T.C.
İSTANBUL ÜNİVERSİTESİ CERRAHPAŞA
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

DOKTORA TEZİ

ÇOK KAYNAKLI WEBER PROBLEMİ İÇİN REVİZE AĞIRLIKLI BULANIK C-ORTALAMALAR VE GÜÇLENDİRİLMİŞ WEISZFELD HİBRİT YÖNTEMİ

M. IŞIK GÜZELGÖZ

ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

ENDÜSTRİ MÜHENDİSLİĞİ

DANIŞMAN

PROF. DR. ŞAKİR ESNAF

İSTANBUL-2019

Bu çalışma 19/12/2019 tarihinde aşağıdaki jüri tarafından Endüstri Mühendisliği Anabilim Dalı Endüstri Mühendisliği programında Doktora olarak kabul edilmiştir.

Tez Jürisi:

İmza
Prof.Dr. Şakir Esnaf (Danışman)
İstanbul Cerrahpaşa Üniversitesi
Mühendislik Fakültesi

İmza
Prof.Dr. Mehmet Mutlu Yenisey
İstanbul Cerrahpaşa Üniversitesi
Mühendislik Fakültesi

İmza
Prof.Dr. Ferhan Çebi
İstanbul Cerrahpaşa Üniversitesi
Mühendislik Fakültesi

İmza
Prof.Dr. Alp Baray
İstanbul Cerrahpaşa Üniversitesi
Mühendislik Fakültesi

İmza
Doç.Dr. Hür Bersam Bolat
İstanbul Teknik Üniversitesi
İşletme Mühendisliği

Endüstrileşmenin özellikle 20 yy. başlarından itibaren hızla yaygınlaşmasıyla birlikte hem üretim miktarı artmış hem de üretilen ürün ve/veya hizmetlerin daha verimli bir şekilde ihtiyaç sahiplerine iletilmesi zorunluluğu ortaya çıkmıştır. Süreçteki maliyetlerin azaltılması çalışmaları, üretim ve hizmet yerlerinin belirlenmesinin daha hassas yollarla yapılmasına yol açmıştır. Çalışmalar kapsamında çeşitli disiplinlerdeki araştırmacılar konuya eğilmiş ve yeni birçok yöntem geliştirmişlerdir.

Bu tez kapsamında, üretim yerinin coğrafi konumunun dağıtım maliyetlerini en aza indirgeyecek şekilde yerleştirebilme problemine yeni bir yöntem sunulmaktadır. Hibrit bir yöntem olan bu yeni öneri, yine aynı amaçla kullanılan iki ve üç farklı algoritmanın birleşmesinden oluşmaktadır.

Gerek bu tez çalışmasında gerekse doktora eğitimim boyunca benden desteğini esirgemeyen değerli hocam Prof. Dr. Şakir Esnaf'a teşekkürlerimi sunarım.

Tez boyunca bana hep olumlu destek veren eşime ve özellikle de oyun zamanını aldığım oğlum Uzay'a sonsuz teşekkür ederim.

Ekim 2019 M Işık GÜZELGÖZ

İÇİNDEKİLER

Sayfa No

İÇİNDEKİLER **4**
TABLO LİSTESİ **6**
ŞEKİLLER LİSTESİ: **8**
SİMGE VE KISALTMA LİSTESİ **9**
ÖZET **10**
SUMMARY **12**
1. GİRİŞ **14**
1.1. KATKILAR **15**
1.2. TEZİN ANA HATLARI **16**
2. GENEL KISIMLAR **18**
2.1. TYS PROBLEMİ **18**
2.2. PROBLEM TANIMI **20**
2.2.1. WEBER PROBLEMİ 21
2.2.2. ÇOK TESİSLİ WEBER PROBLEMİ (ÇTWP) 21
2.2.3. ÇOK KAYNAKLI WEBER PROBLEMİ (ÇKWP) 23
2.3. LİTERATÜR TARAMASI **25**
2.4. REVİZE AĞIRLIKLI BULANIK C-ORTALAMALAR YÖNTEMİ **29**
2.4.1. ÖBEKLEME ANALİZİ 30
2.4.2. BULANIK C-ORTALAMALAR (BCO) YÖNTEMİ 31
2.5. WEISZFELD ALGORİTMASI **34**
2.5.1. WEISZFELD ALGORİTMASI ÖRNEĞİ 36
2.6. MODİFİYE WEISZFELD ALGORİTMASI **39**
2.6.1. MODİFİYE WEISZFELD ALGORİTMASI ÖRNEĞİ 39
2.7. GÜÇLENDİRİLMİŞ WEISZFELD ALGORİTMASI **43**
2.7.1. PARABOLOID YAKLAŞIM 43
2.7.2. OPTİMUM – TALEP NOKTASI EŞLEMESİ 46

2.7.3. ALGORİTMA 46
2.7.4. GÜÇLENDİRİLMİŞ WEISZFELD ALGORİTMASI ÖRNEĞİ 47
2.8. NELDER-MEAD ALGORİTMASI 52
2.8.1. GENEL BİLGİLER: 53
2.8.2. NELDER-MEAD ALGORİTMASININ ADIMLARI: 54
3. MALZEME VE YÖNTEM 57
3.1. TANIMLAMA 57
3.1.1. İKİLİ ALGORİTMA: 58
3.1.2. ÜÇLÜ ALGORİTMA: 60
4. UYGULAMA 63
4.1. RABCO VE WEISZFELD HİBRİT YÖNTEMLERİ UYGULAMASI SONUCU 64
4.2. RABCO VE NELDER-MEAD HİBRİT ALGORİTMASI: 73
4.3. RABCO–WEISZFELD VE RABCO–NM HİBRİT ALGORİTMALARI: . 75
4.4. RABCO, WEISZFELD VE kombinasyonları ile Nelder-Mead ALGORİTMALARININ BİRLİKTE KULLANIMI 78
5. TARTIŞMA VE SONUÇ 84
6. KAYNAKLAR 89
7. EKLER 94
8. ÖZGEÇMİŞ 95

TABLO LİSTESİ

Sayfa No

Tablo 2.1: Örnek Weiszfeld algoritmasında ikinci adım sonucunda elde edilen değerler.....36

Tablo 2.2: Örnek Weiszfeld algoritmasında dördüncü adım sonucunda elde edilen değerler.....37

Tablo 2.3: Weiszfeld algoritması örnek uygulama sonuçları.....38

Tablo 2.4: Modifiye Weiszfeld Algoritması- örnek uygulamada ikinci adım sonuçları.....40

Tablo 2.5: Modifiye Weiszfeld Algoritması- örnek uygulamada dördüncü adım sonuçları.....40

Tablo 2.6: Modifiye Weiszfeld algoritması örnek uygulama sonuçları.....41

Tablo 2.7: Güçlendirilmiş Weiszfeld Algoritması – örnek uygulama sonuçları (λ=1).....49

Tablo 2.8: Güçlendirilmiş Weiszfeld Algoritması – örnek uygulama sonuçları (λ=1.8).....50

Tablo 2.9: Örnek uygulama sonuçları.....51

Tablo 4.1: BCO, BCO&COG, PSO ve RABCO yöntemleri ile yeni yöntemin farklı Weiszfeld kombinasyonlarıyla yapılmış uygulama sonuçlarının karşılaştırılması.....64

Tablo 4.2: RABCO – Weiszfeld (λ=1) algoritmasının kullanıldığı önerilen yöntemin sonuçlarının diğer algoritmalara kıyasla elde edilen yüzdelik iyileşmeler.....66

Tablo 4.3: RABCO ve yeni yöntemin farklı Weiszfeld kombinasyonlarıyla yapılmış uygulama sonuçları.....69

Tablo 4.4: Önerilen yöntemin farklı kombinasyonları ile elde edilen sonuçların RABCO yönteminin sonuçları arasındaki farklar.....71

Tablo 4.5: Yeni yöntemin farklı Weiszfeld kombinasyonlarıyla yapılmış uygulama sonuçlarında, RABCO yöntemi sonuçlarına göre olan iyileşmelerin yüzdelik karşılaştırılması.....72

Tablo 4.6: RABCO ve Nelder-Mead Hibrit Algoritması sonuçları.....74

Tablo 4.7: RABCO–NM ve RABCO–WEISZFELD algoritma sonuçları.....75

Tablo 4.8: RABCO-Güçlendirilmiş Weiszfeld (λ=1,8) ve RABCO-Nelder-Mead algoritmalarının sonuçlarının karşılaştırılması.....76

Tablo 4.9: RABCO, Nelder-Mead ve Güçlendirilmiş Weiszfeld (λ=1.8) algoritmalarının birlikte kullanımı sonuçları....78

Tablo 4.10: RABCO, Güçlendirilmiş Weiszfeld (λ=1.8) ve Nelder-Mead algoritmalarının birlikte kullanımı sonuçları....80

Tablo 4.11 Üçlü algoritmanın karşılaştırılması....82

Tablo 4.12 İyileştirme oranları....83

Tablo 5.1: İkili ve Üçlü Kombinasyonların RABCO yöntemine göre iyileştirmelerinin karşılaştırması....87

Tablo 5.2 Üçlü Kombinasyonların RABCO-NM yöntemine göre iyileştirmelerinin karşılaştırması....87

ŞEKİLLER LİSTESİ:

Şekil 3-1 RABCO - WEISZFELD ALGORİTMASI AKIŞ DİAGRAMI.....59

Şekil 3-2 RABCO - WEISZFELD - NELDER-MEAD ALGORİTMASI AKIŞ DİAGRAMI..59

SİMGE VE KISALTMA LİSTESİ

Simgeler	Açıklama
λ	: Modifiye Weiszfeld algoritmasında kullanılan sabit
U	: Bulanık C-Ortalamalar algoritmasında kullanılan üyelik matrisi
ε	: Algoritma durdurma kriteri sabit sayısı

Kısaltmalar	Açıklama
ABCO	: Ağırlıklı Bulanık C-Ortalamalar
ALA	: En yakın merkeze göre sınıflandırma (Alternate Location-Allocation)
BCO	: Bulanık C-Ortalamalar
COG	**:** Ağırlık Merkezi Algoritması
ÇTYS	: Çok Kaynaklı Tesis Yeri Seçimi
FCM	: Fuzzy C-Means
FCM&COG	: Fuzzy C-Means ve Ağırlık Merkezi hibrit yöntemi
K-Ort	: K Ortalamalar Metodu
PSO	: Parçacık Sürü Optimizasyonu
RABCO	: Revize Ağırlıklı Bulanık C-Ortalamalar
RWFCM	: Revised Weighted Fuzzy C-Means
TKTYS	: Tek Kaynaklı Tesis Yeri Seçimi
ÇTWP	: Çok Tesisli Weber Problemi
ÇKWP	: Çok Kaynaklı Weber Problemi
TYS	: Tesis Yeri Seçimi

ÖZET

DOKTORA TEZİ

ÇOK KAYNAKLI WEBER PROBLEMİ İÇİN REVİZE AĞIRLIKLI BULANIK C-ORTALAMALAR VE GÜÇLENDİRİLMİŞ WEISZFELD HİBRİT YÖTEMİ

M. IŞIK GÜZELGÖZ

İstanbul Üniversitesi Cerrahpaşa

Lisansüstü Eğitim Enstitüsü

Endüstri Mühendisliği A.B.D

Danışman: Prof. Dr. Şakir ESNAF

Bu çalışmada kapasite kısıtsız çok kaynaklı Weber problemlerinin çözümlerinde kullanmak üzere yeni bir hibrit yöntem önerilmiştir. Yöntem, Revize Ağırlıklandırılmış Bulanık C-Ortalamalar (Esnaf ve Küçükdeniz, 2013), Weiszfeld (1937), Modifiye Weiszfeld (Drezner, 1992), Güçlendirilmiş Weiszfeld (Drezner, 2015) ve Nelder-Mead (Nelder ve Mead, 1965) algoritmalarından oluşmaktadır.

Geliştirilen hibrit algoritmasında bahsi geçen algoritmalar ikili ve üçlü gruplar halinde kullanılmaktadır.

Önerilen yöntemle elde edilen sonuçların hem bu yöntemin bileşenlerini oluşturan algoritmalardan hem de bu tür problemlerde kullanılan diğer yöntemlerle elde edilenlerden daha iyi olduğu gösterilmiştir. Çalışmamız bu yöntemleri birleştiren ilk çalışmadır.

Ekim 2019, 95 sayfa.

Anahtar kelimeler:

Çok Kaynaklı Weber Problemi, Weiszfeld Algoritması, Güçlendirilmiş Weiszfeld Algoritması, Revize Ağırlıklı Bulanık C-Ortalamalar Algoritması, Nelder-Mead Algoritması.

SUMMARY

Ph.D. THESIS

REVISED WEIGHTED FUZZY C-MEANS and FORTIFIED WEISZFELD HYBRID METHOD for MULTISOURCE WEBER PROBLEM

M. IŞIK GÜZELGÖZ

Istanbul University Cerrahpaşa

Institute of Graduate Studies

Industrial Engineering Department

Supervisor: Prof. Dr. Sakir ESNAF

In this thesis, a new hybrid method will be proposed to be used for the solutions of the uncapacitated Weber problems. It is proposed a new combination of the algorithm Revised Weighted Fuzzy C-Means (RWFCM) proposed by Esnaf and Küçükdeniz (2013), the Fortified Weiszfeld algorithm proposed by Drezner (2015) and Nelder-Mead algorithm developed by Nelder and Mead (1965).

The RWFCM is a specific form of the Weighted Fuzzy C-Means proposed by Bezdek (1981), Tsekouras (2005) and Tsekouras et al. (2005). The Fortified Weiszfeld Algorithm is a modification of Weiszfeld algortihm that was proposed by Weiszfeld (1937).

Results of our proposed hybrid method we obtained show that the newly proposed sequentially-run method gives better than the previous methods. This study is the first one that explains hybrid use of Revised Weighted Fuzzy C-Means, Fortified Weiszfeld and Nelder-Mead algorithms.

October 2019, 95 pages

Keywords:

Multi-source Weber Problem, Weiszfeld and Fortified Weiszfeld algorithms, Revised Weighted Fuzzy C-Means algorithm, Nelder-Mead algorithm.

1. GİRİŞ

Endüstrinin değişik alanlarında maliyet optimizasyonu yapmak, özellikle yaşadığımız çağda gittikçe önem kazanan bir konu olmaktadır. Özel veya kamu şirketleri, gerek kendilerine doğrudan veya kısmen bağlı alt birimlerine, gerekse müşterilerine verecekleri hizmet ve/veya ürünlerini en düşük maliyetle ve en hızlı şekilde iletmek istemektedirler. Bunun için de acente, şube gibi ara birimler, depo alanları gibi ara istasyonlar kurup, tedarik hizmetlerini bu noktalardan yapmayı tercih etmektedirler. Örneğin, ulusal çapta beyaz eşya üreten bir fabrika ürünlerini bayilerine doğrudan fabrikasından değil, merkezi yerlerde konuşlandırdığı depolardan iletmeyi tercih etmektedir. Banka, sigorta şirketi gibi hizmet kurumları müşterilerine daha rahat ulaşabilmek için bölge müdürlükleri, şube müdürlükleri veya ATM gibi otomat makineler kullanmaktadır. Benzer durum acil hizmet gerektiren sağlık, itfaiye vb. kuruluşlar için de geçerli olup kaza veya yangın yerine ulaşımı en kısa zamanda yapabilmek amacıyla şehrin belli noktalarına ambulans veya itfaiye araçları istasyonları konuşlandırmaktadır.

Tüm bunlarda temel amaç ürün veya hizmetin, üretilen yerden (kaynak noktası veya noktaları) tüketilecek yerlere (talep noktaları) en az maliyetle iletiminin sağlanmasıdır. Maliyetin belirlenmesi ürünün yapısına göre değişmektedir. Örneğin beyaz eşya depolarından bayilere olan ürün iletiminde maliyet "nakliye masrafları" olurken, sağlık hizmetlerindeki maliyet "zaman" olmaktadır.

Alman mühendis Alfred Weber, 20. Yüzyılın başlarında yazdığı On The Location of Industries (Weber, 1909) isimli kitabında, endüstriyel komplekslerin yerleşim yerlerinin belirlenmesinin, toplam maliyet içindeki önemli bir başlık olan ürünün iletimine bağlı maliyetin minimize edilmesinde hayati önem taşıdığını vurgulamıştır. Weber çalışmasında, iki farklı noktada satış yapan bir firmanın üretim yerinin nasıl belirlenebileceğini incelemiştir. Tesis yerini belirlerken toplam nakliyat maliyetini minimum yapacak bir model ortaya koymuştur (Jamshidi, 2009).

Tesis Yeri Seçimi (TYS) problemi olarak adlandırılan bu problem, Weber'in çalışmasının ardından öncelikle teorik alanda birçok araştırmacının ilgisini çekmiş ve çeşitli çözüm yöntemler geliştirilmiştir. Sorunun doğası gereği çok fazla hesaplama yapılması ihtiyacı yüzünden pratik çalışmalar ancak 1960'lı yıllarda bilgisayar teknolojisinin gelişimi ile birlikte hız kazanmıştır.

Günümüzde hem endüstriyel komplekslerin yaygınlaşması hem de lojistik faaliyetlerin artması nedeniyle TYS problemleri büyük/küçük birçok organizasyon için hayati önem kazanmıştır.

1.1.KATKILAR

Bu çalışmada özel bir Tesis Yeri Seçimi (TYS) problemi olarak sınıflandırılan "Çok Kaynaklı Sürekli ve Kapasite Kısıtsız TYS Problemi" için yeni bir hibrit yöntem önerilmektedir. Yeni yöntemde Revize Ağırlıklandırılmış Bulanık C-Ortalamalar (RABCO) (Esnaf ve Küçükdeniz, 2012), Modifiye Weiszfeld Algoritması (Ostresh, 1978, Drezner, 1992), Güçlendirilmiş Weiszfeld Algoritması (Drezner, 2015) ve Nelder-Mead Algoritması (Nelder-Mead,1965) birlikte kullanılacaktır.

Zadeh(1965) tarafından geliştirilen "bulanık mantık" 1970'lerden itibaren öbekleme analizinde kullanılmış ve zamanla çeşitli uygulamaları üretilmiştir. Bunlardan Bulanık C-Ortalamalar (BCO) (Bezdek, 1981) yöntemi özellikle 1990lardan itibaren çeşitli şekillerde öbekleme analizi çalışmalarında yer almıştır. Tez kapsamında bu metodun genelleştirilmiş bir hali olan RABCO (Esnaf ve Küçükdeniz, 2012) metodu kullanılmıştır.

Weiszfeld algoritması Fermat-Weber problemi için geliştirilen gradyan temelli bir çözümdür (Weiszfeld, 1937). Algoritmanın dayandığı teorik temeller, 1960'lardan itibaren yapılan çalışmalarla geliştirilmiş ve algoritma daha genel durumlar için uygulanabilir hale getirilmiştir. Bu çalışmalara paralel olarak algoritmanın daha hızlı

çalışması, başka deyişle daha az sayda adımla optimum noktaya erişebilir olması için de araştırmalar yürütülmüştür. Bu amaçla üretilen Modifiye Weiszfeld Algoritması (Ostresh, 1978 ve Drezner, 1992), Güçlendirilmiş Weiszfeld Algoritması (Drezner, 2015) tez kapsamında kullanılmıştır.

Tez çalışmasında bu iki farklı alanda geliştirilmiş algoritmalar birlikte kullanılarak Çok Kaynaklı TYS problemi için optimum çözümler elde edilmiştir.

Tez kapsamında başka bir optimizasyon yöntemi olan Nelder-Mead algoritması üzerinde de çalışılmış ve bu algoritmanın kullanıldığı hibrit yöntem araştırılmıştır.

1.2. TEZİN ANA HATLARI

Tezin ikinci bölümünde, TYS probleminin tanımı, sınıflandırılması ve çözümünde kullanılan yöntemlerle ilgili teorik bilgiler verilmiştir.

Aynı bölümde, tez konusu olan hibrit algoritmanın paydaşları, Weiszfeld algoritması (Weiszfeld, 1937), Modifiye Weiszfeld algoritması (Obresh, 1978 ve Drezner, 1992), Güçlendirilmiş Weiszfeld algoritması (Drezner, 2015) anlatılmış ve bu algoritmaların çalışma prensiplerini açıklayan örnek bir uygulama da verilmiştir. Yine aynı bölümde Revize Ağırlıklı Bulanık C-Ortalamalar (RABCO) algoritması (Esnaf ve Küçükdeniz, 2012) ve son olarak Nelder-Mead türevsiz optimizasyon algoritması ayrıntılı olarak açıklanmıştır.

Ayrıca literatür taraması da yine bu bölümde verilmiştir.

Tezin üçüncü bölümünde ise öneri konusu olan RABCO-Weiszfeld ikili algoritmaları ve RABCO-Weiszfeld-Nelder-Mead üçlü hibrit algoritmaları anlatılmıştır. Burada hibrit algoritmaların adımları ve kullanılış şekli, algoritma sıraları verilmiştir.

Dördüncü bölümde ise yeni metodun bilinen veri kümeleri üzerindeki uygulama sonuçları verilmiştir. Yeni hibrit metotta Weiszfeld algoritmasının üç farklı seçeneği kullanılmış ve bunların karşılaştırılması yapılmıştır.

Burada kullanılan örnek veri kümeleri, sağlıklı bir karşılaştırma yapılabilmesi amacıyla, RABCO metodunun (Esnaf ve Küçükdeniz, 2012) uygulandığı veri kümeleri ile aynı seçilmiştir.

Algoritmanın farklı kombinasyonları ile elde edilen sonuçlar hem birbirileriyle hem de RABCO metodu (Esnaf ve Küçükdeniz, 2012) ile elde edilen sonuçlarla karşılaştırılmıştır.

Bu bölümde son olarak RABCO-Weiszfeld algoritmaları-Nelder-Mead ve RABCO-Nelder-Mead- Weiszfeld algoritmalarının iki farklı sıra ile hibritlenmiş versiyonları aynı veri kümelerine uygulanmış ve sonuçları karşılaştırmalı olarak verilmiştir.

Tezin sonuç bölümünde RABCO ve Weiszfeld algoritmalarının birleştirilmesinden elde edilen sonuçlarla, RABCO, Weiszfeld ve Nelder-Mead algoritmalarının birleştirilmesinden elde edilen sonuçlar karşılaştırılarak önerilen hibrit yöntemin kazanımları verilmiştir.

Tezin sonunda ayrıca bir kaynakça bölümü bulunmaktadır.

2. GENEL KISIMLAR

Bu bölümde TYS problemi ile ilgili bilgiler verilmiş, tez konusu hibrit algoritmanın bileşenleri olan algoritmalar örneklerle incelenmiştir. Ayrıca TYS ile ilgili çalışmaları içeren literatür başlığı eklenmiştir.

2.1. TYS PROBLEMİ

TYS problemi genel olarak, n adet tesisin m adet konuma ($n < m$), taşıma maliyetlerinin minimize edilecek şekilde, yerleştirilmesi problemidir (Tavakkoli ve Shayan, 1998). Daha detaylı olarak; belli sayıda hizmet veren tesisin (kaynak noktaları), kısıt olmaksızın veya bazı kısıtlar altında, müşterilerin (talep noktaları) taleplerini minimum düzeyde maliyet olacak şekilde uygun konumlara yerleştirilmesi ve her bir müşterinin hizmet veren tesislere atanması kapsayan problemleridir.

Alfred Weber, "Über den Standort der Industrien" adlı kitabında (Weber, 1909), bir endüstriyel yapının kurulum maliyeti haricindeki maliyetlerini azaltma problemini nakliye masraflarını minimize edecek şekilde konuşlandırılması problemine indirgemiş ve böylelikle TYS problemlerinin endüstrinin hemen her alanında optimizasyon çalışmalarının odağında olmasına yol açmıştır. Weber kitabında, tek bir ürün üretip bunu iki farklı noktada satışa sunan bir işletmeyi ele almıştır. Analizinde bu üç noktanın doğrusal olmadığını varsaymıştır (Kuhn, H.W. , Kuenne, R.E. , 1962). Çözüm olarak Fermat problemini genelleştirerek kullanmıştır.

Fermat problemi, Fransız matematikçi Pierre de Fermat'ın yazdığı bir mektupta yer alan "verilen üç noktaya olan uzaklıkları toplamı minimum olacak şekilde dördüncü bir nokta nasıl belirlenir" sorusuna dayanır. İlk olarak İtalyan matematikçi Torricelli

tarafından sadece pergel ve cetvel kullanılarak bulunan çözümdeki bu özel nokta "Torricelli noktası" olarak adlandırılmıştır. Zaman içinde birçok matematikçi ve mühendisin ilgisini çeken problem için çeşitli çözüm yöntemleri geliştirilmiştir. Cavalieri 1647'de bu özel noktayı üçgenin köşelerine bağlayan doğruların arasındaki açının 120° olduğunu göstermiştir. Simpson 1755'de problemin dual haline bir çözüm getirmiştir. 1800'lerde özellikle Franz Heinen ve Jacob Steiner'ın çalışmaları problemin optimizasyon problemi olarak ele alınmasını sağlamıştır (Drezner, Z., Hamacher, H., 2002).

20.yy da ise konuya ilk eğilen kişi Weber (1909) olmuştur. Tüm bu çalışmaların etkisiyle zaman içinde soru, "Fermat Problemi", "Genelleştirilmiş Fermat Problemi", "Fermat-Torricelli Problemi", "Steiner Problemi", "Genelleştirilmiş Steiner Problemi", "Steiner-Weber Problemi", "Fermat-Weber Problemi", "Weber Problemi", "1-medyan problemi", "p-medyan problemi", "minisum problemi" gibi adlar almıştır.

Weber problemi özünde, yerleri bilinen belirli sayıdaki talep noktası için, bu taleplerin karşılanacağı bir adet kaynak noktasının yerinin belirlenmesidir.

Uygulama alanlarının çok çeşitli olması sorunun farklı durumlar içermesine ve birbirinden farklı çözüm yöntemlerinin geliştirilmesine yol açmıştır. Dolayısıyla genel olarak TYS problemi olarak adlandırılan bu problemleri çözmek için zengin bir literatür ortaya çıkmıştır. Çözüm yöntemlerindeki zenginlik sınıflandırılma yapılmasını da etkilemiştir. Örneğin Daskin (1995) TYS problemlerini 10 farklı sınıfa, Sule (2001) ise 5 farklı sınıfa ayrılabileceğini belirtmiştir.

Sınıflandırmalar öncelikle kaynak sayısına bağlı olarak yapılmaktadır: Tek Kaynaklı TYS problemlerinde (TKTYS) bir kaynak yerinden ihtiyaç duyulan tüm noktaların taleplerinin karşılanacağı varsayılmaktadır (Weber problemi). Aynı şekilde Çok Kaynaklı TYS problemlerinde (ÇKTYS) ise talep noktalarının ihtiyaçlarının birden fazla sayıdaki kaynak yerinden karşılanacağı kabul edilmektedir (İyigün C. ve Ben-Israel A., 2012). Çok Kaynaklı TYS problemleri NP zorluğunda olduğundan çözümlerinde genellikle Tek kaynaklı TYS problemlerine indirgenmektedir (Esnaf ve Küçükdeniz, 2013).

Diğer önemli bir sınıflandırma talep noktalarının tanımına göre yapılmaktadır: Sürekli (düzlemsel), Süreksiz (ayrık) veya network (ağ yapısı) çeşitlerine göre farklı çözüm yöntemleri bulunmaktadır.

Çok Kaynaklı TYS problemi için önemli diğer bir ayırım da kaynak noktalarının kapasite sınırı ile ilgilidir. Eğer bazı kaynak noktaları kendilerine gelecek tüm talebi karşılamıyorlarsa bu problem Kapasite Kısıtlı Çok Kaynaklı TYS problemi olarak; eğer tüm kaynaklar kendilerine yapılacak tüm talepleri karşılayabiliyorsa Kapasite Kısıtsız Çok Kaynaklı TYS problemi olarak adlandırılır. İki problem farklı çözüm yöntemleri gerektirmektedir.

Bu tez kapsamında Kapasite Kısıtsız Çok Kaynaklı TYS problemi ele alınacaktır.

Maliyet hesaplaması, iki nokta arasındaki öklidyen mesafe ile talep miktarı dikkate alınarak belirlenecektir.

2.2. PROBLEM TANIMI

Sürekli düzlemde Kapasite Kısıtsız Tesis Yer Seçimi Problemine Weber Problemi (WP) adı da verilmektedir. Weber Problemi, sürekli düzlemde, yerleri belirli olan talep noktalarına hizmet verecek, kapasiteleri sınırlı olmayan bir tesisin yerini optimal şekilde belirlemekle ilgilidir. Weber Probleminde amaç, kurulacak olan bir adet tesis için Öklid uzayında en uygun yeri bularak toplam taşıma maliyetini en küçüklemektir.

Birden çok tesisin kurulması söz konusu olduğunda probleme Çok Tesisli Weber Problemi (ÇTWP) adı verilir. Bu problemde tesislerin kapasite kısıtı göz önüne alınmamaktadır.

2.2.1. WEBER PROBLEMİ

Kapasite kısıtsız sürekli yer seçimi ve tahsis problemi (Uncapacitated Continuous Location-Allocation Problem) olarak da bilinen, bir düzlem üzerindeki a_i noktalarında bulunan müşterilerin w_i miktarındaki taleplerini en az toplam ulaştırma maliyeti ile karşılamak üzere kurulacak olan tek bir tesisin yerini (X) belirlemeyi amaçlayan Weber problemi aşağıdaki gibi ifade edilmektedir. Bu, sınırlandırılmamış lineer olmayan bir optimizasyon problemidir.

WP:

$$\min_{X} \sum_{i=1}^{n} w_i d(x, a_i) \qquad (2.1)$$

öyle ki

$$w_i \geq 0, \quad w_i \in \mathbb{R}, \quad \forall i$$

$$x \in \mathbb{R}^2$$

$\boldsymbol{a_i}$ *:* (a_{i1}, a_{i2}) *talep noktalarının bilinen konumları* $\boldsymbol{i}$*=1,...,n*

$\boldsymbol{x}$ *: tesis konumunu [* $\boldsymbol{x}$ *:* $(\boldsymbol{x_1}, \boldsymbol{x_2})$ *] gösteren karar değişkenleri*

$\boldsymbol{w_i}$ *: talep noktası* $\boldsymbol{i}$*'nin toplam talep miktarı*

$\| \boldsymbol{x} - \boldsymbol{a_i} \|$ *: talep noktası* $\boldsymbol{i}$ *ile tesis arasındaki Öklid uzaklık*

$$\|x - a_i\| = \sqrt{(x_1 - a_{i1})^2 + (x_2 - a_{i2})^2}$$

2.2.2. ÇOK TESİSLİ WEBER PROBLEMİ (ÇTWP)

Weber Probleminin gelişmiş hali olan Çok Tesisli Weber Problemi (ÇTWP) bir düzlem üzerindeki a_i noktalarında bulunan müşterilerin w_i miktarındaki taleplerini en az toplam ulaştırma maliyeti ile karşılamak üzere birden çok sayıda tesis kurulması ile

ilgilenir. Tesislerin kapasiteleri sınırsız kabul edilir. Her tesisin yeri ve her müşterinin hangi tesisten ne kadar hizmet alacağı belirlenecektir. Weber Probleminde (WP) olduğu gibi tesislerin kurulum maliyeti toplam maliyete dahil edilmez, amaç fonksiyonu olarak sadece toplam ulaştırma maliyeti dikkate alınır. Kaç tesis kurulacağı baştan verilmiştir. ÇTWP tesis kapasitelerinin toplam talep miktarından çok büyük olduğu durumlarla ilgilidir. Tesis kapasiteleri sınırsız olduğu için optimal çözümde her bir müşterinin talebi kendisine en yakın olan tesise atanır ve her müşteri birer tesisten hizmet alır. ÇTWP lineer olmayan, ne içbükey, ne de dışbükey bir hedef fonksiyona sahiptir ve genellikle çok sayıda yerel minimum nokta içerir.

ÇTWP:

$$\min_{Z,X} \sum_{i=1}^{n} \sum_{j=1}^{m} w_i z_{ij} \left\| x_j - a_i \right\| \qquad (2.2)$$

öyle ki

$$\sum_{j=1}^{m} z_{ij} = 1 \qquad i = 1, \dots, n \qquad (2.3)$$

$$z_{ij} \in \{0, 1\}, \quad \forall i, j \qquad (2.4)$$

$$x_j \in \mathbb{R}^2, \quad \forall j$$

m *: kurulacak tesis sayısı*

n *: müşteri (talep noktası) sayısı*

$a_i : (a_{i1}, a_{i2})$ *talep noktalarının bilinen iki boyutlu konumları* $i=1,\dots,n$

$X=(x_1,\dots,x_m)$ *: tesis konumlarını [* $x_j : (x_{j1}, x_{j2})$ *] gösteren karar değişkenleri* $j=1,\dots,m$

w_i *: talep noktası* ***i****'nin toplam talep miktarı*

$Z = (z_{ij})$ *: talep noktası* ***i****'nin tesis* ***j****'den hizmet alıp almayacağını gösteren karar değişkenleri (*eğer *i*nci müşteri *j*nci tesis atanmışsa 1; atanmamışsa 0 değerini alır).

|| x_j - a_i || : *talep noktası* ***i*** *ile tesis* ***j*** *arasındaki Öklid uzaklık*

(1) Her talep noktasının toplam talebi karşılanacaktır. (Talep kısıtı)

(2) $z_{ij} = 0$: Talep noktası ***j*** tesis ***i***'den hizmet almayacak.
$z_{ij} = 1$: Talep noktası ***j*** tesis ***i***'den w_i kadar miktarda hizmet alacak.

İlk kısıt (1) (talep kısıtı) müşterilerin taleplerinin tamamen karşılanması gerektiğini belirtir. Talep noktalarının tesislere atanmasını gösteren zij değerleri, tesislerin kapasite kısıtı olmamasından dolayı, her talep noktasını kendisine en yakın tesise atayacak şekilde belirlenir ve bir talep noktası sadece bir tesisten hizmet alır. Bir talep noktası birden çok tesise aynı uzaklıkta ise, bu tesislerden herhangi birisine rastgele atanır. (Tunçbilek, 2012).

2.2.3. ÇOK KAYNAKLI WEBER PROBLEMİ (ÇKWP)

Salhi ve Gamal (2003) ve Drezner ve diğ. (2016) problemi aşağıdaki gibi yeniden formüle etmiş ve çok kaynaklı Weber problemi olarak adlandırmıştır. Problem yukarıdaki notasyona göre aşağıda verilmiştir. Gamal ve Salhi (2003) ve Küçükdeniz ve Esnaf (2018) problemi müşteri taleplerine göre iki alt gruba ayırmışlardır.

ÇKWP:

$$\min_{X} \sum_{i=1}^{n} \sum_{j=1}^{m} w_{ij} \left\| x_j - a_i \right\| \qquad (2.5)$$

öyle ki

$$\sum_{j=1}^{m} w_{ij} = w_i \qquad i = 1, \dots, n \qquad (2.6)$$

$$w_{ij} \geq 0 \qquad , \forall i \; ve \; \forall j \qquad (2.7)$$

$$x_j \in \mathbb{R}^2, \quad \forall j$$

w_{ij} : *j tesisinden talep noktası* ***i****'ye gönderilecek talep miktarı*

W: *nm boyutlu* w_{ij} *karar değişkenleri kümesi*

Bu tez kapsamında ele alınan ÇKWP problemi aşağıdaki varsayımlara göre çözülmüştür. (Lozano, 1998 ve Esnaf, Küçükdeniz, 2012):

1- Kaynak noktaları düzlemde herhangi bir noktada yer alabilir.
2- Kaynak noktaları arasında aktarım olmayacaktır.
3- Her talep noktasının ihtiyacı sadece bir adet kaynak tarafından karşılanacaktır. Başka deyişle talep noktasından yapılan talep miktarı birden fazla sayıda kaynağa bölünmeyecektir.
4- Nakliye masrafları öklidyen uzaklıklarla orantılı kabul edilecektir.
5- Her talep noktası kendisine en yakın kaynak noktasından ihtiyacını karşılayacaktır.
6- Kurulum masrafları göz önüne alınmayacaktır.
7- Talep noktaları düzlemde sabit koordinatlarda bulunacak ve talep miktarları da değişken olmayacaktır.

Yukarıdaki formülde her talep noktası a_i kendisine en yakın (maliyeti en düşük) kaynak noktasına X_j eşleşecektir. Dolayısıyla talep noktaları kümesi, merkezlerinde birer adet kaynak noktası olan alt kümelere ayrılmış olmaktadır. Formüldeki C_j bu alt kümeleri belirtmektedir.

Maliyet hesaplamasında yer alan d metrik fonksiyonu çeşitli uygulamalarda farklı şekiller almaktadır. Genel olarak l_p metrik kullanılmaktadır. $u, v \in \mathbb{R}^n$ herhangi iki nokta olmak üzere bu noktalar arasındaki l_p metrik aşağıdaki formülle verilir:

$$l_p(u,v) = [|u_1 - v_1|^p + |u_2 - v_2|^p + \cdots + |u_n - v_n|^p]^{1/p} \ , p \geq 1 \qquad (2.4)$$

l_p metriği $p = 1$ için Manhattan metriği, $p = 2$ için öklidyen metrik, $p = \infty$ için ise Hilbert metriği adını alır. Bunlar dışında l_p metriği kullanarak farklı formüller de oluşturulabilmektedir. Örneğin, $l_p(u,v) = l_p(u,u) + l_p(v,v)$ şeklinde tanımlanan "British Rail" metriği İngiltere tren seferleri yapısından üretilmiştir.

Bu çalışmada aşağıdaki şekilde tanımlanan ağırlıklı l_2 öklidyen metrik dikkate alınmıştır:

$$l_2(u,v) = w_0[|u_1 - v_1|^2 + |u_2 - v_2|^2]^{1/2} \qquad (2.5)$$

2.3. LİTERATÜR TARAMASI

Weber tarafından önü açılan Tesis Yeri Seçim problemleri zamanla çok çeşitli uygulama alanları bulmuş ve buna bağlı olarak farklı çözüm yöntemleri üretilmiştir. Uygulama alanı ve çözüm yöntemi zenginliği nedeniyle, yukarıda bahsedildiği üzere bu yöntemler çok çeşitli şekillerde sınıflandırılmaktadır.

Tezin bu aşamasında özellikle tez konusu olan Sürekli ÇTYS problemlerine ait çalışmalara ilişkin bir özet verilmiştir.

ÇTYS tipi problemlerin çözümlerinde genellikle, önce alt öbekler oluşturulur ardından her alt öbek için bir TKTYS problemi çözülür. İlk defa Miehle (1958) tarafından önerilen bu yöntem ile Weber probleminin genelleştirilmesi sağlanmıştır. Cooper (1963), Çoklu-Weber problemini tanımlamış, problemde yer alan maliyet fonksiyonun konkav veya konveks olmadığını bu nedenle çok sayıda yerel minimum noktasını olduğunu göstermiştir. Cooper (1964) Weber problemi için sezgisel ve algoritmik bir metot olan ALA (Alternate Location-Allocation) metodunu geliştirmiştir.

Kuhn ve Kuenne (1962) Weber problemi için Weiszfeld algoritmasını yeniden kurmuşlardır. Kaynak noktasının herhangi bir talep noktasına çok yakın olduğu durumlarda algoritmanın yakınsama problemi yaşadığı biliniyordu. Bu nedenle özellikle algoritmanın yakınsama problemi üzerinde durulmuştur. Kuhn (1973), algoritmadaki adımlarda talep noktasına rastlanılmadığı sürece, algoritmanın her zaman optimum sonuca ulaşacağını ispatlamıştır. Aynı çalışmada, algoritmanın optimum sonuca ulaşamayacağı başlangıç noktalarının "sayılabilir çoklukta" olduğunu ileri sürmüş ancak Chandrasekaran ve Tamir (1989) bunun yanlış olduğunu gösteren bir karşı örnek bulmuşlardır. Brimberg (1995) yakınsama problemi yaşanan noktaların hangi şartlarda olduğunu veren teoremi öne sürmüştür. Cánovas, ve diğ. (2002) tüm bunların ayrıntılı bir incelemesini yapmışlardır (Beck, A., Sabach, S., 2010).

Yakınsama probleminin yanı sıra aynı dönemde Weiszfeld algoritmasını daha hızlı çözüme ulaştıracak yöntemler üzerinde de durulmuştur. Algoritma özellikle kaynak noktası ile talep noktası birbirlerine yakın olduğunda oldukça yavaş çalışmaktadır. Ostresh (1978) bunun çözümü için algoritmadaki iki ardışık adım arasında, bir sonraki adım yerine arada kalan uygun bir noktanın kullanılarak çözüme daha hızlı ulaşılabileceğini göstermiştir. Chen (1984) bundan faydalanarak *radyal maliyet* fonksiyonu oluşturmuş ve daha hızlı çözüm elde etmiştir. Drezner (1992) aynı metodu kullanarak önerdiği Modifiye Weiszfeld algoritması ile sonuca daha hızlı ulaşılabileceğini göstermiştir.

Taillard (1988) CLS, LOPT ve DEC yöntemlerinin bileşkesinden oluşan bir yöntem geliştirdi. Bu yöntem özellikle talep noktası sayısı çok fazla olan kümelere uygulanabilmektedir.

İyigün ve Ben-İsrael (2010) Weiszfeld algoritmasını uzaklığa bağlı olasılık dağılımını içerecek şekilde genelleştirilen bir yöntem önermişlerdir.

Drezner (2015) aynı düşünceyi kullanarak Güçlendirilmiş Weiszfeld algoritmasını önermiştir. Bu yöntemle optimum noktaya erişmek için Weiszfeld algoritmasını daha az sayıda çalıştırmanın yeterli olacağını göstermiştir. Bu yöntemde paraboloid yaklaşım metodunu kullanmıştır.

Weber probleminin ayrık uzayda uygulanmış şekli olan p-medyan problemi Hakimi (1964) tarafından tanımlanmış ve ayrıntılı incelenmiştir. P-medyan problemlerinde *p* kaynak noktası yerleri belli *n* farklı konumdan birine yerleştirilmektedir. P-medyan probleminin kombinatoryal NP-zorlukta bir optimizasyon problemi olduğu Cornuejols ve diğ. (1977) ve Hakimi (1979) tarafından ispatlanmıştır (Bastı, 2012). Bunun özel bir hali olan 1-medyan probleminin ise NP zorluğu bilinmemektedir (Hakimi, 2000).

Sürekli Çok Kaynaklı TYS problemi aynı zamanda bir öbekleme problemi olarak ele alınabilir. Bu nedenle öbekleme analizinde kullanılan yöntemlerden de faydalanılmaktadır. Öbekleme çalışmalarında sıklıkla kullanılan K-Ortalamalar (McQuenn, 1967) yöntemini Bulanık Mantık (Zadeh, 1965) teorisiyle birleştiren *Bulanık C-Ortalamalar yöntemi* Dunn (1973) ve Bezdek (1981) tarafından geliştirilmiş ve özellikle 1980'lerden sonra Weber probleminin çözümünde de kullanılmıştır.

Looney (2002), Gauss fonksiyonu ile hesaplanan ağırlıkların kullanıldığı modifiye ağırlıklı beklenen değer fonksiyonuna dayalı bir bulanık öbekleme algoritması önermiştir. Bu algoritma iki çok bilinen veri kümesinde, IRIS ve Wisconsin Göğüs Kanseri veri kümelerinde, uygulanmış ve Bulanık C-Ortalamalara göre daha iyi sonuç vermiştir.

Eschrich ve diğ. (2003), hızlı ama hatasız çalışan bir bulanık öbekleme algoritması oluşturmuştur. brBCO isimli bu algoritmada öbek sayısı, öbeklenecek verileri belli kutulara ayırarak azaltılmaktadır. İlk adımda verinin azaltılması için nicemleme yapılır. İkinci adımda ise kümeleme yapılır. Bunun için benzer özellikteki vektörel büyüklükler aynı ağırlıklı örneğe konulur.

Wang ve diğ. (2004), Ağırlıklı Bulanık C-Ortalamalar (ABCO) algoritması adını verdikleri, azalan gradyan temelli bir yöntem önermişlerdir. Yöntemde ağırlıklar amaç fonksiyonu gradyan temelli tırmanma tekniği ile minimize edilerek belirlenmekte ve sabit kalmaktadır.

D'urso and Giordani (2006), bulanık veriyi karşılaştırmak için "ağırlıklı benzemezlik" metriği kullanan bir Bulanık Öbekleme modeli önermiştir. Modelde "merkez" uzaklığı

ve “yayılma” uzaklığı kullanılmıştır. Modelde ağırlıklar sabit olmayıp her adımda değişebilmektedir.

Tsekouras (2005) öbek merkezinin ağırlığını, öbekteki eleman sayısına bağlı olarak belirlemiştir. Ağırlıklı Bulanık C-Ortalamalar yöntemi bu şekilde uygulandığında, her öbek merkezinin öbeklerin belirlenmesine olan katkısı kendi öbeğinin büyüklüğü ile belirlenmiş olmaktadır.

Tsekouras ve diğ. (2005), hiyerarşik bulanık öbeklemeye dayalı bir bulanık modelleme önermişlerdir. Metodları iki aşamalı olup ilkinde en yakın komşuluk prensibi ile veri kümesi belli sayıda alt öbeğe ayrılır. Ardından bunların merkezleri ABCO ile iyileştirilir. Modeldeki algoritmanın her adımında, bulanık öbekleme üyelik değerleri, öbek merkezlerinin ağırlıklarının hesaplanmasında kullanılırlar.

Turmchokkassam ve Mitaim (2006), Ağırlıklı Bulanık C-Ortalamalar (ABCO) yöntemini kullanarak akustik dengeleme filtresi önermiştir. Burada ABCO kullanımı her öbek için ayrı bir ağırlık elde edilebilmesini sağlamaktadır.

Hore ve diğ. (2007) BCO yöntemindeki amaç fonksiyonunu modifiye ederek ağırlıkların etkisini hesaplamalara dahil etmiştir. Ağırlıklar Tsekouras’ın (2005) çalışmasında olduğu gibi hesaplanmıştır.

2000’lerden başlayarak tek bir yöntem veya bu yöntemin özel/modifiye bir halini kullanmak yerine birden fazla sayıda algoritmayı içeren hibrit yöntemler geliştirilmiştir.

Gamal ve Salhi (2003) ÇTYS problemindeki kaynak noktalarının yerlerini, talep noktalarını belli hücrelere ayırıp ardından Cooper (1964) tarafından önerilen ALA yöntemi ile iyileştirme yapılarak belirleyen bir metot geliştirmişlerdir. Ardından aynı araştırmacılar, hücresel yöntem yerine Genetik Algoritmanın (GA) kullanıldığı bir hibrit metodu geliştirmişleridir (Salhi ve Gamal, 2003). Burada tesis yerlerinin koordinatları GA ile elde edilmekte ardından ALA yöntemi ile iyileştirilmektedir.

Levin ve Ben-Israel (2004) tek kaynaklı TYS için kullanılabilecek Newton-Bracket yöntemi ile Cooper (1964) tarafından geliştirilen En Yakın Merkeze göre sınıflandırma

(ALA) yöntemlerinden oluşan hibrit bir metot geliştirmiştir. Bu yöntemle optimum sonuca Weiszfeld yöntemine göre daha hızlı erişilebileceğini örnekle göstermişlerdir.

Mehdizeded, E. ve diğ. (2008), BCO ile PSO algoritmalarını birlikte kullanmıştır. BPSO (Bulanık Parçacık Sürüsü Optimizasyonu) adlı bu yöntemde BCO ile bulunan öbek merkezleri PSO ile iyileştirilmektedir. Algoritmanın 8 bilinen veri seti üzerindeki denemelerinde PSO algoritmasına göre daha hızlı sonuçlar verdiği görülmektedir.

Esnaf ve Küçükdeniz (2009) talep noktalarının Bulanık C-Ortalamalar ile alt öbeklere ayrıldığı, ardından her alt öbeğe ait öbek merkezinin Ağırlık Merkezi (COG) ile iyileştirildiği hibrit bir metot geliştirmiştir. Bu algoritma kapasite sınırsız sürekli çok kaynaklı TYS problemlerinde kullanılmaktadır.

Kaur ve Kaur (2013) BCO ile Genetik Algoritmanın birleştirildiği hibrit bir yöntem kullanmışlardır. Bu yöntemi sistem güvenlik kontrollerinde kullanmışlar ve hem BCO hem de GA yöntemlerine göre daha iyi sonuçlar elde etmişlerdir.

Küçükdeniz ve Esnaf (2018), RABCO yöntemi ile Nelder-Mead algoritmasının birleştirilmesi ile elde edilen sonuçların RABCO yönteminden daha iyi olduğunu göstermiştir.

Şimdiye kadar yaptığımız incelemelerimizde BCO, Weiszfeld ve Nelder-Mead algoritmalarının birlikte kullanıldığı bir çalışmaya rastlanmamıştır.

2.4. REVİZE AĞIRLIKLI BULANIK C-ORTALAMALAR YÖNTEMİ

Tezde önerilen hibrit algoritmanın ilk adımı olarak Revize Ağırlıklı Bulanık C-Ortalamalar yöntemi (Esnaf ve Küçükdeniz, 2013) kullanılmıştır. Bu yöntem Öbekleme Analizi çalışmaları kapsamında üretilmiştir.

2.4.1. ÖBEKLEME ANALİZİ

Doğada karşılaştığımız nesneleri belli gruplara ayırarak onlar hakkında daha çok bilgi sahibi olabiliyoruz. Bu nedenle çok eski çağlardan beri nesneleri sınıflandırma çalışmaları yapmaktayız. Örneğin Aristo (M.Ö.350) ilk defa sistematik olarak canlıları sınıflandırmıştır. 9 ciltten oluşan Hayvan Tarihi (Historia Animalium) isimli kitabında Yunanistan, Anadolu ve Makedonya'da yaşayan 520 kadar hayvan türünün tanımını yaparak bunların yapıları ve davranışları hakkında bilgi vermiştir. Modern zamanlarda Linneasus (1707-1778), Lamarck (1744-1829), Buffon (1707-1788) ile başlayarak araştırmacılar, canlıların sınıflandırılmasına yeni kıstaslar getirmişlerdir.

Antik Yunanda başlayan maddenin özü çalışmalarında yine Aristo tarafından önerilen modelde tüm maddelerin toprak, su, hava, ateş olarak sınıflandırılması yapılmştır. Zamanla terk edilen bu modelin yerine Dalton'un maddeler atomlardan oluşur düşüncesi hakim olmuş ve ilk defa Mendeleyev (1869) tarafından Periyodik Elementler Tablosu aracılığıyla tüm madde çeşitlerinin sınıflandırılması yapılmıştır. Bunun bir sonucu olarak daha önce bilinmeyen ama tabloda olması gerektiği düşünülen Galyum, Skandiyum ve Germanyum gibi yeni elementlerin varlığı öngörülmüş ve araştırmacılar bunları doğada bulabilmiştir. Tablo bunların özelliklerini, keşiflerinden önce verebilmiştir.

20.yy da taksonomi başta olmak üzere birçok disiplinde sınıflandırma çalışmalarına önem verilmiş ve çok çeşitli yöntemler geliştirilmiştir. Zamanla "öbekleme analizi" adı ile yeni bir disiplin de ortaya çıkmıştır.

Öbekleme analizinde en çok kullanılan yöntem K-Ortalamalar yöntemidir. James McQueen (1967) tarafından önerilen bu yöntemde ana küme, *k* sayıda alt kümeye elemanlarının birbirlerine olan benzerlikleri/yakınlıkları kullanılarak ayrılmaktadır.

K-Ortalamalar yönteminde aşağıdaki adımlar takip edilmektedir.

1- Alt öbek merkezi olmaya aday *k* tane nokta belirlenir.
2- Ana kümedeki her nokta, yerleri belirlenmiş bu *k* tane noktadan kendisine en yakın olan noktayla eşleştirilir. Böylelikle alt kümeler elde edilmiş olur.

3- Her alt küme için yeni bir küme merkezi belirlenir. Burada aynı alt kümedeki noktaların geometrik ortası alınır.
4- Yukarıdaki 2.ve3.adımlar tekrarlanır.
5- Algoritma önceden belirlenmiş kriter(ler) karşılanana kadar devam ettirilir. Kriter olarak genellikle aşağıdaki seçeneklerden biri kullanılmaktadır:
    a. İki ardışık yinelemede elde edilen öbek merkezleri arasındaki toplam mesafe farkı önceden belirlenmiş bir sabit olan $\varepsilon > 0$ sayısından küçük olmalı;
    b. İki ardışık yinelemede elde edilen öbeklemeye göre hesaplanan toplam maliyet değerleri arasındaki fark önceden belirlenmiş bir sabit olan $\varepsilon > 0$ sayısından küçük olmalı;
    c. Yinelemeler önceden belirlenmiş bir sabit sayı $N > 0$ kadar tekrarlanır;

2.4.2. BULANIK C-ORTALAMALAR (BCO) YÖNTEMİ

Klasik küme tanımlamasında bir kümeye ait olma kavramı, kümenin içinde veya tamamen dışında olacak şekilde yapılmaktadır. Başka deyişle klasik küme teorisinde, bir A kümesi için $x \in A$ veya $x \notin A$ seçeneklerinden sadece bir tanesi doğrudur. Burada üyelik için $\{0,1\}$ sayıları kullanılır.

Zadeh (1965) tarafından geliştirilen Bulanık Mantıkta ise bir elemanın bir kümeye ait olması, daha flu tanımlanmıştır. Bir kümeye ait olma durumu $u \in [0,1]$ ile belirtilir. Bu sabit sayıya *üyelik derecesi* denir. Kümelerin bu şekilde tanımlanması, doğal dili daha iyi betimlediğinden, başta bilgisayar mühendisliği olmak üzere birçok disiplinde geniş kullanım alanı bulmuştur.

Bulanık mantıkta herhangi bir $A = \{a_1, a_2, a_3, \dots, a_n\} \subset \mathbb{R}^d$ kümesi k-adet C_j alt kümeye ayrıldığında a_i elemanları birden fazla sayıda alt kümeye ait olabilmektedir. Burada $a_i \in A$ elemanının C_j alt kümesine olan üyelik derecesi $u_{ij} \in [0,1]$ ile belirtilir. Tüm üyelik derecelerinin olduğu $n \times k$ matris ise üyelik matrisi olarak adlandırılır:

$$U = \left[u_{ij}\right]; \quad u_{ij} \in [0,1], \qquad 1 \le i \le n, \qquad 1 \le j \le k \tag{2.6}$$

Öbekleme analizinde ilk olarak Ruspini (1969) ve Dunn(1974) tarafından kullanılan bu düşünceye dayanarak Bezdek (1981) Bulanık C-Ortalamalar (BCO) yöntemini geliştirmiştir.

Bu yöntemde, $c_j \in C_j$ öbek merkezleri, $1 \leq p \leq \infty$ önceden belirlenmiş üyelik katsayısı olmak üzere, maliyet fonksiyonu, $J_p(U, c)$, aşağıdaki formülle tanımlanır.

$$J_p(U, c) = \sum_{i=1}^{n} \sum_{j=1}^{k} w_i (u_{ji})^p \left\| a_i - c_j \right\|^2 \tag{2.7}$$

Maliyet fonksiyonunda görünen $U = [u_{ji}]$ üyelik matrisini, $p \in (1, \infty)$ bulanıklaştırma katsyısını, w_i ise a_i talep noktasının talep miktarını belirtmektedir. Üyelik matrisi, $\sum_{j=1}^{k} u_{ji} = 1$ koşulunu $\forall i = 1 \ldots n$ için sağlar. Talep noktaları ile talep noktasının ait olduğu alt kümenin merkezine olan uzaklık $\left\| a_i - c_j \right\|$ metriği ile belirtilir.

Algoritmada aşağıdaki adımlar takip edilir:

1- Başlangıç olarak $U^{(0)} = [u_{ji}]$ üyelik matrisi rasgele atanır.

2- Her alt küme için öbek merkezi hesaplanır:

$$c_j = \frac{\sum_{i=1}^{n} u_{ij}^p a_i}{\sum_{i=1}^{n} u_{ij}^p} \tag{2.8}$$

3- Yukarıdaki formülle hesaplanan alt küme merkezlerine göre U üyelik matrisi tekrar hesaplanır:

$$u_{ij} = \frac{1}{\sum_{j=1}^{k} \left(\frac{a_i - c_i}{a_i - c_j} \right)^{\frac{2}{p-1}}} \tag{2.9}$$

4- Algoritma ardışık iki üyelik matrisi, $U^{(k)}, U^{(k+1)}$ kullanılarak elde edilen J_p değeri arasındaki fark $\varepsilon > 0$ sabitinden küçük olana kadar devam ettirilir: $\left\| J_p\left(U^{(k)}, c\right) - J_p\left(U^{(k+1)}, c\right) \right\| < \varepsilon$

Bu yöntemden faydalanılarak üretilen Revize Ağırlıklı Bulanık C-Ortalamalar (RABCO) metodunda talep noktalarındaki talep miktarı başlangıçta verildiği ve algoritma süresince sabit kaldığı varsayılır (Esnaf ve Küçükdeniz, 2013). Burada tedarik zincirinin, taleplerin sabit olarak belirlenebileceği ve kaynak noktalarının tüm talepleri karşılama özelliği kullanılmıştır.

RABCO yöntemi de BCO ile aynı maliyet fonksiyonunu kullanmaktadır. Farklı olarak BCO yönteminde ağırlıklar eşit kabul edilirken, RABCO yönteminde her talep noktası için farklı değer alabilmektedir. Burada tedarik zinciri yapısı öngörülmüş olup talep noktalarındaki talep miktarlarının algoritma süreci boyunca sabit kalacağı varsayılmıştır.

Dolayısıyla Revize Ağırlıklı Bulanık C-Ortalamalar algoritmasının adımları aşağıdaki şekilde verilir:

1- Algoritmanın başında aşağıdaki değerler/sabitler belirlenir:
   a. Kümenin ayrılacağı alt küme sayısı, k,
   b. Alt kümelerin öbek merkezleri, $\{c_1, c_2, \dots, c_k\}$
   c. Bulanıklaştırma katsayısı, p,
   d. Durdurma katsayısı, $\varepsilon > 0$.
2- BCO algoritmasının 3.adımı kullanılarak $u_{ij} \in [0,1]$ $(1 \le i \le n,\ 1 \le j \le k)$ değerleri hesaplanır.
3- Yeni alt öbek merkezleri, aşağıdaki formülle tekrar hesaplanır:

$$c_j{}^{yeni} = \frac{\sum_{i=1}^{n} w_i u_{ij}^{p} a_i}{\sum_{i=1}^{n} w_i u_{ij}^{p}} \tag{2.8}$$

4- Algoritma, $max_i \left\{ \left\| c_i - c_i^{yeni} \right\| \right\} < \varepsilon$ şartı sağlanıyorsa durdurulur, aksi halde 2. adımdan itibaren tekrarlanır.

RABCO algoritması, eğer tüm talep noktalarının talep miktarları eşit kabul edilirse, BCO algoritması ile aynı olacaktır.

2.5.WEISZFELD ALGORİTMASI

Weiszfeld (1937), Fermat-Weber probleminin çözümü için Sturm (1884) tarafından daha önce önerilmiş bir metodu kullanarak bir algoritma geliştirmiştir (Beck, A., Sabach, S., 2010). Burada fonksiyonunun gradyan değerinin optimum noktalarda 0 olacağı bilgisini kullanmıştır:

$$\nabla f(\vec{x}) = 0 \tag{2.10}$$

Yukarıda verilen maliyet fonksiyonu dikkate alındığında, $\wp = \{a_1, a_2, \dots, a_n\}$ olmak üzere:

$$\nabla f(\vec{x}) = \sum_{i=1}^{n} w_i \frac{\vec{x} - \boldsymbol{a_i}}{\|\vec{x} - \boldsymbol{a_i}\|} \ , \vec{x} \notin \wp \tag{2.11}$$

Eşitliği ağırlıklı l_2 öklidyen metrik tanımı kullanılarak elde edilir.

Weiszfeld (1937), optimum noktaya aşağıda kuralları verilen algoritmayla erişilebileceğini göstermiştir:

$$\overrightarrow{\boldsymbol{x}_{k+1}} = \frac{1}{\sum_{i=1}^{n} \frac{w_i}{\|\overrightarrow{\boldsymbol{x}_k} - \boldsymbol{a_i}\|}} \sum_{i=1}^{n} \frac{w_i \boldsymbol{a_i}}{\|\overrightarrow{\boldsymbol{x}_k} - \boldsymbol{a_i}\|} \tag{2.12}$$

Bu algoritmada ilk başlangıç noktası için uygun bir $\boldsymbol{x_1}$ noktası belirlemek gerekmektedir. Ayrıca, algoritmanın adımlarında karşılaşılan noktaların talep noktaları olmaması şartı vardır: $\vec{\boldsymbol{x}} \notin \{\boldsymbol{a_1}, \boldsymbol{a_2}, \boldsymbol{a_3}, \dots, \boldsymbol{a_n}\}$. Böyle durumlarda maliyet fonksiyonunun gradyanı hesaplanamaz.

Algoritma aşağıdaki şartlardan biri sağladığında durdurulur:

1- Önceden belirlenmiş bir adım sayısı kadar tekrarlandığında
2- $\varepsilon > 0$ sabit bir sayı olmak üzere, $d\ (\overrightarrow{\boldsymbol{x_k}}, \overrightarrow{\boldsymbol{x_{k+1}}}) < \varepsilon$ olduğunda
3- $\varepsilon > 0$ sabit bir sayı olmak üzere, $|\sum d(\overrightarrow{\boldsymbol{x_{k+1}}}, \overrightarrow{\boldsymbol{a_i}}) - \sum d(\overrightarrow{\boldsymbol{x_k}}, \overrightarrow{\boldsymbol{a_i}})| < \varepsilon$ olduğunda

Macar matematikçi Weiszfeld teoremini bir Japon matematik dergisinde Fransızca yayınlamış ancak gerek dönemin savaş koşulları gerekse optimum noktaya ardışık adımlarla ulaşma düşüncesinin yeni olması nedeniyle uzun süre dikkat çekmemiştir.

Algoritma, 1960'lı yıllarda tekrar keşfedilmiş ve özellikle Miehle (1958), Kuhn ve Kunne (1962), Cooper (1963) gibi araştırmacıların çabalarıyla birçok alanda uygulanabilir hale gelmiştir.

1960'larda algoritma yeniden keşfedildiğinde özellikle yakınsama problemi üzerinde durulmuştur. Problemin içeriğinde kaynak noktalarının talep noktalarından farklı olduğu varsayılmaktadır. Ancak algoritmadaki adımlarda talep noktalarına rastlanıp rastlanmayacağı kesin değildir. Kuhn ve Kuhne (1962) ve Kuhn(1973) bu soru üzerinde durmuş ve aşağıdaki önermeleri yapmışlardır:

a- Algoritmadaki adımlarda karşılaşılan noktalar talep noktası olmadığı sürece yakınsama mümkündür ve optimum noktaya erişilir.

b- Algoritmanın sonuç vermeyeceği bir yere varmak için seçilebilecek "kötü" başlangıç noktalarının sayısı sonlu veya sayılabilir sonsuzdur.

İlk önerme yine Kuhn (1973) tarafından ispat edilmiş ancak ikincisi için karşı örnek bulunmuştur (Chandrasekaran, R., Tamir, A., 1989).

2.5.1. WEISZFELD ALGORİTMASI ÖRNEĞİ

Aşağıdaki örneklemede bu algoritmayla TKTYS problemine nasıl bir çözüm bulunabileceği gösterilmiştir.

$\wp = \{(0,0), (1,0), (0,1), (1,4)\}$ talep noktaları kümesi olsun. Her nokta için talep miktarını veren Talep vektörünün ise $\boldsymbol{w_i} = (1, 1, 1, 4)$ olduğunu varsayalım. Kaynak noktasının yerini belirlemek için Weiszfeld algoritması aşağıdaki şekilde kullanılır:

1. adım: Algoritmanın ilk adımı olarak $\overrightarrow{\boldsymbol{x_1}} = (\boldsymbol{x_1}, \boldsymbol{y}_1)$ noktası belirlenir. Bunun için $\wp$ kümesinin ağırlıklı ortalaması alınır:

$$\overrightarrow{\boldsymbol{x_1}} = (\boldsymbol{x_1}, \boldsymbol{y}_1) = \frac{\sum \boldsymbol{w_i} * \boldsymbol{a_i}}{\sum \boldsymbol{w_i}} = \frac{[(0,0) + (1,0) + (0,1) + 4 * (1,4)]}{7} = \left(\frac{5}{7}, \frac{17}{7}\right) \quad (2.13)$$

2. adım: $\overrightarrow{\boldsymbol{x_1}}$ noktası ile talep noktaları $\boldsymbol{a_i}$ arasındaki mesafeler belirlenir:

Tablo 2.1: Örnek Weiszfeld algoritmasında ikinci adım sonucunda elde edilen değerler

$\boldsymbol{a_i}$	(a_{i_1}, a_{i_2})	$\boldsymbol{w_i}$	$\boldsymbol{w_i} * a_{i_1}$	$\boldsymbol{w_i} * a_{i_2}$	Δx	Δy	$d(x,y)$
$\boldsymbol{a_1}$	(0 , 0)	1	0	0	$\left(0 - \frac{5}{7}\right)$	$\left(0 - \frac{17}{7}\right)$	2.53
$\boldsymbol{a_2}$	(1 , 0)	1	1	0	$\left(1 - \frac{5}{7}\right)$	$\left(0 - \frac{17}{7}\right)$	2.45
$\boldsymbol{a_3}$	(0 , 1)	1	0	1	$\left(0 - \frac{5}{7}\right)$	$\left(1 - \frac{17}{7}\right)$	1.60
$\boldsymbol{a_4}$	(1,4)	4	4	16	$\left(1 - \frac{5}{7}\right)$	$\left(4 - \frac{17}{7}\right)$	1.60
$\sum$		7	5	17	$\overrightarrow{\boldsymbol{x_1}} = \left(\frac{5}{7}, \frac{17}{7}\right)$		

3. adım: $\boldsymbol{w_i}/d(\overrightarrow{\boldsymbol{x_1}}, \overrightarrow{\boldsymbol{a_i}})$ değerleri hesaplanır.

4. adım: $\boldsymbol{w_i a_i}/d(\overrightarrow{\boldsymbol{x_1}}, \overrightarrow{\boldsymbol{a_i}})$ değerleri hesaplanır.

Tablo 2.2: Örnek Weiszfeld algoritmasında dördüncü adım sonucunda elde edilen değerler

$\wp$	(a_{i_1}, a_{i_2})	$\boldsymbol{w_i}$	$d(\overrightarrow{\boldsymbol{x_1}}, \overrightarrow{\boldsymbol{a_i}})$	$\boldsymbol{w_i}/d(\overrightarrow{\boldsymbol{x_1}}, \overrightarrow{\boldsymbol{a_i}})$	$\boldsymbol{w_i} a_{i_2}/d(\overrightarrow{\boldsymbol{x_1}}, \overrightarrow{\boldsymbol{a_i}})$	$\boldsymbol{w_i} a_{i_2}/d(\overrightarrow{\boldsymbol{x_1}}, \overrightarrow{\boldsymbol{a_i}})$
$\boldsymbol{a_1}$	(0 , 0)	1	2.53	0.40	0	0
$\boldsymbol{a_2}$	(1 , 0)	1	2.45	0.41	0.41	0
$\boldsymbol{a_3}$	(0 , 1)	1	1.60	0.63	0	0.63
$\boldsymbol{a_4}$	(5 , 8)	4	1.60	2.50	2.50	10.02
$\sum$				3.93	2.91	10.64

5. adım: Bütün talep noktaları için 3. ve 4. adımlarda elde edilen değerler ayrı ayrı toplanır. Toplamlar algoritmadaki formüle uygun olarak bölündüğünde $\overrightarrow{\boldsymbol{x_2}}$ noktası elde edilir:

$$\overrightarrow{\boldsymbol{x_2}} = \left(\frac{2.91}{3.93}, \frac{10.64}{3.93}\right) = (0.74 , 2.71) \qquad (2.14)$$

6. adım: yukarıdaki adımlar $\overrightarrow{\boldsymbol{x_2}}$ için tekrarlanır ve $\overrightarrow{x_3}$ elde edilir.

7.adım: Ardışık iki iterasyonla elde edilen $\overrightarrow{\boldsymbol{x_k}}$ ve $\overrightarrow{\boldsymbol{x_{k+1}}}$ noktaları arasındaki uzaklık belli bir hata değerinin altına düşene kadar algoritma devam ettirilir.

Örnekteki süreç devam ettirildiğinde aşağıdaki sonuçlar elde edilmiştir:

Tablo 2.3: Weiszfeld algoritması örnek uygulama sonuçları

İterasyon sayısı	$\sum \boldsymbol{w_i} / d(\overrightarrow{\boldsymbol{x_k}}, \overrightarrow{\boldsymbol{a_i}})$	x_{k_1}	x_{k_2}	$d(\overrightarrow{\boldsymbol{x_k}}, \overrightarrow{\boldsymbol{x_{k+1}}})$	Maliyet: $\sum d(\overrightarrow{\boldsymbol{x_k}}, \overrightarrow{\boldsymbol{a_i}})$
1	3.93	0.71	2.42	-	12.9627
2	4.29	0.74	2.71	0.28	12.6635
3	4.87	0.79	2.95	0.25	12.3995
4	5.73	0.84	3.16	0.22	12.1718
5	6.94	0.87	3.34	0.18	11.9809
6	8.60	0.90	3.49	0.15	11.8247
7	10.87	0.92	3.61	0.12	11.6996
8	13.94	0.94	3.70	0.10	11.6011
9	18.09	0.96	3.77	0.07	11.5245
10	13.94	0.97	3.83	0.06	11.4657
11	31.23	0.98	3.87	0.04	11.4208
12	41.39	0.98	3.90	0.03	11.3868
13	55.07	0.99	3.93	0.02	11.3612
14	73.49	0.99	3.95	0.02	11.3420
15	98.27	0.99	3.96	0.01	11.3275

16	131.64	0.99	3.98	0.01	11.3168
17	176.55	1.00	3.98	0.01	11.3087

Bu örnekte 16. ve 17. adımlar arasında toplam maliyet farkı $|11.3168 - 11.3087| = 0.0081$ olarak hesaplanır.

2.6.MODİFİYE WEISZFELD ALGORİTMASI

Araştırmacılar bir yandan teorinin ispatında yer alan yakınsama sorununa çözümler üretirken bir yandan da algoritmanın daha hızlı sonuca ulaşmasını sağlayacak yöntemler geliştirmişlerdir.

Ostresh (1978) ve Drezner(1992) tarafından önerilen modifiye algoritmada, yeni adım noktası, ardışık iki nokta kullanılarak aşağıdaki formülle belirlenmektedir:

$$\bar{X} = X' + \lambda(X'' - X') \qquad (2.15)$$

Drezner(1992) tarafından kullanılan bu notasyonda X' ilk noktayı, X'' ise algoritmayla elde edilen ikinci noktayı betimler. Drezner, ikinci nokta olarak X'' yerine $\bar{X}$ noktasının kullanılmasının algoritmayı hızlandıracağını söyler. Ostresh(1978) bu formüldeki λ için $1 \leq \lambda \leq 2$ aralığında bir değer kullanıldığında algoritmanın yakınsayacağını ispatlamıştır. Drezner (1992) genellikle λ=1.8 için algoritmanın optimal sonucu verdiğini belirtmiştir.

2.6.1. MODİFİYE WEISZFELD ALGORİTMASI ÖRNEĞİ

Yukarıda verilen örneği Modifiye Weiszfeld algoritması kullanarak nasıl çözüleceği aşağıda verilmiştir. Bu örnekte λ=1.8 kullanılmıştır.

1. adım: Algoritmanın ilk adımı olarak $\overrightarrow{\boldsymbol{x_1}} = (\boldsymbol{x_1}, \boldsymbol{y_1})$ noktası belirlenir. Bunun için $\wp$ kümesinin ağırlıklı ortalaması alınır:

$$\overrightarrow{x_1} = (x_1 , y_1) = \frac{\sum w_i * a_i}{\sum w_i} = \frac{[(0,0) + (1,0) + (0,1) + 4 * (1,4)]}{7} = \left(\frac{5}{7}, \frac{17}{7}\right) \quad (2.16)$$

2. adım: $\overrightarrow{x_1}$ noktası ile talep noktaları a_i arasındaki mesafeler belirlenir:

Tablo 2.4: Modifiye Weiszfeld Algoritması- örnek uygulamada ikinci adım sonuçları.

a_i	(a_{i_1}, a_{i_2})	w_i	$w_i * a_{i_1}$	$w_i * a_{i_2}$	Δx	Δy	$d(x,y)$
a_1	(0 , 0)	1	0	0	$\left(0 - \frac{5}{7}\right)$	$\left(0 - \frac{17}{7}\right)$	2.53
a_2	(1 , 0)	1	1	0	$\left(1 - \frac{5}{7}\right)$	$\left(0 - \frac{17}{7}\right)$	2.45
a_3	(0 , 1)	1	0	1	$\left(0 - \frac{5}{7}\right)$	$\left(1 - \frac{17}{7}\right)$	1.60
a_4	(1,4)	4	4	16	$\left(1 - \frac{5}{7}\right)$	$\left(4 - \frac{17}{7}\right)$	1.60
$\sum$		7	5	17	$\overrightarrow{x_1} = \left(\frac{5}{7}, \frac{17}{7}\right)$		

3. adım: $w_i / d(\overrightarrow{x_1}, \overrightarrow{a_i})$ değerleri hesaplanır.

4. adım: $w_i a_i / d(\overrightarrow{x_1}, \overrightarrow{a_i})$ değerleri hesaplanır.

Tablo 2.5: Modifiye Weiszfeld Algoritması- örnek uygulamada dördüncü adım sonuçları

$\wp$	(a_{i_1}, a_{i_2})	w_i	$d(\overrightarrow{x_1}, \overrightarrow{a_i})$	$w_i / d(\overrightarrow{x_1}, \overrightarrow{a_i})$	$w_i a_{i_2} / d(\overrightarrow{x_1}, \overrightarrow{a_i})$	$w_i a_{i_2} / d(\overrightarrow{x_1}, \overrightarrow{a_i})$
a_1	(0 , 0)	1	2.53	0.40	0	0
a_2	(1 , 0)	1	2.45	0.41	0.41	0

$\boldsymbol{a_3}$	(0 , 1)	1	1.60	0.63	0	0.63
$\boldsymbol{a_4}$	(5 , 8)	4	1.60	2.50	2.50	10.02
$\sum$				3.93	2.91	10.64

5. adım: Bütün talep noktaları için 3. ve 4. adımlarda elde edilen değerler ayrı ayrı toplanır. Toplamlar algoritmadaki formüle uygun olarak bölündüğünde $\overrightarrow{\boldsymbol{x_2}}$ noktası elde edilir:

$$\overrightarrow{\boldsymbol{x_2}} = \left(\frac{2.91}{3.93} , \frac{10.64}{3.93}\right) = (0.74 , 2.71)$$

6. adım: $\overrightarrow{\boldsymbol{x_1}}$ ve $\overrightarrow{\boldsymbol{x_2}}$ kullanılarak yeni nokta belirlenir:

$$\bar{X} = \overrightarrow{\boldsymbol{x_1}} + \lambda(\overrightarrow{\boldsymbol{x_2}} - \overrightarrow{\boldsymbol{x_1}})$$

$$\bar{X} = \left(\frac{5}{7}, \frac{17}{7}\right) + \lambda * \left[(0.74 , 2.71) - \left(\frac{5}{7}, \frac{17}{7}\right)\right]$$

$$\bar{X} = (0.76 , 2.93)$$

7. adım: Yukarıdaki 3.-6. adımlar tekrarlanarak yeni noktalar elde edilir.

7.adım: Ardışık iki iterasyonla elde edilen $\overrightarrow{\boldsymbol{x_k}}$ ve $\overrightarrow{\boldsymbol{x_{k+1}}}$ noktaları arasındaki uzaklık belli bir hata değerinin altına düşene kadar algoritma devam ettirilir.

Örnekteki süreç devam ettirildiğinde aşağıdaki sonuçlar elde edilmiştir:

Tablo 2.6: Modifiye Weiszfeld algoritması örnek uygulama sonuçları

İterasyon sayısı	$\sum \boldsymbol{w_i} / d(\overrightarrow{\boldsymbol{x_k}}, \overrightarrow{\boldsymbol{a_i}})$	x_{k_1}	x_{k_2}	$d(\overrightarrow{\boldsymbol{x_k}}, \overrightarrow{\boldsymbol{x_{k+1}}})$	Maliyet: $\sum d(\overrightarrow{\boldsymbol{x_k}}, \overrightarrow{\boldsymbol{a_i}})$

1	3.93	0.71	2.43	-	12.9627
2	4.79	0.74	2.71	0.28	12.6635
3	8.55	0.90	3.48	0.25	11.8287
4	13.89	0.94	3.70	0.22	11.6023
5	23.90	0.97	3.83	0.18	11.4641
6	42.54	0.98	3.91	0.15	11.3841
7	77.24	0.99	3.95	0.12	11.3392
8	13.94	0.99	3.97	0.10	11.3146
9	18.09	1.00	3.98	0.07	11.3012
10	13.94	1.00	3.99	0.06	11.2939
11	31.23	1.00	4.00	0.04	11.2900
12	41.39	1.00	4.00	0.03	11.2879
13	55.07	1.00	4.00	0.02	11.2868
14	73.49	1.00	4.00	0.02	11.2862
15	98.27	1.00	4.00	0.01	11.2859
16	131.64	1.00	4.00	0.01	11.2858
17	176.55	1.00	4.00	0.01	11.2857

Bu örnekte 8. ve 9. adımlar arasında toplam maliyet farkı $|11.3146 - 11.3012| = 0.0134$ olarak hesaplanır.

Oysa ki, 16. ve 17. adımlar arasında toplam maliyet farkı $|11.2858 - 11.2857| = 0.0001$ olarak hesaplanır.

Buradan açıkça görülüyor ki Modifiye Weiszfeld algoritması ile optimum noktaya daha hızlı ulaşılacaktır.

2.7.GÜÇLENDİRİLMİŞ WEISZFELD ALGORİTMASI

Drezner (2015) tarafından önerilen Güçlendirilmiş Weiszfeld algoritmasında, maliyet fonksiyonunun ikinci derecede yaklaşımı kullanılarak çok daha hızlı bir şekilde optimum noktaya erişilmektedir.

2.7.1. PARABOLOID YAKLAŞIM

$F(x,y)$ 2-değişkenli bir fonksiyon olsun. F fonksiyonunun (x_0, y_0) noktası yakınındaki açılımı aşağıdaki formülle belirlendiğini fonksiyonel analizden biliyoruz:

$$F(x,y) \approx F(x_0,y_0) + \left.\frac{\partial F}{\partial x}\right|_{(x_0,y_0)} |x - x_0| + \left.\frac{\partial F}{\partial y}\right|_{(x_0,y_0)} |y - y_0| + \frac{1}{2}\left.\frac{\partial^2 F}{\partial x^2}\right|_{(x_0,y_0)} (x - x_0)^2 + \frac{1}{2}\left.\frac{\partial^2 F}{\partial y^2}\right|_{(x_0,y_0)} (y - y_0)^2 + \left.\frac{\partial^2 F}{\partial x \partial y}\right|_{(x_0,y_0)} |x - x_0||y - y_0| \quad (2.17)$$

Burada $(x_0, y_0) = (0,0)$ noktası için:

$$F(x,y) \approx F(0,0) + \left.\frac{\partial F}{\partial x}\right|_{(0,0)} |x| + \left.\frac{\partial F}{\partial y}\right|_{(0,0)} |y| + \frac{1}{2}\left.\frac{\partial^2 F}{\partial x^2}\right|_{(0,0)} (x)^2 + \frac{1}{2}\left.\frac{\partial^2 F}{\partial y^2}\right|_{(0,0)} (y)^2 + \left.\frac{\partial^2 F}{\partial x \partial y}\right|_{(0,0)} |x||y| \quad (2.18)$$

eşitliği elde edilir. Bu kuadratik yaklaşımı ikinci dereceden bir denklemle belirtebiliriz:

$$v_1x^2 + 2v_2xy + v_3y^2 + 2v_4\mathrm{x} + 2v_5\mathrm{y} + v_6 = 0 \qquad (2.19)$$

Denklemdeki v_i katsayılarını bulmak için $F(\mathrm{x},\mathrm{y})$ fonksiyonun

$$(\theta_j, \psi_j) \in \{(1,1),(1,-1),(-1,1),(-1,-1),(0,1),(0,-1),(1,0),(-1,0)\} \qquad (2.20)$$

noktalarındaki değerleri dikkate alınarak

$$f_j = F(\theta_j, \psi_j) - F(0,0) \qquad (2.21)$$

denklemleri elde edilir. Buradan elde edilecek denklemler vektörel formda $\bar{\boldsymbol{f}} = A\bar{\boldsymbol{v}}$ olarak yazılabilir. Eşitlikteki A matrisi aşağıda verilmiştir:

$$A = \begin{bmatrix} 1 & +2 & 1 & +2 & +2 \\ 1 & -2 & 1 & +2 & -2 \\ 1 & -2 & 1 & -2 & +2 \\ 1 & +2 & 1 & -2 & -2 \\ 0 & 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 0 & -2 \\ 1 & 0 & 0 & 2 & 0 \\ 1 & 0 & 0 & -2 & 0 \end{bmatrix} \qquad (2.22)$$

Eşitlikteki $\bar{\boldsymbol{v}}$ değerlerini bulmak için en küçük kareler yöntemi kullanılır. $(Av-f)^T(Av-f)$ ifadesinin v ye göre türevini alıp 0'a eşitlediğimizde $A^TAv = A^Tf$ eşitliği elde edilir. Böylece v vektörünü aşağıdaki şekilde elde ederiz:

$$v = (A^TA)^{-1}A^Tf \qquad \Rightarrow$$

$$\frac{d}{dv}(Av-f)^T(Av-f) = 0 \qquad \Rightarrow$$

$$\left[\frac{d}{dv}(Av-f)^T\right](Av-f) + (Av-f)^T\left[\frac{d}{dv}(Av-f)\right] = 0 \qquad \Rightarrow$$

$$A^T(Av-f) + (Av-f)^TA = 0 \qquad \Rightarrow$$

$$A^T(Av-f) + \left(A^T(Av-f)\right)^T = 0 \qquad \Rightarrow$$

$$A^T(Av-f) = 0 \qquad \Rightarrow$$

$$A^T A v = A^T f \qquad \Rightarrow$$

$$v = (A^T A)^{-1} A^T f \tag{2.23}$$

Eşitliği elde edilir. Buradaki $(A^T A)^{-1} A^T$ matrisi:

$$(A^T A)^{-1} A^T = \frac{1}{120}\begin{bmatrix} 12 & 12 & 12 & 12 & -24 & -24 & 36 & 36 \\ 15 & -15 & -15 & 15 & 0 & 0 & 0 & 0 \\ 12 & 12 & 12 & 12 & 36 & 36 & -24 & -24 \\ 10 & 10 & -10 & -10 & 0 & 0 & 10 & -10 \\ 10 & -10 & 10 & -10 & 10 & -10 & 0 & 0 \end{bmatrix} \tag{2.24}$$

2.23 formülünde yerine koyduğumuzda v için aşağıdaki değerleri buluruz:

$$v_1 = \frac{1}{120}(12f_1 + 12f_2 + 12f_3 + 12f_4 - 24f_5 - 24f_6 + 36f_7 + 36f_8) \tag{2.25}$$

$$v_2 = \frac{1}{120}(15f_1 - 15f_2 - 15f_3 + 15f_4)$$

$$v_3 = \frac{1}{120}(12f_1 + 12f_2 + 12f_3 + 12f_4 + 36f_5 + 36f_6 - 24f_7 - 24f_8)$$

$$v_4 = \frac{1}{120}(10f_1 + 10f_2 - 10f_3 - 10f_4 + 10f_7 - 10f_8)$$

$$v_5 = \frac{1}{120}(10f_1 - 10f_2 + 10f_3 - 10f_4 + 10f_5 - 10f_6)$$

2.25 denklemlerindeki 1/120 katsayısı ortak olduğundan çözümü etkilemeyecek olup göz önüne alınmayacaktır. Bu v_i değerleri aradığımız paraboloidi verecektir. Paraboloidin tepe noktası aşağıdaki formüllerle gösterilir:

$$\bar{x} = \frac{v_2 v_5 - v_3 v_4}{v_1 v_3 - {v_2}^2} \quad , \quad \bar{y} = \frac{v_2 v_4 - v_1 v_5}{v_1 v_3 - {v_2}^2} \tag{2.26}$$

2.7.2. OPTİMUM – TALEP NOKTASI EŞLEMESİ

Algoritmada elde edilen optimum kaynak noktası, mevcut talep noktalarından birine yakın bir konumda ise algoritma yavaşlayacak, bunun sonucu olarak çok sayıda adım çalıştırılarak optimum noktaya erişilebilecektir. Bu özellik Weiszfeld algoritmasının tanımından kaynaklanmakta olup Modifiye veya Güçlendirilmiş Weiszfeld algoritmaları için de geçerlidir. Drezner ve Simchi-Levi (1992) bunun olasılığının yaklaşık 1/n olduğunu göstermişlerdir. Dolayısıyla, optimum olarak elde edilen noktanın talep noktasına yakın olup olmadığını kontrol edilmesi önemlidir.

$\boldsymbol{a_i} = (x_i\ ,\ y_i\ )$ talep noktasının optimal olması için gerek ve yeter şart (Love ve diğ. (1988), Drezner(2009)):

$$w_i{}^2 \geq \left(\sum_{k\neq i=1}^{n} w_k \frac{x_k - x_i}{d_{ik}}\right)^2 + \left(\sum_{k\neq i=1}^{n} w_k \frac{y_k - y_i}{d_{ik}}\right)^2 \qquad (2.27)$$

eşitsizliğinin sağlanması olacaktır. Talep noktalarının optimal olup olmadığı algoritma esnasında kontrol edilecektir. Bunun için önce n-boyutlu $U = \{u\}$ vektörü alınır. Başlangıç değeri olarak U vektörünün tüm elemanları 0 olarak ayarlanır. i . talep noktası kontrol edildiğinde $u_i = 1$ olarak güncellenir.

2.7.3. ALGORİTMA

Algoritmanın başlangıç noktası olarak X' belirlenir. Durdurma kriteri olarak $\varepsilon > 0$ seçilir. Algoritmanın adımları aşağıdaki sırayı takip eder:

1. $u_i = 0 \quad \forall\, i = 1, \ldots, n$ olarak sabitlenir.
2. Weiszfeld algoritması kullanılarak X'' noktası elde edilir.
3. X' a en yakın talep noktası bulunur: $\boldsymbol{a_k}$ olsun. X'' elde edilirken X' ile tüm talep noktaları arasındaki mesafeler zaten hesaplanmış olduğundan bu adım uzun sürmeyecektir. Eğer $u_k = 0$ ise:
    a. $\boldsymbol{a_k}$ için optimallik şartı kontrol edilir.
    b. Eğer optimal ise, algoritma durdurulur.
    c. Aksi halde $u_k = 1$ olarak güncellenir ve algoritmaya devam edilir.

4. $\Delta = \|X' - X''\|_2$ mesafesi hesaplanır. Burada $\| \quad \|_2$ öklidyen metriktir.,
5. Eğer $\Delta < \varepsilon$ ise algoritma durdurulur. X'' çözüm noktası olarak alınır.
6. $\forall\, j = 1, \ldots, 8$ için $f_j = F\left(x'' + \theta_j \Delta\, , y' + \psi_j \Delta\right) - F\,(x'', y'')$ değerleri hesaplanır.
7. $v = (A^T A)^{-1} A^T f$ değeri hesaplanır. Buradan $\bar{X} = (\bar{x}, \bar{y})$ elde edilir.
8. $X' = X'' + \Delta \bar{X}$ kabul edilir ve 2.adıma gidilir.

Drezner (1992) genellikle λ=1.8 için algoritmanın optimal sonucu verdiğini belirtmiştir.

2.7.4. GÜÇLENDİRİLMİŞ WEISZFELD ALGORİTMASI ÖRNEĞİ

Yukarıda verilen örneği Güçlendirilmiş Weiszfeld algoritması kullanarak nasıl çözüleceği aşağıda verilmiştir. Bu örnekte λ=1 ve $\varepsilon = 10^{-10}$ kullanılmıştır.

Algoritmanın başlangıç noktası olarak X' yukarıda Weiszfeld algoritmasında anlatıldığı üzere talep noktalarının ağırlık merkezi olarak hesaplanır: $X' = \left(\frac{5}{7}, \frac{17}{7}\right) = (0.7143\, , 2.4286)$

1. adım: $u_i = 0 \quad \forall\, i = 1, \ldots, n$ olarak sabitlenir.
2. Weiszfeld algoritması ile $X'' = (0.7405\, , 2.7052)$
3. $X' = \left(\frac{5}{7}, \frac{17}{7}\right)$ noktasına en yakın talep noktası: $\boldsymbol{a}_3$
    a. $u_3 = 0$ olduğundan $\boldsymbol{a}_3$ için optimallik kontrolü yapılır.

$$1 \geq \left(\left(w_1 \frac{x_1 - x_3}{d(\boldsymbol{a}_1, \boldsymbol{a}_3)}\right) + \left(w_2 \frac{x_2 - x_3}{d(\boldsymbol{a}_2, \boldsymbol{a}_3)}\right) + \left(w_4 \frac{x_4 - x_3}{d(\boldsymbol{a}_4, \boldsymbol{a}_3)}\right)\right)^2 + \left(\left(w_1 \frac{y_1 - y_3}{d(\boldsymbol{a}_1, \boldsymbol{a}_3)}\right) + \left(w_2 \frac{y_2 - y_3}{d(\boldsymbol{a}_2, \boldsymbol{a}_3)}\right) + \left(w_4 \frac{y_4 - y_3}{d(\boldsymbol{a}_4, \boldsymbol{a}_3)}\right)\right)^2$$

$$1 \geq \left(\left(w_1 \frac{0-0}{1}\right) + \left(w_2 \frac{1-0}{1.4142}\right) + \left(w_4 \frac{1-0}{3.1623}\right)\right)^2 + \left(\left(w_1 \frac{0-1}{1}\right) + \left(w_2 \frac{0-1}{1.4142}\right) + \left(w_4 \frac{4-1}{3.1623}\right)\right)^2$$

$$1 \geq \left(0 + \left(\frac{1}{1.4142}\right) + \left(4 * \frac{1}{3.1623}\right)\right)^2 + \left((-1) + \left(\frac{-1}{1.4142}\right) + \left(\frac{12}{3.1623}\right)\right)^2$$

$$1 \geq 3.8888 + 0.5752 = 4.4641$$

Eşitsizliğin sağlanmadığı açıkça görülmektedir. Dolayısıyla $\boldsymbol{a}_3$ talep noktası optimum değildir: $u_3 = 1$

Algoritmaya devam edilir.

4. $\Delta = \|X' - X''\|_2 = 0.2779$
5. Eğer $\Delta > \varepsilon$ olduğundan algoritmaya devam edilir.
6. $\forall\, j = 1, \ldots, 8$ için $f_j = F\left(x'' + \theta_j \Delta\, , y' + \psi_j \Delta\right) - F\,(x'', y'')$ değerleri hesaplanır:

f_1 : 0,2307
f_2 : 0,4406
f_3 : 0,0234
f_4 : 0,4419
f_5 : 0,2808
f_6 : 0,2991
f_7 : 0,0981
f_8 : 0,2134

7. $v = (A^T A)^{-1} A^T f$ değeri hesaplanır:

v_1 : 10,9360
v_2 : 3,1286
v_3 : 27,0391
v_4 : 0,9080
v_5 : -6,4672

Bu değerler kullanılarak:

$$\bar{X} = (\bar{x}, \bar{y}) = (-0.1566\ ,\quad 0.2573)$$

8. $X' = X'' + \Delta\bar{X} = (0.6969\, , 2.7767)$

Ardından 2.adımdan devam edilir.

Aşağıdaki tabloda Güçlendirilmiş Weiszfeld algoritmasının örneğe uygulanmış sonuçları bulunmaktadır.

Tablo 2.7: Güçlendirilmiş Weiszfeld Algoritması – örnek uygulama sonuçları (λ=1)

İterasyon sayısı	x_{k_1}	x_{k_2}	$d(\overrightarrow{x_k}, \overrightarrow{x_{k+1}})$	Maliyet: $\sum d(\overrightarrow{x_k}, \overrightarrow{a_i})$
1	0.7143	2.4286	-	12.9627
2	0.6969	2.7767	0.3486	12.6056
3	0.7770	3.0630	0.2972	12.2882
4	0.8385	3.3020	0.2467	12.0284
5	0.8860	3.4918	0.18	11.8240
6	0.9208	3.6368	0.15	11.6691
7	0.9455	3.7442	0.12	11.5550
8	0.99	3.97	0.10	11.4730
9	1.00	3.98	0.07	11.4149
10	1.00	3.99	0.06	11.3744
11	1.00	4.00	0.04	11.3463
12	1.00	4.00	0.03	11.3269

13	1.00	4.00	0.02	11.3137
14	1.00	4.00	0.02	11.3046
15	1.00	4.00	0.01	11.2985
16	1.00	4.00	0.01	11.2940
17	1.00	4.00	0.01	11.2917

Weiszfeld algoritması ile karşılaştırdığımızda, Güçlendirilmiş Weiszfeld (λ=1) algoritması kullanılarak elde edilen sonuçların daha iyi olduğu görülmektedir. Ancak Modifiye Weiszfeld (λ=1.8) ile elde edilen sonuçların ise Güçlendirilmiş Weiszfeld (λ=1) kullanıldığında elde edilenden daha iyi olduğu görülmektedir. Algoritmayı RABCO ve Güçlendirilmiş Weiszfeld (λ=1.8) birlikte olacak şekilde çalıştırdığımızda ise sonucun az bir farkla diğer tüm kombinasyonlardan daha iyi olduğu görülmüştür (Tablo 2.8).

Tablo 2.8: Güçlendirilmiş Weiszfeld Algoritması – örnek uygulama sonuçları (λ=1.8)

İterasyon sayısı	x_{k_1}	x_{k_2}	$d(\overrightarrow{\boldsymbol{x_k}}, \overrightarrow{\boldsymbol{x_{k+1}}})$	Maliyet:$\sum d(\overrightarrow{\boldsymbol{x_k}}, \overrightarrow{\boldsymbol{a_i}})$
1	0.7143	2.4286	-	12.9627
2	0.8802	3.1060	0.70	12.2272
3	0.9527	3.5618	0.46	11.7453
4	0.9665	3.7836	0.22	11.5113
5	0.9805	3.8848	0.10	11.4057
6	0.9892	3.9374	0.05	11.3508

7	0.9940	3.9657	0.03	11.3212
8	0.9967	3.9812	0.02	11.3050
9	0.9982	3.9897	0.01	11.2962
10	0.9990	3.9943	0.00	11.2913
11	0.9994	3.9969	0.00	11.2887
12	0.9997	3.9983	0.00	11.2872
13	0.9998	3.9991	0.00	11.2864
14	0.9999	3.9995	0.00	11.2859
15	0.9999	3.9997	0.00	11.2857
16	1.0000	3.9998	0.00	11.2856
17	1.0000	3.9999	0.00	11.2855

Karşılaştırmanın daha sağlıklı olabilmesi için uyguladığımız tüm kombinasyonların sonuçları Tablo 2.9'da verilmiştir.

Tablo 2.9: Örnek uygulama sonuçları

Yineleme sayısı	WEISZFELD ALGORİTMASI	MODİFİYE WEISZFELD ALGORİTMASI	GÜÇLENDİRİLMİŞ WEISZFELD ALGORİTMASI λ=1.0	GÜÇLENDİRİLMİŞ WEISZFELD ALGORİTMASI λ=1.8
1	12.9627	12.9627	12.9627	12.9627
2	12.6635	12.6635	12.6056	12.2272
3	12.3995	11.8287	12.2882	11.7453

4	12.1718	11.6023	12.0284	11.5113
5	11.9809	11.4641	11.8240	11.4057
6	11.8247	11.3841	11.6691	11.3508
7	11.6996	11.3392	11.5550	11.3212
8	11.6011	11.3146	11.4730	11.3050
9	11.5245	11.3012	11.4149	11.2962
10	11.4657	11.2939	11.3744	11.2913
11	11.4208	11.2900	11.3463	11.2887
12	11.3868	11.2879	11.3269	11.2872
13	11.3612	11.2868	11.3137	11.2864
14	11.3420	11.2862	11.3046	11.2859
15	11.3275	11.2859	11.2985	11.2857
16	11.3168	11.2858	11.2940	11.2856
17	11.3087	11.2857	11.2917	11.2855

2.8.NELDER-MEAD ALGORİTMASI

Bu tez çalışması sürecinde Nelder-Mead (NM) algoritmasının RABCO ile birlikte kullanıldığı çalışmalar yapılmıştır (Küçükdeniz ve Esnaf, 2018). Buradan elde edilen sonuçların RABCO'dan daha iyi olduğu görülmektedir. Bu nedenle Nelder-Mead

algoritması ele alınmış ve tez kapsamında önerilen hibrit yönteme NM algoritması eklenerek elde edilecek sonuçlar incelenecektir.

2.8.1. GENEL BİLGİLER:

Kısıtsız optimizasyon problemlerinde kullanmak için türevden bağımsız yöntemler 1950'lerden itibaren geliştirilmeye başlanmış ve özellikle bilgisayar teknolojisinin gelişim ile birlikte 1960'lardan itibaren hızla geliştirilmiştir. Doğrudan Arama yöntemleri ismi verilen (Hooke ve Jeeves, 1961) bu türev bağımsız yöntemler çok değişkenli fonksiyonları içeren lineer veya lineer olmayan problemlerde optimum noktalara erişebilmeyi sağlamıştır.

Doğrudan arama yöntemleri arasında sık kullanılan simpleks tabanlı doğrudan arama yöntemlerinde, seçilen bir simpleks bölge belli dönüşümlerle değiştirilerek optimum noktaya erişilmeye çalışır. Simpleks bölgeler, *n*-boyutlu bir fonksiyon için *n+1* sayıda nokta konveks bir bölge oluşturacak şekilde birleştirilerek kurulurlar.

İlk defa Spendley ve diğ. (1962) tarafından oluşturulan metotta yeni simpleks bölgeyi elde etmek için iki tip dönüşüm kullanılmaktadır:

- Yansıtma: En kötü noktanın (amaç fonksiyonunun en yüksek değeri aldığı nokta) tam karşısı
- Büzme: En iyi noktaya (amaç fonksiyonunun en düşük değeri aldığı nokta) doğru

Bu algoritmanın her adımında elde edilen simpleks bölgelerin kenarları arasındaki açılar sabit kalmaktadır. Dolayısıyla arana yapılan bölgeler büyüklük olarak değişse bile şekil olarak aynı kalmaktadır.

Nelder ve Mead (1965) bu yönteme iki yeni dönüşüm –genişleme ve kısılma – ekleyerek modifiye etmiş ve simpleks bölgenin sadece büyüklük değil şekil de değiştirerek optimum noktaya erişmesini sağlamıştır. Basitliği ve düşük bellek gereksinimi sayesinde hızla popüler olmuştur. 1970'lerden itibaren bilgisayar

laboratuvarlarındaki standart programlar arasına girmiştir. Nelder-Mead algoritması, doğrudan araştırmalar alanında yapılan tüm ilerlemelere rağmen, günümüzde hala, en popüler yöntemler arasında sayılmaktadır.

2.8.2. NELDER-MEAD ALGORİTMASININ ADIMLARI:

NM algoritması n-boyutlu reel bir fonksiyonun minimum noktalarını bulmak için, n+1 noktadan oluşan simpleks bölgelerin ardışık adımlarla optimum noktaya indirgenmesi yolunu kullanır.

Algoritmanın her adımında simpleks bir bölge oluşturan $\Delta = \{x_1, x_2, x_3 \dots, x_{n+1}\}$ noktaları amaç fonksiyonun değerlerine göre sıralanır:

$$f(x_1) \leq f(x_2) \leq \cdots \leq f(x_{n+1})$$

Sıralamada yer alan ve amaç fonksiyonunun en küçük değeri aldığı ilk nokta "*en iyi nokta*", amaç fonksiyonunun en büyük değeri aldığı son nokta "*en kötü nokta*" olarak adlandırılır.

Algoritmada her adımda yeni bir nokta belirlenir ve en kötü noktanın yerine konulur. Ardından sıralama tekrar yapılarak algoritmaya devam edilir. Yeni noktanın belirlenmesinde aşağıdaki kurallar uygulanır (Küçükdeniz ve Esnaf, 2018):

1- Sıralama: Δ simpleks bölgesinin köşelerini oluşturan n+1 nokta amaç fonksiyonundaki değerlerine göre sıralanır.

2- Yansıtma: Yansıma noktası aşağıdaki formüle göre hesaplanır:

$$x_r = (1 + \alpha)\bar{x} - \alpha x_{n+1}$$

Amaç fonksiyonunun bu noktadaki değeri hesaplanır: $f(x_r)$

Eğer $f(x_1) \leq f(x_r) < f(x_n)$ eşitliği sağlanıyorsa *en kötü nokta* x_{n+1} yerine yansıma noktası x_r konulur.

3- Genişletme: Yansıma noktası için $f(x_r) < f(x_1)$ eşitliği sağlanıyorsa genişletme noktası aşağıdaki şekilde hesaplanır:

$$x_e = \beta x_r + (1-\beta)\bar{x}$$

Amaç fonksiyonunun bu noktadaki değeri hesaplanır: $f(x_e)$

Eğer $f(x_e) < f(x_r)$ eşitliği sağlanıyorsa *en kötü nokta* x_{n+1} yerine genişletme noktası x_e konulur. Aksi halde x_{n+1} yerine yansıma noktası x_e konulur.

4- Dışa Kısılma: Eğer $f(x_n) \leq f(x_r) < f(x_{n+1})$ eşitliği sağlanıyorsa Dış Kısılma noktası aşağıdaki şekilde hesaplanır:

$$x_{oc} = \gamma x_r + (1-\gamma)\bar{x}$$

Amaç fonksiyonunun bu noktadaki değeri hesaplanır: $f(x_{oc})$

Eğer $f(x_{oc}) < f(x_r)$ eşitliği sağlanıyorsa *en kötü nokta* x_{n+1} yerine dış kısılma noktası x_{oc} konulur. Aksi halde aşağıdaki 6.adıma gidilir.

5- İçeri Kısılma: Eğer $f(x_r) \geq f(x_{n+1})$ eşitliği sağlanıyorsa İçeri Kısılma noktası aşağıdaki şekilde hesaplanır:

$$x_{ic} = (1-\gamma)\bar{x} - \gamma x_{n+1}$$

Amaç fonksiyonunun bu noktadaki değeri hesaplanır: $f(x_{ic})$

Eğer $f(x_{ic}) \leq f(x_{n+1})$ eşitliği sağlanıyorsa *en kötü nokta* x_{n+1} yerine iç kısılma noktası x_{ic} konulur. Aksi halde aşağıdaki 6.adıma gidilir.

6- Daraltma: En kötü nokta haricideki tüm noktalar, en iyi noktaya doğru belli bir oranda daraltılır. Bunun için her nokta aşağıdaki şekilde yeniden tanımlanır:

$$\forall i, 2 \leq i \leq n, \quad x_i = \delta x_i + (1-\delta)x_1$$

Algoritmadaki adımlarda kullanılan sabit sayılar, yansıtma katsayısı, α, genişletme katsayısı, β, kısılma katsayısı, γ, daraltma katsayısı, δ işlem başlamadan önce belirlenir. Nelder ve Mead (1965) bu sabitler için $\alpha > 0, \quad \beta > 1, \; 0 < \gamma < 1$ ve $0 < \delta < 1$ şartlarının sağlanması gerektiğini belirtmişlerdir. Yukarıdaki formüllerde kullanılan $\bar{x}$ noktası, tüm noktaların geometrik ortası olarak tanımlanır: $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$

Algoritmada genellikle $\alpha = 1,\ \beta = 2,\ \gamma = \frac{1}{2},\ \delta = \frac{1}{2}$ değerleri kullanılır (Lagarias ve ark, 1998).

Algoritmanın durdurulmasına, iki ardışık adım arasında simpleks bölgenin ne kadar ayrıldığı ölçülerek, karar verilir. Bunun için $\varepsilon > 0$ yeterince küçük ve önceden belirlenmiş sabit reel bir sayı olmak üzere aşağıdaki eşitlik sağlandığında algoritma durdurulur:

$$\frac{1}{2}\sum_{j=1}^{n}\left\|x_j^k - x_j^{k+1}\right\|^2 < \varepsilon$$

3. MALZEME VE YÖNTEM

Weber (1909) tarafından tanımlaması yapılan Tesis Yerleştirme probleminin çözümü için birçok farklı yöntem ileri sürülmüştür. Bu tezde problemin özel bir hali olan "Çok Kaynaklı Sürekli ve Kısıtsız TYS Problemi" için daha önce önerilmiş RABCO, Weiszfeld ve Nelder-Mead algoritmalarının değişik kombinasyonları kullanılarak yeni hibrit yöntemler önerilmiştir.

Bu yöntemlerin, paydaşı olan algoritmalara göre daha iyi sonuçlar verdiği test verileri üzerinde gösterilecektir.

3.1. TANIMLAMA

Tez kapsamında, yukarıda ayrıntıları verilen algoritmalar artarda kullanılarak "Kapasite Kısıtsız Çok Kaynaklı Weber Problemi" için optimal sonuçlar verecek yeni hibrit yöntemler önerilmektedir.

Bu problemde aşağıdaki varsayımlar yapılmaktadır:

1- Tüm talep noktaları düzlemde yer almaktadır, $a_i \in \mathbb{R}^2$, $1 \leq i \leq n$
2- Her talep noktası için sabit bir talep miktarı vardır: $w_i > 0, 1 \leq i \leq n$
3- Talep noktalarının yerleri ve talep miktarları sabit olup önceden bilinmektedir. Bu değerler algoritma çalıştırıldığında değişmeyecektir.
4- Talep noktalarındaki maliyet, bu noktanın kendisine en yakın kaynak noktasına (tesise) olan uzaklığı ve talep miktarı ile doğru orantılı olacaktır.
5- Her talep noktasının talebi sadece bir adet kaynak noktasından karşılanacaktır.
6- Kaynak noktalarında talep karşılama kısıdı olmayacaktır.

Problem talep noktalarındaki tüm talepleri karşılamak üzere belirli sayıda tesisin yerini toplam maliyeti en aza indirgeyerek belirlemektir. Problemin çözümü için iki veya üç farklı algoritmadan oluşan hibrit yöntem önerilmektedir.

3.1.1. İKİLİ ALGORİTMA:

RABCO algoritması ile birlikte Weiszfeld, Modifiye Weiszfeld, Güçlendirilmiş Weiszfeld veya Nelder-Mead algoritmalarından biri kullanılarak elde edilmiştir. Aşağıda bu hibrit algoritmanın aşamaları listelenmiştir.

Adım1: Talep noktaları *k*-adet alt öbeğe RABCO algoritması kullanılarak ayrılır. Bunun için:

Adım 1.1: RABCO ile *k*-adet öbek merkezi belirlenir.

Adım 1.2: Talep noktaları kendilerine en yakın öbek merkezi ile eşleştirilir. Öbek merkezlerine eşleştirilen talep noktaları ile alt öbekler oluşturulur.

Adım2: Her alt öbek için ayrı ayrı olmak üzere, öbek merkezleri Weiszfeld, Modifiye Weiszfeld, Güçlendirilmiş Weiszfeld veya Nelder-Mead algoritmalarından biri kullanılarak yeniden belirlenir.

Adım3: Talep noktaları yeni öbek merkezlerine göre tekrar gruplandırılır. Böylece yeni alt öbekler elde edilir.

Adım4: Amaç fonksiyonu tekrar hesaplanır.

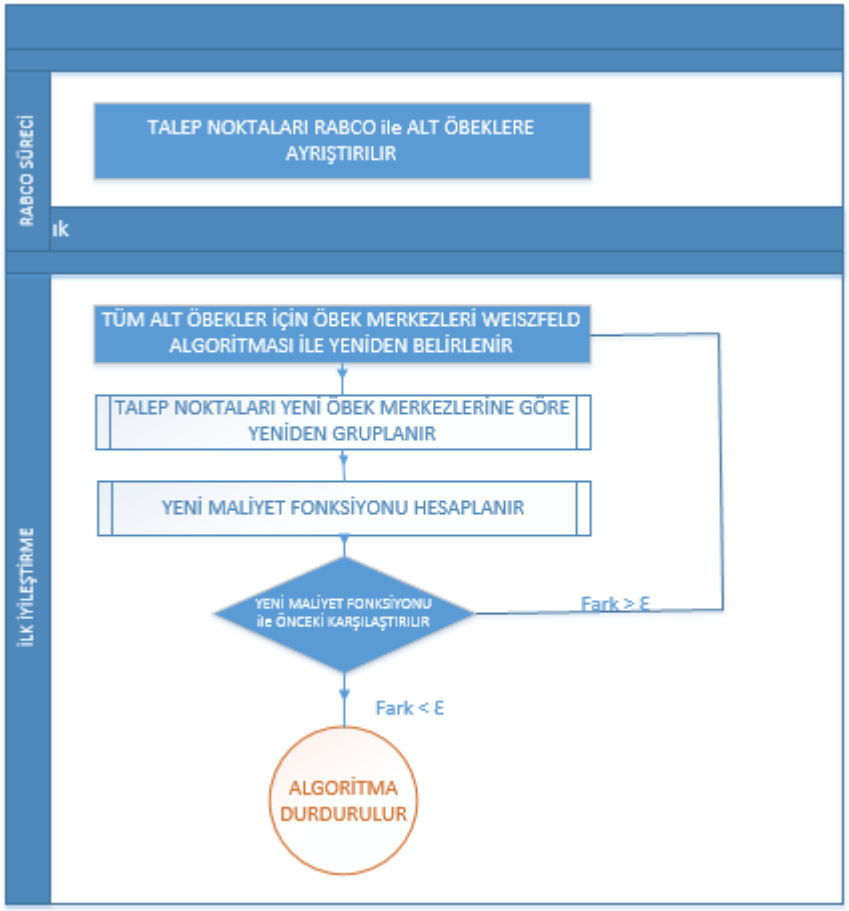
Adım5: Ardışık iki maliyet hesaplaması arasındaki fark belli bir hata değerinden az olana kadar yinelemeler devam ettirilir.

Algoritma durdurulduğunda, öbek merkezlerinin son hali sabitlenir ve talep noktaları bunlara göre alt öbeklere ayrılır. Böylece n adet nokta k adet alt kümeye toplam maliyet en az olacak şekilde ayrışmış olur.

Şekil 3.1'de RABCO ve Weiszfeld algoritmalarının birlikte kullanılması halinde takip edilecek akış verilmiştir. Burada Weiszfeld algortiması yerine Modifiye Weiszfeld, Güçlendirilmiş Weiszfeld veya Nelder-Mead algoritmalarından biri kullanılabilir.

Tüm bu kombinasyonlar tez kapsamında ayrı ayrı denenmiştir. Denemelerde elde edilen sonuçlar aşağıda 4. Uygulama başlığı altında verilmiştir.

Şekil 3-1 RABCO - WEISZFELD ALGORİTMASI AKIŞ DİAGRAMI



3.1.2. ÜÇLÜ ALGORİTMA:

Tez kapsamında öncelikle RABCO ve Weiszfeld algoritmalarının değişik kombinasyonları ile Nelder-Mead algoritması kullanılarak elde edilen hibrit yöntem üzerinde durulmuştur. Ek olarak ikili algoritma yerine üçlü algoritma kullanılması durumu da incelenmiştir. Weiszfeld algoritması gradyan temelli, Nelder-Mead algoritması ise geometrik olması nedeniyle farklı yollarla optimum noktaya erişmektedir. Bu nedenle bu iki algoritmanın birlikte kullanılmasının optimum sonuca erişmeyi kolaylaştıracağı öngörülmüştür.

Üç farklı algoritma ile oluşturulan hibrit yöntem için iki farklı kombinasyon hazırlanmıştır. İlkinde RABCO – Weiszfeld – Nelder-Mead sıralaması, ikincisinde ise RABCO – Nelder-Mead – Weiszfeld sıralaması takip edilmiştir.

RABCO – Weiszfeld – Nelder-Mead sıralaması için aşağıdaki adımlar takip edilmiştir:

Adım1: Talep noktaları *k*-adet alt öbeğe RABCO algoritması kullanılarak ayrılır.

Adım2: Her alt öbek için ayrı ayrı olmak üzere, öbek merkezleri Weiszfeld, Modifiye Weiszfeld, Güçlendirilmiş Weiszfeld algoritmalarından biri kullanılarak yeniden belirlenir.

Adım3: Amaç fonksiyonu tekrar hesaplanır.

Adım4: Bu algoritmadaki iki ardışık maliyet hesaplaması arasındaki fark belli bir hata değerinden az olana kadar yinelemeler devam ettirilir.

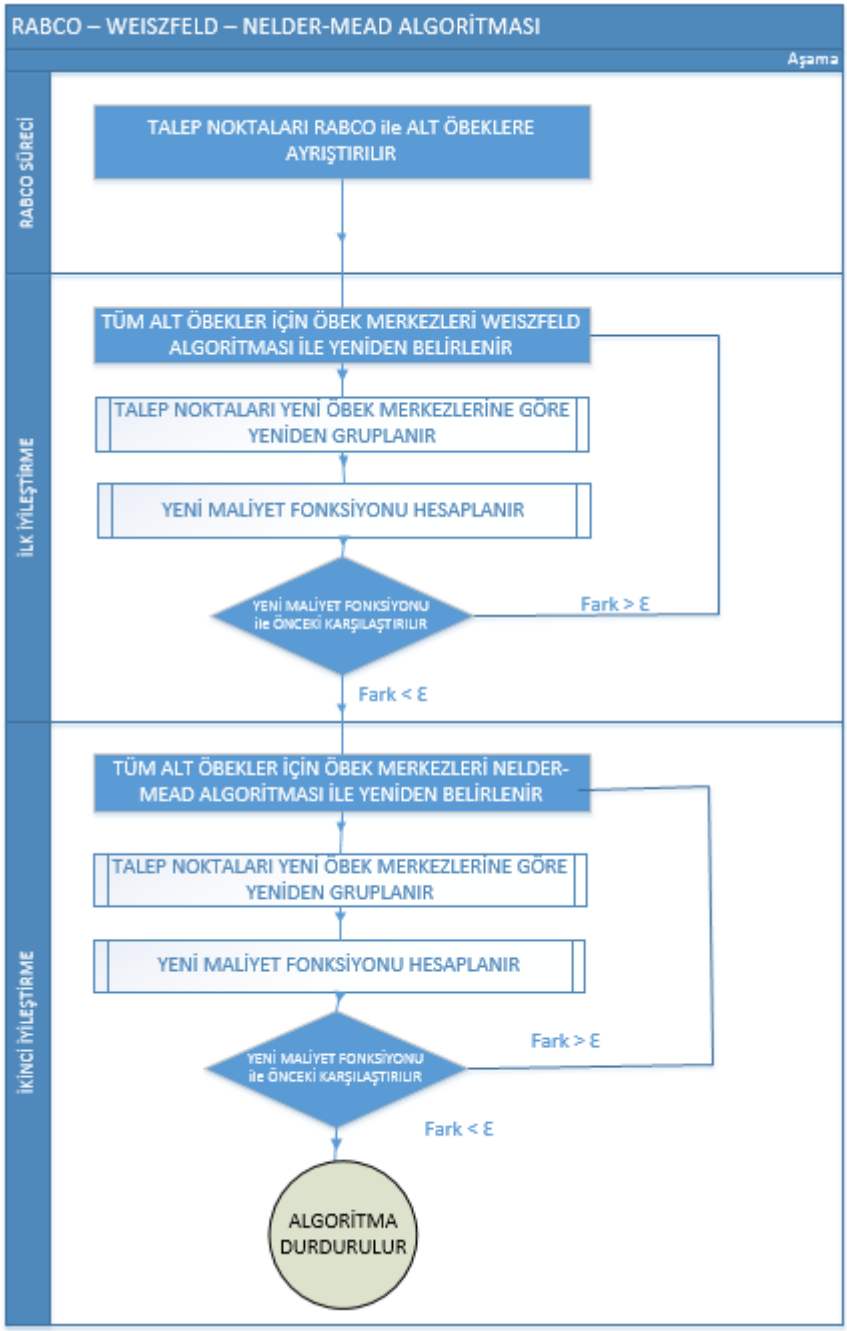
Adım5: Her alt öbek için elde edilen öbek merkezleri Nelder-Mead algoritması kullanılarak yeniden belirlenir.

Adım6: Talep noktaları yeni öbek merkezlerine göre tekrar gruplandırılır. Böylece yeni alt öbekler elde edilir.

Adım7: Amaç fonksiyonu tekrar hesaplanır.

Adım8: Bu algoritmadaki iki ardışık maliyet hesaplaması arasındaki fark belli bir hata değerinden az olana kadar yinelemeler devam ettirilir.

Şekil 3-2 RABCO - WEISZFELD - NELDER-MEAD ALGORİTMASI AKIŞ DİAGRAMI



RABCO – Nelder-Mead – Weiszfeld sıralaması için ise yukarıda verilen aşamalardan Adım5 ve Adım2 yer değiştirilmektedir. Burada RABCO ile elde edilen öbek merkezleri önce Nelder-Mead ile iyileştirilmekte ardından Weiszfeld algoritması ile tekrar iyileştirilmektedir.

Üçlü algoritmada Weiszfeld algoritması yerine Modifiye veya Güçlendirilmiş Weiszfeld algoritmalarından biri de kullanılabilir.

4. UYGULAMA

Önerilen bu yöntemin kapasite kısıtsız sürekli çok kaynaklı TYS problemlerinde daha düşük maliyetli çözümler sağladığını göstermek için, bu tür problemlerin kullanıldığı bazı veri kümeleri üzerinde uygulama yapılmıştır.

Karşılaştırmaların daha sağlıklı olması için RABCO yönteminin kullanıldığı veri kümeleri (Esnaf ve Küçükdeniz, 2012) özellikle seçilmiştir. Bunlar, OR kütüphanesinden (Operations Research Library) alınan pmedcap isimli veri kümesi ve buradan 16 farklı kombinasyon ile elde edilen kümeler (SJC324, vb.), Bongartz ve diğ. (1994) tarafından verilen 287 elemanlı küme, Taillard (2003) tarafından verilen 2863 elemanlı küme olmak üzere toplam 24 adet veri kümesidir. Veri kümeleri müşteri numarası, konum (x ve y koordinatları) ve talep miktarları içermektedir. Algoritma sürecinde talep miktarlarının sabit olduğu varsayılmıştır.

Elde edilen sonuçları karşılaştırmak için de aşağıda tanımlamaları verilen yöntemlerle elde edilen sonuçlar kullanılmıştır:

Bulanık C-Ortalamalar Algoritması (BCO): Ruspini(1969) ve Dunn(1973) tarafından öne sürülen yönteme dayanarak Bezdek(1981) tarafından geliştirilen ve bulanık mantık yöntemlerine dayanan bir algoritmadır.

BCO ve COG Algoritması: Bu hibrit yöntem Bulanık C-Ortalamalar (BCO) ve Ağırlık Merkezi (COG) düşüncesine dayanılarak üretilmiş bir algoritmadır. Burada alt öbekler önce BCO ile belirlenmekte ardından bunların öbek merkezleri COG ile iyileştirilmektedir (Esnaf ve Küçükdeniz, 2009).

Parçacık Sürü Optimizasyonu (PSO): Kuş sürülerinden esinlenilerek oluşturulmuş sezgisel bir yöntemdir (Kennedy ve Eberhart (1995)). Bu çalışmada algoritmanın ön koşullarını RABCO uygulaması (Esnaf ve Küçükdeniz, 2013) ile aynı belirlenmiştir.

4.1. RABCO VE WEISZFELD HİBRİT YÖNTEMLERİ UYGULAMASI SONUCU

RABCO – Weiszfeld hibrit algoritmasını ilgili veri kümeleri üzerindeki uygulama sonuçlarını aşağıdaki tablolarda verilmiştir. Burada ikinci algoritma olarak sırasıyla Weiszfeld ($\lambda = 1.0$), Modifiye Weiszfeld ($\lambda = 1.8$) ve Güçlendirilmiş Weiszfeld algoritmaları kullanılmıştır.

Tablo 4.1, yukarıda listelenen yöntemlerin (BCO, BCO&COG, PSO) sonuçlarını, RABCO algoritmasının tek başına uygulanmış haliyle olan sonuçları ve önerilen yeni hibrit yöntemin sonuçlarını bir arada göstermektedir.

Tablo 4.1: BCO, BCO&COG, PSO ve RABCO yöntemleri ile yeni yöntemin farklı Weiszfeld kombinasyonlarıyla yapılmış uygulama sonuçlarının karşılaştırılması.

Veri Kümesi (öbek sayısı)	BCO	BCO & COG	PSO (Mean, Minimum)	RABCO	RABCO - Weiszfeld	RABCO - Modifiye Weiszfeld	RABCO - Güçlendirilmiş Weiszfeld	RABCO - Modifiye Güçlendirilmiş Weiszfeld
Bongartz (10)	13.632	9.373	8.510 7.882	7.627,66	7.488,59	7.488,59	7.488,59	7.488,59
Bongartz (20)	11.163	6.652	6.356 5.878	4.622,84	4.424,98	4.424,93	4.424,96	4.424,92
SJC3a (25)	1.896.044	1.590.720	1.698.809 1.589.846	1.435.737,80	1.403.309,56	1.403.309,56	1.403.309,56	1.403.309,56
SJC3a (40)	1.463.476	1.175.349	1.384.343 1.344.344	1.035.707,47	1.010.510,33	1.010.508,76	1.010.509,94	1.010.508,70
SJC4a (25)	2.848.104	2.520.080	2.727.118 2.617.243	2.374.352,64	2.339.707,29	2.339.698,20	2.339.702,70	2.339.697,99
SJC4a (40)	2.387.161	1.977.533	2.337.425	1.840.615,85	1.792.436,57	1.792.436,26	1.792.436,36	1.792.436,25

			2.180.875					
SJC324 (20)	2.371.575	2.115.075	2.014.423 1.907.161	1.881.620,88	1.851.063,71	1.851.054,27	1.851.058,96	1.851.054,06
SJC324 (40)	1.582.704	1.250.084	1.447.463 1.384.221	1.163.306,30	1.137.949,00	1.137.947,34	1.137.948,23	1.137.946,92
SJC324 (60)	1.324.625	962.849	1.271.042 1.220.284	850.630,05	810.975,36	810.076,63	810.628,49	809.926,20
SJC500 (20)	5.027.370	4.827.415	4.571.051 4.422.492	4.269.917,31	4.177.700,42	4.177.698,87	4.177.699,42	4.177.698,87
SJC500 (40)	3.529.296	2.975.684	3.503.689 3.282.946	2.723.926,40	2.666.668,10	2.666.655,29	2.666.661,27	2.666.654,98
SJC500 (60)	2.918.302	2.400.974	3.118.047 2.984.051	2.091.680,61	2.043.055,51	2.043.052,13	2.043.054,30	2.043.051,31
SCJ708 (20)	6.920.864	6.599.399	6.606.900 6.338.001	6.017.948,21	5.952.604,78	5.952.604,78	5.952.604,78	5.952.604,78
SCJ708 (40)	5.042.815	4.226.576	4.993.834 4.874.041	3.998.800,63	3.935.826,51	3.935.810,24	3.935.815,31	3.935.810,23
SCJ708 (60)	3.983.870	3.331.938	4.315.360 4.078.910	2.989.036,37	2.934.485,42	2.934.478,10	2.934.481,60	2.934.477,22
SCJ818 (20)	9.867.565	8.733.462	9.103.742 8.893.566	7.624.484,87	7.533.597,20	7.533.597,20	7.533.597,20	7.533.597,20
SCJ818 (40)	6.293.308	5.543.320	6.828.894 6.523.902	5.250.380,21	5.136.108,19	5.136.094,50	5.136.101,82	5.136.092,97
SCJ818 (60)	5.358.010	4.147.632	5.877.620 5.565.993	3.873.383,84	3.797.428,98	3.797.428,83	3.797.428,92	3.797.428,82
CPmedcap2	116.416	115.286	123.632	114.575,00	113.920,97	113.920,99	113.920,98	113.920,99

(25)			121.952					
CPmedcap2 (40)	89.777	88.913	98.147 95.700	88.556,55	87.894,16	87.894,16	87.894,16	87.894,16
Thaillard (25)	968.287.764	748.719.913	852.102.000 780.562.000	656.694.127	645.895.203	645.894.892	645.895.037	645.894.874
Thaillard (50)	637.134.339	497.671.334	642.804.992 602.791.270	452.711.743	443.108.027	443.107.278	443.107.709	443.107.244
Thaillard (100)	456.456.205	332.979.312	517.619.639 494.666.904	302.172.907	295.776.171	295.775.917	295.776.079	295.775.898
Thaillard (200)	322.984.996	212.882.871	390.060.000 362.426.000	192.204.730	184.913.903	184.913.784	184.913.864	184.913.764

Veri kümeleri farklı değerlere sahip olduğu için yukarıdaki tabloda iyileşme oranı karşılaştırılamamaktadır. Bunun yerine iyileştirme yüzdeleri aşağıdaki formülle elde edilmiştir:

$$\Delta = \left(\frac{H-M}{H}\right) x\ 100 \qquad (3.2)$$

Burada H, amaç fonksiyonun karşılaştırması yapılacak yöntemle elde edilmiş sonucunu, M ise yeni yöntemle elde dilmiş sonucu göstermektedir. RABCO – Weiszfeld ($\lambda = 1$) sıralamasının kullanıldığı yeni yöntem ile diğer yöntem sonuçlarının karşılaştırması Tablo 4.2'de verilmiştir.

Tablo 4.2: RABCO – Weiszfeld (λ=1) algoritmasının kullanıldığı önerilen yöntemin sonuçlarının diğer algoritmalara kıyasla elde edilen yüzdelik iyileşmeler.

Veri Kümesi	**Talep Noktası Sayısı**	**Alt Öbek Sayısı**	**BCO**	**BCO & COG**	**PSO (Mean, Minimum)**	**RABCO**

Bongartz	287	10	% 45,066	% 20,105	% 12,002 % 4,991	% 1,823
Bongartz	287	20	% 60,36	% 33,479	% 30,381 % 24,72	% 4,28
SJC3a	300	25	% 25,988	% 11,781	% 17,395 % 11,733	% 2,259
SJC3a	300	40	% 30,951	% 14,025	% 27,004 % 24,832	% 2,433
SJC4a	402	25	% 17,85	% 7,157	% 14,206 % 10,604	% 1,459
SJC4a	402	40	% 24,913	% 9,36	% 23,316 % 17,811	% 2,618
SJC324	324	20	% 21,948	% 12,482	% 8,109 % 2,941	% 1,624
SJC324	324	40	% 28,101	% 8,97	% 21,383 % 17,791	% 2,18
SJC324	324	60	% 38,777	% 15,773	% 36,196 % 33,542	% 4,662
SJC500	500	20	% 16,901	% 13,459	% 8,605 % 5,535	% 2,16
SJC500	500	40	% 24,442	% 10,385	% 23,89 % 18,772	% 2,102
SJC500	500	60	% 29,992	% 14,907	% 34,476	% 2,325

					% 31,534	
SCJ708	708	20	% 13,99	% 9,801	% 9,903 % 6,081	% 1,086
SCJ708	708	40	% 21,952	% 6,879	% 21,186 % 19,249	% 1,575
SCJ708	708	60	% 26,341	% 11,929	% 31,999 % 28,057	% 1,825
SCJ818	818	20	% 23,653	% 13,739	% 17,247 % 15,292	% 1,192
SCJ818	818	40	% 18,388	% 7,346	% 24,789 % 21,272	% 2,176
SCJ818	818	60	% 29,126	% 8,443	% 35,392 % 31,774	% 1,961
CPmedcap	1481	25	% 2,143	% 1,184	% 7,855 % 6,585	% 0,571
CPmedcap	1481	40	% 2,097	% 1,146	% 10,446 % 8,157	% 0,748
Thaillard	2863	25	% 33,295	% 13,733	% 24,2 % 17,253	% 1,644
Thaillard	2863	50	% 30,453	% 10,964	% 31,066 % 26,491	% 2,121
Thaillard	2863	100	% 35,202	% 11,173	% 42,858 % 40,207	% 2,117

Thaillard	2863	200	% 42,748	% 13,138	% 52,593 % 48,979	% 3,793
Ortalama			**% 26,862**	**% 11,722**	**% 23,604** **% 19,759**	**% 2,114**

RABCO yönteminin diğer kıyaslama yöntemlerinden sırasıyla ortalama %23,57, %9,12, %17,13, %21,00 daha iyi sonuç verdiği görülmektedir (Esnaf ve Küçükdeniz, 2013). RABCO ve Weiszfeld algoritmasının (λ=1) kullanıldığı önerilen bu yöntemde ise RABCO yöntemine göre %2,114, BCO&COG yöntemine göre %11,722, PSO (Mean) yöntemine göre %23,604, PSO (Minimum) yöntemine göre %19,759 ve BCO yöntemine göre ise %26,862 oranında daha iyi sonuçlar elde edilmiştir.

Aşağıdaki Tablo 4.3'te yeni hibrit yöntemde Weiszfeld algoritması yerine Modifiye Weiszfeld ve Güçlendirilmiş Weiszfeld algoritmaları kullanıldığında elde edilen sonuçların karşılaştırması görünmektedir. Tabloda açıkça görüleceği üzere Modifiye ve Güçlendirilmiş Weiszfeld algoritmaları arasındaki fark çok azdır.

Tablo 4.3: RABCO ve yeni yöntemin farklı Weiszfeld kombinasyonlarıyla yapılmış uygulama sonuçları.

Veri Kümesi	**RABCO**	**RABCO – Weiszfeld ($\lambda = 1$)**	**RABCO – Modifiye Weiszfeld ($\lambda = 1,8$)**	**RABCO – Güçlendirilmiş Weiszfeld ($\lambda = 1$)**	**RABCO – Güçlendirilmiş Weiszfeld ($\lambda = 1,8$)**
Bongartz (10)	7.627,6645	7.488,5927	7.488,5913	7.488,5913	7.488,5913
Bongartz (20)	4.622,8436	4.424,9768	4.424,9298	4.424,9628	4.424,9201
SJC3a (25)	1.435.737,8027	1.403.309,5617	1.403.309,5616	1.403.309,5616	1.403.309,5616
SJC3a (40)	1.035.707,4675	1.010.510,3297	1.010.508,7580	1.010.509,9396	1.010.508,6961

SJC4a (25)	2.374.352,6372	2.339.707,2866	2.339.698,2010	2.339.702,6961	2.339.697,9922
SJC4a (40)	1.840.615,8539	1.792.436,5705	1.792.436,2634	1.792.436,3598	1.792.436,2473
SJC324 (20)	1.881.620,8752	1.851.063,7147	1.851.054,2696	1.851.058,9579	1.851.054,0557
SJC324 (40)	1.163.306,3040	1.137.948,9967	1.137.947,3424	1.137.948,2286	1.137.946,9248
SJC324 (60)	850.630,0549	810.975,3575	810.076,6283	810.628,4916	809.926,1993
SJC500 (20)	4.269.917,3083	4.177.700,4154	4.177.698,8707	4.177.699,4169	4.177.698,8651
SJC500 (40)	2.723.926,3967	2.666.668,0970	2.666.655,2907	2.666.661,2676	2.666.654,9783
SJC500 (60)	2.091.680,6112	2.043.055,5097	2.043.052,1323	2.043.054,3034	2.043.051,3112
SCJ708 (20)	6.017.948,2138	5.952.604,7847	5.952.604,7847	5.952.604,7847	5.952.604,7847
SCJ708 (40)	3.998.800,6290	3.935.826,5111	3.935.810,2438	3.935.815,3140	3.935.810,2311
SCJ708 (60)	2.989.036,3674	2.934.485,4181	2.934.478,1044	2.934.481,6045	2.934.477,2211
SCJ818 (20)	7.624.484,8711	7.533.597,2014	7.533.597,2014	7.533.597,2014	7.533.597,2014
SCJ818 (40)	5.250.380,2058	5.136.108,1922	5.136.094,5017	5.136.101,8226	5.136.092,9733
SCJ818 (60)	3.873.383,8414	3.797.428,9834	3.797.428,8304	3.797.428,9169	3.797.428,8193
CPmedcap2 (25)	114.574,9996	113.920,9715	113.920,9899	113.920,9774	113.920,9924
CPmedcap2 (50)	88.556,5536	87.894,1588	87.894,1588	87.894,1588	87.894,1588
Taillard (25)	656.694.127	645.895.202	645.894.892	645.895.037	645.894.873
Taillard (50)	452.711.742	443.108.027	443.107.277	443.107.709	443.107.243
Taillard (100)	302.172.906	295.776.171	295.775.916	295.776.078	295.775.897
Taillard (200)	192.204.730	184.913.902	184.913.784	184.913.864	184.913.763

Hibrit yöntemde Modifiye ve Güçlendirilmiş Weiszfeld algoritmalar kullanıldığında elde edilen sonuçlar arasındaki farklar çok azdır. Aşağıdaki tablolar bu farkları göstermek üzere hazırlanmıştır. Tablo 4.4'te, önerilen bu yeni yöntem ve RABCO ile

elde edilen sonuçlar bulunmaktadır. Tabloda RABCO yöntemi ile elde edilen sonuçlar ile yeni metodun farklı kombinasyonları ile elde edilen sonuçlar arasındaki sayısal farklar gösterilmiştir.

Tablo 4.4: Önerilen yöntemin farklı kombinasyonları ile elde edilen sonuçların RABCO yönteminin sonuçları arasındaki farklar

Veri Kümesi	Talep Noktası sayısı	Alt öbek sayısı	RABCO – Weiszfeld ($\lambda = 1$)	RABCO – Güçlendirilmiş Weiszfeld ($\lambda = 1$)	RABCO – Modifiye Weiszfeld ($\lambda = 1.8$)	RABCO – Güçlendirilmiş Weiszfeld ($\lambda = 1.8$)
Bongartz	287	10	139,0718	139,0732	139,0732	139,0732
Bongartz	287	20	197,8668	197,9138	197,8808	197,9235
SJC3a	300	25	32.428,2410	32.428,2411	32.428,2411	32.428,2411
SJC3a	300	40	25.197,1378	25.198,7095	25.197,5279	25.198,7714
SJC4a	402	25	34.645,3506	34.654,4362	34.649,9411	34.654,6450
SJC4a	402	40	48.179,2834	48.179,5905	48.179,4941	48.179,6066
SJC324	324	20	30.557,1605	30.566,6056	30.561,9173	30.566,8195
SJC324	324	40	25.357,3073	25.358,9616	25.358,0754	25.359,3792
SJC324	324	60	39.654,6974	40.553,4266	40.001,5633	40.703,8556
SJC500	500	20	92.216,8929	92.218,4376	92.217,8914	92.218,4432
SJC500	500	40	57.258,2997	57.271,1060	57.265,1291	57.271,4184
SJC500	500	60	48.625,1015	48.628,4789	48.626,3078	48.629,3000
SCJ708	708	20	65.343,4291	65.343,4291	65.343,4291	65.343,4291
SCJ708	708	40	62.974,1179	62.990,3852	62.985,3150	62.990,3979
SCJ708	708	60	54.550,9493	54.558,2630	54.554,7629	54.559,1463
SCJ818	818	20	90.887,6697	90.887,6697	90.887,6697	90.887,6697
SCJ818	818	40	114.272,0136	114.285,7041	114.278,3832	114.287,2325

SCJ818	818	60	75.954,8580	75.955,0110	75.954,9245	75.955,0221
CPmedcap2	1481	25	654,0281	654,0097	654,0222	654,0072
CPmedcap2	1481	40	662,3948	662,3948	662,3948	662,3948
Taillard	2863	25	10.798.925	10.799.235	10.799.090	10.799.254
Taillard	2863	50	9.603.715	9.604.465	9.604.033	9.604.499
Taillard (100)	2863	100	6.396.735	6.396.990	6.396.828	6.397.009
Taillard (200)	2863	200	7.290.828	7.290.946	7.290.866	7.290.967

Yukarıdaki tabloda gösterilen değerler ilgili kümenin özelliklerine bağlı olduğu için uygulanan yöntemlerin ne kadar etkili olduğu açıkça görünmemektedir. Bu nedenle Tablo 4.5'te iyileşmelerin yüzde değerleri gösterilmiştir. Bu tabloda ilk sütunda RABCO sonuçları, diğer sütunlarda ise yüzde farklar verilmiştir.

Tablo 4.5: Yeni yöntemin farklı Weiszfeld kombinasyonlarıyla yapılmış uygulama sonuçlarında, RABCO yöntemi sonuçlarına göre olan iyileşmelerin yüzdelik karşılaştırılması.

Veri Kümesi	**RABCO**	**RABCO – Weiszfeld** $(\boldsymbol{\lambda = 1})$	**RABCO – Güçlendirilmiş Weiszfeld** $(\boldsymbol{\lambda = 1})$	**RABCO – Modifiye Weiszfeld** $(\boldsymbol{\lambda = 1.8})$	**RABCO – Güçlendirilmiş Weiszfeld** $(\boldsymbol{\lambda = 1.8})$
Bongartz287 (10)	7,627.66	% 1,8233	% 1,8233	% 1,8233	% 1,8233
Bongartz287 (20)	4,622.84	% 4,2812	% 4,2805	% 4,2814	% 4,2812
SJC3a (25)	1,435,737.81	% 2,2586	% 2,2586	% 2,2586	% 2,2586
SJC3a (40)	1,035,707.47	% 2,4330	% 2,4329	% 2,433	% 2,433
SJC4a (25)	2,374,352.00	% 1,4595	% 1,4593	% 1,4595	% 1,4595
SJC4a (40)	1,840,615.86	% 2,6176	% 2,6176	% 2,6176	% 2,6176
SJC324 (20)	1,881,620.00	% 1,6245	% 1,6242	% 1,6245	% 1,6245
SJC324 (40)	1,163,306.29	% 2,1799	% 2,1798	% 2,1799	% 2,1799
SJC324 (60)	850,630.00	% 4,7675	% 4,7026	% 4,7851	% 4,7675
SJC500 (20)	4,269,917.30	% 2,1597	% 2,1597	% 2,1597	% 2,1597

SJC500 (40)	2,723,926.34	% 2,1025	% 2,1023	% 2,1025	% 2,1025
SJC500 (60)	2,091,680.61	% 2,3249	% 2,3247	% 2,3249	% 2,3249
SCJ708 (20)	6,017,948.00	% 1,0858	% 1,0858	% 1,0858	% 1,0858
SCJ708 (40)	3,998,800.65	% 1,5752	% 1,5751	% 1,5752	% 1,5752
SCJ708 (60)	2,989,036.00	% 1,8253	% 1,8252	% 1,8253	% 1,8253
SCJ818 (20)	7,624,484.86	% 1,1920	% 1,1920	% 1,1920	% 1,1920
SCJ818 (40)	5,250,380.22	% 2,1767	% 2,1766	% 2,1767	% 2,1767
SCJ818 (60)	3,873,383.00	% 1,9609	% 1,9609	% 1,9609	% 1,9609
CPmedcap2 (25)	114,575.00	% 0,5708	% 0,5708	% 0,5708	% 0,5708
CPmedcap2 (40)	88,556.55	% 0,7480	% 0,7480	% 0,7480	% 0,7480
Taillard (25)	656,694,127	% 1,6445	% 1,6445	% 1,6445	% 1,6445
Taillard (50)	452,711,742	% 2,1215	% 2,1214	% 2,1215	% 2,1215
Taillard (100)	302,172,906	% 2,1170	% 2,1169	% 2,1170	% 2,1170
Taillard (200)	192,204,730	% 3,7933	% 3,7933	% 3,7933	% 3,7933
Ortalama		**% 2,1139**	**% 2,1185**	**% 2,1157**	**% 2,1192**

4.2. RABCO VE NELDER-MEAD HİBRİT ALGORİTMASI:

RABCO ve NM algoritmalarının birlikte kullanılması ÇKTYS problemlerinde daha iyi sonuç verebilmektedir (Küçükdeniz ve Esnaf, 2018). Bunun için aşağıdaki adımlar takip edilir:

1- Düzlemde yerleri belli olacak şekilde n.adet talep noktası belirlenir: $a^k = (x_k, y_k),\ k = 1,2 \dots, n$

2- c adet kaynak noktası için öbek merkezleri RABCO algoritması ile belirlenir.

3- Talep noktaları üyelik dereceleri u_{ik}, kullanılarak birer adet öbek merkezi ile eşleştirilir. Böylelikle talep noktaları öbeklere ayrışmış olur.

4- Her öbek için, öbek merkezinin yeri NM algoritması ile optimize edilir.

5- Talep noktaları NM algoritması ile elde edilen yeni öbek merkezlerine eşleştirilir. Ardından amaç fonksiyonu tekrar hesaplanır.

Algoritmanın bilinen veri kümelerine uygulaması ile elde edilen sonuçlar Tablo 4.6'da listelenmiştir (Küçükdeniz ve Esnaf, 2018).

Tablo 4.6: RABCO ve Nelder-Mead Hibrit Algoritması sonuçları

Veri Kümesi	FCM	FCM – COG	FCM – NM	RABCO	RABCO – NM
Bongartz et al. (1994) (10)	13.633,00	9.373,00	8.202,00	7.627,00	7.257,00
Bongartz et al. (1994) (20)	11.163,00	6.652,00	6.058,00	4.622,00	4.463,00
SJC3a (25)	1.896.044,00	1.590.720,00	1.455.470,00	1.435.737,00	1.397.181,00
SJC3a (40)	1.463.476,00	1.175.349,00	1.068.120,00	1.035.707,00	996.274,00
SJC4a (25)	2.848.104,00	2.520.080,00	2.415.106,00	2.374.352,00	2.328.186,00
SJC4a (40)	2.387.161,00	1.977.533,00	1.853.488,00	1.840.615,00	1.755.552,00
SJC324 (20)	2.371.575,00	2.115.075,00	1.939.756,00	1.881.620,00	1.849.145,00
SJC324 (40)	1.582.704,00	1.250.084,00	1.195.256,00	1.163.306,00	1.122.982,00
SJC324 (60)	1.324.625,00	962.849,00	872.151,00	850.630,00	800.439,00
SJC500 (20)	5.027.370,00	4.827.415,00	4.371.431,00	4.269.917,00	4.126.927,00
SJC500 (40)	3.529.296,00	2.975.684,00	2.790.630,00	2.723.926,00	2.662.007,00
SJC500 (60)	2.918.302,00	2.400.974,00	2.110.570,00	2.091.680,00	2.014.274,00
SJC708 (20)	6.920.864,00	6.599.399,00	5.887.346,00	6.017.948,00	5.924.470,00
SJC708 (40)	5.042.815,00	4.226.576,00	3.951.696,00	3.998.800,00	3.734.418,00
SJC708 (60)	3.983.870,00	3.331.938,00	3.120.330,00	2.989.036,00	2.869.925,00
SJC818 (20)	9.867.565,00	8.733.462,00	7.836.862,00	7.624.484,00	7.507.594,00
SJC818 (40)	6.293.308,00	5.543.320,00	5.195.670,00	5.250.380,00	5.038.369,00
SJC818 (60)	5.358.010,00	4.147.632,00	3.974.910,00	3.873.383,00	3.719.948,00
CPmedcap2 (25)	116.416,00	115.286,00	113.801,00	114.575,00	113.657,00
CPmedcap2 (40)	89.777,00	88.913,00	87.837,00	88.556,00	87.383,00
Taillard (2003) (25)	968.287.764	748.719.913	677.108.079	656.694.127	645.719.011
Taillard (2003) (50)	637.134.339	497.671.334	466.603.329	452.711.742	439.571.070
Taillard (2003) (25)	456.456.205	332.979.312	321.799.135	302.172.906	290.541.198
Taillard (2003) (50)	322.984.996	212.882.871	214.434.343	192.204.730	185.953.715

Bu tabloda görüldüğü üzere, yeni hibrit yöntemde Weiszfeld yerine Nelder-Mead kullanıldığında daha iyi sonuçlar elde edilmektedir.

4.3.RABCO–WEISZFELD VE RABCO–NM HİBRİT ALGORİTMALARI:

Bu tezde ileri sürülen RABCO ve Weiszfeld algoritması ile kıyaslandığında RABCO ve NM hibrit algoritmasının daha iyi sonuçlar verdiği görülmektedir. Aşağıdaki tabloda bu algoritmalarla elde edilen sonuçlar verilmiştir.

Tablo 4.7: RABCO–NM ve RABCO–WEISZFELD algoritma sonuçları

Veri Kümesi	RABCO	RABCO – NM	RABCO – Weiszfeld λ=1	RABCO – Modifiye Weiszfeld λ=1.8	RABCO – Güçlendirilmiş Weiszfeld λ=1.0	RABCO – Güçlendirilmiş Weiszfeld λ=1.8
Bongartz et al. (1994) (10)	7.627,00	**7.257,00**	7.488,59	7.488,59	7.488,59	7.488,59
Bongartz et al. (1994) (20)	4.622,00	4.463,00	4.424,98	4.424,93	4.424,96	**4.424,92**
SJC3a (25)	1.435.737,00	**1.397.181,00**	1.403.309,56	1.403.309,56	1.403.309,56	1.403.309,56
SJC3a (40)	1.035.707,00	**996.274,00**	1.010.510,33	1.010.508,76	1.010.509,94	1.010.508,70
SJC4a (25)	2.374.352,00	**2.328.186,00**	2.339.707,29	2.339.698,20	2.339.702,70	2.339.697,99
SJC4a (40)	1.840.615,00	**1.755.552,00**	1.792.436,57	1.792.436,26	1.792.436,36	1.792.436,25
SJC324 (20)	1.881.620,00	**1.849.145,00**	1.851.063,71	1.851.054,27	1.851.058,96	1.851.054,06
SJC324 (40)	1.163.306,00	**1.122.982,00**	1.137.949,00	1.137.947,34	1.137.948,23	1.137.946,92
SJC324 (60)	850.630,00	**800.439,00**	810.975,36	810.076,63	810.628,49	809.926,20
SJC500 (20)	4.269.917,00	**4.126.927,00**	4.177.700,42	4.177.698,87	4.177.699,42	4.177.698,87
SJC500 (40)	2.723.926,00	**2.662.007,00**	2.666.668,10	2.666.655,29	2.666.661,27	2.666.654,98
SJC500 (60)	2.091.680,00	**2.014.274,00**	2.043.055,51	2.043.052,13	2.043.054,30	2.043.051,31
SJC708 (20)	6.017.948,00	**5.924.470,00**	5.952.604,78	5.952.604,78	5.952.604,78	5.952.604,78
SJC708 (40)	3.998.800,00	**3.734.418,00**	3.935.826,51	3.935.810,24	3.935.815,31	3.935.810,23
SJC708 (60)	2.989.036,00	**2.869.925,00**	2.934.485,42	2.934.478,10	2.934.481,60	2.934.477,22
SJC818 (20)	7.624.484,00	**7.507.594,00**	7.533.597,20	7.533.597,20	7.533.597,20	7.533.597,20

SJC818 (40)	5.250.380,00	**5.038.369,00**	5.136.108,19	5.136.094,50	5.136.101,82	5.136.092,97
SJC818 (60)	3.873.383,00	**3.719.948,00**	3.797.428,98	3.797.428,83	3.797.428,92	3.797.428,82
CPmedcap2 (25)	114.575,00	**113.657,00**	113.920,97	113.920,99	113.920,98	113.920,99
CPmedcap2 (40)	88.556,00	**87.383,00**	87.894,16	87.894,16	87.894,16	87.894,16
Taillard (2003) (25)	656.694.127	**645.719.011**	645.895.202	645.894.892	645.895.037	645.894.873
Taillard (2003) (50)	452.711.742	**439.571.070**	443.108.027	443.107.277	443.107.709	443.107.243
Taillard (2003) (100)	302.172.906	**290.541.198**	295.776.171	295.775.916	295.776.078	295.775.897
Taillard (2003) (200)	192.204.730	185.953.715	184.913.902	184.913.784	184.913.864	**184.913.763**

Tablo 4.7' ye göre sadece Bongartz et al (20 öbek) ve Taillard (2003) (200 öbek) veri kümeleri haricinde tüm örnek veri kümelerinde RABCO–NM uygulamasının daha iyi sonuç verdiği görülmüştür.

Tablo 4.5'te görüldüğü üzere hibrit yöntemde ikinci algoritma olarak Weiszfeld, Modifiye Wesizfeld veya Güçlendirilmiş Weiszfeld yöntemlerinden herhangi birinin kullanılması sonuçlarda çok büyük bir değişiklik yaratmamakla birlikte Güçlendirilmiş Weiszfeld (λ=1,8) yönteminin diğerlerine göre çok az da olsa daha iyi sonuçlar verdiği görülmüştür. Bu nedenle Tablo 4.8'de sadece Güçlendirilmiş Weiszfeld (λ=1,8) algoritmasının sonuçları dikkate alınmıştır.

Tablo 4.8: RABCO-Güçlendirilmiş Weiszfeld (λ=1,8) ve RABCO-Nelder-Mead algoritmalarının sonuçlarının karşılaştırılması

Veri Kümesi	RABCO	RABCO – Güçlendirilmiş Weiszfeld λ=1,8	RABCO – NM	İyileştirme 1 (%)	İyileştirme 2 (%)
Bongartz (10)	7.627,66	7.488,59	7.257,00	% 1,8571	% 5,1076
Bongartz (20)	4.622,84	4.424,92	4.463,00	% 4,4728	% 3,5814
SJC3a (25)	1.435.737,80	1.403.309,56	1.397.181,00	% 2,3108	% 2,7596
SJC3a (40)	1.035.707,47	1.010.508,70	996.274,00	% 2,4937	% 3,9581
SJC4a (25)	2.374.352,64	2.339.697,99	2.328.186,00	% 1,4812	% 1,9829
SJC4a (40)	1.840.615,85	1.792.436,25	1.755.552,00	% 2,6879	% 4,8454
SJC324 (20)	1.881.620,88	1.851.054,06	1.849.145,00	% 1,6513	% 1,7563
SJC324 (40)	1.163.306,30	1.137.946,92	1.122.982,00	% 2,2285	% 3,5908
SJC324 (60)	850.630,05	809.926,20	800.439,00	% 5,0256	% 6,2704
SJC500 (20)	4.269.917,31	4.177.698,87	4.126.927,00	% 2,2074	% 3,4648

SJC500 (40)	2.723.926,40	2.666.654,98	2.662.007,00	% 2,1477	% 2,3260
SJC500 (60)	2.091.680,61	2.043.051,31	2.014.274,00	% 2,3802	% 3,8429
SCJ708 (20)	6.017.948,21	5.952.604,78	5.924.470,00	% 1,0977	% 1,5778
SCJ708 (40)	3.998.800,63	3.935.810,23	3.734.418,00	% 1,6004	% 7,0796
SCJ708 (60)	2.989.036,37	2.934.477,22	2.869.925,00	% 1,8592	% 4,1503
SCJ818 (20)	7.624.484,87	7.533.597,20	7.507.594,00	% 1,2064	% 1,5570
SCJ818 (40)	5.250.380,21	5.136.092,97	5.038.369,00	% 2,2252	% 4,2079
SCJ818 (60)	3.873.383,84	3.797.428,82	3.719.948,00	% 2,0002	% 4,1247
CPmedcap2 (25)	114.575,00	113.920,99	113.657,00	% 0,5741	% 0,8077
CPmedcap2 (50)	88.556,55	87.894,16	87.383,00	% 0,7536	% 1,3430
Taillard (25)	656.694.127	645.894.873	645.719.011	% 1,6720	% 1,6997
Taillard (50)	452.711.742	443.107.243	439.571.070	% 2,1675	% 2,9894
Taillard (100)	302.172.906	295.775.897	290.541.198	% 2,1628	% 4,0035
Taillard (200)	192.204.730	**184.913.763**	185.953.715	% 3,9429	% 3,3616
ORTALAMA				**% 2,1753**	**% 3,3495**

Tablo 4.8'de RABCO-NM ve RABCO-Güçlendirilmiş Weiszfeld (λ=1,8) algoritmalarından elde edilen sonuçlara ait iyileştirme oranları verilmiştir. İyileştirme 1 başlığında RABCO-Güçlendirilmiş Weiszfeld (λ=1,8) algoritması ile elde edilen sonuçların sadece RABCO algoritması kullanılmasına göre elde edilen iyileştirmesi listelenmiştir. İyileştirme 2 başlıklı sütunda ise RABCO-NM algoritması ile elde edilen sonuçların sadece RABCO algoritması kullanılmasına göre elde edilen iyileştirmesi verilmiştir. Buna göre RABCO-Güçlendirilmiş Weiszfeld (λ=1,8) hibrit algoritması ile ortalama %2,1753 oranında bir iyileştirme elde edilirken RABCO-NM yöntemi ile ortalama %3,3495 oranında iyileştirme sağlanmıştır.

4.4. RABCO, WEISZFELD VE KOMBİNASYONLARI İLE NELDER-MEAD ALGORİTMALARININ BİRLİKTE KULLANIMI

Tez kapsamında iki algoritmadan oluşan hibrit algoritmaların yanı sıra yukarıda verilen algoritmaların hepsi birden kullanılarak elde edilen üçlü algoritmalar da incelenmiştir, Bu kapsamda RABCO ile elde edilen öbek merkezleri sırasıyla,

a- NM ve ardından Weiszfeld algoritması (veya Modifiye Weiszfeld, Güçlendirilmiş Weiszfeld) kullanılarak
b- Weiszfeld (veya Modifiye Weiszfeld, Güçlendirilmiş Weiszfeld) ve ardından NM algoritmasının kullanılarak

iyileştirme yapılmıştır. Karşılaştırmanın sağlıklı olması amacıyla yine aynı veri kümeleri kullanılmıştır.

Tablo 4.9'da, RABCO algoritması, RABCO – NM algoritması ve RABCO – NM – Güçlendirilmiş Weiszfeld **(λ=1,8)** hibrit algoritması ile elde edilen sonuçlar bulunmaktadır. Bu uygulamada çok az da olsa diğerlerine göre daha iyi sonuç verdiği için sadece Güçlendirilmiş Weiszfeld **(λ=1.8)** algoritması kullanılmıştır.

Tablo 4.9: RABCO, Nelder-Mead ve **Güçlendirilmiş Weiszfeld (λ=1.8)** algoritmalarının birlikte kullanımı sonuçları

Veri Kümesi	RABCO	RABCO-NM	İyileştirme 1 (%)	RABCO-NM-Güçlendirilmiş Weiszfeld (λ=1.8)	İyileştirme 2 (%)
Bongartz et al. (1994) (10)	7.627,00	7.257,00	% 4,85	7.235,00	% 0,30
Bongartz et al. (1994) (20)	4.622,00	4.463,00	% 3,44	4.456,50	% 0,15
SJC3a (25)	1.435.737,00	1.397.181,00	% 2,69	1.396.349,69	% 0,06

SJC3a (40)	1.035.707,00	996.274,00	% 3,81	992.014,52	% 0,43
SJC4a (25)	2.374.352,00	2.328.186,00	% 1,94	2.327.051,32	% 0,05
SJC4a (40)	1.840.615,00	1.755.552,00	% 4,62	1.754.412,66	% 0,06
SJC324 (20)	1.881.620,00	1.849.145,00	% 1,73	1.848.516,20	% 0,03
SJC324 (40)	1.163.306,00	1.122.982,00	% 3,47	1.121.603,54	% 0,12
SJC324 (60)	850.630,00	800.439,00	% 5,9	786.695,27	% 1,75
SJC500 (20)	4.269.917,00	4.126.927,00	% 3,35	4.123.970,63	% 0,07
SJC500 (40)	2.723.926,00	2.662.007,00	% 2,27	2.658.266,07	% 0,14
SJC500 (60)	2.091.680,00	2.014.274,00	% 3,7	2.006.888,02	% 0,37
SJC708 (20)	6.017.948,00	5.924.470,00	% 1,55	5.922.807,63	% 0,03
SJC708 (40)	3.998.800,00	3.734.418,00	% 6,61	3.724.602,82	% 0,26
SJC708 (60)	2.989.036,00	2.869.925,00	% 3,98	2.865.043,62	% 0,17
SJC818 (20)	7.624.484,00	7.507.594,00	% 1,53	7.506.935,38	% 0,01
SJC818 (40)	5.250.380,00	5.038.369,00	% 4,04	5.013.005,39	% 0,51
SJC818 (60)	3.873.383,00	3.719.948,00	% 3,96	3.712.229,77	% 0,21
CPmedcap2 (25)	114.575,00	113.657,00	% 0,8	113.652,61	% 0
CPmedcap2 (40)	88.556,00	87.383,00	% 1,32	87.373,82	% 0,01
Taillard (2003) (25)	656.694.127	645.719.011	% 1,67	645.441.556,42	% 0,04
Taillard (2003) (50)	452.711.742	439.571.070	% 2,9	438.774.767,71	% 0,18
Taillard (2003) (100)	302.172.906	290.541.198	% 3,85	287.108.176,94	% 1,2
Taillard (2003) (200)	192.204.730	185.953.715	% 3,25	182.459.835,46	% 1,91
ORTALAMALAR			**% 3,22**		**% 0,34**

Tablo 4.9'daki sonuçlardan da görüleceği üzere RABCO ile belirlenen alt kümeler önce Nelder-Mead ardından Güçlendirilmiş Weiszfeld **(λ=1.8)** algoritması ile iyileştirildiğinde RABCO–Nelder-Mead algoritmasına göre ortalama %0.34 oranında (iyileştirme 2) daha iyi sonuç elde edilmektedir. Veri kümelerine bakıldığında ise 25 adet alt kümesi olan CPmedcap verisi için çok az bir iyileştirme sağlanırken (% 0,0039), en iyi sonucun % 1,91 ile 200 adet alt kümesi olan Taillard(2003) veri kümesinde olduğu görülmektedir.

Üçlü algoritmadaki sıralama değiştirilip RABCO–Güçlendirilmiş Weiszfeld **(λ=1.8)**–Nelder-Mead algoritmaları art arda kullanıldığında, yukarıda verilen sonuçlardan farklı değerler elde edilmektedir. Tablo 4.10'da RABCO algoritması, RABCO-NM algoritması ve RABCO–Güçlendirilmiş Weiszfeld **(λ=1.8)**–NM hibrit algoritması ile elde edilen sonuçlar bulunmaktadır. Burada tutarlılık sağlamak amacıyla yine **RABCO-NM** algoritması kullanıldığında elde edilen sonuçlar listelenmiştir.

Tablo 4.10: RABCO, **Güçlendirilmiş Weiszfeld** (λ=1.8) ve Nelder-Mead algoritmalarının birlikte kullanımı sonuçları

Veri Kümesi	RABCO	RABCO-NM	İyileştirme 1 (%)	RABCO – Güçlendirilmiş Weiszfeld (λ=1.8) – NM	İyileştirme 2 (%)
Bongartz et al. (1994) (10)	7.627,00	7.257,00	% 4,85	7.449,45	% -2,65
Bongartz et al. (1994) (20)	4.622,00	4.463,00	% 3,44	4.416,14	% 1,05
SJC3a (25)	1.435.737,00	1.397.181,00	% 2,69	1.399.164,12	% -0,14
SJC3a (40)	1.035.707,00	996.274,00	% 3,81	1.002.756,96	% -0,65
SJC4a (25)	2.374.352,00	2.328.186,00	% 1,94	2.333.807,85	% -0,24
SJC4a (40)	1.840.615,00	1.755.552,00	% 4,62	1.784.540,44	% -1,65
SJC324 (20)	1.881.620,00	1.849.145,00	% 1,73	1.847.749,44	% 0,08
SJC324 (40)	1.163.306,00	1.122.982,00	% 3,47	1.135.966,77	% -1,16
SJC324 (60)	850.630,00	800.439,00	% 5,9	807.344,50	% -0,86
SJC500 (20)	4.269.917,00	4.126.927,00	% 3,35	4.167.349,47	% -0,98
SJC500 (40)	2.723.926,00	2.662.007,00	% 2,27	2.664.204,26	% -0,08
SJC500 (60)	2.091.680,00	2.014.274,00	% 3,7	2.041.067,08	% -1,33
SJC708 (20)	6.017.948,00	5.924.470,00	% 1,55	5.939.155,23	% -0,25
SJC708 (40)	3.998.800,00	3.734.418,00	% 6,61	3.855.361,15	% -3,24
SJC708 (60)	2.989.036,00	2.869.925,00	% 3,98	2.931.340,58	% -2,14

SJC818 (20)	7.624.484,00	7.507.594,00	% 1,53	7.524.885,92	% -0,23
SJC818 (40)	5.250.380,00	5.038.369,00	% 4,04	5.090.910,25	% -1,04
SJC818 (60)	3.873.383,00	3.719.948,00	% 3,96	3.792.976,23	% -1,96
CPmedcap2 (25)	114.575,00	113.657,00	% 0,8	113.821,86	% -0,15
CPmedcap2 (40)	88.556,00	87.383,00	% 1,32	87.711,65	% -0,38
Taillard (2003) (25)	656.694.127	645.719.011	% 1,67	645.594.173	% 0,02
Taillard (2003) (50)	452.711.742	439.571.070	% 2,9	442.421.557	% -0,65
Taillard (2003) (100)	302.172.907	290.541.198	% 3,85	294.848.609	% -1,48
Taillard (2003) (200)	192.204.730	185.953.715	% 3,25	182.690.973,28	% 1,75
ORTALAMA:			**% 3,22**		**% -0,77**

Tablo 4.10'da İyileştirme 1 başlıklı sütunda RABCO – NM algoritması sonuçlarının RABCO algoritmasına göre olan iyileştirme yüzdeleri verilmiştir. İyileştirme 2 başlıklı sütunda ise RABCO – Güçlendirilmş Weiszfeld (λ=1.8) – NM algoritmasının RABCO – NM algoritmasına göre olan iyileştirme yüzdeleri verilmiştir.

Tablo 4.10'daki sonuçlardan da görüleceği üzere RABCO – Weiszfeld – Nelder-Mead sıralı üçlü algoritmayla elde edilen sonuçlar RABCO – NM ikili algoritmasına göre ortalama % -0,77 oranında daha kötü sonuç vermiştir. Örnek kümelerden sadece dört tanesinde – Bongartz(20), SJC324(20), Taillard(25) ve Taillard(200) – üçlü algoritma iyi sonuç verirken diğer 20 veride daha kötü sonuç vermiştir.

Üçlü algoritmada sıralama değiştirilerek örnek veri kümeleri üzerinde uygulama yapıldığında, farklı sonuçlar elde edildiği görülmüştür. Genel olarak üçlü algoritma ile elde edilen sonuçların Tablo 4.3 ve Tablo 4.7'de gösterildiği gibi ikili algoritmalardan daha iyi olduğu görülmektedir. Aşağıdaki Tablo 4.11'de ise yukarıda verilmiş olan üçlü algoritmalardan elde edilen sonuçlar gösterilmiştir.

Tablo 4.11 Üçlü algoritmanın karşılaştırılması

	RABCO	RABCO - Nelder-Mead	RWFCM - Güçlendirilmiş WEISZFELD (λ=1.8) - NM	RWFCM-NM-Güçlendirilmiş WEISZFELD (λ=1.8)
Bongartz (10)	7.627,66	7.257,00	7.449,45	7.235,00
Bongartz (20)	4.622,84	4.463,00	4.416,14	4.456,50
SJC3a (25)	1.435.737,80	1.397.181,00	1.399.164,12	1.396.349,69
SJC3a (40)	1.035.707,47	996.274,00	1.002.756,96	992.014,52
SJC4a (25)	2.374.352,64	2.328.186,00	2.333.807,85	2.327.051,32
SJC4a (40)	1.840.615,85	1.755.552,00	1.784.540,44	1.754.412,66
SJC324 (20)	1.881.620,88	1.849.145,00	1.847.749,44	1.848.516,20
SJC324 (40)	1.163.306,30	1.122.982,00	1.135.966,77	1.121.603,54
SJC324 (60)	850.630,05	800.439,00	807.344,50	786.695,27
SJC500 (20)	4.269.917,31	4.126.927,00	4.167.349,47	4.123.970,63
SJC500 (40)	2.723.926,40	2.662.007,00	2.664.204,26	2.658.266,07
SJC500 (60)	2.091.680,61	2.014.274,00	2.041.067,08	2.006.888,02
SCJ708 (20)	6.017.948,21	5.924.470,00	5.939.155,23	5.922.807,63
SCJ708 (40)	3.998.800,63	3.734.418,00	3.855.361,15	3.724.602,82
SCJ708 (60)	2.989.036,37	2.869.925,00	2.931.340,58	2.865.043,62
SCJ818 (20)	7.624.484,87	7.507.594,00	7.524.885,92	7.506.935,38
SCJ818 (40)	5.250.380,21	5.038.369,00	5.090.910,25	5.013.005,39
SCJ818 (60)	3.873.383,84	3.719.948,00	3.792.976,23	3.712.229,77
CPmedcap2 (25)	114.575,00	113.657,00	113.821,86	113.652,61
CPmedcap2 (40)	88.556,55	87.383,00	87.711,65	87.373,82
Thaillard (25)	656.694.127,25	645.719.011	645.594.173	645.441.556,42
Thaillard (50)	452.711.742,55	439.571.070	442.421.557	438.774.767,71
Thaillard (100)	302.172.906,84	290.541.198	294.848.609	287.108.176,94
Thaillard (200)	192.204.730,40	185.953.715	182.690.961	182.459.835,46

Tablo 4.11'de görüldüğü üzere uyguladığımız 24 veri kümesinden 21 tanesinde RABCO–NM–Güçlendirilmiş Weiszfeld (λ=1,8) algoritması en iyi sonucu vermiştir. Sadece 2 tanesinde ise RABCO–Güçlendirilmiş Weiszfeld (λ=1,8)–NM algoritmasının en iyi sonucu verdiği görülmektedir.

Yukarıda RABCO ile RABCO – NM algoritmalarının sonuçları karşılaştırılmış ve iyileştirme oranları verilmişti (Tablo 4.8). Aşağıdaki tabloda da aynı şekilde üçlü algoritmaların RABCO – NM algoritmasına göre iyileştirme oranları listelenmiştir.

Tablo 4.12 İyileştirme oranları

	RABCO - Nelder-Mead	RWFCM - Güçlendirilmiş WEISZFELD - NM	RWFCM-NM- Güçlendirilmiş WEISZFELD
Bongartz (10)	7.257,00	% -2,65	% 0,30
Bongartz (20)	4.463,60	% 1,05	% 0,15
SJC3a (25)	1.397.179,93	% -0,14	% 0,06
SJC3a (40)	996.317,26	% -0,65	% 0,43
SJC4a (25)	2.328.209,70	% -0,24	% 0,05
SJC4a (40)	1.755.768,26	% -1,65	% 0,06
SJC324 (20)	1.849.136,28	% 0,08	% 0,03
SJC324 (40)	1.122.975,70	% -1,16	% 0,12
SJC324 (60)	800.428,66	% -0,86	% 1,72
SJC500 (20)	4.126.946,99	% -0,98	% 0,07
SJC500 (40)	2.661.951,54	% -0,08	% 0,14
SJC500 (60)	2.014.273,98	% -1,33	% 0,37
SCJ708 (20)	5.924.470,05	% -0,25	% 0,03
SCJ708 (40)	3.734.635,50	% -3,24	% 0,26
SCJ708 (60)	2.869.925,22	% -2,14	% 0,17
SCJ818 (20)	7.507.598,42	% -0,23	% 0,01
SCJ818 (40)	5.038.369,47	% -1,04	% 0,5
SCJ818 (60)	3.719.614,65	% -1,96	% 0,21
CPmedcap2 (25)	113.657,04	% -0,15	% 0
CPmedcap2 (40)	87.383,40	% -0,38	% 0,01
Thaillard (25)	645.719.011	% 0,02	% 0,04
Thaillard (50)	439.571.079	% -0,65	% 0,18
Thaillard (100)	290.541.198	% -1,48	% 1,18
Thaillard (200)	186.007.023	% 1,75	% 1,88
ORTALAMA:		% -0,77	% 0,33

Tablo 4.12'de görüldüğü üzere RABCO - Güçlendirilmiş WEISZFELD (λ=1,8) - NM sıralı algoritmasının RABCO-NM algoritmasına göre sadece 4 adet örnek veri kümesinde daha iyi sonuç verdiği görülmüştür. RABCO – NM – Güçlendirilmiş WEISZFELD (λ=1,8) sıralı algoritmasının ise tüm veri kümelerinde, RABCO – NM algoritmasından daha iyi sonuç verdiği görülmüştür.

5. TARTIŞMA VE SONUÇ

Özellikle 20.yy başında endüstriyel üretimin artmasıyla birlikte maliyetlerin azaltılması gereksinimi ortaya çıkmış ve giderek önem kazanmıştır. Weber (1909) tarafından yapılan ilk çalışmaların ardından bu konu, birçok disiplinden bilim insanının ve mühendisin ilgi alanı olmuştur.

Problem özünde yerleri önceden belli talep noktalarına, ihtiyaçları olan ürün/hizmetin, toplam ulaştırma maliyetini en aza indirecek şekilde belirli sayıda kaynak noktasının yerini tespit etmektir. Ancak problem çözümü $O(n^2)$ düzeyinde olduğu için anlamlı bir zaman diliminde kesin bir çözüm bulunamamaktadır. Bu nedenle optimum çözümler verecek yöntemler araştırılmıştır. Özellikle 1960'lardan itibaren bilgisayar teknolojisinin gelişimi ile birlikte de çözüm yöntemlerinin denenmesi ve buna dayanarak geliştirilmesi hızlanmıştır.

Uygulama alanının çok kapsamlı olması nedeniyle, problem çeşitli şekillerde ve şartlarla tanımlanmış ve buna bağlı olarak birbirlerinden farklı çok sayıda çözüm yöntemi üretilmiştir. Bunun sonucunda çok zengin bir literatür ortaya çıkmıştır.

İlk çözüm yöntemlerinden biri Weiszfeld (1937) tarafından önerilen, sürekli bir fonksiyonun minimum noktasında gradyan değerinin 0 olacağı teoremine dayanarak, bu noktaya ardışık iterasyonlarla erişebilmesini sağlayan algoritmadır. Weiszfeld algoritması diye adlandırılan yöntem, 1960'lı yıllarda araştırmacıların dikkatini çekmiş ve yapılan çalışmalarla Tek Kaynaklı Weber Probleminin çözümünde kullanılabilir hale getirilmiştir. Miehle (1958), Kuhn ve Kuenne (1962), Kuhn (1973), Chandrasekaran ve Tamir (1989) algoritmanın yeniden kurulması ve yakınsaklık sorunu üzerinde durmuştur. Ostresh (1978), Chen (1984), Drezner(1992) ve Drezner(2015) algoritmanın daha hızlı çalıştırılabilmesi için farklı kombinasyonlarını üretmişlerdir.

Zadeh (1965) tarafından oluşturulan Bulanık Mantık, Dunn(1973) ve Bezdek(1981) tarafından kullanılarak *Bulanık C-Ortalamalar* yöntemi üretilmiştir. Öbekleme analizindeki çalışmalar için üretilen yöntem 1990'lardan itibaren TYS problemlerinde de kullanılır olmuştur. Esnaf ve Küçükdeniz (2013) BCO yöntemini geliştirerek ürettikleri *Revize Ağırlıklı Bulanık C-Ortalamalar (RABCO)* yöntemini Çok Kaynaklı Weber probleminde kullanmışlardır.

Bu tez kapsamında Tesis Yeri Seçimi problemlerinin önemli bir sınıfı olan Çok Kaynaklı Weber Problemi için RABCO, Weiszfeld ve Nelder-Mead algoritmalarının farklı kombinasyonlarla birlikte kullanılmasından oluşan yeni bir çözüm yöntemi ileri sürülmüştür.

Bu hibrit yöntemde önce RABCO algoritması ile veri kümesini önceden belirlenmiş sayıda alt kümeye ayırmakta ardından, bunlara ait alt küme merkezleri Weiszfeld, Modifiye Weiszfeld, Güçlendirilmiş Weiszfeld veya Nelder-Mead algoritmalarından biri veya ikisi ile birlikte iyileştirmektedir. Böylelikle maliyet fonksiyonunun yakın optimum sonuçları elde edilebilmektedir.

Önerilen hibrit algoritmada farklı yöntemler sırayla kullanıldığından birbirlerinden farklı sonuçlar veren çeşitli kombinasyonlar bulunmaktadır. Aşağıda bunlar listelenmiştir:

- RABCO – Weiszfeld (λ=1) algoritması
- RABCO – Modifiye Weiszfeld (λ=1.8) algoritması
- RABCO – Güçlendirilmiş Weiszfeld (λ=1) algoritması
- RABCO – Nelder-Mead algoritması
- RABCO – Güçlendirilmiş Weiszfeld (λ=1,8) –Nelder-Mead algoritması
- RABCO – Nelder-Mead – Güçlendirilmiş Weiszfeld (λ=1,8) algoritması

Tüm bunlar kümeleme işlemlerinde sıklıkla kullanılan örnek veri kümelerine uygulanarak denenmiştir. Karşılaştırmaların sağlıklı yapılabilmesi için daha önce RABCO yönteminde (Esnaf ve Küçükdeniz, 2013) kullanılan veri kümeleri özellikle seçilmiştir.

Denemeler sonucunda hibrit algoritmada RABCO yönteminden sonra Weiszfeld, Modifiye Weiszfeld ve Güçlendirilmiş Weiszfeld yöntemlerinden biri kullanılarak elde edilen sonuçların birbirlerine çok yakın olduğu ortaya çıkmıştır. RABCO – Weiszfeld hibrit algoritması ortalama % 2,114 oranında, RABCO – Modifiye Weiszfeld ($\boldsymbol{\lambda = 1,8}$) hibrit algoritması ortalama % 2,118 oranında, RABCO – Güçlendirilmiş Weiszfeld ($\boldsymbol{\lambda = 1,0}$) hibrit algoritması ortalama % 2,116 oranında ve RABCO – Güçlendirilmiş Weiszfeld ($\boldsymbol{\lambda = 1,8}$) hibrit algoritması ise ortalama % 2,119 oranında iyileştirme sağlamaktadır.

Esnaf ve Küçükdeniz (2018) tarafından önerilen RABCO – Nelder-Mead algoritması Küçükdeniz ve Esnaf (2013) tarafından önerilen RABCO algoritmasına göre ortalama %3,22 oranında iyileştirme sağlamaktadır.

Bu aşamada iki farklı kombinasyon yerine üç farklı kombinasyon kullanılarak daha iyi sonuçlar elde edilebileceği gösterilmiştir. Talep noktaları RABCO metodu ile alt kümelere ayrıldıktan sonra ilgili alt küme merkezleri sırasıyla Güçlendirilmiş Weiszfeld algoritması (λ=1,8) ve Nelder-Mead algoritması ile iyileştirildiğinde maliyet fonksiyonu ortalama %2,48 oranında iyileşmektedir. Farklı olarak, talep noktaları RABCO metodu ile alt kümelere ayrıldıktan sonra tüm alt küme merkezleri sırasıyla Nelder-Mead algoritması ve Güçlendirilmiş Weiszfeld (λ=1,8) algoritması ile iyileştirildiğinde ise maliyet fonksiyonu ortalama %3,62 oranında iyileştirilmiştir. Aşağıdaki Tablo 5.1'de önerilen hibrit metodun tüm kombinasyonlarının, RABCO yöntemine göre elde ettikleri ortalama iyileştirme oranları listelenmiştir.

Tablo 5.1: İkili ve Üçlü Kombinasyonların RABCO yöntemine göre iyileştirmelerinin karşılaştırması

	RABCO
RABCO - NM	**%3,2177**
RABCO – WEİSZFELD (λ=1,0)	**%2,1139**
RABCO - MODİFİYE WEİSZFELD (λ=1,8)	**%2,1157**
RABCO - GÜÇLENDİRİLMİŞ WEİSZFELD (λ=1,0)	**%2,1185**
RABCO - GÜÇLENDİRİLMİŞ WEİSZFELD (λ=1,8)	**%2,1192**
RABCO - NM - GÜÇLENDİRİLMİŞ WEİSZFELD (λ=1,8)	**%3,6254**
RABCO - GÜÇLENDİRİLMİŞ WEİSZFELD (λ=1,8) **- NM**	**%2,4891**

Tablo 5.2'de ise RABCO-Güçlendirilmiş Weiszfeld algoritması (λ=1,8)-NM ve RABCO-NM-Güçlendirilmiş Weiszfeld algoritması (λ=1,8) algoritmalarının RABCO-NM algoritmasına göre iyileştirmelerinin karşılaştırması verilmiştir.

Tablo 5.2 Üçlü Kombinasyonların RABCO-NM yöntemine göre iyileştirmelerinin karşılaştırması

	RABCO-NM
RABCO - NM - GÜÇLENDİRİLMİŞ WEİSZFELD (λ=1,8)	**% 0,34**
RABCO - GÜÇLENDİRİLMİŞ WEİSZFELD (λ=1,8) **- NM**	**% -0,77**

Çok Kaynaklı Weber probleminin çözümü için hibrit yöntemlerin kullanılması özellikle 2010 lu yılların başından itibaren dikkat çekmeye başlamıştır. Özellikle BCO ve sezgisel algoritmaların birleştirilerek kullanılması dikkat çekmektedir. Mehdizeded ve diğ. (2008) BCO ile Parçacık Sürüsü Optimizasyonu (PSO) yöntemleri ile oluşturulan hibrit bir metot ile BCO'ya göre daha iyi sonuçlar elde edilebileceğini göstermişlerdir. Kaur ve Kaur (2013) BCO ile Genetik Algoritmadan oluşan hibrit bir metodu kapalı sistemlere olan sızmaların belirlenmesinde kullanmıştır.

Esnaf ve Küçükdeniz (2018) geometrik bir yöntem olan Nelder-Mead Algoritması ile Revize Ağırlıklı BCO yöntemini birleştirerek Çok Kaynaklı Weber problemi için optimal sonuçlar elde edildiğini göstermişlerdir.

Weber probleminin çözümü için, Weiszfeld algoritması kullanılarak oluşturulmuş hibrit bir metot önerisine rastlanmamıştır. Aynı şekilde üç farklı algoritmanın birlikte kullanıldığı bir yönteme veya çalışmaya da rastlanmamıştır.

Bu tezde elde edilen sonuçlara göre Weber problemini çözerken kullanılan bir yöntemin ardından Weiszfeld algoritmaları ile daha yakın optimal sonuçlara erişmek mümkün olmaktadır. Aynı şekilde farklı metasezgisel yöntemler kullanılarak elde edilen sonuçlar Nelder-Mead ve Weiszfeld algoritmaları kullanılarak daha yakın optimal hale getirilebilir.

Gelecek çalışması olarak bu tez çalışmasında önerilen ikili ve üçlü hibrit algoritmalar farklı veri setleri üzerinde de denenecektir. Ayrıca önerilen algoritmalar devam eden çalışmalarda talebin ve konumun stokastik olduğu verilere uygulanabilir bir hale getirilmesine de çalışılacaktır.

6. KAYNAKLAR

Bastı, M., 2012, P.medyan Tesis Yeri Seçim Problemi ve Çözüm Yaklaşımları, *Online Academic Journal of Information Technology*, Spring/Bahar 2012 – Cilt/Vol: 3 . Sayı/Num: 7

Beck, A., Sabach, S., 2010, Weiszfeld's Method: Old and New Results, *Journal of Optimization Theory Applications* (2015) 164:1–40DOI 10,1007/ s10957 . 014.0586.7,

Arabani, A., Farahani, R. Z., 2011, Facility location dynamics: An overview of classifiations and applications, *Computers & Industrial Engineering*, 62(1), 408.420,

Bezdek, J.,C., 1981, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York,

Brimberg, J., 1995, The Fermat.Weber location problem revisited, *Mathematical Programming* 71, 71–76,

Brimberg, J., Chen, R., Chen, D., 1998, Accelerating Convergence in The Fermat-Weber Location Problem Revisited, *Operations Research Letters*, 151–157,

Cánovas, L., Cañavate, R., Marín, A., 2002, On the convergence of the Weiszfeld algorithm, *Mathematical Programming,* 93 327–330,

Chandrasekaran, R., Tamir, A., 1989, Open questions concerning Weiszfeld's algorithm for the Fermat.Weber location problem, *Mathematical Programming,* 44 293–295,

Chen, R., 1984, Solution of location problems with radial cost functions, *Computers and Mathematics with Applications,* 10 87–94,

Cooper, L., 1963, Location.Allocation Problems, *Operations Research* Vol 11 No:3 pp 331.343,

Cooper, L., 1964, Heuristic Methods For Location.Allocation Problems, *SIAM Review* 6, 37.53,

Daskin, M.S., 1995, *Network and Discrete Location: Models, Algorithms, and Applications*, John Wiley & Sons, Inc,, New York,

Drezner, Z., 1992, A note on the Weber location problem, *Annals of Operations*

Research, 40, 153–161

Drezner, Z., 1995, A note on accelerating the Weiszfeld procedure, *Location Science*, Vol3, No,4, 275–279

Drezner, Z., 2015, The fortified Weiszfeld algorithm for solving the Weber problem, *IMA Journal of Management Mathematics,* 26, 1–9 doi:10,1093/imaman/dpt019,

Drezner, Z., Brimberg, J., Mladenoviç, N., Salhi, S., 2016, New local searches for solving the multi-source Weber problem, *Annual Operation Researches,246*, 181–203

Dunn, J., 1973, A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well.Separated Clusters, *Journal of Cybernetics*, 3: 32.57,

Eckhardt, U., 1980, Weber's Problem and Weiszfeld's Algorithm in General Spaces, *Journal Mathematical Programming* 18 (1980) 186.196, North Holland Publishing Company,

Esnaf, Ş., Küçükdeniz, T., 2009, A Fuzzy Clustering.based Hybrid Method for a Multi.Facility Location Problem, *Journal of Intelligent Manufacturing* Vol, 20, pp, 259–265,

Esnaf Ş., Küçükdeniz T., 2013, Solving Uncapacitated Planar Multi.facility Location Problems by a Revised Weighted Fuzzy c.means Clustering Algorithm, *Journal of Multiple.Valued Logic and Soft Computing*, Vol, 21, pp, 147–164,

Esnaf Ş., Küçükdeniz T., 2018, Hybrid Revised Weighted Fuzzy C . Means Clustering with Nelder-Mead Simplex Algorithm for Generalized Multisource Weber Problem, *Journal of Enterprise Information Management*, Vol, 31, issue:6, pp, 908–924,

Gamal, M.D.H., Salhi, S., 2003, A cellular heuristic for the multisource weber problem, *Computers & Operations Research,* Vol. 30, No. 11, pp. 1609-1624

Hakimi, S,L., 1964, Optimum Location of Switching Centers and the Absolute Centers and Medians of a Graph, *Operations Research*, Vol,12, No:3, pp,450.459,

Hakimi, S. L., 2000, *Location theory, Handbook of Discrete and Combinatorial Mathematics*, In: Rosen, K, H,, Michaels, J, G,, Gross, J, L,, Grossman, J, W,, Shier, D, R, (editors), CRC Press, Boca Raton, London, ISBN: 0849301491, 1019.1029,

Hooke, R., Jeeves, T.A., 1961, Direct Search Solution of Numerical and Statistical Problems, *Journal of Ass, Computer Mach*, 8, pp, 212–229,

Iyigun C., Ben.Israel A., 2010, A generalized Weiszfeld method for the multi.facility location problem, *Operations Research Letter*, 38(2010) 207.214,

İyigün C., Ben.Israel A., 2012, The Multy.Facility Location Problem: A

Probabilistic Decomposition Method, http://benisrael,net/LOCATION.JUL.29.12,pdf

Jamshidi, M., 2009, *Median Location Problem*, Facility Location: Concepts, Models, Algorithms and Case Studies, In: R,Z, Farahani and M, Hekmatfar (ed), PhysicaVerlag Heidelberg, pp,177.191,

Kaur, K., Kaur, N., 2013, A Hybrid Approach of Fuzzy C.Means Clustering and Genetic Algorithm (GA) to Improve Introsion Detection Rate, *International Journal of Science and Reseasrch (IJSR)*, ISSN (Online): 2319.7064,

Kennedy, J., R, Eberhart., 1995, Particle Swarm Optimization, *Proceedings of the IEEE International Conference on Neural Networks*, 1995, Perth, Australia, pp, 1942–1945,

Khurrum, A., Hartley, R., 2015, Generalized Weiszfeld Alogoirthms for Lq Optimization, *Math, IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol,37, No,4,

Kuenne, R.E., Soland, R.M., 1972, Exact and approximate solutions to the multisource Weber problem, *Mathematical Programming* 3, 193–209,

Kuhn, H.W., Kuenne, R.E., 1962, An efficient algorithm for the numerical solution of the generalized Weber problem in spatial economics, *Journal of Regional Science,* 4, 21–34

Kuhn, H.W., 1973, A note on Fermat's problem, *Mathematical Programming,* 4, 98–107

Küçükdeniz T., Baray A., Ecerkale K., Esnaf Ş., 2012, Integrated use of fuzzy c.means and convex programming for capacitated multi.facility location problem, *Expert Systems with Applications*, 39(4), 4306.4314,

Küçükdeniz T., Esnaf Ş., 2018, Hybrid revised weighted fuzzy C.means clustering with Nelder-Mead simplex algorithm for generalized multisource Weber problemIntegrated use of fuzzy c.means and convex programming for capacitated multi.facility location problem, *Journal of Enterprise Information Management*, Vol, 31 Issue: 6, pp,908.924,

Lagarias, J.C., Reeds, J.A., Wright, M.H., Wright, P,E,, 1998, Convergence properties of the Nelder–mead simplex method in low dimensions, *SIAM Journal on Optimization*, Vol, 9 No, 1, pp, 112.147,

Levin Y., Ben-Israel A., 2004, A Heuristic Method For Large.scale Multi.facility Location Problems, *Computers & Operations Research* 31 (2): 257.272,

Logendran, R., Terrell, M. P., 1988, Uncapacitated plant location.allocation problems with price sensitive stochastic demands, *Computers and Operations*

Research 15, 189–198,

Love R.F., Morris J. G., Wesolowsky G, O,, 1988, *Facilities Location: Models & Methods*, New York, NY: North Holland,

Lozano S., Guerrero F., Onieva L., Larraneta J., 1998, Kohonen maps for solving a class of location.allocation problems, *Europen Journal of Operational Research* 108 / 106.117,

Mehdizeded, E., Sadinezhad, S., Tavakkolimoghaddam, R., 2008, Optimization of Fuzzy Criteria by a Hybrid PSO and Fuzzy C.Means Clustering Algorithm, *Iranian Journal of Fuzzy Systems*, pp, 1.14,

Mezura-Montes, E., Coello C.A., Constraint-handling in Nature-inspired Numerical Optimization: Past, Present and Future, *Swarm and Evolutionary Computation*, 2011, pp, 173–194,

Miehle, W., 1958, Link-Length Minimization in Networks, *Operations Research* 6, 232–243,

Nelder, J.A., Mead, R., 1965, A Simplex Method for Function Minimization, *The Computer Journal*, Vol, 7 No, 4, pp, 308.313,

Ostresh, L.M., 1978, On the Convergence of a Class of Iterative Methods for Ssolving the Weber Location Problem, *Operations Research*, 26, 597–609,

Rosing, K.E., 1992, An Optimal Method for Solving the (generalized) Multi-Weber Problem, *European Jornal of Operational Research*, 58, 414-426,

Salhi , S., Gamal, M.D.H., 2003. A Genetic Algorithm Based Approach for the Uncapacitated Continous Location Problem, *Annals of Operational Research*, 97, 203-222

Spendley, W., Hext, G.R., Himsworth, F.R., 1962, Sequential Application of Simplex Designs in Optimisation and Evolutionary Operation, *Technometrics*, Vol, 4, pp, 441–461,

Sule, D.R., 2001, *Logistics of facility location and allocation*, Marcel Dekker Inc,, New York,

Tsekouras G., Sarimveis H., Kavakli E., Bafas G., 2005, A Hierarchical Fuzzy Clustering Approach to Fuzzy Modeling, *Fuzzy Sets and Systems*, 150(2), 245-266,

Tsekouras G. E., 2005, On the Use of the Weighted Fuzzy C-Means in Fuzzy Modeling, *Advances in Engineering Software*, 36(5), 287-300,

Tunçbilek, N., Taşgetiren, F., Esnaf, Ş., 2012, Artificial Bee Colony Optimization Algorithm For Uncapacitated Facility Location Problems, *Journal of Economic and*

Social Research, Volume 14(1), ISSN 1302-1060

Wang, X., Wang, Y., Wang, L., 2004, Improving Fuzzy C-Means Clustering Based on Feature Weight Learning, *Pattern Recognition Letters*, 25(10), 1123-1132,

Weber A., 1909, *Über den Standort der Industrien, 1, Teil: Reine Theorie Des Standortes*, Tübingen, English Translation: *On the Location of Industries*, University of Chicago Press, Chicago, 1929,

Weiszfeld, E., 1936, Sur le point pour lequel la somme des distances de n points donnes est minimum, *Tohoku Math, J,*, 43, 355–386,

Weiszfeld, E., Plastra, F., 2008, On the point for which the sum of the distances to n given points is minimum, *Annals of Operations Research*, (2009) 167: 7–41,

7. EKLER

8. ÖZGEÇMİŞ

Fotoğraf
(Bilgisayar
Çıktılı)

Kişisel Bilgiler

Adı Soyadı	M. Işık Güzelgöz
Uyruğu	T.C.
Doğum tarihi, Yeri	1970 / Diyarbakır
Telefon	555 843 77 52
E.mail	isikguzelgoz@gmail,com
Web adres	

Eğitim

Derece	Kurum/Anabilim Dalı/Programı	Yılı
Doktora	İstanbul Cerrahpaşa Üniversitesi Fen Bilimleri Enstitüsü/	
Yüksek Lisans	ODTÜ Fen Bilimleri Enstitüsü/ Matematik	1996
Lisans	Boğaziçi Üniversitesi Fen Fakültesi / Matematik	1993
Lise	Küçükçekmece Lisesi	1987

Makaleler / Bildiriler

Revised Weighted Fuzzy C-Means and Fortified Weiszfeld Hybrid Method for Uncapacitated Multi-Facility Location Problems – M. Işık Güzelgöz, Şakir Esnaf, Tarık Küçükdeniz, International Journal of Operations Research and Information Systems (IJORIS), 10(4) 2019, 53-71.