedef programlama temelli bir model önermiş ve önerilen bu modelin DOP ile çözüm sonuçları klasik modele üstünlük sağlamıştır. PO problemini DOP ya da KP kullanarak çözen araştırmacıların çoğunluğu, çözücü olarak CPLEX ya da MATLAB kuadratik çözücüsünü kullanmışlardır. KP ve DOP çözücülerinin yanında, Lagrangian metotları ([Li vd. 2006](#); [Shaw vd. 2008](#)) ve dal sınır algoritması (DSA) ([Bonami ve Lejeune 2009](#)) O-V temelli PO (OVPO) probleminin çözümü için kullanılmıştır (Tablo 1).

Tablo 1: OVPO problemi için uygulanan kesin çözüm algoritmaları

Teknik	Yayınlar
KP	Aouni vd. (2005) ; Brito ve Vicente (2014) ; Cesarone vd. (2013) ; Cesarone vd. (2015) ; Chen vd. (2015) ; Cui vd. (2014) ; Cui vd. (2013) ; Hu ve Zhangy (2010) ; Jiang vd. (2014) ; Liagkouras ve Metaxiotis (2014) ; Mansour vd. (2007) ; Mayambala vd. (2015) ; Peng vd. (2011) ; Qi (2015) ; Qin (2015)
DOP	Kiriş ve Ustun (2012) ; Kocadağlı ve Keskin (2015)
DSA	Bonami ve Lejeune (2009)
Lagrangian	Li vd. (2006) ; Shaw vd. (2008)

2.2. Yakınsama teknikleri

OVPO problemi NP-zor sınıfta olduğu için yakınsama teknikleri literatürde yaygın olarak kullanılmıştır. Yakınsama teknikleri, metasezgiseller ve makine öğrenme algoritmaları olarak iki grupta incelenmiştir.

2.2.1. Metasezgiseller

Metasezgiseller kombinasyonel problemlerde optimal çözüme yakın çözümler elde etmek için çözüm uzayında arama stratejileri uygulayan iteratif süreçlerdir. Metasezgiseller alt sınıf sezgisellere tatmin edici sonuçlar bulmak için rehberlik eden üst sınıf prosedürlerdir. Her iterasyonda bir popülasyon ya da tek bir çözüm kullanılabilir. Alt sınıf sezgiseller bir yerel arama algoritması ya da başlangıç çözüm metodu olabilir.

Metasezgiseller çözüm uzayında arama yaparken iki mekanizma kullanırlar: keşfetme ve sömürme mekanizmaları. Keşif, çözüm uzayında çok fazla bölgenin taranmasını sağlar, böylece lokal optimuma takılmayı önler. Sömürme ise iyi çözümlerin oluşuna inanılan küçük bir bölgeye odaklanarak arama yapmayı sağlar. Algoritma ilerlerken bu iki çakışan mekanizma arasında bir denge kurmak çok önemlidir, çünkü birisine daha fazla ağırlık vermek diğer mekanizmanın yetersiz kullanımına sebep olur.

Metasezgisel algoritmalar, popülasyon temelli algoritmalar ve tek çözüm temelli algoritmalar olarak iki grupta sınıflandırılmıştır.

2.2.1.1. Popülasyon temelli algoritmalar

Popülasyon temelli algoritmalar çözüm uzayında arama yaparken popülasyonun gücünü kullanırlar. Popülasyon geniş bir alanı taramak için dağılılabılır ya da hassas tarama için yoğunlaşarak özel bir bölgede arama yapabilir. Popülasyon temelli algoritmalar bir başlangıç çözümü yaratarak (genellikle rasgele) her iterasyonda popülasyonu iyileştirme stratejisini izlerler. Popülasyon temelli algoritmalar esinlendikleri kaynağa göre iki gruba ayrılırlar: evrimsel algoritmalar ve sürü temelli algoritmalar.

2.2.1.1.1. Evrimsel algoritmalar

Evrimsel algoritmaların temelinde, kromozomlarla temsil edilen popülasyondaki bireylerden çaprazlama, mutasyon gibi çeşitlilik sağlayan operatörler kullanarak yeni bireyler yaratmak yatmaktadır. En çok kullanılan evrimsel algoritma [Holland \(1975\)](#) tarafından önerilen genetik algoritmadır (GA). OVPO problemi için uygulanan evrimsel algoritmalar Tablo 2'de gösterilmiştir.

Tablo 2: OVPO problemine uygulanan evrimsel algoritmalar

Evrimsel algoritma	Öneren	Amaç tipi
Genetik algoritma (GA)	Holland (1975)	Tek amaçlı
Diferansiyel evrim (DE)	Storn ve Price (1997)	Tek amaçlı
Popülasyon temelli artımlı öğrenme (PTAÖ)	Baluja (1994)	Tek amaçlı
Domine edilmemiş sıralama genetik algoritması 2 (DESGA-II)	Deb vd. (2002)	Çok amaçlı
Güç pareto evrim algoritması 2 (GPEA2)	Zitzler vd. (2001)	Çok amaçlı
Pareto zarf temelli seçim algoritması (PZTSA)	Corne vd. (2000)	Çok amaçlı
Pareto arşiv evrim stratejisi (PAES)	Knowles vd. (2006)	Çok amaçlı
Gösterge temelli evrimsel algoritma (GTEA)	Zitzler ve Künzli (2004)	Çok amaçlı
Çok amaçlı genetik algoritma (ÇAGA)	Fonseca ve Fleming (1993)	Çok amaçlı
Niched pareto genetik algoritma 2 (NPGA-II)	Erickson vd. (2001)	Çok amaçlı
Genelleştirilmiş diferansiyel evrim 3 (GDE3)	Kukkonen ve Lampinen (2005)	Çok amaçlı
Çok amaçlı dağınık arama (ÇKDA)	Glover vd. (2000)	Çok amaçlı
Vektör değerlendirilmiş genetik algoritma (VDGA)	Schaffer (1985)	Çok amaçlı

[Chang vd. \(2000\)](#) OVPO problemi için bir GA geliştirmiş ve yasaklı arama (YA) ve benzetilmiş tavlama (BT) teknikleri ile karşılaştırılmıştır. Elde ettikleri bulgulara göre, geliştirdikleri her algoritmanın farklı veri setleri üzerinde birbirine üstünlükleri gözlemlenmiş ve en etkin portföy seçimi için tüm algoritmaların yarattığı çözüm havuzundan yararlanılabileceği sonucuna varmışlardır. Ancak daha etkin ve hızlı çözüm veren algoritmaların geliştirilmesi gereksinimi açık ortadadır. [Moral-Escudero vd. \(2006\)](#) GA ve kuadratik programlama tekniklerini birleştiren melez bir strateji geliştirmişlerdir. [Aranha ve Iba \(2009\)](#) bir yerel arama prosedürünü de içine alan ağaç yapılı bir GA önermişlerdir. [Loukeris vd. \(2009\)](#) GA ve PSO tekniklerini karşılaştırmış ve performanslarının artırılması için yeni çalışmalara ihtiyaç olduğunu altını çizmişlerdir. [Soleimani vd. \(2009\)](#) GA ile LINGO modelindeki sonuçları karşılaştırmışlar ve kabul edilebilir performansta sonuçlar elde etmişler ve yeni tekniklerin geliştirilmesi gerektiğini belirtmişlerdir. [Branke vd. \(2009\)](#) portföy seçimi için optimize edilmiş bir aktif küme algoritmasını çok amaçlı bir GA ile birleştirerek mevcut çok amaçlı GA tekniklerinden daha iyi performans veren zarf tabanlı bir GA önermişler ve kullandıkları mutasyon operatörünün geliştirilmesi gerektiğini belirtmişlerdir. [Anagnostopoulos ve Mamanis \(2010\)](#) çok amaçlı GA üzerine karşılaştırmalı bir analiz yapmışlardır. [Xu vd. \(2010\)](#) bir popülasyon temelli artımlı öğrenim (PTAÖ) algoritması önermiş ve önerilen algoritma ile [Chang vd. \(2000\)](#) tarafından önerilen GA ve [Cura \(2009\)](#) tarafından önerilen PSO teknikleriyle kıyaslamışlardır. Test

sonuçlarına göre önerilen PTAÖ algoritması ile diğer algoritmaların sonuçlarına yakın değerler elde ettiklerini belirtmişlerdir. [Anagnostopoulos ve Mamanis \(2011a\)](#) çok amaçlı genetik algoritmaların tek amaçlı genetik algoritmalara göre problemin çözümünde daha iyi performans verdiklerini göstermişlerdir. [Anagnostopoulos ve Mamanis \(2011b\)](#) yatırımcı isteklerinin problemi konveks olmayan bir yapıya dönüştürdüğünü vurgulayarak çok amaçlı genetik algoritmaları tam çözüm veren matematiksel programlama teknikleri ile karşılaştırarak GA performansını göstermişlerdir. [Woodside-Oriakhi vd. \(2011\)](#) geliştirdikleri GA metodunu literatürdeki mevcut yöntemlerle karşılaştırmışlar, çözüm kalitesi ve hız kriterlerini göz önüne alındığında daha iyi performans elde etmişlerdir. [Chen vd. \(2011\)](#) GA için yeni bir güdümlü mutasyon operatörü geliştirmişlerdir. [Sadjadi vd. \(2012\)](#) portföydeki varlıkların sayısındaki sınırlama ve belirsizliklerin problemin karmaşıklığı üzerindeki etkisini incelemişler ve GA ile optimale yakın çözümler elde etmişlerdir. [Liagkouras ve Metaxiotis \(2014\)](#) GA için yeni bir mutasyon operatörü geliştirerek performansını farklı veri setleri üzerinde test etmişlerdir.

[Fieldsend vd. \(2004\)](#) ilk defa çok amaçlı evrimsel algoritmayı (ÇAEA) OVPO problemi için uygulamıştır. [Ong vd. \(2005\)](#) üç amaçlı bir PO modeli önermiştir: getiri maksimizasyonu, belirsiz risk minimizasyonu ve ilişki risk minimizasyonu. Önerilen modeli çözmek için bir domine edilmemiş sıralama genetik algoritması 2 (DESGA-II) geliştirmişlerdir ve DESGA-II algoritmasının esnek ve kesin çözümler sağladığını belirtmişlerdir. [Skolpadungket vd. \(2007\)](#) PO problemi için dört farklı çok amaçlı evrimsel algoritma uygulamıştır. Bu algoritmalar: vektör hesaplamalı genetik algoritma (VHGA), çok amaçlı genetik algoritma (ÇAGA), güç pareto evrimsel algoritması 2 (GPEA2) ve DESGA-II. Deneysel sonuçlara göre, ÇAGA ve GPEA2 daha karmaşık algoritmalar olmalarına rağmen kısmen daha iyi çözümler üretmiştir. [Anagnostopoulos ve Mamanis \(2010\)](#) üç amaçlı bir PO modeli önermiştir: getiri maksimizasyonu, risk minimizasyonu ve portföyde tutulan hisse sayısının minimizasyonu. Geliştirilen modeli DESGA-II, pareto zarf temelli seçim algoritması (PZTSA) ve GPEA2 kullanarak çözmüşlerdir ve test sonuçları göstermiştir ki GPEA-II daha iyi sonuçlar el etmiştir. [Liagkouras ve Metaxiotis \(2014\)](#) rehber temelli bir mutasyon operatörü geliştirmiş ve DESGA-II ve GPEA2 algoritmalarının kullanarak geliştirile operatör ve klasik mutasyon operatörünü kıyaslamışlardır. Geliştirilen operatör hem DESGA-II hem de GPEA2 algoritmasında daha iyi sonuçlar vermiştir. [Macedo vd. \(2017\)](#) iki çok amaçlı evrimsel algoritmayı kıyaslamıştır, DESGA-II, GPEA2 algoritmasından daha iyi sonuçlar elde etmiştir.

PO problemine uygulanan evrimsel algoritmalar Tablo 3'de gösterilmiştir.

Tablo 3: PO problemine uygulanan evrimsel algoritmalar

Evrimsel Algoritmalar	Kaynaklar
GA	Ackora-Prah vd. (2014); Aranha ve Iba (2009); Barak vd. (2013); Bhattacharyya vd. (2011); Busetti (2006); Cadenas vd. (2009); Chang vd. (2009); Chang vd. (2000); Chen vd. (2008); ChiangLin (2006); Corazza vd. (2012); Coutino-Gomez vd. (2003); Ehrgott vd. (2004); Guo vd. (2016); Hadi vd. (2016); Hao ve Liu (2009); Hoklie ve Zuhail (2010); Huang (2012); Huang ve Shen (2010); Kuo ve Hong (2013); Lai vd. (2006); Lean vd. (2008); Li (2009); Lin ve Liu (2008); Lin vd. (2005); Liu ve Zhang (2015); Loukeris vd. (2009); Lu ve Wang (2013); Ma vd. (2014); Moral-Escudero vd. (2006); Pai ve Michel (2009); Rong vd. (2009); Ruiz-Torribiano ve Suarez (2010); Ruiz-Torribiano ve Suárez (2007); Ruiz-Torribiano ve Suárez (2015); Sabar ve Song (2014); Sadjadi vd. (2012); Shaikh ve Abbas (2009); Shoaf ve Foster (1998); Soleimani vd. (2009); Talebi vd. (2010); Thomaidis (2010); Xia vd. (2000); Yu vd. (2009)
DE	Loukeris vd. (2009); Lwin ve Qu (2013); Zaheer ve Pant (2016)
PTAÖ	Jin vd. (2015); Lwin ve Qu (2013); Xu vd. (2010)
DESGA-II	Anagnostopoulos ve Mamanis (2010, 2011a, 2011b); Arkeman vd. (2013); Babaei vd. (2015); Branke vd. (2009); Chiam vd. (2007); Chiam vd. (2008); Duran vd. (2009); Fieldsend vd. (2004); García vd. (2012); Liagkouras ve Metaxiotis (2014); Lwin vd. (2014); Macedo vd. (2017); Mashayekhi ve Omrani (2016); Mishra vd. (2016); Ong vd. (2005); Sen vd. (2015); Skolpadungket vd. (2007); Streichen ve Tanaka-Yamawaki (2006); Streichert vd. (2004a); Streichert vd. (2004b)
GPEA2	Anagnostopoulos ve Mamanis (2010, 2011a, 2011b); Babaei vd. (2015); Duran vd. (2009); García vd. (2012); Liagkouras ve Metaxiotis (2014); Lwin vd. (2014); Macedo vd. (2017); Mishra vd. (2016); Skolpadungket vd. (2007)
PZTSA	Anagnostopoulos ve Mamanis (2010, 2011a, 2011b); Lwin vd. (2014)
PAES	Lwin vd. (2014)
ÇAGA	Bermúdez vd. (2012); Bevilacqua vd. (2012); Skolpadungket vd. (2007); Yi ve Yang (2013)
ITEA	Duran vd. (2009)
GDE3	Adebisi Ayodele ve Ayo Charles (2015); García vd. (2012)
ÇADA	Lwin vd. (2013)
NPGA-II	Anagnostopoulos ve Mamanis (2011a)
VDGA	Skolpadungket vd. (2007)

2.2.1.1.2. Sürü temelli algoritmalar

Sürü temelli algoritmalar doğada sürü halinde yaşayan hayvanların doğal yaşamlarından esinlenilerek geliştirilmiş yakınsama teknikleridir. Kuş, balık, karınca, arı gibi sürü halinde yaşayan hayvanların doğal yaşamlarındaki davranışlarını (çoğunlukla avlanma) benzeterek, kombinasyonel problemler için hesaplamalı sistemler geliştirilmektedir. En popüler sürü temelli algoritma [Kennedy ve Eberhart \(1995\)](#) tarafından önerilen parçacık sürü optimizasyonu (PSO) algoritmasıdır. PO problem için uygulanan sürü temelli algoritmalar Tablo 4’de gösterilmiştir.

Tablo 4: PO problemine uygulanan sürü temelli algoritmalar

Sürü temelli algoritma	Öneren	Amaç tipi
Parçacık sürü optimizasyonu (PSO)	Kennedy ve Eberhart (1995)	Tek amaçlı
Karınca kolonisi optimizasyonu (KKO)	Dorigo vd. (1996)	Tek amaçlı
Yapay arı kolonisi (YAK)	Karaboga (2005)	Tek amaçlı
Ateş böceği algoritması (ABA)	Yang (2010b)	Tek amaçlı
Bakteriyel avlanma optimizasyonu (BAO)	Passino (2002)	Tek amaçlı
Yarasa algoritması (YARA)	Yang (2010a)	Tek amaçlı
Havai fişek algoritması (HFA)	Tan ve Zhu (2010)	Tek amaçlı
Kedi sürüsü optimizasyonu (KSO)	Chu ve Tsai (2007)	Tek amaçlı
Krill sürüsü (KS)	Gandomi ve Alavi (2012)	Tek amaçlı
Yayılan ot optimizasyonu (YOO)	Mehrabian ve Lucas (2006)	Tek amaçlı
Kimyasal reaksiyon optimizasyonu (KRO)	Lam ve Li (2010)	Tek amaçlı
Yapay kooperatif arama (YKA)	Civicioglu (2013)	Tek amaçlı
Çok amaçlı parçacık sürü optimizasyonu (ÇAPSO)	Coello vd. (2004)	Çok amaçlı
Çok amaçlı yapay arı kolonisi (ÇAYAK)	Kumar ve Mishra (2017)	Çok amaçlı
Çok amaçlı bakteriyel avlanma optimizasyonu (ÇABAO)	Panigrahi vd. (2011)	Çok amaçlı

PSO algoritması OVPO problemine ilk defa [Chen vd. \(2006\)](#) tarafından uygulanmıştır. [Cura \(2009\)](#) PO problemi için bir PSO algoritması geliştirmiş ve GA, YA ve BT ([Chang vd. 2000](#)) ile kıyaslamıştır. [Cura \(2009\)](#), test sonuçlarına göre dört algoritmanın da birbirine üstünlük sağlayamadığını ancak PSO algoritmasının düşük risk seviyelerinde daha en iyi sonuçları verdiğini belirtmiştir. Karınca kolonisi optimizasyonu (KKO) PO problemine ilk defa [Deng ve Lin \(2010a\)](#) tarafından uygulanmış ve PSO algoritması ile kıyaslanmıştır. Kıyaslama sonuçlarına göre KKO algoritmasının özellikle düşük risk seviyelerinde daha etkin sonuçlar verdiğini belirtmişlerdir. Yapay arı kolonisi (YAK) algoritması PO problemine ilk defa [Hong-mei vd. \(2010\)](#) tarafından uygulanmış ve yazarlar YAK algoritmasının PO problemi için kullanışlı bir yaklaşım olduğunu belirtmişlerdir. [Zhu vd. \(2010\)](#) iki farklı sürü temelli algoritma önermiştir: KKO ve PSO. Yazarlar iki algoritmayı kıyaslamış ve KKO algoritmasının nispeten üstünlük sağladığını belirtmişlerdir. [Golmakani ve Fazel \(2011\)](#) geliştirdikleri PSO algoritmasını, [Soleimani \(2007\)](#) tarafından geliştirilen GA algoritması ile kıyaslamışlar ve PSO algoritmasının daha iyi sonuçlar elde ettiğini belirtmişlerdir. [Xu vd. \(2011\)](#) evrim sürecine süper molekül ekleyerek bir kimyasal reaksiyon optimizasyonu (KRO) önermişlerdir. Simülasyon sonuçları önerilen KRO algoritmasının orijinal KRO algoritmasına göre daha iyi performans verdiğini

göstermiştir. [Deng vd. \(2012\)](#) mutasyon operatörlü bir PSO algoritması geliştirip, GA, YA ve BT ([Chang vd. 2000](#)) algoritmaları ile kıyaslamışlardır. Yazarlar PSO algoritmasının daha iyi sonuçlar verdiğini belirtmiştir. [Chen vd. \(2013\)](#) iki aşamalı YAK algoritması önermiştir. Önerilen algoritmanın ilk aşamasında portföye alınacak hisseler belirlenir ikinci aşamasında ise belirlenen hisselerin ağırlıkları belirlenir. [Kao ve Cheng \(2013\)](#) bakteriyel avlanma optimizasyonu (BAO) algoritması önermiş ve GA, YA, BT ([Chang vd. 2000](#)) ve PSO ([Cura 2009](#)) algoritmaları ile kıyaslamıştır. Çözüm kalitesi ve zaman ölçütleri açısından BAO daha iyi çözümler sağlamıştır. Havai fişek algoritması (HFA) OVPO problemine ilk defa [Tuba vd. \(2013\)](#) tarafından uygulanmış ve yazarlar HFA'nın bu problemi çözmek için potansiyeli olduğunu vurgulamışlardır. [Bacanin vd. \(2014\)](#) YAK algoritması ile GA ve ateş böceği algoritmasını (ABA) kıyaslamış ve YAK algoritmasının diğer algoritmalara göre daha iyi sonuçlar verdiğini belirtmişlerdir. [Tan vd. \(2014\)](#) dinamik OVPO problemi için komşuluk öğrenmeli bir BAO algoritması önermiş ve orijinal BAO ile kıyaslamıştır. Test sonuçları önerilen algoritmanın orijinal BAO algoritmasına üstünlük sağladığını göstermiştir. [Kamili ve Riffi \(2015\)](#) OVPO problemi için bir kedi sürüsü optimizasyonu (KSO) algoritması önermiştir. [Kamili ve Riffi \(2016\)](#) üç farklı sürü temelli algoritmayı kıyaslamıştır: PSO, KSO ve yarasa algoritması (YARA). Kıyaslama sonuçlarına göre üç algortmada yakın sonuçlar elde etmiştir. [Ni vd. \(2017\)](#) rasgele dinamik popülasyon stratejisini temel alan bir PSO algoritması önermiş ve izlenen stratejinin PSO algoritmasının performansını iyileştirdiğini belirtmiştir. [Kalayci vd. \(2017\)](#) eleman sayısı kısıtı için çözümlerin uygulanabilir olmasına zorlayan, alt ve üst sınır kısıt için se uygulanabilir olmayan çözümlere izin veren bir YAK algoritması önermiştir. Önerilen algoritmanın orijinal YAK algoritmasına göre çok iyi çözümler elde etmesinin yanı sıra literatürdeki çalışmalara da (GA, YA, BT ([Chang vd. 2000](#)), PSO ([Deng vd. 2012](#))), PTAÖ-DE ([Lwin ve Qu 2013](#)), ARAAP-KP ([Baykasoğlu vd. 2016](#))) üstünlük sağlamıştır.

[Mishra vd. \(2009\)](#) OVPO problemi için çok amaçlı parçacık sürü optimizasyonu (ÇAPSO) algoritması önermiş ve DESGA-II, GPEA2 algoritmaları ile kıyaslamıştır. [Mishra vd. \(2014b\)](#) çok amaçlı bakteriyel avlanma optimizasyonu (ÇABAO) algoritması önermiş ve DESGA-II, PAES ve PZTSA teknikleri ile kıyaslamıştır. Test sonuçlarına göre, ÇABAO algoritması OVPO problemini çözmekle kalmayıp diğer algoritmalara göre daha iyi sonuçlar vermiştir. ([Mishra vd. 2016](#)) tahmin temelli O-V modeli önermiş ve modeli üç farklı algoritma kullanarak çözmüştür: ÇAPSO, DESGA-II ve GPEA2. ÇAPSO algoritmasının en iyi pareto çözümler elde ettiğini belirtmişlerdir. [Kumar ve Mishra \(2017\)](#) PO problemini çözmek için çok amaçlı yapay arı kolonisi algoritması (ÇAYAK) önermiştir. Test sonuçları ÇAYAK algoritmasının PO problemini çözebileceğini kanıtlamıştır.

PO problemi için uygulanan sürü temelli algoritmalar Tablo 5'de gösterilmiştir.

Tablo 5: PO problemi için uygulanan sürü temelli algoritmalar

Sürü temelli algoritma	Kaynaklar
PSO	Abbas ve Haider (2009); Cao ve Tao (2010); Chang ve Chen (2009); Chang ve Hsu (2008); Chen vd. (2012); Chen ve Cai (2008); Chen vd. (2006); Chen ve Zhang (2010); Corazza vd. (2012); Corazza vd. (2013); Cura (2009); Deng ve Lin (2010b); Deng vd. (2012); Farzi vd. (2013); Gao ve Chu (2009); Golmakani ve Fazel (2011); Kamili ve Riffi (2016); Koshino vd. (2007); Li vd. (2010); Li vd. (2011); Liu vd. (2013); Loukeris vd. (2009); Mozafari vd. (2011); Ni vd. (2017); Niu vd. (2010); Niu vd. (2009); Pouya vd. (2016); Reid ve Malan (2015); Sadigh vd. (2012); Sun vd. (2011); Talebi vd. (2010); Tang vd. (2009); Wang vd. (2015); Wang vd. (2009); Wang vd. (2012); Xu ve Chen (2006); Xu vd. (2007); Yaakob ve Watada (2010); Yin vd. (2015a, 2015b); Zhang vd. (2010); Zhu vd. (2010); Zhu vd. (2011)
YAK	Bacanin vd. (2014); Chen vd. (2012); Chen (2014); Chen vd. (2014); Ge (2015); Hong-mei vd. (2010); Kalayci vd. (2017); Seyedhosseini vd. (2016); Suthiwong ve Sodanil (2016); Tuba ve Bacanin (2014a); Wang vd. (2012)
KKO	Deng ve Lin (2010a); Zhu vd. (2010)
ABA	Bacanin ve Tuba (2014); Tuba ve Bacanin (2014a, 2014b); Tuba vd. (2013)
BAO	Kao ve Cheng (2013); Tan vd. (2013); Tan vd. (2014)
YARA	Kamili ve Riffi (2016); Strumberger vd. (2017)
HFA	Bacanin ve Tuba (2015)
KSO	(Kamili ve Riffi 2015); Kamili ve Riffi (2016)
KS	Tuba vd. (2014)
YOO	Hajnoori vd. (2013); Pouya vd. (2016)
KRO	Xu vd. (2011)
YKS	Kumar ve Bhattacharya (2012)
ÇAPSO	Babaei vd. (2015); Liang ve Qu (2013); Mishra vd. (2016); Mishra vd. (2014a); Sen vd. (2015); Zhou ve Li (2014)
ÇAYAK	Kumar ve Mishra (2017)
ÇABAO	Mishra vd. (2014b)

2.2.1.1. Tek çözüm temelli algoritmalar

Tek çözüm üzerinde çalışan algoritmalarla yörunge metotları da denir (Blum ve Roli 2003). Tek çözüm temelli algoritmalar bir popülasyon yerine tek bir çözümü iyileştirme stratejisini izlerler. Popülasyonun gücünü kullanamamalarına rağmen her iterasyonda bir popülasyonu iyileştirmek yerine tek bir çözümü iyileştirmeye çalıştığı için oldukça hızlıdır.

PO problemi için uygulanan tek çözüm temelli algoritmalar Tablo 6'da verilmiştir.

Tablo 6: PO problemine uygulanan tek çözüm temelli algoritmalar

Tek çözüm temelli algoritma	Öneren	Amaç tipi
Benzetilmiş tavlama (BT)	Kirkpatrick vd. (1983)	Tek amaçlı
Yasaklı arama (YA)	Glover (1986)	Tek amaçlı
Açgözlü rasgele adaptif arama prosedürü (ARAAP)	Feo ve Resende (1995)	Tek amaçlı
Tepe tırmanışı (TT)	Forrest ve Mitchell (1993)	Tek amaçlı
Arşivlenmiş çok amaçlı benzetilmiş tavlama (AÇABT)	Bandyopadhyay vd. (2008)	Çok amaçlı

[Chang vd. \(2000\)](#) ve [Ehr Gott vd. \(2004\)](#) PO problemi için YA, BT ve GA tekniklerini kıyaslamışlardır ancak her iki çalışmada da herhangi bir algoritmanın üstün geldiği belirtilmemiştir. [Anagnostopoulos vd. \(2010\)](#) OVPO problemi için açgözlü rasgele adaptif arama prosedürü (ARAAP) tasarlamıştır. Yazarlar ARAAP algoritmasının OVPO problemini çözmek için güçlü bir algoritma olduğunu belirtmiştir. [Woodside-Oriakhi vd. \(2011\)](#) üç farklı metasezgisel önermiştir (GA, YA, BT) ve elde ettikleri sonuçları [Chang vd. \(2000\)](#) tarafından yayınlanan sonuçlarla kıyaslamıştır. [Fastrich ve Winker \(2012\)](#) BT algoritmasını bir sezgisel algoritmanın içerisine değiştirme mekanizması olarak gömmüştür. [Sen vd. \(2015\)](#) arşivlenmiş çok amaçlı benzetilmiş tavlama (AÇABT) tekniğini uygulamış ve DESGA-II, ÇAPSO teknikleri ile kıyaslamıştır. Yazarlar AÇAPT tekniğinin daha iyi sonuçlar elde ettiğini belirtmiştir. [Chen vd. \(2017\)](#) bir lokal arama temelli çok amaçlı optimizasyon tekniği önermiştir.

PO probleminin çözümü için tek çözüm temelli algoritma uygulayan yayınlar Tablo 7'de verilmiştir.

Tablo 7: PO probleminin çözümü için tek çözüm temelli algoritma uygulayan yayınlar

Tek çözüm temelli algoritma	Yayınlar
BT	Chang vd. (2000) ; Coutino-Gomez vd. (2003) ; Ehr Gott vd. (2004) ; Fastrich ve Winker (2012) ; Li vd. (2009) ; Schaerf (2002) ; Thomaidis (2010)
YA	Buseti (2006) ; Chang vd. (2000) ; Ehr Gott vd. (2004) ; Schaerf (2002)
ARAAP	Anagnostopoulos vd. (2010) ; Baykasoğlu vd. (2015)
TT	Coutino-Gomez vd. (2003) ; Gaspero vd. (2010) ; Schaerf (2002)
AÇABT	Sen vd. (2015)

2.3. Makine öğrenimi algoritmaları

Makine öğrenimi, verileri öğrenebilen ve veriler üzerinde öngöründe bulunabilen algoritmaların çalışmasını ve inşasını araştırır. Bu tür algoritmalar, katı statik program talimatlarını izlemek yerine, veri odaklı tahminler veya çıktılar olarak verilen kararlar vermek için, örnek bir eğitim seti olan giriş gözlemlerinden bir model oluşturarak çalışır.

PO problem için uygulanan makine öğrenimi algoritmaları Tablo 8’de verilmiştir.

Tablo 8: PO problemine uygulanan makine öğrenimi algoritmaları

Makine öğrenimi algoritması	Öneren
Hopfield ağı (HA)	Hopfield (1984)
Yapay bağıklık sistemi (YBS)	De Castro ve Timmis (2002)
k-ortalamlar kümeleme	Lloyd (1982)

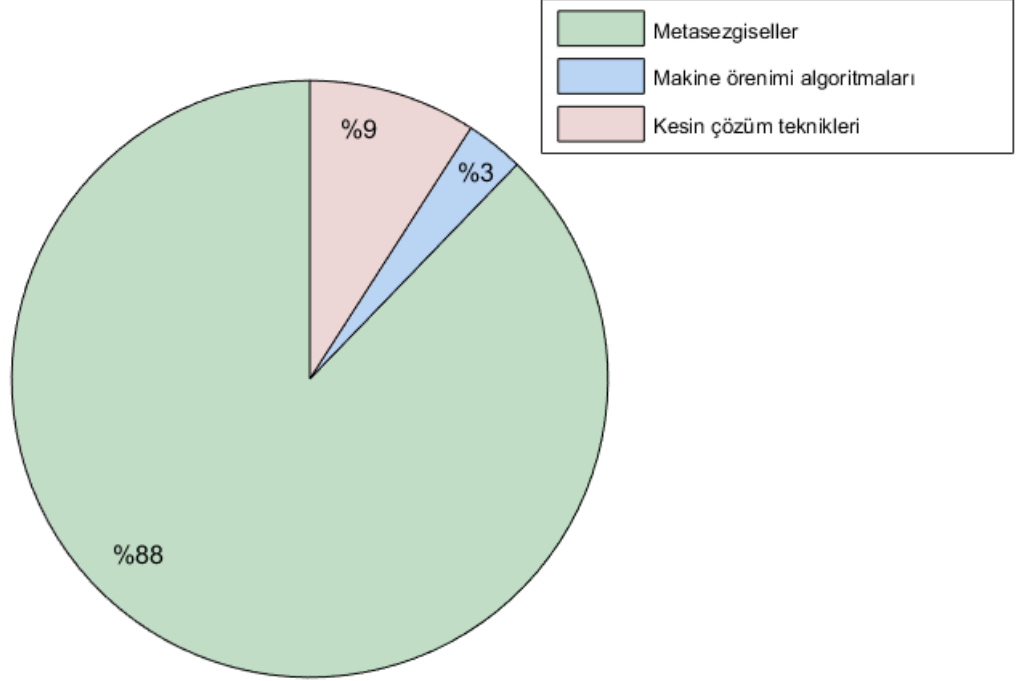
[Fernández ve Gómez \(2007\)](#) OVPO problemini çözmek için Hopfield ağı (HA) önermiştir ve GA, BT ve YA ([Chang vd. 2000](#)) ile kıyaslama sonuçları HA’nın daha iyi sonuçlar verdiğini göstermiştir. [Golmakani ve Alishah \(2008\)](#) bir yapay bağıklık sistemi (YBS) algoritması önermiş ve GA, BT, YA ([Chang vd. 2000](#)) ve HA ([Fernández ve Gómez 2007](#)) teknikleriyle kıyaslamıştır. Yazarlar YBS algoritmasının daha iyi sonuçlar elde etmesine rağmen çözüm elde etmek için daha fazla zamana ihtiyacı olduğunu belirtmiştir.

PO problemi için makine öğrenimi algoritması uygulayan yayınlar Tablo 9’da gösterilmiştir.

Tablo 9: PO problemi için makine öğrenimi algoritması uygulayan yayınlar

Makine öğrenimi algoritması	Yayınlar
Yapay sinir ağları	Fernández ve Gómez (2007) ; Freitas vd. (2009) ; Li vd. (2009) ; Suganya ve Vijayalakshmi Pai (2009)
YBS	Abbas ve Haider (2009) ; Golmakani ve Alishah (2008)
k-ortalamlar kümeleme	Jiang vd. (2014) ; Pai ve Michel (2009) ; Suganya ve Vijayalakshmi Pai (2009)

Bölüm 2.1, Bölüm 2.2 ve Bölüm 2.3’de PO problemini çözmek için uygulanan tekniklerin analizleri göstermiştir ki, NP-zor olduğu kanıtlanan bu problemin çözümü için araştırmacılar büyük çoğunlukla metasezgisel algoritmalara başvurmuşlardır (Şekil 3).



Şekil 3: PO problemi çözümü için uygulanan tekniklerin dağılımı

Çözüm teknikleri arasında, metasezgisel teknikler literatürde en çok tercih edilen yöntemlerdir. Bunun başlıca sebebi PO probleminin NP-zor olmasıdır. PO probleminin çözümü için GA ve PSO teknikleri yaygın olarak kullanılmıştır. Ancak, bu teknikler son zamanlarda pek de ilgi görmemiştir. Bu da göstermektedir ki, bu teknikler PO problemi için literatürdeki popülaritelerini yakın gelecekte kaybedebilirler. Onların yerine YAK algoritması gibi son zamanlarda popüler olan teknikler daha fazla kullanılmaya başlayabilir. Ayrıca diğer bir yakınsama tekniği olan makine öğrenimi algoritmaları PO probleminin çözümü için pek rağbet görmemiştir.

3. GEREÇ VE YÖNTEM

Bu bölümde veri setlerinin toplanması, matematiksel model, deneme çözümleri ve kesin çözüm algoritması ve literatürdeki mevcut yöntemlerin kodlanması iş paketleri detaylı olarak anlatılmıştır.

3.1. Veri setlerinin toplanması

Literatürde yaygın olarak karşılaştırma amaçlı kullanılan veri setleri OR-LIBRARY (<http://people.brunel.ac.uk/~mastjib/jeb/orlib/portinfo.html>) web sitesinden indirilmiş ve etkin sınır oluşturma çalışmaları başlatılmıştır. Bu veri setleri Mart 1992 ile Eylül 1997 tarihleri arasındaki sırasıyla, 31, 85, 89, 98 ve 225 adet hisse bulunduran Hang Seng, DAX 100, FTSE 100, S&P 100 ve Nikkei 225 problemlerinden oluşmaktadır.

Türkiye piyasalarındaki BIST30 ve BIST100 verilerine erişim için, öncelikle, Google Finans veri tabanını çevrimiçi kullanan bir prosedür geliştirilmiştir. Ancak, tarihler eskiye doğru gittikçe verilerde eksiklikler olduğu gözlemlenmiştir. Google Finans veri tabanındaki bilgilere göre BIST30 verileri ancak 01.01.2014 tarihinden itibaren ve BIST100 verileri ise ancak 01.01.2016 tarihinden itibaren eksiksiz çekilebilmektedir. Yahoo Finans veri tabanında da benzer veri eksiklikleri tespit edilmiştir. Bu durum, BIST30 ve BIST100 endekslerinin aynı tarih aralığında analizini olanaksız kılmıştır. Daha sonra, Bloomberg veri tabanından veri erişimi sağlanmıştır. Ancak, Bloomberg veri tabanında da, geçmiş tarihlerde hisselerin halka arzı, tek grupta birleştirme gibi olaylar nedeniyle belli hisselerin belli tarihlerde endekse yeni giriş veya endeksten çıkış yaptığı gözlemlenmiştir. Örneğin, araştırmalarımıza göre, güncel BIST100 endeksindeki hisselerden Carrefoursa'da A ve B Grubu paylarının birleşmesi nedeniyle serbest marjla 27.08.2015 tarihinden yeni işlem kodu CRFSA.E işlem görmeye başlamıştır (Gundem 2015). Başlangıç tarihi olarak 27.08.2015 alınması veri derinliğinde ciddi küçülmeye neden olacağı için, CRFSA.E kodlu hisse BIST100 endeksinden çıkarılmıştır. Ayrıca, halka arz sürecini tamamlayan Pegasus Hava Yolları, "PGSUS" kodu ile BIST30'da 26.04.2013 tarihinde işlem görmeye başlamıştır (Haber 2013b). Odaş Elektrik Üretim Sanayi Ticaret AŞ de "ODAS" kodu ile BIST100'de 21.05.2013 tarihinde işlem görmeye başlamıştır (Haber 2013a). Sonuç olarak, BIST30 verileri eksiksiz olarak 30 adet hisse ile 26.04.2013 tarihinden itibaren, BIST100 verileri ise 99 adet hisse ile 23.05.2013 tarihinden itibaren çekilebilmiştir. Tarih uyumu açısından BIST30 verileri de 23.05.2013 tarihine çekilerek deneyler ve analizler 23.05.2013 ile 18.04.2016 tarihleri arasındaki BIST30 ve BIST 100 verileri kullanılarak yapılmıştır.

BIST30 ve BIST100 veri setleri, diğer araştırmacıların algoritma performanslarını bu proje kapsamında geliştirilen algoritmalar ile kıyaslanmasına olanak sağlamak için, www.pau.edu.tr/portfolio adresinde yayınlanmıştır.

3.2. Matematiksel model, deneme çözümleri ve kesin çözüm algoritması

Portföy seçimi problemi matematiksel ifadeyle, ikinci dereceden amaç fonksiyonu ve lineer kısıtlarla gerçek değerli değişkenlerin optimizasyonu problemidir. Bu nedenle, portföy seçimi iki çakışan hedefin çok amaçlı bir optimizasyon görevidir: kar maksimizasyonu ve risk minimizasyonu. Bu iki amaç aynı anda başarılamayacağı için bu problem pareto optimal çözümlerin kümesi olarak tanımlanan etkin sınırı belirlemek olarak karşımıza çıkar. Bir portföy, eğer verilen bir risk için yatırım ağırlıklarını ayarlayarak daha büyük bir getiri sağlayamıyorsa, o portföy etkin sınırdadır. O portföy aynı zamanda beklenen getirinin değeri için riski minimize edendir de denebilir.

Portföy optimizasyonu problemi en basit şekliyle standart sayısal teknikler ile kolayca çözülebilen bir problemdir. Ancak, çeşitlendirmenin avantajlarından yararlanmak ve toplam riski azaltmak için fazla sayıda varlıklara küçük miktarlarda yatırım yapan portföy oluşturmakta bir takım zorluklar bulunmaktadır. Bu tip yatırım stratejisi, yüksek işlem maliyetleri ve fazla sayıda varlığı yönetmenin zorluğundan dolayı pratikte çok zordur. Bu zorluğu aşmak için sermayenin varlıklara dağıtılması üzerine birçok kısıt getirilebilir. Portföydeki varlık sayısını sınırlanabilir ya da her varlığa yatırılan sermayenin oranlarına alt ve üst limitler koyulabilir. Bu kısıtlar, problemin standart optimizasyon teknikleriyle çözülmesini zorlaştırmakta ve nitekim, problemi NP-Zor haline getirmektedir.

En basit haliyle standart Markowitz Ortalama- Varyans (O-V) modeli (Markowitz 1952, 1959) kullanılarak elde edilen portföy optimizasyonu probleminin matematiksel formülasyonu aşağıda verilmiştir:

Parametreler:

- N Uygun varlıkların sayısı
 μ_i i . varlığın beklenen getirisi
 σ_{ij} i . ve j . varlıkların arasındaki kovaryans değeri
 R^* İstenen düzeydeki beklenen getiri

Karar Değişkenleri:

w_i i . varlığın oranı

$$\text{Amaç Fonksiyonu: } \min \sum_{i=1}^N \sum_{j=1}^N w_i w_j \sigma_{ij} \quad (1)$$

$$\text{Kısıtlar: } \sum_{i=1}^N w_i \mu_i = R^* \quad (2)$$

$$\sum_{i=1}^N w_i = 1 \quad (3)$$

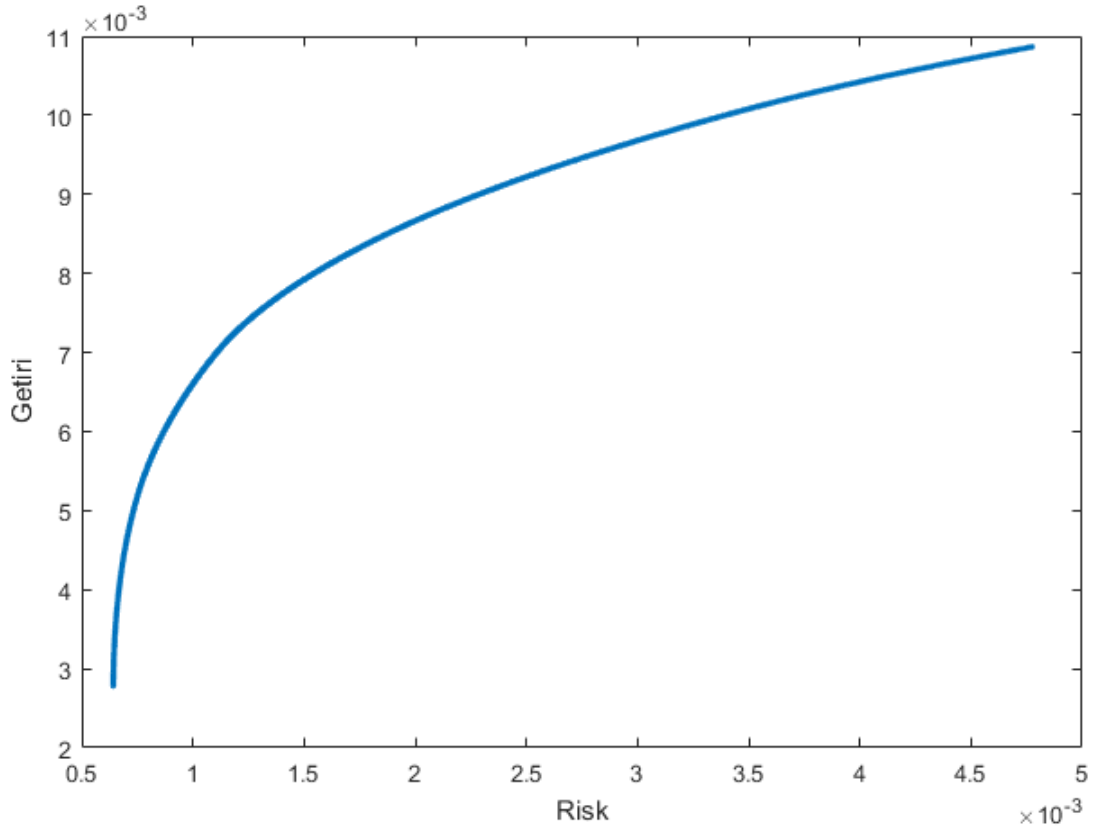
$$0 \leq w_i \leq 1, \quad i = 1, \dots, N \quad (4)$$

Denklem (1), portföydeki toplam riski minimize ederken, denklem (2) ise portföyün R^* beklenen getirisini sağlamasını garanti eder. Denklem (3) ise portföyde kullanılan varlıkların oranlarının toplamının 1 olması kısıtını ifade eder. Bu formülasyon ile herhangi bir veri seti için optimal çözümün hesaplanması pratikte mümkün olabilmektedir. Bu modeli çözerek, R^* hedeflenen getirisinin değişen değerlerine göre kesiksiz artan bir eğri ile etkin sınır bulunabilir. Bu etkin sınır, elde edilmesi hedeflenen getiriye karşılık katlanılması gereken riskin en iyi dengesini ifade eder. Etkin sınırı bir λ ağırlıklandırma parametresi ile takip etmek mümkündür (Chang vd. 2000). Denklem (5)' deki λ parametresi 0 olduğunda, riske bakılmaksızın beklenen getiri maksimum olmakta ve optimal çözüm en yüksek getiri veren bir adet varlıktan oluşmaktadır. λ parametresi 1 olduğunda ise, beklenen getiriye bakılmaksızın risk minimum olmakta ve optimal çözüm birden fazla varlıktan oluşabilmektedir. λ parametresinin $0 < \lambda < 1$ değerleri için, $\lambda = 0$ ve $\lambda = 1$ sınır değerleri arasında getiri ile risk arasında ödünleşerek etkin sınır üzerindeki noktaları oluşturur. Bu noktaları birleştiren eğri, etkin sınır eğrisidir. Hang Seng için literatürde yer alan kısıtsız etkin sınır Şekil 4'de verilmiştir.

$$\text{Amaç Fonksiyonu: } \min \lambda \left[\sum_{i=1}^N \sum_{j=1}^N w_i w_j \sigma_{ij} \right] - (1 - \lambda) \left[\sum_{i=1}^N w_i \mu_i \right] \quad (5)$$

$$\text{Kısıtlar: } \sum_{i=1}^N w_i = 1 \quad (6)$$

$$0 \leq w_i \leq 1 \quad i = 1, \dots, N \quad (7)$$



Şekil 4: Hang Seng veri seti için kısıtsız etkin sınır

Denklem (5) - (7) ile verilen formülasyona portföye girmesi istenilen varlık sayısı kısıtı ile bir hissenin alması gereken minimum ve maksimum ağırlık kısıtları eklendiğinde eleman sayısı kısıtlı portföy optimizasyonu problemi elde edilmektedir. Eklenen kısıtlar ve karar değişkeni aşağıdaki karma tamsayı doğrusal olmayan programlama modeline (MINLP) ilişkin formülasyonla gösterilmiştir:

Modele eklenen parametreler:

- K Portföydeki istenen varlık sayısı
- ε_i i . varlığın portföydeki minimum ağırlığı
- δ_i i . varlığın portföydeki maksimum ağırlığı

Modele eklenen karar değişkeni:

$$z_i = \begin{cases} 1 & \text{Eğer } i. \text{ varlık portföyde ise} \\ 0 & \text{Aksi halde} \end{cases}$$

$$\text{Amaç Fonksiyonu: } \min \lambda \left[\sum_{i=1}^N \sum_{j=1}^N w_i w_j \sigma_{ij} \right] - (1 - \lambda) \left[\sum_{i=1}^N w_i \mu_i \right] \quad (8)$$

$$\text{Kısıtlar: } \sum_{i=1}^N w_i = 1 \quad (9)$$

$$\sum_{i=1}^N z_i = K \quad (10)$$

$$\varepsilon_i z_i \leq w_i \leq \delta_i z_i \quad i = 1, \dots, N \quad (11)$$

$$z_i \in (0,1) \quad i = 1, \dots, N \quad (12)$$

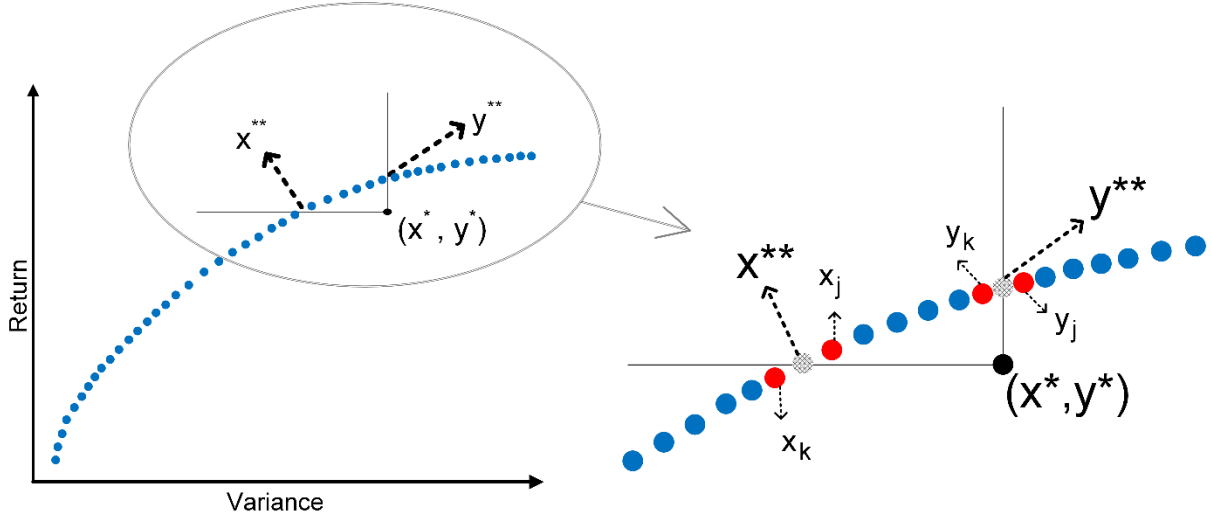
$$0 \leq w_i \leq 1, \quad i = 1, \dots, N \quad (13)$$

$$0 \leq \varepsilon_i \leq \delta_i \leq 1, \quad i = 1, \dots, N \quad (14)$$

Denklem (8) ve Denklem (9) kısıtsız modelden eklenmiş olup, denklem (10), portföyde toplam K sayıda varlığın bulunmasını garanti ederken, denklem (11) ise portföye alınan varlığın ağırlığının belirlenen minimum ve maksimum değerler arasında olmasını sağlar. Denklem (12) de karar değişkeninin 0 veya 1 olmasını sağlayan bütünlük kısıtıdır. Denklem (13) ile ifade edilen kısıt bir varlığın ağırlığının 0 ile 1 arasında olmasını sağlarken, Denklem (14) bir varlığın alabileceği minimum ve maksimum ağırlık kısıtlarını sağlamaktadır.

Literatürde algoritmaların oluşturduğu kısıtlı etkin sınır ile standart kısıtsız etkin sınır arasındaki hataları hesaplamak yaygın olarak kullanılan performans ölçütleri bu projede de kullanılmıştır. Bu ölçütler: Yüzde Hatasının Ortalaması (YHO), Yüzde Hatasının Minimumu (YHMİN), Yüzde Hatasının Maksimumu (YHMAKS), Yüzde Hatasının Medyanı (YHMED), Getiri Hatasının Varyansı 1(GHV-I), Ortalama Getiri Hatasını 1(OGH-1), Ortalama Öklid Uzaklığı (OÖU), Getiri Hatasının Varyansı 2(GHV-II) ve Ortalama Getiri Hatasını 2(OGH-II).

(x_i, y_i) , $(i = 1, \dots, 2000)$, standart etkin sınır üzerindeki risk ve getiri değerlerini temsil etmekte ve (x^*, y^*) , $(j = 1, \dots, E)$, algoritmalar tarafından üretilen etkin sınır üzerindeki risk ve getiri değerlerini temsil etmektedir (Şekil 5). Basit interpolasyon kullanılarak, geliştirilen algoritma tarafından elde edilen noktaların, standart etkin sınıra göre olan hataları hesaplanabilir (Chang vd. 2000).



Şekil 5: Temsili bir hata hesaplama

$$\varphi_j = 100 \left| \frac{y^* - y^{**}}{y^{**}} \right| \quad \text{/*getirin için yüzde hata*/} \quad (15)$$

$$\omega_j = 100 \left| \frac{x^* - x^{**}}{x^{**}} \right| \quad \text{/*varyans için yüzde hata*/} \quad (16)$$

$$x^{**} = x_k + (x_j - x_k) \left[\frac{(y^* - y_k)}{(y_j - y_k)} \right] \quad \text{/*}(x^*, y^*) \text{ noktasının standart etkin sınır} \quad (17)$$

üzerine yatay izdüşümü*/

$$y^{**} = y_k + (y_j - y_k) \left[\frac{(x^* - x_k)}{(x_j - x_k)} \right] \quad \text{/*}(x^*, y^*) \text{ noktasının standart etkin sınır} \quad (18)$$

üzerine dikey izdüşümü*/

$$y_j = \min[y_i | y_i \geq y^*] \quad (19)$$

$$y_k = \max[y_i | y_i \leq y^*] \quad (20)$$

$$x_j = \min[x_i | x_i \geq x^*] \quad (21)$$

$$x_k = \max[x_i | x_i \leq x^*] \quad (22)$$

Yukarıda verilen formülasyonları kullanarak elde edilen performans ölçütleri aşağıda verilmiştir:

YHO, YHMED, YHMİN, YHMAKS (Chang vd. 2000):

$$YHO = \frac{\sum_{j=1}^E \min(\varphi_j, \psi_j)}{E} \quad (23)$$

$$\psi_j = 100 \left| \frac{(\sqrt{x^{**}} - \sqrt{x^*})}{x^{**}} \right|$$

$$YHMED = \min\{\text{medyan}\{\varphi_1, \varphi_2, \dots, \varphi_E\}, \text{medyan}\{\psi_1, \psi_2, \dots, \psi_E\}\} \quad (24)$$

$$YHMİN = \min\{\min\{\varphi_1, \varphi_2, \dots, \varphi_E\}, \min\{\psi_1, \psi_2, \dots, \psi_E\}\} \quad (25)$$

$$YHMAKS = \max\{\text{maks}\{\varphi_1, \varphi_2, \dots, \varphi_E\}, \text{maks}\{\psi_1, \psi_2, \dots, \psi_E\}\} \quad (26)$$

GHV-I OGH-I (Fernández ve Gómez 2007):

$$GHV-I = \frac{\sum_{j=1}^E \varphi_j}{E} \quad (27)$$

$$OGH-I = \frac{\sum_{j=1}^E \omega_j}{E} \quad (28)$$

(x^{***}, y^{***}) , standart etkin sınır üzerindeki noktalardan (x^*, y^*) noktasına en yakın noktadır. OÖU, GHV-II, OGH-II (Cura 2009):

$$OÖU = \frac{\sum_{j=1}^E \sqrt{(x^{***}-x^*)+(y^{***}-y^*)}}{E} \quad (29)$$

$$GHV-II = \frac{\sum_{j=1}^E \frac{100|(x^{***}-x^*)|}{x^*}}{E} \quad (30)$$

$$OGH-II = \frac{\sum_{j=1}^E \frac{100|(y^{***}-y^*)|}{y^*}}{E} \quad (31)$$

Eleman sayısı kısıtlı portföy optimizasyonu problemi için Denklem (8) - (14)'de verilen matematiksel model, yüksek seviye bir modelleme dili olan, GAMS (*General Algebraic Modeling System*) yardımı ile kodlanmıştır. Tablo 10'da eleman sayısı kısıtlı portföy optimizasyonu problemi için geliştirilen GAMS programı kaynak kodları yer almaktadır. GAMS programında, MINLP modelinin çözümünde kullanılan COUNNE ve BONMIN çözümleri ile çözüm aranmıştır. Ayrıca eleman sayısı kısıtlı portföy optimizasyonu problemi için Denklem (8) - (14)'de verilen matematiksel model, doğası gereği aynı zamanda kuadratik programlama kullanılarak da çözümlenebilmektedir. Bu kapsamda doğrusal programlama modellerinde olduğu kadar kuadratik programlama modellerinin çözümünde de yaygın olarak kullanılan CPLEX çözümleri ile de çözüm aranmıştır. BONMIN çözümleri ile problem ayrıca kuadratik programlama modeli olarak da çözülmüştür.

GAMS ara yüzünde yapılan işlemlerde sapma miktarları (Optca ve Optcr) hassasiyeti 0.0000 noktasına çekilerek çözümler çalıştırılmıştır (Tablo 10).

Tablo 10: Eleman sayısı kısıtlı portföy optimizasyonu problemi için geliştirilen GAMS kaynak kodları

```

SCALAR
  lambda      Risk değişkeni
  StockMax    Portföydeki hisse sayısı /10/;
POSITIVE VARIABLES
  x(i)        Hisselerin ağırlıkları;
PARAMETER
  xlow(i)     Belirtilen minimum ağırlık değeri;

  xlow(i) = 0.01;
BINARY VARIABLE
  Y(i)        Portföyde yatırım yapılan hisseleri belirten ikili değişken;
VARIABLES
  PortVariance    Portföy varyansı
  PortReturn      Portföy getirisi
  Z               Amaç fonksiyonun değeri;
EQUATIONS
  ReturnDef       Portföy getirisi kısıtı
  VarDef          Portföy varyansı kısıtı
  NormalCon       Portföydeki ağırlıkların toplamını 1'e eşitleyen kısıt
  LimitCon        Portföyde istenilen varlık sayısı kısıdı
  LoBounds(i)    Ağırlıkların minimum değerleri
  ObjDef          Amaç fonksiyonu;
ReturnDef .. PortReturn =1= SUM(i, ExpectedReturns(i)*x(i));
VarDef .. PortVariance =E= SUM((i,j), x(i)*VarCov(i,j)*x(j));
LoBounds(i).. x(i) =G= xlow(i)* Y(i);
NormalCon .. SUM(i, x(i)) =E= 1;
LimitCon .. SUM(i, Y(i)) =E= StockMax;
ObjDef .. Z =E= (1-lambda) * PortReturn - lambda * PortVariance;
MODEL LAMBDAPORT /Returndef, VarDef, NormalCon, ObjDef, UpBounds, LoBounds,
LimitCon/ ;
SCALAR StartTime; StartTime=jnow;
SET p/1*51/
PARAMETERS
  PortfolioVariance(p)
  PortfolioReturn(p)
  Optimalallocation(p,i)
  RiskWeight(p)
  SummaryReport(*,*)
  AssetsOnPort(p,i) ;
RiskWeight(p) = (ORD(p)-1)/(CARD(p)-1);
FILE Results /Results.txt/;
PUT Results ;
PUT 'Variance      Return'//;
LOOP(p,
  lambda= RiskWeight(p);
  SOLVE LAMBDAPORT MAXIMIZING Z USING MINLP;
  PortfolioVariance(p) = PortVariance.l;
  PortfolioReturn(p) = PortReturn.l;
  Optimalallocation(p,i)= x.l(i);
  put PortfolioVariance(p):10:8'      'PortfolioReturn(p):10:8/;
);
SCALAR Elapsed; Elapsed=(jnow-StartTime)*24*3600;
DISPLAY PortfolioVariance, PortfolioReturn, Optimalallocation,elapsed;
PUTCLOSE Results;
SummaryReport(i,p) = OptimalAllocation(p,i);
SummaryReport('Variance',p) = PortfolioVariance(p);
SummaryReport('Return',p) = PortfolioReturn(p);

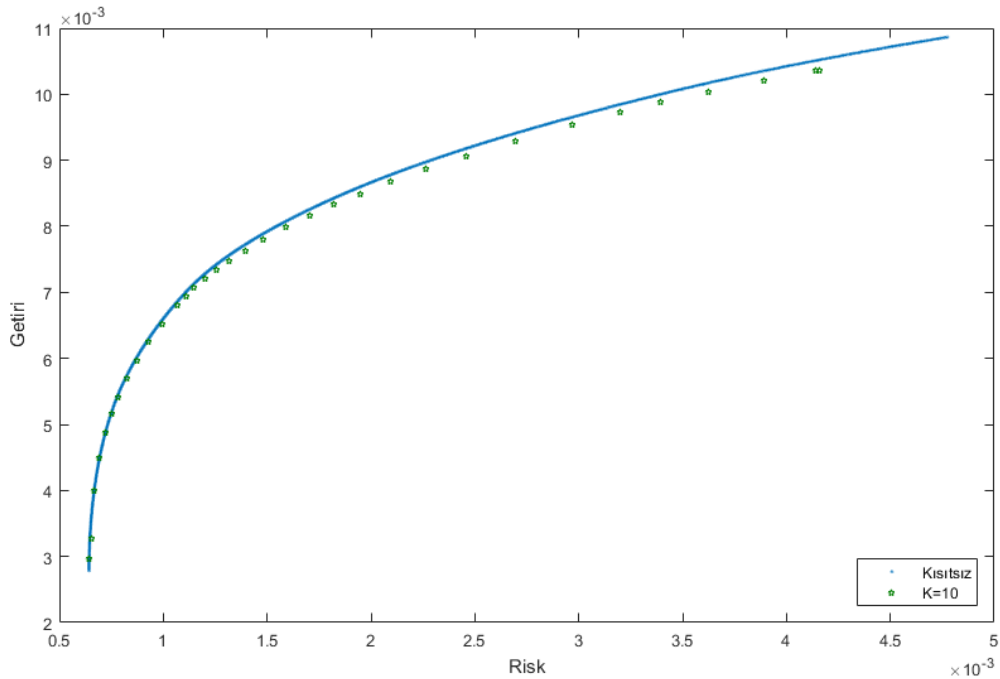
```

Kodlanan modelin doğrulanması ve geçerliliğinin sınanması için küçük çaplı bir ver seti olan Hang Seng indeksine ait hisseler kullanılmıştır. Modelde portföy içerisindeki varlık sayısı kısıtı $K = 10$ olarak alınmıştır. Portföydeki varlıkların minimum ağırlığı ise $\varepsilon = 0.01$ olarak kabul edilmiştir. Kullanılan 3 farklı çözücü ile küçük çaplı veri seti için elde edilen sonuçlar OÖU,

GHV-II(%), OGH-II(%) ve çözüm zamanlarına göre Tablo 11’de yer almaktadır. Bu çözücüler arasında COUNNE çözücüsü, Hang Seng veri seti için önceden belirlenmiş bir zaman limitinde (2 saat) çözüm üretememiştir. Diğer çözücüler arasında en iyi çözümü sunan çözücü CPLEX çözücüsüdür. CPLEX çözücüsünde MINLP problem tipi mevcut değildir. Hang Seng için literatürde yer alan kısıtsız ve CPLEX ile elde edilen kısıtlı etkin sınırlar Şekil 6’da verilmiştir.

Tablo 11: Farklı çözücüler için Hang Seng veri seti ile GAMS çözümleri

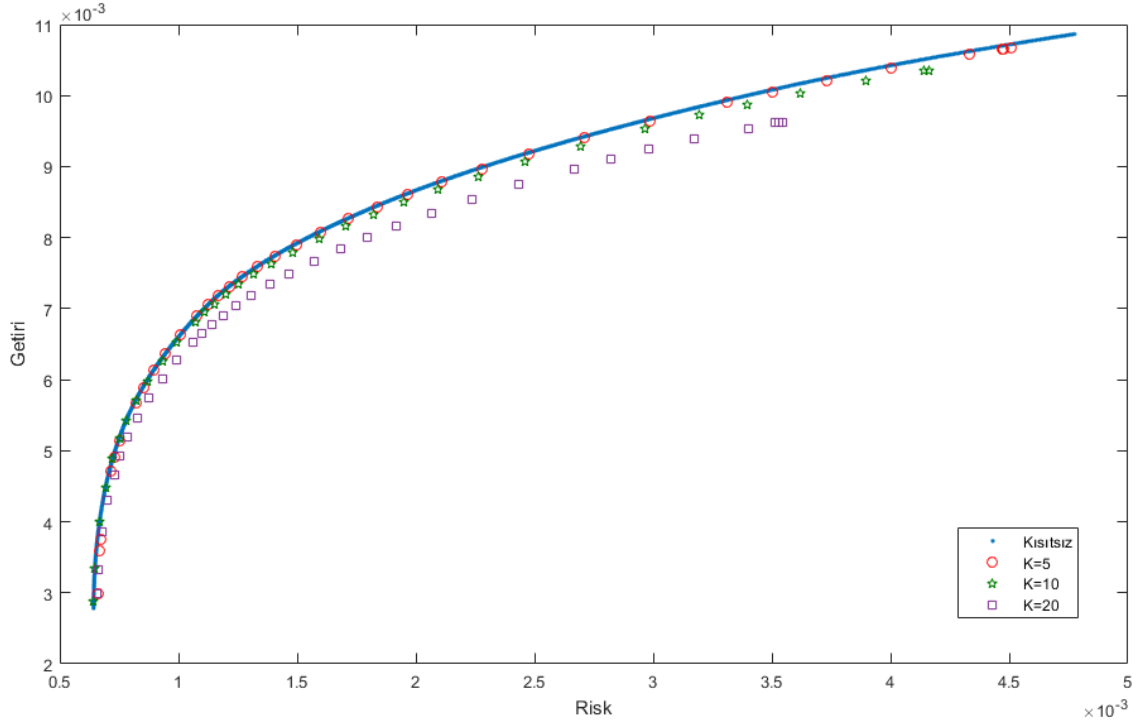
Hatalar	MINLP				MIQCP			
	OÖU	GHV-II (%)	OGH-II (%)	Zaman (s)	OÖU	GHV-II (%)	OGH-II (%)	Zaman (s)
Çözücü								
COUNNE	-	-	-	-	-	-	-	-
BONMIN	0.0001	1.6463	0.6019	18.832	0.0001	1.6463	0.6020	18.930
CPLEX		Mevcut değil			0.0001	1.6275	0.6034	8.746



Şekil 6: Hang Seng veri setinde $K = 10$ değeri için GAMS ile bulunan etkin sınır

Şekil 7’de görüldüğü gibi kısıtsız modele, toplam varlık sayısı (K) kısıtı ve minimum ağırlık (ε) kısıtı eklendiğinde, kısıtlı etkin sınır üzerindeki noktalar kısıtsız etkin sınırdan uzaklaşmıştır. Bunun sebebi modele portföyde K adet varlık bulundurması zorunluluğu getirilmesidir. Model kısıtsız olduğunda, portföy bir risk değerindeki K ’dan az sayıda varlık içerirken, kısıtlandırıldığı zaman varlık sayısını K ’ya tamamlamak için getirisi daha düşük varlıklara yatırım yapmak zorunda kalabilmektedir. Bu da o risk değeri için etkin sınırdaki noktayı daha az getiri

sağlayacak şekilde aşağıya çekmektedir. Eğer kısıttaki K değeri artırılırsa, model portföye daha az getirisi olan varlıklardan daha fazla koyacağı için kısıtlı etkin sınır ve kısıtsız etkin sınırdaki noktalar arasındaki mesafe artacaktır. Şekil 7’de Hang Seng veri seti için K değerinin sırası ile 5, 10 ve 20 olduğu durumlar için GAMS CPLEX ile bulunan kısıtlı sınırlar ve literatürde yer alan kısıtsız etkin sınır yer almaktadır. Model, düşük risk seviyeleri için beklenen getiriyi rahatlıkla karşılayabiliyor iken, risk seviyesi arttığında beklenen getirinin karşılanamaması nedeniyle noktalar arasındaki mesafe artmaktadır. Tablo 12’de farklı K değerleri için GAMS CPLEX ile bulunan sonuçların belirlenen performans ölçütlerine göre analizi de bu durumu özetlemektedir.



Şekil 7: Hang Seng veri setinde farklı K değerleri için GAMS ile bulunan etkin sınırlar

Tablo 12: Hang Seng veri setinde farklı K değerleri için GAMS ile bulunan sonuçların performans analizi

	Performans Ölçütü:		
	OÖU	GHV-II (%)	OGH-II (%)
5	0.0000	0.5365	0.1665
K 10	0.0001	1.6282	0.6037
20	0.0002	6.9439	1.7929

Küçük çaplı veri setlerinin optimal olarak çözümünde genel olarak başarı sağlayan test edilen çözümler, GAMS dilinde DAX 100 gibi karmaşık veri setlerinin çözümünde ise başarı sağlayamamıştır. Bu sonuçlar göstermektedir ki kesin çözüm üreten bu araçlar, ele alınan probleme ilişkin veri setleri büyüdüğünde, karmaşıklıkları arttığında ve/veya yeni kısıtlar probleme eklendiğinde yetersiz kalabilmektedir. Bu nedenle, kısa sürelerde yaklaşık çözüm

üretebilen, büyük çaplı karmaşık veri setlerinin çözümünde kullanılabilecek sezgisel yaklaşımlı algoritmaların geliştirilmesine ihtiyaç duyulmaktadır.

Proje raporunun bir sonraki bölümde literatürdeki en çok uygulanmış olan yöntemlerin kodlanması aşaması ile devam etmektedir.

3.3. Literatürdeki mevcut yöntemlerin kodlanması

Yapılan kapsamlı literatür çalışmasına göre, ortalama-varyans modellenli portföy optimizasyonu probleminin çözümü için genetik algoritma (GA) ile parçacık sürü optimizasyonu (PSO) metotları en çok çalışılan ve iyi sonuçlar alınmış yöntemler olarak göze çarpmaktadır. Bu nedenle, etkin algoritma tasarımına ışık tutması ve referans noktası olması beklenen bu iki metot kodlanmış ve sonuçları raporlanmıştır. GA, PSO ve altıncı iş paketinde detayları verilen yapay arı kolonisi (YAK) algoritmalarında ortak olarak kullanılan prosedürler sıradaki bölümde detaylandırılmıştır. Bu sayede, aynı değerlendirme ve tamir prosedürlerini kullanan algoritmaların performanslarını adil bir şekilde karşılaştırmak mümkün olabilmektedir.

3.3.1. Algoritmalarda kullanılan ortak prosedürler

Kodlanan algoritmaların ortak kullandığı iki prosedür oluşturulmuştur. Bunlardan ilki tamir fonksiyonudur (Şekil 8). Algoritmaların ürettiği çözümler istenilen kısıtları her zaman karşılayamayabilir. Yani, toplam yatırım yapılan hisse senedi sayısı K sayısına eşit olmayabilir ya da bir hisseye ε değerinden daha az ağırlık verilmiş olabilir. Böyle çözümler uygunsuz çözümlerdir ve tamir prosedürü yardımıyla uygun çözüm haline getirilmelidir. Tamir prosedürünün ilk aşaması portföydeki toplam yatırım yapılan hisse sayısını K' 'ya eşitlemektir. Eğer portföydeki yatırım yapılan hisse sayısı K değerinden büyük ise, K' 'ya eşit olana kadar en küçük ağırlığa sahip hisseden başlayarak hisselerin ağırlıklarını ve o varlık için oluşturulan ikili değişkeni sıfır yapar. Eğer portföydeki yatırım yapılan hisse sayısı K değerinden küçük ise, K' 'ya eşit olana kadar veri setinde yatırım yapılmayan rasgele bir hisselerin ikili değişkenini bir yapar. Böylelikle, portföyde bulunması gereken toplam hisse senedi sayısı kısıtı sağlanmış olur. İkinci aşamada ise, yatırım yapılan hisselerin minimum ağırlıkları ε değerine eşitlenmektedir. Öncelikle, portföydeki K adet yatırım yapılan hisselerin ağırlıkları toplanır (CS). Daha sonra portföyde bir varlığın alabileceği maksimum ağırlık (FP) hesaplanır. Daha sonra her hisselerin ağırlığı, o hisselerin ağırlığı ile FP/CS oranının çarpımının ve ε değerinin toplamına eşitlenir. Bu iki aşama sonucunda ilk başta uygunsuz olan bir çözüm hem toplam yatırım yapılan hisse sayısı kısıtını hem de minimum ağırlık kısıtını sağlayarak uygun bir çözüm haline gelir.

```

1:  Prosedür: Tamir
2:  Girdi:  $N, K, S, W$ 
3:  Çıktı:  $W^r$ 
4:   $N$ : İndeksteki toplam varlık sayısı
5:   $K$ : Portföyde bulunmasına izin verilen toplam varlık sayısı
6:   $S$ : İndeksteki varlıklar kümesi
7:   $W$ : Tamire giren portföyün ağırlıkları kümesi
8:   $w_i$ : Tamire giren  $i$ . varlığın ağırlığı  $i \in S$ 
9:   $w_i^r$ : Tamirden çıkan  $i$ . varlığın ağırlığı  $i \in S$ 
10:  $W^r$ : Tamire giren portföyün ağırlıkları kümesi
11:  $z_i$ :  $i$ . varlığın portföyde yer alıp almama durumunu gösteren ikili değişken,  $i \in S$ 
12: Başla
13: Tekrar et
14:   Eğer  $\sum_{i=1}^N z_i > K$ 
15:     Portföyde bulunan en küçük  $w_i$  değerine sahip varlık için  $z_i = 0$ ,  $w_i = 0$  yap,  $i \in S$ 
16:   Eğer  $\sum_{i=1}^N z_i < K$ 
17:     Portföyde bulunmayan rasgele bir  $z_i$  değeri için  $z_i = 1$  yap,  $i \in S$ 
18:    $\sum_{i=1}^N z_i = K$  olana kadar
19:    $CS = \sum_{i=1}^N w_i$   $i \in S$ , /* portföydeki varlıkların ağırlıkları toplamı */
20:    $FP = 1 - \varepsilon * K$  /* portföydeki varlıklara atanabilecek maksimum ağırlık */
21:    $w_i^r = \varepsilon * z_i + w_i * z_i * FP / CS$   $\forall i \in W, \forall i \in W^r$ 
22: Bitir

```

Şekil 8: Uygunsuz çözüm tamir prosedürü

Ortak kullanılan ikinci prosedür ise uygunluk hesaplama prosedürüdür (Şekil 9). Bu prosedür sayesinde bir çözümün toplam getirisi, riski ve uygunluk değeri hesaplanır. İlk olarak portföydeki her varlık ile o varlığın ortalama getirisi çarpımları toplamı ile portföyün ortalama getirisi hesaplanır. Daha sonra varyans-kovaryans matrisi kullanılarak portföyün riski hesaplanır. Varyans-kovaryans matrisi veri setindeki varlıkların birbirleriyle ilişkisini gösteren matristir. Bunun için yatırım yapılan her varlığın ağırlığı teker teker portföydeki diğer varlıkların ağırlıkları ve o iki varlığın varyans-kovaryans değeri ile çarpılır. Bu çarpımlar toplamı toplam riski belirler. Son olarak uygunluk değeri hesaplanır. Uygunluk değeri hesaplanırken bir ağırlıklandırma kullanılır. Böylece iki amaç fonksiyonu (maksimum getiri, minimum risk) tek bir amaç fonksiyonu olarak tanımlanır. Ağırlık değeri (λ) 0'dan başlayarak 1'e kadar istenilen aralıklarla ilerler. $\lambda = 0$ olduğunda portföy maksimum getiriye amaçlar, $\lambda = 1$ olduğunda ise amaç minimum risktir.

1:	Prosedür: Uygunluk hesapla	
2:	Girdi: $W^r, N, VC_{ij}, R_i, \lambda$	
3:	Çıktı: R^*, V^*, f	
4:	W^r : Varlıkların ağırlıkları kümesi	
5:	N : İndeksteki toplam varlık sayısı	
6:	f : Portföyün uygunluk değeri	
7:	R^* : Portföyün toplam getirisi	
8:	V^* : Portföyün varyansı (Riski)	
9:	VC_{ij} : i . ve j . varlıklar arasındaki varyans-kovaryans değeri $i, j \in S$	
10:	w_i^r : i . Varlığın ağırlığı $w_i^r \in W^r$	
11:	R_i : i . Varlığın ortalama getirisi $i \in S$	
12:	Başla	
13:	$R^* = \sum_{i=1}^N w_i^r R_i$	<i>/* portföyün toplam getirisi */</i>
14:	$V^* = \sum_{i=1}^N \sum_{j=1}^N w_i^r w_j^r VC_{ij}$	<i>/* portföyün toplam risk değeri */</i>
15:	$f = \lambda V^* - (1 - \lambda)R^*$	<i>/* λ ile hesaplanan uygunluk değeri */</i>
16:	Bitir	

Şekil 9: Uygunluk Hesaplama Prosedürü

3.3.2. Genetik Algoritma (GA)

İlk defa [Holland \(1975\)](#) tarafından ortaya atılan genetik algoritmalar (GA) karmaşık optimizasyon problemlerini çözmek için uyarlanabilen bir arama metodudur. Biyolojik doğal seçim ve genetik kalıtım prensiplerini temel alarak bir çözümler popülasyonunu nesiller boyunca evrimleştiren yinelemeli arama prosedürleri ile yol alan GA, popülasyondaki mevcut çözümlerin seçim, çaprazlama ve mutasyon işlemleriyle yeni bir çözümler kümesi oluşturulur. Popülasyondaki uygunluğu değerlendirilen her birey eldeki probleme potansiyel bir çözümü temsil eder. Bireyler çaprazlama ve mutasyon gibi genetik operatörler vasıtasıyla stokastik bir dönüşüm geçirerek yeni bireyler oluşturur. Çaprazlama, çözümler arasındaki genetik materyallerin rasgele, ancak yapılandırılmış değişimi ile iyi çözümlerden daha iyi çözümlerin oluşturulmasına olanak sağlar. Mutasyonun rolü ise algoritmanın olgunlaşmamış yakınsama göstermesini (yerel optimal sonuçlara takılması) engellemek için kaybedilmiş ya da keşfedilmemiş genetik materyallerin kazanılmasını sağlar. Birçok iterasyondan sonra, algoritma optimal ya da optimele yakın bir çözüme işaret eden en iyi bireye yakınsar. Genetik algoritmaların en temel adımları [Şekil 10](#)'da verilmiştir. [Goldberg \(1989\)](#) genetik algoritmaların arama ve optimizasyon konularındaki geniş uygulama alanlarına dikkat çekmiştir. Nitekim, literatür özeti bölümünde de belirtildiği gibi GA, portföy optimizasyonu araştırma alanında da yaygın olarak kullanılmıştır.

Adım	Açıklama
1	Başlangıç popülasyonunu rasgele yarat.
2	Popülasyondaki her bireyin uygunluk değerini hesapla.
3	Genetik operatörleri uygulamak için bireyler kümesini seç.
4	Genetik operatörleri uygula ve oluşan yeni bireylerin uygunluklarını değerlendir.
5	Uygunluk değerlerine göre yeni nesli oluştur.
6	Durdurma ölçütü henüz sağlanmadıysa adım 3'e git.
7	Evrimi sonlandır ve sonuçları raporla.

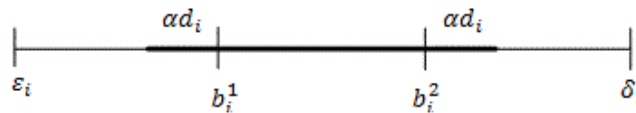
Şekil 10: Genetik Algoritmalar'ın ana adımları

Bu projede geliştirilen GA, popülasyon içerisinde çeşitliliği sağlamak için 2 tane operatör kullanılmaktadır. Bunlar çaprazlama ve mutasyon operatörleridir. Çaprazlama operatörü için Şekil 11'de verilen $BLX - \alpha$ operatörü (Picek vd. 2013) kullanılmıştır.

1:	Prosedür: Çaprazlama
2:	Girdi: S, N, P_1, P_2 ,
3:	Çıktı: OS
4:	S : İndeksteki varlıklar kümesi
5:	N : İndeksteki toplam varlık sayısı
6:	P_1, P_2 : Rulet tekeri yöntemiyle popülasyondan seçilen ebeveynler.
7:	b_i^1, b_i^2 : Ebeveynlerin i . elemanı için ağırlıkları, $b_i^1 \in P_1, b_i^2 \in P_2, i \in S$
8:	os_i : çaprazlama sonucu oluşan çocuğun i . elemanının ağırlığı, $os_i \in OS$
9:	α : pozitif bir parametre
10:	r : düzgün rassal sayı, $r \in [0,1]$
11:	Başla
12:	$d_i = b_i^1 - b_i^2 , \forall i \in S$
13:	$os_i = (\min(b_i^1, b_i^2) - \alpha d_i) + r * [(\max(b_i^1, b_i^2) + \alpha d_i) - (\min(b_i^1, b_i^2) - \alpha d_i)] , \forall i \in S$
14:	Bitir

Şekil 11: Çaprazlama Prosedürü

İlk olarak $BLX - \alpha$ çaprazlama, iki bireyin i . ağırlıklarının mutlak farklarını (d_i) hesaplar. Daha sonra küçük olan ağırlıktan $\alpha * d_i$ kadar dışa doğru ve büyük ağırlıktan $\alpha * d_i$ kadar dışa doğru uzaklıkta iki nokta belirler (Şekil 12). Daha sonra bu iki nokta arasında rasgele bir nokta seçerek çocuğun i . elemanın ağırlığını belirler.



Şekil 12: $BLX - \alpha$ çaprazlama operatörü

GA'nın kullandığı diğer operatör ise mutasyon operatörüdür. Mutasyon operatörü çaprazlamadan sonra belirli bir ihtimalle yapılır ve popülasyona farklı genler kazandırır. Önerilen algoritmada Şekil 13'de verilen gaussian mutasyon (Yao vd. 1999) kullanılmıştır. Bu

mutasyon, bir parametre içermektedir (mp) ve kromozomun rasgele bir geninin belirli bir miktarda rasgele azaltarak ya da artırarak değiştirir.

-
- 1: **Prosedür:** Mutasyon
 - 2: **Girdi:** S, N, OS
 - 3: **Çıktı:** OS
 - 4: S : İndeksteki varlıklar kümesi
 - 5: N : İndeksteki toplam varlık sayısı
 - 6: OS : çaprazlama sonucu oluşan çocuk
 - 7: os_i : çocuğun i . elemanı için ağırlık, $i \in S$
 - 8: mp : mutasyon parametresi
 - 9: r_1 : düzgün rassal tamsayı $r_1 \in [0, N]$
 - 10: r_2 : ortalaması 0 ve varyansı 1 olan normal dağılımlı bir rassal sayı
 - 11: **Başla**
 - 12: $os_{r_1} = r_2 * \sqrt{mp} + os_{r_1}$
 - 13: **Bitir**
-

Şekil 13: Mutasyon Prosedürü

Çaprazlama işlemi yapabilmek için iki bireye ihtiyaç duyulmaktadır. Önerilen GA'da bu iki bireyin seçimi için Şekil 14'de prosedürü verilen rulet tekeri yöntemi (Horn vd. 1994) kullanılmaktadır. Rulet tekeri yöntemi popülasyondaki her kromozom için uygunluk değerine bağlı olarak bir ağırlık vermektedir. Seçilecek bireyler bu ağırlığa göre seçilir. Yani, uygunluk değeri ne kadar iyi ise ağırlığı o kadar fazla olur ve dolayısıyla seçilme ihtimali de o kadar artar.

-
- 1: **Prosedür:** Rulet tekeri
 - 2: **Girdi:** f, ps
 - 3: **Çıktı:** sm
 - 4: S : İndeksteki varlıklar kümesi
 - 5: N : indisteki toplam varlık sayısı
 - 6: ps : popülasyon büyüklüğü
 - 7: f_i : portföyün uygunluk değeri, $i \in S$
 - 8: sm : rulet tekeri yöntemi sonucunda seçilen çözümün indisi
 - 9: P_i : varlıkların uygunluk değerlerine göre ağırlıkları kümesi, $i \in S$
 - 10: C_k : uygunluk değeri için kümülatif toplamlar kümesi, $k \in S$
 - 11: r : düzgün bir rassal sayı, $r \in [0,1]$
 - 12: **Başla**
 - 13:
$$P_i = f_i / \sum_{i=1}^N f_i \quad i = 1, \dots, N$$
 - 14:
$$C_k = \sum_{i=1}^k f_i \quad i = 1, \dots, N \quad k = 1, \dots, N$$
 /* kümülatif toplam */
 - 15: **Eğer** $P_{i-1} \leq r \leq P_i$ ise $ps = i$
 - 16: **Bitir**
-

Şekil 14:Rulet tekeri

Önerilen GA'nın aşamaları Şekil 15'de gösterilmiştir. İlk aşama olarak rasgele bir başlangıç popülasyonu yaratılır ve uygunluk değerleri hesaplanır. Daha sonra sırasıyla seçim, çaprazlama ve mutasyon uygulanır. Mutasyondan sonra kromozomun uygun çözüm haline gelmesi için tamir prosedürü kullanılır. Oluşan çocuk popülasyonun en kötü uygunluk değerine

sahip bireyle deđiřtirilir. Bu adımlar belirlenen iterasyon sayısı kadar yapılır ve bir sonraki λ deđeri için aynı işlemler tekrar edilir. Önerilen GA'da $E = 51$ olarak sabitlenmiştir. Böylelikle λ deđeri 0 ile 1 arasında 0.02 aralıklarla artar ve etkin sınır üzerinde 51 nokta oluşturur. Her λ deđeri için iterasyonlar bittikten sonra popülasyondaki en iyi çözüm risk-getiri grafiđinde işaretleir ve etkin sınır oluşturulur.

```

1: Algoritma: Genetik Algoritma
2: Girdi: Veri, parametre ve yapılar
3: Çıktı:  $H$ 
4:  $S$ : İndeksteki varlıklar kümesi
5:  $N$ : İndeksteki toplam varlık sayısı
6:  $ps$ : Popülasyon büyüklüğü
7:  $W$ : Portföyün ağırlıkları kümesi
8:  $POP$ : popülasyon
9:  $E$ : tanımlanan  $\lambda$  sayısı
10:  $IT$ : iterasyon sayısı
11:  $H$ : Pareto optimal çözümler
12:  $P_1, P_2$ : çaprazlamaya girecek ebeveynler
13:  $OS$ : çaprazlama sonucu oluşan çocuk
14: Başla
15:  $H = \emptyset$ 
16:  $e = 1$ 
17: Tekrar Et
18:    $\lambda = (e - 1)/(E - 1)$ 
19:    $W_i = \varepsilon + r(1 - \varepsilon), \quad i = 1, \dots, ps \quad \forall W_i \in POP$ 
20:    $POP = \{W_1, \dots, W_i, \dots, W_{ps}\}$  /*rasgele bir başlangıç popülasyonu yarat*/
21:    $iterasyon = 1$ 
22:   Tekrar Et
23:      $\{P_1, P_2\} \leftarrow Rulet\ Teker\ (POP)$ 
24:      $OS \leftarrow \text{Çaprazlama}(S, N, P_1, P_2)$ 
25:      $OS \leftarrow \text{Mutasyon}(OS)$ 
26:      $OS \leftarrow \text{Tamir}(OS)$ 
27:      $\{R^*, V^*, f\} \leftarrow \text{Uygunluk\ hesapla}(OS)$ 
28:      $POP_{en\ kötü} \leftarrow OS$  /*oluşan çocuđu popülasyonun en küçük uygunluk deđerine sahip elemanıyla deđiřtir*/
29:      $iterasyon = iterasyon + 1$ 
30:    $iterasyon = IT$  oluncaya kadar
31:    $H_e \leftarrow POP_{en\ iyi}, \quad e = 1, \dots, E$ 
32:    $e = e + 1$ 
33:  $e = E$  oluncaya kadar
34: Bitir

```

Şekil 15: PO problemi için önerilen Genetik Algoritma

3.3.3. Parçacık Sürü Optimizasyonu (PSO)

İlk defa [Kennedy ve Eberhart \(1995\)](#) tarafından ortaya atılan parçacık sürüsü optimizasyonu (PSO) tekniđi, sürü halinde hareket eden kuş ve balıkların yiyecek bulmak için birbirleri arasındaki bilgi paylaşımını modelleyen, sürü zekâsı konseptini temel alan bir metasezgiseldir. Bilim insanları kuşların ve balıkların yiyeceđe ulaşmak için gösterdikleri sosyal davranışları gözlemlemiş ve simüle etmişlerdir. Bilim insanlarının bulgularına göre, yiyecek ararken her parçacık hem kendi tecrübesine, hem de sürüdeki diđer parçacıkların tecrübelerini

değerlendirerek konumunu ve hızını ayarlar. PSO tekniğinde, bir parçacık olarak adlandırılan her kuş ya da her balık, çok boyutlu çözüm uzayında bir pozisyondan başka bir pozisyona bir hızla hareket ederek optimal çözüme yakınsamaya çalışır. Çözüm uzayındaki pozisyonu üzerinde çalışılan problemin bir çözümüne karşılık gelir. PSO terminolojisinde, her iterasyondaki uygun çözümler topluluğuna GA'daki popülasyona eşdeğer olan sürü adı verilir. GA'da olduğu gibi, PSO'da da parçacıkların her iterasyonda hesaplanan bir hız ile pozisyonunu belirlediği popülasyon temelli bir arama prosedürüdür. Bu yüzden, hız optimal uzaklığa ulaşmak için pozisyonu ayarlama önemli bir rol oynar. GA'lardan farklı olarak PSO'da popülasyon filtrelenerek evrimleşmek yerine korunarak ilerler. Genel olarak, her parçacığın davranışı grup hafızası ile bireysel hafızanın bir uzlaşmaya varmasıdır denebilir. PSO'nun en temel adımları Şekil 16'da verilmiştir.

Adım	Açıklama
1	Başlangıç popülasyonunu rasgele yarat.
2	Popülasyondaki her bireyin uygunluk değerini hesapla.
3	Her birey için bireysel en iyi pozisyonu belirle
4	Popülasyondaki en iyi bireyin global en iyi pozisyonunu belirle.
5	Her bireyin hızını güncelle.
6	Her bireyin pozisyonunu güncelle.
7	Durdurma ölçütü sağlanmadıysa adım 2'ye git.
8	Algoritmayı bitir ve sonuçları raporla.

Şekil 16: Parçacık Sürü Optimizasyonu algoritmasının ana adımları

Bu projede kodlanan PSO algoritması Deng vd. (2012) çalışmasından esinlenerek hazırlanmıştır. Bu algoritma da diğer algoritmalar gibi ortak prosedürleri kullanmaktadır. Geliştirilen PSO algoritması Şekil 17'de verilmiştir.

Bir sonraki bölümde proje kapsamında portföy optimizasyonu problemine uyarlanan yapay arı kolonisi (YAK) algoritması anlatılmıştır.

1: **Algoritma:** Parçacık Sürü Optimizasyonu
2: **Girdi:** Veri (R_i, VC_{ij}) ve parametreler
 $(R_i, VC_{ij}, K, \varepsilon, E, ps, c_1^{min}, c_1^{maks}, c_2^{min}, c_2^{maks}, Iw_{min}, Iw_{maks}, IT)$
3: **Çıktı:** H
4: S : İndeksteki varlıklar kümesi
5: N : İndeksteki toplam varlık sayısı
6: ps : Popülasyon büyüklüğü
7: W : Portföyün ağırlıkları kümesi
8: POP : Popülasyon matrisi
9: VEL : Parçacık hızları matrisi
10: E : Tanımlanan λ sayısı
11: IT : Maksimum iterasyon sayısı
12: H : Pareto optimal çözümler
13: **Başla**
14: $H = \emptyset$
15: $e = 1$
16: **Tekrar Et**
17: $\lambda = (e - 1)/(E - 1)$
18: $W_i = \varepsilon + r(1 - \varepsilon), \quad i = 1, \dots, ps \quad \forall W_i \in POP$
19: $W_i^{rep} \leftarrow Tamir(N, K, S, W_i) \quad i = 1, \dots, ps \quad \forall W_i \in POP$
20: $POP = \{W_1^{rep}, \dots, W_i^{rep}, \dots, W_{ps}^{rep}\}$ /*rasgele bir başlangıç popülasyonu oluştur*/
21: $PBEST \leftarrow POP$ /*Popülasyondaki parçacıkların pozisyonlarını kaydet*/
22: $PBEST \leftarrow POP_{en.iyi}$ /*Popülasyondaki parçacıkların arasındaki en iyi pozisyonu kaydet*/
23: $iterasyon = 1$
24: **Tekrar Et**
25: $c_1 = (c_1^{min} - c_1^{maks})iterasyon/IT + c_1^{maks}$
26: $c_2 = (c_2^{maks} - c_2^{min})iterasyon/IT + c_2^{min}$
27: $Iw_{iterasyon} = (Iw_{maks} - Iw_{min})(IT - iterasyon)/IT + Iw_{maks}$
28: $VEL_{sj} = VEL_{sj}Iw_{iterasyon} + c_1r(PBEST_{sj} - POP_{sj}) + c_2r(GBEST_{sj} - POP_{sj}),$
 $s = 1, \dots, ps \quad j = 1, \dots, N$
29: $POP_{sj} = POP_{sj} + VEL_{sj} \quad s = 1, \dots, ps \quad j = 1, \dots, N$
30: $POP_s \leftarrow Tamir(N, K, S, POP_s)$
31: $\{R^*, V^*, f_s\} \leftarrow Uygunluk\ hesapla(POP_s)$
32: **Eğer** $f_s < f_{PBEST_s}$ **ise**
33: $PBEST \leftarrow POP_s$
34: **Eğer** $f_{POP_{en.iyi}} < f_{GBEST}$ **ise**
35: $GBEST = f_{POP_{en.iyi}}$
36: $iterasyon = iterasyon + 1$
37: $iterasyon = IT$ **oluncaya kadar**
38: $H_e \leftarrow GBEST, \quad e = 1, \dots, E$
39: $e = e + 1$
40: $e = E$ **oluncaya kadar**
41: **Bitir**

Şekil 17: Parçacık Sürü Optimizasyonu Algoritması

3.4. Yapay Arı Kolonisi (YAK) tekniğinin probleme uyarlanması

Projenin 1. gelişme raporunda verilen Yapay Arı Kolonisi (YAK) algoritması her ne kadar kabul edilebilir çözümler üretebilse de performansı literatüre kıyasla daha kötü bir performans sergilemekteydi. Bu projenin ana amacı olan en etkin algoritma tasarımının gerçekleştirilmesi doğrultusunda proje ekibi, YAK ve diğer algoritmaların düşük performansının nedenlerini araştırmış ve en temel sıkıntının literatürde yaygın olarak kullanılan tamir fonksiyonundan

(Şekil 8) kaynaklandığını tespit etmiştir. Bu zayıf tamir prosedürüne alternatif olarak ikinci bir tamir prosedürü Bölüm 3.4.1’de verilmiştir. Bu yeni prosedür, uyumsuz çözüm kontrolü (Bölüm 3.4.2) ve uyumsuzluğa izin veren seçim (Bölüm 3.4.3) prosedürlerini kullanmaktadır. Bu yeni prosedürleri içeren YAK-II algoritması da Bölüm 3.4.4’de verilmiştir.

Bir algoritma tarafından oluşturulan bir çözümün uyumluluğunun (feasibility) garanti altına alınması gerekmektedir. Şekil 8’de verilen tamir fonksiyonu ilk defa Chang vd. (2000) tarafından literatüre kazandırılmış ve daha sonra birçok çalışmada (Anagnostopoulos ve Mamanis 2010, 2011a; Khalidji vd. 2009; Li 2009; Lwin vd. 2014; Mashayekhi ve Omrani 2016; Mishra vd. 2014b; Moral-Escudero vd. 2006) oluşturulan çözümü uyumlu hale getirmek için kullanılmıştır. Bu prosedür ile öncelikle yatırım yapılan toplam hisse senedi sayısını K ’ya eşitlenmekte, daha sonra ise, yatırım yapılan hisselerin minimum ağırlıkları ε değerinin üzerine çıkartılarak, ilk başta uygunsuz olan bir çözüm, hem yatırım yapılan toplam hisse sayısı kısıtını hem de minimum ağırlık kısıtını sağlayarak uygun bir çözüm haline getirilmektedir.

Şekil 8’de verilen bu tamir fonksiyonuna giren bir çözümdeki bir hisse ağırlığı, alt sınır olan ε değerinden daha küçük ya da eşit olmasına rağmen, tamir fonksiyonu bu değerleri ε ’dan yukarıya öteleyip, ağırlığının fazla olmaması gereken bir hisse ağırlığını artırmakta ve bu da çözümü kötüleştirmektedir. Bir başka deyişle, bu prosedür algoritmaya özgürce çalışma olanağı tanımamaktadır. Bu tamir prosedürünü Denklem (32), denklem (33) ve denklem (34) ile formülize edilmektedir.

$$L = \sum_{i \in Q} w_i \quad L : \text{Portföyde bulunan hisselerin ağırlıklar toplamı} \quad (32)$$

$$F = 1 - \sum_{i \in Q} \varepsilon_i \quad F : \text{Portföydeki serbest oran} \quad (33)$$

$$w_i = \varepsilon_i + w_i F/L \quad \forall i \in Q \quad Q : \text{portföyde bulunan hisseler kümesi} \quad (34)$$

Denklem (34)’de görüldüğü gibi, her ağırlık alt sınır değeri (ε) ve $s_i F/L$ değeri ile toplanmaktadır. Bu durumun neden olduğu sıkıntıları basit bir örnekle gösterebiliriz.

$w_0 = \{0.085, 0, 0.65, 0.06, 0\}$ tamir fonksiyonuna girecek olan ağırlıklar kümesi olsun. Bu örneğimiz için alt sınır değeri $\varepsilon = 0.1$ ve portföyde bulunması gereken minimum hisse sayısı $k = 3$ olsun. w_0 çözümü bu tamir fonksiyonu ile tamir edildikten sonra $w_1 = \{0.1748, 0, 0.6723, 0.1528, 0\}$ çözümü elde edilir. Tamire girmeden önce en yüksek ağırlığa sahip üçüncü hisse ağırlığının tamirden sonraki değişim oranı en azdır. Bu örneğimiz için istenilen w_1 çözümü $\{0.1, 0, 0.8, 0.1, 0\}$ gibi olabilir. Fakat Chang vd. (2000) tarafından önerilen tamir fonksiyonu buna imkân vermemektedir.

Portföy optimizasyonu probleminde bazı durumlarda da (özellikle de $\lambda = 0$) sadece bir hisseye yatırım yapmak ve $k - 1$ tane hisseye ε ağırlığını vermek amaç fonksiyonunu maksimum etmektedir. Bu tamir prosedürü elde edilen çözümü bozmakta ve istenilen çözüme ulaşılmasını zorlaştırmaktadır.

3.4.1. Tamir Prosedürü – II

Birinci tamir prosedüründeki eksiklikten hareketle, yeni bir prosedür arayışına girilmiştir. Deb (2000) bir algoritmanın uyumsuz çözüm uzayında çalışmasına imkân veren bir metot önermiştir. Bu metot, uyumsuz çözümlere penaltı değerleri vererek zamanla uyumsuz çözümlerden kurtulmayı hedeflemektedir. Tüm ağırlıkların bire eşit olması kısıtı, portföyde tam olarak K adet hisse bulunması kısıtı ve bir hissenin alabileceği minimum ε kısıtı, ele alınan portföy optimizasyonu probleminin olmazsa olmazlarıdır. Deb (2000) tarafından önerilen kısıt yönetme metodunun bizim problemimize uyarlanması için birçok deneme yapılmıştır. Yapılan ön çalışmalar göstermiştir ki, tüm ağırlıkların bire eşit olması kısıtı ile portföyde tam olarak K adet hisse bulunması kısıtı serbest bırakıldığı takdirde algoritmaların uyumlu çözüm bulamadığı gözlemlenmiştir. Bu nedenle, yeni tamir fonksiyonunda, portföyde K adet hisse bulundurma kısıtı ve ağırlıklar toplamının bire eşit olma kısıtı sağlanırken, minimum ağırlık değeri kısıtı serbest bırakılmıştır. Bu yöntemle, Şekil 8’de verilen tamir prosedürünün oluşturduğu sıkıntının azaltılmasını hedefleyen ikinci bir tamir prosedürü (Şekil 18) oluşturulmuştur. Bu prosedür, uyumsuz bir çözüme izin verebildiği için bir uyumsuz çözüm kontrolü prosedürüne ihtiyaç duyulmaktadır.

-
- 1: **Prosedür:** Tamir-II
 - 2: **Girdi:** N, K, S, W
 - 3: **Çıktı:** W^r
 - 4: N : İndeksteki toplam varlık sayısı
 - 5: K : Portföyde bulunmasına izin verilen toplam varlık sayısı
 - 6: S : İndeksteki varlıklar kümesi
 - 7: W : Tamire giren portföyün ağırlıkları kümesi
 - 8: w_i : Tamire giren i . varlığın ağırlığı $i \in S$
 - 9: w_i^r : Tamirden çıkan i . varlığın ağırlığı $i \in S$
 - 10: W^r : Tamire giren portföyün ağırlıkları kümesi
 - 11: z_i : i . varlığın portföyde yer alıp almama durumunu gösteren ikili değişken, $i \in S$
 - 12: **Başla**
 - 13: **Tekrar et**
 - 14: **Eğer** $\sum_{i=1}^N z_i > K$
 - 15: Portföyde bulunan en küçük w_i değerine sahip varlık için $z_i = 0$, $w_i = 0$ yap, $i \in S$
 - 16: **Eğer** $\sum_{i=1}^N z_i < K$
 - 17: Portföyde bulunmayan rasgele bir z_i değeri için $z_i = 1$ yap, $i \in S$
 - 18: $\sum_{i=1}^N z_i = K$ olana kadar
 - 19: $CS = \sum_{i=1}^N w_i$ $i \in S$, /* portföydeki varlıkların ağırlıkları toplamı */
 - 21: $w_i^r = w_i * z_i / CS$ $\forall i \in W, \forall i \in W^r$
 - 22: **Bitir**
-

Şekil 18: Tamir-II prosedürü

3.4.2. Uyumsuz Çözüm Kontrol Prosedürü

Tamir-II prosedürü sadece portföyde k adet varlık bulundurma ve portföydeki varlıkların toplamının bire eşit olma kısıtı sağlandığı için portföyde alt sınır olan ε 'dan daha küçük ağırlıklara sahip varlıklar olabilir. Bu da çözümü uyumsuz yapmaktadır. Kullandığımız seçim prosedüründe, çözümler kıyaslanırken iki uyumsuz çözüm arasında daha az uyumsuz olan

tercih edilmektedir. Bu nedenle, çözümlerin uyumsuz olup olmadıkları ve uyumsuzlar ise ne kadar uyumsuz olduklarının bilinmesi gerekir. Geliştirdiğimiz algoritmada bu işlem için uyumsuz çözüm kontrol prosedürü (Şekil 19) kullanılmıştır. Bu prosedür, uyumsuz çözüme neden olan bir hisse eğer alt sınırın altında ise o çözümün uygunluk değerine bir ceza değeri eklemektedir. Bu prosedürü kullanarak uyumsuzluğa izin veren seçim prosedürü Bölüm 3.4.3'de verilmiştir.

-
- 1: **Prosedür:** Uyumsuz Çözüm Kontrol
 - 2: **Girdi:** n, k, ε, w
 - 3: **Çıktı:** w
 - 4: N : İndeksteki toplam varlık sayısı
 - 5: K : Portföyde bulunmasına izin verilen toplam varlık sayısı
 - 6: S : İndeksteki varlıklar kümesi
 - 7: ε : bir varlığın alabileceği minimum ağırlık
 - 8: W : Tamire giren portföydeki toplam varlık sayısı
 - 9: w_i : Tamire giren i . varlığın ağırlığı $i \in S$
 - 10: vio : çözümün ne kadar uygun olmadığını gösteren değişken
 - 11: u : çözümün uygulanabilir olup olmadığını gösteren ikili değişken
 - 12: **Başla**
 - 13: $vio = |W - K|$
 - 14: **Tekrar et**
 - 15: **Eğer** $w_i = 0$ **ise Devam et**
 - 16: **Değilse eğer** $w_i < \varepsilon$
 - 17: $vio = vio + 1$
 - 18: **Son**
 - 19: $i = n$ **olana kadar**
 - 20: **Eğer** $vio = 0$ **ise** $u = 0$ /*uyumlu (feasible) bir çözüm değil*/
 - 21: **Değilse** $u = 1$
 - 22: **Bitir**
-

Şekil 19: Uyumsuz çözüm kontrol prosedürü

3.4.3. Uyumsuzluğa İzin Veren Seçim Prosedürü

Çözüm uzayında çözüm aranırken uyumsuz çözümlere izin veren bir prosedür ile iterasyonlar boyunca uyumsuz bir çözümden iyi uygunluk değerlerine sahip uyumlu çözümler elde edilebilmektedir. Bu prosedür Şekil 20'de gösterilmiştir. Her iki çözümün de uyumlu olup olmamasına göre hareket eden bu prosedür ile uyumsuz çözümlere geçici olarak izin verilmektedir.

```

1: Prosedür: Uyumsuzluğa İzin Veren Seçim
2: Girdi:  $w_1, w_2, N$ 
3: Çıktı:  $w_i, gelişme$ 
4:  $w_1$ : güncel çözüm
5:  $w_2$ : güncel çözüm ile kıyaslanan çözüm
6:  $u(w_i)$ :  $i$ . çözümün kısıtları sağlayıp sağlamadığını gösteren ikili değişken
7:  $f(w_i)$ :  $i$ . çözümün uygunluk değeri
8:  $f(v_i)$ :  $i$ . çözümün ne kadar uygun olmadığını gösteren değişken
9: Başla
10:  $gelişme = 0$ 
11: Eğer  $u(w_2) = 1$  ve  $u(w_1) = 1$  ise /*her iki çözüm de uyumlu ise*/
12:   Eğer  $f(w_2) < f(w_1)$  ise
13:      $w_1 = w_2, gelişme = 1$ 
14:   Son
15: Değilse Eğer  $u(w_2) = 1$  ve  $u(w_1) = 0$  ise /*kıyaslanan çözüm uyumlu bir çözüm değil ise*/
16:    $w_1 = w_2, gelişme = 1$ 
17: Değilse Eğer  $u(w_2) = 0$  ve  $u(w_1) = 0$  ise /*her iki çözüm de uyumlu değil ise*/
18:   Eğer  $f(v_2) < f(v_1)$  ise
19:      $w_1 = w_2, gelişme = 1$ 
20:   Son
21: Son
22: Bitir

```

Şekil 20: Uyumsuzluğa izin veren seçim prosedürü

3.4.4. İyileştirilmiş YAK Algoritması (YAK-II)

Birinci gelişme raporunda verilen YAK algoritması yukarıdaki bölümlerde verilen alt prosedürler eklenerek güncellenmiş ve YAK-II algoritması oluşturulmuştur. İşçi arı ve gözcü arı aşamalarında üretilen yeni çözümler eski çözümlerle uyumsuzluğa izin veren seçim prosedüründe karşılaştırılmış ve prosedürden çıkan çözüm popülasyona eklenmiştir. Geliştirilen YAK-II algoritması adımları [Şekil 21](#)'de verilmiştir.

```

1: Algoritma: YAK-II
2: Girdi: Veri  $(R_i, VC_{ij})$  ve parametreler  $(\varepsilon, K, E, ps, klimit, IT)$ 
3: Çıktı:  $H$ 
4:  $n$ : İndeksteeki toplam varlık sayısı
5:  $ps$ : Popülasyon büyüklüğü
6:  $POP$ : popülasyon
7:  $E$ : tanımlanan  $\lambda$  sayısı
8:  $T$ : maksimum iterasyon sayısı
9:  $H$ : son popülasyon
10:  $EB$ : işçi arı safhasından üretilen ağırlıklar kümesi
11:  $OB$ : gözcü arı safhasından üretilen ağırlıklar kümesi
12:  $f$ : portföyün uygunluk değeri
13:  $limit_s$ : popülasyondaki çözümlerin limit değerleri,  $s = 1, \dots, ps$ 
14:  $r$ : düzgün bir rassal sayı,  $r \in [-1, 1]$ 
15:  $rk$ : düzgün rassal bir tamsayı,  $rk \in [1, ps]$ 
16:  $klimit$ : kaşif arıların devreye girmesi için aşılması gereken limit değeri
17: Başla
18:  $H = \emptyset$ 
19:  $e = 1$ 
20: Tekrar Et
21:  $\lambda = (e - 1)/(E - 1)$ 
22:  $W_i = \varepsilon + r(1 - \varepsilon)$ ,  $i = 1, \dots, ps \quad \forall W_i \in POP$ 
23:  $W_i^{rep} \leftarrow Tamir(N, K, S, W_i, z_i)$   $i = 1, \dots, ps \quad \forall W_i \in POP$ 
24:  $EB \leftarrow Uygunluk\ hesapla(EB, N, VC_{ij}, R_i, \lambda)$ 
25:  $POP = \{W_1^{rep}, \dots, W_i^{rep}, \dots, W_{ps}^{rep}\}$ 
26: /*işçi arı aşaması*/
27:  $s = 1$ 
28: Tekrar Et
29:  $k = rk, rk \in [1, ps], rk \neq s$ 
30:  $EB_i \leftarrow POP_{si} + rPOP_{ki} - POP_{ki}$  /*  $i = 1, \dots, N, s = 1, \dots, ps$  */
31:  $EB \leftarrow TamirII(N, K, S, EB)$ 
32:  $\{R^*, V^*, f_{EB}\} \leftarrow Uygunluk\ hesapla(EB, N, VC_{ij}, R_i, \lambda)$ 
33:  $EB \leftarrow Uygulanabilirlik\ kontrol(n, k, \varepsilon, EB)$ 
34:  $EB$  /*seçim prosedüründe iki çözüm karşılaştırılıp daha iyi çözüm EB çözümüyle değiştiriliyor ve gelişme olup olmadığı ikili değişkene atılıyor*/
35:  $POP_s \leftarrow EB$ 
36: Eğer gelişme = 1 ise
37:  $limit_s = 0$ 
38: Aksi halde  $limit_s = limit_s + 1$ 
39:  $s = s + 1$ 
40:  $s = ps$  oluncaya kadar
41: /*gözcü arı aşaması*/
42:  $s = 1$ 
43: Tekrar Et
44:  $l \leftarrow rulet\ tekeri(N, POP)$ 
45:  $OB = POP_l$ 
46:  $k = rk, rk \in [1, ps], rk \neq l$ 
47:  $OB_i = POP_{li} + r * (POP_{li} - POP_{ki})$ ,  $i = 1, \dots, N$ 
48:  $OB \leftarrow TamirII(N, K, S, OB)$ 
49:  $\{R^*, V^*, f_{OB}\} \leftarrow Uygunluk\ hesapla(OB, N, VC_{ij}, R_i, \lambda)$ 
50:  $OB \leftarrow Uyumsuzluk\ kontrol(n, k, \varepsilon, OB)$ 
51:  $OB$  /*seçim prosedüründe iki çözüm karşılaştırılıp daha iyi çözüm OB çözümüyle değiştiriliyor ve gelişme olup olmadığı ikili değişkene atılıyor*/
52:  $POP_s \leftarrow OB$ 
53: Eğer gelişme = 1 ise
54:  $limit_s = 0$ 
55: Aksi halde  $limit_s = limit_s + 1$ 
56:  $s = s + 1$ 
57:  $s = ps/2$  oluncaya kadar
58: /*kaşif arı aşaması*/
59:  $s = 1$ 
60: Tekrar Et
61: Eğer  $limit_s > klimit$  ise
62:  $W_i = \varepsilon + r(1 - \varepsilon)$ ,  $i = 1, \dots, ps \quad \forall W_i \in POP$ 
63:  $SB \leftarrow TamirII(N, K, S, W_i)$ 
64:  $\{R^*, V^*, f_{SB}\} \leftarrow Uygunluk\ hesapla(SB, N, VC_{ij}, R_i, \lambda)$ 
65:  $POP_s \leftarrow SB$ 
66:  $limit_s = 0$ 
67:  $s = ps$  oluncaya kadar
68: /*popülasyonun en iyi çözümünü pareto optimal çözümler kümesine atılır*/
69:  $H_e = POP_{en\ iyi}$ 
70:  $e = E$  oluncaya kadar
71: Bitir

```

Şekil 21: YAK-II algoritması

3.5. Karınca Kolonisi Optimizasyonu (KKO) tekniğinin probleme uyarlanması

Karınca kolonisi optimizasyonu (KKO) tekniği ilk defa [Dorigo ve Gambardella \(1997\)](#) tarafından gezgin satıcı problemini çözmek için önerilmiştir. Bu teknik yapısı gereği çeşitli kesikli problemlerin çözümü için yaygın olarak uygulanmıştır. Ancak, yapılan literatür araştırması KKO tekniğinin sürekli yapıdaki problemlere çok az sayıda uygulanmış olduğu göstermiştir. Daha önce bazı denemeler yapılmış olsa da, ilk defa [Socha ve Dorigo \(2008\)](#) tarafından iyi bir performans ortaya koyan sürekli fonksiyonların çözümü için bir KKO algoritması önerilmiştir. Daha sonra, [Liao vd. \(2014\)](#) tarafından bu algoritma geliştirilmiştir. KKO algoritması ana adımları Şekil 22'de verilmiştir.

Karınca Kolonisi Optimizasyonu

Tekrar et

Karınca Esaslı Çözüm yapısı oluştur

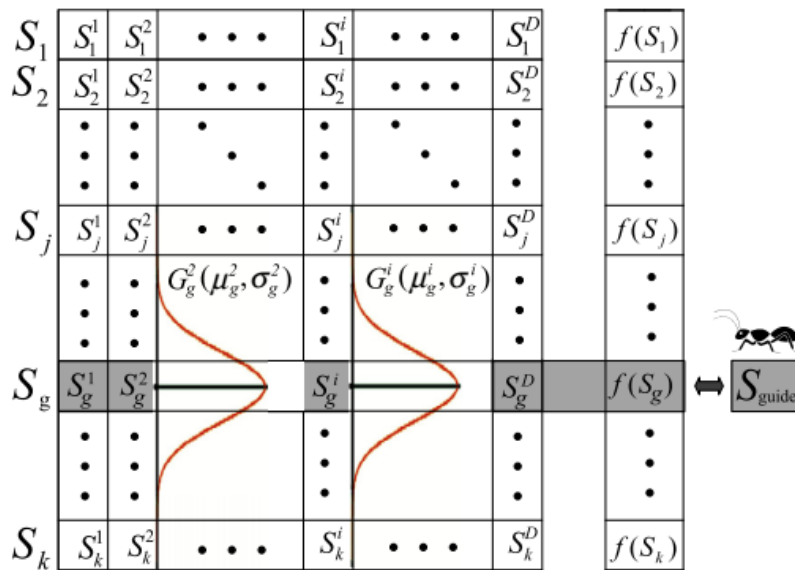
Feromon güncelle

Lokal arama (opsiyonlu)

Sonlandırma kriteri sağlanıncaya kadar

Şekil 22: KKO algoritması ana adımları ([Socha ve Dorigo 2008](#))

Sürekli alanda çalışan KKO algoritmasında feromonlar, kesikli problem tiplerinde kenarlar üzerinde biriktirilmesi yerine, bir çözüm arşivi olarak temsil edilmektedir. Şekil 23'de bir çözüm arşivinin yapısı verilmiştir. Bu algoritmada, düzgün dağılıma sahip rassal olarak oluşturulan bir arşivdeki çözümlerin ortalama ve standart sapma değerlerini kullanarak dinamik olarak olasılık yoğunluk fonksiyonları (özellikle Gauss dağılımı) ile yeni çözümler üretilmektedir. Bu çözüm arşivi geçmiş arama bilgisini içeren bir hafıza yapısı olarak görülebilir.



Şekil 23: KKO algoritmasında çözüm arşivinin yapısı ([Liao vd. 2014](#))

Bu sözü edilen çalışmaları (Liao vd. 2014; Socha ve Dorigo 2008) esas alan portföy optimizasyonu probleminin çözümüne yönelik bir KKO algoritması geliştirilmiştir. KKO algoritmasında Şekil 23'de temsili olarak verilen arşivi düzgün dağılıma sahip rasgele ve uyumlu çözümlerden oluşacak şekilde başlatılır (Şekil 24 – Adım 25) ve artan uygunluk değerlerine göre sıralanır. Daha sonra, seçim olasılığına göre rehber karınca olarak arşivdeki en iyi çözüm seçilir ya da rulet tekeri yöntemi kullanılarak seçilir (Adım 28, Adım 29). Adım 31'de eğer rasgele sayı, değiştirme oranından (mr) düşük ise, rehber çözüm baz alınarak o problem boyutu için yeni bir değer oluşturulur. Seçilen çözümün her boyutu için, o boyuttaki değeri ortalaması (Adım 33) ve arşivdeki diğer çözümlere olan ortalama uzaklığını standart sapması (Adım 32) olan bir Gauss dağılımına sahip rassal değişken yaratılır (Adım 34). Oluşturulan yeni arşiv, tamir (Adım 39), uyumsuz çözüm kontrolü (Adım 40) ve uygunluk hesaplama prosedürleri (Adım 41) uygulanır. Daha sonra arşivdeki çözümler bir lokal arama prosedürleri ile iyileştirilir (Adım 42). Bu aşamada kullanılan loka arama prosedürü Bölüm 3.6.1.2.2'de anlatılmıştır. Oluşturulan yeni arşiv (NS) karınca sayısı kadar çözüm içerir, bu çözümler ana arşivdeki elit çözümler ile birleştirilir (Adım 45). Bu işlemden sonra ana arşivin boyutunun değişmemesi için, belirlenen elit çözüm sayısı, arşiv büyüklüğü ve karınca sayısının farkına eşittir. Arşivdeki en iyi çözüm o ana kadar elde edilen en iyi (GB) çözümden daha iyi ise, GB güncellenir ve *limit* sayacı sıfırlanır (Adım 46). Eğer *limit* sayacı, belirlenen *ilimit* değerini aşmış ise arşiv kötü çözümlerden başlayarak rR oranında yenilenir (Adım 47). Bu adımda arşivin ilk $(1 - rR)$ oranındaki çözümler (iyi çözümler) korunur, geri kalan çözümlerin yerine ise rasgele çözümler yaratılır. İterasyon limiti aşıncaya kadar bu işlemler tekrarlanır. Bütün λ değerleri için maksimum iterasyon sayısı kadar KKO algoritması çalıştırılır ve her λ değeri için elde edilen en iyi çözüm pareto optimal çözümler kümesine atılır (Adım 50).

```

1: Algoritma: KKO
2: Girdi: Veri  $(R, VC)$  ve parametreler  $(n, \varepsilon, k, E, as, ks, T, qBest, mr, \xi)$ 
3: Çıktı:  $H$ 
4:  $S$ : arşivdeki çözümler kümesi
4:  $GB$ : o ana kadar elde edilen en iyi çözüm
5:  $n$ : indeksteki toplam varlık sayısı
6:  $as$ : arşiv büyüklüğü
7:  $AR$ : arşiv
8:  $E$ : tanımlanan  $\lambda$  sayısı
9:  $T$ : maksimum iterasyon sayısı
10:  $H$ : pareto optimal çözümler kümesi
11:  $\xi$ : feromon uçurma parametresi
12:  $ks$ : karınca sayısı
13:  $NS$ : yeni çözüm arşivi
14:  $f$ : portföyün uygunluk değeri
15:  $qBest$ : arşivdeki en iyi çözümün seçilme olasılığı
16:  $mr$ : seçilen çözümde değiştirilecek problem boyutu oranı
17:  $ilimit$ : arşivin yenilenmesi için aşılması gereken değer
18:  $rR$ : arşivin hangi oranda yenileneceğini belirleyen parametre
19:  $r$ :  $[0,1]$  aralığında düzgün dağılıma sahip rasgele sayı
20: Başla
21:  $H = \emptyset$ 
22:  $e = 1$ 
23: Tekrar Et
24:  $\lambda = (e - 1)/(E - 1)$ 
25: Başlangıç arşivi  $AR = \{S_1, \dots, S_{as}\}$ 
26: Tekrar Et
27: Tekrar Et
28: Eğer  $r < qBest$  ise  $rehber \leftarrow AR_{en\ iyi}$ 
29: Aksi Halde  $rehber \leftarrow rulet\_tekeri(AR)$ 
30: Tekrar Et
31: Eğer  $r < mr$  ise
32:  $\sigma_j^i = \xi \sum_{r=1}^{as} \frac{|S_r^i - S_{rehber}^i|}{as - 1}$ 
33:  $\mu_j^i = S_{rehber}^i$ 
34:  $G^i(x) = \sum_{l=1}^k \omega_l \frac{1}{\sigma_l^i \sqrt{2\pi}} e^{-\frac{(x-\mu_l^i)^2}{2\sigma_l^i}}$ 
35:  $NS_j^i = G^i(x)$ 
36: Aksi Halde  $NS_j^i = AR_{rehber}^i$ 
37:  $i = i + 1$ 
38:  $i = n$  oluncaya kadar
39:  $NS \leftarrow TamirII(NS, n, \varepsilon, k)$ 
40:  $NS \leftarrow Uyumsuz\ çözüm\ kontrolü(NS, n, \varepsilon, k)$ 
41:  $NS \leftarrow Uygunluk\ hesapla(NS, R, VC, \lambda)$ 
42:  $NS \leftarrow Nelder\_mead(NS)$ 
43:  $limit = limit + 1$ 
44:  $j = j + 1$ 
45:  $j = ks$  oluncaya kadar
46:  $AR \leftarrow Arşiv\ güncelle(NS, AR)$ 
47: Eğer  $f(AR_1) < f(GB)$  ise  $GB = AR_1$  ve  $limit = 0$ 
48: Eğer  $limit > ilimit$  ise  $AR \leftarrow Arşivi\ yenile(AR, rR)$ 
49:  $iter = iter + 1$ 
50:  $iter = T$  oluncaya kadar
51:  $H_e = AR_{en\ iyi}$ 
52:  $e = e + 1$ 
53:  $e = E$  oluncaya kadar
54: Bitir

```

/* $ks = as * rR^*$ */
/* $NS = \{S_1, \dots, S_{ks}\}^*$ */

/*rasgele bir başlangıç arşivi yarat ve uygunluk değerlerin göre artan sırayla sırala*/

/* $qBest$ ihtimalle arşivdeki en iyi çözüm rehber karınca seçilir*/
/* aksi halde rehber karınca rulet tekeri yöntemi ile seçilir*/

/*Gaussian fonksiyonu*/

Şekil 24: KKO algoritması

3.6. Değişken Komşuluk Arama (DKA) tekniğinin probleme uyarlanması

Değişken Komşuluk Arama (DKA) tekniği literatürde ilk defa [Mladenović ve Hansen \(1997\)](#) tarafından komşuluk değiştirerek arama yapma fikriyle önerilmiştir. Gezgin satıcı probleminden başlayarak çeşitli kesikli problemlerin çözümü için yaygın olarak uygulanmıştır. Ancak, yapılan literatür araştırmasında DKA tekniğinin de sürekli yapıdaki problemlere çok az sayıda uygulanmış olduğu gözlemlenmiştir. [Mladenović vd. \(2008\)](#) sürekli alanda çözüm arayan genel bir DKA algoritması geliştirmiş, [Carrizosa vd. \(2012\)](#) ise bu metodu Gauss olasılık dağılımları ile besleyerek performansını geliştirmişlerdir.

İki temel aşamadan oluşan DKA algoritması, ilk aşamada çözüm uzayındaki keşfedilmemiş alanlara yönelmek için olasılık dağılım fonksiyonlarını kullanarak her boyutta rassal bir sarsım (shaking for exploration), ikinci aşamada ise ulaşılan alandaki yerel optimumlara ulaşmayı hedefleyen bir yerel arama (local search for exploitation) yapmaktadır. Bu nedenle, DKA algoritması sürekli düzlemde çalışırken olasılık dağılım fonksiyonlarının yanı sıra yerel arama prosedürlerine ihtiyaç duymaktadır.

Bu çalışmada kullanılan yerel arama prosedürleri ([Bölüm 3.6.1](#)) türev bilgisi kullanan ([Bölüm 3.6.1.1](#)) ve kullanmayan ([Bölüm 3.6.1.2](#)) olarak gruplanmış ve her bir yerel arama metodu ayrıntısıyla açıklanmıştır.

3.6.1 Yerel Arama Prosedürleri

DKA algoritmasında üç tanesi türev bilgisini kullanan prosedürler ([Bölüm 3.6.1.1](#)) ve iki tanesi türev bilgisini kullanmayan prosedürler ([Bölüm 3.6.1.2](#)) olmak üzere toplam beş tane yerel arama prosedürü kullanılmıştır.

3.6.1.1. Türev Bilgisi Kullanan Yerel Arama Prosedürleri

Türev bilgisi kullanan arama metotlarında, arama yönü fonksiyonun eğimine göre yapılandırılır. Türev bilgisi kullanılan metotların uygulanması zordur ([Venkataraman 2009](#)). Bu çalışmada, Steepest-Descent Algoritması ([Bölüm 3.6.1.1.1](#)) ve Fletcher-Reeves Algoritması ([Bölüm 3.6.1.1.2](#)) bu kapsamda yerel arama prosedürleri olarak çalışmaya eklenmiştir.

3.6.1.1.1. Steepest-Descent Algoritması

Steepest-Descent (SD) metodu türev tabanlı teknikler için doğal evrimleşmeyi sağlar ([Fox 1971](#)). Bir fonksiyonun bir noktadaki türevi, fonksiyonun o noktadaki artış yönünü gösterir. Dolayısıyla iniş yönü ters fonksiyon ile hesaplanabilir. İniş yönü belirlendikten sonra iniş vektörüyle fonksiyonun değeri iterasyonlar boyunca azaltılmaya çalışılır.

DKA algoritmasında kullanılan SD algoritması aramaya girecek çözümün türevinin alınmasıyla başlar ([Şekil 25, Adım 17](#)). Daha sonra çözümün türevi kullanılarak yeni çözüm oluşturulur ([Adım 18](#)) ve oluşturulan yeni çözüm ilk çözüm ile karşılaştırılır ([Adım 22](#)). Eğer yeni çözümün

uygunluk değeri daha iyi ise arama prosedürü içindeki en iyi çözüm olur. Aksi halde arama adımı büyüklüğü arama adımı küçültme parametresi ile çarpılır ve iterasyon artırılarak adımlara devam edilir. SD algoritması Şekil 25'de gösterilmiştir.

```

1: Prosedür: Steepest-Descent (SD)
2: Girdi:  $w_0, T, n, s$ 
3: Çıktı:  $w_1$ 
4:  $w_0$ : lokal aramaya girecek çözüm
5:  $f$ : çözümün uygunluk değeri
6:  $n$ : İndeksteki toplam varlık sayısı
7:  $s$ : iterasyon içerisindeki arama adımı büyüklüğü
8:  $\rho$ : arama adımı küçültme parametresi
9:  $S$ : belirlenen başlangıç arama adımı büyüklüğü
10:  $T$ : maksimum iterasyon sayısı
11:  $\nabla f$ :  $f$  fonksiyonunun birinci dereceden türevi
12: Başla
13:  $s = S$ 
14:  $w_{en\ iyi} = w_1$ 
15:  $iter = 0$ 
16: Tekrar Et
17:    $T_1 = \nabla f(w_1)$ 
18:    $w_1 = w_{en\ iyi} - s(T_1)$ 
19:    $w_1 \leftarrow TamirII(w_1)$ 
20:    $w_1 \leftarrow Uyumsuz\ çözüm\ kontrol(w_1)$ 
21:    $w_1 \leftarrow Uygunluk\ hesapla(w_1)$ 
22:   Eğer  $f(w_1) < f(w_{en\ iyi})$  ise  $w_{en\ iyi} = w_1$ 
23:   Değilse  $s = s * \rho$  Son
24:    $iter = iter + 1$ 
25:    $iter = T$  oluncaya kadar
26: Bitir

```

Şekil 25:Steepest-Descent algoritması

3.6.1.1.2. Fletcher-Reeves Algoritması

Fletcher-Reeves (FR) metodu (Fletcher ve Reeves 1964), SD metodunda küçük bir değişiklik yaparak çözüm performansı üzerinde etkili bir sonuç sağlamaktadır. Bu gelişmenin başarısı örüntü arama (pattern search) stratejisine dayanmaktadır. Bu metotta, ikinci dereceden yakınsama özelliği sayesinde, arama yönü ikinci dereceden türev matrisiyle n değişkenli bir kuadratik problemde n 'den daha az iterasyon ile yakınsama sağlanabilmektedir.

FR algoritmasında, SD metodunda olduğu gibi arama prosedürüne giren çözümün türevi kullanılarak ikinci bir çözüm elde edilir (Şekil 26, Adım-21) ve o ana kadarki en iyi çözümle karşılaştırılır. Daha sonra ikinci çözümün türevi kullanılarak üçüncü bir çözüm elde edilir ve üçüncü çözüm arama prosedürünün o ana kadarki en iyi çözümü ile karşılaştırılır (Adım 41). Eğer üçüncü çözüm daha iyi bir çözüm ise en iyi çözüm üçüncü çözüm olur, aksi halde arama adımı büyüklüğü arama adımı küçültme parametresi ile çarpılır ve iterasyon sayısı bir artırılır. Sonlandırma kriterlerinden birisi sağlanana kadar prosedür aramaya devam eder. FR algoritması Şekil 26'da gösterilmiştir.

```

1: Prosedür: Fletcher-Reeves
2: Girdi:  $w_0, T, n, s$ 
3: Çıktı:  $w_2$ 
4:  $w_0$ : lokal aramaya girecek çözüm
5:  $f$ : çözümün uygunluk değeri
6:  $n$ : İndeksteki toplam varlık sayısı
7:  $s$ : iterasyon içerisindeki arama adımı büyüklüğü
8:  $\rho$ : arama adımı küçültme parametresi
9:  $S$ : Belirlenen başlangıç arama adımı büyüklüğü
10:  $T$ : maksimum iterasyon sayısı
11:  $d_1, d_2, d_3$ : durdurma kriterleri
12:  $\nabla f$ :  $f$  fonksiyonunun birinci dereceden türevi
13: Başla
14:  $s = S$ 
15:  $w_{en\ iyi} = w_0$ 
16:  $w_1 = w_{en\ iyi}$ 
17:  $iter = 0$ 
18: Tekrar Et
19: Eğer  $iter = 1$  ise
20:  $T_1 = -\nabla f(w_1)$ 
21:  $w_2 = w_1 + sT_1$ 
22:  $w_2 \leftarrow TamirII(w_2)$ 
23:  $w_2 \leftarrow Uyumsuz\ çözüm\ kontrol(w_2)$ 
24:  $w_2 \leftarrow Uygunluk\ hesapla(w_2)$ 
25:  $T_2 = -\nabla f(w_2)$ 
26:  $iter = iter + 1$ 
27: Eğer  $f(w_2) < f(w_{en\ iyi})$  ise  $w_{en\ iyi} = w_2$  Son
28: Değilse
29:  $\beta = \frac{\nabla f(w_2) * \nabla f(w_2)^T}{\nabla f(w_1) * \nabla f(w_1)^T}$ 
30:  $T_3 = T_2 + \beta T_1$ 
31:  $w_3 = w_2 + sT_3$ 
32:  $w_3 \leftarrow TamirII(w_3)$ 
33:  $w_3 \leftarrow Uyumsuz\ çözüm\ kontrol(w_3)$ 
34:  $w_3 \leftarrow Uygunluk\ hesapla(w_3)$ 
35: Eğer  $f(w_3) < f(w_{en\ iyi})$  ise  $w_{en\ iyi} = w_3$ 
36: Değilse  $s = s * \rho$  Son
37:  $T_3 = -\nabla f(w_3)$ 
38:  $\Delta f = f(w_3) - f(w_2)$ 
39: Eğer  $|\Delta f| \leq d_1$  ise Bitir
40:  $\Delta w = w_3 - w_2$ 
41: Eğer  $\Delta w \Delta w^T \leq d_2$  ise Bitir
42: Eğer  $\nabla f(w_3) \nabla f(w_3)^T \leq d_3$  ise Bitir
43:  $w_1 = w_2$ 
44:  $T_1 = T_2$ 
45:  $w_2 = w_3$ 
46:  $T_2 = T_3$ 
47: Son
48:  $iter = T$  oluncaya kadar
49: Bitir

```

Şekil 26:Fletcher-Reeves Algoritması

3.6.1.2. Türev Bilgisi Kullanmayan Yerel Arama Prosedürleri

Türev bilgisi kullanmayan yerel arama prosedürlerinden olan Hooke-Jeeves Algoritması (Bölüm 3.6.1.2.1) ve Nelder-Mead algoritması (Bölüm 3.6.1.2.2) çalışmaya entegre edilmiştir.

3.6.1.2.1. Hooke-Jeeves Algoritması

Hooke-Jeeves (HJ) metodu (Hooke ve Jeeves 1961) tarafından önerilmiş olup türev bilgisi gerektirmektedir. İki tip hareket içerir:

- *Keşif arama (exploratory search)*: Gelişme yönünde küçük adımlar atarak ilerleme yapar.
- *Örüntü arama (pattern search)*: Gelişme yönünde daha büyük aramalar yapar. İlerleme kaydedildikçe aramaya devam eder.

Şekil 27’de verilen HJ algoritmasının keşif arama safhasında algoritmadaki en iyi çözümden arama adımı büyüklüğü kadar pozitif yönde ilerlenerek yeni bir çözüm elde edilir (Şekil 27, Adım-22). Eğer yeni çözüm en iyi çözümden daha iyi ise, en iyi çözüm yeni çözüm olur, değil ise en iyi çözümden arama adımı büyüklüğü kadar negatif yönde ilerlenerek yeni bir çözüm elde edilir (Adım-30). Eğer keşif arama safhasında gelişme kaydedilememiş ise arama adımı büyüklüğü arama adımı küçültme parametresi ile çarpılır, aksi halde örüntü safhasına girilir. Örüntü arama safhasına gelen çözümden yeni bir çözüm elde edilir (Adım-45) ve yeni çözüm örüntü aramaya giren çözümlerle kıyaslanır (Adım-50). Eğer örüntü aramaya giren çözüm yeni çözümden daha iyi ise en iyi en iyi çözüm olarak belirlenir, aksi halde yeni çözüm en iyi çözüm olur (Adım-52). Eğer bu iki çözümün uygunluk değerleri arasındaki fark belirlenen tolerans değerinden küçük ise algoritma sonlandırılır (Adım-53). Maksimum iterasyon sayısına ulaşıncaya kadar algoritma çalışmaya devam eder.

```

1: Prosedür: Hooke-Jeeves
2: Girdi:  $w_0, T, n, s$ 
3: Çıktı:  $w_{en\ iyi}$ 
4:  $w_0$ : lokal aramaya girecek çözüm
5:  $w_1$ : keşif arama aşamasından çıkan çözüm
6:  $f$ : çözümün uygunluk değeri
7:  $n$ : İndeksteki toplam varlık sayısı
8:  $s$ : iterasyon içerisindeki arama adımı büyüklüğü
9:  $S$ : belirlenen başlangıç arama adımı büyüklüğü
10:  $T$ : maksimum iterasyon sayısı
11:  $\rho$ : arama adımı küçültme parametresi
12: tolerans: yakınsama toleransı
13: Başla
14:  $s = S$ 
15:  $w_{en\ iyi} = w_0$ 
16:  $iter = 0$ 
17: Tekrar Et
18:  $gelişme = 0$ 
19: /*keşif arama safhası*/
20: Tekrar Et
21:    $w_0^i = w_{en\ iyi}^i + s$ 
22:    $w_0 \leftarrow TamirII(w_0)$ 
23:    $w_0 \leftarrow Uyumsuz\ çözüm\ kontrol(w_0)$ 
24:    $w_0 \leftarrow Uygunluk\ hesapla(w_0)$ 
25:   Eğer  $f_0 < f_{en\ iyi}$  ise  $gelişme = 1$  Son
26:   Eğer  $gelişme = 1$  ise
27:      $w_1 = w_0$ 
28:     Değilse
29:      $w_0^i = w_{en\ iyi}^i - s$ 
30:      $w_0 \leftarrow TamirII(w_0)$ 
31:      $w_0 \leftarrow Uyumsuz\ çözüm\ kontrol(w_0)$ 
32:      $w_0 \leftarrow Uygunluk\ hesapla(w_0)$ 
33:     Eğer  $f_0 < f_{en\ iyi}$  ise  $gelişme = 1$  Son
34:     Eğer  $gelişme = 1$  ise
35:        $w_1 = w_0$ 
36:       Son
37:     Son
38:    $i = n$  oluncaya kadar
39:   Eğer  $gelişme = 0$  ise
40:      $s = s * \rho$ 
41:     Değilse
42:     /*örüntü arama safhası*/
43:      $s = S$ 
44:      $w_2 = w_1 + \alpha(w_1 - w_{en\ iyi})$ 
45:      $w_2 \leftarrow TamirII(w_2)$ 
46:      $w_2 \leftarrow Uyumsuz\ çözüm\ kontrol(w_2)$ 
47:      $w_2 \leftarrow Uygunluk\ hesapla(w_2)$ 
48:      $w_2 \leftarrow Keşif\ arama(w_2)$ 
49:     Eğer  $f(w_2) > f(w_1)$  ise  $w_{en\ iyi} = w_1$  Bitir
50:     Değilse
51:        $w_{en\ iyi} = w_2$ 
52:       Eğer  $|f(w_2) - f(w_1)| < tolerans$  ise Bitir
53:     Son
54:   Son
55:    $iter = T$  oluncaya kadar
56:   Bitir
57:

```

/*çözüm Adım 17'de başlayan keşif arama safhasına yeniden girer*/

Şekil 27: Hooke-Jeeves Algoritması

3.6.1.2.2. Nelder-Mead Algoritması

Nelder-Mead (NM) algoritması (Nelder ve Mead 1965) yansıma, genişleme, daralma ve büzülme safhalarından oluşmaktadır. Yansıma safhasında oluşturulan çözüm (w_1) eğer simpleks matristeki sondan bir önceki çözümden iyi ve ilk çözümden kötü ise simpleks matrisin son çözümü bu çözüm olur (Şekil 28, Adım-54), aksi halde eğer w_1 çözümü simplekstekteki en iyi çözümden daha iyi ise genişleme safhasına girilir. Genişleme safhasında oluşturulan çözüm (w_2), w_1 çözümünden daha iyi ise simpleks matrisinin son çözümü w_2 olur, aksi halde w_1 çözümü olur (Adım-61,62). Eğer (w_1) çözümü simpleks matrisindeki son çözümden daha kötü ise daralma safhasına girilir (Adım-64). Daralma safhasında elde edilen çözüm (w_3) simpleks matristeki son çözümden daha iyi ise simpleks matrisin son çözümü bu çözüm olur (Adım-79), değil ise yenileme sayacı ($resCount$) bir artırılır ve büzülme safhasına girilir. Büzülme safhasında algoritmada sayılan yenileme sayacı kadar, simpleks matrisinin sonundan başlayarak (yani en kötü çözümden) arama prosedüründeki en iyi çözüme göre yeniden çözümler oluşturulur (Adım-87). Böylelikle simpleks matrisi yenilenmiş olur. Sonlandırma kriterlerinden birisi sağlanıncaya kadar algoritma çalışmaya devam eder (Adım-100,101) NM algoritması Şekil 28'da verilmiştir.

```
1: Prosedür: Nelder-Mead
2: Girdi:  $w_0, n, \varepsilon$ 
3: Çıktı:  $w_{en\ iyi}$ , gelişme
4:  $n$ : endeksteki toplam varlık sayısı
4:  $\varepsilon$ : endeksteki bir varlığın alabileceği minimum ağırlık
5:  $w_0$ : lokal aramaya giren çözüm
6:  $f$ : çözümün uygunluk değeri
7:  $I$ : birim matris
8:  $\alpha$ : yansıma parametresi
9:  $\gamma$ : genişleme parametresi
10:  $\beta$ : daralma parametresi
11:  $\delta$ : büzülme parametresi
12:  $T$ : maksimum iterasyon sayısı
13:  $IS$ : simpleks matrisi
14:  $resCount$ : simpleks matrisindeki kaç değişkenin yineleneneğini belirleyen sayaç
15:  $resLimit$ : simpleks matrisinin yeniden oluşturulması için aşılması gereken değer
16:  $tolerans$ : yakınsama toleransı
17:  $w_b$ : simpleks matriste tutulan en iyi çözüm
18:  $w_s$ : simpleks matriste tutulan en kötü ikinci çözüm
19:  $w_w$ : simpleks matriste tutulan en kötü çözüm
20: Başla
21:  $w_{en\ iyi} = w_0$ 
22:  $IS_1 = w_{en\ iyi}, i = 2, j = 1$ 
23: Tekrar et
24:   Tekrar et
25:   Eğer  $I_{i-1,j} = 0$  ise  $IS_{i,j} = w_{en\ iyi}^j$ 
26:   Değilse  $IS_{i,j} = w_{en\ iyi}^j + \varepsilon$ 
27:    $j = j + 1$ 
28:    $j = n$  oluncaya kadar
29:    $IS_i \leftarrow Tamir(IS_i)$ 
30:    $IS_i \leftarrow Uyumsuz\ çözüm\ kontrol(IS_i)$ 
31:    $IS_i \leftarrow Uygunluk\ hesapla(IS_i)$ 
32:    $i = i + 1$ 
35:  $i = n + 1$  oluncaya kadar /*IS matrisini satırlardaki çözümlerin uygunluk değerlerini göre sırala*/
36:  $iter = 0$ 
37:  $büzülme = 0$ 
38: Tekrar et
39: Eğer  $büzülme = 0$  ise
40:    $w_1^j = \left( \sum_{i=1}^n IS_i^j \right) / n + \alpha \left( \left( \sum_{i=1}^n IS_i^j \right) / n - IS_{n+1}^j \right)$   $j = 1, 2, \dots, n$  /*yansıma*/
```

```

41:  $w_1 \leftarrow \text{TamirII}(w_1)$ 
42:  $w_1 \leftarrow \text{Uyumsuz çözüm kontrol}(w_1)$ 
43:  $w_1 \leftarrow \text{Uygunluk hesapla}(w_1)$ 
44: Eğer  $f(w_1) < f(w_b)$  ise  $\text{gelişme1} = 1$ 
51: Eğer  $f(w_1) \leq f(w_s)$  ise  $\text{gelişme2} = 1$ 
52: Eğer  $f(w_1) \leq f(w_w)$  ise  $\text{gelişme3} = 1$ 
53: Eğer  $\text{gelişme1} = 0$  ve  $\text{gelişme2} = 1$  ise
54:  $w_w = w_1, \text{resCount} = 0$  /*IS matrisini satırlardaki çözümlerin uygunluk değerlerini göre sırala*/
55: Değilse eğer  $\text{gelişme1} = 1$  ise
57:  $w_2^j = \left( \sum_{i=1}^n IS_i^j \right) / n + \beta \left( w_1^j - \left( \sum_{i=1}^n IS_i^j \right) / n \right) \quad j = 1, 2, \dots, n$  /*genişleme*/
58:  $w_2 \leftarrow \text{TamirII}(w_2)$ 
59:  $w_2 \leftarrow \text{Uyumsuz çözüm kontrol}(w_2)$ 
60:  $w_2 \leftarrow \text{Uygunluk hesapla}(w_2)$ 
61: Eğer  $f(w_2) \leq f(w_1)$  ise  $w_w = w_2$  /*IS matrisini satırlardaki çözümlerin uygunluk
62: Değilse  $w_w = w_1$  değerlerini göre sırala*/
63:  $\text{resCount} = 0$ 
64: Değilse eğer  $\text{gelişme2} = 0$  ise
65: Eğer  $\text{gelişme3} = 1$ 
66:  $w_3^j = \left( \sum_{i=1}^n IS_i^j \right) / n + \gamma \left( w_1^j - \left( \sum_{i=1}^n IS_i^j \right) / n \right) \quad j = 1, 2, \dots, n$  /*daralma*/
67:  $w_3 \leftarrow \text{Tamir}(w_3)$ 
68:  $w_3 \leftarrow \text{Uygulanabilirlik kontrol}(w_3)$ 
69:  $w_3 \leftarrow \text{Uygunluk hesapla}(w_3)$ 
72: Eğer  $f(w_3) < f(w_1)$  ise  $w_w = w_3, \text{resCount} = 0$  /*IS matrisini satırlardaki çözümlerin uygunluk
74: Değilse değerlerini göre sırala*/
75:  $w_3^j = \left( \sum_{i=1}^n IS_i^j \right) / n - \gamma \left( \left( \sum_{i=1}^n IS_i^j \right) / n - IS_{n+1}^j \right) \quad j = 1, 2, \dots, n$ 
76:  $w_3 \leftarrow \text{Tamir}(w_3)$ 
77:  $w_3 \leftarrow \text{Uygulanabilirlik kontrol}(w_3)$ 
78:  $w_3 \leftarrow \text{Uygunluk hesapla}(w_3)$ 
79: Eğer  $f(w_3) < f(w_w)$  ise  $w_w = w_3, \text{resCount} = 0$ 
80: Değilse  $\text{resCount} = \text{resCount} + 1, \text{büzülme} = 1$ 
81: Son
82: Son
83: Değilse
84:  $w_{en\ iyi} = IS_1, \quad i = n + 1, j = 1$ 
85: Tekrar et
86: Tekrar et
87:  $IS_i^j = IS_i^j + \delta(IS_i^j - w_{en\ iyi}^j)$  /*büzülme*/
88:  $j = j + 1$ 
89:  $j = n$  oluncaya kadar
90:  $IS_i \leftarrow \text{TamirII}(IS_i)$ 
91:  $IS_i \leftarrow \text{Uyumsuz çözüm kontrol}(IS_i)$ 
92:  $IS_i \leftarrow \text{Uygunluk hesapla}(IS_i)$ 
93:  $i = i - 1$ 
94:  $i = n + 1 - \text{resCount}$  oluncaya kadar /*IS matrisini satırlardaki çözümlerin uygunluk değerlerini göre
95:  $\text{büzülme} = 0$  sırala*/
96: Son
97:  $w_{en\ iyi} = IS_1$ 
98: Eğer  $f(w_{en\ iyi}) < f(w_0)$  ise  $w_0 = w_{en\ iyi}$  Son /*sadece DKA-I algoritmasında uygulanır*/
99:  $\text{iter} = \text{iter} + 1$ 
100: Eğer  $|f(IS_1) - f(IS_{n+1})| / |f(IS_1) - f(IS_{n+1})| + \text{tolerans} \leq \text{tolerans}$  ise Bitir
101:  $\text{iter} = T$  oluncaya kadar
102: Bitir

```

Şekil 28: Nelder-Mead Algoritması

3.6.2 DKA Algoritması

DKA algoritmasında sarsım aşamasında 4 farklı prosedür (*Shake1, Shake2, Shake3, Shake4*), lokal arama aşamasında da 4 farklı prosedür (*fletcher_reeves, hooke_jeves, steepest_descent, nelder_mead*) kullanılmıştır. Lokal arama aşamasında kullanılan

prosedürler Bölüm 3.6.1'de anlatılmıştır. Sarsım aşamasında kullanılan prosedürler ise [Mladenović vd. \(2008\)](#) tarafından önerilmiştir. Bu prosedürler:

Shake1: d yönü, birim top içerisinde düzgün dağılım kullanılarak üretilmiştir. Birim top, merkezden 1 birim uzaklıktaki noktalar kümesidir. Arama uzayında birim top, mevcut çözümden n -boyutta tam olarak 1 birimlik sıçramayı temsil eder.

Shake2: d yönü, birim küre içerisinde düzgün dağılım kullanılarak üretilmiştir. Birim küre, merkezden 1 birim veya 1 birimden daha az uzaklıktaki noktalar kümesidir. Arama uzayında birim küre, 1 birim veya daha az uzaklıktaki sıçramayı temsil eder.

Shake3: d yönü, birim küre üzerinde özel tasarlanmış bir dağılıma göre şu şekilde üretilmiştir: (i) d_1 koordinatı $[-1,1]$ aralığında düzgün dağılıma göre alınmıştır, d_k koordinatı ise $[-A_k, A_k]$ aralığında düzgün dağılıma göre alınmıştır, burada $A_k = 1 - |d_1| - \dots - |d_{k-1}|, k = 2, \dots, n - 1$ ve son d_n yönü ise rasgele alınmıştır. (ii) bütün d koordinatları rasgele karıştırılır. Bu tasarım sayesinde d koordinatlarından birkaçının 0'dan önemli ölçüde farklı olması, problem boyutu büyüdüğünde DKA algoritmasını hızlandırmaktadır.

Shake4: ilk d yönü birim top içerisinde düzgün dağılım kullanılarak rasgele üretilir. Daha sonra, o anki nokta x ve kısıt vektörleri a ve b ile d_i koordinatları şöyle ölçeklenir: (i) eğer $d_i \geq 0$ ise ölçekleme faktörü $(b_i - x_i)/r_{k_{max}}$, (ii) $d_i < 0$ ise ölçekleme faktörü $(x_i - a_i)/r_{k_{max}}$ olarak belirlenir. Böylece nokta her zaman belirlenen sınırlar içerisinde kalır.

DKA algoritması, düzgün dağılımlı rasgele bir başlangıç çözümü (w_1) yaratılarak başlar (Şekil 29 – Adım 14). Daha sonra, w_1 çözümü kurulan rasgele diziye göre sarsım aşamasındaki prosedürlerden bir tanesine girer (Adım20-23). Sarsım aşamasından çıkan çözüm lokal arama aşamasına girer, bu aşamada çözüm, rasgele sırayla bütün yerel arama prosedürleri ile geliştirilmeye çalışılır (Adım29-33). Lokal arama aşamasından w_1 çözümü eğer algoritmanın o ana kadar elde ettiği en iyi çözümden daha iyi ise, en iyi çözüm w_1 çözümü olur ve sarsım sayacı 1'e eşitlenir, aksi halde, sayaç 1 artırılır. DKA algoritması Şekil 29'da verilmiştir.

```

1: Algoritma: DKA
2: Girdi: Veri  $(R_i, VC_{ij})$  ve parametreler  $(\varepsilon, k, E, T)$ 
3: Çıktı:  $H$ 
4:  $n$ : İndeksteki toplam varlık sayısı
5:  $k$ : Arşiv büyüklüğü
6:  $E$ : tanımlanan  $\lambda$  sayısı
7:  $T$ : maksimum iterasyon sayısı
8:  $H$ : pareto optimal çözümler kümesi
9:  $rp(n)$ : 1'den  $n$ 'ye kadar olan tam sayıların rasgele dizilim kümesi
10:  $f$ : portföyün uygunluk değeri
11: Başla
12:  $H = \emptyset$ 
13:  $e = 1$ 
14:  $w_1 = \varepsilon + r(1 - \varepsilon)$ 
15:   Tekrar Et
16:      $\lambda = (e - 1)/(E - 1)$ 
17:      $SHK = rp(4)$ 
18:     Tekrar Et
19:       Tekrar Et
20:         Eğer  $SHK_i = 1$  ise  $w_1 \leftarrow Shake1$ 
21:         Eğer  $SHK_i = 2$  ise  $w_1 \leftarrow Shake2$ 
22:         Eğer  $SHK_i = 3$  ise  $w_1 \leftarrow Shake3$ 
23:         Eğer  $SHK_i = 4$  ise  $w_1 \leftarrow Shake4$ 
24:          $w_1 \leftarrow Tamir(w_1)$ 
26:          $w_1 \leftarrow Uyumsuz\ çözümler\ kontrol(w_1)$ 
27:          $w_1 \leftarrow Uygunluk\ hesapla(w_1)$ 
28:         Tekrar et
29:           Eğer  $rp(4)_j = 1$  ise  $w_2 \leftarrow fletcher\_reeves(w_1)$ 
31:           Eğer  $rp(4)_j = 2$  ise  $w_2 \leftarrow hooke\_jeeves(w_1)$ 
32:           Eğer  $rp(4)_j = 3$  ise  $w_2 \leftarrow steepest\_descent(w_1)$ 
33:           Eğer  $rp(4)_j = 4$  ise  $w_2 \leftarrow nelder\_mead(w_1)$ 
34:            $j = j + 1$ 
35:            $j = 4$  oluncaya kadar
36:           Eğer  $f(w_2) < f(w_{en\ iyi})$  ise
37:              $w_{en\ iyi} = w_2, i = 1$ 
38:           Değilse  $i = i + 1$ 
39:            $i = 4$  oluncaya kadar
40:            $iter = iter + 1$ 
41:            $iter = T$  oluncaya kadar
42:            $H_e = w_{en\ iyi}$  /*algoritmanın en iyi çözümü pareto optimal çözümler kümesine atılır*/
43:            $e = e + 1$ 
44:            $e = E$  oluncaya kadar
45: Bitir

```

Şekil 29: DKA algoritması

4. BULGULAR

Bu bölümde geliştirilen algoritmaların, [Bölüm 3.1](#)'de verilen veri setleri kullanılarak yapılan deneysel çalışmalardan elde edilen bulgular verilmiştir. Geliştirilen algoritmalar MATLAB R2016b kullanılarak modellendi ve Microsoft Visual C++ kullanılarak test edilmiştir. Deneysel testler ise Intel Xeon E5-2650 2.0 GHz işlemcili 32 GB RAM kapasiteli bir çalışma istasyonu kullanılarak yapılmıştır.

Eleman sayısı ve minimum, maksimum ağırlık kısıtları için problem parametreleri ε ve K , literatürdeki diğer çalışmalar ile kıyaslama yapabilmek için sırası ile 0.01 ve 10 değerlerine sabitlenmiştir.

Sonraki bölümlerde geliştirilen algoritmalar ile yapılan deneysel testlerin sonuçları verilmiştir.

4.1. YAK-II algoritmasının deneysel sonuçlar

Bu bölümde YAK-II algoritması kullanılarak yapılan deneysel sonuçlar verilmiştir. [Bölüm 4.1.1](#)'de algoritmanın parametrelerini belirlemek için yapılan analizler, [Bölüm 4.1.2](#)'de ise YAK-I ve YAK-II algoritmaları ile oluşturulan etkin sınırlar ve algoritmaların performans karşılaştırmaları verilmiştir.

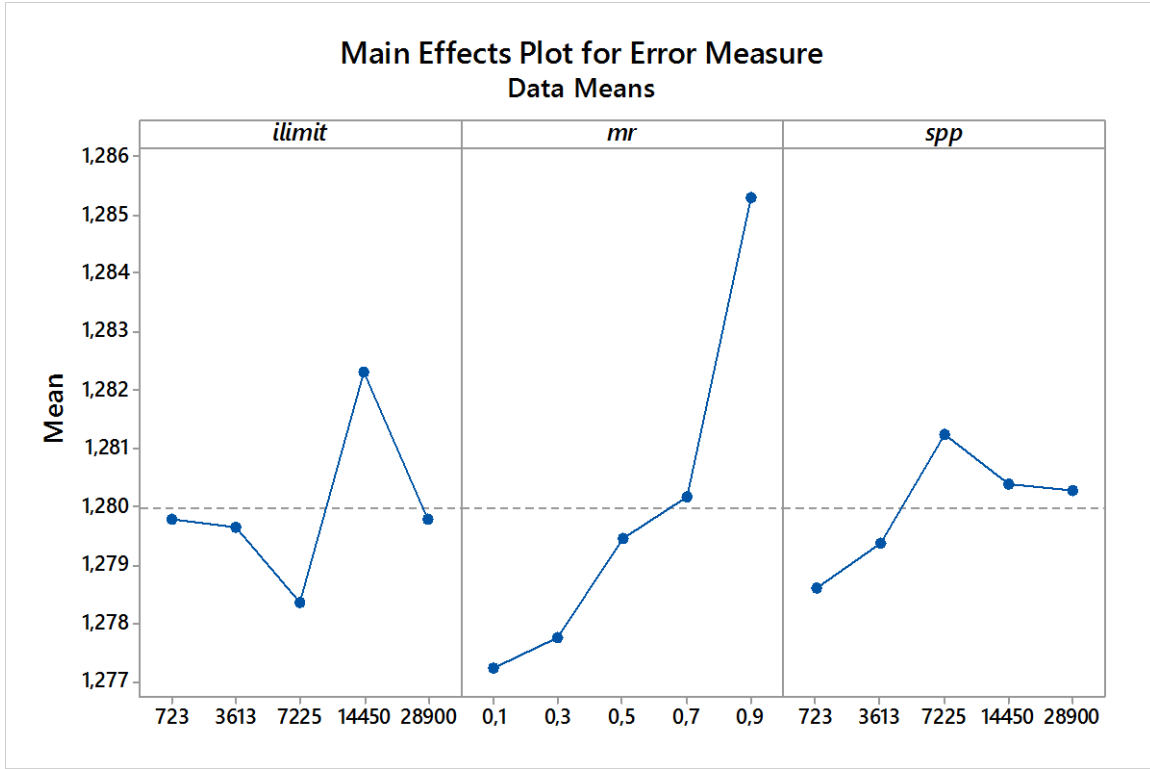
4.1.1. Parametre Analizi

Geliştirilen YAK algoritmaları üç parametre içermektedir, bunlar: kaşif arıların üretilme periyodu (spp), kaşif arıların devreye girmesi için aşılması gereken limit değeri ($ilimit$) ve modifikasyon oranı (mr). Optimal parametre setinin belirlenmesi için yapılan deneysel tasarımda kullanılan seviyeleri Tablo 13'de verilmiştir.

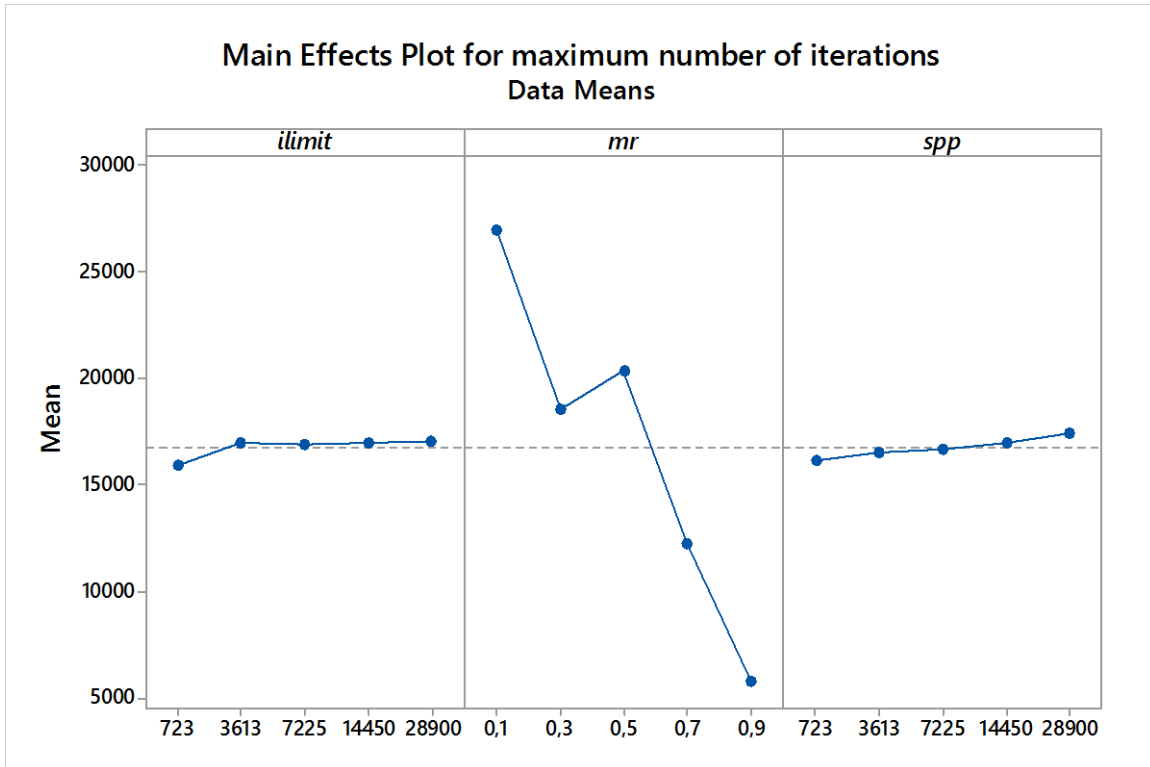
Tablo 13: Deneysel tasarımda kullanılan seviyeler

Level	spp	$ilimit$	mr
1	$0.1 \times PS \times N$	$0.1 \times PS \times N$	0.1
2	$0.5 \times PS \times N$	$0.5 \times PS \times N$	0.3
3	$1 \times PS \times N$	$1 \times PS \times N$	0.5
4	$2 \times PS \times N$	$2 \times PS \times N$	0.7
5	$4 \times PS \times N$	$4 \times PS \times N$	0.9

Tablo 13'deki seviyeler ile 85 hisseli DAX 100 veri seti kullanılarak, tam faktöriyel deney tasarımı yapılmıştır. Minitab kullanılarak elde edilen ana etki grafikleri, hata ölçütü için (GHV-II) Şekil 30'da ve maksimumu iterasyon sayısı için Şekil 31'de verilmiştir.



Şekil 30: Hata ölçütü için ana etki grafikleri



Şekil 31: Maksimum iterasyon sayısı için ana etki grafikleri

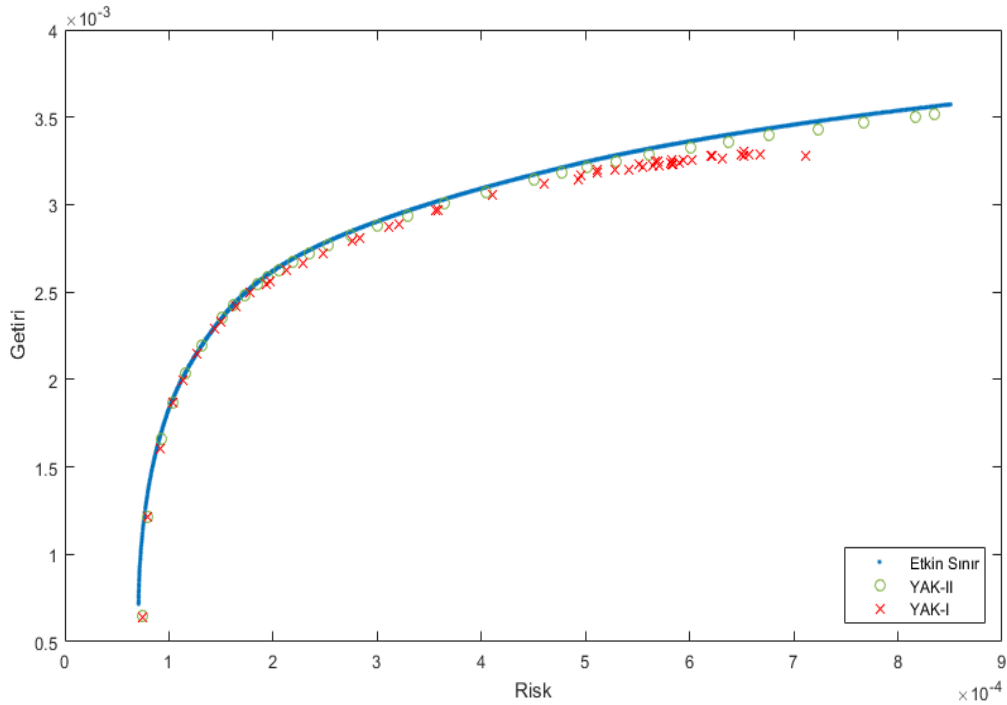
Şekil 30 göstermektedir ki mr ve spp parametrelerinin küçük değerleri için GHV-II hatası düşmektedir. Ancak Şekil 31'de görüldüğü üzere $ilimit$ ve spp değerlerinin düşük değerleri için maksimum iterasyon sayısı düşerken mr parametresi için bunun tam tersidir.

Sonuç olarak yapılan deneylerde YAK-I ve YAK-II algoritmalarında $spp = 0.1 \times PS \times N$, $ilimit = 1 \times PS \times N$ ve $mr = 0.1$ olarak sabitlenmiştir.

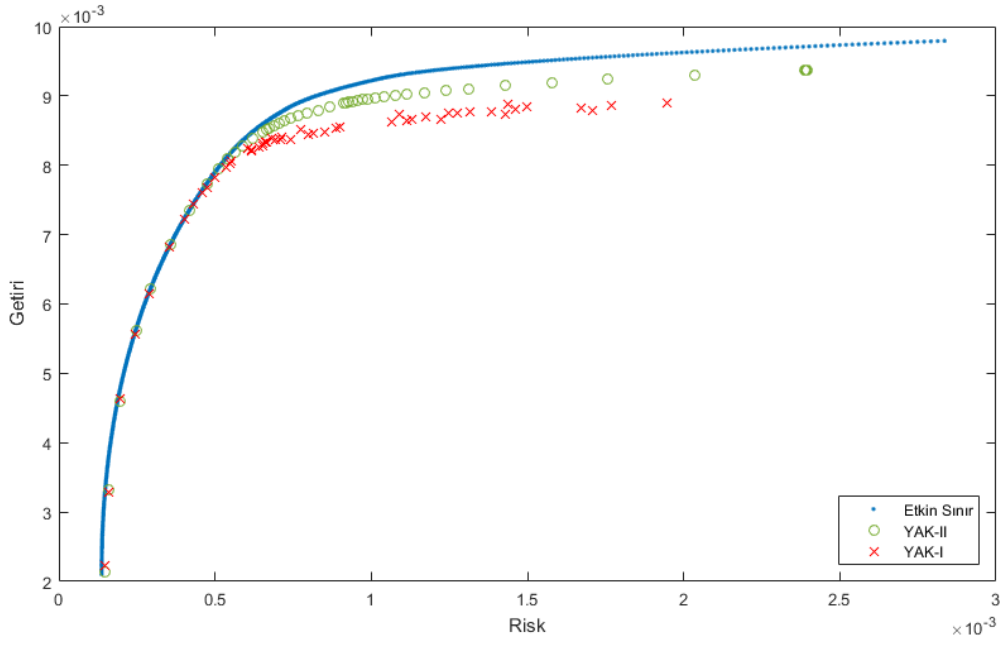
4.1.2. YAK-I ve YAK-II algoritmaları için deneysel sonuçlar

Bu bölümde YAK-I ve YAK-II algoritmaları ile elde edilen kısıtlı etkin sınırlar ve Bölüm 3.1'de verilen performans ölçütleri kullanılarak hesaplanan performans sonuçları karşılaştırmalı olarak verilmiştir.

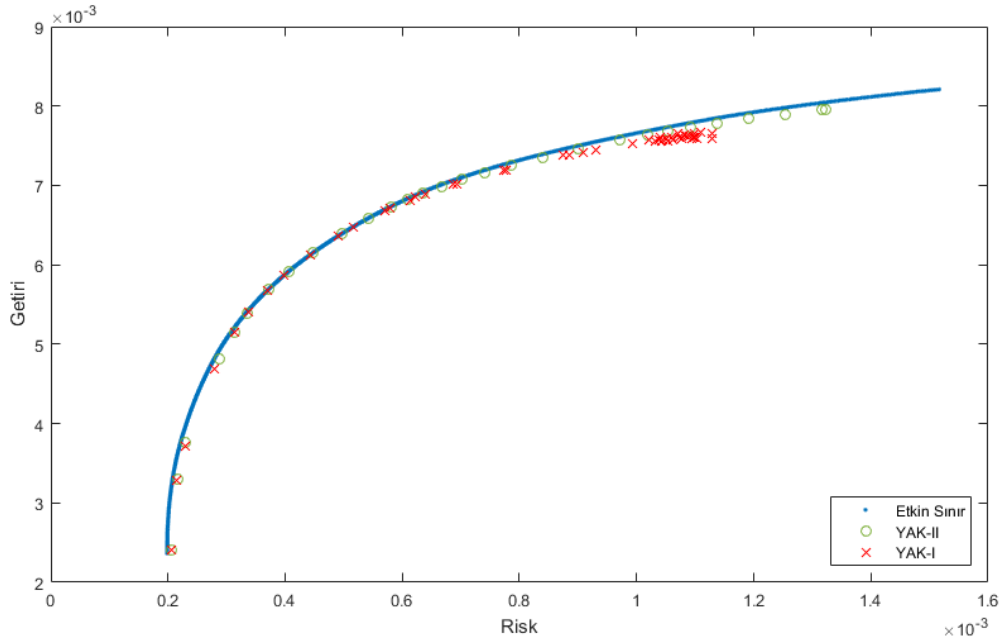
Tablo 14'de görüldüğü üzere YAK-II algoritması YAK-I algoritmasına bütün veri setlerinde bütün performans ölçütleri için üstünlük sağlamıştır. YAK-I ve YAK-II algoritmaları ile çizilen kısıtlı etkin sınırlar: Hang Seng Şekil 32'de, DAX 100 Şekil 33'de, FTSE 100 Şekil 34'de, S&P 100 Şekil 35'de, NIKKEI Şekil 36'da, BİST30 Şekil 37'de ve BİST100 Şekil 38'de verilmiştir.



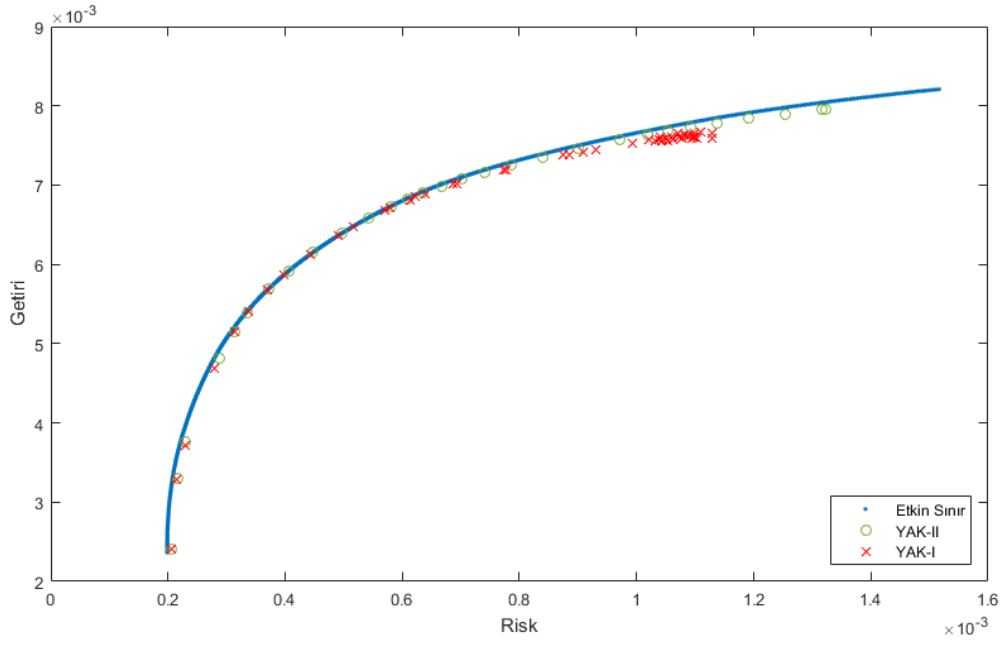
Şekil 32: Hang Seng veri seti için YAK-I ve YAK-II algoritmaları ile çizilen etkin sınırlar



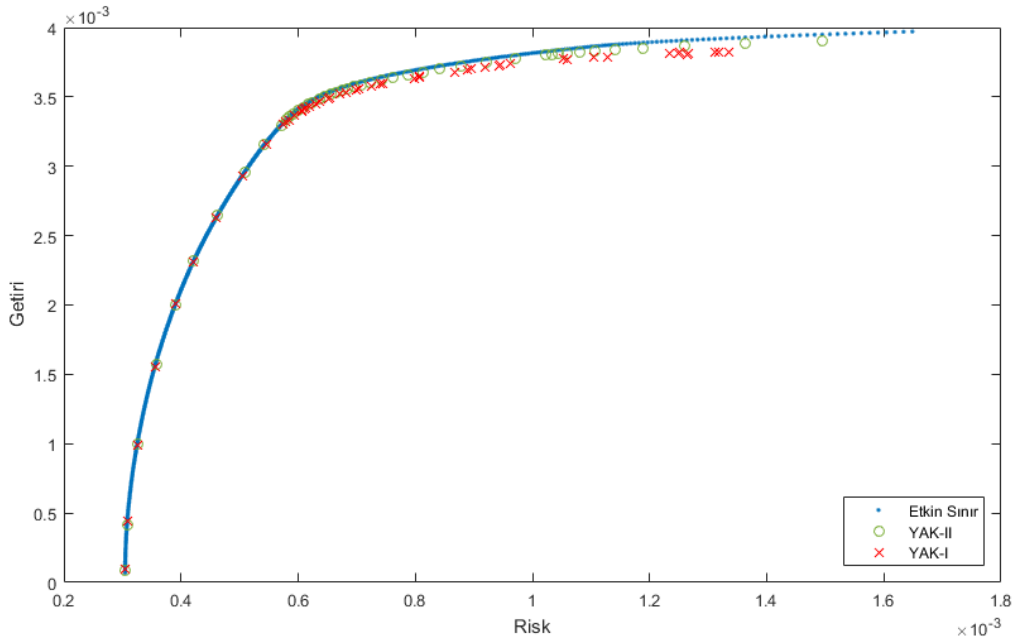
Şekil 33: DAX 100 veri seti için YAK-I ve YAK-II algoritmaları ile çizilen etkin sınırlar



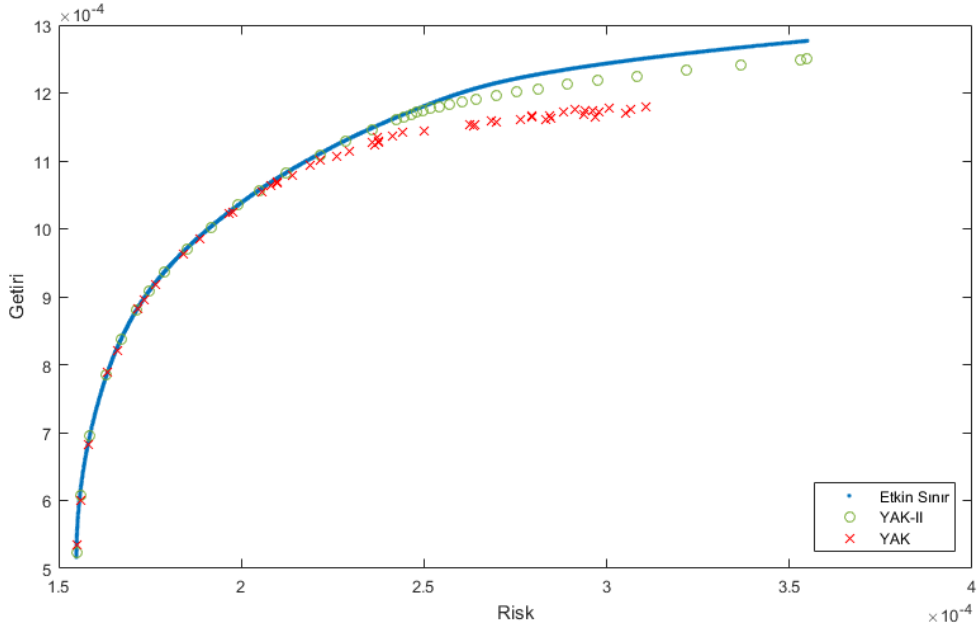
Şekil 34: FTSE 100 veri seti için YAK-I ve YAK-II algoritmaları ile çizilen etkin sınırlar



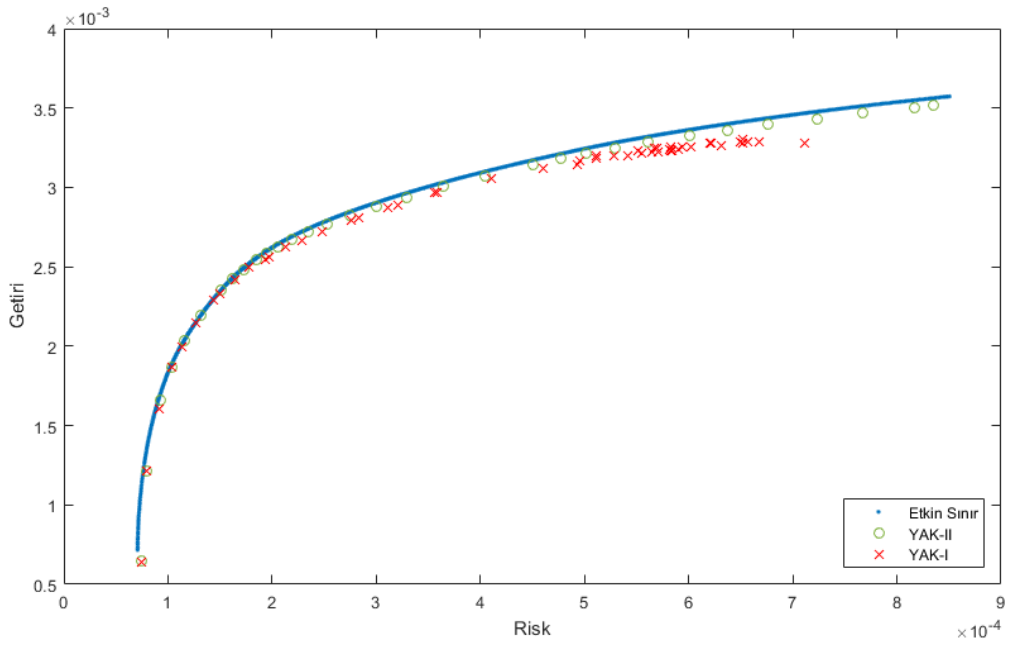
Şekil 35: S&P 100 veri seti için YAK-I ve YAK-II algoritmaları ile çizilen etkin sınırlar



Şekil 36: Nikkei veri seti için YAK-I ve YAK-II algoritmaları ile çizilen etkin sınırlar



Şekil 37: BİST 30 veri seti için YAK-I ve YAK-II algoritmaları ile çizilen etkin sınırlar



Şekil 38: BİST 100 veri seti için YAK-I ve YAK-II algoritmaları ile çizilen etkin sınırlar

Tablo 14: YAK-I ve YAK-II algoritmalarının performans karşılaştırması

INDEKS N	Hang Seng 31			DAX 100 85			FTSE 100 89			S&P 100 98		
	P. Ölçütü	YAK-I	YAK-II	Δ	YAK -I	YAK -II	Δ	YAK -I	YAK -II	Δ	YAK -I	YAK -II
YHO	2.4327	1.0953	54.9760	4.4799	2.3295	48.0011	1.4883	0.8481	43.0155	2.0948	1.2930	38.2757
YHMED	2.6909	1.2181	54.7326	4.4023	2.5678	41.6714	1.7776	1.0841	39.0133	2.0063	1.1369	43.3335
;YHMİN	0.0329	0.0000	100.0000	0.0859	0.0023	97.3225	0.0310	0.0047	84.8387	0.0194	0.0000	100.0000
YHMAKS	4.4815	1.5538	65.3286	7.9642	4.0275	49.4299	3.2289	2.0638	36.0835	5.7988	5.4422	6.1495
GHV-I	8.9629	4.1302	53.9189	41.9042	32.6157	22.1660	7.3210	4.5658	37.6342	14.5024	11.5133	20.6111
OGH-I	2.7448	1.1364	58.5981	5.2692	3.0743	41.6553	2.1839	1.4328	34.3926	3.2236	2.3309	27.6926
OÖÜ	0.0002	0.0001	50.0000	0.0002	0.0001	50.0000	0.0001	0.0000	100.0000	0.0001	0.0001	0.0000
GHV-II	4.0351	1.6432	59.2773	15.9677	6.7925	57.4610	4.8366	2.4397	49.5575	5.4076	2.5260	53.2880
OGH-II	1.1899	0.6047	49.1806	1.6927	1.2761	24.6116	0.3926	0.3255	17.0912	1.2345	0.8885	28.0275

INDEKS N	NIKKE 225			XU030 30			XU100 99		
	P. Ölçütü	YAK -I	YAK -II	Δ	YAK -I	YAK -II	Δ	YAK -I	YAK -II
YHO	1.2600	0.5781	54.1190	3.2030	1.1256	64.8579	2.3370	1.0272	56,0462
YHMED	1.3379	0.5856	56.2299	4.5870	1.2549	72.6422	2.4420	1.1586	52,5553
;YHMİN	0.0114	0.0000	100.0000	0.0268	0.0012	95.5224	0.3779	0.0132	96,5070
YHMAKS	2.6911	1.1606	56.8727	5.9630	2.1286	64.3032	3.7930	1.7384	54,1682
GHV-I	9.0457	5.5757	38.3608	11.7200	6.5390	44,2065	12.22	5.6169	54,0352
OGH-I	2.6666	1.6764	37.1334	3.5100	1.2537	64,2821	3.325	2.0126	39,4707
OÖÜ	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
GHV-II	1.9814	0.8396	57.6259	7.2240	1.9324	73,2503	6.6050	2.4973	62,1908
OGH-II	0.8514	0.4127	51.5269	1.1670	0.7358	36,9494	1.1200	0.7306	34,7679

Δ performanstaki iyileştirmeyi temsil etmekte ve şu formüle göre hesaplanmaktadır: $\frac{(YAK-II)-(YAK-I)}{YAK-I}$

4.2. KKO algoritmasının deneysel sonuçlar

Bu bölümde Bölüm 3.5’de anlatılan KKO algoritması için yapılan parametre analizi ve KKO algoritması ile elde edilen kısıtlı etkin sınırlar ve performans sonuçları verilmiştir.

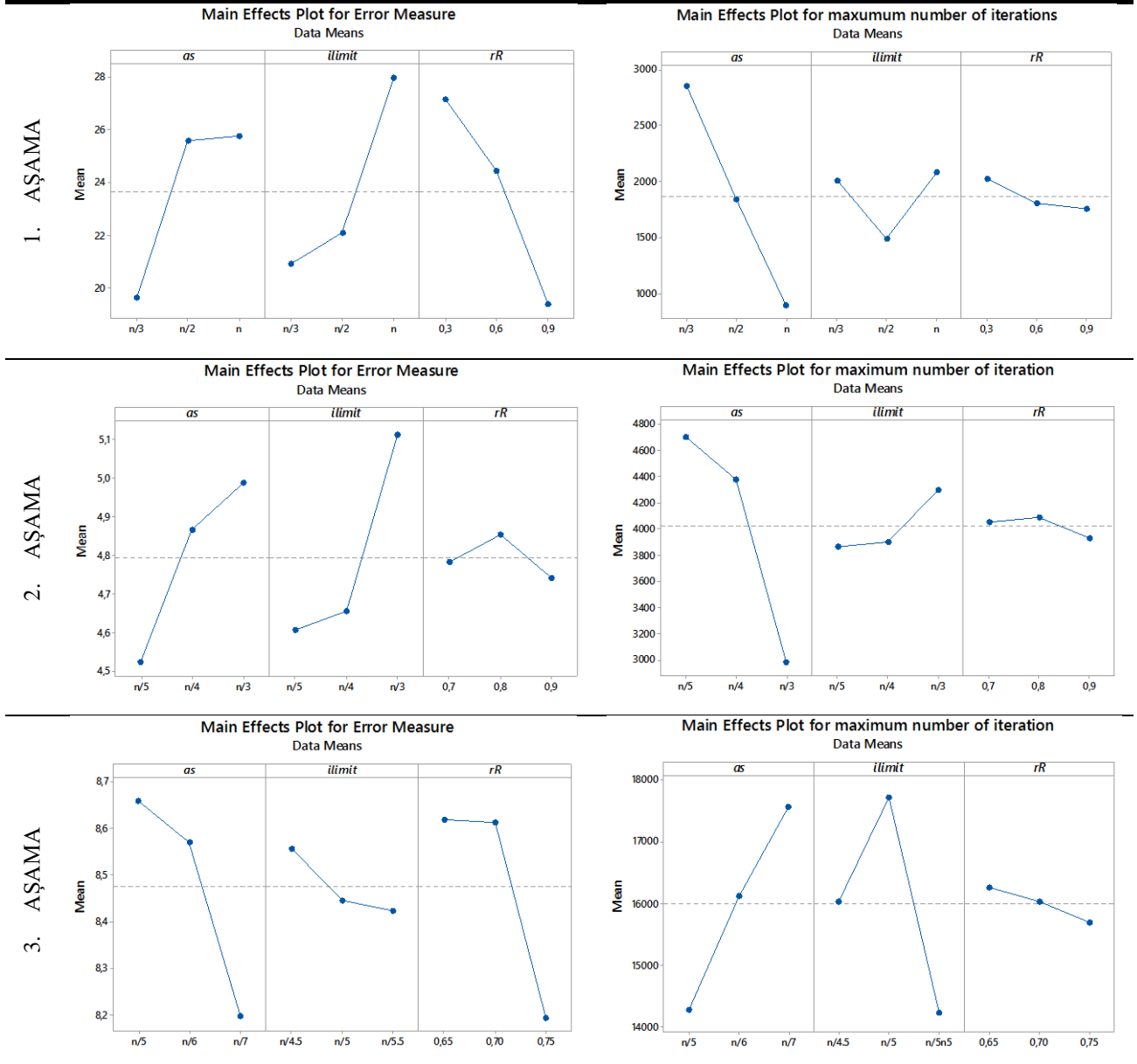
4.2.1. Parametre Analizi

KKO algoritmasındaki değiştirme oranı (mr), arşivdeki en iyi çözümün seçilme olasılığı ($qBest$) ve feromon uçurma (ξ) parametreleri algoritma içerisinde iterasyon sayısına bağlı lineer olarak değişmektedir. Bu parametrelerin başlangıç değerleri ve bitiş değerlerine göre iterasyonlar boyunca azalmakta ya da artmaktadır (Tablo 15).

Tablo 15: mr , $qBest$ ve ξ parametrelerinin başlangıç ve bitiş değerleri

	Başlangıç değeri	Bitiş değeri
mr	0.3	0.1
$qBest$	0.25	0.5
ξ	0.7	0.3

KKO algoritmasının diğer üç parametresi (arşiv büyüklüğü: as , arşivin yenilenme oranı: rR ve arşivin yenilenmesi için aşılması gereken değer: $ilimit$) için DAX 100 veri seti kullanılarak üç aşamalı bir deneysel tasarım yapılmıştır. Yapılan parametre analizi çalışmasının ilk aşamasında geniş bir aralıklı seviyeler belirlenmiş, ikinci ve üçüncü aşamalarında ise bir önceki aşamanın sonuçları doğrultusunda daha dar aralıklar belirlenmiştir. Parametre analizinin her aşamasında belirlenen seviyeler için tam faktöriyel deneysel tasarım kullanılmıştır. Yapılan üç aşamalı parametre analizi sonucunda elde edilen ana etki grafikleri Şekil 39’da verilmiştir. Her aşamadaki ana etki grafikleri hata ölçütü için (GHV-II) ve maksimum iterasyon sayısı için yapılmıştır.

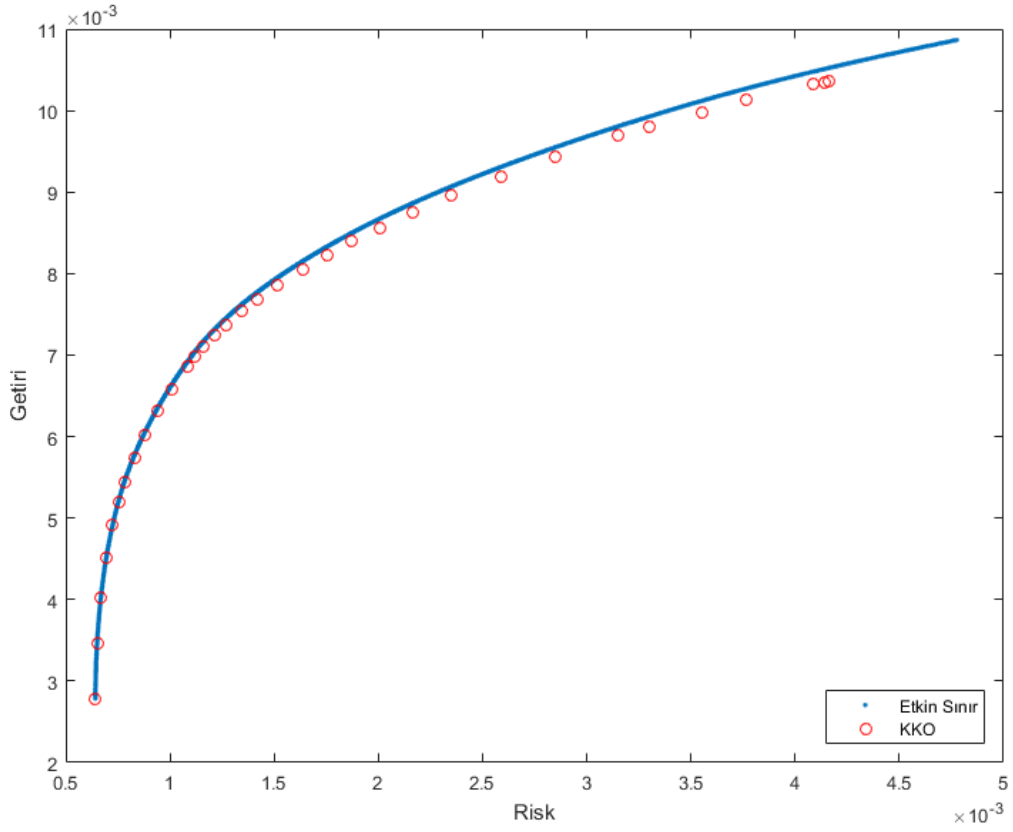


Şekil 39: *as*, *ilimit* ve *rR* parametreleri için yapılan üç aşamalı parametre analizi

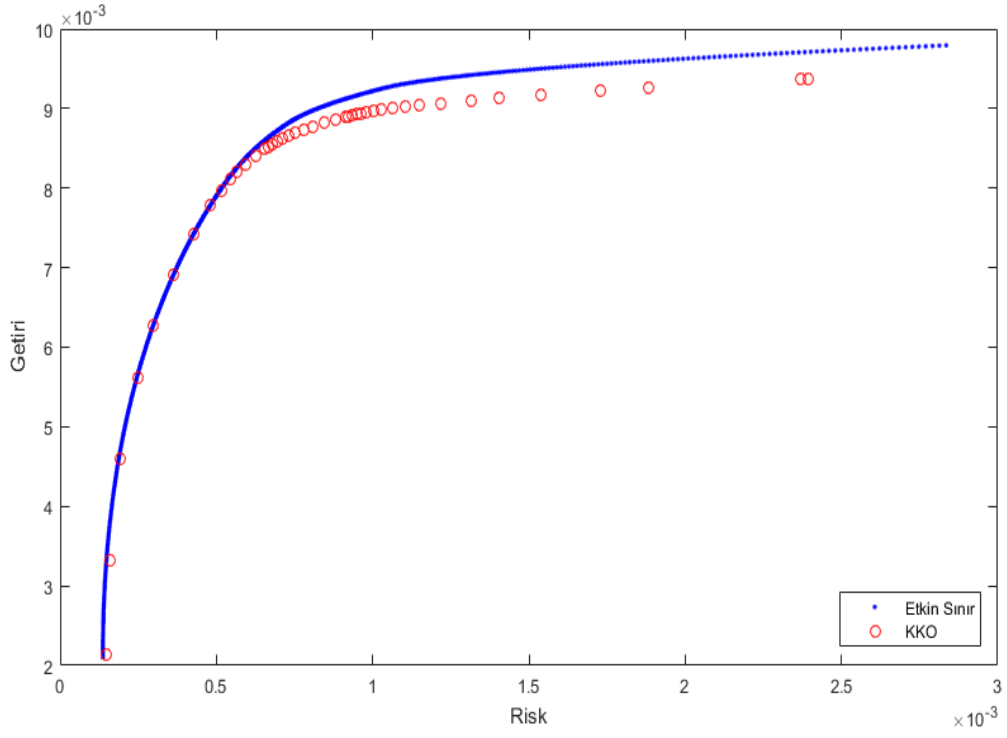
Yapılan parametre analizi göstermektedir ki *as* parametresinin düşük değerleri, hata ölçütü için iyi performans sergilese de maksimum iterasyon sayısını artırmaktadır. Düşük arşiv büyüklüğünde iterasyon sayısının artması her ne kadar istenmeyen bir şey olsa da, algoritmadaki bir iterasyon düşük arşiv büyüklüğünde daha az süre almaktadır. Yani toplam çalışma süresinde arşiv büyüklüğünün etkisi çok azdır. Bu sebepten ötürü bu parametre için hata ölçütünde en iyi performansı veren seviye (*n/7*) seçilmiştir. *ilimit* için *n/5.5* seviyesi hem hata ölçütünde hem de maksimum iterasyon sayısında en iyi sonuçları verdiği için *ilimit* parametresi bu seviyeye sabitlenmiştir. Aynı şekilde *rR* parametresinin her iki değerlendirme kriterindeki en iyi performansı 0.75 olduğu için *rR* parametresi bu değere sabitlenmiştir.

4.2.2. KKO Algoritması İçin Deneysel Sonuçlar

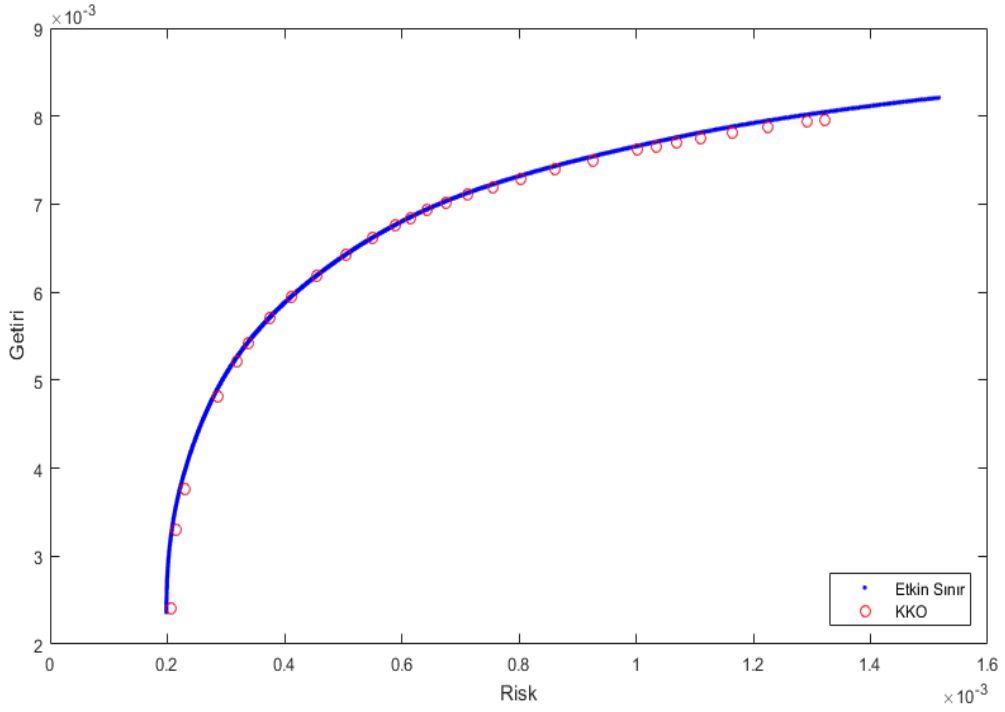
Tablo 16'da KKO algoritmasının bütün veri setleri için performans sonuçları verilmiştir. KKO algoritması ile çizilen kısıtlı etkin sınırlar: Hang Seng Şekil 40'da, DAX 100 Şekil 41'de, FTSE 100 Şekil 42'de, S&P 100 Şekil 43'de, Nikkei Şekil 44'de, BİST30 Şekil 45'de ve Şekil 46'da verilmiştir.



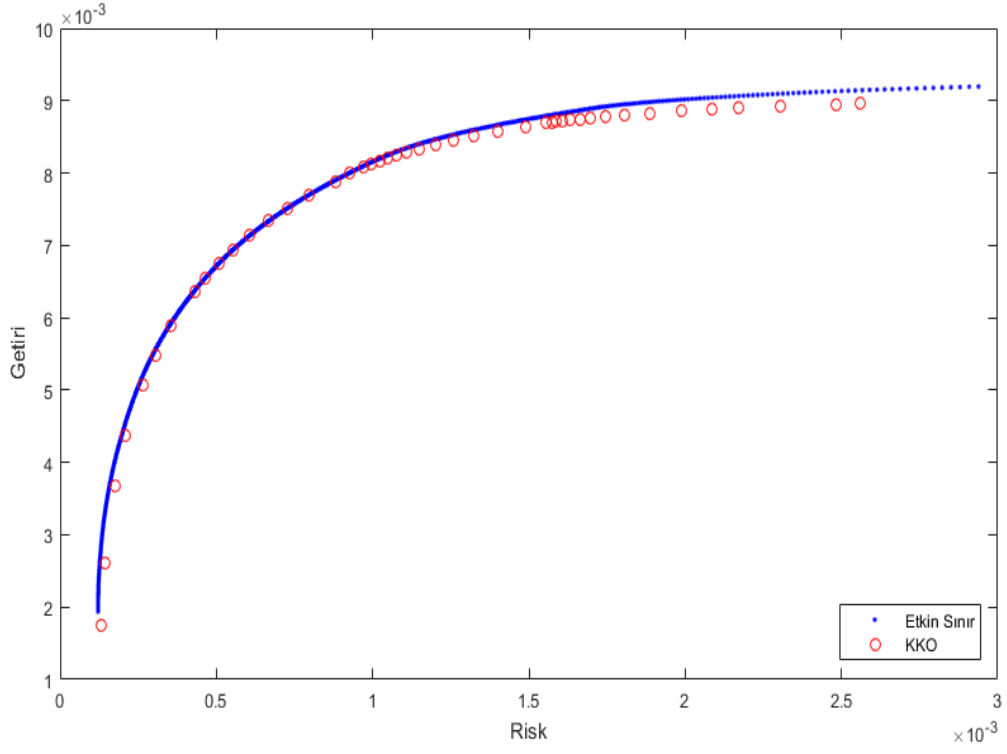
Şekil 40: Hang Seng veri seti için KKO algoritması ile çizilen etkin sınır



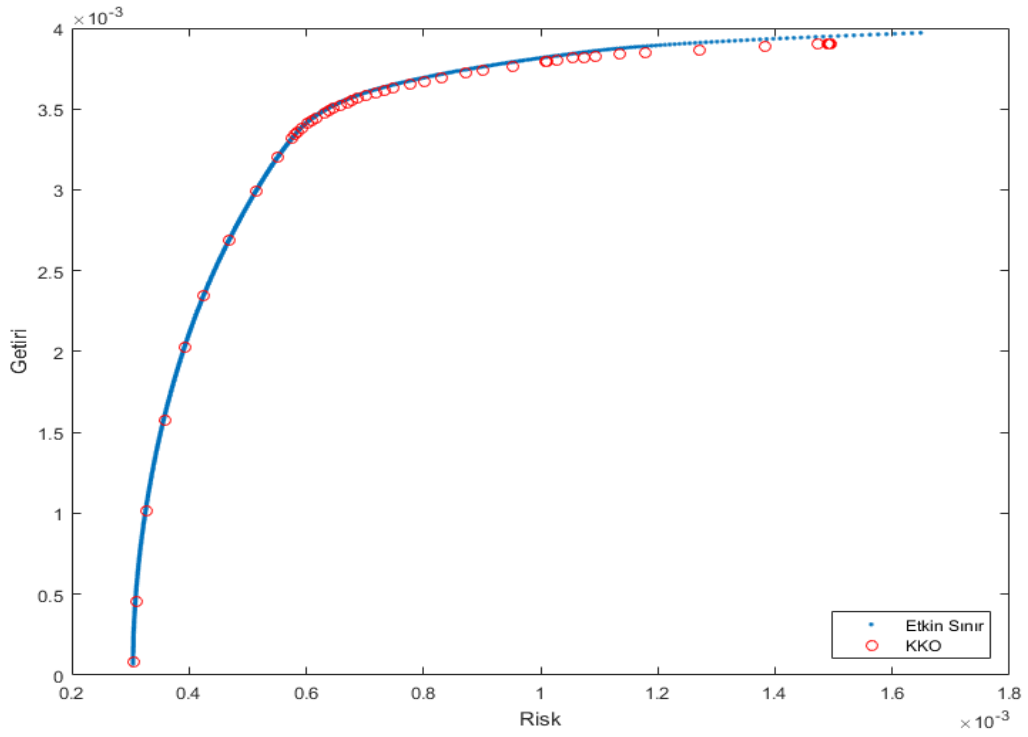
Şekil 41: DAX 100 veri seti için KKO algoritması ile çizilen etkin sınır



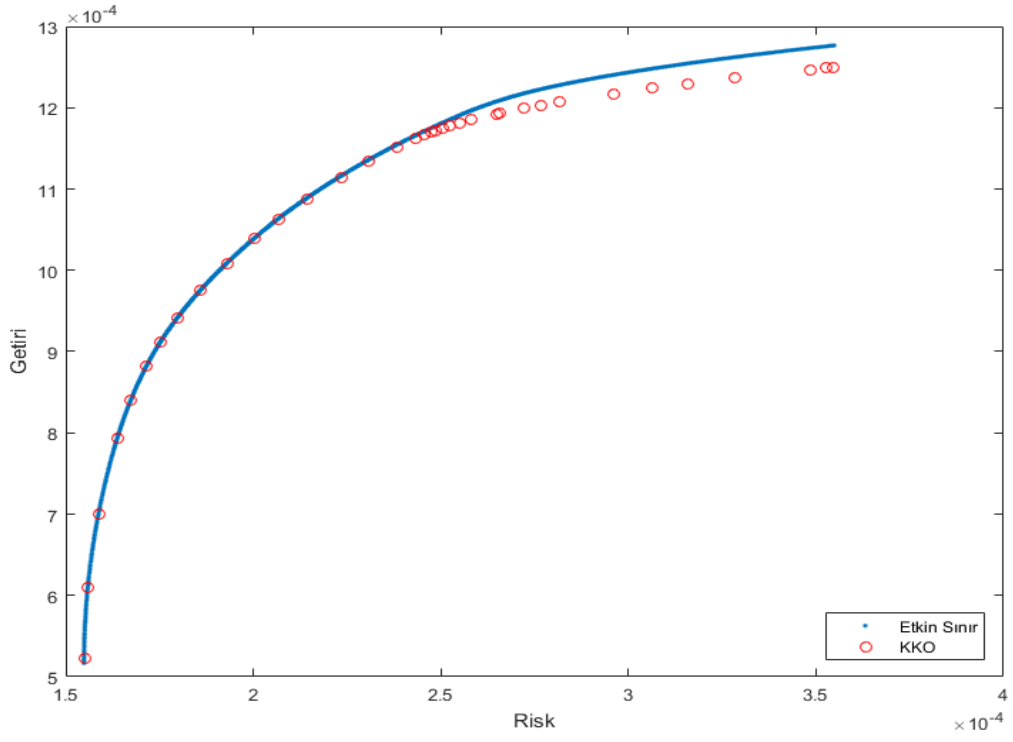
Şekil 42: FTSE 100 veri seti için KKO algoritması ile çizilen etkin sınır



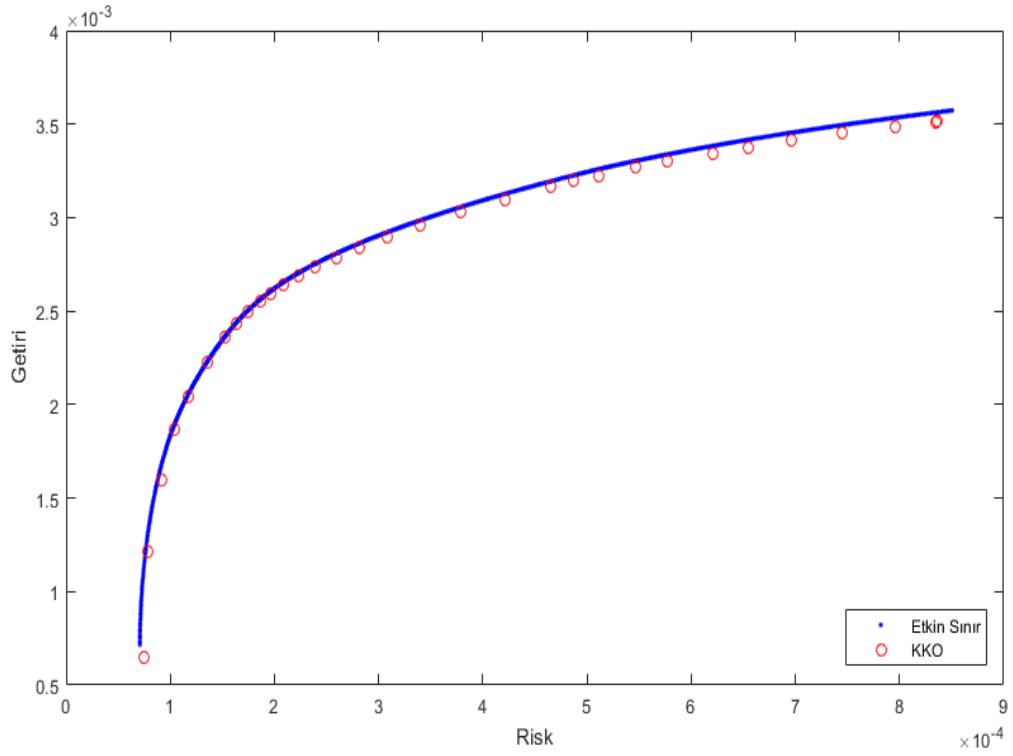
Şekil 43: S&P 100 veri seti için KKO algoritması ile çizilen etkin sınır



Şekil 44: Nikkei veri seti için KKO algoritması ile çizilen etkin sınır



Şekil 45: BİST 30 veri seti için KKO algoritması ile çizilen etkin sınır



Şekil 46: BİST 100 veri seti için KKO algoritması ile çizilen etkin sınır

Tablo 16: KKO algoritması için performans sonuçları

İndeks	N	Performans Ölçütü	KKO
Hang Seng	31	YHO	1.0960
		YHMED	1.2080
		YHMİN	0.0000
		YHMAKS	1.554
		GHV-I	4.1310
		OGH-I	1.1370
		OÖÜ	0.0000
		GHV-II	1.6350
DAX 100	85	YHO	2.3298
		YHMED	3.5715
		YHMİN	0.0022
		YHMAKS	4.0260
		GHV-I	32.7535
		OGH-I	3.1607
		OÖÜ	0.0001
		GHV-II	6.7701
FTSE 100	89	YHO	1.2767
		YHMED	0.8400
		YHMİN	1.0840
		YHMAKS	0.0047
		GHV-I	2.1165
		OGH-I	4.5600
		OÖÜ	1.5370
		GHV-II	0.0000
S&P 100	98	YHO	2.4329
		YHMED	0.3259
		YHMİN	1.2850
		YHMAKS	1.1377
		GHV-I	0.0000
		OGH-I	5.4340
		OÖÜ	11.8786
		GHV-II	1.6518
Nikkei	225	YHO	0.0001
		YHMED	2.3956
		YHMİN	0.6908
		YHMAKS	0.5844
		GHV-I	0.5894
		OGH-I	0.0000
		OÖÜ	1.1680
		GHV-II	5.6280
Bist30	30	YHO	1.7820
		YHMED	0.0000
		YHMİN	0.8385
		YHMAKS	0.4229
		GHV-I	1.1210
		OGH-I	1.3150
		OÖÜ	0.0004
		GHV-II	2.0840
Bist100	99	YHO	6.5930
		YHMED	1.2500
		YHMİN	0.0000
		YHMAKS	1.9220
		GHV-I	1.0438
		OGH-I	1.2067
		OÖÜ	0.0185
		GHV-II	1.8538
		YHO	5.6353
		YHMED	2.0626
		YHMİN	0.0000
		YHMAKS	2.5234
		GHV-I	0.7404
		OGH-I	0.7404
		OÖÜ	0.7404
		GHV-II	0.7404

4.3. DKA algoritmasının deneysel sonuçlar

4.3.1. Parametre Analizi

DKA algoritmasındaki parametrelerin etki analizi yapılırken, yerel arama operatörlerinde kullanılan parametre değerleri üzerinde birçok öncü deney yapılmış, ancak önemli bir performans farkı yakalanamamıştır. Bu nedenle, genel algoritma performansındaki artış için yerel arama algoritmalarında bazı değişikliklere gidilmiştir. Yerel Arama Prosedürlerinde kullanılan parametreler ve yapılan değişiklikler Tablo 17’de özetlenmiştir.

Tablo 17: Yerel Arama Prosedürlerinde kullanılan parametreler ve yapılan değişiklikler

Literatürdeki yaygın kullanım					Yapılan değişiklik		
Türev bilgisi kullanan	Steepest-Descent Algoritması	S	ρ		S	ρ	
			- Sabitlenmiş adım uzuluğu - Geri izlemeleli doğru arama - Tam doğru arama		$r \sim (0,1)$	0.5	
	Fletcher-Reeves Algoritması	S	ρ	d_1, d_2, d_3	S	ρ	
			- Sabitlenmiş adım uzuluğu - Geri izlemeleli doğru arama - Tam doğru arama	1e-10	$r \sim (0,1)$	0.5	
Türev bilgisi kullmayan	Hooke-Jeeves Algoritması	S	ρ	<i>tolerans</i>	S	ρ	
		1	0.5	1e-10	$r \sim (0,1)$	0.5	
	Nelder-Mead Algoritması		α	γ	β	δ	<i>tolerans</i>
		1	0.5	2	0.5	1e-10	Standart parametre değerleri kullanılmış, ancak başlangıç simpleksi oluşturma tekniği değiştirilmiştir.

S: belirlenen başlangıç arama adımı büyüklüğü; ρ : arama adımı küçültme parametresi; d_1, d_2, d_3 : durdurma kriterleri; *tolerans*: yakınsama toleransı; α : yansıma parametresi; γ : genişleme parametresi; β : daralma parametresi; δ : büzülme parametresi

Türev bilgisi kullanan algoritmalarından Steepest-Descent ve Fletcher-Reeves Algoritmalarında kısıtsız fonksiyon optimizasyonunda, genellikle; sabitlenmiş adım uzuluğu, geri izlemeleli doğru arama veya tam doğru arama teknikleri kullanılarak parametre optimizasyonuna gidilmektedir. Ancak, eleman sayısı kısıtlı portföy optimizasyonu probleminde, bu algoritmalar çok fazla sayıda çağırıldığı için bu tarz uygulamalar aşırı zaman kaybına neden olmaktadır. Bu nedenle, başlangıç adım uzunluğu düzgün rassal bir sayı ($r \sim (0,1)$) ile başlatılmış ve her adımda yarıya düşürülerek hassas arama yapılmasına olanak tanınmıştır. Böyle bir uygulamada, başlangıç adım uzunluğunun rassal bir sayı ile belirlenmesi sayesinde arama adımı küçültme parametresinin önemini azaltmış ve $\rho = 0.5$ değerine sabitlenerek literatürdeki

birçok sabitlenmiş adım uzunluğu uygulamalarındaki değer ile eşitlenmiştir. Hooke-Jeeves algoritmasında da benzer şekilde bir yöntem kullanılmıştır. Bu sayede, DKA algoritması içerisinde çağırılan yerel arama algoritmaları, her çağırıldıklarında farklı rassal sayı ile aramaya başlayarak çeşitliliğin önünü açmaktadır. Nitekim, deney sonuçları da çok hızlı ve literatüre kıyasla daha az hatayla etkin sınıra yakınsadığını göstermektedir. Nelder-Mead algoritmasında ise fazla parametre olması nedeniyle birçok öncü deney çalışması yapılmış, ancak önemli bir performans artışı yakalanamamıştır. Başlangıç deneylerinde düşük performans gösteren Nelder-Mead algoritmasındaki portföy optimizasyonu için ana problemin başlangıç simplexinin oluşturulması olduğunu fark edilerek, Nelder-Mead algoritması portföy optimizasyonuna özgü yeni bir simplex oluşturma formülasyonu geliştirmiştir.

Şekil 28, adım 25-26'da da verilen formülasyon (Denklem 35), literatürde yaygın olarak kullanılan standart formülasyondan (Denklem 36) farklıdır.

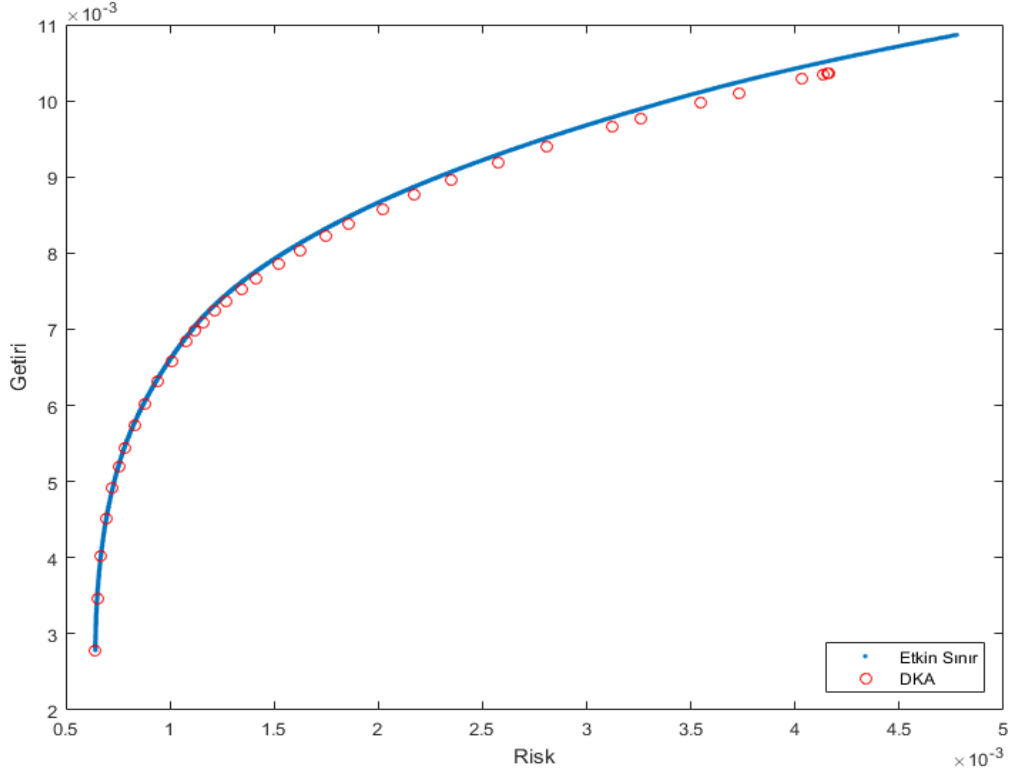
$$IS_{i,j} = \begin{cases} 0 & \text{eğer } I_{i-1,j} = 0 \\ \varepsilon & \text{aksi halde} \end{cases} \quad (35)$$

$$IS_{i,j} = \begin{cases} 0.05 & \text{eğer } I_{i-1,j} = 0 \\ 0.00025 & \text{aksi halde} \end{cases} \quad (36)$$

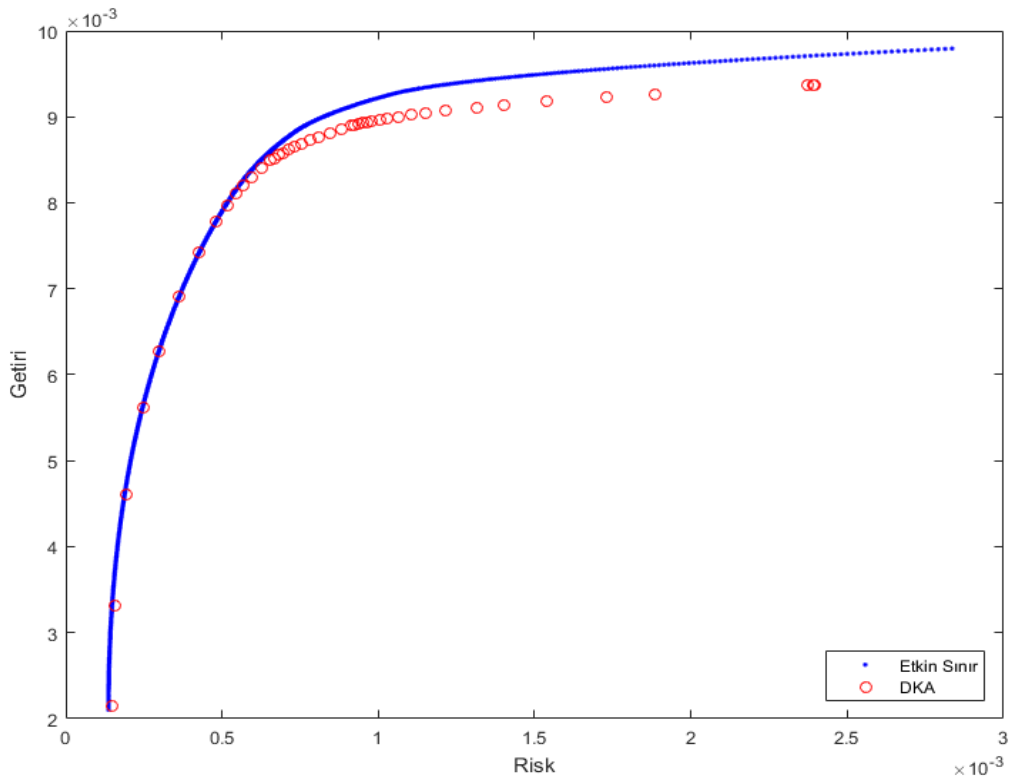
Denklem 35 ile verilen formülasyonun eleman sayısı kısıtlı portföy optimizasyonu probleminde, bilhassa yüksek λ değerlerinde Nelder-Mead algoritmasının çok üstün bir performans göstermesini sağladığı görülmüştür.

4.3.2. DKA Algoritması İçin Deneysel Sonuçlar

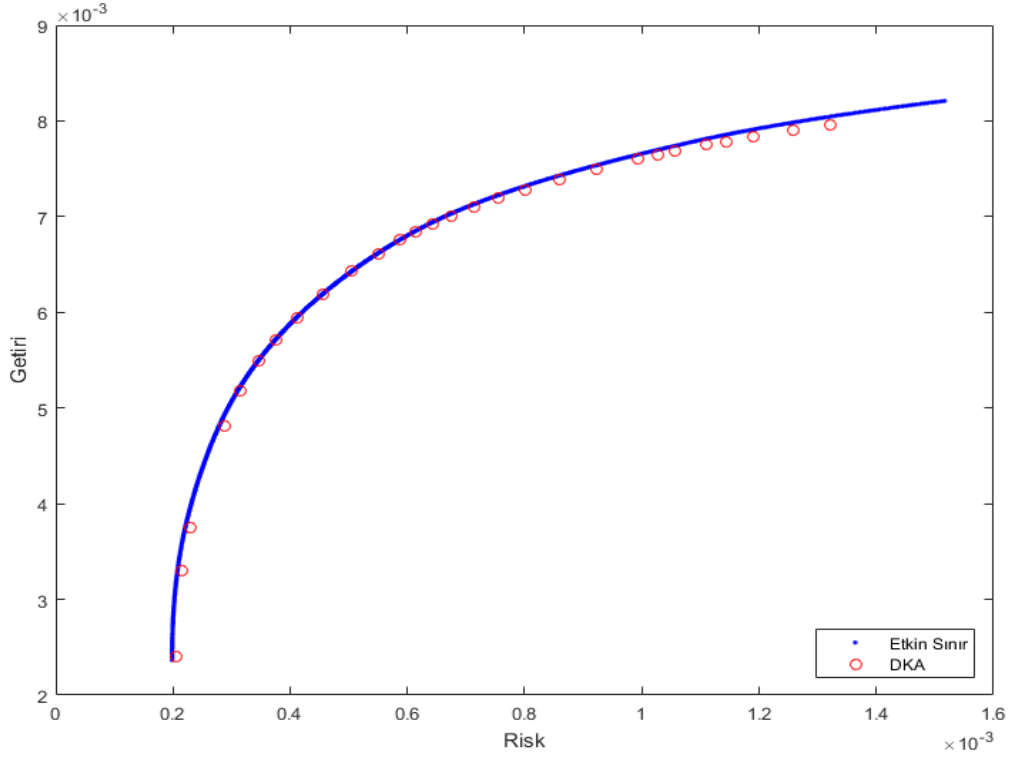
Tablo 18'de DKA algoritmasının bütün veri setleri için performans sonuçları verilmiştir. DKA algoritması ile çözülen kısıtlı etkin sınırlar: Hang Seng Şekil 47'de, DAX 100 Şekil 48'de, FTSE 100 Şekil 49'da, S&P 100 Şekil 50'de, Nikkei Şekil 51'de, BİST 30 Şekil 52'de ve BİST 100 Şekil 53'de verilmiştir.



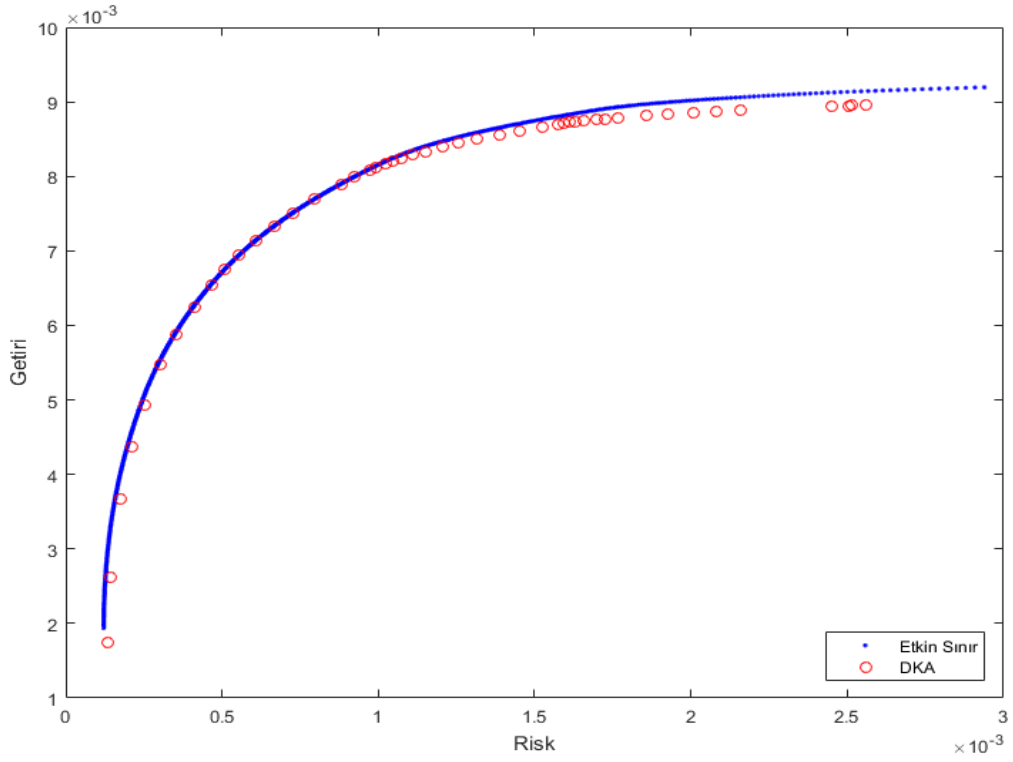
Şekil 47: Hang Seng veri seti için DKA algoritması ile çizilen etkin sınır



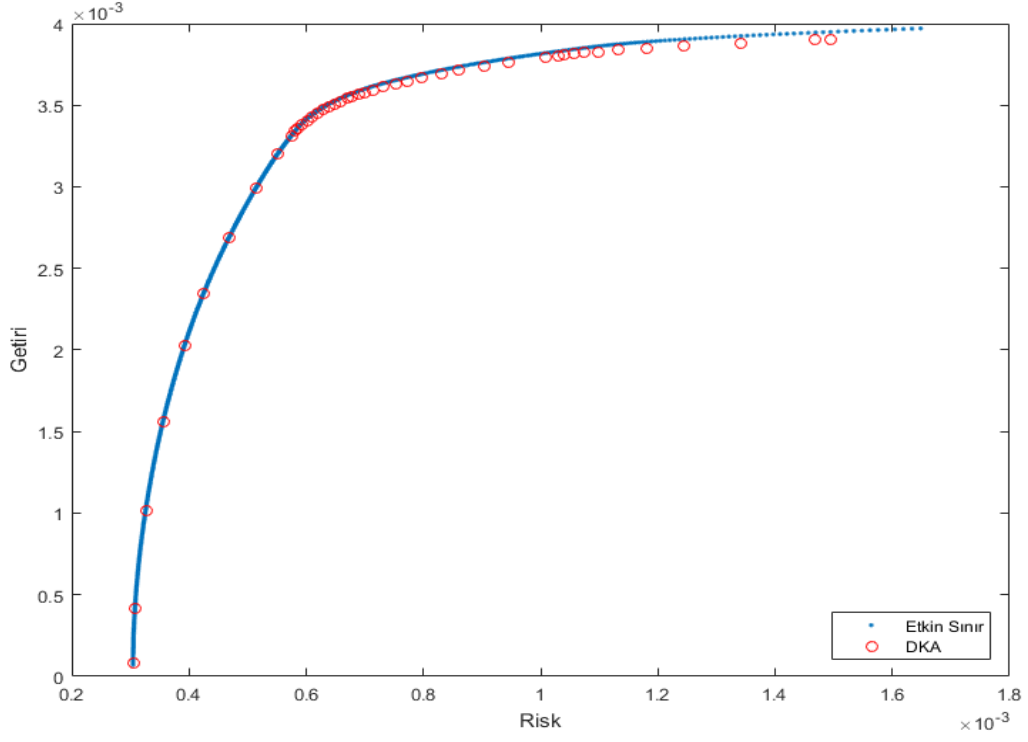
Şekil 48: DAX 100 veri seti için DKA algoritması ile çizilen etkin sınır



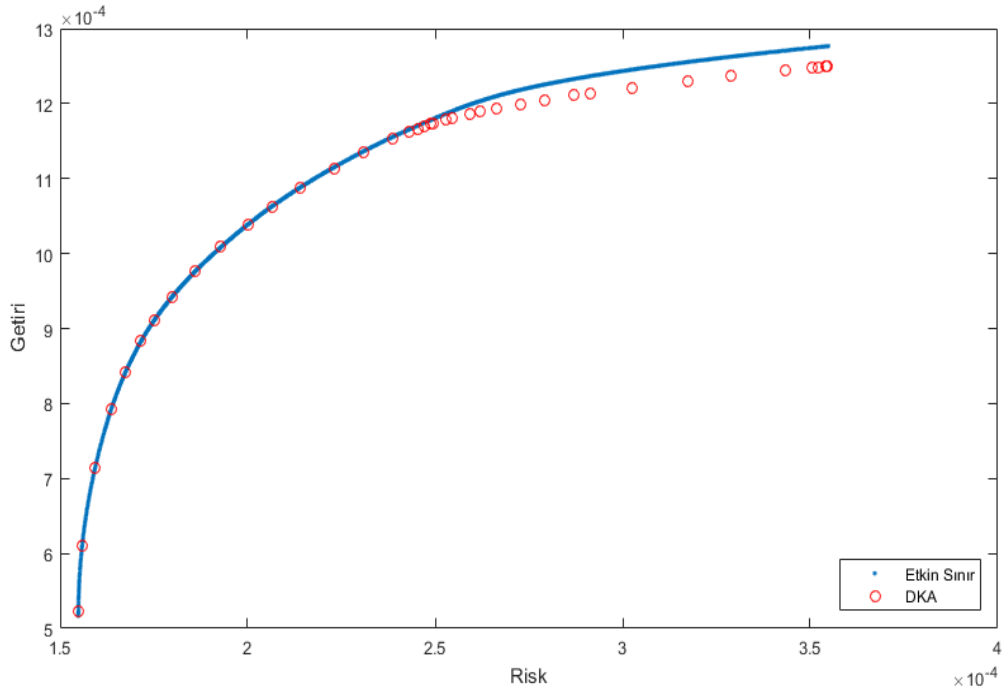
Şekil 49: FTSE 100 veri seti için DKA algoritması ile çizilen etkin sınır



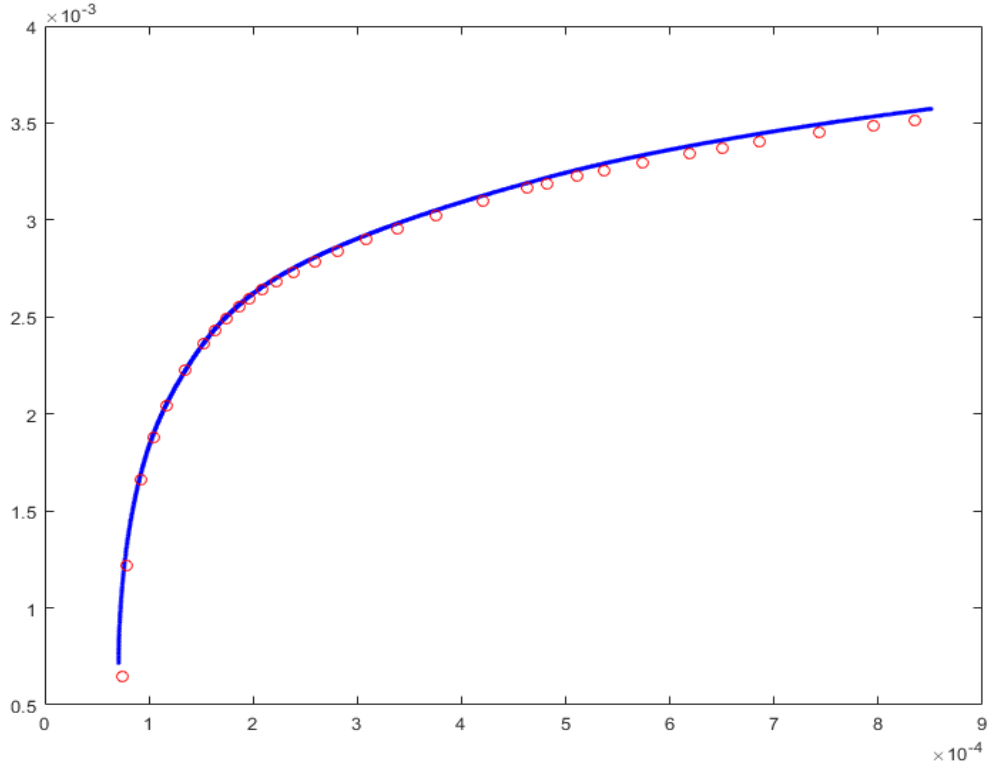
Şekil 50: S&P 100 veri seti için DKA algoritması ile çizilen etkin sınır



Şekil 51: Nikkei veri seti için DKA algoritması ile çizilen etkin sınır



Şekil 52: BİST 30 veri seti için DKA algoritması ile çizilen etkin sınır



Şekil 53: BİST 100 veri seti için DKA algoritması ile çizilen etkin sınır

Tablo 18: DKA algoritması için performans sonuçları

İndeks	N	Performans Ölçütü	DKA
Hang Seng	31	YHO	1.0990
		YHMED	1.2240
		YHMİN	0.0000
		YHMAKS	1.5570
		GHV-I	4.1420
		OGH-I	1.1400
		OÖÜ	0.0000
		GHV-II	1.6270
DAX 100	85	YHO	2.3199
		YHMED	2.5715
		YHMİN	0.0023
		YHMAKS	4.0261
		GHV-I	32.7536
		OGH-I	3.1608
		OÖÜ	0.0001
		GHV-II	6.7702
FTSE 100	89	YHO	1.2768
		YHMED	0.8438
		YHMİN	1.0842
		YHMAKS	0.0047
		GHV-I	2.1166
		OGH-I	4.5359
		OÖÜ	1.5302
		GHV-II	0.0000
S&P 100	98	YHO	2.4232
		YHMED	0.3243
		YHMİN	1.2898
		YHMAKS	1.1464
		GHV-I	0.0000
		OGH-I	5.4340
		OÖÜ	11.7266
		GHV-II	2.4611
Nikkei	225	YHO	0.0001
		YHMED	2.4744
		YHMİN	0.7041
		YHMAKS	0.5821
		GHV-I	0.5853
		OGH-I	0.0000
		OÖÜ	1.1810
		GHV-II	5.5870
Bist30	30	YHO	1.6960
		YHMED	0.0000
		YHMİN	0.8557
		YHMAKS	0.4216
		GHV-I	1.1240
		OGH-I	1.2740
		OÖÜ	0.0004
		GHV-II	2.1210
Bist100	99	YHO	6.6010
		YHMED	1.2520
		YHMİN	0.0000
		YHMAKS	1.9230
		GHV-I	6.6010
		OGH-I	1.2520
		OÖÜ	0.0000
		GHV-II	1.9230
Bist100	99	YHO	0.7460
		YHMED	1.0102
		YHMİN	1.1492
		YHMAKS	0.0184
		GHV-I	1.7391
		OGH-I	5.5230
		OÖÜ	1.1828
		GHV-II	0.0000
Bist100	99	YHO	2.4025
		YHMED	0.5104
		YHMİN	1.0102
		YHMAKS	1.1492
		GHV-I	0.0184
		OGH-I	1.7391
		OÖÜ	5.5230
		GHV-II	1.1828
Bist100	99	YHO	0.5104
		YHMED	1.0102
		YHMİN	1.1492
		YHMAKS	0.0184
		GHV-I	1.7391
		OGH-I	5.5230
		OÖÜ	1.1828
		GHV-II	0.0000
Bist100	99	YHO	2.4025
		YHMED	0.5104
		YHMİN	1.0102
		YHMAKS	1.1492
		GHV-I	0.0184
		OGH-I	1.7391
		OÖÜ	5.5230
		GHV-II	1.1828

4.4. Karşılaştırmalı sonuçlar

4.4.1. Proje kapsamında geliştirilen algoritmaların hata performansı karşılaştırılması

Proje kapsamında geliştirilen algoritmaların hata performansı karşılaştırılması Tablo 19'da verilmiştir. Bu sonuçlara göre, çözüm performans bakımından her bir algoritmanın birbirine yakın performans sergilediği gözlemlenmiştir. Ancak, DKA algoritmasının çözüm süresi bakımından YAK ve KKO algoritmalarına göre çok daha iyi bir performans verdiği rahatlıkla söylenebilir.

Tablo 19: Proje kapsamında geliştirilen algoritmaların hata performansı karşılaştırılması

İndeks	N	Performans Ölçütü	YAK	KKO	DKA		
Hang Seng	31	YHO	1.0953	1.0960	1.0990		
		YHMED	1.2181	1.2080	1.2240		
		YHMİN	0.0000	0.0000	0.0000		
		YHMAKS	1.5538	1.554	1.5570		
		GHV-I	4.1302	4.1310	4.1220		
		OGH-I	1.1364	1.1370	1.1400		
		OÖÜ	0.0001	0.0000	0.0000		
		GHV-II	1.6432	1.6350	1.6270		
		OGH-II	0.6047	0.6082	0.6122		
		Süre	56	99	16		
		DAX 100	85	YHO	2.3295	2.3298	2.3199
				YHMED	2.5678	3.5715	2.5715
				YHMİN	0.0023	0.0022	0.0023
YHMAKS	4.0275			4.0260	4.0261		
GHV-I	32.6157			32.7535	32.7536		
OGH-I	3.0743			3.1607	3.1605		
OÖÜ	0.0001			0.0001	0.0001		
GHV-II	6.7925			6.7701	6.7702		
OGH-II	1.2761			1.2767	1.2768		
Süre	135			210	41		
FTSE 100	89			YHO	0.8481	0.8400	0.8438
				YHMED	1.0841	1.0840	1.0842
				YHMİN	0.0047	0.0047	0.0047
		YHMAKS	2.0638	2.1165	2.1166		
		GHV-I	4.5658	4.5600	4.5359		
		OGH-I	1.4328	1.5370	1.5302		
		OÖÜ	0.0000	0.0000	0.0000		
		GHV-II	2.4397	2.4329	2.4232		
		OGH-II	0.3255	0.3259	0.3243		
		Süre	109	205	31		
		S&P 100	98	YHO	1.2930	1.2850	1.2898
				YHMED	1.1369	1.1377	1.1464
				YHMİN	0.0000	0.0000	0.0000
YHMAKS	5.4422			5.4340	5.4340		
GHV-I	11.5133			11.8786	11.7266		
OGH-I	2.3309			2.6518	2.4611		
OÖÜ	0.0001			0.0001	0.0001		
GHV-II	2.5260			2.3956	2.3744		
OGH-II	0.8885			0.6908	0.7041		
Süre	160			312	51		
Nikkei	225			YHO	0.5781	0.5844	0.5821
				YHMED	0.5856	0.5894	0.5853
				YHMİN	0.0000	0.0000	0.0000
		YHMAKS	1.1606	1.1680	1.1810		
		GHV-I	5.5757	5.6280	5.5870		
		OGH-I	1.6764	1.7820	1.6660		
		OÖÜ	0.0000	0.0000	0.0000		
		GHV-II	0.8396	0.8385	0.8557		
		OGH-II	0.4127	0.4229	0.4216		
		Süre	410	773	79		
		Bist30	30	YHO	1.1256	1.1210	1.1240
				YHMED	1.2549	1.3150	1.2740
				YHMİN	0.0012	0.0004	0.0004
YHMAKS	2.1286			2.0840	2.1210		
GHV-I	6.5390			6.5930	6.5010		
OGH-I	1.2537			1.2500	1.2520		
OÖÜ	0.0000			0.0000	0.0000		
GHV-II	1.9324			1.9220	1.9210		
OGH-II	0.7358			0.7438	0.7460		
Süre	50			98	14		
Bist100	99			YHO	1.0272	1.0438	1.0102
				YHMED	1.1586	1.2067	1.1492
				YHMİN	0.0132	0.0185	0.0184
		YHMAKS	1.7384	1.8538	1.7481		
		GHV-I	5.6169	5.6353	5.6230		
		OGH-I	2.0126	2.0626	1.1828		
		OÖÜ	0.0000	0.0000	0.0000		
		GHV-II	2.4973	2.5234	2.4025		
		OGH-II	0.7306	0.7404	0.5104		
		Süre	170	311	53		

4.4.2. Literatürdeki diğer yöntemlerle hata performansı karşılaştırması

Literatürdeki diğer yöntemlerle hata performansı karşılaştırmaları çeşitli hata hesaplama formülleri cinsinden sırasıyla Tablo 20, Tablo 21 ve Tablo 22'de verilmiştir. Bu sonuçlara göre,

proje kapsamında geliştirilen algoritmaların literatürdeki diğer algoritmalarla rekabet edebilir düzeyde performans gösterdiklerini ortaya koymuştur.

Tablo 20: YHO, YHMED, YHMİN, YHMAKS hataları için performans karşılaştırması

İndeks	N	Performans ölçütü	GRASP-								
			GA	TS	SA	PSO	PBILD	QP	YAK	KKO	DKA
			(Chang vd. 2000)	(Chan g vd. 2000)	(Chan g vd. 2000)	(Deng vd. 2012)	E (Lwin ve Qu 2013)	(Baykasoğlu u vd. 2016; Baykasoğlu vd. 2015)			
Hang Seng	31	YHO	1.0974	1.1217	1.0957	1.0953	1.1431	1.0965	1.0953	1.0960	1.0990
		YHMED	1.2181	1.2181	1.2181	-	1.2390	1.2155	1.2181	1.2080	1.2240
		YHMİN	-	-	-	-	-	0.0000	0.0000	0.0000	0.0000
		YHMAKS	-	-	-	-	-	1.5538	1.5538	1.554	1.5570
		Süre(s)	172	74	79	-	-	62	56	99	16
DAX 100	85	YHO	2.5424	3.3049	2.9297	2.5417	2.4251	2.3126	2.3295	2.3298	2.3199
		YHMED	2.5466	2.6380	2.5661	-	2.5866	2.5630	2.5678	3.5715	2.5715
		YHMİN	-	-	-	-	-	0.0059	0.0023	0.0022	0.0023
		YHMAKS	-	-	-	-	-	4.0275	4.0275	4.0260	4.0261
		Süre(s)	544	199	210	-	-	139	135	210	41
FTSE 100	89	YHO	1.1076	1.6080	1.4623	1.0628	0.9706	0.8451	0.8481	0.8400	0.8438
		YHMED	1.0841	1.0841	1.0841	-	1.0840	1.0841	1.0841	1.0840	1.0842
		YHMİN	-	-	-	-	-	0.0016	0.0047	0.0047	0.0047
		YHMAKS	-	-	-	-	-	2.0576	2.0638	2.0665	2.0666
		Süre(s)	573	246	215	-	-	143	109	205	31
S&P 100	98	YHO	1.9328	3.3092	3.0696	1.6890	1.6386	1.2937	1.2930	1.2850	1.2898
		YHMED	1.2244	1.2882	1.1823	-	1.1692	1.1420	1.1369	1.1377	1.1464
		YHMİN	-	-	-	-	-	0.0009	0.0000	0.0000	0.0000
		YHMAKS	-	-	-	-	-	5.4551	5.4422	5.4340	5.4340
		Süre(s)	638	225	242	-	-	172	160	312	51
Nikkei	225	YHO	0.7961	0.8975	0.6066	0.6870	0.5972	0.5782	0.5781	0.5844	0.5821
		YHMED	0.6133	0.6093	0.6732	-	0.5896	0.5857	0.5856	0.5894	0.5853
		YHMİN	-	-	-	-	-	0.0000	0.0000	0.0000	0.0000
		YHMAKS	-	-	-	-	-	1.1606	1.1606	1.1680	1.1810
		Süre(s)	1964	545	553	-	-	438	410	773	79
Ortalama		YHO	1.4952	2.0483	1.8328	1.4152	1.3549	1.2252	1.2280	1.2270	1.2269
		YHMED	1.3373	1.3675	1.3448	-	1.3337	1.3181	1.3185	1.5181	1.3222
		YHMİN	-	-	-	-	-	0.0017	0.0014	0.0013	0.0014
		YHMAKS	-	-	-	-	-	2.8509	2.8495	2.8497	2.8529
		Süre(s)	778	257	260	-	-	191	174	320	44

Tablo 21: GHV-I, OGH-I hata performansı karşılaştırması

İndeks	N	Performans Ölçütü	GRASP-QP		YAK	KKO	DKA
			NN (Fernández ve Gómez 2007)	(Baykasoğlu vd. 2016; Baykasoğlu vd. 2015)			
Hang Seng	31	GHV-I	4.1039	4.1341	4.1302	4.1310	4.1220
		OGH-I	1.4530	1.1380	1.1364	1.1370	1.1400
		Süre(s)	390	62	56	99	16
DAX 100	85	GHV-I	12.5914	32.7815	32.6157	32.7535	32.7536
		OGH-I	2.2060	3.0504	3.0743	3.1607	3.1605
		Süre(s)	1069	139	135	210	41
FTSE 100	89	GHV-I	4.4663	4.5594	4.5658	4.5600	4.5359
		OGH-I	1.9636	1.5188	1.4328	1.5370	1.5302
		Süre(s)	1106	143	109	205	31
S&P 100	98	GHV-I	8.3811	12.0795	11.5133	11.8786	11.7266
		OGH-I	2.6816	1.5599	2.3389	2.6518	2.4611
		Süre(s)	1211	172	160	312	51
Nikkei	225	GHV-I	6.5924	5.5753	5.5757	5.6280	5.5870
		OGH-I	3.1050	1.6595	1.6764	1.7820	1.6660
		Süre(s)	2827	438	410	773	79
Ortalama		GHV-I	7.2270	11.8260	11.6801	11.7902	11.7450
		OGH-I	2.2818	1.7853	1.9358	2.0537	1.9915
		Süre(s)	1321	191	174	320	44

Tablo 22: OÖU, GHV-II, OGH-II hata performansı karşılaştırması

Index	N	Performans Ölçütü	PSO (Cura 2009)	PSO-SA (Mozafari vd. 2011)	PSO-HNN (Sadigh vd. 2012)	GRASP-QP (Baykasoğlu vd. 2015)	ABC-FA (Tuba ve Bacanin 2014a)			
							YAK	KKO	DKA	
		OÖU	0.0049	0.0001	0.0001	0.0001	0.0004	0.0001	0.0000	0.0000
Hang Seng	3	GHV-II	2.2421	1.6388	2.5908	1.6400	1.3952	1.6432	1.6350	1.6270
		OGH-II	0.7427	0.6059	0.7335	0.6060	0.5289	0.6047	0.6082	0.6122
		Süre(s)	34	-	-	62	12	56	99	16
DAX 100	8	OÖU	0.0090	0.0001	0.0000	0.0001	0.0009	0.0001	0.0001	0.0001
		GHV-II	6.8588	6.7806	5.7585	6.7593	7.2649	6.7925	6.7701	6.7702
		OGH-II	1.5885	1.2770	0.1466	1.2769	1.3523	1.2761	1.2767	1.2768
		Süre(s)	179	-	-	139	62	135	210	41
FTSE 100	8	OÖU	0.0022	0.0000	0.0000	0.0000	0.0003	0.0000	0.0000	0.0000
		GHV-II	3.0596	2.4701	5.4141	2.4350	2.6721	2.4397	2.4329	2.4232
		OGH-II	0.3640	0.3247	0.3095	0.3245	0.3187	0.3255	0.3259	0.3243
		Süre(s)	190	-	-	143	76	109	205	31
S&P 100	9	OÖU	0.0052	0.0001	0.0000	0.0001	0.0001	0.0001	0.0001	0.0001
		GHV-II	3.9136	2.6281	5.1456	2.5211	3.7598	2.5260	2.3956	2.3744
		OGH-II	1.4040	0.7846	0.2925	0.9063	0.9529	0.8885	0.6908	0.7041
		Süre(s)	214	-	-	172	125	160	312	51
Nikkei	2	OÖU	0.0019	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
		GHV-II	2.4274	0.9583	4.7779	0.8359	1.6982	0.8396	0.8385	0.8557
		OGH-II	0.7997	1.7090	0.7040	0.4184	0.6719	0.4127	0.4229	0.4216
		Süre(s)	919	-	-	438	329	410	773	79
Ortalama		OÖU	0.0046	0.0001	0.0000	0.0001	0.00034	0.0001	0.0000	0.0000
		GHV-II	3.7003	2.8952	4.7374	2.8383	3.35804	2.8482	2.8144	2.8101
		OGH-II	0.9798	0.9402	0.4372	0.7064	0.76494	0.7015	0.6649	0.6678
		Süre(s)	307	-	-	191	120.8	174	320	44

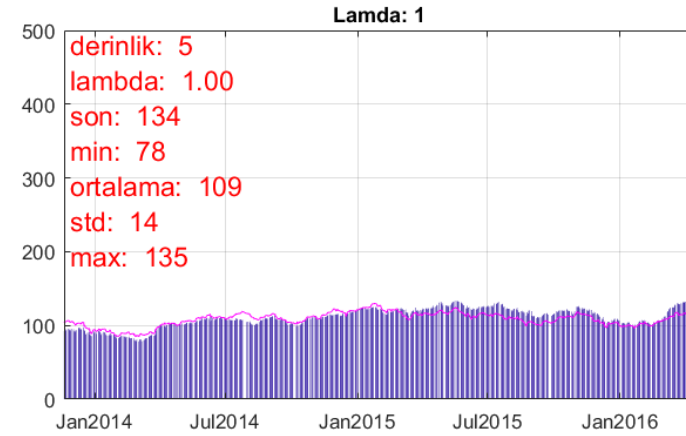
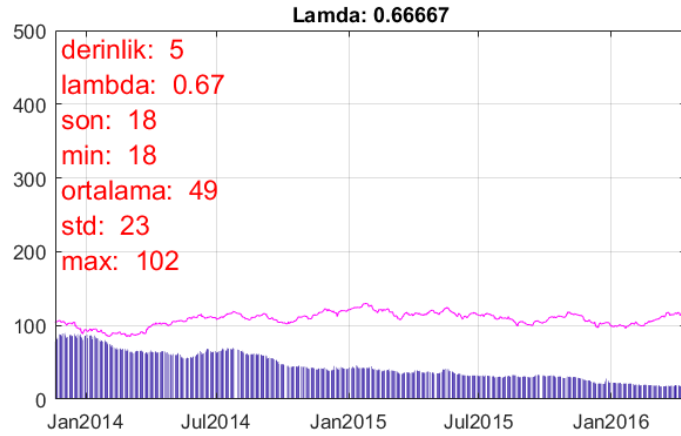
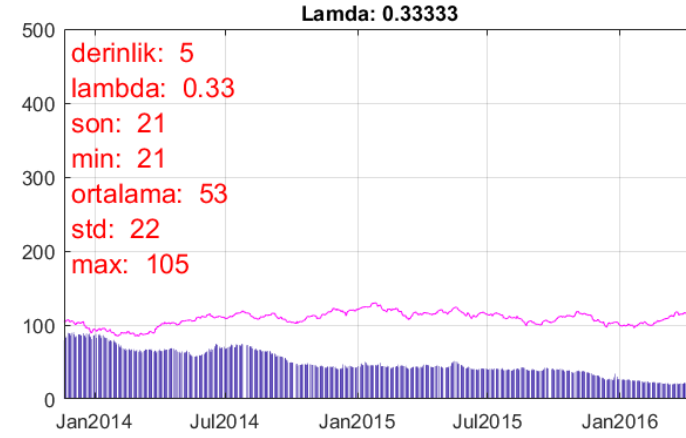
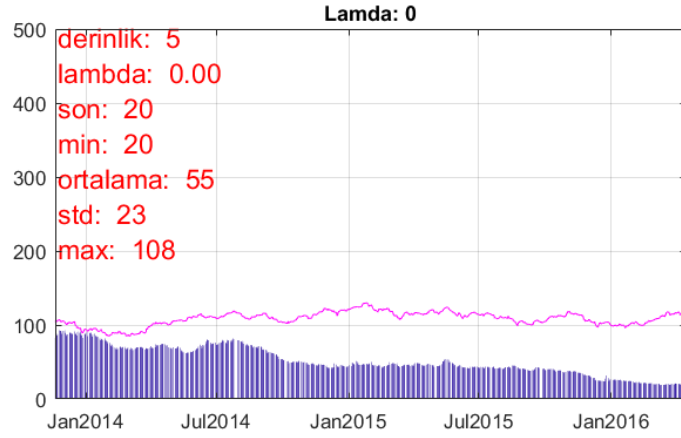
4.5. Borsa İstanbul Getiri Analizi

Geliştirilen algoritmalar (YAK, KKO veya DKA) ile BİST 30 ve BİST 100 veri setleri üzerinde farklı λ (lamda) değerleri ile kullanılan geçmiş veri derinliğini değiştirerek anapara değişimi izlenmiştir.

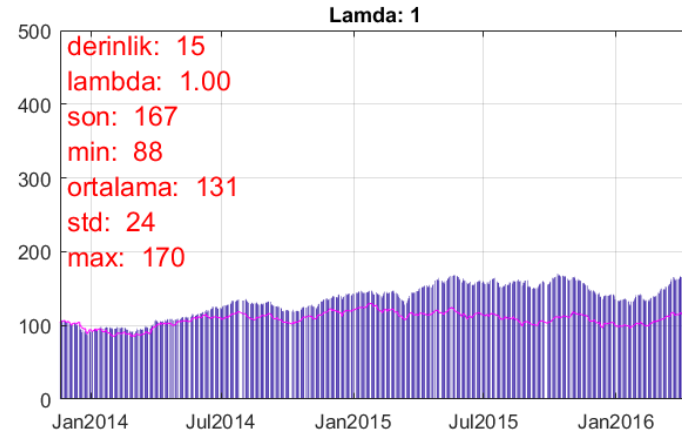
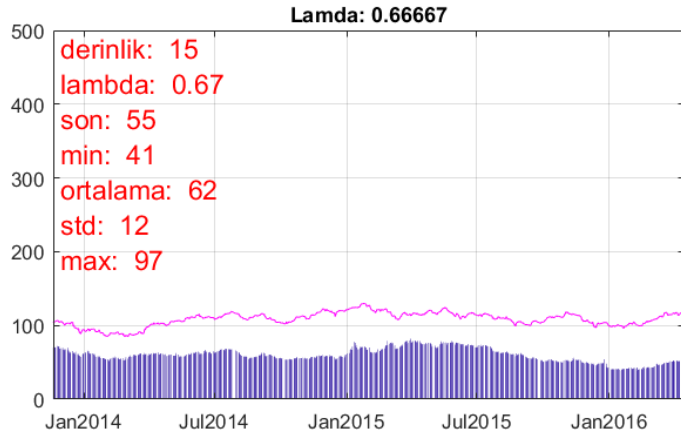
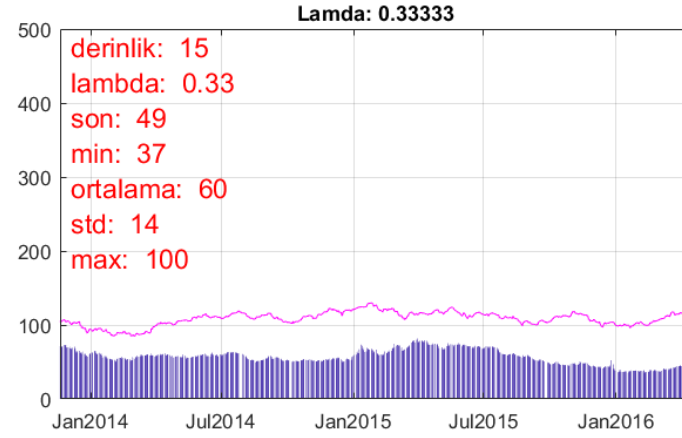
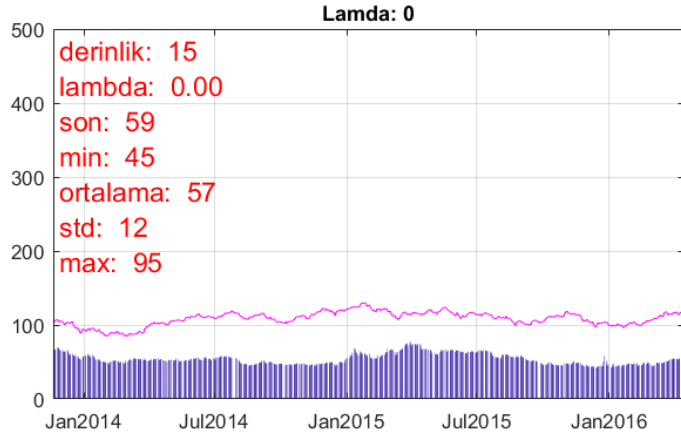
Anaparanın başlangıçta 100 birim olduğu kabul edilerek, hali hazırda Bloomberg veri tabanından indirilmiş olan Borsa İstanbul verilerini kullanarak, deneyler ve analizler 23.05.2013 ile 18.04.2016 tarihleri arasındaki BIST 30 ve BIST 100 verileri kullanılarak yapılmıştır. Derinlik testlerin eşit şartlarda kıyaslanabilmesi amacıyla 23.05.2013 tarihinden 01.01.2014 tarihine kadar olan veriler test verisi olarak korunmuş, simülasyon Ocak 2014'den itibaren başlatılmıştır. Bu tarihten itibaren, simülasyondan elde edilen günlük BIST 30 ve BIST 100 endekslerine yönelik yatırım önerileri gerçekleştirildiğinde, anaparanın kümülatif olarak değer artışları gözlemlenmiştir. Simülasyonda kullanılan algoritmanın günlük yatırım önerileri, veri derinliği olarak belirlenen gün sayısı kadar geçmiş veriyi girdi olarak kabul etmekte ve bu verilere göre elde edilen portföy optimizasyon sonuçlarını yansıtmaktadır. Yapılan yatırım önerileri ile o gün için belirlenen hisseler belirlenen oranlarda sabah borsa açılış saati ile birlikte yatırım yapıldığı ve akşam kapanış saati ile birlikte de satış yapıldığı varsayılmıştır. Ertesi gün, yatırıma elde kalan/ulaşılabilir tüm parayla devam edilmiştir. Şekil 54-Şekil 63 numaralı grafiklerde kırmızı renk ile verilen 'derinlik' değeri kaç günlük geçmiş veri kullanılarak portföy oluşturulduğunu, 'lambda' değeri yatırımda kullanılan risk seviyesini, 'son', 'min', 'ortalama', 'std', 'max' değerleri sırasıyla, simülasyon sonunda elde kalan anaparayı, simülasyon boyunca düştüğü en küçük değeri, ortalama değeri, standart sapmayı ve ulaştığı en büyük değeri göstermektedir. Bu grafiklerdeki X-ekseni anaparanın hangi gün hangi düzeyde olduğunu gösteren tarihleri vermektedir.

BIST 30 veri setleri üzerinde, 5 günlük, 15 günlük, 30 günlük, 45 günlük ve 60 günlük derinlik (geçmiş veri) kullanılarak yapılan yatırımların anapara üzerindeki etkileri sırasıyla; Şekil 54, Şekil 55, Şekil 56, Şekil 57 ve Şekil 58'de verilmiştir. Bu grafikler incelendiğinde, $\lambda = \{0, 0.33, 0.66\}$ değerleri için yatırımların başarısız olduğu gözlemlenmiştir. Ancak, $\lambda = 1$ değeri için, yani minimum risk seviyesinde anaparayı koruduğu, ayrıca, düşük derinlik seviyelerinde endeks ortalamasının üstünde bir getiri sağladığı gözlemlenmiştir. BIST 100 veri setleri üzerinde ise, 5 günlük, 15 günlük, 30 günlük, 45 günlük ve 60 günlük derinlik kullanılarak yapılan yatırımların anapara üzerindeki etkileri sırasıyla; Şekil 59, Şekil 60, Şekil 61, Şekil 62 ve Şekil 63'te verilmiştir. BIST 100 yatırımlarının anapara değişim grafiği incelendiğinde, $\lambda = 1$ değeri için, yani minimum risk seviyesinde BIST 30 endeksinde olduğu gibi, anaparayı koruduğu ve endeks ortalamasının üstünde bir getiri sağladığı gözlemlenmiştir. Diğer risk seviyelerinde ise, yer yer büyük kayıplar gözlemlenmiş olsa da getirinin belli periyotlarda çok yüksek olduğu görülebilmektedir. Bazı seviyelerde, örneğin $\lambda = 0.66$ ve 60 günlük veri

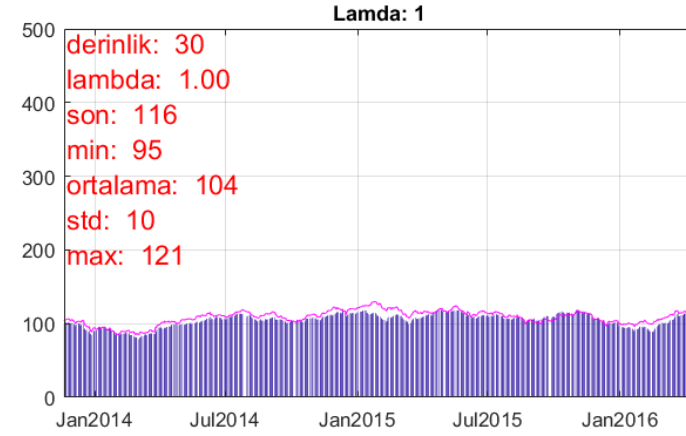
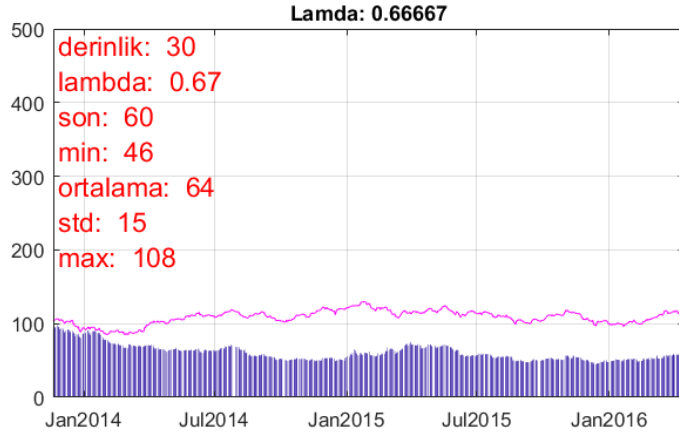
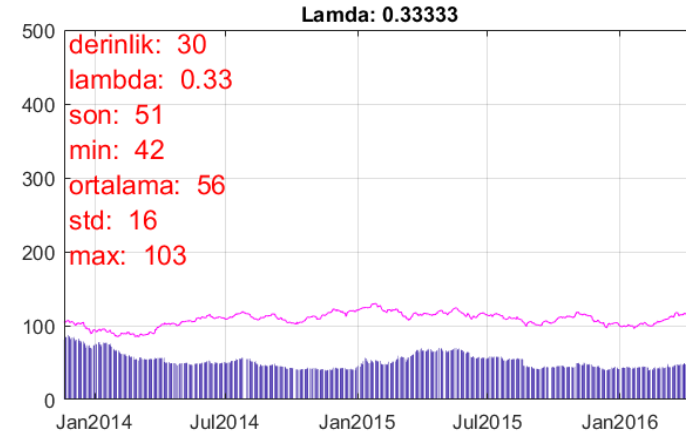
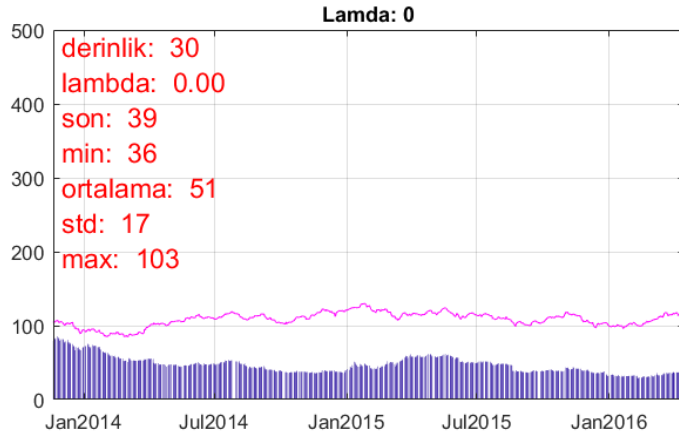
derinliğinde, anaparanın 8.6 katına kadar ulaşabildiği gözlemlenmiştir. Hem BIST 30, hem de BIST 100 grafiklerinde pembe renk ile verilen çizgi endeksin belirlenen tarihler arasındaki değişim grafiğini temsil etmektedir.



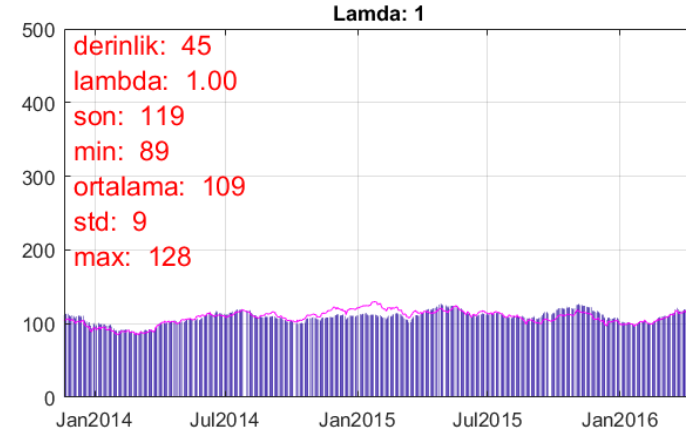
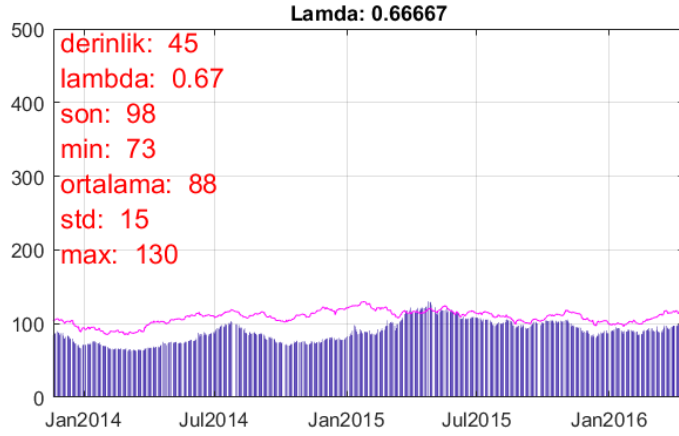
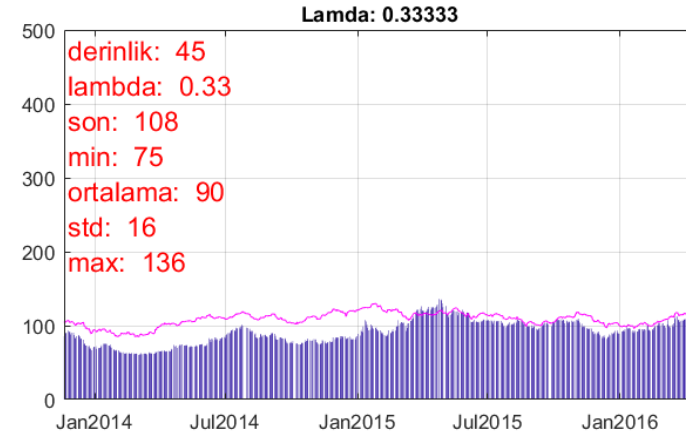
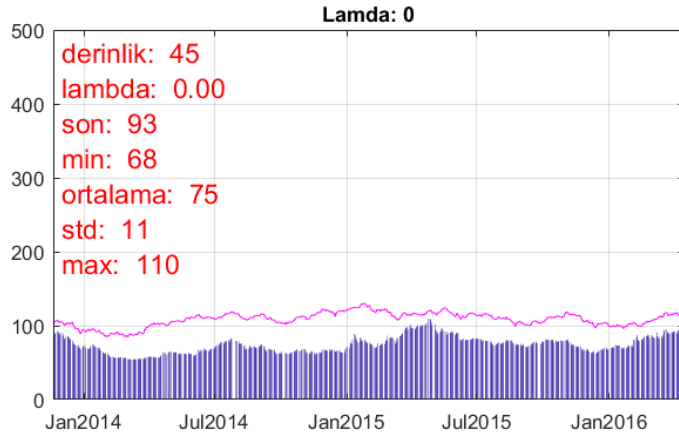
Şekil 54: Beş günlük geçmiş veriyle BİST 30 endeksi için anapara değişim grafiği



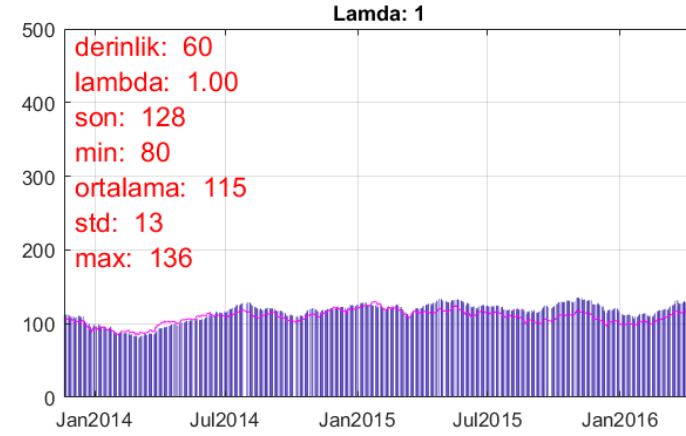
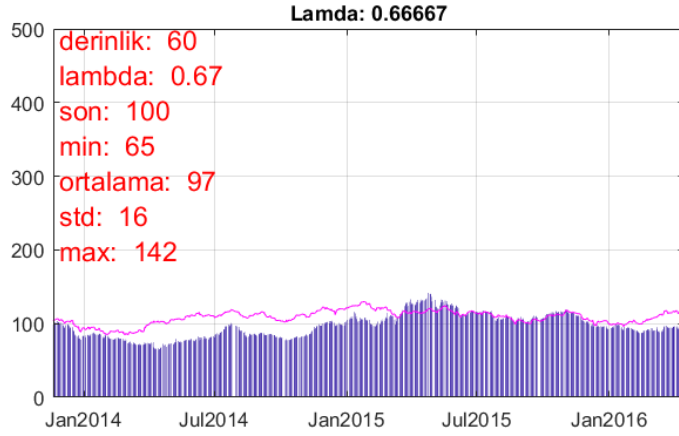
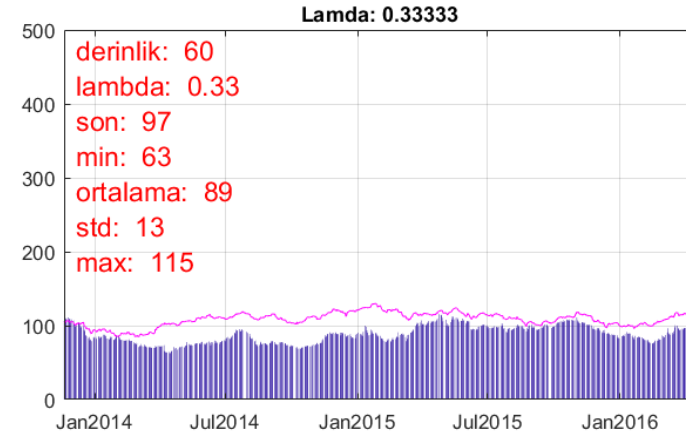
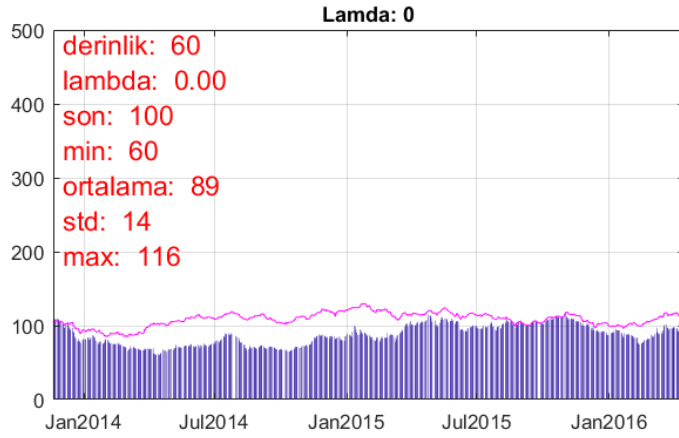
Şekil 55: On beş günlük geçmiş veriyle BİST 30 endeksi için anapara değişim grafiği



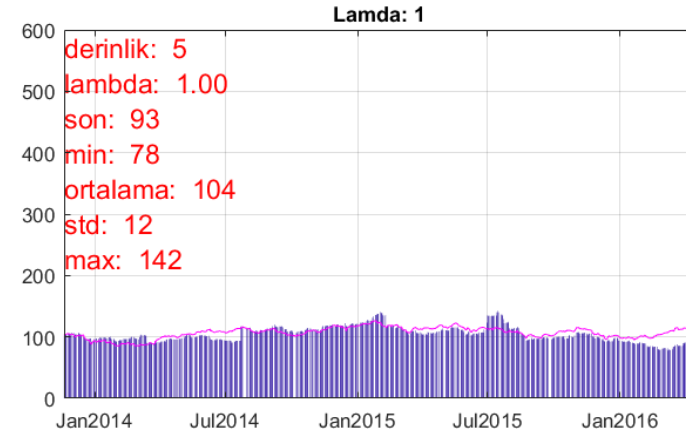
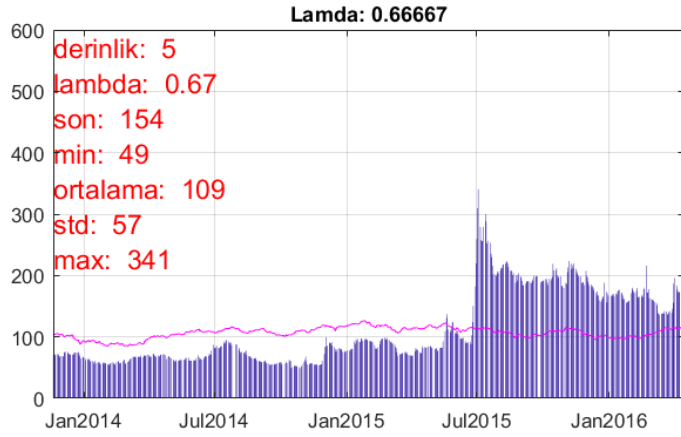
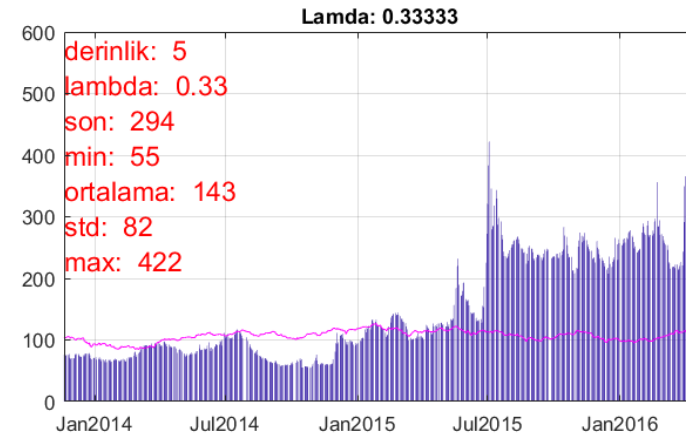
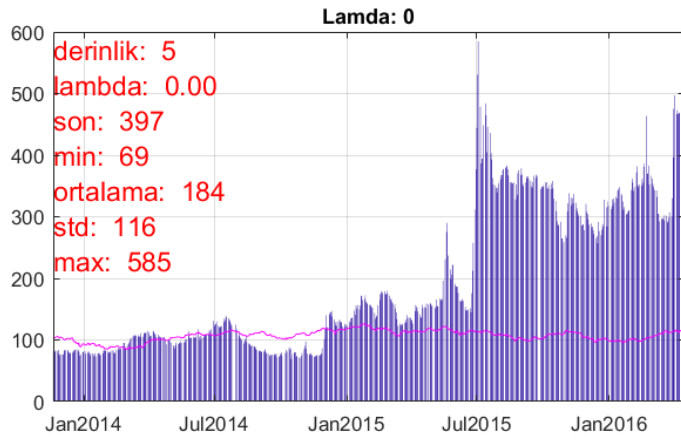
Şekil 56: Otuz günlük geçmiş veriyle BİST 30 endeksi için anapara değişim grafiği



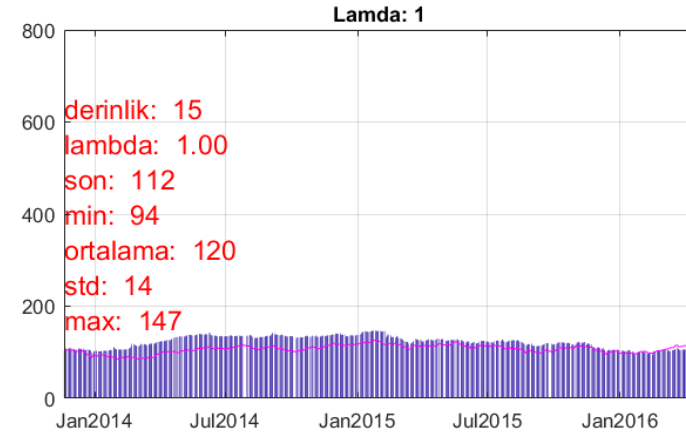
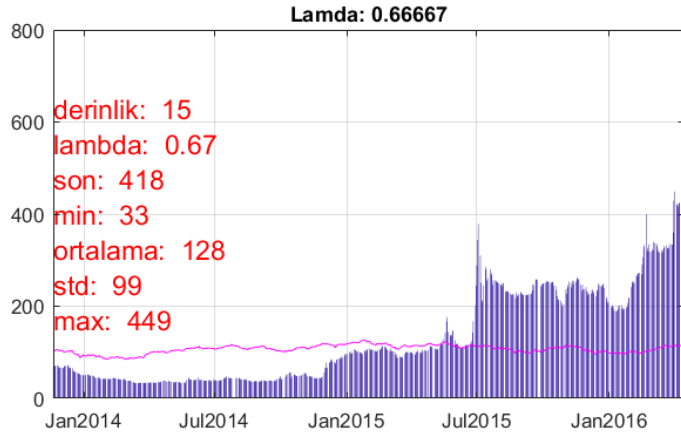
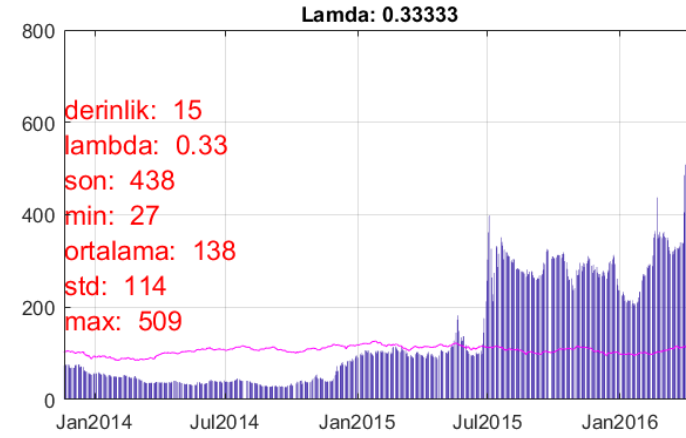
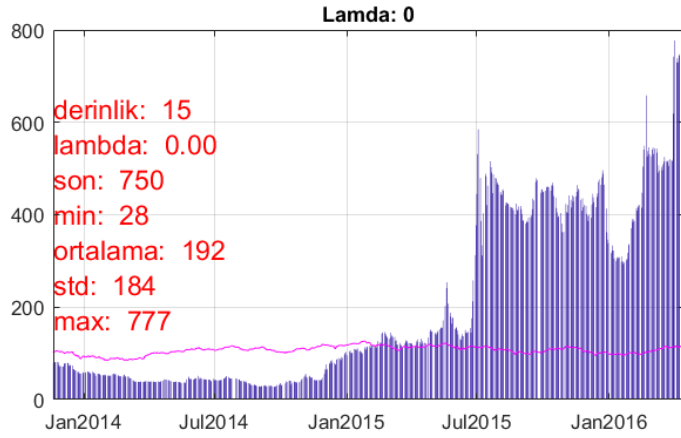
Şekil 57: Kırk beş günlük geçmiş veriyle BİST 30 endeksi için anapara değişim grafiği



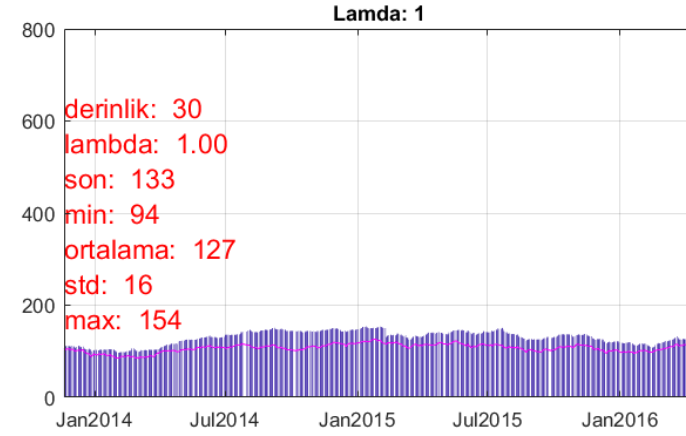
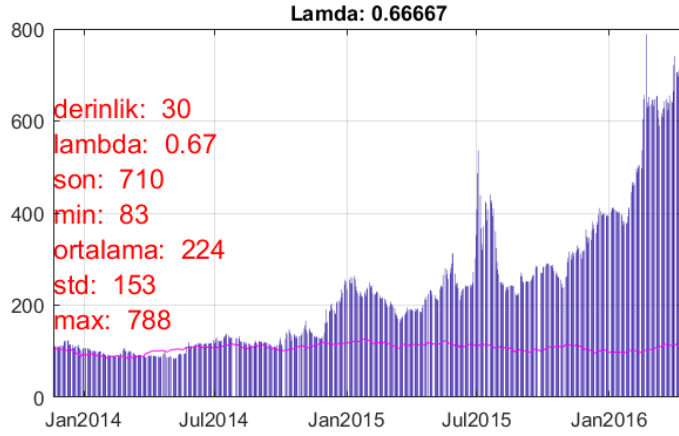
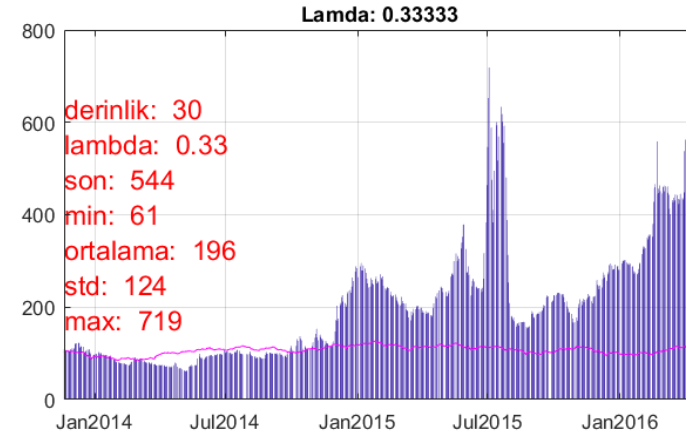
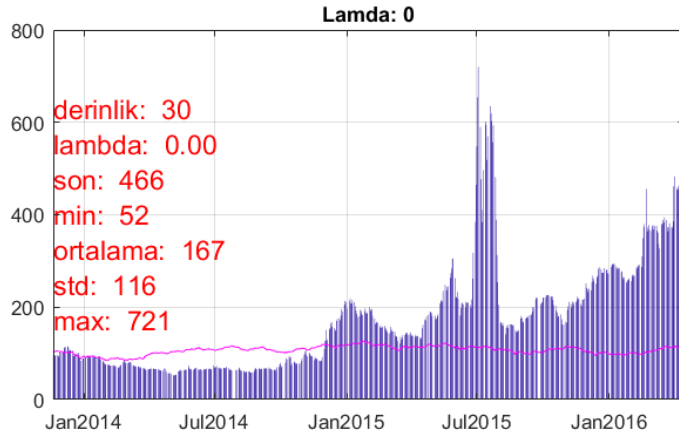
Şekil 58: Altmış günlük geçmiş veriyle BİST 30 endeksi için anapara değişim grafiği



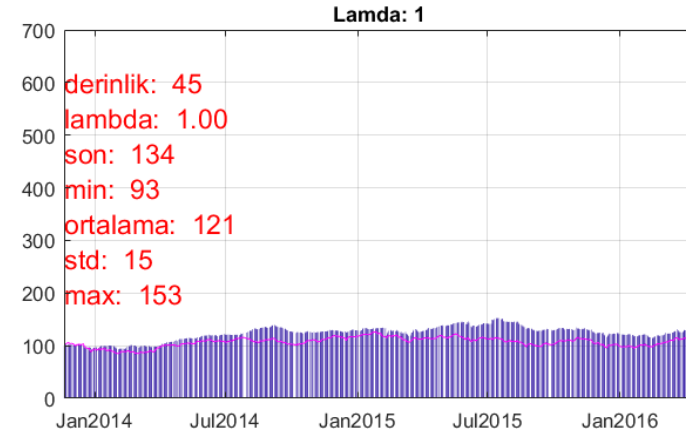
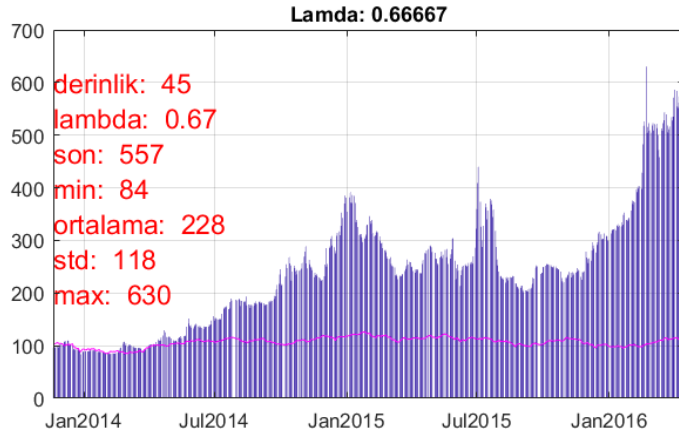
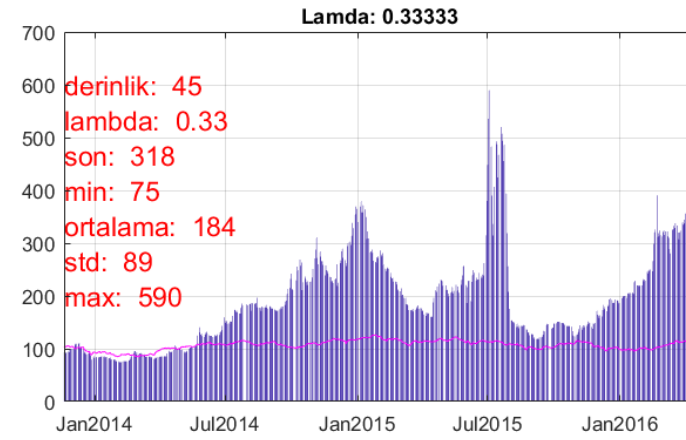
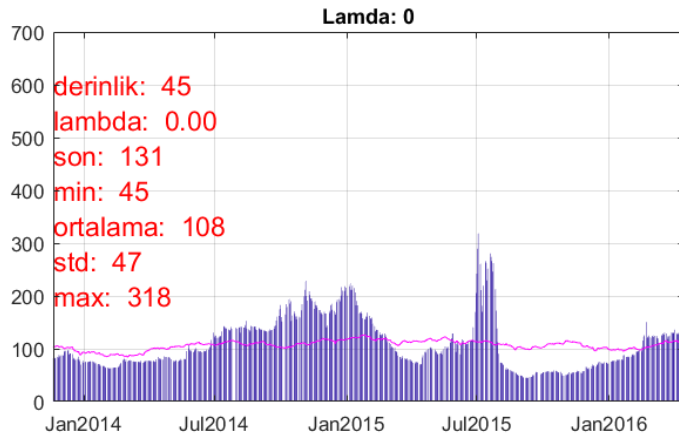
Şekil 59: Beş günlük geçmiş veriyle BİST 100 endeksi için anapara değişim grafiği



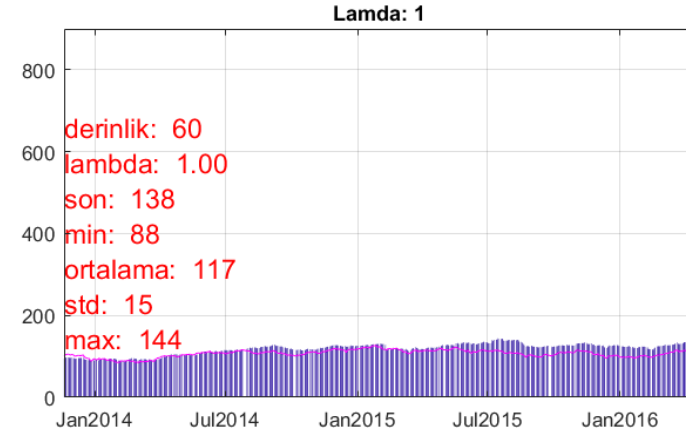
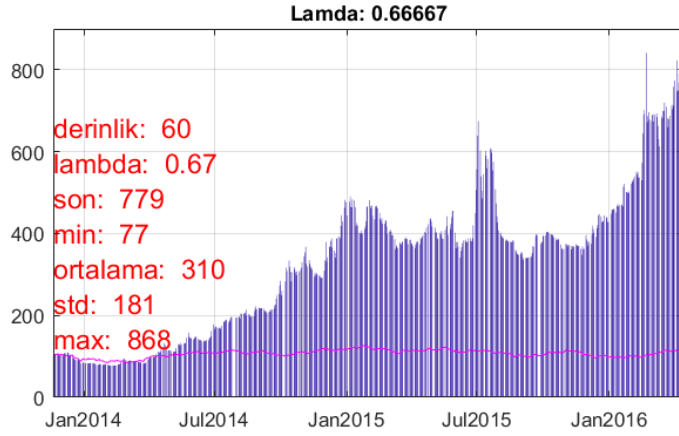
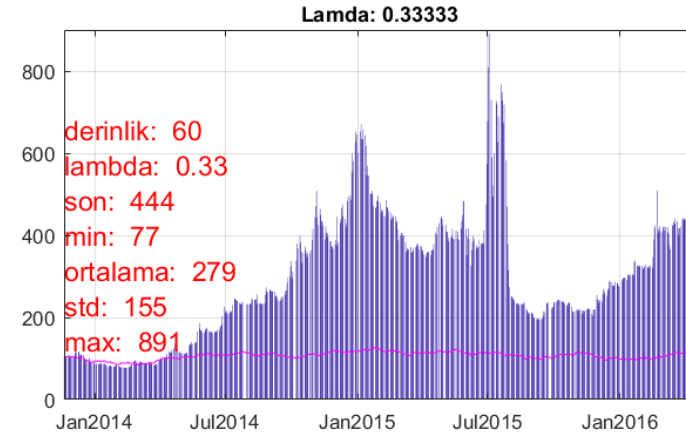
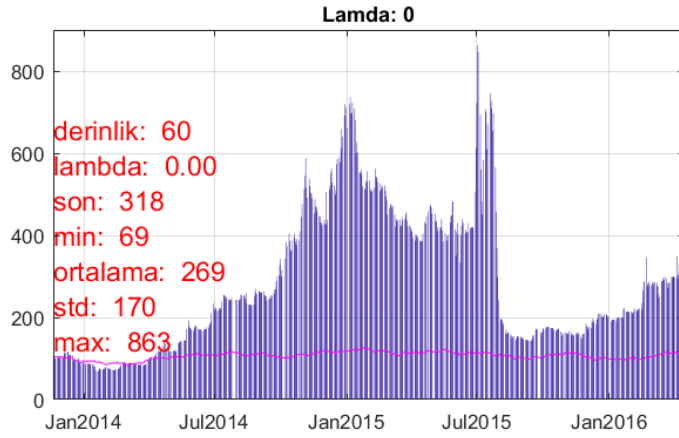
Şekil 60: On beş günlük geçmiş veriyle BİST 100 endeksi için anapara değişim grafiği



Şekil 61: Otuz günlük geçmiş veriyle BİST 100 endeksi için anapara değişim grafiği

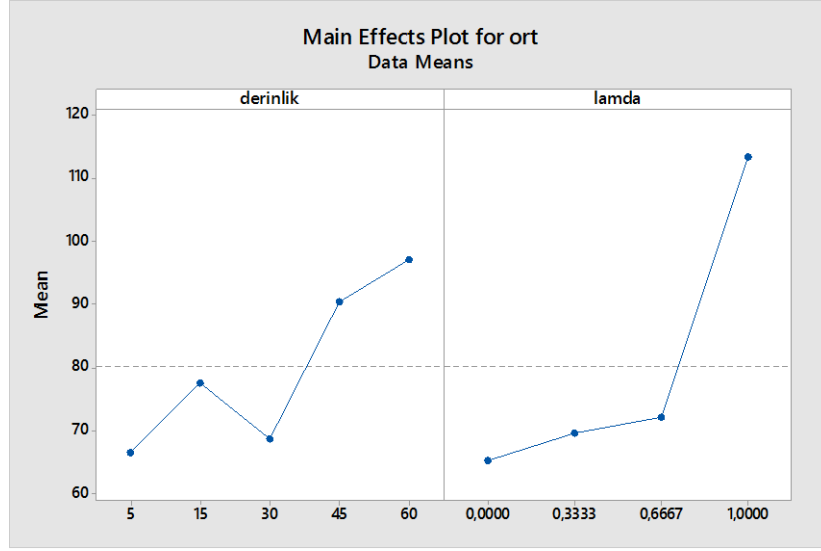


Şekil 62: Kırk beş günlük geçmiş veriyle BİST 100 endeksi için anapara değişim grafiği

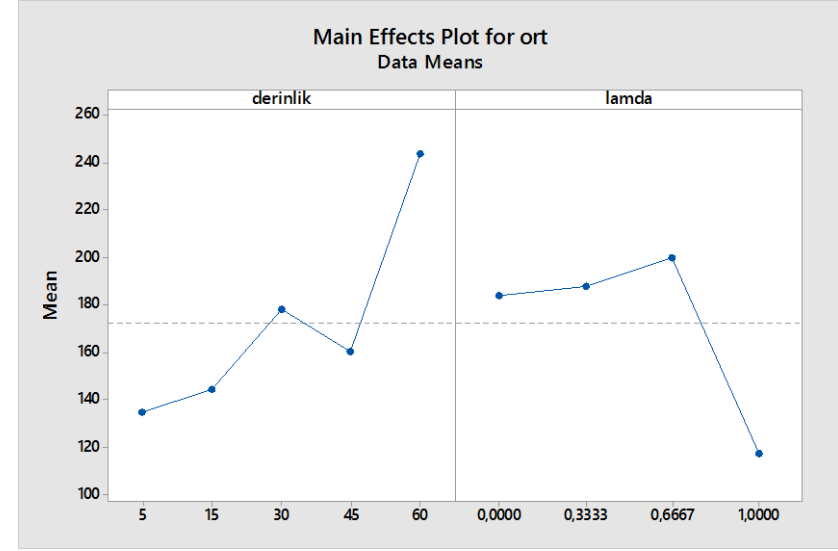


Şekil 63: Altmış günlük geçmiş veriyle BİST 100 endeksi için anapara değişim grafiği

BIST 30 endeksinde yapılan analizlere göre (Şekil 64), ortalamda en iyi getiriye 60 günlük derinlik ve minimum risk seviyesi ($\lambda = 1$) verirken, BIST 100 endeksinde ise (Şekil 65), 60 günlük derinlik ve düşük risk seviyesi ($\lambda = 0.66$) vermiştir. BIST 100 endeksindeki yatırımların BIST 30 endeksinde göre ortalamda daha fazla getiri elde ettiği söylenebilir. Minimum risk seviyesi BIST 30 endeksinde en yüksek getiriye elde etmişken, BIST 100 endeksinde ise tam tersi en düşük getiriye elde etmiştir.

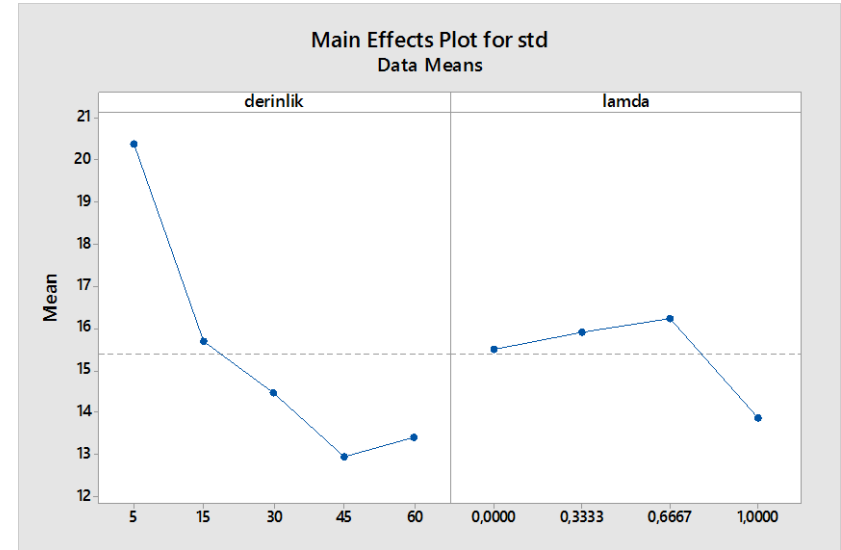
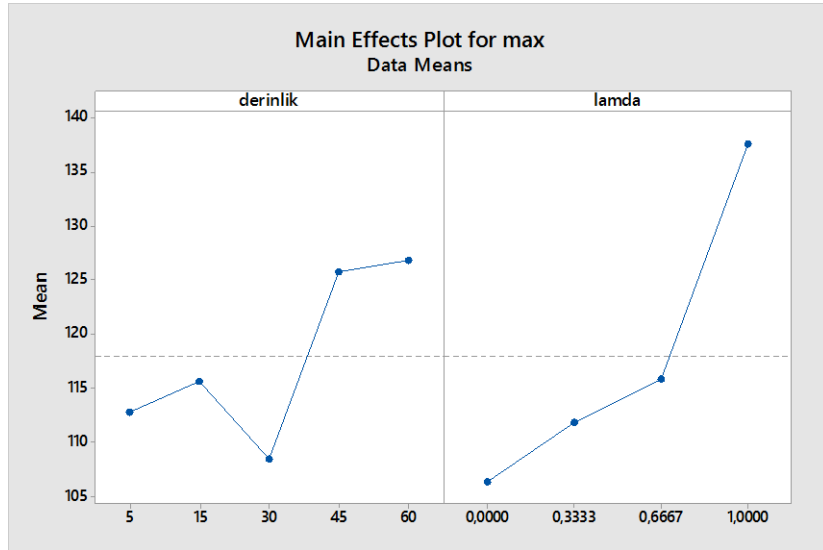
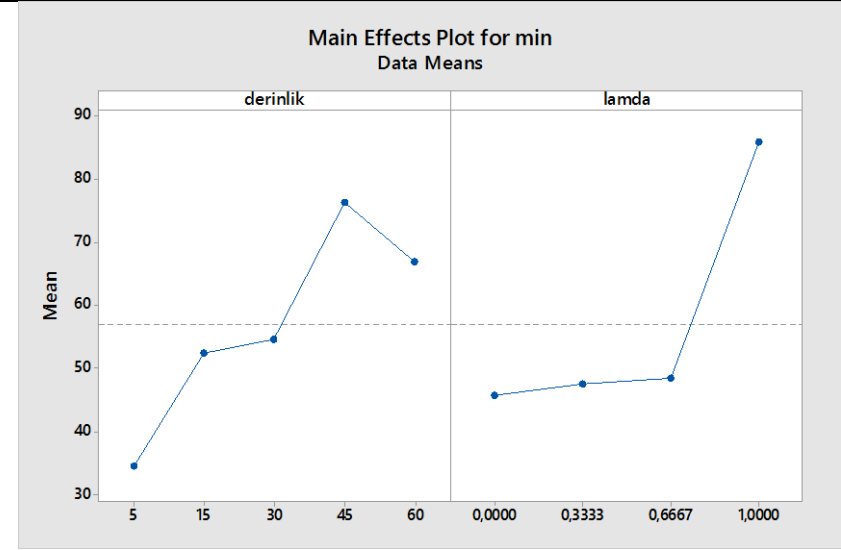
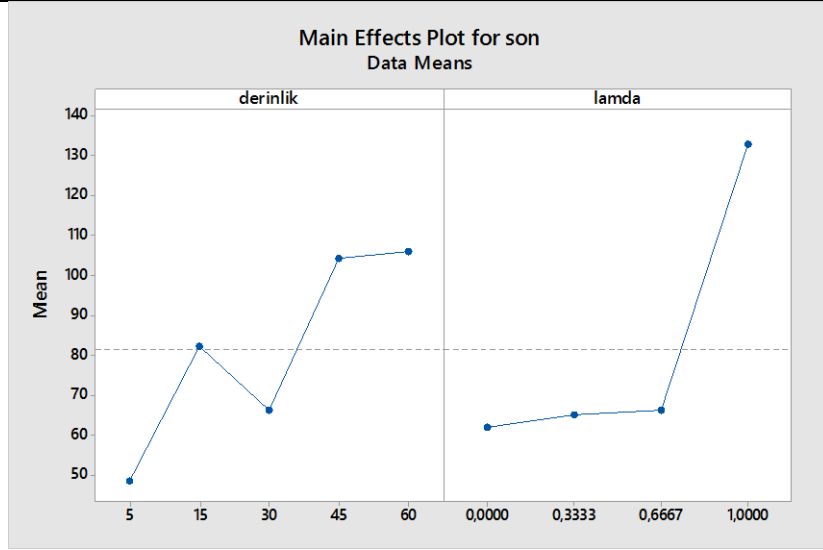


Şekil 64: BIST 30 endeksi için ana etki grafikleri: ortalama değerler

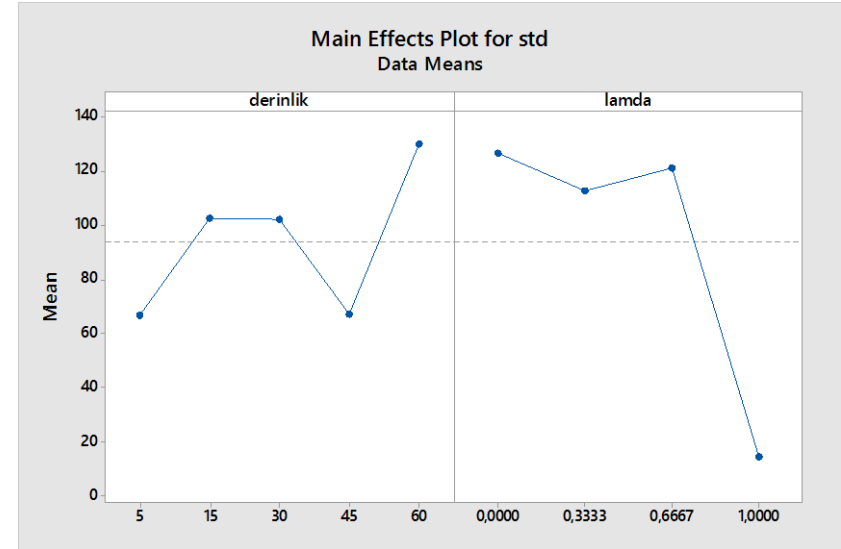
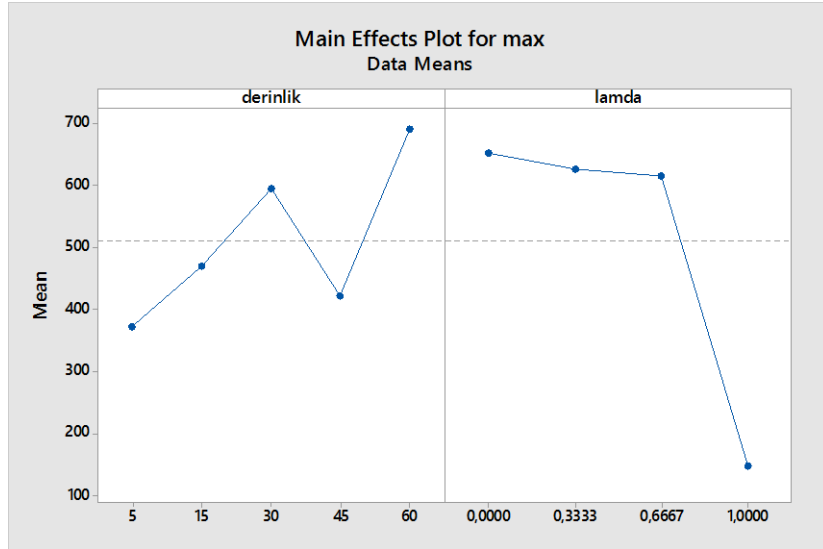
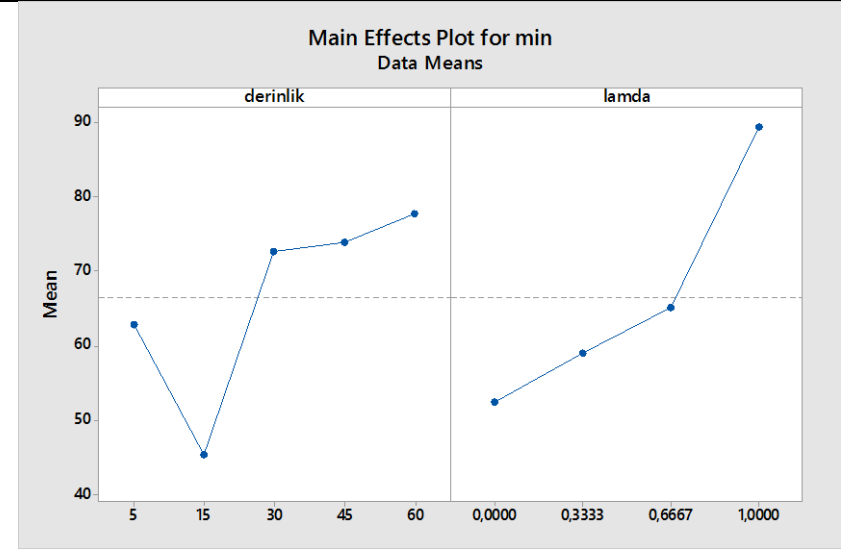
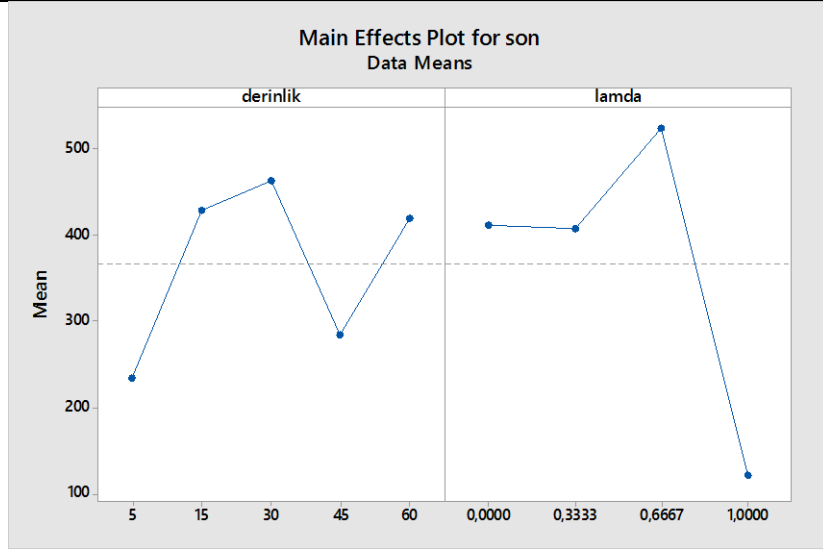


Şekil 65: BIST 100 endeksi için ana etki grafikleri: ortalama değerler

BIST 30 endeksi için ana etki grafikleri (son, minimum, maksimum ve standart sapma değerleri) Şekil 66'da, BIST 100 endeksi için ana etki grafikleri ise Şekil 67'de verilmiştir. BIST 30 endeksi için ortalamada en yüksek getiriye sağlayan 60 günlük derinliğin standart sapmasının da düşük olduğu gözlemlenmiştir. Yine, en iyi getiriye sağlayan minimum risk seviyesinin de standart sapmasının en düşük olduğu gözlemlenmiştir. BIST 30'un aksine, BIST 100 endeksinde, ortalamada en iyi getiriye sağlayan 60 günlük derinlik, en yüksek standart sapmaya sahiptir. BIST 100 endeksinde, düşük risk seviyesindeki ($\lambda = 0.66$) yatırımların ortalamada en yüksek getiriye sağlamasına rağmen, en yüksek standart sapmaya sahiptir. BIST 30 ve BIST 100 endeks sonuçları, en iyi yatırım stratejisinin endeksten endekse farklılık gösterebileceğini ortaya koymuştur.



Şekil 66: BIST 30 endeksi için ana etki grafikleri: son, minimum, maksimum ve standart sapma değerleri



Şekil 67: BIST 100 endeksi için ana etki grafikleri: son, minimum, maksimum ve standart sapma değerleri

5. SONUÇ VE TARTIŞMA

Proje önerisinde, projenin tam anlamıyla başarıya ulaşmış sayılabilmesi için iş-zaman çizelgesinde yer alan her bir ana iş paketinin hedefi, başarı ölçütü (ne ölçüde gerçekleşmesi gerektiği) ve projenin başarısındaki önem derecesi başarı ölçütleri tablosunda belirtilmiştir. Projenin sonuçlanması ile birlikte, Tablo 23’de verilen başarı ölçütleri tablosuna, gerçekleştirilen iş paketlerini özetlemek için sonuç ve yorum sütunu eklenmiştir.

Tablo 23: Başarı ölçütleri gerçekleşme tablosu

#	İş Paketi Hedefi	Başarı Ölçütü (%, sayı, ifade, vb.)	Önem (%)**	Sonuç ve Yorum
1	Literatürün kapsamlı olarak taranarak ulusal ve uluslararası çalışmaların incelenmesi ve projenin güncel kalmasının sağlanması	İş-zaman çizelgesinde sunulduğu şekilde ilgili dönemlerde hazırlanacak güncel literatür değerlendirme raporunun oluşturulması	5	Proje süresi boyunca, literatür kapsamlı olarak taranmış ve projenin güncel kalması sağlanmıştır. Güncel literatür değerlendirme raporunun oluşturulmasına ek olarak, proje ekibi yapılan bu detaylı araştırmaların literatüre katkı sağlaması için ULAKBİM kapsamında taranan bir derleme çalışması yayınlamıştır. Ayrıca, proje ekibinin SCI-Expanded kapsamında taranan dergilerde (<i>Swarm and Evolutionary Computation</i> ve <i>Computers & Operations Reserach</i>) değerlendirilme aşamasında olan iki adet derleme çalışması daha bulunmaktadır. Sözü edilen bu çalışmalar, halen değerlendirilme aşamasında olmaları nedeniyle henüz ARDEB proje takip sistemine yüklenmemiş olup, yayımlandıktan sonra sisteme yüklenecektir.
2	Algoritmanın performans testleri ve analizi için uluslararası veri setlerinin yanında Türkiye’deki BIST30 ve BIST100 endekslerine ait verilerin elde edilmesi	Verileri barındıran veri seti dosyalarının hazırlanması	5	Dünyadaki gelişmiş ülke endekslerin performans testlerine ek olarak, gelişmekte olan Türkiye piyasalarındaki Borsa İstanbul verileri (XU030 ve XU100) de Bloomberg veri tabanından indirilmiş, proje hedeflerinde de belirtildiği gibi, SCI-Expanded kapsamında taranan <i>Expert System with Applications</i> (Kalaycı vd. 2017) isimli dergide analizleri yapılmıştır. Ayrıca, Pamukkale Üniversitesi web alanında (http://www.pau.edu.tr/portfolio/) araştırmacıların dikkatine sunulmuştur. Bu sayede, Türkiye piyasalarındaki veri setlerinin de uluslararası literatüre kazandırılması sağlanmıştır.
3	Literatürde tanımlı olan portföy optimizasyonu probleminin matematiksel modelinin kurularak ilk çözümlerin oluşturulması	Modelin kurulması	5	Model birinci gelişme raporunda da belirtildiği gibi başarıyla kurulmuştur.

4	Matematiksel çözümler kullanılarak küçük boyuttaki bir veri setinin optima çözülmesi, modelin doğrulanması ve geliştirilen sezgisel algoritmalar için bir referans noktası olarak yapılacak karşılaştırmalara rehberlik etmesi	Küçük ölçekli problemler için optimum sonuçların elde edilmesi	5	Küçük ölçekli veri setini BONMIN ve CPLEX çözümleri ile optimal olarak çözebilen bir model GAMS ara yüzü ile kurulmuş, büyük ölçekli veri setlerinde hiçbir çözümlerle optimal çözümün yakalanamadığı görülmüştür. Bu nedenle, iş paketi amacına ulaşmış ve etkin algoritmaların geliştirilmesi kapsamında sezgisel algoritmaların uyarlanması aşamasına geçilmiştir.
5	Projenin devam eden iş paketlerine yönelik rehberlik etmesi için literatürdeki mevcut yöntemlerin kodlanması, veri setleri üzerinde kabul edilebilir süreler içerisinde çözümlerinin test edilmesi	Literatürde mevcut yöntemlerden GA, PSO, BT ve YA tekniklere dayalı algoritmaların kurulmuş olması ve test problemlerinin çözümlenmiş olması	5	Projenin birinci döneminde yapılan literatür araştırması, portföy optimizasyonu probleminin çözümü için literatürde en çok tercih edilen yöntemlerin genetik algoritmalar (GA) ve parçacık sürü optimizasyonu (PSO) tekniklerinin olduğunu göstermiştir. Bu nedenle, GA ve PSO teknikleri probleme uyarlanmış ve test problemlerinin kabul edilebilir süreler içerisinde çözümlendiği görülmüştür. Projenin devam eden iş paketlerine yönelik rehberlik eden bu iş paketi de başarıyla tamamlanarak proje önerisinde belirtilen tekniklerin uyarlanması aşamasına geçilmiştir.
6	YAK tekniğinin yapılacak deneysel tasarım ile en iyi performansı yakalandıktan sonra, problemi kabul edilebilir süreler içerisinde etkin bir şekilde çözümlerinin test edilmesi	Literatürdeki GA, PSO, BT ve YA tekniklerine kıyasla etkin sınıra yakınsama, çözüm sayısı veya işlem süresi bakımından daha iyi sonuçlar elde edilmiş olması.	5	Yapay Arı Kolonisi (YAK) tekniği probleme başarıyla uyarlanmış, etkin sınıra yakınsama ve çözüm süresi performansı da üst düzeye ulaştırılarak, literatürde yayınlanmış olan tekniklerle rekabet edebilir düzeye getirilmiş ve SCI-Expanded kapsamında taranan <i>Expert System with Applications</i> isimli dergide TÜBİTAK desteği belirtilerek yayımlanmıştır (Kalaycı vd. 2017).
7	KKO tekniğinin yapılacak deneysel tasarım ile en iyi performansı yakalandıktan sonra, problemi kabul edilebilir süreler içerisinde etkin bir şekilde çözümlerinin test edilmesi	Literatürdeki GA, PSO, BT ve YA tekniklerine kıyasla etkin sınıra yakınsama, çözüm sayısı veya işlem süresi bakımından daha iyi sonuçlar elde edilmiş olması.	5	Karınca Kolonisi Optimizasyonu (KKO) tekniği probleme başarıyla uyarlanmış, etkin sınıra yakınsama performansı üst düzeye ulaştırılarak, literatürde yayınlanmış olan tekniklerle rekabet edebilir düzeye getirilmiş ve SCI-Expanded kapsamında taranan bir dergide (<i>Annals of Operations Research</i>) yayınlanmak üzere hazırlanmaktadır.
8	DKA tekniğinin yapılacak deneysel tasarım ile en iyi performansı yakalandıktan sonra, problemi kabul edilebilir süreler içerisinde etkin bir şekilde çözümlerinin test edilmesi	Literatürdeki GA, PSO, BT ve YA tekniklerine kıyasla etkin sınıra yakınsama, çözüm sayısı veya işlem süresi bakımından daha iyi sonuçlar elde edilmiş olması.	5	Değişken komşuluk arama (DKA) tekniği de probleme başarıyla uyarlanmış, etkin sınıra yakınsama ve çözüm süresi performansı da üst düzeye ulaştırılarak, literatürde yayınlanmış olan tekniklerle rekabet edebilir düzeye getirilmiş ve SCI-Expanded kapsamında taranan bir dergide (<i>Computers & Operations Research</i>) yayınlanmak üzere hazırlanmaktadır.

<p>9 Uyarlanan her metodun performans testleri, karşılaştırılmaları ve analizlerinin yapılması. Bu aşama, melezleştirme fikirlerinin oluşması açısından büyük önem arz etmektedir.</p>	<p>Performans testlerinin tamamlanması.</p>	<p>10</p>	<p>Proje boyunca yapılan literatür araştırmalar ve karşılaştırmalı performans testleri, portföy optimizasyonu probleminin çözümünde yerel arama prosedürlerinin algoritma performansına olumlu katkı sağlayabileceğini göstermiştir. Bu nedenle, literatürde henüz portföy optimizasyonunun çözümüne uyarlanmamış ancak sürekli düzlemdeki birçok problemde başarısı kanıtlanmış yerel arama prosedürlerinden hem türev bilgisi kullanan (Steepest Descent,ve Fletcher Reeves), hem de türev bilgisi kullanmayan (Hooje Jeeves ve Nelder Mead) algoritmalar portföy optimizasyonu problemini çözebilecek şekilde uyarlanmıştır. Tek başına yerel optimuma takılan bu algoritmalar, KKO ve DKA algoritmalarının performansını artırmak için gömülü olarak çalışacak şekilde tasarlanmıştır.</p>
<p>10 İP4, İP 5, İP 6 ve İP7 ile önerilen çözüm yaklaşımlarının birbirlerinin eksikliklerini tamamlayacak, literatürdeki mevcut yöntemlerin kodlama yapıları ve operatör stratejilerinden yararlanılarak elde edilen melezleştirme fikirleriyle en etkin algoritmanın tasarlanması ve uluslararası veri setlerinde elde edilen sonuçların literatürdeki tüm çalışmaların sonuçlarıyla karşılaştırılmasıyla daha iyi sonuçların elde edilmesi</p>	<p>Literatürdeki tüm çalışmalara kıyasla; etkin sınırdan yüzdelik sapma oranı, etkin sınır üzerinde bulunabilen çözüm sayısı veya işlem süresi bakımından daha etkin algoritmaların geliştirilmesi ve daha iyi sonuçların elde edilmesi</p>	<p>40</p>	<p>Bu projede, literatüre yeni bir problem öne sürülmemiş, literatürde hali hazırda tanımlanmış ve hakkında yüzlerde çalışma yayınlanmış bir problem üzerine odaklanıldığı için, projenin en önemli özgünlüğünün literatürdeki yayınlanmış tekniklerle rekabet edebilir nitelikte algoritmaların literatüre kazandırılması olduğu rahatlıkla söylenebilir. Bu kapsamda, geliştirilmiş olan algoritmalarından YAK algoritmasının SCI-Expanded kapsamında taranan saygın bir akademik dergide yayınlanmaya kabul görmüş olması da bu başarı ölçütünün sağlandığının bir göstergesi olabilir. Ayrıca, proje kapsamında geliştirilen KKO ve DKA algoritmaları da göstermiş oldukları performansları ile saygın dergilerde yayınlanmaya adaydır. Onuncu iş paketi kapsamında Yapay Arı Kolonisi (YAK), Karınca Kolonisi Optimizasyonu (KKO) ve Değişken Komşuluk Arama (DKA) algoritmalarının saf halleri üzerinde yapılan eleman sayısı kısıtlı portföy optimizasyonu problemine özgü yapılan iyileştirmeler ve oluşan melez yapılar aşağıdaki tabloda özetlenmiştir:</p>

	YAK	KKO	DKA
Alternatif tamir prosedürü	✓	✓	✓
Uyumsuzluğa izin veren seçim prosedürü	✓	✓	
Steepest Descent ve düzgün rassal başlangıç adım uzunluğu			✓
Fletcher Reeves ve düzgün rassal başlangıç adım uzunluğu			✓
Hookee Jeeves ve düzgün rassal başlangıç adım uzunluğu			✓
Nelder Mead ve başlangıç simplexli oluşturma formülasyonu		✓	✓

11	Sonuçların çeşitli senaryolar altında analiz edilmesi ve farklı parametrelere göre hassasiyet gösterebilecek olan algoritmalar için deneysel tasarım yapılması	Algoritmaların tam performansla çalışması için en uygun parametre setlerinin belirlenmesi ve gerekli istatistiksel analizlerin tamamlanması	5	Algoritmalar için en uygun parametre seti deneysel tasarım yapılarak belirlenmiş, MINITAB yazılımı ile yapılan istatistiksel analizler raporda paylaşılmıştır.
12	Proje kapsamında önerilen modellerin katkılarının sonuçlar üzerinde ortaya konması ve ISI Web of Knowledge'a kayıtlı uluslararası dergilerde makale(ler) olarak yayınlanması	Geliştirilen modellerin ve sunulan çözüm yöntemlerinin uluslararası konferanslarda sunulması ve en az iki adet makalenin uluslararası literatüre kazandırılması	5	<p>Projede hedefleri arasında olan proje çıktılarının konferanslarda sunulması ve saygın akademik dergilerde yayınlanması hedefi gerçekleştirilmiş ve gerçekleşmeye devam etmektedir.</p> <p>Proje kapsamındaki çalışmaları sunmak üzere katılımcısı olunan konferanslar:</p> <ul style="list-style-type: none"> - Textile Science and Economy, 16-19 Mayıs 2016, Zrenjanin, Sırbistan - Yöneylem Araştırması ve Endüstri Mühendisliği 36. Ulusal Kongresi, 13-15 Temmuz 2016, Yaşar Üniversitesi, İzmir, Türkiye - International Conference on Science, Innovation and Management, 20-21 Temmuz 2016, Tokyo, Japonya - International Conference on Engineering Technology and Innovation, 22-26 Mart 2017, Sarajevo, Bosna Hersek <p>Proje kapsamındaki çalışmaların yayımlanmış olduğu akademik dergiler:</p> <ul style="list-style-type: none"> - Pamukkale University Journal of Science - Expert Systems with Applications <p>Yayına hazırlanan/ değerlendirme aşamasında olan çalışmaların yayınlanması planlanan akademik dergiler:</p> <ul style="list-style-type: none"> - Swarm and Evolutionary Computation - Computers and Operations Research - Annals of Operations Research

Bu projede, literatürde hesaplama karmaşıklığının yüksek olduğu ispatlanmış ve NP-Zor sınıfında bir problem olarak yer alan eleman sayısı kısıtlı portföy optimizasyonu problemi ele alınmıştır. Bu problemin çözümü için metasezgisel tekniklerden yararlanılarak etkin algoritmaların geliştirilmesi öncelikli olarak amaçlanmıştır. Bilhassa, finans, matematik ve bilgisayar bilimi alanlarındaki birçok araştırmacı tarafından ilgi gören bu problemin çözümü için proje ekibi, iki yıllık proje süresi boyunca yoğun bir şekilde çalışarak, proje önerisinde öngörülen hedeflere ulaşmayı başarmıştır. Proje amaç ve hedefleri doğrultusunda gerçekleştirilen çalışmalar aşağıda maddeler halinde özetlenmiştir:

- ✓ Bu projede, karmaşık portföy optimizasyonu için etkin algoritmaların geliştirilmiş ve literatürdeki mevcut metotlara kıyasla daha iyi sonuçlar elde edilmiştir. Çalışmalardan bir tanesi SCI Expanded kapsamında taranan bir dergide yayınlanmış, diğerleri de ortaya koydukları performans ile yayınlanmaya adaydır.
- ✓ Dünyadaki gelişmiş ülke endekslerin performans testlerine ek olarak, gelişmekte olan Türkiye piyasalarındaki Borsa İstanbul verileri (XU030 ve XU100) de Bloomberg veri tabanından indirilmiş, proje hedeflerinde de belirtildiği gibi, SCI-Expanded kapsamında taranan Expert System with Applications (Kalayci vd. 2017) isimli dergide analizleri yapılmıştır. Ayrıca, Pamukkale Üniversitesi web alanında (<http://www.pau.edu.tr/portfolio/>) araştırmacıların dikkatine sunulmuştur. Bu sayede, Türkiye piyasalarındaki veri setlerinin de uluslararası literatüre kazandırılması sağlanmıştır.
- ✓ Projedeki bilgi ve tecrübelerin paylaşılması kapsamında, Pamukkale Üniversitesi lisans müfredatına IENG 473 kodlu Portföy Optimizasyonu isimli bir seçmeli ders eklenmiş ve bizzat proje yürütücüsü tarafından yönetilmiştir.
- ✓ Potansiyel genç araştırmacıların ilgisi çekilmiş, yüksek lisans ve doktora öğrencileri proje kapsamında yetiştirilmiştir. Bu kapsamda, bir doktora öğrencisi mezun edilmiş ve bir yüksek lisans öğrencisi de en kısa zamanda mezun edilecektir.
- ✓ Proje yürütücüsün kariyerini geliştirmesi amacını da taşıyan bu proje, proje çıktılarıyla birlikte, proje yürütücüsünün ÜAK tarafından belirlenen doçentlik kriterlerini sağlamasına yardımcı olmuştur.
- ✓ Problemi hızlı ve etkin biçimde çözebilen algoritmaların, yapılan analizler ile ticari yazılıma dönüştürülebileceği ispatlanmış olması sayesinde, finans alanında yatırımcılara tavsiye niteliğinde öneriler getirebilecek, yeni bir şirket kurulmasının önünü açmıştır. Bu bulgular, projenin son aşamasında yapılan bir çalıştay da tartışılarak destek bulmuştur.

Referanslar

- Abbas A., Haider S. Comparison of AIS and PSO for constrained portfolio optimization. In: Proceedings - 2009 International Conference on Information and Financial Engineering, ICIFE 2009, 2009. pp 50-54. doi:10.1109/ICIFE.2009.32
- Ackora-Prah J., Gyamerah S. A., Andam P. S. 2014. "A heuristic crossover for portfolio selection", *Applied Mathematical Sciences*, (65-68), 3215-3227.
- Adebisi Ayodele A., Ayo Charles K. 2015. "Portfolio selection problem using generalized differential evolution 3", *Applied Mathematical Sciences*, 9 (41-44), 2069-2082.
- Anagnostopoulos K. P., Chatzoglou P. D., Katsavounis S. 2010. "A reactive greedy randomized adaptive search procedure for a mixed integer portfolio optimization problem", *Managerial Finance*, 36 (12), 1057-1065.
- Anagnostopoulos K. P., Mamanis G. 2010. "A portfolio optimization model with three objectives and discrete variables", *Computers & Operations Research*, 37 (7), 1285-1297.
- Anagnostopoulos K. P., Mamanis G. 2011a. "The mean-variance cardinality constrained portfolio optimization problem: An experimental evaluation of five multiobjective evolutionary algorithms", *Expert Systems with Applications*, 38 (11), 14208-14217.
- Anagnostopoulos K. P., Mamanis G. 2011b. "Multiobjective evolutionary algorithms for complex portfolio optimization problems", *Computational Management Science*, 8 (3), 259-279.
- Aouni B. d., Ben Abdelaziz F., Martel J.-M. 2005. "Decision-maker's preferences modeling in the stochastic goal programming", *European Journal of Operational Research*, 162 (3), 610-618.
- Aranha C., Iba H. 2009. "The Memetic Tree-based Genetic Algorithm and its application to Portfolio Optimization", *Memetic Computing*, 1 (2), 139-151.
- Arkeman Y., Yusuf A., Mushthofa, Laxmi G. F., Seminar K. B. 2013. "The formation of optimal portfolio of mutual shares funds using multi-objective genetic algorithm", *Telkomnika*, 11 (3), 625-636.
- Babaei S., Sepehri M. M., Babaei E. 2015. "Multi-objective portfolio optimization considering the dependence structure of asset returns", *European Journal of Operational Research*, 244 (2), 525-539.
- Bacanin N., Tuba M. 2014. "Firefly algorithm for cardinality constrained mean-variance portfolio optimization problem with entropy diversity constraint", *Scientific World Journal*, 2014.
- Bacanin N., Tuba M. Fireworks algorithm applied to constrained portfolio optimization problem. In: 2015 IEEE Congress on Evolutionary Computation, CEC 2015 - Proceedings, 2015. pp 1242-1249. doi:10.1109/CEC.2015.7257031
- Bacanin N., Tuba M., Pelevic B. 2014. "Constrained portfolio selection using artificial bee colony (ABC) algorithm", *International Journal of Mathematical Models and Methods in Applied Sciences*, 8, 190-198.
- Baluja S. (1994) Population-based incremental learning. a method for integrating genetic search based function optimization and competitive learning. DTIC Document,
- Bandyopadhyay S., Saha S., Maulik U., Deb K. 2008. "A simulated annealing-based multiobjective optimization algorithm: AMOSA", *IEEE Transactions on Evolutionary Computation*, 12 (3), 269-283.
- Barak S., Abessi M., Modarres M. 2013. "Fuzzy turnover rate chance constraints portfolio model", *European Journal of Operational Research*, 228 (1), 141-147.
- Baykasoğlu A., Avci M. G., Burcin Özsoydan F. 2016. "Erratum to "A GRASP based solution approach to solve cardinality constrained portfolio optimization problems" [Comput. Indus. Eng. 90 (2015) 339-351]", *Computers & Industrial Engineering*, 96, 249-250.
- Baykasoğlu A., Yunusoglu M. G., Burcin Özsoydan F. 2015. "A GRASP based solution approach to solve cardinality constrained portfolio optimization problems", *Computers & Industrial Engineering*, 90, 339-351.
- Bermúdez J. D., Segura J. V., Vercher E. 2012. "A multi-objective genetic algorithm for cardinality constrained fuzzy portfolio selection", *Fuzzy Sets and Systems*, 188 (1), 16-26.
- Bevilacqua V., Pacelli V., Saladino S. (2012) A Novel Multi Objective Genetic Algorithm for the Portfolio Optimization. In: Huang D-S, Gan Y, Bevilacqua V, Figueroa JC (eds) *Advanced Intelligent Computing: 7th International Conference, ICIC 2011, Zhengzhou, China, August 11-14, 2011. Revised Selected Papers*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 186-193. doi:10.1007/978-3-642-24728-6_25
- Bhattacharyya R., Kar S., Majumder D. D. 2011. "Fuzzy mean-variance-skewness portfolio selection models by interval analysis", *Computers and Mathematics with Applications*, 61 (1), 126-137.
- Blum C., Roli A. 2003. "Metaheuristics in combinatorial optimization: Overview and conceptual comparison", *ACM Computing Surveys (CSUR)*, 35 (3), 268-308.

- Bonami P., Lejeune M. A. 2009. "An Exact solution approach for portfolio optimization problems under stochastic and integer constraints", *Operations Research*, 57 (3), 650-670.
- Branke J., Scheckenbach B., Stein M., Deb K., Schmeck H. 2009. "Portfolio optimization with an envelope-based multi-objective evolutionary algorithm", *European Journal of Operational Research*, 199 (3), 684-693.
- Brito R. P., Vicente L. N. (2014) Efficient cardinality/mean-variance portfolios. *IFIP Advances in Information and Communication Technology*, vol 443. doi:10.1007/978-3-662-45504-3_6
- Busetti F. Heuristic approaches to realistic portfolio optimisation. In: *WIT Transactions on Modelling and Simulation*, 2006. pp 361-370. doi:10.2495/CF060351
- Cadenas J. M., Carrillo J. V., Garrido M. C., Ivorra C., Lamata T., Liern V. 2009. "A hybrid meta-heuristic to solve the portfolio selection problem", *risk*, 11, 1.
- Cao J., Tao L. Improved particle swarm algorithm for portfolio optimization problem. In: *ICIMA 2010 - 2010 2nd International Conference on Industrial Mechatronics and Automation*, 2010. pp 561-564. doi:10.1109/ICINDMA.2010.5538246
- Carrizosa E., Dražić M., Dražić Z., Mladenović N. 2012. "Gaussian variable neighborhood search for continuous optimization", *Computers & Operations Research*, 39 (9), 2206-2213.
- Cesarone F., Scozzari A., Tardella F. 2013. "A new method for mean-variance portfolio optimization with cardinality constraints", *Annals of Operations Research*, 205 (1), 213-234.
- Cesarone F., Scozzari A., Tardella F. 2015. "Linear vs. quadratic portfolio selection models with hard real-world constraints", *Computational Management Science*, 12 (3), 345-370.
- Chang J. F., Chen K. L. 2009. "Applying new investment satisfied capability index and particle swarm optimization to stock portfolio selection", *ICIC Express Letters*, 3 (3), 349-354.
- Chang J. F., Hsu S. W. The construction of stock's portfolios by using particle swarm optimization. In: *Second International Conference on Innovative Computing, Information and Control, ICICIC 2007*, 2008. doi:10.1109/ICICIC.2007.568
- Chang T.-J., Yang S.-C., Chang K.-J. 2009. "Portfolio optimization problems in different risk measures using genetic algorithm", *Expert Systems with Applications*, 36 (7), 10529-10537.
- Chang T. J., Meade N., Beasley J. E., Sharaiha Y. M. 2000. "Heuristics for cardinality constrained portfolio optimisation", *Computers & Operations Research*, 27 (13), 1271-1302.
- Chen A. H. L., Liang Y. C., Liu C. C. Portfolio optimization using improved artificial bee colony approach. In: *Proceedings of the 2013 IEEE Conference on Computational Intelligence for Financial Engineering and Economics, CIFE 2013 - 2013 IEEE Symposium Series on Computational Intelligence, SSCI 2013*, 2013. pp 60-67. doi:10.1109/CIFE.2013.6611698
- Chen A. H. L., Yun-Chia L., Chia-Chien L. An artificial bee colony algorithm for the cardinality-constrained portfolio optimization problems. In: *Evolutionary Computation (CEC), 2012 IEEE Congress on*, 10-15 June 2012 2012. pp 1-8. doi:10.1109/CEC.2012.6252920
- Chen B., Lin Y., Zeng W., Xu H., Zhang D. 2017. "The mean-variance cardinality constrained portfolio optimization problem using a local search-based multi-objective evolutionary algorithm", *Applied Intelligence*, 1-21.
- Chen W. 2014. "An artificial bee colony algorithm for uncertain portfolio selection", *Scientific World Journal*, 2014.
- Chen W., Cai Y. M. Study on the efficient frontier in portfolio selection by using particle swarm optimization. In: *Chinese Control and Decision Conference, 2008, CCDC 2008*, 2008. pp 269-272. doi:10.1109/CCDC.2008.4597313
- Chen W., Ma H., Yang Y., Sun M. 2014. "Application of Artificial Bee Colony Algorithm to Portfolio Adjustment Problem with Transaction Costs", *Journal of Applied Mathematics*, 2014, 12.
- Chen W., Xu W. J., Yang L., Cai Y. M. Genetic algorithm with an application to complex portfolio selection. In: *Proceedings - 4th International Conference on Natural Computation, ICNC 2008*, 2008. pp 333-337. doi:10.1109/ICNC.2008.323
- Chen W., Zhang R. T., Cai Y. M., Xu F. S. Particle swarm optimization for constrained portfolio selection problems. In: *Proceedings of the 2006 International Conference on Machine Learning and Cybernetics*, 2006. pp 2425-2429. doi:10.1109/ICMLC.2006.258773
- Chen W., Zhang W. G. 2010. "The admissible portfolio selection problem with transaction costs and an improved PSO algorithm", *Physica A: Statistical Mechanics and its Applications*, 389 (10), 2070-2076.
- Chen Y., Mabu S., Hirasawa K. 2011. "Genetic relation algorithm with guided mutation for the large-scale portfolio optimization", *Expert Systems with Applications*, 38 (4), 3353-3363.
- Chen Z., Li Z., Wang L. 2015. "Concentrated portfolio selection models based on historical data", *Applied Stochastic Models in Business and Industry*, 31 (5), 649-668.

- Chiam S. C., Al Mamun A., Low Y. L. A realistic approach to evolutionary multiobjective portfolio optimization. In: 2007 IEEE Congress on Evolutionary Computation, CEC 2007, 2007. pp 204-211. doi:10.1109/CEC.2007.4424473
- Chiam S. C., Tan K. C., Mamun A. 2008. "Evolutionary multi-objective portfolio optimization in practical context", *International Journal of Automation and Computing*, 5 (1), 67-80.
- ChiangLin C. Y. Applications of genetic algorithm to portfolio optimization with practical transaction constraints. In: *Proceedings of the 9th Joint Conference on Information Sciences, JCIS 2006*, 2006. doi:10.2991/jcis.2006.273
- Chu S.-C., Tsai P.-W. 2007. "Computational intelligence based on the behavior of cats", *International Journal of Innovative Computing, Information and Control*, 3 (1), 163-173.
- Civicioglu P. 2013. "Artificial cooperative search algorithm for numerical optimization problems", *Information Sciences*, 229, 58-76.
- Coello C. A. C., Pulido G. T., Lechuga M. S. 2004. "Handling multiple objectives with particle swarm optimization", *IEEE Transactions on Evolutionary Computation*, 8 (3), 256-279.
- Corazza M., Fasano G., Gusso R. Portfolio selection with an alternative measure of risk: Computational performances of Particle Swarm Optimization and Genetic Algorithms. In: *Mathematical and Statistical Methods for Actuarial Sciences and Finance*, 2012. pp 123-130. doi:10.1007/978-88-470-2342-0_15
- Corazza M., Fasano G., Gusso R. 2013. "Particle Swarm Optimization with non-smooth penalty reformulation, for a complex portfolio selection problem", *Applied Mathematics and Computation*, 224 (0), 611-624.
- Corne D., Knowles J., Oates M. The Pareto envelope-based selection algorithm for multiobjective optimization. In: *Parallel problem solving from nature PPSN VI*, 2000. Springer, pp 839-848
- Coutino-Gomez C. A., Torres-Jimenez J., Villarreal-Antelo B. M. (2003) Heuristic Methods for Portfolio Selection at the Mexican Stock Exchange. In: Liu J, Cheung Y-m, Yin H (eds) *Intelligent Data Engineering and Automated Learning: 4th International Conference, IDEAL 2003*, Hong Kong, China, March 21-23, 2003. Revised Papers. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 919-923. doi:10.1007/978-3-540-45080-1_130
- Cui T., Cheng S., Bai R. A combinatorial algorithm for the cardinality constrained portfolio optimization problem. In: *Proceedings of the 2014 IEEE Congress on Evolutionary Computation, CEC 2014*, 2014. pp 491-498. doi:10.1109/CEC.2014.6900357
- Cui X. T., Zheng X. J., Zhu S. S., Sun X. L. 2013. "Convex relaxations and MIQCQP reformulations for a class of cardinality-constrained portfolio selection problems", *Journal of Global Optimization*, 56 (4), 1409-1423.
- Cura T. 2009. "Particle swarm optimization approach to portfolio optimization", *Nonlinear Analysis: Real World Applications*, 10 (4), 2396-2406.
- De Castro L. N., Timmis J. 2002. *Artificial immune systems: a new computational intelligence approach*. Springer Science & Business Media.
- Deb K. 2000. "An efficient constraint handling method for genetic algorithms", *Computer Methods in Applied Mechanics and Engineering*, 186 (2-4), 311-338.
- Deb K., Pratap A., Agarwal S., Meyarivan T. 2002. "A fast and elitist multiobjective genetic algorithm: NSGA-II", *IEEE Transactions on Evolutionary Computation*, 6 (2), 182-197.
- Deng G.-F., Lin W.-T. (2010a) Ant Colony Optimization for Markowitz Mean-Variance Portfolio Model. In: Panigrahi BK, Das S, Suganthan PN, Dash SS (eds) *Swarm, Evolutionary, and Memetic Computing: First International Conference on Swarm, Evolutionary, and Memetic Computing, SEMCCO 2010*, Chennai, India, December 16-18, 2010. Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 238-245. doi:10.1007/978-3-642-17563-3_29
- Deng G.-F., Lin W.-T. (2010b) Swarm intelligence for cardinality-constrained portfolio problems. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol 6423 LNAI. doi:10.1007/978-3-642-16696-9_44
- Deng G.-F., Lin W.-T., Lo C.-C. 2012. "Markowitz-based portfolio selection with cardinality constraints using improved particle swarm optimization", *Expert Systems with Applications*, 39 (4), 4558-4566.
- Dorigo M., Gambardella L. M. 1997. "Ant colony system: a cooperative learning approach to the traveling salesman problem", *Evolutionary Computation*, *IEEE Transactions on*, 1 (1), 53-66.
- Dorigo M., Maniezzo V., Coloni A. 1996. "Ant system: optimization by a colony of cooperating agents", *Systems, Man, and Cybernetics, Part B: Cybernetics*, *IEEE Transactions on*, 26 (1), 29-41.
- Duran F. C., Cotta C., Fernández A. J. (2009) Evolutionary optimization for multiobjective portfolio selection under Markowitz's model with application to the caracas stock exchange. *Studies in Computational Intelligence*, vol 193. doi:10.1007/978-3-642-00267-0_18

- Ehrgott M., Klamroth K., Schwehm C. 2004. "An MCDM approach to portfolio optimization", *European Journal of Operational Research*, 155 (3), 752-770.
- Erickson M., Mayer A., Horn J. The niched pareto genetic algorithm 2 applied to the design of groundwater remediation systems. In: *International Conference on Evolutionary Multi-Criterion Optimization*, 2001. Springer, pp 681-695
- Farzi S., Shavazi A. R., Pandari A. R. 2013. "Using Quantum-Behaved Particle Swarm Optimization for portfolio selection problem", *International Arab Journal of Information Technology*, 10 (2).
- Fastrich B., Winker P. 2012. "Robust portfolio optimization with a hybrid heuristic algorithm", *Computational Management Science*, 9 (1), 63-88.
- Feo T., Resende M. C. 1995. "Greedy Randomized Adaptive Search Procedures", *Journal of Global Optimization*, 6 (2), 109-133.
- Fernández A., Gómez S. 2007. "Portfolio selection using neural networks", *Computers & Operations Research*, 34 (4), 1177-1191.
- Fieldsend J. E., Matatko J., Peng M. (2004) Cardinality Constrained Portfolio Optimisation. In: Yang ZR, Yin H, Everson RM (eds) *Intelligent Data Engineering and Automated Learning – IDEAL 2004: 5th International Conference*, Exeter, UK. August 25-27, 2004. Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 788-793. doi:10.1007/978-3-540-28651-6_117
- Fletcher R., Reeves C. M. 1964. "Function minimization by conjugate gradients", *The computer journal*, 7 (2), 149-154.
- Fonseca C. M., Fleming P. J. Genetic Algorithms for Multiobjective Optimization: Formulation Discussion and Generalization. In: *Icga*, 1993. vol July. Citeseer, pp 416-423
- Forrest S., Mitchell M. 1993. "Relative building-block fitness and the building block hypothesis", *D Whitley, Foundations of Genetic Algorithms*, 2, 109-126.
- Fox R. L. 1971. *Optimization methods for engineering design*. Addison-Wesley Pub. Co.
- Freitas F. D., De Souza A. F., de Almeida A. R. 2009. "Prediction-based portfolio optimization model using neural networks", *Neurocomputing*, 72 (10-12), 2155-2170.
- Gandomi A. H., Alavi A. H. 2012. "Krill herd: a new bio-inspired optimization algorithm", *Communications in Nonlinear Science and Numerical Simulation*, 17 (12), 4831-4845.
- Gao J., Chu Z. An improved particle swarm optimization for the constrained portfolio selection problem. In: *Computational Intelligence and Natural Computing*, 2009. CINC'09. International Conference on, 2009. IEEE, pp 518-522
- García S., Quintana D., Galván I. M., Isasi P. 2012. "Time-stamped resampling for robust evolutionary portfolio optimization", *Expert Systems with Applications*, 39 (12), 10722-10730.
- Gaspero L. D., Tollo G. D., Roli A., Schaerf A. 2010. "Hybrid metaheuristics for constrained portfolio selection problems", *Quantitative Finance*, 11 (10), 1473-1487.
- Ge M. Artificial bee colony algorithm for portfolio optimization. In: *5th International Conference on Intelligent Control and Information Processing, ICICIP 2014 - Proceedings*, 2015. pp 449-453. doi:10.1109/ICICIP.2014.7010297
- Glover F. 1986. "Future paths for integer programming and links to artificial intelligence", *Computers & Operations Research*, 13 (5), 533-549.
- Glover F., Laguna M., Martí R. 2000. "Fundamentals of scatter search and path relinking", *Control and cybernetics*, 29 (3), 653-684.
- Goldberg D. E. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc.
- Golmakani H. R., Alishah E. J. Portfolio selection using an artificial immune system. In: *2008 IEEE International Conference on Information Reuse and Integration, IEEE IRI-2008*, 2008. pp 28-33. doi:10.1109/IRI.2008.4583000
- Golmakani H. R., Fazel M. 2011. "Constrained Portfolio Selection using Particle Swarm Optimization", *Expert Systems with Applications*, 38 (7), 8327-8335.
- Gundem B. (2015) "Carrefoursa'da kod değişikliği ve referans fiyat". <http://www.borsagundem.com/haber/carrefoursada-kod-degisikligi-ve-referans/230756>. Son erişim tarihi: 13 May 2016
- Guo S., Yu L., Li X., Kar S. 2016. "Fuzzy multi-period portfolio selection with different investment horizons", *European Journal of Operational Research*, 254 (3), 1026-1035.
- Haber H. (2013a) "Borsa İstanbul'da gong, Odaş Enerji için çaldı". <http://www.hurriyet.com.tr/borsa-istanbulda-gong-odas-enerji-icin-caldi-23332759>. Son erişim tarihi: 13 May 2016
- Haber H. (2013b) "Pegasus Hava Yolları, "PGSUS" kodu ile işlem görmeye başladı". <http://www.hurriyet.com.tr/pegasus-hava-yollari-pgsus-kodu-ile-islem-gormeye-basladi-23139952>. Son erişim tarihi: 13 May 2016
- Hadi A. S., El Naggat A. A., Abdel Bary M. N. 2016. "New model and method for portfolios selection", *Applied Mathematical Sciences*, 10 (5-8), 263-288.

- Hajnoori A., Amiri M., Alimi A. 2013. "Forecasting stock price using grey-fuzzy technique and portfolio optimization by invasive weed optimization algorithm", *Decision Science Letters*, 2 (3), 175-184.
- Hao F. F., Liu Y. K. 2009. "Mean-variance models for portfolio selection with fuzzy random returns", *Journal of Applied Mathematics and Computing*, 30 (1-2), 9-38.
- Hoklie, Zuhail L. R. Resolving multi objective stock portfolio optimization problem using genetic algorithm. In: 2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE), 26-28 Feb. 2010. pp 40-44. doi:10.1109/ICCAE.2010.5451372
- Holland J. H. 1975. *Adaptation in natural and artificial systems*. Ann Arbor, MI: The University of Michigan Press.
- Hong-mei L., Zhuo-fu W., Hui-min L. Artificial bee colony algorithm for real estate portfolio optimization based on risk preference coefficient. In: *Management Science and Engineering (ICMSE), 2010 International Conference on*, 2010. IEEE, pp 1682-1687
- Hooke R., Jeeves T. A. 1961. "Direct Search" Solution of Numerical and Statistical Problems", *Journal of the ACM (JACM)*, 8 (2), 212-229.
- Hopfield J. J. 1984. "Neurons with graded response have collective computational properties like those of two-state neurons", *Proceedings of the national academy of sciences*, 81 (10), 3088-3092.
- Horn J., Nafpliotis N., Goldberg D. E. A niched Pareto genetic algorithm for multiobjective optimization. In: *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence.*, Proceedings of the First IEEE Conference on, 1994. IEEE, pp 82-87
- Hu J., Zhangy G. 2010. "Comparison of portfolio optimization models with real features: An empirical study based on Chinese stock market", *Dynamics of Continuous, Discrete and Impulsive Systems Series B: Applications and Algorithms*, 17 (1), 83-100.
- Huang X. 2012. "Mean-variance models for portfolio selection subject to experts' estimations", *Expert Systems with Applications*, 39 (5), 5887-5893.
- Huang X., Shen W. Multi-period mean-variance model with transaction cost for fuzzy portfolio selection. In: *Proceedings - 2010 7th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2010, 2010*. pp 894-898. doi:10.1109/FSKD.2010.5569108
- Jiang K., Li D., Gao J., Yu J. X. Factor model based clustering approach for cardinality constrained portfolio selection. In: *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 2014. pp 10713-10718
- Jin Y., Qu R., Atkin J. A population-based incremental learning method for constrained portfolio optimisation. In: *Proceedings - 16th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2014, 2015*. pp 212-219. doi:10.1109/SYNASC.2014.36
- Kalayci C. B., Ertenlice O., Akyer H., Aygoren H. 2017. "An artificial bee colony algorithm with feasibility enforcement and infeasibility toleration procedures for cardinality constrained portfolio optimization", *Expert Systems with Applications*, 85, 61-75.
- Kamili H., Riffi M. E. 2015. "Portfolio selection using the cat swarm optimization", *Journal of Theoretical and Applied Information Technology*, 74 (3), 374-380.
- Kamili H., Riffi M. E. A comparative study on portfolio optimization problem. In: *Proceedings - 2016 International Conference on Engineering and MIS, ICEMIS 2016, 2016*. doi:10.1109/ICEMIS.2016.7745339
- Kao Y., Cheng H.-T. 2013. "Bacterial Foraging Optimization Approach to Portfolio Optimization", *Computational Economics*, 42 (4), 453-470.
- Karaboga D. (2005) *An idea based on Honey Bee Swarm for Numerical Optimization*. Erciyes University, Engineering Faculty, Computer Engineering Department, Kayseri
- Kennedy J., Eberhart R. Particle swarm optimization. In: *Neural Networks, 1995. Proceedings., IEEE International Conference on*, Nov/Dec 1995. pp 1942-1948 vol.1944. doi:10.1109/ICNN.1995.488968
- Khalidji M., Zeiaee M., Taei A., Jahed-Motlagh M. R., Khaloozadeh H. Dynamically weighted continuous ant colony optimization for biobjective portfolio selection using value-at-risk. In: *Proceedings - 2009 3rd Asia International Conference on Modelling and Simulation, AMS 2009, 2009*. pp 230-235. doi:10.1109/AMS.2009.133
- Kırış Ş., Ustun O. 2012. "An integrated approach for stock evaluation and portfolio optimization", *Optimization*, 61 (4), 423-441.
- Kirkpatrick S., Gelatt C. D., Jr., Vecchi M. P. 1983. "Optimization by simulated annealing", *Science*, 220 (4598), 671-680.
- Knowles J., Thiele L., Zitzler E. 2006. "A tutorial on the performance assessment of stochastic multiobjective optimizers", *Tik report*, 214, 327-332.
- Kocadağlı O., Keskin R. 2015. "A novel portfolio selection model based on fuzzy goal programming with different importance and priorities", *Expert Systems with Applications*, 42 (20), 6898-6912.

- Koshino M., Murata H., Kimura H. 2007. "Improved particle swarm optimization and application to portfolio selection", *Electronics and Communications in Japan, Part III: Fundamental Electronic Science (English translation of Denshi Tsushin Gakkai Ronbunshi)*, 90 (3), 13-25.
- Kukkonen S., Lampinen J. GDE3: The third evolution step of generalized differential evolution. In: *Evolutionary Computation, 2005. The 2005 IEEE Congress on, 2005. IEEE*, pp 443-450
- Kumar D., Mishra K. K. 2017. "Portfolio optimization using novel co-variance guided Artificial Bee Colony algorithm", *Swarm and Evolutionary Computation*, 33, 119-130.
- Kumar R., Bhattacharya S. 2012. "Cooperative search using agents for cardinality constrained portfolio selection problem", *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 42 (6), 1510-1518.
- Kuo R. J., Hong C. W. 2013. "Integration of genetic algorithm and particle swarm optimization for investment portfolio optimization", *Applied Mathematics and Information Sciences*, 7 (6), 2397-2408.
- Lai K. K., Yu L., Wang S., Zhou C. (2006) A Double-Stage Genetic Optimization Algorithm for Portfolio Selection. In: King I, Wang J, Chan L-W, Wang D (eds) *Neural Information Processing: 13th International Conference, ICONIP 2006, Hong Kong, China, October 3-6, 2006. Proceedings, Part III*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 928-937. doi:10.1007/11893295_102
- Lam A. Y., Li V. O. 2010. "Chemical-reaction-inspired metaheuristic for optimization", *IEEE Transactions on Evolutionary Computation*, 14 (3), 381-399.
- Lean Y., Wang S., Lai K. K. (2008) Portfolio optimization using evolutionary algorithms. In: *Reflexing Interfaces: The Complex Coevolution of Information Technology Ecosystems*. pp 235-245. doi:10.4018/978-1-59904-627-3.ch014
- Li D., Sun X., Wang J. 2006. "Optimal lot solution to cardinality constrained mean-variance formulation for portfolio selection", *Mathematical Finance*, 16 (1), 83-101.
- Li J. (2009) Compromise Approach-Based Genetic Algorithm for Constrained Multiobjective Portfolio Selection Model. In: Shi Y, Wang S, Peng Y, Li J, Zeng Y (eds) *Cutting-Edge Research Topics on Multiple Criteria Decision Making: 20th International Conference, MCDM 2009, Chengdu/Jiuzhaigou, China, June 21-26, 2009. Proceedings*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 697-704. doi:10.1007/978-3-642-02298-2_104
- Li L., Xue B., Tan L., Niu B. (2010) Improved particle swarm optimizers with application on constrained portfolio selection. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol 6215 LNCS. doi:10.1007/978-3-642-14922-1_72
- Li X., Zhang Y., Wong H. S., Qin Z. 2009. "A hybrid intelligent algorithm for portfolio selection problem with fuzzy returns", *Journal of Computational and Applied Mathematics*, 233 (2), 264-278.
- Li Y., Wang B., Watada J. Building a fuzzy multi-objective portfolio selection model with distinct risk measurements. In: *IEEE International Conference on Fuzzy Systems, 2011*. pp 1096-1102. doi:10.1109/FUZZY.2011.6007314
- Liagkouras K., Metaxiotis K. 2014. "A new Probe Guided Mutation operator and its application for solving the cardinality constrained portfolio optimization problem", *Expert Systems with Applications*, 41 (14), 6274-6290.
- Liang J. J., Qu B. Y. Large-scale portfolio optimization using multiobjective dynamic multi-swarm particle swarm optimizer. In: *Proceedings of the 2013 IEEE Symposium on Swarm Intelligence, SIS 2013 - 2013 IEEE Symposium Series on Computational Intelligence, SSCI 2013, 2013*. pp 1-6. doi:10.1109/SIS.2013.6615152
- Liao T., Stützle T., Montes de Oca M. A., Dorigo M. 2014. "A unified ant colony optimization algorithm for continuous optimization", *European Journal of Operational Research*, 234 (3), 597-609.
- Lin C.-C., Liu Y.-T. 2008. "Genetic algorithms for portfolio selection problems with minimum transaction lots", *European Journal of Operational Research*, 185 (1), 393-404.
- Lin D., Li X., Li M. 2005. "A genetic algorithm for solving portfolio optimization problems with transaction costs and minimum transaction lots", *Advances in Natural Computation*, 441-441.
- Liu Y.-J., Zhang W.-G., Zhang P. 2013. "A multi-period portfolio selection optimization model by using interval analysis", *Economic Modelling*, 33, 113-119.
- Liu Y. J., Zhang W. G. 2015. "A multi-period fuzzy portfolio optimization model with minimum transaction lots", *European Journal of Operational Research*, 242 (3), 933-941.
- Lloyd S. 1982. "Least squares quantization in PCM", *IEEE transactions on information theory*, 28 (2), 129-137.
- Loukeris N., Donnelly D., Khuman A., Peng Y. 2009. "A numerical evaluation of meta-heuristic techniques in portfolio optimisation", *Operational Research*, 9 (1), 81-103.

- Lu Z., Wang X. Improved portfolio optimization with non-convex and non-concave cost using genetic algorithms. In: Proceedings - 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer, MEC 2013, 2013. pp 2567-2570. doi:10.1109/MEC.2013.6885468
- Lwin K., Qu R. 2013. "A hybrid algorithm for constrained portfolio selection problems", *Applied Intelligence*, 39 (2), 251-266.
- Lwin K., Qu R., Kendall G. 2014. "A learning-guided multi-objective evolutionary algorithm for constrained portfolio optimization", *Applied Soft Computing*, 24 (0), 757-772.
- Lwin K., Qu R., Zheng J. Multi-objective scatter search with external archive for portfolio optimization. In: *IJCCI 2013 - Proceedings of the 5th International Joint Conference on Computational Intelligence*, 2013. pp 111-119
- Ma Y., Gong X., Tian G. A mean-semi-variance portfolio optimization model with full transaction costs. In: *Proceedings - 2014 6th International Conference on Computational Intelligence and Communication Networks, CICN 2014*, 2014. pp 623-627. doi:10.1109/CICN.2014.139
- Macedo L. L., Godinho P., Alves M. J. 2017. "Mean-semivariance portfolio optimization with multiobjective evolutionary algorithms and technical analysis rules", *Expert Systems with Applications*, 79, 33-43.
- Mansour N., Rebai A., Aouni B. 2007. "Portfolio selection through the imprecise goal programming model: Integration of manager's preferences", *Journal of Industrial Engineering International*, 3 (5), 1-8.
- Markowitz H. M. 1952. "Portfolio Selection", *The Journal of Finance*, 7 (1), 77-91.
- Markowitz H. M. 1959. *Portfolio Selection: Efficient Diversification of Investments*. New York: Yale University Press.
- Mashayekhi Z., Omrani H. 2016. "An integrated multi-objective Markowitz–DEA cross-efficiency model with fuzzy returns for portfolio selection problem", *Applied Soft Computing*, 38, 1-9.
- Mayambala F., Rönneberg E., Larsson T. Eigendecomposition of the mean-variance portfolio optimization model. In: *Springer Proceedings in Mathematics and Statistics*, 2015. pp 209-232. doi:10.1007/978-3-319-18567-5_11
- Mehrabian A. R., Lucas C. 2006. "A novel numerical optimization algorithm inspired from weed colonization", *Ecological informatics*, 1 (4), 355-366.
- Mishra S. K., Panda G., Majhi B. 2016. "Prediction based mean-variance model for constrained portfolio assets selection using multiobjective evolutionary algorithms", *Swarm and Evolutionary Computation*.
- Mishra S. K., Panda G., Majhi R. 2014a. "A comparative performance assessment of a set of multiobjective algorithms for constrained portfolio assets selection", *Swarm and Evolutionary Computation*, 16, 38-51.
- Mishra S. K., Panda G., Majhi R. 2014b. "Constrained portfolio asset selection using multiobjective bacteria foraging optimization", *Operational Research*, 14 (1), 113-145.
- Mishra S. K., Panda G., Meher S. Multi-objective particle swarm optimization approach to portfolio optimization. In: *2009 World Congress on Nature and Biologically Inspired Computing, NABIC 2009 - Proceedings*, 2009. pp 1612-1615. doi:10.1109/NABIC.2009.5393659
- Mladenović N., Dražić M., Kovačević-Vujčić V., Čangalović M. 2008. "General variable neighborhood search for the continuous optimization", *European Journal of Operational Research*, 191 (3), 753-770.
- Mladenović N., Hansen P. 1997. "Variable neighborhood search", *Computers & Operations Research*, 24 (11), 1097-1100.
- Moral-Escudero R., Ruiz-Torrubiano R., Suarez A. Selection of Optimal Investment Portfolios with Cardinality Constraints. In: *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on, 0-0 0 2006*. pp 2382-2388. doi:10.1109/CEC.2006.1688603
- Mozafari M., Tafazzoli S., Jolai F. 2011. "A new IPSO-SA approach for cardinality constrained portfolio optimization", *International Journal of Industrial Engineering Computations*, 2 (2), 249-262.
- Nelder J. A., Mead R. 1965. "A simplex method for function minimization", *The computer journal*, 7 (4), 308-313.
- Ni Q., Yin X., Tian K., Zhai Y. 2017. "Particle swarm optimization with dynamic random population topology strategies for a generalized portfolio selection problem", *Natural Computing*, 16 (1), 31-44.
- Niu B., Tan L., Xue B., Li L., Chai Y. Constrained portfolio selection using multiple swarms. In: *IEEE Congress on Evolutionary Computation, 2010. IEEE*, pp 1-7
- Niu B., Xue B., Li L., Chai Y. (2009) Symbiotic Multi-swarm PSO for Portfolio Optimization. In: Huang D-S, Jo K-H, Lee H-H, Kang H-J, Bevilacqua V (eds) *Emerging Intelligent Computing Technology and Applications. With Aspects of Artificial Intelligence: 5th International Conference on Intelligent Computing, ICIC 2009 Ulsan, South Korea, September 16-19, 2009*

- Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 776-784. doi:10.1007/978-3-642-04020-7_83
- Ong C. S., Jih-Jeng H., Tzeng G. H. 2005. "A novel hybrid model for portfolio selection", *Applied Mathematics and Computation*, 169 (2), 1195-1210.
- Pai G. A. V., Michel T. 2009. "Evolutionary optimization of constrained k-means clustered assets for diversification in small portfolios", *IEEE Transactions on Evolutionary Computation*, 13 (5), 1030-1053.
- Panigrahi B., Pandi V. R., Sharma R., Das S., Das S. 2011. "Multiobjective bacteria foraging algorithm for electrical load dispatch problem", *Energy Conversion and Management*, 52 (2), 1334-1342.
- Passino K. M. 2002. "Biomimicry of bacterial foraging for distributed optimization and control", *IEEE Control Systems*, 22 (3), 52-67.
- Peng H., Kitagawa G., Gan M., Chen X. 2011. "A new optimal portfolio selection strategy based on a quadratic form mean-variance model with transaction costs", *Optimal Control Applications and Methods*, 32 (2), 127-138.
- Picek S., Jakobovic D., Golub M. On the recombination operator in the real-coded genetic algorithms. In: *Evolutionary Computation (CEC), 2013 IEEE Congress on*, 20-23 June 2013 2013. pp 3103-3110. doi:10.1109/CEC.2013.6557948
- Pouya A. R., Solimanpur M., Rezaee M. J. 2016. "Solving multi-objective portfolio optimization problem using invasive weed optimization", *Swarm and Evolutionary Computation*, 28, 42-57.
- Qi Y. 2015. "On the criterion vectors of lines of portfolio selection with multiple quadratic and multiple linear objectives", *Central European Journal of Operations Research*, 1-14.
- Qin Z. 2015. "Mean-variance model for portfolio optimization problem in the simultaneous presence of random and uncertain returns", *European Journal of Operational Research*, 245 (2), 480-488.
- Reid S. G., Malan K. M. Constraint handling methods for portfolio optimization using particle swarm optimization. In: *Proceedings - 2015 IEEE Symposium Series on Computational Intelligence, SSCI 2015*, 2015. pp 1766-1773. doi:10.1109/SSCI.2015.246
- Rong X., Lu M., Deng L. 2009. "Multi-period model of portfolio investment and adjustment based on hybrid genetic algorithm", *Transactions of Tianjin University*, 15 (6), 415-422.
- Ruiz-Torrubiano R., Suarez A. 2010. "Hybrid Approaches and Dimensionality Reduction for Portfolio Selection with Cardinality Constraints", *Computational Intelligence Magazine, IEEE*, 5 (2), 92-107.
- Ruiz-Torrubiano R., Suárez A. Use of heuristic rules in evolutionary methods for the selection of optimal investment portfolios. In: *2007 IEEE Congress on Evolutionary Computation, CEC 2007*, 2007. pp 212-219. doi:10.1109/CEC.2007.4424474
- Ruiz-Torrubiano R., Suárez A. 2015. "A memetic algorithm for cardinality-constrained portfolio optimization with transaction costs", *Applied Soft Computing*, 36, 125-142.
- Sabar N. R., Song A. (2014) Dual population genetic algorithm for the cardinality constrained portfolio selection problem. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol 8886.
- Sadigh A. N., Mokhtari H., Iranpoor M., Fatemi Ghomi S. M. T. 2012. "Cardinality constrained portfolio optimization using a hybrid approach based on particle swarm optimization and hopfield neural network", *Advanced Science Letters*, 17 (1), 11-20.
- Sadjadi S. J., Gharakhani M., Safari E. 2012. "Robust optimization framework for cardinality constrained portfolio problem", *Applied Soft Computing*, 12 (1), 91-99.
- Schaerf A. 2002. "Local Search Techniques for Constrained Portfolio Selection Problems", *Computational Economics*, 20 (3), 177-190.
- Schaffer J. D. Multiple objective optimization with vector evaluated genetic algorithms. In: *Proceedings of the 1st international Conference on Genetic Algorithms*, 1985. L. Erlbaum Associates Inc., pp 93-100
- Sen T., Saha S., Ekbal A., Laha A. K. Bi-objective portfolio optimization using archive multi-objective simulated annealing. In: *2014 International Conference on High Performance Computing and Applications, ICHPCA 2014*, 2015. doi:10.1109/ICHPCA.2014.7045343
- Seyedhosseini S. M., Esfahani M. J., Ghaffari M. 2016. "A novel hybrid algorithm based on a harmony search and artificial bee colony for solving a portfolio optimization problem using a mean-semi variance approach", *Journal of Central South University*, 23 (1), 181-188.
- Shaikh R. A., Abbas A. Genetic algorithm and MS solver for portfolio optimization under exogenous influence. In: *2009 International Conference on Computer and Electrical Engineering, ICCEE 2009*, 2009. pp 555-558. doi:10.1109/ICCEE.2009.173
- Shaw D. X., Liu S., Kopman L. 2008. "Lagrangian relaxation procedure for cardinality-constrained portfolio optimization", *Optimization Methods and Software*, 23 (3), 411-420.

- Shoaf J., Foster J. A. Efficient set GA for stock portfolios. In: Proceedings of the IEEE Conference on Evolutionary Computation, ICEC, 1998. pp 354-359
- Skolpadungket P., Dahal K., Harnpornchai N. Portfolio optimization using multi-objective genetic algorithms. In: 2007 IEEE Congress on Evolutionary Computation, CEC 2007, 2007. pp 516-523. doi:10.1109/CEC.2007.4424514
- Socha K., Dorigo M. 2008. "Ant colony optimization for continuous domains", *European Journal of Operational Research*, 185 (3), 1155-1173.
- Soleimani H. 2007. "Portfolio selection using genetic algorithm. MS Degree Thesis, Amirkabir University of Technology", Industrial Engineering Department, Tehran.
- Soleimani H., Golmakani H. R., Salimi M. H. 2009. "Markowitz-based portfolio selection with minimum transaction lots, cardinality constraints and regarding sector capitalization using genetic algorithm", *Expert Systems with Applications*, 36 (3, Part 1), 5058-5063.
- Storn R., Price K. 1997. "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces", *Journal of Global Optimization*, 11 (4), 341-359.
- Streichen F., Tanaka-Yamawaki M. The effect of local search on the constrained portfolio selection problem. In: 2006 IEEE Congress on Evolutionary Computation, CEC 2006, 2006. pp 2368-2374
- Streichert F., Ulmer H., Zell A. Evaluating a hybrid encoding and three crossover operators on the constrained portfolio selection problem. In: Evolutionary Computation, 2004. CEC2004. Congress on, 19-23 June 2004 2004a. pp 932-939 Vol.931. doi:10.1109/CEC.2004.1330961
- Streichert F., Ulmer H., Zell A. (2004b) Evolutionary algorithms and the cardinality constrained portfolio optimization problem. In: Operations Research Proceedings 2003. Springer, pp 253-260
- Strumberger I., Bacanin N., Tuba M. 2017. "Constrained Portfolio Optimization by Hybridized Bat Algorithm".
- Suganya N. C., Vijayalakshmi Pai G. A. Constrained portfolio rebalancing with transaction costs using evolutionary wavelet hopfield network strategy. In: 2009 World Congress on Nature and Biologically Inspired Computing, NABIC 2009 - Proceedings, 2009. pp 451-456. doi:10.1109/NABIC.2009.5393347
- Sun J., Fang W., Wu X., Lai C.-H., Xu W. 2011. "Solving the multi-stage portfolio optimization problem with a novel particle swarm optimization", *Expert Systems with Applications*, 38 (6), 6727-6735.
- Suthiwong D., Sodanil M. Cardinality-constrained Portfolio optimization using an improved quick Artificial Bee Colony Algorithm. In: Computer Science and Engineering Conference (ICSEC), 2016 International, 2016. IEEE, pp 1-4
- Talebi A., Molaei M. A., Sheikh M. J. Performance investigation and comparison of two evolutionary algorithms in portfolio optimization: Genetic and particle swarm optimization. In: Proceedings - 2010 2nd IEEE International Conference on Information and Financial Engineering, ICIFE 2010, 2010. pp 430-437. doi:10.1109/ICIFE.2010.5609394
- Tan L., Niu B., Lin F., Duan Q., Li L. 2013. "Modified bacterial foraging optimization for constrained portfolio optimization", *Information Technology Journal*, 12 (23), 7918-7921.
- Tan L., Niu B., Wang H., Huang H., Duan Q. (2014) Bacterial foraging optimization with neighborhood learning for dynamic portfolio selection. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol 8590 LNBI. doi:10.1007/978-3-319-09330-7_48
- Tan Y., Zhu Y. 2010. "Fireworks algorithm for optimization", *Advances in swarm intelligence*, 355-364.
- Tang J., Wang K., Zhao X. A novel constraint handling technique for complex portfolio selection. In: 6th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2009, 2009. pp 156-160. doi:10.1109/FSKD.2009.537
- Thomaidis N. S. (2010) Active Portfolio Management from a Fuzzy Multi-objective Programming Perspective. In: Chio C, Brabazon A, Caro GA et al. (eds) Applications of Evolutionary Computation: EvoApplications 2010: EvoCOMNET, EvoENVIRONMENT, EvoFIN, EvoMUSART, and EvoTRANSLOG, Istanbul, Turkey, April 7-9, 2010, Proceedings, Part II. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 222-231. doi:10.1007/978-3-642-12242-2_23
- Tuba M., Bacanin N. 2014a. "Artificial bee colony algorithm hybridized with firefly algorithm for cardinality constrained mean-variance portfolio selection problem", *Appl Math Inf Sci*, 8 (6), 2831-2844.
- Tuba M., Bacanin N. Upgraded firefly algorithm for portfolio optimization problem. In: Proceedings - UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, UKSim 2014, 2014b. pp 113-118. doi:10.1109/UKSim.2014.25
- Tuba M., Bacanin N., Pelevic B. 2013. "Framework for constrained portfolio selection by the firefly algorithm", *International Journal of Mathematical Models and Methods in Applied Sciences*, 7 (10), 888-896.

- Tuba M., Bacanin N., Pelevic B. 2014. "Krill Herd (KH) algorithm applied to the constrained portfolio selection problem", *International Journal of Mathematics and Computers in Simulation*, 8 (1), 94-102.
- Venkataraman P. 2009. *Applied optimization with MATLAB programming*. John Wiley & Sons.
- Wang J.-b., Chen W.-N., Zhang J., Lin Y. (2015) A Dimension-Decreasing Particle Swarm Optimization Method for Portfolio Optimization. Paper presented at the Proceedings of the Companion Publication of the 2015 on Genetic and Evolutionary Computation Conference, Madrid, Spain,
- Wang W., Wang H., Wu Z., Dai H. A Simple and Fast Particle Swarm Optimization and Its Application on Portfolio Selection. In: *Intelligent Systems and Applications, 2009. ISA 2009. International Workshop on, 2009. IEEE*, pp 1-4
- Wang Z., Liu S., Kong X. 2012. "Artificial bee colony algorithm for portfolio optimization problems", *International Journal of Advancements in Computing Technology*, 4 (4), 8-16.
- Woodside-Oriakhi M., Lucas C., Beasley J. E. 2011. "Heuristic algorithms for the cardinality constrained efficient frontier", *European Journal of Operational Research*, 213 (3), 538-550.
- Xia Y., Liu B., Wang S., Lai K. K. 2000. "A model for portfolio selection with order of expected returns", *Computers & Operations Research*, 27 (5), 409-422.
- Xu F., Chen W. Stochastic portfolio selection based on velocity limited particle swarm optimization. In: *Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on, 2006. IEEE*, pp 3599-3603
- Xu F., Chen W., Yang L. Improved particle swarm optimization for realistic portfolio selection. In: *Proceedings - SNPD 2007: Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007. pp 185-190. doi:10.1109/SNPD.2007.313*
- Xu J., Lam A. Y. S., Li V. O. K. 2011. "Stock portfolio selection using chemical reaction optimization", *World Academy of Science, Engineering and Technology*, 77, 458-463.
- Xu R. T., Zhang J., Liu O., Huang R. Z. An estimation of distribution algorithm based portfolio selection approach. In: *Proceedings - International Conference on Technologies and Applications of Artificial Intelligence, TAAI 2010, 2010. pp 305-313. doi:10.1109/TAAI.2010.57*
- Yaakob S. B., Watada J. 2010. "A hybrid particle swarm optimization approach to mixed integer quadratic programming for portfolio selection problems", *International Journal of Simulation: Systems, Science and Technology*, 11 (5), 68-74.
- Yang X.-S. 2010a. "A new metaheuristic bat-inspired algorithm", *Nature inspired cooperative strategies for optimization (NICSO 2010)*, 65-74.
- Yang X. S. 2010b. "Firefly algorithm", *Engineering Optimization*, 221-230.
- Yao X., Liu Y., Lin G. 1999. "Evolutionary programming made faster", *Evolutionary Computation, IEEE Transactions on*, 3 (2), 82-102.
- Yi H., Yang J. Multi-objective portfolio optimization based on fuzzy genetic algorithm. In: *Proceedings - 9th International Conference on Computational Intelligence and Security, CIS 2013, 2013. pp 90-94. doi:10.1109/CIS.2013.26*
- Yin X., Ni Q., Zhai Y. (2015a) A novel particle swarm optimization for portfolio optimization based on random population topology strategies. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol 9140. doi:10.1007/978-3-319-20466-6_18
- Yin X., Ni Q., Zhai Y. A novel PSO for portfolio optimization based on heterogeneous multiple population strategy. In: *2015 IEEE Congress on Evolutionary Computation, CEC 2015 - Proceedings, 2015b. pp 1196-1203. doi:10.1109/CEC.2015.7257025*
- Yu L., Wang S., Lai K. K. 2009. "Multi-attribute portfolio selection with genetic optimization algorithms", *INFOR*, 47 (1), 23-30.
- Zaheer H., Pant M. Solving portfolio optimization problem through differential evolution. In: *International Conference on Electrical, Electronics, and Optimization Techniques, ICEEOT 2016, 2016. pp 3982-3987. doi:10.1109/ICEEOT.2016.7755462*
- Zhang X. L., Zhang W. G., Xu W. J., Xiao W. L. 2010. "Possibilistic Approaches to Portfolio Selection Problem with General Transaction Costs and a CLPSO Algorithm", *Computational Economics*, 36 (3), 191-200.
- Zhou J., Li J. An improved multi-objective particle swarm optimization for constrained portfolio selection model. In: *11th International Conference on Service Systems and Service Management, ICSSSM 2014 - Proceeding, 2014. doi:10.1109/ICSSSM.2014.6874155*
- Zhu H., Chen Y., Wang K. (2010) *Swarm Intelligence Algorithms for Portfolio Optimization*. In: Tan Y, Shi Y, Tan KC (eds) *Advances in Swarm Intelligence: First International Conference, ICSI 2010, Beijing, China, June 12-15, 2010, Proceedings, Part I*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 306-313. doi:10.1007/978-3-642-13495-1_38

- Zhu H., Wang Y., Wang K., Chen Y. 2011. "Particle Swarm Optimization (PSO) for the constrained portfolio optimization problem", *Expert Systems with Applications*, 38 (8), 10161-10169.
- Zitzler E., Künzli S. Indicator-based selection in multiobjective search. In: *International Conference on Parallel Problem Solving from Nature*, 2004. Springer, pp 832-842
- Zitzler E., Laumanns M., Thiele L., Zitzler E., Zitzler E., Thiele L., Thiele L. (2001) SPEA2: Improving the strength Pareto evolutionary algorithm. Eidgenössische Technische Hochschule Zürich (ETH), Institut für Technische Informatik und Kommunikationsnetze (TIK),

TÜBİTAK
PROJE ÖZET BİLGİ FORMU

Proje Yürütücüsü:	Yrd. Doç. Dr. CAN BERK KALAYCI
Proje No:	214M224
Proje Başlığı:	Eleman Sayısı Kısıtlı Portföy Optimizasyonu Problemi İçin Etkin Algoritmaların Tasarlanması
Proje Türü:	3501 - Kariyer
Proje Süresi:	24
Araştırmacılar:	OLCAY POLAT
Danışmanlar:	AŞKINER GÜNGÖR
Projenin Yürütüldüğü Kuruluş ve Adresi:	PAMUKKALE Ü. MÜHENDİSLİK F. ENDÜSTRİ MÜHENDİSLİĞİ B.
Projenin Başlangıç ve Bitiş Tarihleri:	15/04/2015 - 15/04/2017
Onaylanan Bütçe:	155950.0
Harcanan Bütçe:	112475.64
Öz:	<p>Portföy seçimi problemi, farklı kısıtlar altında finansal riski minimize ederek mümkün olan en iyi getiri elde etmek için portföyde bulundurulacak olan varlıkların oranlarını belirleme işlemidir. Matematiksel ifadeyle, ikinci dereceden amaç fonksiyonu ile lineer kısıtlar altında gerçek değerli değişkenlerin optimizasyonu problemidir. Sermayenin varlıklara dağıtılması üzerine portföydeki varlık sayısı sınırlanması ve her varlığa yatırılan sermayenin oranlarına alt ve üst limitler koyulması gibi birçok kısıt uygulanabilir. Bu kısıtlar ele alındığında, literatürde probleme eleman sayısı kısıtlı portföy optimizasyonu adı verilmiş ve bu problemin NP-Zor olduğu ispatlanmıştır. Bu proje önerisinde, portföy optimizasyonu problemine uygulanmamış olan tekniklerden, karınca kolonisi optimizasyonu, yapay arı kolonisi optimizasyonu ve değişken komşuluk arama teknikleri bu projede ilk defa portföy optimizasyonu problemini çözmek için uyarlanmıştır. Daha etkin algoritmaların geliştirilmesine ihtiyaç duyan bu problem için özel olarak geliştirilen alternatif çözüm yaklaşımları finans alanına öncelikle metodolojik olarak katkı sağlanmıştır. Ayrıca, literatürdeki önerilen mevcut yöntemler, şu ana kadar sadece gelişmiş ülkelerdeki varlık endekslerine uygulanmıştır. Ancak, bu projede hem mevcut yöntemler hem de geliştirilen yöntemler Türkiye'deki BIST 30 ve BIST 100 endekslerine de uygulanmış ve uyarlanan metotların karşılaştırmalı analizleri sunulmuştur. Bu sayede, Türkiye piyasalarındaki endekslerine ait veri setleri de uluslararası literatüre kazandırılmıştır. Bu proje sayesinde, teorik altyapısı başarıyla tamamlanarak geliştirilen metodolojiler, yakın gelecekte finans alanındaki uygulama çalışmalarında bir yer bulabilecektir. Ayrıca, bu projede geliştirilen algoritmaların, proje ekinde de detayları verilen, proje kapsamında düzenlenen portföy optimizasyonu çalıştayında, ilerideki aşamalarda kazançlı ticari bir yazılıma dönüştürülerek yatırımcılara bir karar destek sistemi sağlayabileceği ispatlanmıştır. Nitekim, böyle bir yazılım, değişken piyasa şartlarında yatırımcıya güvenilir bir yön gösterebilir. Böylece, finans alanında faaliyet gösterebilecek olan yeni bir şirket için zemin hazırlanmıştır. Şu andan itibaren, projenin yürütüldüğü üniversitenin finans sektöründe aktif rol alması ve daha fazla işbirliği sağlaması mümkün olabilmektedir. Son olarak, elde edilen proje çıktıları sayesinde, proje yürütücüsü kariyerini bir seviye daha yükselterek yetiştirdiği mezun öğrencilerin gururuyla doçentlik unvanına bir adım daha yaklaşmıştır.</p>
Anahtar Kelimeler:	Portföy optimizasyonu, karınca kolonisi, değişken komşuluk arama, yapay arı kolonisi
Fikri Ürün Bildirim Formu Sunuldu Mu?:	Hayır
Projeden Yapılan Yayınlar:	1- Portföy Yönetiminde Sezgisel Algoritmalar: Bir BIST 30 Uygulaması (Bildiri - Ulusal Bildiri - Sözlü Sunum), 2- AN APPLICATION OF ARTIFICIAL BEE COLONY ALGORITHM TO CARDINALITY CONSTRAINED PORTFOLIO OPTIMIZATION PROBLEM (Bildiri - Uluslararası Bildiri - Sözlü Sunum),