



T.C.
KONYA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ



**ÇOK TESİSLİ BÜTÜNLEŞİK ÜRETİM VE
DAĞITIM ÇİZELGELEME PROBLEMLERİ
İÇİN ÇÖZÜM YÖNTEMLERİ**

Gözde CAN ATASAGUN

DOKTORA TEZİ

Endüstri Mühendisliği Anabilim Dalı

Haziran-2022
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Gözde CAN ATASAGUN tarafından hazırlanan “Çok Tesisli Bütünleşik Üretim ve Dağıtım Çizelgeleme Problemleri İçin Çözüm Yöntemleri” adlı tez çalışması 20/06/2022 tarihinde aşağıdaki jüri tarafından oy birliği ile Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü Endüstri Mühendisliği Anabilim Dalı’nda DOKTORA TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Başkan

Prof. Dr. Saadettin Erhan KESEN

.....

Danışman

Prof. Dr. İsmail KARAOĞLAN

.....

Üye

Doç. Dr. Talip KELLEGÖZ

.....

Üye

Doç. Dr. Çağrı KOÇ

.....

Üye

Dr. Öğretim Üyesi Alper DÖYEN

.....

Yukarıdaki sonucu onaylarım.

Prof. Dr. Saadettin Erhan KESEN
Enstitü Müdürü

Bu tez çalışması Konya Teknik Üniversitesi Bilimsel Araştırma Projeleri Koordinatörlüğü (BAP) tarafından 18101021 numaralı proje ile desteklenmiştir.

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Gözde CAN ATASAGUN

Tarih: 20.06.2022

ÖZET

DOKTORA TEZİ

ÇOK TESİSLİ BÜTÜNLEŞİK ÜRETİM VE DAĞITIM ÇİZELGELEME PROBLEMLERİ İÇİN ÇÖZÜM YÖNTEMLERİ

Gözde CAN ATASAGUN

Konya Teknik Üniversitesi
Lisansüstü Eğitim Enstitüsü
Endüstri Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. İsmail KARAOĞLAN

2022, 128 Sayfa

Jüri

Prof. Dr. İsmail KARAOĞLAN
Prof. Dr. Saadettin Erhan KESEN
Doç. Dr. Talip KELLEGÖZ
Doç. Dr. Çağrı KOÇ
Dr. Öğr. Üyesi Alper DÖYEN

Günümüzde firmalar diğer firmalarla rekabet edebilmek için üretim ve dağıtım operasyonlarını birlikte yürütmek zorundadır. Üretim faaliyetleri, müşteri siparişlerinin tesis içerisindeki işlem sıraları ve bu işlemlerin çizelgelenmesi operasyonlarını, dağıtım faaliyetleri ise üretilen ürünlerin müşterilere teslim sürecini kapsamaktadır. Zamana duyarlı (bozulabilir) ürünler, üretimi tamamlandıktan sonra kısıtlı bir zaman diliminde müşteriye teslim edilmek zorundadır. Bozulabilir ürünler için bütünlük bir yaklaşımın uygulama alanlarına örnek olarak gazete, gıda ürünleri, hazır beton karışımları, nükleer ilaç ve endüstriyel yapıştırıcı malzemeleri üretim ve dağıtım verilebilir. Literatürde üretim planlama ve dağıtım planlamasının ayrı ayrı ele alındığı yaklaşım çok uzun yıllardır çalışılmasına rağmen bütünlük üretim ve dağıtım çizelgeleme problemi yaklaşık 20 yıldır yoğun olarak çalışılmaktadır. Bu çalışmaların çok büyük bir kısmında ise bozulabilir ürünler göz önüne alınmamıştır. Oysa bu tarz bozulabilir ürünlerin üretim ve dağıtımının planlanması, yaşanacak gecikmelerde ürünlerin kısmen veya tamamen değerlerini yitirmelerine yol açacağından, çok daha önemlidir.

Tez kapsamında literatürde Bütünlük Üretim ve Dağıtım Çizelgeleme (BÜDÇ) Problemi olarak adlandırılan problem ele alınmıştır. Bu problem bünyesinde makine çizelgeleme ve araç rotalama problemlerini barındırmaktadır. Çok sayıda çeşidi olan BÜDÇ problemi ele alınırken rotalama kararları da göz önüne alınmıştır. Rotalama kararlarının göz önüne alındığı bu problemde dağıtım aşamasında birden fazla müşterinin ziyaret edilmesine izin verilmektedir. Tez kapsamında literatürde daha önce ele alınmamış olan BÜDÇ problemleri incelenmiştir. Tez çalışmasında ilk olarak sistemde birden fazla tesisin bulunduğu ve müşterilere her tesiste bulunan tek araçla hizmet edilen Çok Tesisli ve Tek Araçlı Bütünlük Üretim ve Dağıtım Çizelgeleme Problemi (ÇT_TA_BÜDÇ) ele alınmıştır. Problemin çözümü için öncelikle bir matematiksel model geliştirilmiştir. Hem makine çizelgeleme hem de araç rotalama problemi NP-zor problem sınıfında yer aldığı için BÜDÇ probleminin de NP-zor yapıda olduğu bilinmektedir. Bu nedenle büyük boyutlu problemlerin çözümü için daha kısa sürelerde optimal ya da optime yakın çözümler elde edilebilmesi amacıyla Değişken Komşu Arama (DKA) Algoritması geliştirilmiştir. Geliştirilen matematiksel model ve DKA algoritması 6 farklı parametre üzerinden değerlendirilmiş ve sonuçlar karşılaştırılmıştır. Geliştirilen 2592 adet test probleminin matematiksel model ile çözülmesi sonucunda 7200 saniye süre sınırı içerisinde 833 adet test probleminde uygun bir çözüm elde edilememişken DKA algoritması ile tüm problemler için uygun bir çözüm bulunmuştur. Matematiksel model ile 74 adet test probleminde optimal sonuçlar elde edilmiştir ve bu problemlerin hepsi 10 müşteri problemi problemlerdir. Optimal sonuçların elde edildiği problemler için ortalama çözüm süresi 3085 saniyedir. DKA algoritması ile

matematiksel model ile elde edilen sonuçlar bir saniyenin altında bir sürede elde edilmiştir. Matematiksel model ile DKA algoritmasının sonuçları karşılaştırıldığı zaman elde edilen Yüzde Sapma Değeri (YSD) ortalama %15.16 olarak hesaplanmıştır. Matematiksel modelin çözümü için gereken süre ortalama 7083 saniye iken bu süre DKA algoritmasında yaklaşık 70 saniyedir. DKA algoritmasını hem süre hem de performans açısından çok daha iyi sonuçlar sağlamaktadır.

Tez çalışmasında çalışılan bir diğer problem ise yine sistemde birden çok tesisin bulunduğu ve tesislerde de birden çok aracın mevcut olduğu Çok Tesisli ve Çok Araçlı Bütünleşik Üretim ve Dağıtım Çizelgeleme Problemi (ÇT_ÇA_BÜDÇ)'dir. Problemin çözümü için ilk problemde olduğu gibi öncelikle bir matematiksel model geliştirilmiştir. Geliştirilen matematiksel model geliştirilen geçerli eşitsizlikler ile kuvvetlendirilmiştir. Büyük boyutlu problemler için daha kısa sürelerde optimal ya da optime yakın çözümler elde edilebilmesi amacıyla Memetik Algoritma (MA) geliştirilmiştir. ÇT_ÇA_BÜDÇ için gerçekleştirilen deneysel çalışmalar farklı parametre seviyeleri için ele alınmış ve sonuçlar raporlanmıştır. 2592 adet test probleminin 1283 adedinde uygun çözümler elde edilmiştir. Matematiksel modele göre elde edilen YSD değeri %0.48 olarak hesaplanmıştır. Bu sonuçların elde edilebilmesi için geçen yaklaşık ortalama süre 4920 saniyedir. Optimal çözümler ise yaklaşık ortalama 539 saniyede elde edilmiştir. MA ile bu çözümlerin elde edilmesi için gereken ortalama süre 3 saniyedir. Bu da algoritmanın ne kadar hızlı ve doğru sonuçlar verdiğini göstermektedir. Problemlerin MA ile çözümünden elde edilen en küçük değerlerin matematiksel model ile aralarındaki farka göre matematiksel model ve MA birbirine çok yakın sonuçlar vermektedir. Elde edilen en küçük değerler göz önüne alındığında MA %0.08 daha iyi sonuç vermektedir. MA ile elde edilen 5 farklı çözümün ortalaması göz önüne alındığında ise matematiksel model çözümlerinin ve MA çözümlerinden %0.01 daha iyi olduğu söylenebilir. Aynı ya da çok yakın sonuçlara MA ile çok kısa sürelerde ulaşılması mümkündür. MA algoritmasının çözümü için gereken süre en küçük 0.01 saniye iken en uzun sürede çözülen problem 26 saniyede çözülmüştür. Ayrıca MA ile tüm problemler için uygun bir çözüm elde edilmişken matematiksel model ile çözüm elde edilemeyen 1309 adet problem mevcuttur. Bu nedenle MA'nın matematiksel modele göre çok daha başarılı olduğu söylenebilir. Müşteri sayısı arttıkça matematiksel model ile uygun çözüm elde etmek zorlaşırken MA ile matematiksel modelden çok daha iyi sonuçlar elde edilmiştir. Matematiksel model ile 50 müşterilik 5 problem için uygun bir çözüm elde edilmiş bu problemlerin YSD değeri %11.03 olarak hesaplanmıştır. MA ile bu 5 problem için %7.17 daha iyi sonuçlar elde edilmiştir. Diğer tüm parametreler için MA ile elde edilen en küçük değerler matematiksel model ile elde edilen değerlerden daha iyidir.

Anahtar Kelimeler: Araç Rotalama Problemi, Bütünleşik Üretim ve Dağıtım Çizelgeleme Problemi, Değişken Komşu Arama Algoritması, Karma Tamsayı Programlama, Memetik Algoritma, Üretim Çizelgeleme Problemi

ABSTRACT

PhD THESIS

SOLUTION METHODS FOR INTEGRATED PRODUCTION AND OUTBOUND DISTRIBUTION SCHEDULING PROBLEM WITH MULTIPLE PLANTS

Gözde CAN ATASAGUN

**Konya Technical University
Institute of Graduate Studies
Department of Industrial Engineering**

Advisor: Prof. Dr. İsmail KARAOĞLAN

2022, 128 Pages

Jury

**Prof. Dr. İsmail KARAOĞLAN
Prof. Dr. Saadettin Erhan KESEN
Assoc. Prof. Dr. Talip KELLEGÖZ
Assoc. Prof. Dr. Çağrı KOÇ
Assist. Prof. Dr. Alper DÖYEN**

Nowadays, companies must carry out production and distribution operations together to compete with other companies. While the production activities include the order of operations of customer demands within the facility and the scheduling of these operations, distribution activities include the process of delivering the products to customers. One of these operations is time sensitive products. Time-sensitive (perishable) products must be delivered to the customer in a limited time, after their production is completed. Newspapers, food products, ready mixed concrete mixes, nuclear medicine, and industrial adhesive materials can be given as examples of application areas of an integrated approach for perishable products. Although production planning and distribution planning are studied separately for many years, the integrated production and distribution scheduling problem has been studied extensively for about 20 years. Perishable products are not considered in most of these studies. However, the planning of the production and distribution of perishable products is more important because of it will cause the products to lose their value partially or completely in delays.

In this thesis, the problem has been examined called as Integrated Production and Outbound Distribution Scheduling (IPODS) Problem in the literature. This problem includes machine scheduling and vehicle routing problems and has many variants. In this study, routing decisions are also considered allowing to visit more than one customer during the same tour. In this thesis, IPODS problems that have not been dealt with before in the literature have been examined. As a first problem, the Integrated Production and Outbound Distribution Scheduling Problem with Multiple Plants and Single Vehicle (IPODS_MP_SV) is discussed. In this problem, there are more than one plant in the system and customers are served with a single vehicle in each facility. A mathematical model is developed to solve the problem firstly. The Variable Neighborhood Search (VNS) Algorithm has been developed to obtain optimal or near-optimal solutions in a shorter time for the solution of large-scale problems. Developed mathematical model and the VNS algorithm are evaluated with 6 different parameters and compared the results. For the developed 2592 test problems, although the mathematical model could not be solved 833 test problems, the VNS algorithm found a feasible solution for all problems. Optimal results are obtained in 74 test problems with the mathematical model, and all these problems are in the case of 10 customers. The average solution times for problems with optimal results is 3085 seconds. The results obtained with the VNS algorithm, and the mathematical model were obtained in less than 1 seconds. When the results of the mathematical model and the VNS algorithm is compared, the difference between the GAP values is calculated as 15.16% on average. While the time required for the solution of the mathematical model is 7083 seconds on average,

this time is about 70 seconds in the VNS algorithm. The VNS algorithm provides much better results both in terms of time and performance.

Another study of the thesis is The Integrated Production and Outbound Distribution Scheduling Problem with Multiple Plants and Multiple Vehicles (IPODS_MP_MV). This problem includes many plants in the system, and more than one vehicle in these plants. For the solution to the problem, firstly a mathematical model was developed as in the first problem. The developed mathematical model is strengthened by the valid inequalities. Since the IPODS problem is known to be NP-hard class, the Memetic Algorithm (MA) has been developed to obtain optimal or near-optimal solutions for large-sized problems in a shorter time. Test instances for the IPODS_MP_MV are examined for different parameters and the results are reported. Optimal or feasible solutions are obtained in 1283 problems of 2592 test problems. The GAP value obtained according to the mathematical model was calculated as 0.48%. The GAP value is very close to "0" indicates that these results very close to the optimal result are obtained. The average time to obtain these results is 4920.55 seconds in mathematical model. Optimal solutions were obtained in an average of 538.98 seconds in mathematical model. The average time required to obtain these solutions with the MA is 3 seconds. This shows how fast and accurate results the algorithm gives. Considering that the smallest values obtained with MA, MA gives 0.08% better results. Considering that the average of 5 different solutions obtained with MA, it can be said that the mathematical model solutions 0.01% better than MA solutions. It is possible to say the same or very close results in a very short time with MA. The problems are solved by the MA algorithm at least 0.01 seconds, at most 26 seconds. In addition, while a feasible solution is obtained for all problems with MA, there are 1309 problems that cannot be solved with the mathematical model. Therefore, it can be said that the MA algorithm is more efficient than the mathematical model. As the number of customers increases, while it becomes difficult to obtain a feasible solution with the mathematical model, obtained much better results with MA. With the mathematical model, a feasible solution was obtained for 5 problems within the case of 50 customers, and the GAP value of these problems was calculated as 11.03%. 7.17% better results were obtained for these 5 problems with MA.

Keywords: Integrated Distribution and Scheduling Problem, Memetic Algorithm, Mixed Integer Programming, Production Scheduling Problem, Variable Neighborhood Search Algorithm, Vehicle Routing Problem,

ÖNSÖZ

Doktora sürecim boyunca değerli bilgi ve tecrübelerini büyük bir sabır ve anlayışla bana aktaran, yüksek lisans sürecimden itibaren danışmanlığımı yapmış, o günden bugüne kadar da bana her türlü fedakarlığı yaparak destek sağlayan ve alanında yetkin bir akademisyen olma yolunda kendisini örnek aldığım değerli hocam ve tez danışmanım Sayın Prof. Dr. İsmail KARAOĞLAN'a sonsuz teşekkürlerimi sunarım.

Doktora sürecimde kıymetli fikir ve önerilerini her daim benimle paylaşan değerli tez izleme komitesi üyesi hocalarım Sayın Prof. Dr. Saadettin Erhan KESEN'e ve Sayın Doç. Dr. Talip KELLEĞÖZ'e, doktora eğitimim boyunca bölümde her türlü fiziksel ve sosyal olanağı sağlayarak bizlere huzurlu bir çalışma ortamı sunan, başta Endüstri Mühendisliği Bölüm Başkanı Sayın Prof. Dr. Ahmet PEKER olmak üzere tüm bölüm hocalarıma ve meslektaşlarıma teşekkürlerimi sunarım.

Bu çalışmayı, günlere gelmemde şüphesiz ki en büyük emeğin sahibi, her an her koşulda yanımda olan canım annem Ümran ve babam Murat CAN'a, canımdan öte kardeşlerim Beyzanur ve Feyzanur'a, doktora çalışmalarım boyunca hem mesleki hem de manevi olarak sabır ve anlayışla beni her koşulda destekleyen değerli eşim ve meslektaşım Yakup ATASAGUN'a ve en güzel yaşlarında bensiz geçirdiği zamanlarının özü olarak canım oğlum Yusuf Yiğit'e ithaf ediyorum.

Gözde CAN ATASAGUN
KONYA-2022

İÇİNDEKİLER

ÖZET	iv
ABSTRACT.....	vi
ÖNSÖZ	viii
İÇİNDEKİLER	ix
SİMGELER VE KISALTMALAR	xi
1. GİRİŞ	1
2. BÜTÜNLEŞİK ÜRETİM VE DAĞITIM ÇİZELGELEME (BÜDÇ) PROBLEMİ VE LİTERATÜR TARAMASI.....	5
2.1. Problem Tanımı	5
2.2. Bütünleşik Üretim ve Dağıtım Çizelgeleme Problemi İçin Literatür Taraması ..	10
2.2.1. Tek makineli üretim ortamı için literatür taraması	10
2.2.2. Paralel makineli üretim ortamı için literatür taraması	19
2.2.3. Diğer üretim ortamları için literatür taraması	23
3. ÇOK TESİSLİ VE TEK ARAÇLI BÜTÜNLEŞİK ÜRETİM VE DAĞITIM ÇİZELGELEME (ÇT_TA_BÜDÇ) PROBLEMİ.....	29
3.1. Problem Tanımı ve Açıklayıcı Örnek	29
3.2. ÇT_TA_BÜDÇ Problemi İçin Önerilen Matematiksel Model.....	34
3.3. Değişken Komşu Arama (DKA) Algoritması	37
3.3.1. ÇT_TA_BÜDÇ problemi için önerilen değişken komşu arama (DKA) algoritması	39
3.3.2. DKA algoritmasının örnek probleme uygulanması	51
3.4. ÇT_TA_BÜDÇ Problemi İçin Deney Tasarımı	56
3.5. ÇT_TA_BÜDÇ Problemi İçin Deneysel Karşılaştırmalar	57
3.5.1. Önerilen matematiksel modelin performansı.....	57
3.5.2. Matematiksel model ve DKA algoritması ile elde edilen çözümlerin karşılaştırılması	63
4. ÇOK TESİSLİ VE ÇOK ARAÇLI BÜTÜNLEŞİK ÜRETİM VE DAĞITIM ÇİZELGELEME (ÇT_ÇA_BÜDÇ) PROBLEMİ	66
4.1. Problem Tanımı ve Açıklayıcı Örnek	66
4.2. ÇT_ÇA_BÜDÇ Problemi İçin Önerilen Matematiksel Model.....	72
4.3. Memetik Algoritma (MA)	77
4.3.1. ÇT_ÇA_BÜDÇ problemi için önerilen memetik algoritma (MA).....	80
4.3.2. MA'nın örnek probleme uygulanması	85
4.4. ÇT_ÇA_BÜDÇ Problemi İçin Deney Tasarımı	95
4.5. ÇT_ÇA_BÜDÇ Problemi İçin Deneysel Karşılaştırmalar	97
4.5.1. Önerilen matematiksel modelin performansı.....	97
4.5.2. Matematiksel model ve MA ile elde edilen çözümlerin karşılaştırılması ..	105

5. SONUÇLAR VE ÖNERİLER	109
KAYNAKLAR	113



SİMGELER VE KISALTMALAR

Kısaltmalar

AFD	: Amaç Fonksiyonu Deęeri
ARP	: Araç Rotalama Problemi
BÜDÇ	: Bütünleşik Üretim ve Dağıtım Çizelgeleme Problemi
ÇT_TA_BÜDÇP	: Çok Tesisli ve Tek Araçlı Bütünleşik Üretim ve Dağıtım Çizelgeleme Problemi
ÇT_ÇA_BÜDÇP	: Çok Tesisli ve Çok Araçlı Bütünleşik Üretim ve Dağıtım Çizelgeleme Problemi
DKA	: Deęişken Komşu Arama Algoritması
DKİ	: Deęişken Komşu İniş
GA	: Genetik Algoritma
GDKA	: Genel Deęişken Komşu Arama
GRASP	: Açgözlü Rassallaştırılmış Uyarlamalı Arama Algoritması
İDKA	: İndirgenmiş Deęişken Komşu Arama
KTDP	: Karma Tamsayılı Doğrusal Programlama
MA	: Memetik Algoritma
TA	: Tabu Arama
TDKA	: Temel Deęişken Komşu Arama
TDP	: Tamsayılı Doğrusal Programlama
YSD	: Yüzde Sapma Deęeri

1. GİRİŞ

Günümüzde firmalar birçok nedenden dolayı diğer firmalarla rekabet edebilmek için üretim ve dağıtım operasyonlarını birlikte yürütmek zorundadır. Üretim faaliyetleri, müşteri siparişlerinin tesis içerisindeki işlem sıraları ve bu işlemlerin çizelgelenmesi operasyonlarını, dağıtım faaliyetleri ise üretilen ürünlerin müşterilere teslim sürecini kapsamaktadır. Bu faaliyetleri birlikte ele alma zorunluluğunun bir diğer nedeni de zamana duyarlı siparişlerdir. Zamana duyarlı ürünler, üretimi tamamlandıktan sonra kısıtlı bir zaman diliminde müşteriye teslim edilmek zorundadır. Sistemde sınırlı kaynakların mevcut olması ve üretim ve dağıtım maliyetlerinin firmaların kârları üzerindeki etkisinin çok yüksek olması sebebi ile üretim ve dağıtım problemleri literatürde çok sık çalışılan problemlerdir. Dağıtım problemleri, maliyet açısından önemli olduğu kadar, ürün kalitesinin korunması ve müşteri tatmini için de oldukça önemlidir. Özellikle çabuk bozulabilir ürünler göz önüne alındığında ürün dağıtımının mümkün olduğunca hızlı gerçekleştirilmesi ürünün kalitesinin korunması ve hatta ürünün fiyatının belirlenmesinde de oldukça etkilidir.

Firmalar müşteri tatminini artırmak ve pazarda rekabeti sağlayabilmek amacıyla stok seviyelerini de mümkün olduğunca azaltmaya çalışmaktadırlar. Üretim ve dağıtım operasyonlarının bütünleşik gerçekleştirilmesi stokların azalmasına, dolayısıyla da firmaların bu zorunlulukları yerine getirebilmesine olanak sağlamaktadır. Bu nedenle, üretim ve dağıtım işlemlerini birlikte ele almak neredeyse kaçınılmaz hale gelmiştir. Son yıllarda pek çok işletme siparişe göre üretim (*Make-To-Order*) modeline yönelmeye başlamıştır. Bu modelde ürünler siparişe göre üretilmekte ve ürünler tesislerden müşterilere çok kısa sürelerde ulaştırılmaktadır. Sonuç olarak bitmiş ürün stoku oldukça düşük seviyelere çekilmekte, dolayısıyla da firma stok tutmanın maliyet, güvenlik, ürünün bozulması gibi birçok dezavantajından kurtulmaktadır. Bu süreci etkin bir şekilde yönetebilmek için de operasyonel seviyedeki üretim ve dağıtım çizelgelerinin bütünleşik bir şekilde ele alınması gerekmektedir.

Bozulabilir ürünlerin planlamasında ise üretim ve dağıtım operasyonlarının birlikte ele alınması daha fazla önem arz etmektedir. Zamana bağlı olarak kalitesi ve/veya değeri azalan ya da kullanılamaz hale gelen bu ürünlerin ne zaman ve ne kadar üretileceği, nasıl ve ne şekilde dağıtılacağı belirlenmesi oldukça önem kazanmıştır. Bozulabilir ürünler için bütünleşik bir yaklaşımın uygulama alanlarına örnek olarak gazete, gıda ürünleri,

hazır beton karışımları, nükleer ilaç ve endüstriyel yapıştırıcı malzemeleri üretim ve dağıtımını verilebilir.

Literatürde üretim planlama ve dağıtım planlamasının ayrı ayrı ele alındığı yaklaşım çok uzun yıllardır çalışılmasına rağmen bütünleşik üretim ve dağıtım çizelgeleme problemi ise son yıllarda büyük ilgi görmektedir. Bu çalışmaların çok büyük bir kısmında ise bozulabilir ürünler göz önüne alınmamıştır. Oysa yukarıda örneği verilen bu tarz bozulabilir ürünlerin üretim ve dağıtımının planlaması, yaşanacak gecikmelerde ürünlerin kısmen veya tamamen değerlerini yitirmelerine yol açacağından çok daha önemlidir.

Yukarıdaki örnekler göz önüne alındığında üretim ve dağıtım planlarının bütünleşik bir şekilde gerçekleştirilmesi birçok durum için kaçınılmaz derecede önemli hale gelmiştir. Literatürde Bütünleşik Üretim ve Dağıtım Çizelgeleme (BÜDÇ - Integrated Production and Outbound Distribution Scheduling) Problemi olarak adlandırılan bu problem bünyesinde makine çizelgeleme ve araç rotalama problemlerini barındırmaktadır. Makine çizelgeleme ve araç rotalama problemleri uzun yıllardır çok yaygın olarak çalışılmasına rağmen BÜDÇ problemleri yoğun olarak son 20 yılda çalışılmıştır.

Tez kapsamında çok sayıda çeşidi olan BÜDÇ problemi üzerinde durulacaktır. Problem ele alınırken rotalama kararları da göz önüne alınmıştır. Rotalama kararlarının göz önüne alındığı bu problemde dağıtım aşamasında birden fazla müşterinin ziyaret edilmesine izin verilmektedir. Müşterilerin hangi sırada ziyaret edileceği kararı da içeren rotalama problemi, NP-zor yapıda olduğu bilinen gezgin satıcı probleminin özel bir durumunu oluşturduğu için bu problem de NP-zor sınıfındadır. Rotalama kararını barındıran bu problemler BÜDÇ problemleri içerisinde çözümü en zor olan problemlerdir. Tez kapsamında literatürde daha önce ele alınmamış olan BÜDÇ problemleri incelenmiştir.

Tez çalışmasında ilk olarak sistemde birden fazla tesisin bulunduğu ve müşterilere her tesiste bulunan tek araçla hizmet edilen Çok Tesisli ve Tek Araçlı Bütünleşik Üretim ve Dağıtım Çizelgeleme Problemi (ÇT_TA_BÜDÇ) ele alınmıştır. Belirli varsayımlar altında ÇT_TA_BÜDÇ probleminde amacımız müşteri siparişlerinin üretilip dağıtımının gerçekleştirileceği en kısa zaman içerisinde, hangi müşterinin hangi depodan hizmet göreceğini, hangi müşteri siparişlerinin hangi parti içerisinde olacağını, her bir partinin üretimine ne zaman başlanıp bitirileceğini ve her bir tur içerisinde müşterilerin hangi sırada ziyaret edileceğini tespit etmektir. Bu amaç doğrultusunda her müşteriye yalnızca

bir kere ziyaret gerçekleştirilmeli, araca yüklenen ürünlerin toplam alan, hacim, ağırlık vb. gereksinimi araç kapasitesini geçmemeli, her rota bir tesiste başlamalı ve aynı tesiste sonlanmalı ve bir partinin üretimi tamamlandıktan sonra o parti içerisinde bulunan tüm ürünler (siparişler) ilgili müşterilere B birimlik zaman içerisinde teslim edilmelidir. Problemin çözümü için öncelikle bir matematiksel model geliştirilmiştir. Hem makine çizelgeleme hem de araç rotalama problemi NP-zor problem sınıfında yer aldığı için BÜDÇ probleminin de NP-zor yapıda olduğu bilinmektedir. Bu nedenle büyük boyutlu problemler için daha kısa sürelerde optimal ya da optimale yakın çözümler elde edilebilmesi amacıyla Değişken Komşu Arama (DKA) Algoritması geliştirilmiştir.

Tez çalışmasında çalışılan bir diğer problem ise yine sistemde çok tesisin bulunduğu fakat tesislerde birden çok aracın mevcut olduğu Çok Tesisli ve Çok Araçlı Bütünleşik Üretim ve Dağıtım Çizelgeleme Problemi (ÇT_ÇA_BÜDÇ)'dir. Belirli varsayımlar altında bu problemde amacımız, müşteri siparişlerinin üretilip dağıtımının gerçekleştirileceği en az maliyet ile, hangi müşterinin hangi tesisten ve hangi araçla hizmet göreceğini, hangi müşteri siparişlerinin hangi parti içerisinde olacağını, her bir partinin üretimine ne zaman başlanıp bitirileceğini, sistemde her tesis için kaç araç kullanılacağını ve her bir tur içerisinde müşterilerin hangi sırada ziyaret edileceğini tespit etmektir. Bu amaç doğrultusunda her müşteriye yalnızca bir kere ziyaret gerçekleştirilmeli, araçlara yüklenen ürünlerin toplam alan, hacim, ağırlık vb. gereksinimi araç kapasitesini aşmamalı, her rota bir tesiste başlamalı ve aynı tesiste sonlanmalı, bir partinin üretimi tamamlandıktan sonra o parti içerisinde bulunan tüm ürünler ilgili müşterilere B birimlik zaman içerisinde teslim edilmeli, bir tesise atanan müşterilerin talepleri o tesisin kapasitesini geçmemeli ve tüm üretim ve dağıtım işlemleri planlama ufkundan önce bitirilmelidir. Problemin çözümü için öncelikle bir matematiksel model geliştirilmiştir. Büyük boyutlu problemler için ise daha kısa sürelerde optimal ya da optimale yakın çözümler elde edilebilmesi amacıyla Memetik Algoritma (MA) geliştirilmiştir.

Tez çalışmasının diğer bölümleri şu şekilde düzenlenmiştir. Çalışmanın ikinci bölümünde BÜDÇ problemi genel olarak ele alınmış ve tanımlanmıştır. BÜDÇ problemi ile ilgili literatürde daha önce çalışılmış olan problemler üretim ortamlarına göre sınıflandırılarak detaylı olarak incelenmiş daha sonra ise literatürün genel bir değerlendirilmesi yapılmıştır. Üçüncü ve dördüncü bölümde sırasıyla tez çalışması kapsamında incelenen ÇT_TA_BÜDÇ ve ÇT_ÇA_BÜDÇ problemleri ele alınmış ve ayrıntılı olarak açıklanmıştır. Bu bölümlerde öncelikle problem tanımları verilmiş, daha

sonra problemlerin çözümleri için geliştirilen matematiksel modeller ayrıntılı olarak sunulmuştur. Geliştirilen matematiksel modelin ve problemin daha iyi anlaşılabilmesi için açıklayıcı bir örnek verilmiştir. Sonraki kısımda her bir problemin çözümüne ilişkin önerilen sezgisel yöntemler açıklanmış ve bu yöntemlerin daha iyi anlaşılabilmesi için bir örnek üzerinde gösterimi sağlanmıştır. Son olarak geliştirilen yöntemlerin performanslarını değerlendirmek amacıyla karşılaştırmalı sonuçlar verilmiştir. Her iki problem ayrıntılı olarak ele alındıktan sonra son bölümde ise tez çalışması kapsamında elde edilen bulguların genel bir değerlendirmesi yapılarak, gelecekte yapılması planlanan çalışmalara ışık tutabilecek öneriler yer almaktadır.



2. BÜTÜNLEŞİK ÜRETİM VE DAĞITIM ÇİZELGELEME (BÜDÇ) PROBLEMİ VE LİTERATÜR TARAMASI

Bu bölümde BÜDÇ probleminin tanımı yapılarak, 3. ve 4. bölümde ayrıntılı olarak ele alınacak olan ÇT_TA_BÜDÇ ve ÇT_ÇA_BÜDÇ problemlerinin daha iyi anlaşılabilmesi için genel bilgiler verilecektir. Ayrıca BÜDÇ problemi ile ilgili literatürde daha önce çalışılan problemler de detaylı olarak bu bölümde incelenecektir.

2.1. Problem Tanımı

Günümüzde rekabetçi ortama uyum sağlamak isteyen şirketler için üretim ve dağıtım faaliyetlerini birlikte ele almak önemli bir problem haline gelmiştir. Üretim faaliyetleri, müşteri siparişlerinin tesis içerisindeki işlem sıraları ve bu işlemlerin çizelgelenmesi operasyonlarını, dağıtım faaliyetleri ise üretilen ürünlerin müşterilere teslim sürecini kapsamaktadır. Bu iki faaliyetin bütünleştirilmesi adına yapılan ilk çalışmalarda üretim ve dağıtım kararları, stratejik- taktik seviye kararlar olarak kabul edilerek bütünleştirilmiştir (Bilgen ve Ozkarahan (2004); Erengüç ve ark. (1999); Sarmiento ve Nagi (1999); Chen (2004)). Bu çalışmalarda, üretim ve dağıtım aşamaları genelde bir ara stok vasıtasıyla birbirine bağlanmaktadır. Bu nedenle sözü edilen çalışmaların neredeyse hepsinde üretim ve dağıtım kararlarının yanında stok kararları da bulunmaktadır. Yapılan bu çalışmalarla üretim ve dağıtım planlarının bütünleştirilmesinin (eş zamanlı karar verilmesi), maliyetleri önemli ölçüde azalttığı ve pek çok durumda müşterilere sunulan hizmet kalitesini iyileştirdiği görülmüştür.

Tüm bu çalışmalara karşın, detaylı çizelgeleme seviyesinde bu iki kararın bütünleştirilmesi çok yeni bir kavram olup bu alanda yapılmış olan çalışmalar yoğun olarak son 20 yıl içerisinde gerçekleşmiştir. Bu çalışmalarda önerilen çözüm yöntemlerinde, her sipariş için gelir, maliyet ve müşteri hizmet seviyeleri göz önünde bulundurularak detaylı üretim ve dağıtım çizelgeleri belirlenmeye çalışılmıştır. Pratikte bütünleşik kararlar ardışık bir yol izlenerek verilmektedir. Öncelikle operasyon biriminde üretim çizelgeleri oluşturulmakta ardından bitmiş ürünler için lojistik departmanı tarafından üretim çizelgesine bağlı kalınarak bir dağıtım çizelgesi oluşturulmaktadır. Bu tür sıralı bir yaklaşımın etkin sonuçlar üretmediği, her iki kararın bütünleşik verilmesi ile daha iyi sonuçlara ulaşılabileceği literatürde değişik çalışmalar tarafından gösterilmiştir (Chen ve Vairaktarakis (2005); Pundoor ve Chen (2005)).

Günümüzde, global pazardaki rekabet ve müşterilerin artan beklentileri, firmaları stok seviyelerini azaltmaya zorlamaktadır. Üretim ve dağıtım operasyonlarının bütünleşik gerçekleştirilmesi stokların azalmasına, dolayısıyla da firmaların bu zorunlulukları yerine getirebilmesine olanak sağlamaktadır. Örneğin, son yıllarda pek çok işletme siparişe göre üretim (*Make-To-Order*) modeline yönelmeye başlamıştır. Bu modelde ürünler siparişe göre üretilmekte ve teslimat fabrikadan müşterilere çok kısa sürelerde ulaştırılmaktadır. Sonuç olarak bitmiş ürün stoku oldukça düşük seviyelerde tutulmakta, dolayısıyla da firma stok tutmanın maliyet, güvenlik, ürünün bozulması gibi birçok dezavantajından kurtulmaktadır. Bu süreci etkin bir şekilde yönetebilmek için de operasyonel seviyedeki üretim ve dağıtım çizelgelerinin bütünleşik bir şekilde ele alınması gerekmektedir.

Çizelgelerin bütünleşik olarak ele alınmasını gerektirecek bir diğer durum ise zamana duyarlı ürünlerin üretimi ve dağıtımıdır. Kolay bozulan hazır beton karışımları (García ve Lozano (2004); (2005)) endüstriyel yapıştırıcı materyaller (Devapriya ve ark. (2006); Armstrong ve ark. (2008); Geismar ve ark. (2008)) bu tarz ürünlere örnek olarak verilebilir. Bu ürünlerin yaşam süreleri (raf ömrü) çok kısadır. Beton karışımı için gereken hammaddeler (su, kum, çimento, çakıl vb.) karıştırıldıktan sonra kısa bir süre sonunda katılaşıp ve kullanılamaz hale gelir. Benzer şekilde bazı yapıştırıcı materyaller yapıştırıcı özelliğini belirli bir süre sonunda kaybeder. Bu nedenle bu tarz ürünler üretildikten hemen sonra tüketicilere ulaştırılmak zorundadır. Bu durumda üretim ve dağıtım aşamaları arasında ya hiç ya da yok denecek kadar az bir süre bulunmak zorundadır. Bir diğer zamana duyarlı ürün grubu ise gazetelerin basılması ve dağıtılmasıdır (Van Buer ve ark. (1999)). Günlük olarak üretilen ve dağıtılması gereken bu gazetelerin gece yarısından sonra üretimine başlanarak en kısa sürede araçlara yüklenip gazete bayilerine teslim edilmek üzere yola çıkmak zorundadır.

Yukarıdaki örnekler göz önüne alındığında üretim ve dağıtım planlarının bütünleşik bir şekilde gerçekleştirilmesi birçok durum için kaçınılmaz derecede önemli hale gelmiştir. Literatürde BÜDÇ Problemi olarak adlandırılan bu problem bünyesinde makine çizelgeleme ve araç rotalama problemlerini barındırmaktadır. Makine çizelgeleme ve araç rotalama problemleri uzun yıllardır çok yaygın olarak çalışılmasına rağmen BÜDÇ problemleri yoğun olarak son yıllarda çalışılmaktadır.

Chen (2010) yapmış olduğu çalışmada, makine çizelgeleme problemlerinden esinlenerek, BÜDÇ problemlerinin gösterimini standartlaştırmak amacıyla $(\alpha | \beta | \pi | \delta | \gamma)$ şeklindeki bir beşli gösterim önermiştir. Bu gösterimde α , β ve γ

gösterimi üretim çizelgeleme problemleri için kullanılan klasik gösterimle aynı olup aşağıdaki şekilde ifade edilmektedir;

α : üretim sistemlerindeki makine konfigürasyonunu,

β : siparişlerle ilgili kısıtları ve sınırlandırmaları,

γ : modelde kullanılan amaç fonksiyonunu,

π : teslimat süreciyle ilgili karakteristikleri ve

δ : müşteri sayısını temsil etmektedir.

Bu notasyonlar ayrıntılı olarak izleyen bölümde sırasıyla açıklanmıştır.

Makine konfigürasyonu (α):

Literatürde çalışılan pek çok BÜDC probleminde tek bir tesis incelenmiştir. Bu problemlerde kullanılan temel makine konfigürasyonları şu şekildedir:

1 : Tüm siparişlerin tek bir makinede işlendiği tek makineli üretim ortamı

P_m : m adet özdeş makinenin yer aldığı paralel makineli üretim ortamı (Her bir sipariş bu makinelerden yalnızca birinde işlenmektedir.)

F_m : m adet makinenin yer aldığı akış tipi üretim ortamı (Her bir sipariş sırasıyla tüm makineler tarafından işlenmektedir.)

B_m : Gruplandırma (bundling) tipi üretim ortamı (her bir sipariş m adet farklı operasyon içermekte, bu operasyonlar m adet makinede birbirinden bağımsız olarak gerçekleştirilmekte ve bu operasyonların hepsi tamamlandıktan sonra siparişin üretimi tamamlanmaktadır.)

$F(m_1, m_2)$: İki aşamalı esnek akış tipi üretim ortamı (Birinci aşamada m_1 adet paralel özdeş makine ve ikinci aşamada m_2 adet paralel özdeş makine bulunmak üzere her bir sipariş öncelikle birinci aşamadaki bir makinede ardından ikinci aşamadaki bir makinede işlenmektedir.)

Çok az sayıda çalışmada farklı lokasyonlarda bulunan birden fazla tesis incelenmiştir. Bu çalışmalarda her bir sipariş sadece tek bir tesiste üretilip, sadece aynı tesiste tamamlanan siparişler birlikte teslim edilebilmektedir. Çoklu tesisin bulunduğu bu problemleri ifade etmek için gösterimin bu bölümü CT şeklinde ifade edilmektedir.

Sipariş parametreleri ile ilgili kısıtlar (β):

r_j : Siparişler farklı serbest kalma zamanlarına sahiptir. (Bu kısıt olmadan tüm siparişlerin $t = 0$ anında işlenmeye hazır oldukları varsayımı yapılmaktadır.)

$d_j \equiv d$: Tüm siparişler aynı teslim tarihine sahiptirler.

\bar{d}_j : Her bir j siparişi \bar{d}_j ömrüne sahiptir ve siparişler bu süre tamamlanmadan önce müşterilere ulaştırılmak zorundadır.

$[a_j, b_j]$: Her bir j siparişi bir zaman penceresine sahiptir ve bu zaman aralığında müşteriye ulaştırılmak zorundadır.

fd_j : Her bir j siparişi sabit bir teslim süresine sahiptir ve tam olarak bu anda müşteriye ulaştırılmak zorundadır.

$s_f(s_{ij})$: Siparişlerin üretilebilmesi için ardışık siparişler arasında sıra bağımsız (bağımlı) bir hazırlık süresi gereklidir.

$prec$: Siparişler arasında öncelik ilişkileri bulunmaktadır.

$pmtn$: Siparişler işlenirken yarıda kesilebilir ve sonrasında işlenmeye devam edebilir.

$pickup$: Siparişler üretilmeden önce müşterilerden toplanmak zorundadır ve bitmiş ürünler tekrar aynı araçla ilgili müşterilere teslim edilmektedir.

$no - wait$: Akış tipi üretim ortamında bir siparişin bir makineden diğerine geçişi sırasında herhangi bir bekleme izin verilmemektedir.

$r - a (nr - a)$: Makinenin herhangi bir zamanda makine bakımı gibi sebeplerden dolayı kullanım dışı olma durumunu ifade etmektedir. Bu durumda siparişler sonrasında işlenmek üzere (işlenemeyecek şekilde) yarıda kesilmektedir.

$\sum D_j \leq D_0 (D_{max} \leq D_0)$: Toplam (en büyük) teslimat süresi önceden tanımlanan bir D_0 eşik değerini geçemez.

Teslimat süreciyle ilgili karakteristikler (π):

Literatürde yer alan çalışmaların büyük bir kısmında homojen araçlar incelenmiştir. Homojen araçlar için geçerli olan karakteristikler aşağıdaki şekildedir.

Araç karakteristikleri

$V(x, y)$: Her biri y kapasiteye sahip x adet araç kullanılmaktadır. Burada x ve y parametrelerinin değişik girişler aşağıdaki durumları ifade etmektedir.

$x = 1$: Ortamda tek bir araç vardır.

$x = v$: Ortamda sonlu sayıda v adet araç vardır.

$x = \infty$: Ortamda yeteri miktarda araç vardır.

$y = 1$: Her bir turda tek bir sipariş teslim edilebilir.

$y = c$: Her bir turda en fazla c adet sipariş teslim edilebilir.

$y = \infty$: Her bir turda herhangi bir sayıda sipariş teslim edilebilir, sınırlandırma yapılmamıştır.

$y = Q$: Her bir turda maksimum Q birim sipariş teslim edilebilir (Bu durumda her bir j müşterisinin q_j birimlik siparişi vardır).

Teslimat yöntemleri

iid : Bireysel ve hızlı teslimat (her bir sipariş tamamlandıktan hemen sonra ilgili müşteriye teslim edilir).

direct : Parti halinde direkt teslimat (sadece aynı müşteriye giden siparişler birlikte aynı turda taşınarak teslim edilir).

routing : Parti halinde rotalanarak teslimat (farklı müşterilere giden siparişler aynı turda taşınarak teslim edilir).

fdep : Sabit ayrılma zamanlı teslimat (her bir araç tesisten siparişleri teslim etmek üzere sabit zamanda ayrılmaktadır).

split : Bölünmüş talepli teslimat (sipariş bölünerek farklı partiler halinde taşınabilir).

Müşteri sayısı (δ):

Sistemdeki müşteri sayısına ilişkin üç olası durum vardır.

1 : Sistemde tek bir müşterinin olduğu durum

k : Her bir müşterinin farklı bir lokasyonda bulunduğu birden fazla müşterinin olduğu durum

n : Her siparişin farklı bir müşteriye ait olduğu durumdaki n adet müşterinin olduğu durum

Amaç Fonksiyonu (γ):

BÜDÇ probleminde amaç fonksiyonu;

- Müşteri hizmet düzeyi,

C_{max} : Maksimum tamamlanma zamanı

D_{max} : Maksimum teslim zamanı

$\sum D_j$: Toplam teslim zamanı

L_{max} : Maksimum gecikme vs.

- Toplam maliyet,
 TC : Ulaşım maliyeti
 VC : Araç sayısına bağlı ulaşım maliyeti
 PC : Toplam üretim maliyeti
- Toplam gelir,
 R_j : Toplam gelir amaçlarının birinin veya birkaçının eniyilenmesini içermektedir.

Bu sınıflandırmalara göre tez kapsamında çalışılması planlanan problemlerden birincisinde birden çok tesisin ve her tesiste tek bir makine ve tek bir aracın bulunduğu durum göz önüne alınacaktır ve standart gösterimi $(1, \zeta T | V(1, Q), routing | n | D_{max})$ şeklindedir. İkinci problemde ise daha genel bir durum ele alınacak olup sistemde yine çok sayıda tesis mevcuttur. Her tesiste sınırlı sayıda homojen araç ve tek bir makinenin mevcut olduğu durum ele alınmıştır. Problemin standart gösterimi ise $(1, \zeta T | V(v, Q), routing | n | TC + VC + PC)$ şeklindedir.

2.2. Bütünleşik Üretim ve Dağıtım Çizelgeleme Problemi İçin Literatür Taraması

Bu bölümde BÜDÇ problemleri ile ilgili literatürde yapılan çalışmalara yer verilmiştir. Bugüne kadar yapılmış olan çalışmaların büyük çoğunluğunda müşteri siparişlerinin üretimi için tek bir tesis ve o tesiste de tek bir makine bulunmaktadır. Siparişler partiler halinde üretilmektedir. Tek makineli üretim ortamı olan çalışmalardan sonra ise en çok çalışma yapılan üretim ortamı ise paralel makineli üretim ortamıdır. Son yıllarda ise akış tipi üretim ortamının da çalışılmaya başlandığı görülmektedir. Literatürde bugüne kadar yapılmış olan çalışmalar tek makineli üretim ortamı, paralel makineli üretim ortamı ve diğer üretim ortamları olmak üzere 3 ana grupta toplanmıştır. Bu şekilde literatürdeki boşlukların daha net bir şekilde görülmesi planlanmıştır.

2.2.1. Tek makineli üretim ortamı için literatür taraması

Tek makineli üretim ortamında müşteri siparişlerinin üretilmesi için tek bir makine vardır ve siparişler bu makinede sırasıyla işlem görmektedir. Yapılan çalışmaların çoğunda bu siparişler bir parti halinde birleştirilmektedir. Tek makineli üretim ortamı için yapılan çalışmalar Çizelge 2.1'de özetlenmiştir.

Çizelge 2.1. Tek makineli üretim ortamı için yapılmış olan çalışmalar

	Üretim					Dağıtım							Amaç Fonksiyonu								
	Tek Tesis	Çok Tesis	Parti Üretimi	Üretim Maliyeti	Hazırlık Zamanı	Hazırlık Maliyeti	Tek Araç	Homojen Araçlar	Heterojen Araçlar	Sınırsız Sayıda Araç	Sınırlı Sayıda Araç	Çoklu Tur	Yükleme Zamanı	Boşaltma Zamanı	Maliyet	Toplam Kazanç	Servis	Talebin Karşlanması	Kullanılan Araç Sayısı	Toplam Mesafe	Çevre
Potts (1980)	*		*				*		*						*						
Hurter ve Van Buer (1996)	*		*		*		*			*		*	*	*	*						
Van Buer ve ark. (1999)	*		*		*		*			*	*	*			*						
Garcia ve ark. (2004)		*	*				*			*	*		*		*						
Chang ve Lee (2004)	*		*				*				*					*					
Li ve ark. (2005)	*		*				*				*					*					
Chen ve Vairaktarakis (2005)	*		*				*		*						*	*					
Naso ve ark. (2007)		*					*		*		*	*	*	*	*	*					
Geismar ve ark. (2008)	*		*				*				*					*					
Armstrong ve ark. (2008)	*						*			*						*					
Chen ve ark. (2009)	*		*	*			*			*			*		*						
Park ve Hong (2009)	*		*	*	*	*	*			*		*	*	*	*						
Li ve Ferrell (2011)	*		*					*		*	*				*						
Geismar ve ark. (2011)	*						*			*					*						
Condotta ve ark. (2013)	*		*				*			*						*					
Low ve ark. (2013)	*		*				*			*			*			*					
Low ve ark. (2014)	*		*					*	*				*	*							
Viergutz ve Knust (2014)	*						*										*				
Hajiaghahi-Kesteli ve ark.(2014)	*							*		*				*							

Çizelge 2.1. Tek makineli üretim ortamı için yapılmış olan çalışmalar (Devam)

	Üretim						Dağıtım						Amaç Fonksiyonu								
	Tek Tesis	Çok Tesis	Parti Üretimi	Üretim Maliyeti	Hazırlık Zamanı	Hazırlık Maliyeti	Tek Araç	Homojen Araçlar	Heterojen Araçlar	Sınırsız Sayıda Araç	Sınırlı Sayıda Araç	Çoklu Tur	Yükleme Zamanı	Boşaltma Zamanı	Maliyet	Toplam Kazanç	Servis	Talebin Karşlanması	Kullanılan Araç Sayısı	Toplam Zaman	Çevre
Zu ve ark. (2014)	*		*				*			*	*				*						
Li ve Ling (2015)	*		*				*			*	*				*						
Li ve ark. (2016)	*		*				*			*					*	*					
Kang ve ark. (2016)	*			*		*	*			*					*						
Jamili ve ark. (2016)	*		*				*		*						*	*					
Cheref ve ark. (2016)	*		*				*				*					*					
Low ve ark. (2017)	*		*					*	*				*		*						
Devapriya ve ark. (2017)	*		*				*		*		*				*						
Karaođlan ve Kesen (2017)	*		*				*				*					*					
Zou ve ark. (2018)	*		*				*			*						*					
Lacomme ve ark. (2018)	*						*			*					*		*				
Gharaei ve Jolai (2018)		*					*		*						*		*				
Wang ve ark. (2019)	*							*		*											*
Liu ve ark. (2020)	*						*			*						*					
Ganji ve ark. (2020)	*							*		*					*	*					
He ve ark. (2021)	*						*			*					*	*					
ÇT_TA_BÜDÇ Problemi		*	*				*				*									*	
ÇT_ÇA_BÜDÇ Problemi		*	*	*			*			*	*				*						

Literatürde BÜDÇ problemiyle ilgili olarak yapılan ilk çalışma Potts tarafından yapılmıştır (Potts, 1980). Bu çalışmada $(1|r_j|V(\infty, 1), iid|n|D_{max})$ şeklinde ifade edilen direkt teslimatın göz önünde bulundurulduğu birden fazla müşteriye sınırsız araçla teslimatın gerçekleştirildiği durum incelenmiştir. Çalışma literatürde yer alan direkt teslimat yönteminin incelendiği pek çok çalışmaya örnek teşkil etmiştir.

Bütünleşik çalışmaların ilklerinden olan, Hurter ve Van Buer (1996) tarafından yapılan çalışmada yazarlar bir basım evinde gazete üretim ve dağıtım problemini araştırmışlardır. Problem için matematiksel bir model önermişler ve arama algoritmalarına dayalı sezgisel bir çözüm yaklaşımı geliştirmişlerdir.

Van Buer ve ark. (1999), daha önceki çalışmalarına benzer, araçların birden fazla tur yapabildiği doğrusal olmayan bir problemi ele almışlardır. Hurter ve Van Buer (1996)'de kullanılan araç sayısının enküçüklenmesi amaçlanmışken, Van Buer ve ark. (1999)'da hem araç sayısı hem de kullanım maliyetlerinin enküçüklenmesine çalışılmıştır.

Garcia ve ark. (2004) çalışmalarında hazır beton üretimi ve dağıtımını için bütünleşik bir çalışma yapmışlardır. Çalışmada birden çok tesis mevcuttur. Her tesiste, aynı anda birden çok sipariş üretmek için yeterli kapasite vardır. Çalışmada zaman penceresi yoktur ancak her siparişin müşteriye teslim edilmesi gereken bir teslim tarihi vardır. Problemin çözümü için geliştirdikleri tamsayılı doğrusal programlama (TDP) modelinin amacı, dağıtım maliyetlerini hesaba katarak kârı maksimize edecek siparişleri seçmektir.

Chang ve Lee (2004) iki müşteri ve tek makineli üretim ortamı için BÜDÇ problemini ele almışlardır. Problem için üç farklı senaryo tanımlamışlardır. Çalışmalarında bu senaryolar için sezgisel bir yöntem geliştirmiş bunun yanı sıra inceledikleri bir senaryonun NP-zor yapısını kanıtlamışlardır.

Li ve ark. (2005) ise çalışmalarında ikiden fazla müşteriyle Chang ve Lee (2004)'deki problemin daha genel bir hali üzerinde çalışma yapmışlardır. Bir siparişin tek bir turda teslim edilmesi gerekir, ancak aynı müşterinin farklı siparişleri farklı turlarda teslim edilebilir. Her iki çalışmada da amaç, aracın tüm siparişleri teslim edip tesise geri dönmesi için gereken toplam süreyi ve müşterilerin siparişe ulaşma sürelerinin toplamını minimize etmektir.

Chen ve Vairaktarakis (2005), problemi hazır yemek (catering) ve bilgisayar endüstrisine uyarlamışlardır. Yazarlar hem müşteri hizmet düzeyini hem de dağıtım maliyetlerini optimize etmeyi amaçlarken; tek ve paralel makineli üretim ortamlarını incelemişlerdir. Her bir problem için kesin ve sezgisel bir yönteme başvuran yazarlar aynı zamanda problemi bütünlük olarak ele almanın ayrı ayrı ele almaktan daha etkin olduğunu kanıtlamışlardır.

Naso ve ark. (2007) yapmış oldukları çalışmada hazır karıştırılmış beton ürününü dikkate almışlardır. Sistemde birden fazla tesis mevcuttur ve her tesiste, ürünü işlemek ve doğrudan bir kamyonu yüklemek için tek bir yükleme platformu mevcuttur. Bazı tesisler homojen bir araç filosuna sahipken, bazıları ise diğer tesislerin filosuna ihtiyaç duymaktadır. Dış firmalardan (3. Parti Lojistik -3PL) da araç kiralanabilir. Ayrıca problemde zaman pencereleri de mevcuttur. Eğer bir araç yükleme ve boşaltma yapmak için beklemek zorunda kalırsa bir ceza maliyeti ortaya çıkar. Doğrusal olmayan problemin amacı, nakliye, yükleme ve boşaltma bekleme sürelerini, fason üretimi, kiralanmış araç sayılarını ve sürücüler için fazla mesai ile ilgili maliyetleri en aza indirmektir.

Geismar ve ark. (2008) sınırlı bir yaşam ömrüne sahip bir ürün için çalışmalar yapmışlardır. Geismar ve ark. tarafından formüle edilen bütünlük problemde, tesisin içindeki makine çizelgelemesi açık bir şekilde dikkate alınmamıştır. Odak noktası, müşteri siparişlerini üretim işlemlerine atamak ve her rotanın büyüklüğünü ve başlangıç saatini belirlemek üzerinedir. Ayrıca, hangi müşterinin hangi rotada ve hangi sırada servis alacağı belirlenmektedir. Amaç, minimum yayılım zamanının belirlenmesidir. Çalışmada hem en iyi çözüm için alt sınır geliştirilmiş hem de iki aşamalı bir sezgisel önerilmiştir.

Armstrong ve ark. (2008), sıfır stokla çalışılan bir sistemde üretim ve dağıtım süreçlerini incelemişlerdir. Çalışmada müşterilerin ziyaret sıralamasının sabit olduğu varsayılarak amaç karşılanan talep sayısının en büyüklenmesi olarak belirlenmiştir. Problemin çözümü için dal-sınır algoritmasının önerildiği çalışmada en iyi çözüm için alt sınırlar geliştirilmiştir.

Chen ve ark. (2009) tarafından yapılan çalışmada, belirli bir bozulma oranına sahip olan ürünler için doğrusal olmayan tamsayı bir model geliştirilmiştir. Bu çalışmada her müşterinin gevşek bir zaman penceresi vardır. Eğer araç müşteriye erken gelirse, teslim zamanına kadar beklemek zorunda kalır, geç teslimat için ise ceza maliyeti vardır.

Müşteri talebi stokastik olduğundan, belirtilen plan ne kadar üretim yapılacağı, üretime ne zaman başlanacağı ve ürünün fiyatı, üretimi, taşınması ve müşteri memnuniyeti ile ilgili tüm maliyetleri de göz önüne alarak tedarikçinin beklenen kârını maksimize edecek rotaların belirlenmesini kapsamaktadır.

Tek dönemlik stok ürünleri için bütünlük bir çalışma, Park ve Hong (2009) tarafından yapılmıştır. Tek bir üretim hattı, ürünün farklı modellerini de işlemektedir. Her model bir kez üretilir ve bu nedenle aynı ürün için müşteri siparişleri bir seri halinde gerçekleştirilir. Müşteriler sıkı ve gevşek teslim zamanlarına sahiptir. Gevşek teslim zamanının ihlali gecikme maliyeti ile cezalandırılırken, sıkı teslim tarihinin ihlaline ise izin verilmemektedir. Aynı ürünün bölünmüş teslimatlarına izin verilmez, ancak müşteriler birden fazla ürün sipariş ettiğinde, her ürünü farklı bir araçla teslim etmek mümkündür. Geliştirdikleri karma tamsayılı doğrusal programlama (KTDP) modelinin amacı, üretim, nakliye ve gecikme maliyetlerini en aza indirmektir.

Li ve Ferrell (2011), bozulabilir bir ürün için bütünlük bir çalışma yapmışlardır. Yapılan çalışmada kullanılan araçlar heterojendir. Kapasite ve maliyet açısından farklılık gösteren araçlarla dağıtımını incelemişlerdir.

Geismar ve ark. (2011) sıfır stokla çalışılan bir üretim dağıtım sisteminde “havuz noktası (pool point)” teslimatın avantajını vurgulayarak her havuz noktasının talebi birden fazla araçla karşılandıktan sonra her bir havuz noktası içerisinde yer alan bireysel müşterilerinin siparişlerinin de 3PL firmalarınca sağlandığı durumu incelemişlerdir. Yazarlar toplam üretim ve dağıtım maliyetlerinin en küçüklenmesini amaçlamışlardır.

Condotta ve ark. (2013) her bir işin kendine ait üretime başlama ve teslim zamanının olduğu ve bu işlerin tek bir makinede üretildiği durumu incelemişlerdir. Çalışmada en büyük gecikmenin en küçüklenmesi amaçlanırken karma tamsayılı programlama yaklaşımıyla problem için alt sınırlar geliştirilmiştir. Bunun yanı sıra tabu arama sezgiselinden de yararlanılmıştır.

Low ve ark. (2013); (2014) yapmış oldukları çalışmalarda çok ürünlü bir dağıtım merkezi için bütünlük bir problemi ele almışlardır. Eğer bir müşteri farklı ürünlerden sipariş ederse, bu ürünlerin hepsi birlikte gruplandırılarak teslim edilir. Low ve ark. (2013) tüm siparişleri müşterilere ulaştırma süresini en aza indirmek için doğrusal olmayan tamsayılı bir model önermişlerdir. Büyük boyutlu problemlerin çözümü için

genetik algoritmadan yararlanmışlardır. Low ve ark. (2014) yılındaki çalışmalarında amaç, sabit araç maliyetleri, taşıma maliyetleri ve zaman penceresi nedeniyle ortaya çıkan ceza maliyetlerini minimize etmektir.

Viergutz ve Knust (2014) endüstriyel kimyasallarla ilgili bir çalışma yapmışlardır. Armstrong ve ark. (2008)'nin ele aldıkları problemi genişleterek üretime başlarken meydana gelen gecikmeleri de göz önünde bulundurmuşlardır. Ürünün sınırlı bir kullanım ömrü mevcuttur ve sadece tek bir teslimat turu vardır. Bu nedenle, tüm talepler müşteri bölgelerindeki zaman pencereleri içinde karşılanamayabilir. Geliştirilen KTDP modelinin amacı, maksimum talebin karşılanmasıdır.

Hajiaghahi-Keshteli ve ark. (2014), problemi demiryolu endüstrisindeki çalışmalar için ele almışlardır. Problem üretim çizelgeleri ve dağıtım ataması kararlarını içermektedir. Amaç en düşük maliyetle müşteri hizmet düzeyinin en büyüklenmesidir. Problem için sezgisel ve metasezgisel yaklaşımlardan yararlanılmıştır. Bunun yanı sıra sezgisel yöntemlerde kullanılan parametrelerin belirlenmesi için Taguchi yöntemine başvurulmuştur.

Zu ve ark. (2014) ve Li ve Ling (2015), Li ve Ferrell (2011)'in yapmış olduğu çalışmaya topla dağıt araç rotalama problemini entegre ederek geliştirmişlerdir. Tedarikçiler, tesis ve müşterilerden oluşan üç aşamalı tedarik zinciri için bütünlük bir çalışma yapmışlardır. Toplama ve dağıtım işlemleri aynı turda yapılmalıdır. Bu problem, hammaddelerin tedarikçiden alınıp, ürünlerin müşterilere teslim edilmesi gereken topla dağıt araç rotalama problemi olarak düşünülebilir. Geliştirdikleri KTDP modelin amacı, toplam taşıma maliyetini en aza indirmektir.

Li ve ark. (2016) hem taşıma maliyetini hem de toplam müşteri bekleme süresini minimize etmek için iki amaçlı doğrusal olmayan tamsayılı bir model geliştirmişlerdir. Taşıma maliyeti, kullanılan her bir araç için oluşan sabit bir maliyet ve seyahat zamanına bağlı olarak değişen bir maliyetten oluşur. Toplam müşteri bekleme süresi, teslim sürelerinin toplamına eşittir.

Kang ve ark. (2016) çalışmalarında dışarıdan hizmet alınan firmaları incelemişlerdir. Problemi karma tamsayılı bir modelle formüle etmişler ve en iyi sonuçlara yakın sonuçlar elde etmek amacıyla genetik algoritmadan yararlanmışlardır.

Jamili ve ark. (2016) tarafından yapılan çalışmada iki amaçlı tamsayı doğrusal bütünlük bir model önerilmiştir. Hem dağıtım maliyetini hem de dağıtım sürelerinin ortalamasını en aza indirmek amaçlanmıştır. İki amaç fonksiyonu, karar vericinin tercihini temsil eden ağırlıklar kullanılarak tek bir amaç olarak birleştirilir. Ürünün üretime başlayabilmesi için belirlenen en erken başlama zamanları mevcuttur.

Cheref ve ark. (2016) belirsiz bir ortamda bütünlük bir problemi inceleyen ilk yazarlardır. Jamili ve ark. (2016)'nin çalışmasında olduğu gibi, ürünler için belirlenen en erken başlama zamanları mevcuttur. En erken başlama tarihleri, işlem süreleri, seyahat süreleri ve teslim tarihleri belirsizdir. Karma tamsayı doğrusal modelin amaç fonksiyonu, teslim sürelerindeki maksimum gecikmeyi en küçüklemektir.

Low ve ark. (2017), iki kademeli bir tedarik zincirinde heterojen araçlarla dağıtımın gerçekleştirildiği ve zaman pencerelerinin dikkate alındığı bir problem için genetik algoritmanın iki farklı varyantının performansını karşılaştırmışlardır.

Devapriya ve ark. (2017), Geismar ve ark. (2008)'nin çalışmasını genişletmişlerdir. Bu çalışmada, araç sayısı tek bir araç değil, bir karar değişkenidir. Her araç birden fazla tur yapabilir. İkinci bir fark ise, sonlu bir planlama ufku olmasıdır. KTDP modelinin amacı, sabit araç maliyetlerini ve değişken taşıma maliyetlerini en aza indirmektir.

Karaođlan ve Kesen (2017) yapmış oldukları çalışmada belirli bir ürün ömrü olan ürünlerin üretimi ve dağıtılması ile ilgilenmişlerdir. Çalışmada amaç siparişlerin üretim ve dağıtım için gereken zamanın en küçüklenmesidir. Problemin çözümü için dal-kesme algoritması geliştirilmiştir. Alt sınırların iyileştirilmesinde geçerli eşitsizlikler geliştirilirken üst sınırların iyileştirilmesi amacıyla tavlama benzetimine dayanan bir yerel arama stratejisi geliştirmişlerdir.

Zou ve ark. (2018) yaptıkları çalışmada tek makineli ve sınırlı sayıdaki araç için BÜDÇ problemini ele almışlardır. Problemin çözümü için bir Genetik Algoritma (GA) geliştirilmiştir. GA'nın performansı literatürde bu probleme benzerlik gösteren diğer çalışmalardaki sezgisel algoritmalarla karşılaştırılmıştır. Ayrıca geliştirilen GA iki aşamalı olarak çözüm bulan bir algoritma ile de karşılaştırılmıştır. Yapılan deneysel çalışmalar, önerilen genetik algoritmanın önerilen iki aşamalı algoritmadan ve ilgili problemleri inceleyen iki yayınlanmış algoritmadan daha yüksek kalitede çözümler sağlayabileceğini göstermektedir.

Lacomme ve ark. (2018) çalışmalarında sınırlı ürün ömrüne sahip bir ürün için BÜDÇ problemini ele almışlardır. Problemi çözmek için yerel arama algoritması ve Açgözlü Rassallaştırılmış Uyarlamalı Arama Algoritması (GRASP)'nı birlikte kullanmışlardır.

Gharaei ve Jolai (2018) çok tesisli BÜDÇ Problemini incelemişlerdir. Sistemde birden çok tesis bulunmaktadır. Parti üretimi mevcuttur. Problemin çözümü için bir matematiksel model önermişler ve büyük boyutlu problemlerin çözümü için ise arı algoritmasından yararlanmışlardır.

Wang ve ark. (2019a) tarafından yapılan çalışmada BÜDÇ problemleri ilk kez çevresel faktörlerle beraber ele alınmıştır. Havaya salınan CO2 emisyonunun enküçüklenmesi için tabu arama algoritması geliştirilmiştir. Ayrıca üretimler arasında makinelerin kapatılmasına izin verilerek elektrik tüketimi de incelenmiştir.

Liu ve ark. (2020) yapmış oldukları çalışmada siparişlerin teslim zamanlarının toplamını en küçüklemek amacıyla değişken komşu arama algoritması geliştirmişlerdir.

Dağıtımda kullanılan araçların heterojen olduğu bir problem Ganji ve ark. (2020), tarafından ele alınmıştır. Çok amaçlı olarak ele aldıkları bu problemde maliyet ve hizmet amaçlarının yanı sıra yakıt tüketimi ve havaya salınan CO2 emisyonunun dikkate alınmıştır. Çalışmada parçacık sürüsü optimizasyonu, karınca kolonisi optimizasyonu ve BSGA-II algoritmalarının performansı karşılaştırılmıştır.

He ve ark. (2021) üç boyutlu yazıcılar ile tam zamanında dağıtım sistemini bütünleştirerek dağıtım maliyetleri ile teslimat zamanları toplamını ağırlıklı toplam şeklinde yeni bir küme kapsama modeli olarak formüle etmişler ve problem için kesin bir yöntem olan dal-fiyat algoritmasına başvurmuşlardır.

Literatürde tek makineli üretim ortamı için yapılmış olan çalışmalar göz önüne alındığında birçoğunun tek tesisli olduğu açıkça görülmektedir. Ancak günlük hayatta özellikle bozulabilir ürünlerin mevcut olduğu sistemlerde ürünlerin hızlı bir şekilde üretilip dağıtımını gerçekleştirebilmek için birden çok tesise ihtiyaç duyulabilmektedir. Bu nedenle birden çok tesisin ve her tesiste dağıtımını gerçekleştirecek birden çok aracın bulunması pratikte daha uygun bir durumu temsil etmektedir. Tek makineli üretim ortamında, ürünlerin belirli bir raf ömrüne sahip olduğu, birden çok tesis ile müşterilere hizmetin gerçekleştirildiği, ürünlerin partiler halinde üretilip dağıtıldığı ve araçların çoklu

kullanımına izin verilirken rotalama kararlarının da alındığı bir çalışmaya literatürde rastlanmamıştır. Bu nedenle yapılan bu tez çalışmasının bu alandaki bir boşluğu dolduracağı düşünülmektedir.

2.2.2. Paralel makineli üretim ortamı için literatür taraması

BÜDÇ üzerindeki çalışmaların büyük çoğunluğu tek makineli üretim ortamını kapsarken, daha sonra en fazla çalışılan üretim ortamı da paralel makineli üretim ortamıdır. Bu çalışmaların çoğunda, özdeş paralel makineler (*identical, P_m*) kullanılmıştır. Diğer paralel makineli üretim ortamları ise özdeş olmayan paralel makinelerle (*uniform, Q_m*) ve birbirinden bağımsız (ilişkisiz) paralel makineler (*unrelated, R_m*)'dir. Tek bir makine ortamıyla yapılan çalışmalara benzer şekilde, çoğu çalışmalarda partiler halinde üretim yapılmakta ve hazırlık süreleri ihmal edilmektedir. Çizelge 2.2'de paralel makineli üretim ortamı için yapılmış çalışmalar listelenmiştir.

Tek makineli üretim ortamındaki çalışmalarının yanı sıra Chen ve Vairaktarakis (2005) paralel bir makine ortamını da göz önüne almışlardır. Tek makineli üretim ortamında olduğu gibi, dağıtım maliyetleri ile müşteri hizmet seviyesi arasındaki ödünleşim amaç fonksiyonu olarak kabul edilmiştir.

Hurter ve Van Buer (1996) ve Van Buer ve ark. (1999) tarafından tek makineli üretim ortamı için yapılan çalışmaya benzer bir çalışma, Russell ve ark. (2008) tarafından ele alınmıştır. Bu çalışmada iki adet paralel gazete üretim hattı mevcuttur. Gazetelerin basımının gece yarısından önce başlayamayacağı varsayılmaktadır, bu da üretimin en erken başlama tarihi olarak değerlendirilmiştir. Çözümde tabu arama algoritması kullanılmışlardır.

Chiang ve ark. (2009) Russell ve ark.'ın çalışmasına benzer bir çalışma yapmıştır. Ancak bu çalışmada ek bir gazete baskısı daha vardır. Ek baskının üretimi, diğer iki basımdan birinin üretilmesinden önce tamamlanmak zorundadır. Her iki çalışmada da zaman pencerelerinin ve bölge kısıtlamalarının (zoning constraints) göz önüne alındığı açık uçlu araç rotalama problemi için KTDP modeli önerilmiştir. Tek bir araçla teslim edilebilecek bölgelerin sayısında bir sınırlama vardır. Modelin genel amacı, seyahat edilen toplam mesafeyi enküçükleme iken, bölgeler için amaç araç sayısını minimize etmektir.

Çizelge 2.2. Paralel makineli üretim ortamı için yapılmış olan çalışmalar

	Üretim					Dağıtım							Amaç Fonksiyonu								
	Tek Tesis	Çok Tesis	Parti Üretimi	Üretim Maliyeti	Hazırlık Zamanı	Hazırlık Maliyeti	Tek Araç	Homojen Araçlar	Heterojen Araçlar	Sınırsız Sayıda Araç	Sınırlı Sayıda Araç	Çoklu Tur	Yükleme Zamanı	Boşaltma Zamanı	Maliyet	Toplam Kazanç	Servis	Talebin Karşlanması	Kullanılan Araç	Toplam Mesafe	Çevre
Chen ve Vairaktarakis (2005)	*		*				*		*					*		*					
Russel vd. (2008)	*		*					*		*		*	*					*	*		
Chiang vd. (2009)	*		*					*		*		*	*					*	*		
Farahani vd. (2012)	*		*		*	*	*			*				*							*
Ullrich (2013)	*							*		*	*	*	*			*					
Amorim vd. (2013)	*			*	*	*	*		*				*	*							
Lee vd. (2014)	*		*	*				*		*	*	*	*	*							
Chang vd. (2014)	*		*				*			*				*		*					
Belo – Filho vd. (2015)	*			*	*	*	*		*				*	*							
Zhong ve Jiang (2015)	*		*				*		*					*		*					
Fu vd. (2017)	*			*	*	*		*		*				*							
Kergosien vd. (2017)	*								*		*					*					
Tavares-Neto ve Nagano (2018)	*				*				*							*					
Kesen ve Bektaş (2019)	*						*			*						*					

Ullrich (2013)'in yapmış olduğu çalışmada, müşteri siparişleri özdeş paralel makinelerden birinde işlem görmektedir. Dağıtımını gerçekleştirecek olan araçlar heterojendir. Zaman penceresi dikkate alınmıştır. Erken teslimata izin verilmezken geç teslimata izin verilmektedir. KTDP modelinin amacı toplam gecikmenin en küçüklenmesidir. Problemin çözümü için GA tabanlı bir yöntem geliştirilmiştir.

Amorim ve ark. (2013), bazı ürünlerin bozulabileceği özdeş olmayan paralel makineler için BÜDÇ problemini incelemişlerdir. Bu çalışmanın literatüre asıl katkısı, parti büyüklüğü kararlarının ele alınmış olmasıdır. Yani bir müşteri siparişinin daha küçük parçalara bölünerek farklı makinelerde işlenmesinin, tamamının tek bir makinede işlenmesine kıyasla daha iyi sonuçlar çıkarıp çıkaramayacağına da değinmişlerdir. Hazırlık süreleri ve maliyetleri, sonuçları çok ciddi bir şekilde etkileyebileceğinden dikkate alınmıştır. KTDP modelinin amacı, toplam üretim, hazırlık ve dağıtım maliyetlerini en aza indirmektir.

Nükleer ilaç için bir çalışma Lee ve ark. (2014) tarafından yapılmıştır. Tıbbi tedavi amacıyla kullanılan radyoaktif maddelerin üretim ve dağıtımını ile ilgili bir çalışma yapmışlardır. Birden fazla sipariş, makine kapasitesi ihlal edilmediği sürece, eşzamanlı olarak üretilebilir. Problemden sıkı zaman pencereleri mevcuttur. Zaman penceresinin üst sınırı, ilacın kullanım süresidir. Geliştirilen KTDP modelinin amacı, üretim ve dağıtım maliyetini en aza indirmektir. Farklı komşuluk algoritmalarına da başvurmuşlardır.

Chang ve ark. (2014)'in çalışmalarında, farklı müşterilere ait farklı siparişlerin, özdeş olmayan paralel makinelerde işlenmesi gerekir. Aynı araç tarafından teslim edilecek olan tüm müşteri siparişleri, sırasıyla üretilmektedir. Doğrusal olmayan matematiksel modelin amacı, teslimat sürelerinin ve toplam dağıtım maliyetlerinin minimize edilmesidir.

Belo-Filho ve ark. (2015), Amorim ve ark. (2013) tarafından geliştirilen modeli parti büyüklüğü yaklaşımını kullanarak ele almışlardır. Karma tamsayı doğrusal model geliştirilmiş ve büyük boyutlu problemlerin çözümü için uyarlanabilir geniş komşuluk arama stratejisinden yararlanmışlardır.

Zhong ve Jiang (2015) yapmış oldukları çalışmalarında özdeş paralel makineli üretim ortamında BÜDÇ problemini ele almışlardır. 2 müşterinin ve birden fazla siparişin bulunduğu durum için sezgisel algoritma geliştirmişlerdir. Amaç fonksiyonu toplam maliyet enküçüklemesi ve maksimum servis seviyesidir.

Fu ve ark. (2017) işlerin bölünebildiği ve özdeş olmayan paralel makinelerin kullanıldığı bir metal paketleme fabrikasında çalışma yapmışlardır. Yapılan çalışmada hem üretim hem de dağıtım için teslim zamanları mevcuttur. Ayrıca işlerin üretim sırasına göre de hazırlık süresi ve hazırlık maliyeti de sisteme dahil edilmiştir. Üretilen siparişler heterojen araçlarla belirli bir zaman penceresine göre teslim edilmektedir. Amaç toplam hazırlık ve dağıtım maliyetlerinin enküçüklenmesidir. Yazarlar problemin çözümü için iki aşamalı iteratif bir sezgisel geliştirmişlerdir.

Kergosien ve ark. (2017) ilaç sektöründe bir uygulama yaparak kemoterapi ilaçlarının üretim ve dağıtım problemini ele almışlardır. Çalışmada teknisyenler özdeş paralel makineler, kemoterapi hazırlama süreci de iş olarak kabul edilmiştir. Sistemde dağıtım işlemi için tek bir araç ve personel bulunmaktadır. Bu çalışmada önerilen çözüm yöntemi, uygun çözümler ve alt sınırların bulunmasını sağlayan “Benders Ayırıştırma” temelli bir sezgiseldir. Bu yaklaşımın avantajı, problemin iki aşamalı olarak ayrıştırılabilen yapısının kolaylıkla kullanabilmesidir.

Tavares-Neto ve Nagano (2018) çalışmalarında özdeş paralel makineli BÜDÇ problemini ele almışlardır. Problemde ayrıca sıraya bağlı hazırlık zamanı da dikkate alınmaktadır. Sistemde sınırsız stok alanı ve tek bir araç mevcuttur. Küçük boyutlu problemler için KTDP modeli geliştirmişler. Daha büyük boyutlu problemlerin çözümü içinse GA, ekle ve sırala algoritması (insert-and-order) ve yinelemeli açgözlü sezgiseli üzerine çalışmalar yapmışlardır. Yapılan deneysel çalışmalar, algoritmaların, iyi bir sonuç elde edebildiğini, ancak, makinelerin sayısının, her algoritmanın performansını farklı şekilde etkilediğini göstermiştir. Problemin amacı yayılım zamanının enküçüklenmesidir.

Kesen ve Bektaş (2019) özdeş paralel makineli üretim ortamı ve sınırlı sayıda homojen aracın yer aldığı dağıtım ortamını ele almışlardır. Problemin çözümünde müşterilerin zaman pencerelerini dikkate alarak müşterilere toplam erken varma ve gecikme zamanlarını enküçükmeye çalışmışlardır.

Çizelge 2.2’de görüldüğü üzere paralel makineli üretim ortamları ile ilgili yapılan çalışmaların tamamı sistemde tek tesisin mevcut olduğu durum için ele alınmıştır. Tüm çalışmalarda tesiste birden çok araç olduğu varsayılmaktadır ve bu araçlar birçok çalışmada homojen ve sınırlı sayıdadır.

2.2.3. Diğer üretim ortamları için literatür taraması

BÜDÇ problemleri ile ilgili olarak tek makineli üretim ortamı ve paralel makineli üretim ortamının yanı sıra, dört farklı üretim ortamı ile ilgili de çalışmalar yapılmıştır. Bunlar; gruplandırma tipi (bundling), akış tipi, iki aşamalı esnek akış tipi ve atölye tipi üretim ortamlarıdır. Çizelge 2.3'te bu konularla ilgili yapılan çalışmalar listelenmiştir.

Li ve Vairaktarakis (2007)'in çalışması makine konfigürasyonu açısından yapılan diğer çalışmalardan farklılık göstermektedir. Yazarlar bir müşteri siparişinin iki görevinin de özel bir makinede işlenmesi gereken bütünleşik bir problemi ele almışlardır. İki görev birbirinden bağımsızdır ve gerektiğinde aynı anda paralel olarak işlem görebilir. Her iki görev de tamamlandığında teslimat başlayabilir. Bu tür üretim sistemleri gruplandırılmalı üretim olarak adlandırılabilir. Aynı araç ile teslim edilecek olan müşteri siparişleri ardışık olarak üretilmektedir. Amaç dağıtım maliyetlerinin ve müşterilerin bekleme maliyetlerinin en küçüklenmesidir. Yazarlar problem için polinom zamanlı bir sezgisel algoritma önermişlerdir.

Daha önceki çalışmalar tek bir üretim seviyesini dikkate alırken, Scholz-Reiter ve ark. (2011)'in yapmış olduğu çalışma akış tipi üretim ortamı ile araç rotalama probleminin bütünleşik bir şekilde ele alındığı ilk çalışma olmuştur. Bu çalışmada bir üretim ve dağıtım çizelgesi tanımlanmasına ihtiyaç vardır. Scholz-Reiter ve ark. tarafından gerçekleştirilen bu çalışma, stokun ilk üretim seviyesinden önce, ardışık üretim seviyeleri arasında ve bir teslimattan hemen önce oluşturulabileceğini ve maliyetlerini de hesaba katan literatürdeki bilinen ilk çalışmadır. Çalışmada bir müşterinin siparişinin bölünemeyeceği varsayılmıştır ancak aynı müşterinin farklı siparişleri farklı rotalarda teslim edilebilir. Bir planlama ufku mevcuttur ve stokastik olaylar planlamayı etkileyebilir. Her siparişin teslim edileceği zamanlar mevcuttur erken teslimata izin verilmezken, geç teslimatlar için ceza maliyeti oluşmaktadır. Geliştirilen KTDP modelinin amacı, üretim, stok, ceza ve taşıma maliyetleri dahil toplam maliyeti en aza indirmektir.

Çizelge 2.3. Diğer üretim ortamları için yapılmış olan çalışmalar

	Üretim						Dağıtım							Amaç Fonksiyonu							
	Tek Tesis	Çok Tesis	Parti Üretimi	Üretim Maliyeti	Hazırlık Zamanı	Hazırlık Maliyeti	Tek Araç	Homojen Araçlar	Heterojen Araçlar	Sınırsız Sayıda Araç	Sınırlı Sayıda Araç	Çoklu Tur	Yükleme Zamanı	Boşaltma Zamanı	Maliyet	Toplam Kazanç	Servis	Talebin Karşlanması	Kullanılan Araç	Toplam Mesafe	Çevre
Gruplandırma Tipi Üretim Ortamı																					
Li ve Vairaktarakis (2007)	*		*	*				*		*				*		*					
Akış Tipi Üretim Ortamı																					
Scholz - Reiter vd. (2011)	*			*				*		*	*			*							
Rohmers ve Billaut (2015)	*			*			*		*		*			*							
Marandi ve Zegordi (2017)	*							*		*				*							
Ramezani ve ark. (2017)	*							*		*				*							
Wang ve ark. (2019)	*						*				*					*					
Yağmur ve Kesen (2020)	*						*				*										
Yağmur ve Kesen (2021)	*							*		*	*					*					
İki Aşamalı Esnek Akış Tipi Üretim Ortamı																					
Yan vd. (2018)	*				*		*		*					*							
Atölye Tipi Üretim Ortamı																					
Meinecke ve Scholz-Reiter (2014)	*			*		*	*			*				*							
Mohammadi ve ark. (2020)	*							*		*						*					
Yağmur ve Kesen (2022)	*							*		*	*					*					*

Rohmer ve Billaut (2015) yapmış oldukları çalışmalarında akış tipi üretim ortamını ve aynı zamanda dağıtımın başka bir şirket (3PL) tarafından gerçekleştirildiği durumu ele almışlardır. Modelin ayrıca daha gerçekçi olması için stok maliyetleri de dikkate alınmıştır. İki tür stok göz önünde bulundurulmuştur: yarı ürün ve ürün stoku. Ayrıca stok maliyeti, gecikmelerden doğan ceza maliyeti ve taşıma maliyetleri de göz önüne alınmıştır. Müşteriler için belirli teslim zamanları mevcuttur. Problemi çözmek için KTDP modelinden faydalanmış ve problemin amacını da maliyet enküçüklemesi olarak belirlemişlerdir.

Meinecke ve Scholz-Reiter (2014) tarafından gerçekleştirilen çalışma, atölye tipi üretim ortamının kullanıldığı ilk çalışmadır. Yazarlar çalışmalarında stoku, sadece üretim ve dağıtım işlevi arasında bir bağlantı elemanı olarak kullanmışlardır. Her müşteri siparişi, her bir üretim seviyesinde bulunan makinelerin birinde işlenmelidir. Geliştirilen tamsayı doğrusal modelde amaç, üretim, hazırlık, dağıtım, stok, üretim ve teslim tarihlerinin ihlali ile ilgili maliyetleri en aza indirmektir.

Marandi ve Zegordi (2017) bozulabilir gıda ürünleri için üretim ve dağıtım süreçlerini ele almışlardır. Üretim ve dağıtım arasında geçen sürenin azaltılarak ürün kalitesinin artırılmasını amaçlamışlar ve problemin çözümü için parçacık sürüsü optimizasyonundan yararlanmışlardır.

Akış tipi üretim ortamıyla ilgili yapılan çalışmalardan birisi de Ramezani ve ark. (2017) tarafından gerçekleştirilmiştir. Dağıtım sürecinde direkt teslimat ve rotalama yapılarak teslimat olmak üzere iki farklı yöntemi karşılaştırmışlardır.

Yan ve ark. (2018) iki aşamalı yarı-sürekli bir akış tipi üretim ortamında çalışma yapmışlardır. Bu atölye tipinde hammaddeler ilk aşamada sürekli kaynaklara dönüştürülür ve daha sonra ikinci aşamada, son ürüne dönüştürülür. Ayrıca, ilk aşamadaki işlem sırası, hazırlık sürelerini ve kaynak maliyetlerini belirler. Cam kapların seri üretimi bu üretim ortamı için uygun bir örnektir (Almada-Lobo ve ark., 2008). İlk aşamada, kum, soda külü, kireçtaşı ve kırık cam gibi hammadde karışımı, sürekli kaynak olan yaklaşık 1500°C'lik cam macununun içine karıştırılır. Elde edilen karışım daha sonra ikinci aşamada nihai ürün haline getirileceği üretim hatlarına gönderilir. Başka bir örnek de iplik endüstrisidir. Lifler (farklı elyaf karışımları) ilk üretim aşamasında işlenir, daha sonra farklı iplik türlerini üretmek için paralel makinelere gönderilir (Camargo ve ark., 2014). Bu çalışmada, üretim ve dağıtım planlamasının bir aylık planlama ufku için

gerçekleştirildiği bir model araştırılmaktadır. Üretim planlama modeli ve dağıtım modeli ayrı ayrı oluşturulmuş ve iki model stok parametreleri ile birleştirilmiştir. Tabu arama ve parçacık sürü optimizasyonu algoritmasına dayalı iki aşamalı bir sezgisel algoritma geliştirilmiştir.

Wang ve ark. (2019b), üç aşamalı hibrit akış tipi çizelgeleme problemini ele almışlardır. Problemin çözümünde amaç en büyük dağıtım tamamlanma zamanının enküçüklenmesidir. Hibrit akış tipi üretimin ilk aşamasında sıra bağımlı hazırlık süresine sahip özdeş paralel makineler yer alırken ikinci aşamada hazırlık süresi olmayan özdeş paralel makineler yer almaktadır. Son aşamada ise siparişe özel makineler bulunmaktadır. Yazarlar problem için öncelikle karma tamsayılı bir programlama yaklaşımı geliştirmişler ardından değişken komşuluk arama stratejisine dayalı üç farklı sezgisel yöntem önermişlerdir.

Mohammadi ve ark. (2020) mobilya imalatı yapan bir firmada esnek atölye tipi üretim ortamında BÜDÇ problemini ele almışlardır. Problemden amaç üretim ve dağıtım maliyetlerinin toplamı ile teslim tarihlerine göre erken gelme ve gecikme zamanlarının ağırlıklı toplamının en küçükleşmesidir. Problemin çözümü için hibrit parçacık sürüsü optimizasyonu geliştirilmiştir.

Yağmur ve Kesen (2020), permütasyon akış tipi üretim ortamı ile sistemde tek bir aracın bulunduğu dağıtım ortamında toplam gecikme ile toplam tur zamanı toplamını en küçükleme için karma tam sayılı bir matematiksel model geliştirmişler ve büyük boyutlu problemlerin çözümü için ise MA yaklaşımından faydalanmışlardır.

Yağmur ve Kesen (2021) sınırlı sayıdaki heterojen araçlar kullanılarak permütasyon akış tipi üretim ortamını ele almışlardır. Problemin amacı, araçların kat ettiği toplam mesafeyi ve teslimatların gecikmesinden dolayı kaynaklanabilecek gecikme miktarını en küçükleme. Karma tam sayılı matematiksel modelin yanı sıra, büyük boyutlu problemlerin çözümü için ise memetik algoritma ve tabu arama algoritması geliştirmişlerdir.

Atölye tipi üretim ortamı için yapılan çalışmalardan bir diğeri de Yağmur ve Kesen (2022) tarafından gerçekleştirilmiştir. Her aracın birden fazla kullanılmasına izin verilen, sistemde heterojen ve sınırlı sayıda aracın bulunduğu atölye tipi üretim ortamını ele almışlardır. Problemden araçlar tarafından yayılan toplam CO₂ miktarının en aza

indirilmesi ve geç teslimatlardan kaynaklanan azami gecikmenin en aza indirilmesi olmak üzere birbiriyle çelişen iki farklı hedef göz önünde bulundurulmaktadır.

Ele alınan üretim sistemlerinin özelliklerine daha yakından bakıldığında, önceki çalışmaların, genel olarak her siparişin tek bir işlemden oluştuğu nispeten basit üretim ortamları olduğu görülmektedir. Üretim maliyetleri çok az sayıdaki çalışmaya dahil edilmiştir. Çoğu çalışmada parti üretimi yapılmaktadır. Parti üretimi ile ilgili, ardışık partiler arasında hazırlık işlemleri vardır. Ancak hazırlık işlemlerinin genellikle ihmal edildiği görülmektedir. Üretim teslim tarihi sadece paralel makineli üretim ortamı ve atölye tipi üretim ortamındaki çalışmalarda dikkate alınmıştır. Probleme ürünlerin erken başlama tarihinin eklenmesi, planlama ufkunun başlangıcında tüm emirler mevcut olmadığından problemi daha gerçekçi hale getirir. Fakat bu durum problemi daha da karmaşık hale getirecektir.

Tek ve paralel makineli üretim ortamı için dikkat çeken bir diğer durum ise bu konuda yapılmış olan çalışmalarda stokların ve stok tutma maliyetlerinin ihmal edilmesidir. Yalnızca birkaç çalışmadaki modellerde üretim ve dağıtım arasındaki stoklar ve ilgili stok tutma maliyetleri açık bir şekilde hesaba katılmıştır. Seyahat süreleri ise tüm çalışmalara dahil edilmiştir. Ayrıca, çalışmaların çoğunda, değişken ve sabit taşıma maliyetleri de göz önüne alınmıştır. Taşıma maliyetlerini dikkate almayan çalışmaların hepsinin bir hizmet amacı vardır. Bazı araştırmacılar hizmet sürelerini ayrıca ele alırken, bazıları ise bu süreyi seyahat sürelerine dahil eder. Tek makineli üretim ortamında, parti üretimi yapılmayan hemen hemen tüm çalışmalarda zaman pencereleri mevcut iken, parti üretimi yapılan çok az çalışma zaman pencerelerini içerir. Paralel makineli üretim ortamında ise parti üretimi yapılmayan, tüm çalışmalar zaman pencerelerini hesaba katmaktadır.

Üretim ve dağıtım fonksiyonları aynı anda çözüldüğünde, problem yapısının karmaşıklığı artar. Bütünleşik bir planlama probleminin formülasyonu çok sayıda değişken ve kısıt içermektedir. BÜDÇ probleminin bu karmaşıklığı nedeniyle, kesin yöntemler yalnızca basit tek makineli ortamdaki çalışmalar için uygulanmıştır. Ayrıca, tek makineli ortamla ilgili yapılan çalışmalarda sezgisel ve metasezgisel yöntemler birlikte kullanılmıştır. Paralel makineli üretim ortamı ile ilgili yapılan tüm çalışmalarda ise, Tabu Arama (TA), Adaptif geniş komşu arama, GA ve Karınca kolonisi optimizasyonu gibi bir sezgisel veya metasezgisel yöntemler kullanılmıştır. Genel olarak, geliştirilen sezgisel ve metasezgisellerin performansını değerlendirmek için en fazla

sistemde en fazla 100 müşterinin olduğu örnekler kullanılmıştır. Birkaç çalışmada ise, 200 müşteriye kadar olan test problemleri kullanılmıştır. Ayrıca, her iki çözüm yaklaşımının sonuçlarını karşılaştırmak için CPLEX ve LINGO gibi çözücüler kullanılmıştır. Çözücülerle, genellikle küçük boyutlu problemler için optimal sonuçlar elde edilmiştir. Yazarlar, 21 müşteriye kadar optimal çözümü bulan Park ve Hong (2009) hariç, en fazla 7 müşteriye kadar olan örnekler için optimal çözüme ulaşabilmişlerdir. Ayrıca, tek makineli ortamda, Karaoğlan ve Kesen (2017), dal kesme algoritması kullanarak 50 müşteriye kadar olan test problemlerini çözebilmişlerdir (Moons ve ark., 2017). Bütünleştirmede, ayrı ayrı çözüm yaklaşımına kıyasla %5 ile %20 arasında bir ortalama iyileşmeye yol açabilir, ancak %40'a varan oranlarda iyileştirmeler de elde edilebilir.

Tez kapsamında çalışılan ÇT_TA_BÜDÇ ve ÇT_ÇA_BÜDÇ problemleri literatürde az çalışılan çok tesisli ve tek makineli üretim ortamını ele almaktadır. Tek makineli üretim ortamı ile ilgili yapılan birçok çalışma mevcut olmasına rağmen, sistemde çok tesisin bulunduğu problemler oldukça azdır. Dağıtım süreçlerinde ise gerek tek araçlı gerekse de homojen çok araçlı her iki durum için de çoklu tur gerçekleştirilen çalışma oldukça az sayıdadır. Çoklu tur kullanımı problemin çözümünü daha zor hale getirirse de günlük hayata daha uygundur. Ayrıca üretilen ürünlerin belirli bir kullanım ömrüne sahip olması da yine günlük hayata uygun ve az çalışılmış problem türlerindedir. Tüm bu açılardan değerlendirildiğinde tez kapsamında çalışılan her iki problemin de literatüre önemli katkıları olacağı düşünülmektedir.

3. ÇOK TESİSLİ VE TEK ARAÇLI BÜTÜNLEŞİK ÜRETİM VE DAĞITIM ÇİZELGELEME (ÇT_TA_BÜDÇ) PROBLEMİ

Bu bölümde çok tesisli ve tek araçlı bütünleşik üretim ve dağıtım çizelgeleme probleminin tanımı yapılarak, problemin çözümü için geliştirilen matematiksel model ve metasezgisel yöntem anlatılmıştır. Ayrıca problemin ve çözüm yöntemlerinin daha iyi anlaşılabilmesi için her iki çözüm yöntemi açıklayıcı örnekler üzerinden detaylandırılmıştır.

3.1. Problem Tanımı ve Açıklayıcı Örnek

ÇT_TA_BÜDÇ probleminde sistemde sabit bir üretim hızı olan birden çok homojen tesis bulunmaktadır. Her tesiste tek bir makine vardır ve bu makine her tesis için aynıdır. Üretilen ürünlerin dağıtımının partiler halinde yapılacağı varsayılmıştır. Bir partide birden çok müşterinin siparişi bulunabilir yani aynı rotada birden fazla müşteri ziyaret edilebilir. Sınırlı bir yaşam süresine (B) sahip tek bir çeşit ürün mevcuttur. Yaşam süresinin o parti içerisinde bulunan tüm ürünlerin üretimi tamamlandıktan sonra başladığı varsayılmaktadır. Yani parti içerisinde hangi sırada üretilmiş olursa olsun tüm müşterilerin talebinin ürün ömrü partinin üretimi tamamlandıktan sonra başlayacaktır. Sistemde her biri coğrafi olarak farklı konumlarda yer alan n adet müşteri ve her müşterinin de sabit bir talebi vardır.

Her tesiste, belirli bir Q kapasitesine sahip tek bir araç mevcuttur. Araç kapasitesi sebebiyle tüm taleplerin tek bir turla sevk edilmesi birçok durumda mümkün olmamaktadır. Bu nedenle araçların yeniden tesise dönerek kalan siparişleri alıp, yeni müşterilere dağıtması gerekmektedir. Yani araçlar birden fazla rota için kullanılabilir. Planlama ufkunun başında araç tesiste hazır olarak beklemektedir. İlk partinin üretimi tamamlanır tamamlanmaz dağıtımına başlanabilir. Araçların hangi tesiste hizmet ettiği belirlidir ve dağıtım işlemi bittikten sonra tekrar aynı tesise dönmeleri gerekmektedir. Araç hızı sabit olup tüm düğüm çiftleri arasındaki mesafe ve geçiş süreleri belirli ve bilinmektedir. Araçların yükleme ve boşaltma süreleri ihmal edilmiştir.

Bu varsayımlar altında problemde amacımız, müşteri siparişlerinin üretilip dağıtımının gerçekleştirileceği en kısa zamanı (D_{max}) belirlerken, hangi müşterinin hangi depodan hizmet göreceğini, hangi müşteri siparişlerinin hangi parti içerisinde olacağını,

her bir partinin üretimine ne zaman başlanıp bitirileceğini ve her bir tur içerisinde müşterilerin hangi sırada ziyaret edileceğini de tespit etmektir.

Bu amaç doğrultusunda her müşteriye yalnızca bir kere ziyaret gerçekleştirilmeli, araca yüklenen ürünlerin toplam alan, hacim, ağırlık vb. gereksinimi araç kapasitesini geçmemeli, her rota bir tesiste başlamalı ve aynı tesiste sonlanmalı ve bir partinin üretimi tamamlandıktan sonra o parti içerisinde bulunan tüm ürünler (siparişler) ilgili müşterilere B birimlik zaman içerisinde teslim edilmelidir.

ÇT_TA_BÜDÇ probleminin daha iyi anlaşılabilmesi için test problemleri içerisinde açıklayıcı bir örnek verilmiştir. Ele alınan problemde müşteri sayısı ve tesis sayısı sırasıyla 10 ve 2 olarak ele alınmıştır ve üretim hızı 2'dir. Dolayısıyla müşteri taleplerinin üretim süresi müşteri talebinin yarısına eşittir. Araç kapasitesi ve ürün ömrü 300 birim olarak göz önüne alınmıştır. Müşteriler $(-100,100)$ 'lük bir koordinat düzleminde rassal olarak konumlanmışlardır. Sistemde her tesiste yalnızca tek bir araç hizmet etmektedir. Çizelge 3.1'de müşteri koordinatları ve talepleri verilmiştir.

Çizelge 3.1. Açıklayıcı Örnek İçin Müşteri Koordinatları ve Talepleri

No	X Koordinatı	Y Koordinatı	Talep
T1	-74	13	0
T2	46	12	0
1	-62	25	125
2	11	-45	139
3	80	-14	199
4	-87	-12	180
5	14	49	153
6	42	69	111
7	77	16	127
8	49	21	187
9	-74	27	187
10	49	-12	144

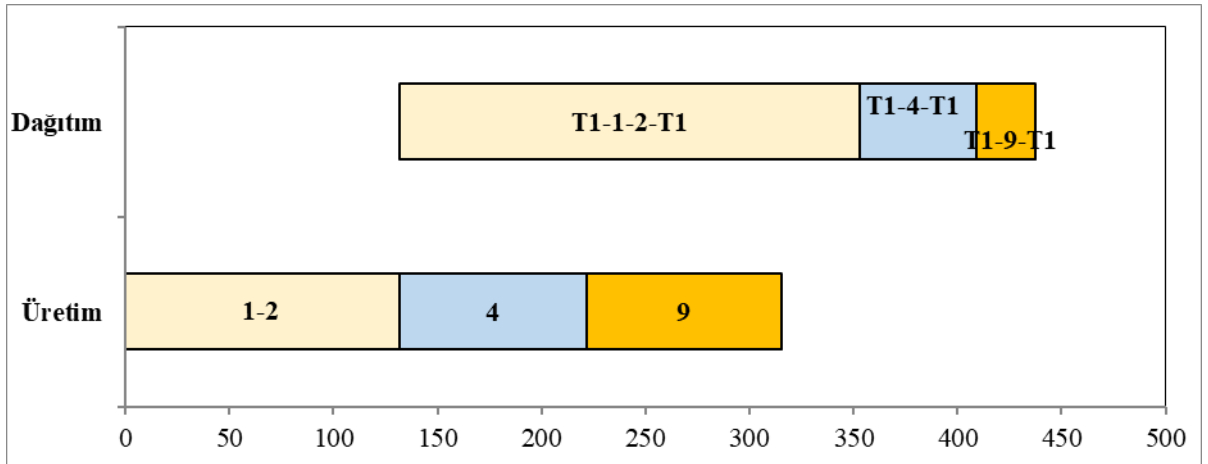
Çizelge 3.1.'de örnek problem için oluşturulan X ve Y koordinatları ile $(-100,100)$ arasında rassal oluşturulan müşteri talepleri verilmiştir. Buna göre ilk iki satırdaki T1 ve T2 değerleri tesisleri, diğer satırlar ise 10 adet müşteriyi göstermektedir. T1 ve T2 tesislerinin konumları diğer müşterilerin konumlarına bağlı olarak belirlenmiş ve ağırlık merkezlerine göre konumlandırılmıştır. Bu problem için sistemde 2 adet tesis mevcut olduğu için koordinat sistemi $(-100,0)$ ve $(0,100)$ olmak üzere 2 eşit parçaya ayrılmıştır. Her iki parçadaki müşterilerin koordinatları göz önüne alınarak düzlemin ağırlık merkezi

belirlenmiş ve tesisler bu ağırlık merkezlerine kurulmuştur. Tesislerin herhangi bir talebi bulunmamaktadır. Açıklayıcı örneğin uygun bir çözümüne göre tesislerdeki atamalar ve rotalar Çizelge 3.2'deki gibidir.

Çizelge 3.2. Açıklayıcı Örnek İçin Elde Edilen Rota Atamaları

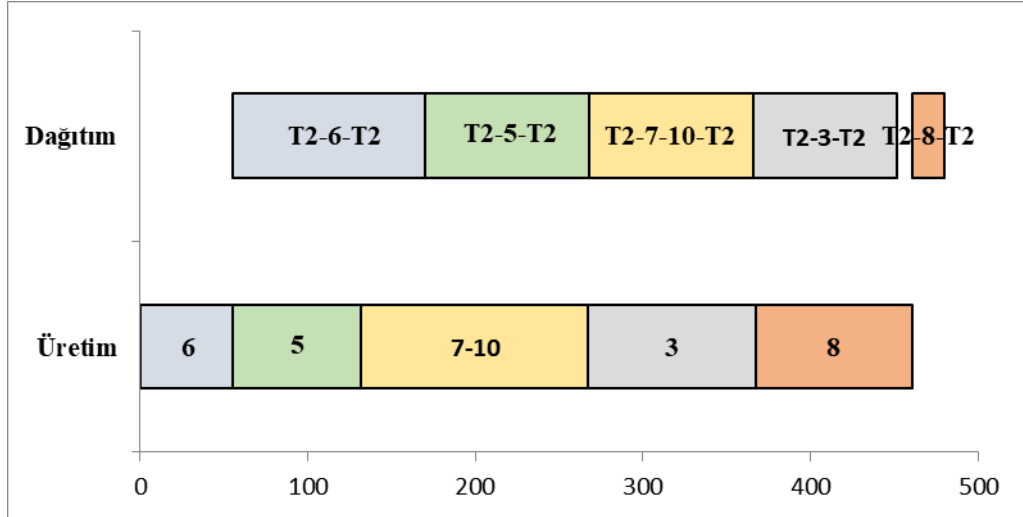
	1.Tesis	Üretim Süresi	Dağıtım Süresi	2.Tesis	Üretim Süresi	Dağıtım Süresi
Atanan Rotalar	T1-1-2-T1	132	221.01	T2-6-T2	55.5	114.28
	T1-4-T1	90	56.36	T2-5-T2	76.5	97.84
	T1-9-T1	93.5	28	T2-7-10-T2	135.5	95.05
				T2-3-T2	99.5	85.6
				T2-8-T2	93.5	18.98

Çizelge 3.2'de hangi müşterilerin hangi tesise atandığı, müşteri taleplerinin hangi rotada ve hangi sırayla karşılanacağı, rotaların toplam üretim süreleri ve rotaların toplam dağıtım süreleri verilmiştir. Buna göre 1-2-4 ve 9 numaralı müşteriler 1.Tesis (T1)'e atanırken, 3-5-6-7-8 ve 10 numaralı müşteriler 2.Tesis (T2)'e atanmıştır. T1 için ilk olarak tesiste 1 ve 2 numaralı müşterilerin talepleri üretilecektir. Dağıtım aşamasında araç 1 numaralı müşteriye uğrayacak daha sonra 2 numaralı müşteriye geçerek tesise dönecektir. Daha sonra sonraki rota olan 4 numaralı müşterinin talebinin üretimi tamamlandıysa o müşterinin talebi için dağıtıma çıkacak yoksa üretimin bitmesini bekleyecektir. T1 için rota sıralaması T1-1-2-T1 / T1-4-T1 / T1-9-T1 olarak belirlenmiştir. Benzer şekilde T2 için T2-6-T2 / T2-5-T2 / T2-7-10-T2 / T2-3-T2 / T2-8-T2 olacak şekilde 5 rota mevcuttur. Araç kapasitesinin 300 birim olmasından dolayı birçok müşteri aynı rotada birleştirilememektedir. Benzer şekilde bazı müşteriler ise ürün ömrü kısıtlamasından dolayı aynı rota içerisine alınamamaktadır. T1 tesisinde 1 ve 2 numaralı müşterilerin talepleri birlikte üretilip dağıtılırken, T2 tesisinde ise 7 ve 10 numaralı müşterilerin talepleri birlikte üretilip dağıtılmaktadır. Çizelge 3.2'deki verilere göre elde edilen Gantt şemaları Şekil 3.1 ve Şekil 3.2'de gösterilmiştir.



Şekil 3.1. T1 Tesisinin Gantt Şeması

T1 tesisinin Gantt Şeması Şekil 3.1’de gösterilmektedir. Tesise atanan müşteriler ile 3 rota oluşturulmuş ve bu rotalardaki müşterilerin talepleri üretilip dağıtılmıştır. İlk rota olan 1 ve 2 numaralı müşterilerin toplam talebi 264 birimdir. Üretim hızının 2 olması nedeniyle bu 264 birimlik talep $264/2=132$ birim zamanda üretilecektir. Araç bu sürede tesiste hazır olarak beklemektedir. 132. zaman diliminde müşteri taleplerinin üretimi bitmiş ve dağıtım işlemi başlamıştır. Araç 1 numaralı müşteriye 148.97 zaman diliminde ulaşmıştır. Sonrasında 1 numaralı müşteriden 2 numaralı müşteriye yola çıkmış ve 250.11 zaman diliminde de 2 numaralı müşteriye ulaşmıştır. Aracın tekrar tesise döndüğü süre 353.01 zaman dilimi olarak hesaplanmıştır. İkinci rotanın üretimi ilk rotanın üretiminden hemen sonra başlamıştır. 4 numaralı müşterinin talebi 222 zaman diliminde tamamlandığında araç henüz tesise geri dönmemiştir. Bu nedenle 4 numaralı müşterinin talebi tesiste bekleyecektir. Aracın dağıtımının çok uzun olması nedeniyle 4 numaralı müşterinin ürün tesliminde ürün ömrünün aşılması söz konusu olursa 4 numaralı müşterinin talebinin üretimi ürün ömrü aşılmayacak şekilde ötelenebilir. Toplam 3 rotadaki tüm müşterilerin taleplerinin üretimi 315.5 zaman diliminde tamamlanmış ve son rotanın dağıtımını ise 437.37 zaman diliminde sona ermiştir. Bu süre bize aynı zamanda T1 için D_{max} değerini vermektedir.



Şekil 3.2. T2 Tesisinin Gantt Şeması

T2 tesisine atanan müşteriler ve rotalara göre oluşturulan Gantt şeması Şekil 3.2’de gösterilmektedir. Buna göre T1’de olduğu gibi 0 zamanında araç tesiste hazır beklemekte ve üretim tamamlanır tamamlanmaz ilk rotadaki müşterilerin taleplerini teslim etmek üzere yola çıkmaktadır. Şekilde de görüldüğü üzere 6 numaralı müşterinin talebi ilk olarak üretilmiş ve 55.5 zaman diliminde üretimi tamamlanarak dağıtıma çıkmıştır. Bu süre içerisinde bir sonraki rota olan 5 numaralı müşterinin talebi üretilmeye başlanmıştır. T1’de olduğu gibi 5 numaralı müşterinin talebi üretilip tamamlandığı zaman araç henüz tesise dönmediği için üretilen ürünlerin tesiste beklemesi gerekmektedir. 3 numaralı müşterinin talebi üretilip dağıtımını tamamlandığında araç 448.39 zaman diliminde tesise dönmüş olacaktır. Bu süre içerisinde bir sonraki rota olan 8 numaralı müşterinin talebinin üretimi tamamlanmadığı için araç 460.50 zaman dilimine kadar tesiste üretimin tamamlanmasını bekleyecektir. Tüm müşteri taleplerinin tesliminin tamamlanması ve aracın rotaya dönmesi için gerekli toplam zaman 479.48’dir. Bu süre bize aynı zamanda T2 için D_{max} değerini vermektedir.

Her iki tesis için T1’in D_{max} değeri 437.37 ve T2’nin D_{max} değeri ise 479.48 olarak hesaplanmıştır. Bu durumda amaç fonksiyonumuz D_{max} minimizasyonu olduğu için $EnBüyükD_{max}$ T2 numaralı tesisteki 479.48’dir. Tüm bu üretim ve dağıtım işlemleri her iki tesis için 479.48 zaman diliminde tamamlanacaktır.

Aynı problemin farklı türleri için farklı D_{max} değerleri elde edilebilir. Örneğin üretim hızının 1 olması durumunda üretim süreleri uzayacak dolayısıyla D_{max} değeri de değişecektir. Ya da araç kapasitesinin 600 olması durumunda birkaç farklı müşterinin daha aynı rotada hizmet görmesi sağlanacak ve yine amaç fonksiyonu değişecektir.

3.2. ÇT_TA_BÜDÇ Problemi İçin Önerilen Matematiksel Model

Bu bölümde ÇT_TA_BÜDÇ problemi için önerilen matematiksel model ve modeldeki kısıtlamaların açıklamaları aşağıda verilmiştir.

Dizin Kümeleri ve Parametreler

- N_0 : Tesisler Kümesi
 N_C : Müşteriler Kümesi
 N : Tüm Dğümler Kümesi ($N = N_0 \cup N_C$)
 MS : Müşteri Sayısı
 B : Ürün Ömrü
 t_{ij} : i düğümünden j düğümüne geçiş süresi ($\forall i, j \in N$)
 d_i : i müşterisinin talebi
 r : Tesislerdeki üretim hızı
 p_i : i müşterisinin talebinin üretim süresi ($p_i = d_i/r$)
 Q : Araç kapasitesi
 M : Büyük bir sayı

İkili Karar Değişkenleri

- $x_{ij} = \begin{cases} 1 & i \text{ düğümünden } j \text{ düğümüne gidilirse } (\forall i, j \in N) \\ 0 & \text{diğer durumda} \end{cases}$
 $y_{ik} = \begin{cases} 1 & i \text{ düğümü } k \text{ deposuna atanırsa } (\forall i \in N_C, \forall k \in N_0) \\ 0 & \text{diğer durumda} \end{cases}$
 $w_{ij} = \begin{cases} 1 & \text{rota } i \text{ düğümünde bitip, sonraki rota } j \text{ düğümünde başlıyorsa} \\ & (\forall i, j \in N) \\ 0 & \text{diğer durumda} \end{cases}$
 $z_k = \begin{cases} 1 & k \text{ tesisine atama yapılırsa } (\forall k \in N_0) \\ 0 & \text{diğer durumda} \end{cases}$

Ek Karar Değişkenleri

- u_i : i düğümüne girmeden hemen önce araçtaki yük miktarı ($\forall i \in N_C$)
 s_i : i düğümünün servis başlangıç zamanı ($\forall i \in N_C$)
 f_i : i düğümünün talebinin üretiminin bitiş zamanı ($\forall i \in N_C$)

Model

$$\text{En Küçük } D_{max} \quad (1)$$

$$\sum_{i \in N} x_{ij} = 1 \quad \forall j \in N_C \quad (2)$$

$$\sum_{j \in N} x_{ji} = \sum_{j \in N} x_{ij} \quad \forall i \in N \quad (3)$$

$$\sum_{k \in N_0} y_{ik} = 1 \quad \forall i \in N_C \quad (4)$$

$$x_{ik} \leq y_{ik} \quad \forall i \in N_C, \forall k \in N_0 \quad (5)$$

$$x_{ki} \leq y_{ik} \quad \forall i \in N_C, \forall k \in N_0 \quad (6)$$

$$x_{ij} + y_{ik} + \sum_{\substack{m \in N_0 \\ m \neq k}} y_{jm} \leq 2 \quad \forall i, j \in N_C; i \neq j, \forall k \in N_0 \quad (7)$$

$$u_j - u_i + Qx_{ij} + (Q - d_i - d_j)x_{ji} \leq Q - d_i \quad \forall i, j \in N_C; i \neq j \quad (8)$$

$$u_i \geq d_i + \sum_{\substack{j \in N_C \\ i \neq j}} d_j x_{ij} \quad \forall i \in N_C \quad (9)$$

$$u_i \leq Q - (Q - d_i) \sum_{k \in N_0} x_{ik} \quad \forall i \in N_C \quad (10)$$

$$\sum_{\substack{j \in N_C \\ i \neq j}} w_{ij} \leq \sum_{k \in N_0} x_{ik} \quad \forall i \in N_C \quad (11)$$

$$\sum_{\substack{i \in N_C \\ i \neq j}} w_{ij} \leq \sum_{k \in N_0} x_{kj} \quad \forall j \in N_C \quad (12)$$

$$\sum_{j \in N_C} \sum_{k \in N_0} x_{kj} - \sum_{i \in N_C} \sum_{j \in N_C} w_{ij} = \sum_{k \in N_0} z_k \quad (13)$$

$$s_i - s_j + Mx_{ij} + (M - t_{ij} - t_{ji})x_{ji} \leq M - t_{ij} \quad \forall i, j \in N_C; i \neq j \quad (14)$$

$$s_i - s_j + Mw_{ij} \leq M - \sum_{k \in N_0} t_{ik} x_{ik} - \sum_{k \in N_0} t_{kj} x_{kj} \quad \forall i, j \in N_C; i \neq j \quad (15)$$

$$f_i - s_j + Mw_{ij} \leq M - u_j/r - \sum_{k \in N_0} t_{kj} x_{kj} \quad \forall i, j \in N_C; i \neq j \quad (16)$$

$$s_j \geq u_j/r + \sum_{k \in N_0} t_{kj} x_{kj} - M \left(1 - \sum_{k \in N_0} x_{kj} + \sum_{\substack{i \in N_C \\ i \neq j}} w_{ij} \right) \quad \forall j \in N_C \quad (17)$$

$$s_j \leq u_j/r + \sum_{k \in N_0} t_{kj} x_{kj} + M \left(1 - \sum_{k \in N_0} x_{kj} + \sum_{\substack{i \in N_C \\ i \neq j}} w_{ij} \right) \quad \forall j \in N_C \quad (18)$$

$$f_i - f_j + Mx_{ij} + (M - p_i - p_j)x_{ji} \leq M - p_j \quad \forall i, j \in N_C; i \neq j \quad (19)$$

$$f_i - f_j + Mw_{ij} \leq M - p_j \quad \forall i, j \in N_C; i \neq j \quad (20)$$

$$f_j \geq p_j + \sum_{\substack{i \in N_C \\ i \neq j}} p_i (x_{ij} + w_{ij}) \quad \forall j \in N_C \quad (21)$$

$$f_j \leq M - (M - p_j) \left(\sum_{k \in N_0} x_{kj} - \sum_{\substack{i \in N_C \\ i \neq j}} w_{ij} \right) \quad \forall j \in N_C \quad (22)$$

$$s_i - f_i \leq B + M(1 - \sum_{k \in N_0} x_{ik}) \quad \forall i \in N_C \quad (23)$$

$$\sum_{k \in N_0} (k + 1)(y_{ik} - y_{jk}) + M(1 - w_{ij}) \geq 0 \quad \forall i, j \in N_C; i \neq j \quad (24)$$

$$\sum_{k \in N_0} (k + 1)(y_{ik} - y_{jk}) - M(1 - w_{ij}) \leq 0 \quad \forall i, j \in N_C; i \neq j \quad (25)$$

$$\sum_{i \in N_C} y_{ik} - MS(z_k) \leq 0 \quad \forall k \in N_0 \quad (26)$$

$$\sum_{i \in N_C} y_{ik} - MS(z_k) \geq 1 - MS \quad \forall k \in N_0 \quad (27)$$

$$D_{max} \geq s_i + t_{ik}x_{ik} \quad \forall i \in N_C, \forall k \in N_0 \quad (28)$$

$$s_i, u_i, f_i \geq 0 \quad \forall i \in N_C \quad (29)$$

$$x_{ij}, w_{ij}, z_k \in \{0,1\} \quad \forall i, j \in N_C, \forall k \in N_0 \quad (30)$$

Matematiksel modelde amaç (1), müşteri taleplerinin üretimi ve dağıtımını için gerekli olan zamanın enküçülenmesidir (yayımlım zamanı / makespan). (2) numaralı kısıt her düğüme (müşteriye) bir kez uğramayı garantilerken, (3) numaralı kısıt bütün düğümlere giren ve çıkan hat sayısını birbirine eşitlemektedir. Her müşterinin bir tesise atanmasını garantileyen kısıt (4) numaralı kısıttır.

(5) – (7) numaralı kısıtlar bir rotanın aynı tesiste başlayıp yine aynı tesiste sonlanmasını garantilemektedir. (5) numaralı kısıt rota üzerindeki son müşterinin dönüş yaptığı tesise, (6) numaralı kısıt rota üzerindeki ilk müşterinin çıkış yaptığı tesise atanmasını sağlarken, (7) numaralı kısıt rota üzerinde ardışık olan iki müşterinin farklı tesislere atanmasını engellemektedir. (8) – (10) numaralı kısıtlar alt tur eleme kısıtlarıdır. (8) numaralı kısıt aracın tesisten çıkış anındaki yük miktarını belirtirken (9) ve (10) numaralı kısıtlar ise herhangi bir düğümden çıkıldığı zaman araçtaki yük miktarı ile ilgili alt sınırları vermektedir. Bu değerler aynı zamanda alt turları engellemek için kullanılan u_i geçici değişkeninin alt ve üst sınırlarını belirlemektedir.

(11) ve (12) numaralı kısıtlar ardışık iki rota için sırasıyla o rotalardaki ilk ve son müşteri olma durumlarını garantilemektedir. (13) numaralı kısıt, toplam oluşturulan rota sayısı ile ardışık rota sayısı arasındaki farkın kullanılan tesis sayısına eşit olmasını garanti etmektedir. (14) – (22) numaralı kısıtlar zaman kısıtlarıdır. (14) numaralı kısıt aynı rotada bulunan ardışık iki düğüm için servis başlangıç zamanlarını belirlerken, (15) numaralı kısıt ardışık rotalardaki ilk müşteriler için servis başlangıç zamanlarını belirtmektedir. (16) numaralı kısıt herhangi bir düğümün üretim tamamlanma zamanı ile servise başlama zamanı arasındaki ilişkiyi belirlemektedir. Servis başlangıç zamanı için alt ve üst sınırlar ise (17) ve (18) numaralı kısıtlar ile belirlenmiştir.

Benzer şekilde (19) numaralı kısıt ile aynı rotada bulunan ardışık iki düğüm için üretim tamamlanma zamanları belirlenmektedir. (20) numaralı kısıt, ardışık rotalardaki ilk müşterilerin üretim bitiş zamanlarını vermektedir. Üretim bitiş zamanları için alt ve üst sınır değerleri ise (21) ve (22) numaralı kısıtlar ile belirlenmiştir.

(23) numaralı kısıt ürün ömrünün aşılmasını garanti etmektedir. (24) ve (25) numaralı kısıtlar ardışık rotaların aynı tesislere atanmasını garantilemektedir. (26) ve (27) numaralı kısıtlar ise herhangi bir tesise atama yapıp yapılmadığını belirlemektedir.

(28) numaralı kısıt yayılım zamanını belirlemektedir. Son olarak (29) ve (30) numaralı kısıtlar ise işaret kısıtlarıdır.

3.3. Değişken Komşu Arama (DKA) Algoritması

DKA algoritması Mladenović ve ark. (1997) tarafından geliştirilen ve kombinatoriyal optimizasyon problemlerinde kullanılan yerel arama tabanlı bir metasezgisel yöntemdir. Klasik yerel arama prosedüründe tek bir komşuluk yapısı kullanılırken, DKA algoritmasında birden fazla komşuluk yapısı kullanılmaktadır. Herhangi bir komşulukta elde edilen yerel optimal nokta diğer komşuluklar için farklılık gösterebileceğinden, farklı komşuluk yapılarının kullanımı arama uzayının farklı bölgelerine ve oradaki yerel optimal noktalara erişim sağlanması açısından oldukça avantajlıdır. DKA algoritmasının önemli bir özelliği de komşuluk yapıları arasındaki değişim stratejisidir. Herhangi bir komşuluk yapısı yerel optimal noktaya takıldığında, komşuluk yapıları değiştirilerek durdurma kriteri sağlanana kadar algoritma devam ettirilir. DKA'da N_k , ($k = 1, 2, \dots, k_{max}$) sınırlı sayıdaki komşuluk yapılarının kümesini ve $N_k(S)$ s çözümünün k . komşuluk yapısıyla elde edilen çözümlerinin kümesini ifade eder. Genellikle, her s çözümü için bir komşuluk yapısından iç içe geçmiş komşuluklar serisi elde edilir. Bu k . komşuluğa doğru yapılacak hareket başlangıç komşuluğuna k defa yapılacak hareketle gerçekleşir. $s' \in S$ (çözüm uzayı) çözümü N_k 'ya göre bir yerel optimumdur. Bu durum s' değerinden daha iyi bir çözüm olmadığında geçerlidir (Hansen ve Mladenović, 2001). DKA'nın Değişken Komşu İniş (DKİ), Temel Değişken Komşu Arama (TDKA), İndirgenmiş Değişken Komşu Arama (İDKA) ve Genel Değişken Komşu Arama (GDKA) olmak üzere 4 farklı türü bulunmaktadır. Tez kapsamında problem için Değişken Komşu İniş algoritmasına dayalı bir algoritma geliştirilmiştir. Çizelge 3.3'te DKA algoritmasının genel adımları verilmiştir.

Çizelge 3.3. DKA Algoritmasının Adımları

Prosedür:	DKA Algoritması Prosedürü
Girdi:	S_0 : Başlangıç Çözümü
Çıktı:	S : DKA Algoritması Sonrası Elde Edilen Çözüm
Adım 1:	N_k Komşuluk yapılarını belirle ve sırala ($N_k; k = 1, 2, \dots, k_{max}$)
Adım 2:	$k \leftarrow 1, S \leftarrow S_0$ olarak ayarla.
Adım 3:	$k = k_{max}$ oluncaya kadar aşağıdaki adımları tekrarla.
Adım 3.1:	k . komşuluk yapısını kullanarak uygun çözümler arasından (S) en iyi uygun çözümü (S') seç.
Adım 3.2:	<i>Amaç Fonksiyonu Hesaplama</i> prosedürü ile $f(S')$ değerini elde et.
	- <i>eğer</i> $f(S') < f(S)$ ise $S \leftarrow S', f(S) \leftarrow f(S')$ ve $k \leftarrow 1$
	- <i>değilse</i> $k \leftarrow k + 1$

DKA komşuluk yapılarının belirlenmesi ve sıralanmasıyla başlamaktadır (Adım 1). İlk komşuluk yapısından başlanarak tüm komşuluk yapıları çözüme sırasıyla uygulanmaktadır (Adım 2). Her komşuluk yapısında tüm çözümler arasından en iyi çözüm belirlenir (Adım 3). Eğer elde edilen çözüm şimdiye kadar bulunan en iyi çözüm ise algoritma ilk komşuluk yapısından yeni çözüm ile tekrar başlamaktadır. Elde edilen yeni çözüm en iyi çözüm değilse bir sonraki komşuluk yapısı ile algoritmaya devam edilir. Son komşuluk yapısı sonucunda da herhangi bir iyileşme elde edilmediyse algoritma sonlandırılır (Adım 3.1 – Adım 3.2).

DKA algoritması önceden tanımlanmış olan komşuluklarda rassal veya belirli bir sırada yerel arama yaparak yerel optimal çözümlerden kaçınmayı amaçlar. Global optimal çözüm tüm komşuluk yapılarına göre en iyi çözüm olacağı için bütün komşuluklarda yerel optimal çözümleri arayarak global optimal çözüme ulaşılması hedeflenmektedir (Talbi, 2009). DKA'da birden fazla komşuluk yapısı kullanıldığı için dikkat edilmesi gereken bazı noktalar vardır (Hansen ve Mladenović, 2001): *i*) Kullanılacak olan komşuluk yapıları problem için uygun ve iyi çözümler verebilecek komşuluk yapıları olmalıdır. Her bir komşuluk yapısında farklı çözümler elde edileceği için komşulukların seçimi önemlidir. *ii*) Kullanılacak olan komşuluk yapıları seçildikten sonra bu komşuluk yapılarının hangi sırada uygulanacağı da önemli bir diğer noktadır. Bir komşulukta mevcuttan daha iyi bir çözüm elde edildiğinde bu çözüm hafızaya alınmakta ve tekrar ilk sıradaki komşuluğa dönülmektedir. İyileşme olmadığı sürece ise sırasıyla bütün komşuluk yapıları uygulanmaya devam edilir. Dolayısıyla sonraki çözümlerin verimi önceki döngülerde bulunan çözümlere de bağlıdır. *iii*) Komşuluk yapılarının değiştirilmesi için stratejinin ne olacağı da önemli bir konudur.

3.3.1. ÇT_TA_BÜDÇ problemi için önerilen değişken komşu arama (DKA) algoritması

Bir önceki bölümde genel adımları verilen DKA algoritmasının ÇT_TA_BÜDÇ problemine uyarlanmış hali bu bölümde ayrıntılı olarak açıklanmıştır. Algoritmanın başlangıç çözümü için çizelgeleme ve araç rotalama problemlerinde kullanılan hızlı ve etkili sezgisellerden birisi olan Clarke ve Wright (1964) tarafından geliştirilen tasarruf algoritması kullanılarak probleme özgü bir sezgisel algoritma geliştirilmiştir. Başlangıç algoritmasının adımları Çizelge 3.4’te verilmiştir.

Çizelge 3.4. ÇT_TA_BÜDÇ Problemi İçin Başlangıç Uygun Çözüm Elde Etme Prosedürü

Prosedür:	Başlangıç Çözümünün (<i>BaşUygunÇözüm</i>) Elde Edilmesi
Girdi:	ÇT_TA_BÜDÇ Problem Bilgileri
Çıktı:	$S_{Baş}$: Başlangıç Çözümü
Adım 1:	Tüm müşterileri kendisine en yakın olan tesise ata.
Adım 2:	Tüm müşteriler için atandıkları tesisten (T_i) rotalar oluşturarak ilk uygun çözümleri $S_{Baş(i)}$ elde et. ($i = 1, 2, \dots, N$; $N = Tesis Sayısı$)
Adım 3:	Her tesis için <i>Amaç Fonksiyonu Hesaplama</i> prosedürünü uygula ve ilk çözümün amaç fonksiyonu değerlerini ($f(S_{Baş(i)})$) hesapla.
Adım 4:	Her tesis için <i>Birleştirme</i> prosedürünü uygula ve $f(S_{Baş(i)})$ değerlerini güncelle.
Adım 5:	$EnBüyükD_{max} \leftarrow EnBüyükf(S_{Baş(i)})$

Çizelge 3.2’de görüldüğü üzere başlangıç algoritmasında öncelikle müşteriler, kendilerine en yakın tesise atanır (Adım 1). Bütün müşteriler bir tesise atandıktan sonra her müşteri için atandığı tesisten her turda sadece o müşteri olacak şekilde rotalar oluşturulur. Böylelikle her müşterinin ayrı birer rota oluşturduğu ilk uygun çözümler elde edilmiş olur (Adım 2). Her tesis için rota sıralaması o tesise atanma sırasına eşittir. Rota sıralamaları belirlendikten sonra ise her tesis için *Amaç Fonksiyonu Hesaplama* prosedürüne geçilir (Adım 3). *Amaç Fonksiyonu Hesaplama* prosedürü ayrıntılı bir şekilde Çizelge 3.5’te ifade edilmiştir. Elde edilen ilk rotalar *Birleştirme* prosedürü kullanılarak tesislerin D_{max} değerinde en fazla iyileştirmeyi sağlayacak şekilde birleştirilmektedir ve bu adım herhangi bir uygun birleştirme işlemi kalmayana kadar devam etmektedir (Adım 4). Her tesis için üretim ve dağıtım operasyonlarının toplam tamamlanma zamanı D_{max} belirlendikten sonra, bu değerler arasından en büyük olanı ($EnBüyükD_{max}$) belirlenerek komşu arama prosedürleriyle bu sonucun iyileştirilmesi amaçlanmaktadır (Adım 5).

Çizelge 3.5. Amaç Fonksiyonu Hesaplama Prosedürü

Prosedür:	Amaç Fonksiyonu Hesaplama Prosedürü
Girdi:	$S_{Baş(i)}$ Çözümleri
Çıktı:	$f(S_{Baş(i)})$ ve $EnBüyükf(S_{Baş(i)})$ Değerlerinin belirlenmesi
Adım 1:	Her bir tesis için aşağıdaki adımları tekrarla ($k \in N_0$).
Adım 1.1:	$PT_k \leftarrow 0$ ve $DT_k \leftarrow 0$
Adım 1.2:	k tesisine atanan tüm rotalara sırasıyla aşağıdaki adımları uygula.
Adım 1.3:	$PT_k \leftarrow PT_k + P(R_j)$ <i>eğer</i> $PT_k \geq DT_k$ <i>ise</i> $DT_k \leftarrow PT_k + D(R_j)$ <i>eğer</i> $(DT_k + D(R_j) - d(R_{jk}, p_k) - PT_k) > B$ <i>ise</i> $PT_k \leftarrow PT_k + [DT_k + D(R_j) - d(R_{jk}, p_k) - PT_k - B]$ <i>değilse</i> $DT_k \leftarrow DT_k + D(R_j)$
Adım 1.4:	$D_{max}(k) \leftarrow DT_k$
Adım 2:	$EnBüyükDmax \leftarrow EnBüyükf(S_{Baş(i)})$
Adım 3:	$EnBüyükDmax$ değerine sahip tesise atanan rotaların sırasını değiştirerek daha iyi bir çözüm olup olmadığını kontrol et.

Amaç Fonksiyonu Hesaplama Prosedüründe önemli olan nokta, üretim ve dağıtım sürelerinin değerlerinin birbirlerine göre belirlenecek olmasıdır (Adım 1). Yani üretim zamanının daha uzun olduğu durumlarda araç boş beklemesi meydana gelebilecekken, dağıtımın uzun olduğu durumlarda ise üretimler arasında boşluklar meydana gelebilecektir. Her tesis için sıranın ilk başından başlanarak rotaların üretim ve dağıtım sürelerine göre tesislerin toplam üretim (PT_k) ve toplam dağıtım zamanları (DT_k) belirlenir. Çözüme başlarken PT_k ve DT_k değerleri 0 olarak atanır (Adım 1.1). İlk rotanın üretim süresi $PT_i = P(R_1)$ olarak hesaplanır. Bu süre içerisinde araç tesiste boş beklemektedir. Üretim biter bitmez araç dağıtıma çıkacaktır. Dağıtıma başlama zamanı PT_k iken dağıtım bitiş zamanı ise $DT_k = P(R_1) + D(R_1)$ olacaktır. Daha sonraki rotalar için üretim ve dağıtım zamanları mevcut PT_k ve DT_k süreleri ile toplanarak her tesis için o tesise atanan rotaların hepsi bitene kadar adımlar devam ettirilir. Bu hesaplamalarda 3 farklı durum söz konusudur. Bunlar (Adım 1.3):

- Eğer $PT_k \geq DT_k$ ise $DT_k = PT_k + D(R_j)$ olarak hesaplanır. Yani araç tesise dönüş yaptıktan sonra diğer rotanın üretiminin bitmesini bekleyerek üretim bittikten sonra dağıtıma çıkabilir.
- Eğer $PT_i \leq DT_i - d(R_{(2)k}, T_i) - B$ ise $P(R_2)$ 'nin üretim tamamlanma zamanını $DT_k - d(R_{(j)k}, T_k) - B - PT_k$ kadar ötele. ($d(R_{(j)k}, T_k)$): R_j rotasının son elemanı ile atandığı tesis arasındaki uzaklık). Bu durumda bir diğer rotanın

üretim süresi ürün ömrünü aşmayacak şekilde mümkün olan en az şekilde ötelenmektedir.

- Eğer bu iki durumun hiç birisi meydana gelmiyorsa da $DT_k = D(R_j)$ olarak hesaplanır. Bu durumda sadece rotanın uzunluğu göz önüne alınmış olur.

Tüm tesisler ve tesislere atanan rotalar için bu hesaplamalar yapıldıktan sonra her tesisin DT_k değeri belirlenmiş olur (Adım 1.4). Bu değerlerden en büyük olanı algoritma için $EnBüyükD_{max}$ değeri olarak belirlenir (Adım 2). $EnBüyükD_{max}$ değerine sahip olan tesis için tesisteki rotaların sıralamasının değişmesi Amaç Fonksiyonunda iyileşme sağlayabilir. Bu nedenle o tesisteki mevcut rotaların sıralaması değiştirilerek tüm alternatif rota sıralaması için Amaç fonksiyonu hesaplanarak D_{max} değerinde en fazla iyileşme sağlayan çözüm yeni çözüm olarak kabul edilir ve $EnBüyükD_{max}$ değeri olarak güncellenir (Adım 3).

Bundan sonraki adımlarda bu değer iyileştirme prosedürü kullanılarak mümkün olduğunca azaltılmaya çalışılacaktır. *İyileştirme* prosedürünün adımları Çizelge 3.6'da verilmektedir. *İyileştirme* prosedüründe, etkin bir çözüm yöntemi için önemli özellikler olan basit, hızlı ve etkin komşu arama mekanizmaları kullanılmıştır. Bu prosedürde kullanılan komşu arama mekanizmaları şu şekildedir:

- *Birleştirme*: Bu mekanizmada tüm rota çiftleri için ($R_i, R_j: R_i \neq R_j$), ilk rotanın son müşterisi ile ikinci rotanın ilk müşterisi birbirine bağlanarak birleştirilir. Birleştirme sonucunda elde edilen yeni rotanın (R_y) araç kapasitesi ve ürün ömrü bakımından uygunluğu kontrol edilir ve eğer uygun ise bir tesise atanır. Tesis ataması için değişik alternatifler söz konusudur. Bunlar sırasıyla; i) R_i 'in atandığı tesiste birleştirme (T_i), ii) R_j 'nin atandığı tesiste birleştirme (T_j), iii) T_i ve T_j 'den farklı açık olan başka bir tesiste birleştirmedir. Birleştirme alternatifleri içerisinde Amaç Fonksiyonu değerinde en fazla iyileştirmeyi sağlayan birleştirme işlemi gerçekleştirilerek hiç uygun birleştirme alternatifi kalmayana kadar prosedüre devam edilir.
- *Yer değiştirme (Swap)*: Herhangi iki müşterinin rotadaki yerlerinin değiştirilmesidir. Bu yer değiştirme işlemi için farklı alternatifler söz konusudur. Bunlar; i) Aynı rotadaki müşteriler ile yer değiştirme ($Swap(0,0)$) ii) Aynı tesis fakat farklı rotadaki müşteriler ile yer değiştirme ($Swap(1,0)$) ve iii) Farklı tesisteki müşteriler ile yer değiştirme ($Swap(1,1)$)'dir.

- *Yerleştirme/Ekleme (Insert)*: Herhangi bir müşterinin bir rotadaki herhangi bir yere eklenmesidir. Farklı alternatifler söz konusudur. Bunlar; i) Aynı rotaya ekleme (*Insert (0,0)*) ii) Aynı tesis fakat farklı rotaya ekleme (*Insert (1,0)*) ve iii) Farklı tesisteki rotaya (*Insert (1,1)*) eklemedir.
- *Rota Taşıma*: Herhangi bir tesisteki bir rotanın tamamının diğer tesislerdeki herhangi bir sıraya taşınmasıdır.

ÇT_TA_BÜDÇ problemi için amacımızın D_{max} enküçüklemesi olduğu göz önüne alınırsa tüm bu komşu arama prosedürlerinin öncelikle $EnBüyükD_{max}$ değerine sahip olan tesis için uygulanması problemin amaç fonksiyonu değerine daha çok katkı sağlayacak ve algoritmanın performansını artıracaktır.

Bu komşu arama mekanizmaları, değişik ARP problemleri için başarılı bir şekilde uygulanmış mekanizmalardır. Başarılı bir birleştirme işleminden sonra mutlaka bir rota bazen de bir depo kapatılır. Bu durum birçok ARP problemi için amaç fonksiyonu değerinde de iyileşme sağlamış olsa da ÇT_TA_BÜDÇ problemi için her zaman amaç fonksiyonunu iyileştirmeyebilir. Diğer komşu arama mekanizmalarında da olduğu gibi bazı durumlarda gerçekleştirilen işlemler çeşitli nedenlerle (araç boş beklemesinin artması, üretimin çok fazla ötelenmesi, ürün ömrü kısıtının ihlal edilmesi vs.) daha kötü amaç fonksiyonu değerlerine neden olabilir. Bu nedenle her komşu arama prosedüründen sonra Amaç Fonksiyonu değerinin yeniden hesaplanması gerekir.

Çizelge 3.6. ÇT_TA_BÜDÇ Problemi İçin İyileştirme Prosedürü

Prosedür:	İyileştirme Prosedürü
Girdi:	$S_{Baş(i)}$: ÇTBÜDÇ probleminin BaşUygunÇözümü
Çıktı:	S_{Yeni} : İyileştirilmiş ÇTBÜDÇ problemi Çözümü
Adım 1:	Her rota için (R_i) için rota taşıma prosedürünü uygula.
Adım 2:	Aynı rotadaki müşteriler için ekleme prosedürünü uygula (<i>Insert 0,0</i>).
Adım 3:	Aynı tesis farklı rotadaki müşteriler için ekleme prosedürünü uygula (<i>Insert 1,0</i>).
Adım 4:	Farklı tesislerdeki müşteriler için ekleme prosedürünü uygula (<i>Insert 1,1</i>).
Adım 5:	Aynı rotadaki müşteriler için yer değiştirme prosedürünü uygula (<i>Swap 0,0</i>).
Adım 6:	Aynı tesis farklı rotadaki müşteriler için yer değiştirme prosedürünü uygula (<i>Swap 1,0</i>).
Adım 7:	Farklı tesislerdeki müşteriler için yer değiştirme prosedürünü uygula (<i>Swap 1,1</i>).
Adım 8:	Her rota çifti (R_i, R_j) için rota birleştirme prosedürünü uygula.
Adım 9:	Eğer hiçbir iyileştirme gerçekleştirilemiyorsa prosedürü sonlandır.

İyileştirme prosedüründe 8 farklı komşu arama mekanizması kullanılmıştır. Bunlar *RotaTaşıma*, *Insert (0,0)* (Aynı rota içerisine ekleme), *Insert (1,0)* (Aynı tesis farklı rota içerisine ekleme) ve *Insert (1,1)* (Farklı tesislerdeki rotalar için ekleme), *Swap(0,0)* (Aynı rota içerisinde yer değiştirme), *Swap(1,0)* (Aynı tesis farklı rota

içerisinden yer değiştirme), *Swap*(1,1) (Farklı tesislerdeki rotalar için yer değiştirme) ve *RotaBirleştirme*'dir. İyileştirme prosedürüne herhangi bir rotanın bir tesisten diğer bir tesise taşınması ile başlanmaktadır (Adım 1). Bu prosedürün adımları Çizelge 3.7'de anlatılmaktadır.

Çizelge 3.7. ÇT_TA_BÜDÇ Problemi İçin Rota Taşıma Prosedürü

Prosedür:	Rota Taşıma Prosedürü
Girdi:	$S_{Baş(i)}$: ÇTBÜDÇ probleminin BaşUygunÇözümü
Çıktı:	S_{Yeni} : İyileştirilmiş ÇTBÜDÇ problemi Çözümü
Adım 1:	$EnBüyükD_{max}$ 'in elde edildiği tesisten rassal olarak bir rota seç.
Adım 2:	D_{max} değeri en küçük olan tesisi belirle.
Adım 3:	Seçilen rota için atanabileceği yeni tesis için Ürün Ömrü kontrolü yap. <ul style="list-style-type: none"> - eğer R_i rotası yeni tesis için ürün ömrü (B) kısıtını sağlıyor ise Adım 4'e git. - değilse sırasıyla tüm atanabileceği tesis alternatifleri için Adım 3'ü tekrarla. <ul style="list-style-type: none"> - eğer R_i rotasının yeni tesis için ürün ömrünü (B) aşmıyor ise Adım 4'e git. - değilse bir sonraki prosedüre geç.
Adım 4:	Seçilen rotayı aday tesisteki tüm alternatif sıralara ekle.
Adım 5:	Bütün alternatifler için Amaç Fonksiyonu Hesaplama prosedürünü uygula S_{Yeni} ve $f(S_{Yeni})$ değerlerini belirle.
Adım 6:	Çözümün iyileşip iyileşmediğini kontrol et. <ul style="list-style-type: none"> - eğer $EnBüyükf(S_{Yeni(i)}) \leq EnBüyükD_{max}$ ise $EnBüyükD_{max} \leftarrow EnBüyükf(S_{Yeni(i)})$ ve Adım 1'e dön. - değilse bir sonraki prosedüre geç.

Rota taşıma prosedürü için ilk olarak $EnBüyükD_{max}$ değerine sahip tesisten rassal olarak bir rota seçilir (Adım 1). Bu adımdan sonra D_{max} değeri en küçük olan tesis belirlenmelidir (Adım 2). İlk adımda seçilen rotanın D_{max} değeri en küçük olan tesise atanıp atanamayacağı kontrol edilir. Eğer ürün ömrü kısıtı sağlanıyorsa mevcut rotanın ilgili tesise atanması sağlanırken, kısıtın sağlanmaması halinde ise alternatif tesislere atama yapıp yapılamayacağı değerlendirilir. Eğer atama yapılamıyorsa bir sonraki prosedüre geçilir (Adım 3). Eğer ürün ömrü kısıtı sağlanıyorsa seçilen rotanın sırasıyla tüm alternatif sıralara eklenerek yeni çözümler oluşturulur (Adım 4). Bütün çözümler için Amaç Fonksiyonu Hesaplama prosedürü uygulanarak $EnBüyükD_{max}$ üzerinde en çok iyileşme sağlayan çözüm belirlenir (Adım 5). Yeni elde edilen D_{max} değeri ile mevcut D_{max} değeri karşılaştırılarak iyileşme olup olmadığı kontrol edilir. Eğer D_{max} iyileşmişse yeni bir rota rassal olarak seçilerek prosedür tekrarlanır. İyileşme olmaması halinde bir sonraki prosedür ile DKA algoritmasına devam edilir (Adım 6).

DKA algoritmasında bir sonraki prosedür ekleme prosedürüdür. Bu prosedür için 3 farklı durum söz konusudur. İlk olarak aynı rota içerisindeki müşterilerden birisinin diğer müşterilerin önüne veya arkasına eklenmesi göz önüne alınmıştır (*Ekleme* (0,0)) (Adım 2). Bu prosedür ayrıntılı olarak Çizelge 3.8'de anlatılmıştır.

Çizelge 3.8. ÇT_TA_BÜDÇ Problemi İçin Ekleme (0,0) Prosedürü

Prosedür:	<i>Ekleme (0,0) Prosedürü</i>
Girdi:	S_{yeni} : Rota Taşıma Prosedüründen Sonraki Çözüm
Çıktı:	S_{yeni}' : <i>Ekleme (0,0) Prosedüründen Sonraki Çözüm</i>
Adım 1:	$EnBüyükD_{max}$ 'in elde edildiği tesisten rassal olarak bir müşteri seç.
Adım 2:	Seçilen müşteriyi o rotada bulunan diğer tüm müşterilerin önüne veya arkasına ekle.
Adım 3:	Ürün ömrü kısıtının sağlanıp sağlanmadığını kontrol et. <ul style="list-style-type: none"> - <i>eğer</i> kısıt sağlanıyorsa <i>Amaç Fonksiyonu Hesaplama</i> prosedürünü uygula. - <i>değilse</i> Adım 1'e dön.
Adım 4:	Çözümde iyileşme olup olmadığını kontrol et. <ul style="list-style-type: none"> - <i>eğer</i> çözümde iyileşme varsa çözümü ve iyileşme miktarını elinde tut. - <i>değilse</i> Adım 1'e dön.
Adım 5:	O tesise atanan tüm müşteriler bitene kadar yukarıdaki adımları tekrarla.
Adım 6:	Çözümün iyileşip iyileşmediğini kontrol et. <ul style="list-style-type: none"> - <i>eğer</i> $EnBüyükf(S') \leq EnBüyükD_{max}$ ise $EnBüyükD_{max} \leftarrow EnBüyükf(S')$ yap ve <i>Rota Taşıma</i> prosedürüne dön. - <i>değilse</i> bir sonraki prosedüre geç.

Ekleme (0,0) için $EnBüyükD_{max}$ 'e sahip olan tesisten rassal olarak bir müşteri seçilir (Adım 1). Bu müşteri atanmış olduğu rotada bulunan tüm müşterilerin önüne veya arkasına eklenerek yeni çözümler elde edilir (Adım 2). Eğer seçilen müşterinin o rotadaki sırası ekleneceği müşterinin sırasından büyük ise o müşterinin önüne, küçük ise arkasına eklenecektir. Burada önemli olan nokta ürün ömrünün aşılmamasıdır. Eklemelere sonra her bir yeni çözüm için *Amaç Fonksiyonu Hesaplama* prosedürü uygulanır (Adım 3). Eğer amaç fonksiyonu değerinde iyileşme sağlanıyorsa bu çözüm hafızada tutulur ve amaç fonksiyonu değerinde en fazla iyileşme sağlayan çözüm belirlenmeye çalışılır (Adım 4). Yeni bir alternatif çözüm için Adım 1'e dönülür (Adım 5). En fazla iyileşmeyi sağlayan yeni çözümün $EnBüyükD_{max}$ 'i azaltıp azaltmadığı kontrol edilir. Eğer $EnBüyükD_{max}$ azalıyorsa yeni çözüm kabul edilir. Bu süreçten sonra bir diğer önemli nokta $EnBüyükD_{max}$ 'e sahip olan tesisin değişip değişmediğidir. Eğer $EnBüyükD_{max}$ 'e sahip olan tesis değiştiyse prosedürler yeni $EnBüyükD_{max}$ 'e sahip tesis için devam ederken, tesisin değişmemesi halinde ise aynı tesis için *Rota Taşıma* prosedürüne dönerek devam edilir. İyileşme olmaması halinde bir sonraki prosedüre (*Ekleme (1,0)*) geçilir (Adım 6).

Çizelge 3.9. ÇT_TA_BÜDÇ Problemi İçin Ekleme (1,0) Prosedürü

Prosedür:	<i>Ekleme</i> (1,0) Prosedürü
Girdi:	S_{yeni} : <i>Ekleme</i> (0,0) Prosedüründen Sonraki Çözüm
Çıktı:	S_{yeni}' : <i>Ekleme</i> (1,0) Prosedüründen Sonraki Çözüm
Adım 1:	$EnBüyükD_{max}$ 'in elde edildiği tesisten rassal olarak bir rota ve o rotadan rassal olarak bir müşteri seç.
Adım 2:	Yine aynı tesisten rassal olarak ilk rotadan farklı başka bir rota daha seç ve ilk rotadaki seçilen müşteriye ikinci rotadaki her müşterinin önüne veya arkasına ekle.
Adım 3:	Her iki rota için de ürün ömrü ve araç kapasitesi kısıtları sağlanıyorsa <i>Amaç Fonksiyonu Hesaplama</i> prosedürünü uygula.
Adım 4:	Çözümde iyileşme olup olmadığını kontrol et. - <i>eğer</i> çözümde iyileşme varsa çözümü ve iyileşme miktarını elinde tut. - <i>değilse</i> Adım 1'e dön.
Adım 5:	O tesise atanan tüm müşteriler bitene kadar yukarıdaki adımları tekrarla.
Adım 6:	Çözümün iyileşip iyileşmediğini kontrol et. - <i>eğer</i> $EnBüyükf(S') \leq EnBüyükD_{max}$ ise $EnBüyükD_{max} \leftarrow EnBüyükf(S')$ yap ve <i>Rota Taşıma</i> prosedürüne dön. - <i>değilse</i> bir sonraki prosedüre geç.

Çizelge 3.9'da görüldüğü gibi *Ekleme* (1,0) prosedüründe aynı tesisteki farklı rotalar için araya ekleme operasyonu yapılmaktadır (Adım 3). Burada $EnBüyükD_{max}$ 'e sahip olan tesisten rassal olarak bir rota ve o rotadan yine rassal olarak bir müşteri seçilir (Adım 1). Daha sonra aynı tesisten yine rassal olarak ve ilk rotadan farklı bir rota daha seçilmektedir (Adım 2). Daha sonra ilk rotadaki müşteri, sırasıyla ikinci rotadaki tüm müşterilerin önüne ve son müşterinin ise arkasına eklenir. Yeni çözümlerin ürün ömrü ve araç kapasitesi kısıtlarını sağlayıp sağlamadığı kontrol edilir. Eğer ürün ömrü ya da araç kapasitesinden herhangi biri ya da ikisi de aşıyorsa bu yeni çözümler göz önüne alınmamaktadır. Ürün ömrü ve araç kapasitesi kısıtlarının sağlanması halinde her iki çözüm için de *Amaç Fonksiyonu Hesaplama* prosedürü uygulanır (Adım 3). Eğer amaç fonksiyonu değerinde iyileşme sağlanıyorsa bu çözüm hafızada tutulur ve amaç fonksiyonu değerinde en fazla iyileşme sağlayan çözüm belirlenmeye çalışılır (Adım 4). Yeni bir alternatif çözüm için Adım 1'e dönülür. Bu işleme o tesisteki bütün müşteriler tamamlanana kadar devam edilir ve bu alternatifler içerisinde en fazla iyileştirmeyi sağlayan çözüm belirlenir (Adım 5). Eğer $EnBüyükD_{max}$ azalıyorsa ve yeni durumda $EnBüyükD_{max}$ 'e sahip tesis değişmiyorsa, yeni çözüm kabul edilerek aynı tesiste başka rotalarla iyileştirme aramaya devam edilirken, $EnBüyükD_{max}$ 'e sahip tesisin değişmesi halinde yeni tesisteki rotalar ile *Rota Taşıma* prosedürüne dönerek algoritmaya devam edilirken, iyileşme olmaması halinde bir diğer prosedüre (*Ekleme* (1,1)) geçilir (Adım 6).

Çizelge 3.10. ÇT_TA_BÜDÇ Problemi İçin Ekleme (1,1) Prosedürü

Prosedür:	<i>Ekleme (1,1) Prosedürü</i>
Girdi:	S_{yeni} : <i>Ekleme (1,0) Prosedüründen Sonraki Çözüm</i>
Çıktı:	S_{yeni}' : <i>Ekleme (1,1) Prosedüründen Sonraki Çözüm</i>
Adım 1:	$EnBüyükD_{max}$ 'in elde edildiği tesisten rassal olarak bir rota ve o rotadan rassal olarak bir müşteri seç.
Adım 2:	Rassal olarak tesislerden birini ve seçilen tesisten bir müşteriyi seç.
Adım 3:	İlk tesisten seçilen müşteriyi, ikinci tesisten seçilen müşterinin olduğu rotadaki her sıraya ekle.
Adım 4:	Her iki rota için de ürün ömrü ve kapasite kısıtlarını kontrol et. <ul style="list-style-type: none"> - <i>eğer</i> kısıtlar sağlanıyor <i>ise</i> her iki tesis için de <i>Amaç Fonksiyonu Hesaplama</i> prosedürünü uygula. - <i>değilse</i> Adım 1'e dön.
Adım 5:	Çözümde iyileşme olup olmadığını kontrol et. <ul style="list-style-type: none"> - <i>eğer</i> çözümde iyileşme varsa çözümü ve iyileşme miktarını elinde tut. - <i>değilse</i> Adım 1'e dön.
Adım 6:	İlk tesise atanan tüm müşteriler bitene kadar yukarıdaki adımları tekrarla.
Adım 7:	Çözümün iyileşip iyileşmediğini kontrol et. <ul style="list-style-type: none"> - <i>eğer</i> $EnBüyükf(S') \leq EnBüyükD_{max}$ <i>ve</i> $EnBüyükf(S') \leq EskiEnBüyükD_{max}$ <i>ise</i> $EnBüyükD_{max} \leftarrow EnBüyükf(S')$ <i>ve</i> <i>Rota Taşıma</i> prosedürüne dön. - <i>değilse</i> bir sonraki prosedüre geç.

Eklemenin son prosedürü olan *Insert (1,1)*'de farklı iki tesisteki rotalar göz önüne alınmaktadır ve bu prosedürün adımları Çizelge 3.10'da ayrıntılı olarak açıklanmıştır (Adım 4). Burada $EnBüyükD_{max}$ 'e sahip olan tesisten rassal olarak bir rota ve o rotadan yine rassal olarak bir müşteri seçilir (Adım 1). Daha sonra tesisler içerisinde ilk seçilen tesisten farklı bir tesis yine rassal olarak belirlenir (Adım 2). İlk rotadaki müşteri, sırasıyla ikinci rotadaki tüm müşterilerin önüne, son müşterinin ise arkasına eklenerek yeni çözümler oluşturulur (Adım 3). Ürün ömrü ve araç kapasitesi kısıtlarının sağlanması halinde her iki tesis için *Amaç Fonksiyonu Hesaplama* prosedürü uygulanır (Adım 4). Birinci tesiste yani $EnBüyükD_{max}$ 'e sahip olan tesisin D_{max} değerinde herhangi bir iyileşme söz konusu ise o çözüm hafızada tutulur (Adım 5). Diğer tüm alternatiflerin değerlendirilebilmesi için Adım 1'e dönülür. Bu işleme o tesisteki bütün müşteriler tamamlanana kadar devam edilir ve bu alternatifler içerisinde en fazla iyileştirmeyi sağlayan çözüm belirlenir (Adım 6). Eğer $EnBüyükD_{max}$ azalıyorsa, bakılması gereken ikinci nokta ise diğer tesisin D_{max} değerinin daha önce elde edilen $EnBüyükD_{max}$ değerini aşmıyor olmasıdır. Bu iki durumun da sağlanması halinde yeni elde edilen çözüm mevcut çözüm olarak kabul edilerek *Rota Taşıma* prosedürüne dönülürken, iyileşme olmaması halinde bir sonraki prosedürle (*Swap 0,0*) algoritmaya devam edilir (Adım 7).

Ekleme prosedürlerinden sonraki prosedür yer değiştirme prosedürüdür. Bu prosedür de ekleme prosedürüne benzer olarak 3 farklı şekilde ele alınmıştır.

Çizelge 3.11. ÇT_TA_BÜDÇ Problemi İçin Yer Değiştirme (0,0) Prosedürü

Prosedür:	<i>Yer Değiştirme</i> (0,0) Prosedürü
Girdi:	S_{Yeni} : Ekleme (1,1) Prosedüründen Sonraki ÇTBÜDÇ Problemi Çözümü
Çıktı:	S_{Yeni}' : <i>Yer Değiştirme</i> (0,0) Prosedüründen Sonraki Çözüm
Adım 1:	$EnBüyükD_{max}$ 'in elde edildiği tesisten rassal bir müşteri seç.
Adım 2:	Seçilen müşteriyi o rotada bulunan diğer tüm müşterilerle yer değiştir.
Adım 3:	Ürün ömrü kısıtının sağlanıp sağlanmadığını kontrol et. - eğer kısıt sağlanıyorsa <i>Amaç Fonksiyonu Hesaplama</i> prosedürünü uygula. - değilse Adım 1'e dön.
Adım 4:	Çözümde iyileşme olup olmadığını kontrol et. - eğer çözümde iyileşme varsa çözümü ve iyileşme miktarını elinde tut. - değilse Adım 1'e dön.
Adım 5:	O tesise atanan tüm müşteriler bitene kadar yukarıdaki adımları tekrarla.
Adım 6:	Çözümün iyileşip iyileşmediğini kontrol et. - eğer $EnBüyükf(S') \leq EnBüyükD_{max}$ ise $EnBüyükD_{max} \leftarrow EnBüyükf(S')$ yap ve <i>Rota Taşıma</i> prosedürüne dön. - değilse bir sonraki prosedüre geç.

DKA algoritmasındaki yer değiştirme prosedürünün ilki, aynı rota içerisindeki müşterilerin yer değiştirilmesidir. *Yer Değiştirme* (0,0) olarak adlandırılan prosedür Çizelge 3.11'de ayrıntılı bir şekilde gösterilmiştir (Adım 5). Burada $EnBüyükD_{max}$ değerine sahip olan tesisten rassal olarak bir rota ve seçilen rotadan yine rassal olarak bir müşteri seçilir (Adım 1). Bu müşteri atanmış olduğu rotada bulunan tüm müşterilerle yer değiştirilerek yeni çözümler elde edilir (Adım 2). Ürün ömrü kısıtının sağlandığı her çözüm için *Amaç Fonksiyonu Hesaplama* prosedürleri uygulanarak yeni çözümün $EnBüyükD_{max}$ 'i azaltıp azaltmadığı kontrol edilir (Adım 3). Yeni elde edilen çözümler içerisinde $EnBüyükD_{max}$ değerinde en fazla iyileşme sağlayan çözüm hafızada tutulur (Adım 4). Tüm yer değiştirme alternatifleri için Adım 1'e dönülerek prosedüre devam edilir ve bu alternatifler içerisinde en fazla iyileştirmeyi sağlayan çözüm belirlenir (Adım 5). Tüm alternatifler içerisinde $EnBüyükD_{max}$ 'i en fazla iyileştiren çözüm en iyi çözüm olarak kabul edilir ve *Rota Taşıma* prosedürüne dönerek DKA Algoritmasına devam edilir. Eğer $EnBüyükD_{max}$ 'te herhangi bir iyileşme meydana gelmiyorsa algoritma Swap (1,0) prosedürü ile devam eder (Adım 6).

Çizelge 3.12'de aynı tesisteki farklı rotalara ait müşterilerin yer değiştirilmesi prosedürü (*Yer Değiştirme* (1,0)) ayrıntılı şekilde anlatılmıştır (Adım 6).

Çizelge 3.12. ÇT_TA_BÜDÇ Problemi İçin Yer Değiştirme (1,0) Prosedürü

Prosedür:	<i>Yer Değiştirme</i> (1,0) Prosedürü
Girdi:	S_{yeni} : <i>Yer Değiştirme</i> (0,0) Prosedüründen Sonraki Çözüm
Çıktı:	S_{yeni}' : <i>Yer Değiştirme</i> (1,0) Prosedüründen Sonraki Çözüm
Adım 1:	$EnBüyükD_{max}$ 'in elde edildiği tesisten rassal olarak bir rota ve o rotadan rassal olarak bir müşteri seç.
Adım 2:	Yine aynı tesisten rassal olarak ve ilk rotadan farklı bir rota seç.
Adım 3:	İlk rotadaki müşteriyi ikinci rotadaki her müşteri ile yer değiştir.
Adım 4:	Ürün ömrü ve kapasite kısıtının sağlanıp sağlanmadığını kontrol et. <ul style="list-style-type: none"> - <i>eğer</i> kısıtlar sağlanıyorsa <i>Amaç Fonksiyonu Hesaplama</i> prosedürünü uygula. - <i>değilse</i> Adım 1'e dön.
Adım 5:	Çözümde iyileşme olup olmadığını kontrol et. <ul style="list-style-type: none"> - <i>eğer</i> çözümde iyileşme varsa çözümü ve iyileşme miktarını elinde tut. - <i>değilse</i> Adım 1'e dön.
Adım 6:	O tesise atanan tüm müşteriler bitene kadar yukarıdaki adımları tekrarla.
Adım 7:	Çözümün iyileşip iyileşmediğini kontrol et. <ul style="list-style-type: none"> - <i>eğer</i> $EnBüyükf(S') \leq EnBüyükD_{max}$ ise $EnBüyükD_{max} \leftarrow EnBüyükf(S')$ yap ve <i>Rota Taşıma</i> prosedürüne dön. - <i>değilse</i> bir sonraki prosedüre geç.

Yer Değiştirme (1,0) prosedürünün ilk adımında yine diğer prosedürlerde olduğu gibi $EnBüyükD_{max}$ 'e sahip olan tesisten rassal olarak bir rota ve seçilen rotadan yine rassal olarak bir müşteri seçilir (Adım 1). Sonrasında aynı tesisten yine rassal olarak ilk seçilen rotadan farklı bir rota daha seçilmektedir (Adım 2). Daha sonra ilk rotadaki müşteri ile ikinci rotadaki tüm müşteriler sırasıyla yer değiştirilir (Adım 3). Ürün ömrü ve araç kapasitesinin aşılması halinde yer değiştirme gerçekleştirilmezken kısıtların sağlanması halinde *Amaç Fonksiyonu Hesaplama* prosedürü uygulanmaktadır (Adım 4). Yeni çözüm için elde edilen D_{max} değerinin iyileşip iyileşmediği kontrol edilir. Eğer çözümde iyileşme varsa yeni çözüm ve iyileşme miktarı hafızada tutulurken iyileşme olmaması halinde alternatif çözümler için Adım 1'e dönlülmektedir (Adım 5). Bu işlem o tesisteki tüm müşteriler için tekrarlanacaktır. Tüm çözümler değerlendirildikten sonra eğer çözümde iyileşme varsa hafızadaki çözüm mevcut çözüm olarak kabul edilerek *Rota Taşıma* prosedürüne dönlür. Eğer iyileştirme olmadıysa bir sonraki prosedüre geçilir (Adım 6).

DKA algoritmasınının 7.adımı *Yer Değiştirme* (1,1) prosedürüdür (Adım 7). Yer değiştirme prosedürünün son basamağı olan bu prosedürde farklı iki tesisteki rotaların yer değiştirilmesi (*Yer Değiştirme* (1,1)) göz önüne alınmaktadır. Prosedürün adımları Çizelge 3.13'te verilmiştir.

Çizelge 3.13. ÇT_TA_BÜDÇ Problemi İçin Swap (1,1) Prosedürü

Prosedür:	<i>Yer Değiştirme</i> (1,1) Prosedürü
Girdi:	S_{yeni} : <i>Yer Değiştirme</i> (1,0) Prosedüründen Sonraki Çözüm
Çıktı:	S_{yeni}' : <i>Yer Değiştirme</i> (1,1) Prosedüründen Sonraki Çözüm
Adım 1:	$EnBüyükD_{max}$ 'in elde edildiği tesisten rassal olarak bir rota ve o rotadan rassal olarak bir müşteri seç.
Adım 2:	Diğer tesislerden birisini rassal olarak seç ve yine rassal olarak o tesisten bir rota seç.
Adım 3:	İlk tesisten seçilen müşteriyi, ikinci tesisten seçilen müşterinin olduğu rotadaki her müşteriyle yer değiştir.
Adım 4:	Her iki rota için de ürün ömrü ve kapasite kısıtlarını kontrol et. <ul style="list-style-type: none"> - <i>eğer</i> kısıtlar sağlanıyor <i>ise</i> her iki tesis için de <i>Amaç Fonksiyonu Hesaplama</i> prosedürünü uygula. - <i>değilse</i> Adım 1'e dön.
Adım 5:	Çözümde iyileşme olup olmadığını kontrol et. <ul style="list-style-type: none"> - <i>eğer</i> çözümde iyileşme varsa çözümü ve iyileşme miktarını elinde tut. - <i>değilse</i> Adım 1'e dön.
Adım 6:	İlk tesise atanan tüm müşteriler bitene kadar yukarıdaki adımları tekrarla.
Adım 7:	Çözümün iyileşip iyileşmediğini kontrol et. <ul style="list-style-type: none"> - <i>eğer</i> $f(S') \leq EnBüyükD_{max}$ ve $EnBüyükf(S') \leq EskiEnBüyükD_{max}$ <i>ise</i> $EnBüyükD_{max} \leftarrow EnBüyükf(S')$ yap ve <i>Rota Taşıma</i> prosedürüne dön. - <i>değilse</i> sonraki prosedüre geç.

Burada $EnBüyükD_{max}$ 'e sahip olan tesisten rassal olarak bir rota ve yine rassal olarak bir müşteri seçilir (Adım 1). Daha sonra tesisler içerisinde farklı bir tesis yine rassal olarak belirlenir (Adım 2). Daha sonra seçilen ikinci tesisten yine rassal olarak bir rota seçilmektedir. İlk rotadaki müşteri ile ikinci rotadaki tüm müşteriler sırasıyla yer değiştirilir (Adım 3). Bir sonraki adımda ürün ömrü ve araç kapasitesinin kontrolleri gerçekleştirilirken, kısıtların aşılması halinde yer değiştirme gerçekleştirilemez. Kısıtlar sağlanıyorsa seçilen iki tesis için *Amaç Fonksiyonu Hesaplama* prosedürleri uygulanır (Adım 4). Elde edilen yeni değerlerde iyileşme meydana geldiyse yeni çözüm ve iyileşme miktarı elde tutulurken iyileşme yoksa diğer alternatifler değerlendirilmeye üzere Adım 1'e dönlür (Adım 5). Bu adımlar ilk tesise atanan tüm müşteriler için tekrarlanır (Adım 6). Tüm müşteriler için adımlar tekrarlandıktan sonra *eğer* $EnBüyükD_{max}$ değeri azalıyorsa, bakılması gereken ikinci nokta ise diğer tesisin D_{max} değerinin daha önce elde edilen $EnBüyükD_{max}$ değerini aşmıyor olmasıdır. Bu durum sağlanıyorsa hafızadaki çözüm uygulanarak *Rota Taşıma* prosedürüne dönlür, *eğer* iyileşme mevcut değilse bir sonraki prosedüre geçilerek algoritma devam ettirilir (Adım 7).

DKA algoritmasının son prosedürü başlangıç çözümü elde edilirken kullanılan revize C&W algoritmasının son adımı olan *Birleştirme* prosedürüdür (Adım 8). Prosedürün adımları Çizelge 3.14'te ayrıntılı olarak gösterilmektedir.

Çizelge 3.14. ÇT_TA_BÜDÇ Problemi İçin Birleştirme Prosedürü

Prosedür:	<i>Birleştirme</i> Prosedürü
Girdi:	S_{yeni} : <i>Yer Değiştirme</i> (1,1) Prosedüründen Sonraki Çözüm
Çıktı:	S_{yeni}' <i>Birleştirme</i> Prosedüründen Sonraki Çözüm
Adım 1:	Tüm rota çiftlerinden sıradaki rota çifti için kapasite kontrolü yap. - <i>eğer</i> $Q \geq d(R_i) + d(R_j)$ <i>ise</i> Adım 2'ye git. (d : talep) - <i>değilse</i> sıradaki rota çifti ile Adım 1'i tekrarla.
Adım 2:	Mevcut rota çifti için Ürün Ömrü kontrolü yap. - <i>eğer</i> R_i ve R_j rota uzunluklarının birleştirilmesi ürün ömrünü (B) aşmıyor <i>ise</i> Adım 3'e git. - <i>değilse</i> sıradaki rota çifti ile Adım 1'e git.
Adım 3:	Mevcut rota çifti için rota uzunluğunun durumunu kontrol et. - <i>eğer</i> R_i ve R_j rotalarının k . tesiste birleştirilmesi toplam rota uzunluğunu azaltıyor <i>ise</i> R_i , R_j ve k bilgilerini elinde tut. - <i>değilse</i> sıradaki rota çifti ile Adım 1'e git.
Adım 4:	Bütün birleştirme alternatifleri içerisinde rotada en çok iyileştirmeyi sağlayan çifti belirle ve k . tesiste birleştir.
Adım 5:	Her tesis için Amaç Fonksiyonu Hesaplama prosedürünü uygula S_{yeni} ve $f(S_{yeni})$ değerlerini belirle.
Adım 6:	Tüm rota çiftleri için birleştirme alternatiflerinin hepsi tamamlanana kadar Adım 1'e git.
Adım 7:	Çözümün iyileşip iyileşmediğini kontrol et. - <i>eğer</i> $EnBüyükf(S_{yeni(i)}) \leq EnBüyükD_{max}$ <i>ise</i> $EnBüyükD_{max} \leftarrow EnBüyükf(S_{yeni(i)})$ ve Rota Taşıma prosedürüne dön. - <i>değilse</i> rotaların birleştirilmesini iptal et.

Bu adımda sırasıyla tüm rota çiftleri göz önüne alınır. Birleştirilmesi planlanan iki rota için müşteri taleplerinin toplamının araç kapasitesini aşmaması gerekmektedir (Adım 1). Eğer bu iki rotanın birleştirilmesi araç kapasitesini aşıyorsa bu birleştirme işlemi yapılamamaktadır. Eğer kapasite aşılmıyorsa bir diğer dikkat edilmesi gereken husus da ürün ömrü kısıtıdır (Adım 2). Bu adımda tüm rota çiftleri için her iki rotanın tek bir rota olarak birleştirilmesi ile elde edilen rota uzunluğunun, $d(R_{(j+1)k}, T_i)$ değerinden farkı ürün ömrünü aşmamalıdır. Eğer ürün ömrü aşılmıyorsa yeni elde edilen rota uzunluğunun ayrı iki rota iken elde edilen rota uzunluğundan kısa olup olmadığı kontrol edilir (Adım 3). Yeni oluşturulan rota uzunluğu hafızadaki çözümle karşılaştırılarak daha iyiyse hafızada tutulup, tüm rota çiftleri için hesaplama bittiği zaman en fazla iyileştirmeyi sağlayan rota çifti birleştirilir (Adım 4). Her tesis için *Amaç Fonksiyonu Hesaplama* prosedürünü uygulanarak en iyi çözüm ve iyileşme miktarı hafızada tutulur (Adım 5). Tüm birleştirme alternatifleri için aynı adımlar devam ettirilir (Adım 6). Yeni bulunan sonuçlar içerisinde mevcut çözümden daha iyi bir sonuç elde edilmişse yani $EnBüyükD_{max}$ değerinde iyileşme sağlanmışsa rota birleştirilmesi gerçekleştirilerek *Rota Taşıma* prosedürüne dönmektedir. Eğer çözümde iyileşme meydana gelmiyorsa algoritma diğer alternatif birleştirmeler ile değerlendirilmektedir (Adım 6).

DKA algoritmasının ilk iterasyonu tamamlandıktan sonra başlangıç çözümü olarak kullanılan revize C&W algoritması yerine farklı rassal başlangıç çözümleri elde edilerek algoritmanın etkinliği artırılmaya çalışılmıştır. Başlangıçta elde edilen sonuç hafızada tutulurken, tüm müşteriler rassal olarak bir depoya atanarak yeni başlangıç çözümleri ile problem tekrar çözülerek yeni çözümler ile şu ana kadar elde edilmiş olan mevcut çözümün iyileşip iyileşmediği kontrol edilir. Başlangıç atamaları rassal olarak oluşturulduktan sonra tıpkı C&W’da olduğu gibi her müşteri tek bir rota oluşturacak şekilde rotalar oluşturulur. Her tesisin D_{max} değeri *Amaç Fonksiyonu Hesaplama* prosedürü kullanılarak belirlenir. DKA algoritmasındaki mevcut komşu arama operatörleri ile çözümün iyileşip iyileşmediği kontrol edilir. Yeni elde edilen D_{max} değeri hafızadaki değerden daha düşük ise yeni çözüm hafızada tutulurken çözümün daha kötü sonuç elde etmesi halinde hafızadaki çözüm muhafaza edilmeye devam eder. Tüm bu işlemler küçük ve orta boyutlu problemler için 20 kere tekrar edilirken, büyük boyutlu problemler için 5 kere tekrarlanır.

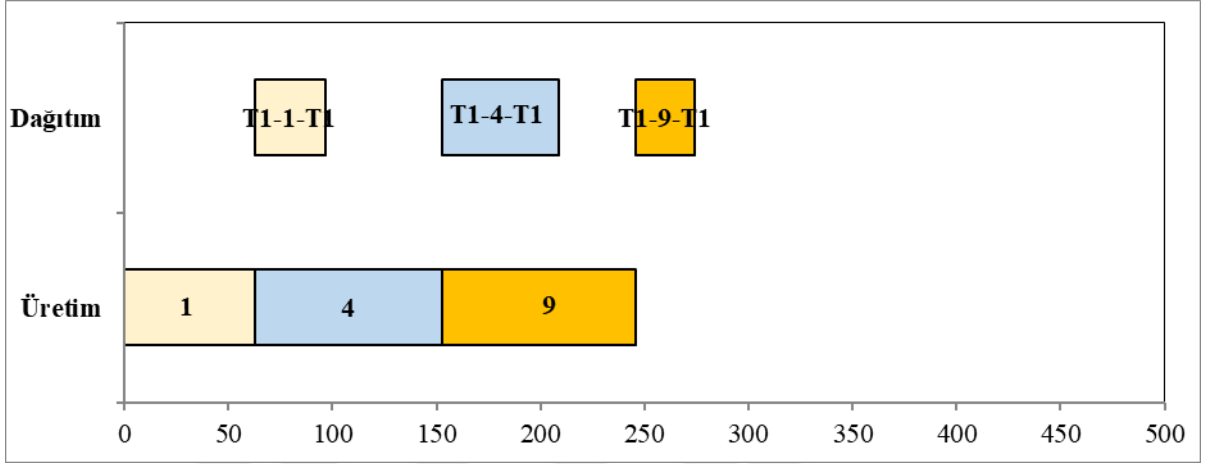
3.3.2. DKA algoritmasının örnek probleme uygulanması

Bu bölümde daha önce matematiksel model ile çözümü örnek olarak verilen test probleminin önerilen DKA algoritması ile çözümü ele alınmıştır. Buna göre algoritma ilk olarak C&W algoritması ile başlayacaktır. C&W algoritmasında ilk olarak tüm müşteriler kendisine en yakın tesise atanır ve her biri tek bir rota olacak şekilde ilk çözümler oluşturulur. Çizelge 3.15’te bu ilk atamalar gösterilmiştir.

Çizelge 3.15. C&W Başlangıç Atamaları ve Rotaları

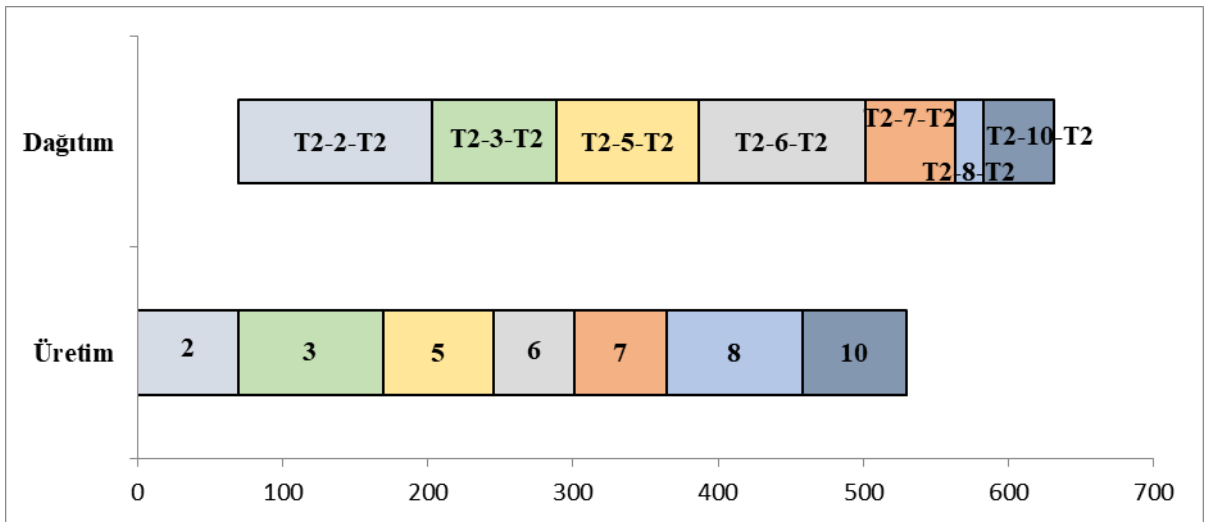
Tesis	Rota	Üretim	Dağıtım
T1	T1-1-T1	62.5	33.94
	T1-4-T1	90	56.36
	T1-9-T1	93.5	28
T2	T2-2-T2	69.5	133.78
	T2-3-T2	99.5	85.6
	T2-5-T2	76.5	97.84
	T2-6-T2	55.5	114.28
	T2-7-T2	63.5	62.52
	T2-8-T2	93.5	18.98
	T2-10-T2	72	48.38

Çizelge 3.15'te görüldüğü üzere, toplam 10 adet müşteriden 1-4 ve 9 numaralı müşteriler T1'e ve 2-3-5-6-7-8 ve 10 numaralı müşteriler ise T2 tesisine atanmıştır. Bu atamalar sırasıyla her müşteriye gidiş geliş olacak şekilde rotalara ayrılır. Buna göre T1 tesisi için 3 ve T2 tesisi için de 7 ayrı rota mevcuttur. Her iki tesisin de Gantt şeması sırasıyla Şekil 3.3 ve Şekil 3.4'te verilmiştir.



Şekil 3.3. T1 Tesisinin İlk Atamaya Göre Elde Edilen Gantt Şeması

T1 tesisinin ilk atama sonuçlarına göre elde edilen Gantt şeması Şekil 3.3'te verilmiştir. Şekilde görüldüğü üzere her 3 müşterinin de üretim süreleri uzun, dağıtım sıraları kısadır. Bu nedenle araç herhangi bir müşterinin ürün talebini dağıttıktan sonra tesise dönmekte ve diğer müşterinin talebinin üretiminin tamamlanmasını beklemektedir. T1 tesisi için D_{max} değeri 274 birim olarak hesaplanmıştır. Şekil 3.4'te ise T2 tesisi için ilk atama sonucunda elde edilen Gantt şeması verilmiştir.



Şekil 3.4. T2 Tesisinin İlk Atamaya Göre Elde Edilen Gantt Şeması

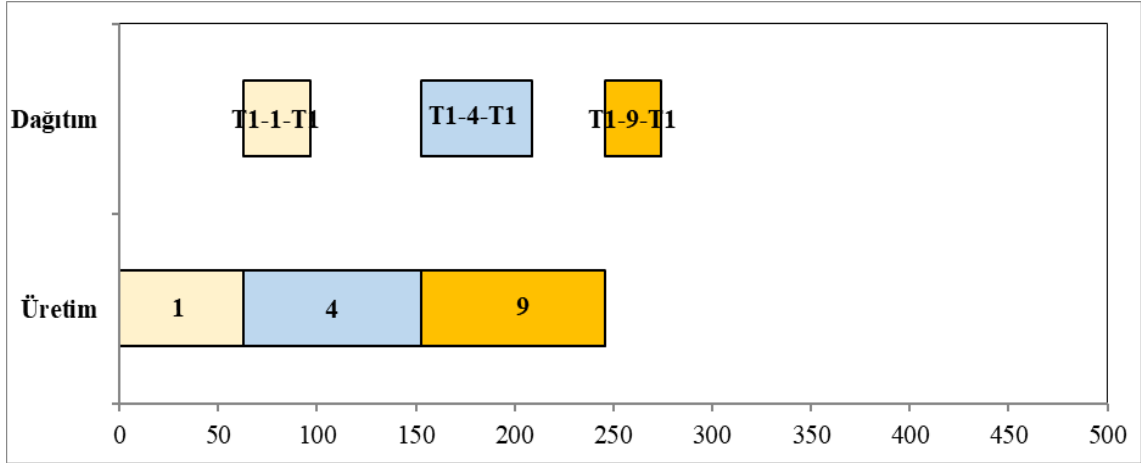
T2 tesisine atanmış olan tüm müşterilerin sırasıyla taleplerinin üretilmesi ve dağıtılması işlemleri Şekil 3.4'te gösterilmiştir. Bu durumda T2 tesisine atanan tüm müşterilerin taleplerinin üretilmesi T1 tesisinin aksine daha kısa sürmektedir. Bu durumda araç üretilen müşteri taleplerini teslim ederken bir sonraki müşterinin talebinin üretimi tamamlanmış olmaktadır. Araç tesise gelir gelmez yeni müşterinin taleplerini dağıtmak için yola çıkmaktadır. T2 tesisinde tüm müşterilerin talebini üretmek için gereken zaman 530 zaman birimi iken, en son müşteri olan 10 numaralı müşterinin talebinin dağıtım işlemi tamamlanıp aracın tesise geri döndüğü zaman 630.88 zaman birimidir. Bu da T2 tesisi için D_{max} değerine eşittir. Bu durumda T1 ve T2 tesisleri göz önüne alındığında $EnBüyükD_{max}$ değeri 630.88'dir. Daha sonraki işlemler bu değeri azaltmak için gerçekleştirilecektir.

C&W algoritmasında her tesis kendisine en yakın tesise atanmakta ve sonrasında ilk rotalar oluşturulmaktadır. Bir sonraki adımda ise bu müşterilerin birleştirilmesi ve taleplerinin ortak üretilip dağıtılması işlemi gelmektedir. Çizelge 3.16'da C&W algoritması sonucunda elde edilen sıralamalar gösterilmiştir ve T1 ve T2 için oluşturulan Gantt şemaları sırasıyla Şekil 3.5 ve 3.6'da verilmiştir.

Çizelge 3.16. C&W Sonrası Rotalar

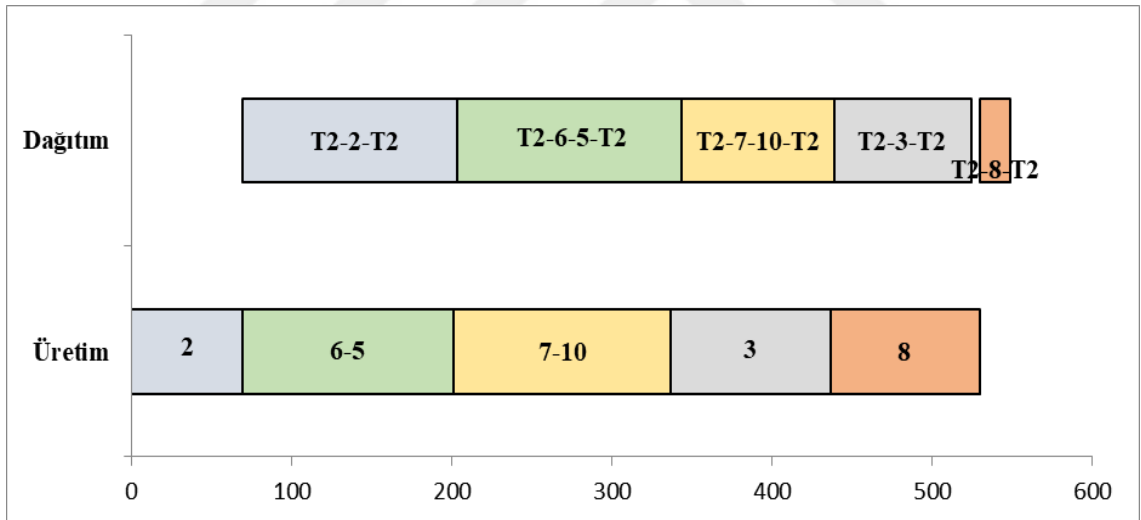
Tesis	Rota	Üretim Süresi	Dağıtım Süresi
T1	T1-1-T1	62.5	33.94
	T1-4-T1	90	56.36
	T1-9-T1	93.5	28
T2	T2-2-T2	69.5	133.78
	T2-6-5-T2	132	140.47
	T2-7-10-T2	135.5	95.05
	T2-3-T2	99.5	85.6
	T2-8-T2	93.5	18.98

Çizelge 3.16'ya göre T1 tesisinde rotalarda ve sıralamada herhangi bir değişiklik söz konusu değil iken, T2 tesisi için hem rotalar hem de sıralamalar değişmiştir. 5 ve 6 numaralı müşteriler aynı rotada birleştirilmiş ve üretim süresi 132 birim, dağıtım süresi de 140.47 olarak güncellenmiştir. Benzer şekilde 7 ve 10 numaralı müşteriler de birlikte işlem göreceklendir.



Şekil 3.5. T1 Tesisinin C&W Sonrası Elde Edilen Gantt Şeması

Şekil 3.5'te T1 tesisinin C&W sonrası elde edilen Gantt şeması yer almaktadır. Şekilde de görüldüğü gibi C&W algoritmasından sonra T1 tesisindeki atamalar ve rotalar ile ilgili herhangi bir değişiklik olmamıştır. Başlangıç çözümünde T1 tesisinin D_{max} değeri 274 zaman birimidir.

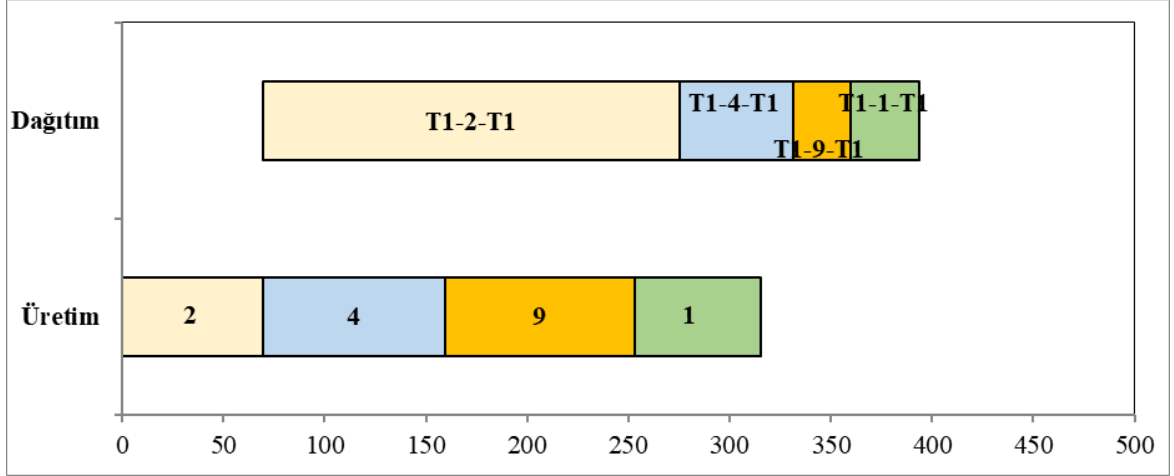


Şekil 3.6. T2 Tesisinin C&W Sonrası Elde Edilen Gantt Şeması

Şekil 3.6'da T2 tesisinin C&W sonrası elde edilen Gantt şeması yer almaktadır. Şekilde de görüldüğü gibi C&W algoritmasından sonra 2 rota kapatılmış ve toplam rota sayısı 5'e düşmüştür. 8 numaralı müşteriye kadar tüm talepler üretilmiş ve çok kısa süre sonra araç tesise dönerek diğer dağıtımlara başlamıştır. Ancak 8 numaralı müşterinin üretimi tamamlanmadan araç tesise dönmüş durumdadır. Bu nedenle araç üretim

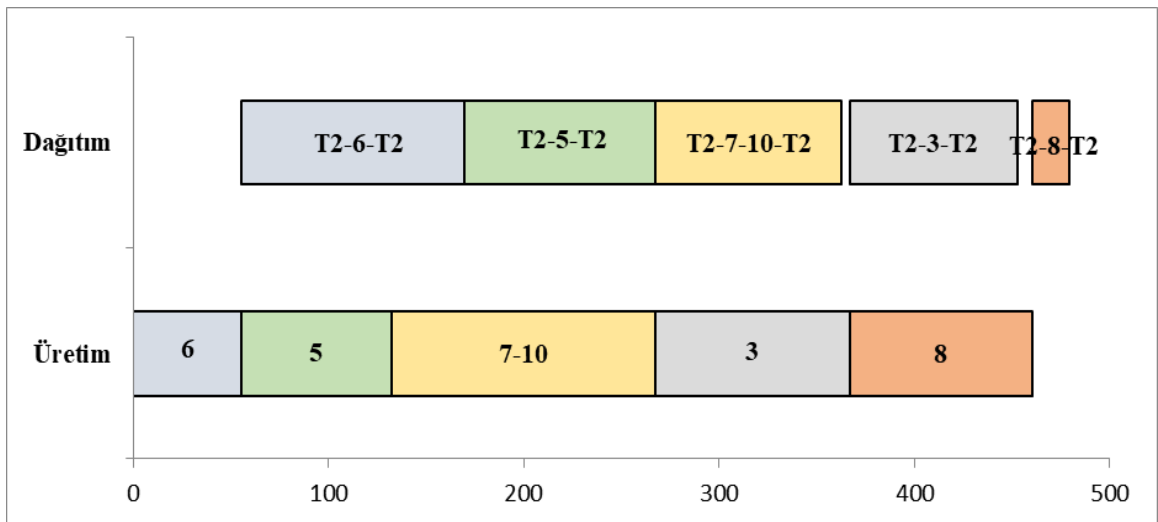
tamamlanana kadar bir müddet tesiste beklemektedir. Başlangıç çözümünde T2 tesisinin D_{max} değeri 630.88 zaman biriminden 548.98 zaman birimine iyileştirilmiştir.

C&W başlangıç algoritması ile DKA algoritmasına başladıktan sonra elde edilen D_{max} değeri komşu arama operatörleri ile iyileştirilmeye devam edilir. Algoritmanın tüm adımları tamamlandıktan sonra elde edilen D_{max} değerleri ve çözümler aşağıda gösterilmektedir.



Şekil 3.7. T1 Tesisinin DKA Sonrası Elde Edilen Gantt Şeması

Şekil 3.7’de T1 tesisinin DKA ile elde edilen Gantt şeması verilmiştir. Buna göre 2 numaralı müşteri T1 tesisine eklenmiş ve sıralama buna göre oluşturulmuştur. 4 müşterinin hepsi ayrı rotalar olarak oluşturulmuş ve dağıtımı bu şekilde gerçekleştirilmiştir. DKA sonrası T1 tesisinin D_{max} değeri 393.60 zaman birimi olarak hesaplanmıştır.



Şekil 3.8. T2 Tesisinin DKA Sonrası Elde Edilen Gantt Şeması

Şekil 3.8’de T2 tesisinin DKA ile elde edilen Gantt şeması verilmiştir. Buna göre 6 adet müşteri 5 rotada gruplanarak üretim ve dağıtım işlemleri gerçekleştirilmiştir. 3 ve 8 numaralı müşterilerin taleplerinin üretimi araç tesise döndüğü zaman henüz tamamlanmadığı için araç tesiste bir müddet beklemektedir. DKA sonrası T2 tesisinin D_{max} değeri 479.48 zaman birimi olarak hesaplanmıştır. C&W ile elde edilen başlangıç çözümünde bu değer $EnBüyükD_{max}$ değeri 548.98 iken komşu arama operatörleri ile yapılan iyileştirmelerle $EnBüyükD_{max}$ değeri 479.48’e düşürülmüştür.

3.4. ÇT_TA_BÜDÇ Problemi İçin Deney Tasarımı

Bir önceki bölümde önerilen matematiksel modelin etkinliğini araştırmak amacıyla Geismar ve ark. tarafından 2008 yılında üretilmiş olan test problemlerindeki yapıya benzer bir yapı kullanılarak test problemleri üretilmiştir. Bu test problemlerinde 2 farklı araç kapasitesi mevcuttur ($Q = 300, 600$). Ayrıca ürün ömrü için de yine $B = 300$ ve 600 olmak üzere 2 farklı durum söz konusudur. Üretim hızı $r = 1, 2, 3$ olmak üzere 3 farklı şekilde ele alınmıştır.

Müşteri taleplerinin (100 – 200) ve (100 – 300) arasında rassal oluşturulduğu 2 farklı durum mevcuttur. Müşterilerin konumları belirlenirken 3 farklı koordinat düzlemi kullanılmıştır. Bunlar; (-100×100) , (-150×150) ve (-200×200) koordinat düzlemidir.

Müşteri sayıları küçük boyutlu problemler için 10; orta boyutlu problemler için 20, 30, 40, 50 ve büyük boyutlu problemler içinse 100 olarak belirlenmiştir. Küçük boyutlu problemlerde tesis sayısı 2, orta boyutlu problemlerde tesis sayısı 2 ve 3, büyük boyutlu problemler içinse tesis sayısı 3, 4 ve 5 olarak belirlenmiştir.

Her parametre kombinasyonu için 3 farklı test problemi üretilmiştir. Toplamda 10 müşterilik problemler için $1 \times 2 \times 2 \times 3 \times 2 \times 3 \times 3 = 216$ adet test problemi üretilmiştir. Orta boyutlu problemlerin her biri için 432 adet test problemi oluşturulmuştur. Orta boyutlu problemler toplamda $4 \times 432 = 1728$ adet iken büyük boyutlu problemler içinse bu sayı 648 adettir. Toplamda 2592 adet test problemi üretilmiştir.

3.5. ÇT_TA_BÜDÇ Problemi İçin Deneysel Karşılaştırmalar

Bu bölümde ÇT_TA_BÜDÇ probleminin çözümü için geliştirilmiş olan matematiksel modelin ve DKA algoritmasının sonuçları ele alınmış, elde edilen sonuçlar birbiriyle kıyaslanmıştır.

3.5.1. Önerilen matematiksel modelin performansı

Önerilen matematiksel model “GAMS” ara yüzünde kodlanmış ve matematiksel model çözücüsü olarak “CPLEX 20.1” kullanılmıştır. Bütün koşullarda çözücünün varsayılan parametre seviyeleri kullanılmıştır. Her bir koşul “Intel Xeon (R) E-2236 (12 Core) 3.40 GHz” hızında “32 GB” ara belleğe sahip, “Windows 10” işletim sistemi ile çalışan bilgisayarda gerçekleştirilmiştir. Bütün koşullar 2 saat (7200 saniye) çözüm süresi ile sınırlandırılmıştır. Küçük boyutlu (10 müşteri) 216 adet problemin 74 adedinde optimal sonuca ulaşılırken orta ve büyük boyutlu problemlerin hiçbirinde optimal sonuç elde edilememiştir. Çizelge 3.17’de müşteri taleplerinin durumuna göre elde edilen sonuçlar verilmiştir.

Çizelge 3.17. Farklı Talepler İçin Deneysel Çalışmalar

	Çözüm Durumu	Adet	Ortalama Çözüm Süresi
D (100-200)	Optimal	25	3799
	Uygun Çözüm	822	7200
	Çözüm Bulunamayan	449	7200
	Toplam (Ortalama)	1296	7134
D (100-300)	Optimal	49	2720
	Uygun Çözüm	863	7200
	Çözüm Bulunamayan	384	7200
	Toplam (Ortalama)	1296	7030

Çizelge 3.17’de görüldüğü üzere, çizelgenin ilk satırında gösterilen ve müşteri taleplerinin 100 ile 200 arasında rassal üretildiği durumu gösteren toplam 1296 adet test probleminin 25 adedinde optimal sonuçlara ulaşılmıştır. 822 adet problemde uygun bir çözüm elde edilmişken 449 adet problemde ise belirlenen 7200 saniye süre sınırı içerisinde herhangi bir uygun çözüm bulunamamıştır. Ortalama çözüm süresi 7134 sn’dir. Benzer şekilde müşteri taleplerinin 100 ile 300 arasında rassal üretildiği toplam 1296 adet test probleminin 49 adedinde optimal sonuçlar elde edilmiştir. Optimal çözüm sayısının

bu problem için daha fazla olmasının nedeninin çözüm uzayının daha küçük olmasıyla bağlantılı olduğu söylenebilir. 863 adet problem için uygun çözüm elde edilirken 384 adet problemde çözüm elde edilememiştir.

Çizelge 3.18’de müşteri sayılarına göre deneysel çalışmaların sonuçları gösterilmektedir. Buna göre optimal sonuca ulaşılan problemler yalnızca 10 müşterilik problemler iken, 30 müşterilik problemlerle beraber 7200 saniye süre sınırı içerisinde herhangi bir çözümün bulunamadığı problemler ortaya çıkmaktadır.

Çizelge 3.18. Müşteri Sayılarına Göre Deneysel Çalışmalar

	Müşteri Sayısı	Çözüm Durumu		Toplam	
		Optimal	Uygun Çözüm		Çözüm Bulunamayan
D (100-200)	10	25	83	108	
	20		216	216	
	30		209	7	216
	40		181	35	216
	50		128	88	216
	100		5	319	324
	Toplam	25	822	449	1296
D (100-300)	10	49	59	108	
	20		216	216	
	30		213	3	216
	40		199	17	216
	50		154	62	216
	100		22	302	324
	Toplam	49	863	384	1296

Müşteri sayısı arttıkça problemin çözümünün daha da zorlaştığı çizelgede açıkça görülmektedir. Çözüm uzayının daha küçük olduğu, müşteri talebinin 100 ile 300 arasında rassal değiştiği test problemlerinde diğer problem türüne göre daha fazla uygun çözüm bulunmuştur. Talebin 100 ile 200 arasında olduğu durumda belirlenen süre sınırı içerisinde uygun çözüm bulunamayan problem sayısı 449 adet iken bu sayı talebin 100 ile 300 arasında rassal değiştiği problem türü için 384 adettir.

Matematiksel modellerin CPLEX çözücü ile belirli bir süre sınırı için elde edilen üst sınır (Z^{US}) ile alt sınır (Z^{AS}) değerleri arasındaki sapma değeri Yüzde Sapma Değeri (YSD) olarak adlandırılmaktadır ve her bir test problemi için aşağıdaki eşitlik kullanılarak hesaplanmaktadır.

$$YSD = \left(\frac{Z^{\text{ÜS}} - Z^{\text{AS}}}{Z^{\text{ÜS}}} \right) * 100$$

Bu değer “0”a ne kadar yakın olursa elde edilen sonuç en iyi çözüme o kadar yakın olmaktadır. Optimal veya uygun çözüm bulunan problemler için müşteri sayılarına ve müşteri taleplerine göre elde edilen YSD değerleri Çizelge 3.19’da verilmiştir.

Çizelge 3.19. Müşteri Taleplerine Göre YSD

	Müşteri Sayısı	YSD*	Ortalama Çözüm Süresi
D (100-200)	10	33.22	6412
	20	88.59	7200
	30	94.11	7200
	40	96.57	7200
	50	97.29	7200
	100	99.07	7200
	Ortalama	85.97	7134
D (100-300)	10	25.34	5167
	20	82.68	7200
	30	91.90	7200
	40	94.59	7200
	50	94.88	7200
	100	98.76	7200
	Ortalama	83.09	7030

*: Ortalama değerler üst sınır değerinin elde edildiği test problemleri kullanılarak elde edilmiştir.

Çizelge 3.19’da görüldüğü üzere her iki problem tipi için de müşteri sayısı arttıkça YSD değeri de artmaktadır. Özellikle büyük boyutlu problemlerde uygun çözüm bulabilmek oldukça zordur. YSD değeri müşteri talebinin 100-200 olduğu durumda ortalama %85.97 iken 100-300 olduğu durumda ise %83.09 olarak belirlenmiştir. Müşteri talebinin 100-300 arasında olduğu durumda araç kapasitesi ve ürün ömrü kısıtlarından dolayı çözüm uzayı daralmakta ve bu da daha kolay çözüm elde edilmesine neden olmaktadır. YSD değerlerinin 10 müşterilik test problemleri için bile bu kadar yüksek olması problemin çözümünün zorluğunu açıkça göstermektedir.

Test problemlerindeki diğer parametrelerin değişimine göre elde edilen sonuçlar aşağıdaki çizelgelerde ayrıntılı şekilde ele alınmıştır. Buna göre ilk olarak tesis sayısının değişiminin sonuçlar üzerindeki etkisi Çizelge 3.20’de gösterilmektedir.

Çizelge 3.20. Tesis Sayısına Göre Deneysel Sonuçlar

Tesis Sayısı	Çözüm Durumu			Toplam	Ortalama YSD*	Ortalama Çözüm Süresi	
	Optimal	Uygun Çözüm	Çözüm Bulunamayan				
D (100-200)	2	25	481	34	540	81.13	7042
	3	-	341	199	540	93.16	7200
	4	-	-	108	108	-	7200
	5	-	-	108	108	-	7200
	Toplam	25	822	449	1296	85.97	7134
D (100-300)	2	49	465	26	540	76.56	6793
	3	-	394	146	540	91.45	7200
	4	-	2	106	108	98.56	7200
	5	-	2	106	108	97.85	7200
	Toplam	49	863	384	1296	83.09	7030

*: Ortalama değerler üst sınır değerinin elde edildiği test problemleri kullanılarak elde edilmiştir.

Test problemlerinde tesis sayısı için 4 farklı durum söz konusudur. Küçük boyutlu problemler için 2 adet tesis mevcutken, orta boyutlu problemler için 2 ve 3 adet tesis mevcuttur. Büyük boyutlu problemler için ise 3- 4 ve 5 adet tesisin bulunduğu durumlar incelenmiştir. Çizelge 3.20’de de görüldüğü üzere her iki talep durumu için de optimal sonuçların elde edilebildiği tek problemler tesis sayısının 2 adet olduğu problemlerdir. Müşteri taleplerinin 100-200 arasında olduğu durumda tesis sayısının 4 ve 5 olduğu problemler için herhangi bir uygun çözüm bulunamazken, müşteri kapasitesinin 100-300 arasında olduğu problemlerde hem 3 adet tesis hem de 4 adet tesisin mevcut olduğu durumda 2 adet problemde uygun çözüm elde edilebilmiştir. Daha önce de bahsedildiği üzere bu durum müşteri talebinin 100-300 arasında olduğu durum için çözüm uzayının daha küçük olmasının bir sonucudur.

Üretim hızının değişmesi de problemlerin performansını etkilemektedir. Problemlerin çözümü için 3 farklı hız belirlenmiş ve sonuçlar raporlanmıştır. Buna göre üretim hızının 1 olduğu durumda müşteri taleplerinin üretim süresinin talep miktarı ile aynı olduğu varsayılırken, üretim hızının 2 olduğu durumda talep miktarının yarısı ve 3 olduğu durumda ise talep miktarının üçte birine eşit olduğu varsayılmıştır. Kısacası hız arttıkça üretim için gerekli olan süre azalmaktadır. Çizelge 3.21’de test problemlerinin sonuçlarının üretim hızına bağlı değişimi gösterilmiştir.

Çizelge 3.21. Üretim Hızına Göre Deneysel Sonuçlar

Üretim Hızı	Çözüm Durumu			Toplam	Ortalama YSD*	Ortalama Çözüm Süresi	
	Optimal	Uygun Çözüm	Çözüm Bulunamayan				
D (100-200)	1	8	274	150	432	86.27	7138
	2	9	272	151	432	85.47	7132
	3	8	276	148	432	86.17	7131
	Toplam	25	822	449	1296	85.97	7134
D (100-300)	1	14	287	131	432	83.35	7069
	2	18	289	125	432	82.71	7010
	3	17	287	128	432	83.22	7011
	Toplam	49	863	384	1296	83.09	7030

*: Ortalama değerler üst sınır değerinin elde edildiği test problemleri kullanılarak elde edilmiştir.

Çizelge 3.21’de görüldüğü üzere tesis üretim hızlarının değişmesinin problemlerin çözümü açısından çok büyük farklar yaratmadığı açıktır. Her üretim hızı için optimal sonuç elde edilen problem sayıları birbirine çok yakındır. Benzer şekilde uygun çözüm bulunan ve bulunamayan problemler için de problem sayıları yakın sonuçlar vermiştir.

Çizelge 3.22. Araç Kapasitesine Göre Deneysel Sonuçlar

Araç Kapasitesi	Çözüm Durumu			Toplam	Ortalama YSD*	Ortalama Çözüm Süresi	
	Optimal	Uygun Çözüm	Çözüm Bulunamayan				
D (100-200)	300	25	434	189	648	82.81	7068
	600	-	388	260	648	89.71	7200
	Toplam	25	822	449	1296	85.97	7134
D (100-300)	300	49	454	145	648	77.29	6861
	600	-	409	239	648	90.22	7200
	Toplam	49	863	384	1296	83.09	7030

*: Ortalama değerler üst sınır değerinin elde edildiği test problemleri kullanılarak elde edilmiştir.

Çizelge 3.22, araç kapasitesi değişimin test problemleri üzerindeki etkisini göstermektedir. Çizelgede de açıkça görüldüğü üzere optimal çözüm elde edilen tüm problemlerin araç kapasiteleri 300 birimdir. Araç kapasitesinin 300 birim olduğu durumda çözüm uzayının daha kısıtlı olduğu bu nedenle de daha hızlı ve daha iyi çözümler elde edildiği ortadadır. Müşteri taleplerinin 100-300 olduğu durumda çözüm elde edilen problem sayısı 912 iken bu sayı müşteri talebinin 100-200 olduğu durumda 847 adettir.

Çizelge 3.23. Ürün Ömrüne Göre Deneysel Sonuçlar

Ürün Ömrü	Çözüm Durumu			Toplam	Ortalama YSD*	Ortalama Çözüm Süresi	
	Optimal	Uygun Çözüm	Çözüm Bulunamayan				
D (100-200)	300	14	404	230	648	85.59	7126
	600	11	418	219	648	86.35	7142
	Toplam	25	822	449	1296	85.97	7134
D (100-300)	300	25	426	197	648	82.82	7031
	600	24	437	187	648	83.35	7030
	Toplam	49	863	384	1296	83.09	7030

*: Ortalama değerler üst sınır değerinin elde edildiği test problemleri kullanılarak elde edilmiştir.

Çizelge 3.23'te ürün ömrünün test problemleri üzerindeki etkisi gösterilmektedir. Çizelgede de görüldüğü üzere üretim hızında olduğu gibi ürün ömründe de 2 farklı ürün ömrü seviyesi için sonuçlar birbirine yakındır. Ürün ömrünün 300 olduğu durumda araç kapasitesi uygun olmasına rağmen birçok müşteri ürün ömrü kısıtından dolayı rotaya eklenemezken, ürün ömrünün 600 olduğu durumda ise çözüm uzayı genişlediği için çözüm süreleri artmaktadır.

Çizelge 3.24. Koordinat Düzlemine Göre Deneysel Sonuçlar

Koordinat Düzlemi	Çözüm Durumu			Toplam	Ortalama YSD*	Ortalama Çözüm Süresi	
	Optimal	Uygun Çözüm	Çözüm Bulunamayan				
D (100-200)	1	4	264	164	432	86.77	7162
	2	6	279	147	432	86.85	7154
	3	15	279	138	432	84.39	7086
	Toplam	25	822	449	1296	85.97	7134
D (100-300)	1	14	275	143	432	83.24	7041
	2	18	277	137	432	81.71	7014
	3	17	311	104	432	84.19	7036
	Toplam	49	863	384	1296	83.09	7030

*: Ortalama değerler üst sınır değerinin elde edildiği test problemleri kullanılarak elde edilmiştir.

Test problemlerinin sonuçlarının koordinat düzlemine göre değerlendirilmesi Çizelge 3.24'te gösterilmektedir. Ürün ömrünün etkisine benzer şekilde koordinat düzleminin değişik durumları için de birbirine yakın sonuçlar elde edilmiştir. Koordinat düzlemi genişledikçe çözüm uzayı da genişlemektedirken ürün ömrü kısıtından dolayı aynı rotada müşterileri birleştirmek zorlaşmaktadır. Benzer şekilde koordinat düzlemi daraldıkça rotada birleşebilecek müşteri sayıları artarken bu sefer araç kapasitesi bağlayıcı kısıt haline gelmektedir.

3.5.2. Matematiksel model ve DKA algoritması ile elde edilen çözümlerin karşılaştırılması

DKA algoritması C# programlama dilinde kodlanmıştır. Her bir koşum “Intel Xeon (R) E-2236 (12 Core) 3.40 GHz” hızında “32 GB” ara belleğe sahip, “Windows 10” işletim sistemi ile çalışan bilgisayarda gerçekleştirilmiştir. DKA algoritması için herhangi bir süre sınırı belirtilmemiştir. Matematiksel model ve DKA algoritmasının tüm parametrelere göre karşılaştırmalı sonuçları Çizelge 3.25’te ayrıntılı olarak verilmiştir.

Çizelge 3.25. DKA Algoritması ve Matematiksel Modelin Karşılaştırılması

Parametre	Seviye	Matematiksel Model				DKA Algoritması			
		Optimal Çözüm	Uygun Çözüm	Çözüm Bulunamayan	Ort. Çözüm Süresi	Ort. Çözüm Süresi (opt)	Ort. YSD En küçük	Ort.	Ort. Çözüm Süresi (sn)
Müşteri Sayısı	10	74	142	0	5790	3085	-1.57	0.09	0.09
	20	0	432	0	7200	-	-25.25	-4.39	0.48
	30	0	422	10	7200	-	-49.38	-13.87	3.87
	40	0	380	52	7200	-	-66.10	-22.34	20.56
	50	0	282	150	7200	-	-81.88	-30.80	75.42
	100	0	27	621	7200	-	-72.11	-65.08	212.40
Talep Aralığı	~U(100-200)	25	822	449	7134	3799	-81.88	-16.96	71.01
	~U(100-300)	49	863	384	7031	2721	-80.87	-13.48	68.65
Araç Kapasitesi	300	74	888	334	6965	3085	-72.11	-10.74	44.56
	600	0	797	499	7200	-	-81.88	-20.48	95.10
Ürün Ömrü	300	39	830	427	7079	3175	-81.80	-14.70	56.54
	600	35	855	406	7086	2985	-81.88	-15.61	83.12
Koordinat Düzlemi	(-100 x 100)	18	539	307	7102	2479	-81.80	-11.49	85.90
	(-150 x 150)	24	556	284	7085	3052	-79.06	-14.25	70.65
	(-200 x 200)	32	590	242	7061	3451	-81.88	-19.28	52.94
Üretim Hızı	1	22	561	281	7104	3438	-76.39	-9.96	79.65
	2	27	561	276	7072	3096	-79.06	-17.00	65.64
	3	25	563	276	7072	2763	-81.88	-18.46	64.20
Toplam (Ortalama)		74	1685	833	(7083)	(3085)	(-83.84)	(-15.16)	(69.83)

Çizelgenin ilk sütununda yer alan bilgiler test problemlerindeki parametreleri göstermektedir. Buna göre test problemleri 6 farklı parametre üzerinden değerlendirilmiştir. Geliştirilen 2592 adet test probleminin matematiksel model ile çözülmesi sonucunda 7200 sn süre sınırı içerisinde 833 adet test probleminde çözüm elde edilememişken DKA algoritması ile çözüm elde edilemeyen problem yoktur. Tüm problemler için uygun bir çözüm bulunmuştur.

Matematiksel model ile müşteri sayısının 10 olduğu problemler için 74 adet problemde optimal sonuçlar elde edilmiştir. Optimal sonuçların elde edildiği problemler için ortalama çözüm süresi 3085 saniyedir. DKA algoritması ile matematiksel model ile elde edilen tüm sonuçlardan daha iyi ya da o sonuçlara yakın sonuçlar 1 saniyenin altında bir sürede elde edilmiştir. Matematiksel model ile DKA algoritmasının sonuçları karşılaştırıldığı zaman elde edilen YSD değerleri içerisinde en fazla fark %83.84 iken ortalama %15.16 olarak hesaplanmıştır. DKA algoritması hem süre hem de performans açısından çok daha iyi sonuçlar sağlamaktadır. Matematiksel modelin çözümü için gereken süre ortalama 7083 saniye iken bu süre DKA algoritmasında yaklaşık 70 saniyedir.

Matematiksel model sonucunda 10 müşterilik toplam 216 adet problemin 74 adedinde optimal sonuç elde edilmiştir. Matematiksel modelle elde edilen optimal sonuçların hepsine DKA algoritması ile de ulaşılmıştır. Ayrıca DKA kullanılarak çözülen tüm problemlerde uygun bir çözüm elde edilmiştir. DKA algoritmasındaki çözüm sürelerinin özellikle küçük ve orta boyutlu problemler için çok kısa olduğu da açıkça görülmektedir. Çözüm süresinin 10 dakikanın üzerinde olduğu problem sayısı D(100-200) için 37 adet ve D(100-300) için 34 adettir. Müşteri sayısı arttıkça çözüm süresinin arttığı da açıkça görülmektedir. 2592 adet test probleminin 58 adedinde DKA algoritması matematiksel modelden daha kötü sonuçlar elde etmiştir. Ancak DKA algoritması ile elde edilen sonuçlar ile matematiksel model ile elde edilen sonuçlar arasında YSD açısından çok büyük farklar yoktur. Daha kötü sonuç elde edilen test problemlerinin YSD değerleri karşılaştırıldığında D(100-200) olduğu problemler için ortalama YSD değeri %1.66 iken D(100-300) olduğu problem için bu fark %1.36'dır. Müşteri sayısının 30'dan büyük olduğu problemlerde ise DKA algoritması ile elde edilen sonuçlar 2 problem türü için de daha iyidir.

Matematiksel modelde elde edilen tüm optimal sonuçların araç kapasitesinin 300 birim olduğu durumlar için olduğu bilinmektedir. Araç kapasitesinin 600 olduğu hiçbir problemde optimal çözüme ulaşılamamıştır. Matematiksel modelde optimal çözümlere ulaşmak için gereken ortalama süre 1 saate yakinken bu sonuçlar DKA algoritması ile milisaniye ile ifade edilebilecek kadar kısa sürelerde bulunabilmektedir.

Tabloya göre matematiksel modelde D(100-200) olduğu test problemleri için harcanan süre ortalama 7134 saniye iken D(100-300) olduğu problemler için bu süre 7031 saniyedir. DKA algoritmasında ortalama sürelerin yaklaşık 1 dakika civarında olduğu göz önüne alınacak olursa algoritmanın hızlı ve doğru sonuçlar verdiği açıkça söylenebilir.

Özellikle 10 müşteri problem için YSD değerine göre yakın sonuçlar elde edildiği görülürken, 10 müşteri problemlerin dışında algoritmanın çözümlerinin çok daha iyi olduğu söylenebilir.



4. ÇOK TESİSLİ VE ÇOK ARAÇLI BÜTÜNLEŞİK ÜRETİM VE DAĞITIM ÇİZELGELEME (ÇT_ÇA_BÜDÇ) PROBLEMİ

Bu bölümde çok tesisli ve çok araçlı bütünleşik üretim ve dağıtım çizelgeleme probleminin tanımı yapılarak, problemin çözümü için geliştirilen matematiksel model ve Memetik Algoritma (MA) anlatılmıştır. Ayrıca problemin ve çözüm yöntemlerinin daha iyi anlaşılabilmesi için her iki çözüm yöntemi açıklayıcı örnekler üzerinden detaylandırılmıştır.

4.1. Problem Tanımı ve Açıklayıcı Örnek

Tez kapsamında çalışılması planlanan ikinci problem olan ÇT_ÇA_BÜDÇ problemi için varsayımlar genel olarak ÇT_TA_BÜDÇ problemi ile benzerlik göstermektedir. Sistemde farklı üretim kapasitelerine sahip birden çok tesis bulunmaktadır. Tüm tesislerin toplam kapasitesi müşterilerin taleplerini karşılayabilecek seviyededir. Yani müşteri taleplerinin tamamı karşılanmaktadır. Her tesiste üretim hızı sabit bir adet makine bulunmaktadır. Tesislerdeki makinelerin hızları ise birbirinden farklıdır. Bu durumda bir müşterinin talebinin herhangi iki tesisteki üretim süreleri birbirinden farklı olacaktır. Dolayısıyla hangi müşterinin talebinin hangi tesiste üretileceği ve ne kadar süreceği de önemli bir kriterdir. Tesislerdeki makine hızlarının farklı olması üretim maliyetlerinin de farklı olmasına neden olmaktadır. Bu sebeple her tesisin üretim maliyetleri de göz önüne alınmıştır. Ürünlerin dağıtım maliyetlerini azaltmak amacıyla parti üretimi yapılmaktadır. Bu nedenle rotalama kararlarına da ihtiyaç vardır. Hangi müşterinin hangi partide olacağı ve parti içerisinde hangi sırada üretileceği de belirlenmeye çalışılan bir başka konudur. Sınırlı bir yaşam süresine (B) sahip tek bir çeşit ürün üretilmektedir. Yaşam süresi o parti içerisinde bulunan tüm ürünlerin üretimi tamamlandıktan sonra başlamaktadır. Dolayısıyla ürünlerin parti içerisindeki sıralaması ürün ömrü açısından fark etmemektedir. Sistemde her biri coğrafi olarak farklı konumlarda yer alan n adet müşteri ve her müşterinin de sabit bir talebi vardır. Üretim yapılan her tesiste her biri Q kapasitesine sahip en az bir araç mevcuttur. Araç kapasitesi sebebiyle tüm taleplerin tek bir turla sevk edilmesi birçok durumda mümkün olmamaktadır. Bu nedenle araçların yeniden tesise dönerek kalan siparişleri alıp, yeni müşterilere dağıtması gerekmektedir. Yani araçlar birden fazla kere kullanılabilir. Araçların belirli bir kullanım maliyeti söz konusudur ve aracın kaç kez dağıtım

çıktığından bağımsız olarak bir kez göz önüne alınır. Araçların atandıkları ve hizmet edecekleri tesisler belirlidir ve dağıtım işlemi bittikten sonra tekrar aynı tesise dönmeleri gerekmektedir. Sistemde ürünlerin üretilmesi ve dağıtılması için gereken zaman planlama ufku ile belirtilmiştir. Bu süreden önce tüm müşterilerin siparişleri teslim edilmeli ve araçlar tesislere dönmüş olmalıdır. Planlama ufkunun başında tüm araçlar atandıkları tesislerde hazır olarak beklemektedir. Araç hızı sabit olup tüm düğüm çiftleri arasındaki mesafe ve geçiş süreleri belirli ve bilinmektedir. Araçların yüklenme ve boşaltma süreleri ihmal edilmiştir.

Bu varsayımlar altında problemde amacımız, müşteri siparişlerinin üretilip dağıtımının gerçekleştirileceği en az maliyet ile, hangi müşterinin hangi tesisten ve hangi araçla hizmet göreceğini, hangi müşteri siparişlerinin hangi parti içerisinde olacağını, her bir partinin üretimine ne zaman başlanıp bitirileceğini, sistemde her tesis için kaç araç kullanılacağını ve her bir tur içerisinde müşterilerin hangi sırada ziyaret edileceğini tespit etmektir. Bu amaç doğrultusunda her müşteriye yalnızca bir kere ziyaret gerçekleştirilmeli, Araçlara yüklenen ürünlerin toplam alan, hacim, ağırlık vb. gereksinimi araç kapasitesini aşmamalı, her rota bir tesiste başlamalı ve aynı tesiste sonlanmalı, bir partinin üretimi tamamlandıktan sonra o parti içerisinde bulunan tüm ürünler ilgili müşterilere B birimlik zaman içerisinde teslim edilmeli, bir tesise atanan müşterilerin talepleri o tesisin kapasitesini geçmemeli ve tüm üretim ve dağıtım işlemleri planlama ufkundan önce bitirilmelidir.

ÇT_ÇA_BÜDÇ probleminin daha iyi anlaşılabilmesi için geliştirilen test problemleri içerisinde açıklayıcı bir örnek ele alınmıştır. Verilen problemde müşteri sayısı ve tesis sayısı sırasıyla 10 ve 2 olarak ele alınmıştır. Araç kapasitesi ve ürün ömrü 300 birim olarak belirlenmiştir. Müşteriler $(-150,150)$ 'lik bir koordinat düzleminde rassal olarak konumlanmışlardır. Sistemde her tesiste kaç aracın hizmet edeceği de matematiksel model sonucunda belirlenecektir. Üretim hızının 3 olarak belirlendiği bu problemde her tesisin üretim hızı ve kapasitesi birbirinden farklı olacağı için her tesisin hızı 3 ± 0.5 olarak rassal belirlenmiştir. Buna göre T1 tesisinin hızı 3.18 ve T2 tesisinin hızı da 2.85 olarak verilmiştir. T1 tesisinde 1 birim ürün üretme maliyeti 4.3 birim, T2 için 4.2 birim olarak hesaplanmıştır. Her tesisin üretim kapasitesi birbirinden farklıdır ve bu problem için T1 1200 birim ve T2 1300 birim olarak hesaplanmıştır. Ayrıca planlama ufku dediğimiz ve tüm müşteri taleplerinin üretim ve dağıtımının tamamlanması için gerekli olan zaman da 800 birim olarak belirtilmiştir. Bu parametreler doğrultusunda Çizelge 4.1'de müşteri koordinatları ve talepleri verilmiştir.

Çizelge 4.1. Açıklayıcı Örnek İçin Müşteri Koordinatları ve Talepleri

No	X Koordinatı	Y Koordinatı	Talep
T1	-106	46	0
T2	89	-29	0
1	-92	52	150
2	126	-44	160
3	-93	-38	180
4	-113	145	180
5	-98	-73	140
6	147	2	140
7	50	-122	200
8	117	98	150
9	-134	145	130
10	6	-79	190

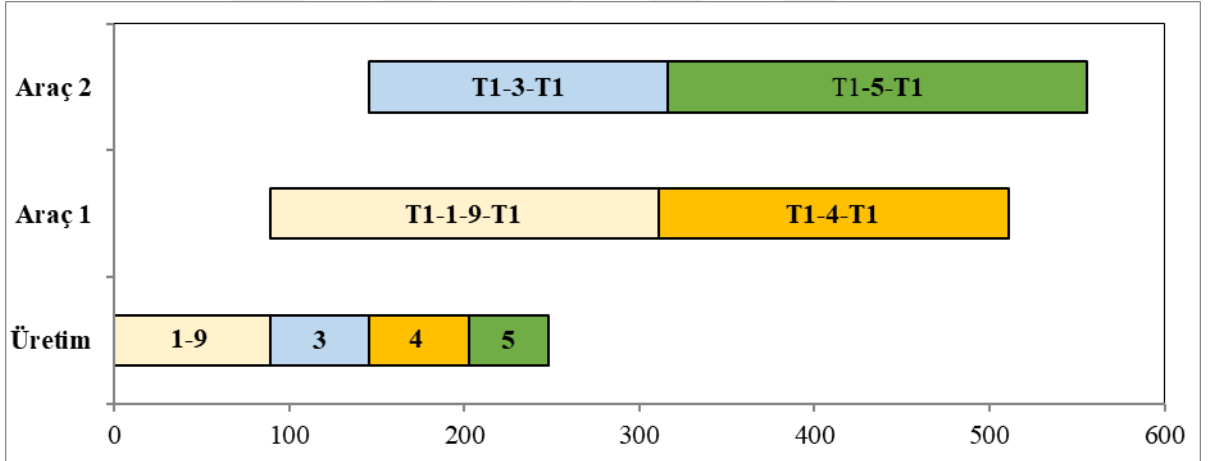
Çizelge 4.1.'de örnek problem için oluşturulan X ve Y koordinatları ile (100-200) arasında rassal oluşturulan müşteri talepleri verilmiştir. Buna göre ilk iki satırdaki T1 ve T2 değerleri tesisleri, diğer satırlar ise 10 adet müşteriyi göstermektedir. T1 ve T2 tesislerinin konumları diğer müşterilerin konumlarına bağlı olarak belirlenmiş ve ağırlık merkezlerine göre konumlandırılmıştır. Bu problem için sistemde 2 adet tesis mevcut olduğu için koordinat sistemi (-100,0) ve (0,100) olmak üzere 2 eşit parçaya ayrılmış ve her iki parçadaki müşterilerin koordinatları göz önüne alınarak düzlemin ağırlık merkezi belirlenmiş ve tesisler bu ağırlık merkezlerine kurulmuştur. Tesislerin herhangi bir talebi bulunmamaktadır. Matematiksel model sonucunda elde edilen optimal çözüme göre tesislerdeki atamalar ve rotalar Çizelge 4.2'deki gibidir.

Çizelge 4.2. Açıklayıcı Örnek İçin Elde Edilen Rota Atamaları

	1.Tesis	Araç No	Üretim Süresi	Dağıtım Süresi	2.Tesis	Araç No	Üretim Süresi	Dağıtım Süresi
	T1-1-9-T1	1	89	222	T2-2-T2	3	57	80
Atanan	T1-3-T1	2	57	170	T2-10-T2	4	67	194
Rotalar	T1-4-T1	1	57	200	T2-6-8-T2	3	102	298
	T1-5-T1	2	45	240	T2-7-T2	4	71	202

Çizelge 4.2'de hangi müşterilerin hangi tesise atandığı, müşteri taleplerinin hangi rotada ve hangi sırayla karşılanacağı, hangi rotanın hangi araçla hizmet göreceği ve rotaların toplam üretim süresi ve dağıtım süresi verilmiştir. Buna göre 1-3-4-5 ve 9 numaralı müşteriler T1'e atanırken, 2-6-7-8 ve 10 numaralı müşteriler T2'ye atanmıştır. T1 için ilk olarak 1 numaralı araç 1 numaralı müşteriye uğrayacak daha sonra 9 numaralı müşteriye geçerek tesise dönecektir. Bir sonraki rota 3 numaralı müşteriye gidiş dönüş

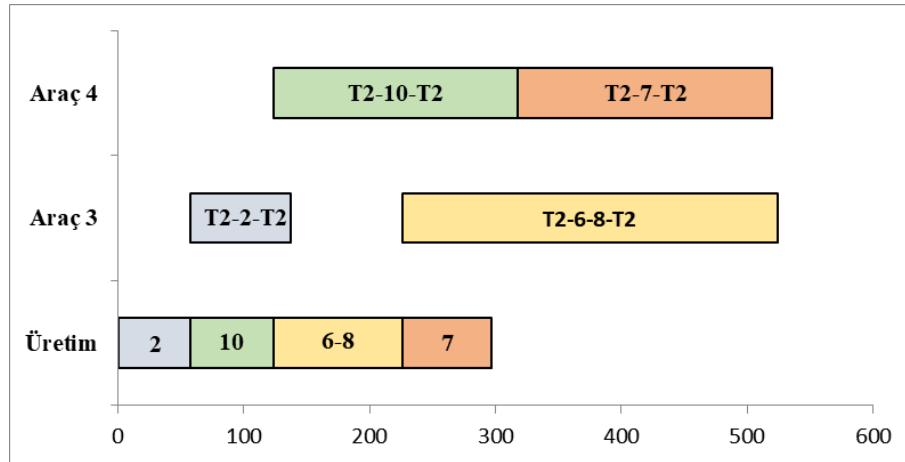
şeklinde. İlk problemimiz ÇT_TA_BÜDÇ probleminde ikinci rotanın hizmete başlayabilmesi için üretiminin tamamlanması ve aracın da bir önceki rotayı tamamlayıp tesise dönmüş olması gerekmektedir. Ancak ÇT_ÇA_BÜDÇ problemi için bir önceki rotanın tamamlanma zorunluluğu bulunmamaktadır. Eğer üretim tamamlandıysa müşteriye dağıtım için bir başka araçtan faydalanılabilir. Bu problem için her tesiste 2 araç olmak üzere toplam 4 araç kullanılmıştır. T1 için optimal çözümdeki rota sıralaması T1-1-9-T1 / T1-3-T1 / T1-4-T1 / T1-5-T1 olarak belirlenmiştir. Benzer şekilde T2 için optimal çözümdeki rota sıralaması T2-2-T2 / T2-10-T2 / T2-6-8-T2 / T2-7-T2 olacak şekilde 4 rotadan oluşmaktadır. Araç kapasitesinin 300 birim olmasından dolayı birçok müşteri aynı rotada birleştirilememektedir. Benzer şekilde bazı müşteriler ise ürün ömrü kısıtlamasından dolayı aynı rota içerisine alınamamaktadır. T1 tesisinde 1 ve 9 numaralı müşterilerin talepleri birlikte üretilip dağıtılırken, T2 tesisinde ise 6 ve 8 numaralı müşterilerin talepleri birlikte üretilip dağıtılmaktadır. Çizelge 4.2’deki verilere göre elde edilen Gantt şemaları Şekil 4.1 ve Şekil 4.2’de gösterilmiştir.



Şekil 4.1. T1 Tesisinin Gantt Şeması

T1 tesisinin Gantt Şeması Şekil 4.1’de gösterilmektedir. Tesise atanan müşteriler ile 4 rota oluşturulmuş ve bu rotalardaki müşterilerin talepleri üretilip dağıtılmıştır. İlk rota olan 1 ve 9 numaralı müşterilerin toplam talebi 280 birimdir. Üretim hızının 3.18 olması nedeniyle bu 264 birimlik talep $280/3.18=88.05$ birim zamanda üretilecektir. Tüm üretim zamanları elde edilen zaman biriminin bir üst tamsayıya yuvarlanmasıyla elde edilmektedir. Dolayısıyla ilk rotanın üretim süresi 89 zaman birimi olarak kabul edilmiştir. Araç(lar) bu süre içerisinde tesiste hazır olarak beklemektedir. 89. zaman diliminde müşteri taleplerinin üretimi bitmiş ve dağıtım işlemi başlamıştır. 1 numaralı

araç 1 numaralı müşteriye 105 zaman diliminde ulaşmıştır. Sonrasında 1 numaralı müşteriden 9 numaralı müşteriye yola çıkmış ve 208 zaman diliminde de 9 numaralı müşteriye ulaşmıştır. Aracın tekrar tesise döndüğü süre 311 zaman dilimi olarak hesaplanmıştır. İkinci rotanın üretimi ilk rotanın üretiminden hemen sonra başlamıştır. 3 numaralı müşterinin talebi 146 zaman diliminde tamamlandığında araç henüz tesise geri dönmemiştir. ÇT_TA_BÜDÇ problemi için 3 numaralı müşterinin talebi dağıtım için aracın gelmesini beklemek zorundayken, ÇT_ÇA_BÜDÇ problemi için diğer araçla dağıtım işlemi başlayabilir. 3 numaralı müşterinin dağıtımı 146 zaman diliminde 2 numaralı araç ile başlayacaktır. 4 numaralı müşterinin talebi 203 zaman diliminde üretildiğinde her iki araç da henüz tesise dönmemiştir. Bu nedenle ya yeni bir araçla 203 zaman diliminde dağıtım başlanacak ya da araçların tesise dönmesi beklenecektir. Matematiksel modelde amacımız maliyet enküçüklemesi olduğu için araçların tesise dönmesini beklemek ve sonra dağıtım başlamak daha az maliyetlidir. Bu nedenle araçların tesise dönmesi beklenecektir. İlk olarak 1 numaralı araç 311 zaman diliminde tesise dönmekte ve hemen 4 numaralı müşterinin talebinin dağıtım için yola çıkmaktadır. Bu bekleme sırasında dağıtım için gerekli süre de göz önüne alınarak, eğer ürün ömrünün aşılması söz konusu olursa 4 numaralı müşterinin talebinin üretimi ürün ömrü aşılmayacak şekilde ötelenebilir. Toplam 4 rotadaki tüm müşterilerin taleplerinin üretimi 248 zaman diliminde tamamlanmış ve son rotanın dağıtım ise 556 zaman diliminde sona ermiştir. Bu süre bize aynı zamanda T1 için D_{max} değerini vermektedir. Planlama ufku belirtilen problem için 800 zaman dilimidir. Dolayısıyla bu süreden önce araçların tamamının tesise dönmesi gerekmektedir. Sistemde tek araç mevcut olsaydı bu süre içerisinde tüm üretim ve dağıtım işlemlerinin gerçekleştirilemeyeceği de açıkça görülmektedir. T1 tesisi için amaç fonksiyonu değerini oluşturan üretim maliyeti, dağıtım maliyeti ve kullanılan araç sayısı maliyeti göz önüne alındığı zaman bu maliyetler sırasıyla 3354, 832 ve 200 olarak hesaplanmaktadır. Toplam maliyet T1 tesisi için 4386 birimdir.



Şekil 4.2. T2 Tesisinin Gantt Şeması

T2 tesisine atanan müşteriler ve rotalara göre oluşturulan Gantt şeması Şekil 4.2’de gösterilmektedir. Buna göre T1’de olduğu gibi 0 zamanında araç(lar) tesiste hazır beklemekte ve üretim tamamlanır tamamlanmaz ilk rotadaki müşterilerin taleplerini teslim etmek üzere araçlardan birisi yola çıkmaktadır. Şekilde de görüldüğü üzere 2 numaralı müşterinin talebi ilk olarak üretilmiş ve 57 zaman diliminde üretimi tamamlanarak 3 numaralı araç ile dağıtımına çıkmıştır. Bu süre içerisinde bir sonraki rota olan 10 numaralı müşterinin talebi üretilmeye başlanmıştır. 10 numaralı müşterinin talebinin üretiminin tamamlanması 124 zaman birimidir. Bu zaman diliminde 3 numaralı araç henüz tesise dönmemiştir. Bu nedenle 10 numaralı müşterinin talebinin dağıtımını 4 numaralı araç ile gerçekleştirilecektir. 3 numaralı araç tesise 137 zaman diliminde dönerken, 4 numaralı araç ise 318 zaman diliminde tesise dönecektir. Bu süreler içerisinde 6 ve 8 numaralı müşterilerin talepleri üretilmeye devam etmektedir. 3 numaralı aracın tesise döndüğü 137 zaman diliminde teslimat için hazır bekleyen herhangi bir sipariş bulunmamaktadır. Dolayısıyla 3 numaralı araç 6 ve 8 numaralı müşterilerin taleplerinin üretilmesini beklemelidir. Tüm müşteri taleplerinin tesliminin tamamlanması ve araçların rotaya dönmesi için gerekli toplam zaman 524 zaman birimidir. Bu süre aynı zamanda T2 için D_{max} değerini vermektedir. Sistemde tek bir aracın mevcut olması durumunda planlama ufku süresince tüm üretim ve dağıtım işlemlerinin gerçekleştirilemeyeceği de açıkça görünmektedir. Özellikle ürün ömrünün bağlayıcı olduğu, belirli süreler içerisinde hizmet etmesi gereken işletmeler için birden fazla aracın sistemde kullanılması kaçınılmazdır. T2 tesisi için amaç fonksiyonu değerini oluşturan üretim maliyeti, dağıtım maliyeti ve kullanılan araç sayısı maliyeti göz önüne alındığı zaman bu maliyetler sırasıyla 3528, 774 ve 200 olarak hesaplanmaktadır. Toplam maliyet T1 tesisi için 4502 birimdir.

Her iki tesisin toplam maliyetleri göz önüne alındığı zaman problemin toplam maliyeti 8888 para birimi olarak hesaplanmıştır. Aynı problemin farklı türleri için farklı maliyetler elde edilebilir. Örneğin üretim hızının 1 olması durumunda üretim süreleri uzayacak dolayısıyla planlama ufkuyla bağlı kalmak için araç sayısının artırılması söz konusu olabilecektir. Ya da araç kapasitesinin 600 olması durumunda birkaç farklı müşterinin daha aynı rotada hizmet görmesi sağlanacak ve yine amaç fonksiyonu değişecektir. Koordinat düzleminin değişmesi müşterilerin tesise olan uzaklıklarını etkileyecek, ürün ömrü kısıtına ve uzaklıklara bağlı olarak farklı rotalar oluşmasına sebep olacaktır.

4.2. ÇT_ÇA_BÜDÇ Problemi İçin Önerilen Matematiksel Model

ÇT_ÇA_BÜDÇ probleminin çözümü için geliştirilen matematiksel model aşağıdaki gibidir:

Dizin Kümeleri ve Parametreler

- N_0 : Tesisler Kümesi
 N_C : Müşteriler Kümesi
 N : Tüm Dğümler Kümesi ($N = N_0 \cup N_C$)
 V : Araçlar Kümesi
 MS : Müşteri Sayısı
 Q : Araç Kapasitesi
 B : Ürün Ömrü
 CV : Araç Kullanım Maliyeti
 PH : Planlama Ufku
 d_i : i müşterisinin talebi
 t_{ij} : i müşterisinden j müşterisine gidiş mesafesi ($\forall i, j \in N$)
 c_{ij} : i müşterisinden j müşterisine gidiş maliyeti ($\forall i, j \in N$) ($c_{ij} = t_{ij}$)
 PQ_k : k tesisinin kapasitesi
 r_k : k tesisinin üretim hızı
 CP_k : k tesisinde 1 birim ürün üretme maliyeti
 p_{ik} : i müşterisinin talebinin k tesisindeki üretim süresi
 M : Büyük bir sayı

İkili Karar Değişkenleri

$$x_{ij} = \begin{cases} 1 & i \text{ düğümünden } j \text{ düğümüne gidilirse } (\forall i, j \in N) \\ 0 & \text{diğer durumda} \end{cases}$$

$$y_{ik} = \begin{cases} 1 & i \text{ müşterisi } k \text{ tesisine atanırsa } (\forall i \in N_c, k \in N_0) \\ 0 & \text{diğer durumda} \end{cases}$$

$$w_{ij} = \begin{cases} 1 & \text{aynı tesise atanan rotalardan önceki rota } i \text{ düğümünde bitip,} \\ & \text{sonraki rota } j \text{ düğümünde başlıyorsa } (\forall i, j \in N_c) \\ 0 & \text{diğer durumda} \end{cases}$$

$$l_{ij} = \begin{cases} 1 & \text{aynı araca atanan rotalardan önceki rota } i \text{ düğümünde bitip,} \\ & \text{sonraki rota } j \text{ düğümünde başlıyorsa } (\forall i, j \in N_c) \\ 0 & \text{diğer durumda} \end{cases}$$

$$a_{vk} = \begin{cases} 1 & v \text{ aracı } k \text{ tesisine atanırsa } (\forall v \in V, k \in N_0) \\ 0 & \text{diğer durumda} \end{cases}$$

$$g_{iv} = \begin{cases} 1 & i \text{ müşterisi } v \text{ aracına atanırsa } (\forall i \in N_c, v \in V) \\ 0 & \text{diğer durumda} \end{cases}$$

$$z_k = \begin{cases} 1 & k \text{ tesisine atama yapılırsa } (\forall k \in N_0) \\ 0 & \text{diğer durumda} \end{cases}$$

Ek Karar Değişkenleri

u_i : i müşterisine girmeden hemen önce araçtaki yük miktarı ($\forall i \in N_c$)

s_i : i müşterisinin servis başlangıç zamanı ($\forall i \in N_c$)

f_i : i müşterisinin talebinin üretiminin bitiş zamanı ($\forall i \in N_c$)

vs_i : i müşterisi ile aynı rotada bulunan ve i müşterisinden sonra gelen müşteri taleplerinin üretilme süresi ($\forall i \in N_c$)

Model

$$\text{En Küçük } \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij} + \sum_{i \in N_c} \sum_{k \in N_0} CP_k d_i y_{ik} + CV \sum_{v \in V} \sum_{k \in N_0} a_{vk} \quad (1)$$

$$\sum_{i \in N} x_{ij} = 1 \quad \forall j \in N_c \quad (2)$$

$$\sum_{j \in N} x_{ji} = \sum_{j \in N} x_{ij} \quad \forall i \in N \quad (3)$$

$$\sum_{k \in N_0} y_{ik} = 1 \quad \forall i \in N_c \quad (4)$$

$$x_{ik} \leq y_{ik} \quad \forall i \in N_c, \forall k \in N_0 \quad (5)$$

$$x_{ki} \leq y_{ik} \quad \forall i \in N_c, \forall k \in N_0 \quad (6)$$

$$x_{ij} + y_{ik} + \sum_{\substack{m \in N_0 \\ m \neq k}} y_{jm} \leq 2 \quad \forall i, j \in N_c; i \neq j, \forall k \in N_0 \quad (7)$$

$$\sum_{v \in V} g_{iv} = 1 \quad \forall i \in N_c \quad (8)$$

$$\sum_{i \in N_c} g_{iv} \leq MS \sum_{k \in N_0} a_{vk} \quad \forall v \in V \quad (9)$$

$$\sum_{k \in N_0} a_{vk} \leq 1 \quad \forall v \in V \quad (10)$$

$$a_{vk} + g_{iv} + \sum_{\substack{l \in N_0 \\ l \neq k}} y_{il} \leq 2 \quad \forall i \in N_C, \forall k \in N_0, \forall v \in V \quad (11)$$

$$x_{ij} + g_{iv} + \sum_{\substack{c \in V \\ c \neq v}} g_{jc} \leq 2 \quad \forall i, j \in N_C; i \neq j, \forall v \in V \quad (12)$$

$$u_j - u_i + Qx_{ij} + (Q - d_i - d_j)x_{ji} \leq Q - d_i \quad \forall i, j \in N_C; i \neq j \quad (13)$$

$$u_i \geq d_i + \sum_{\substack{j \in N_C \\ i \neq j}} d_j x_{ij} \quad \forall i \in N_C \quad (14)$$

$$u_i \leq Q - (Q - d_i) \sum_{k \in N_0} x_{ik} \quad \forall i \in N_C \quad (15)$$

$$\sum_{\substack{j \in N_C \\ i \neq j}} w_{ij} \leq \sum_{k \in N_0} x_{ik} \quad \forall i \in N_C \quad (16)$$

$$\sum_{\substack{i \in N_C \\ i \neq j}} w_{ij} \leq \sum_{k \in N_0} x_{kj} \quad \forall j \in N_C \quad (17)$$

$$\sum_{\substack{j \in N_C \\ i \neq j}} l_{ij} \leq \sum_{k \in N_0} x_{ik} \quad \forall i \in N_C \quad (18)$$

$$\sum_{\substack{i \in N_C \\ i \neq j}} l_{ij} \leq \sum_{k \in N_0} x_{kj} \quad \forall j \in N_C \quad (19)$$

$$\sum_{j \in N_C} \sum_{k \in N_0} x_{kj} - \sum_{i \in N_C} \sum_{j \in N_C} w_{ij} = \sum_{k \in N_0} z_k \quad (20)$$

$$\sum_{j \in N_C} \sum_{k \in N_0} x_{kj} - \sum_{i \in N_C} \sum_{j \in N_C} l_{ij} = \sum_{k \in N_0} \sum_{v \in V} a_{vk} \quad (21)$$

$$l_{ij} \geq \sum_{t \in N_C} w_{it} + \sum_{t \in N_C} w_{tj} - 2 - \sum_{\substack{t \in N_C \\ i, j \neq t}} l_{tj} - \quad \forall i, j \in N_C; i \neq j, \forall v \in V \quad (22)$$

$$M(2 - g_{iv} - g_{jv}) \\ w_{ij} + y_{ik} + \sum_{\substack{l \in N_0 \\ l \neq k}} y_{jl} \leq 2 \quad \forall i, j \in N_C; \forall k \in N_0 \quad (23)$$

$$l_{ij} + g_{iv} + \sum_{\substack{c \in V \\ c \neq v}} g_{jc} \leq 2 \quad \forall i, j \in N_C; \forall v \in V \quad (24)$$

$$\sum_{i \in N_C} y_{ik} - MS(z_k) \leq 0 \quad \forall k \in N_0 \quad (25)$$

$$\sum_{i \in N_C} y_{ik} - MS(z_k) \geq 1 - MS \quad \forall k \in N_0 \quad (26)$$

$$s_i - s_j + Mx_{ij} + (M - t_{ij} - t_{ji})x_{ji} \leq M - t_{ij} \quad \forall i, j \in N_C; i \neq j \quad (27)$$

$$s_i - s_j + Ml_{ij} \leq M - \sum_{k \in N_0} t_{ik} x_{ik} - \quad \forall i, j \in N_C; i \neq j \quad (28)$$

$$\sum_{k \in N_0} t_{kj} x_{kj} \\ s_i \geq f_i + v s_i - \sum_{k \in N_0} p_{ik} y_{ik} + \sum_{k \in N_0} t_{ki} x_{ki} \quad \forall i \in N_C \quad (29)$$

$$f_i - f_j + Mx_{ij} + (M - p_{ik} - p_{jk})x_{ji} \leq M - \quad \forall i, j \in N_C; i \neq j, \forall k \in N_0 \quad (30)$$

$$p_{jk} y_{jk} + M(1 - y_{ik}) \\ f_i - f_j + M w_{ij} \leq M - \sum_{k \in N_0} p_{jk} y_{jk} \quad \forall i, j \in N_C; i \neq j \quad (31)$$

$$f_j \geq p_{jk} y_{jk} + \sum_{\substack{i \in N_C \\ i \neq j}} p_{ik} x_{ij} + M(y_{jk} - 1) \quad \forall j \in N_C, \forall k \in N_0 \quad (32)$$

$$f_j \leq M - \left(M - p_{jk} \right) \left(\sum_{k \in N_0} x_{kj} - \sum_{\substack{i \in N_C \\ i \neq j}} w_{ij} \right) + M(1 - y_{jk}) \quad \forall j \in N_C, \forall k \in N_0 \quad (33)$$

$$s_i - f_i \leq B + M(1 - \sum_{k \in N_0} x_{ik}) \quad \forall i \in N_C \quad (34)$$

$$s_i + \sum_{k \in N_0} t_{ik} x_{ik} \leq PH \quad \forall i \in N_C \quad (35)$$

$$\sum_{i \in N_C} d_i y_{ik} \leq PQ_k \quad \forall k \in N_0 \quad (36)$$

$$vs_j - vs_i + Mx_{ij} \leq M - \sum_{k \in N_0} p_{ik} y_{ik} \quad \forall i, j \in N_C; i \neq j \quad (37)$$

$$vs_i \geq \sum_{k \in N_0} p_{ik} y_{ik} \quad \forall i \in N_C \quad (38)$$

$$vs_i \leq \sum_{k \in N_0} p_{ik} y_{ik} + M(1 - \sum_{k \in N_0} x_{ik}) \quad \forall i \in N_C \quad (39)$$

$$s_i, f_i, u_i, vs_i \geq 0 \quad \forall i \in N_C \quad (40)$$

$$x_{ij}, y_{ik}, a_{vk}, g_{iv}, w_{ij}, l_{ij}, z_k \in \{0,1\} \quad \forall i, j \in N_C, \forall k \in N_0, \forall v \in V \quad (41)$$

Matematiksel modelde amaç, toplam mesafe, üretim ve araç kullanım maliyetlerinin enküçüklenmesidir. (2) numaralı kısıt her düğüme (müşteriye) bir kez uğramayı garantilerken, (3) numaralı kısıt bütün düğümlere giren ve çıkan hat sayısını birbirine eşitlemektedir. Her müşterinin mutlaka bir tesise atanmasını garantileyen kısıt (4) numaralı kısıttır.

(5) – (7) numaralı kısıtlar bir rotanın aynı tesiste başlayıp yine aynı tesiste sonlanmasını sağlamaktadır. (5) numaralı kısıt rota üzerindeki son müşterinin dönüş yaptığı tesise, (6) numaralı kısıt rota üzerindeki ilk müşterinin çıkış yaptığı tesise atanmasını sağlarken, (7) numaralı kısıt rota üzerinde birbirine bağlı olan iki müşterinin farklı tesislere atanmasını engellemektedir.

(8) – (10) numaralı kısıtlar araç ataması ile ilgili kısıtlardır. (8) numaralı kısıt her müşterinin mutlaka bir araca atanmasını garanti etmektedir. Bir tesise atanacak olan araç sayısının belirlenmesini sağlayan kısıt (9) numaralı kısıttır. (10) numaralı kısıt ise her aracın en fazla bir tesiste kullanılmasını garantilemektedir. (11) numaralı kısıt bir araca atanan müşterinin aracın atandığı tesis dışında başka bir tesise atanmasını engellemektedir. (12) numaralı kısıt rota üzerinde birbirine bağlı olan iki müşterinin farklı araçlara atanmamasını garantilemektedir. (13) – (15) numaralı kısıtlar alt tur eleme kısıtlarıdır. (13) numaralı kısıt aracın tesisten çıkış anındaki yük miktarını belirtirken aynı zamanda alt tur oluşumunu engellemektedir. (14) ve (15) numaralı kısıtlar ise alt turları engellemek için kullanılan u_i geçici değişkeninin alt ve üst sınırlarını belirlemektedir.

(16) ve (17) numaralı kısıtlar ardışık iki rota için o rotalardaki ilk ve son müşteri olma durumunu garantilemektedir. Benzer şekilde (18) ve (19) numaralı kısıtlar ise aynı

araca atanan ve ardışık iki rota için o rotalardaki ilk ve son müşteri olma durumunu garantilemektedir. (20) numaralı kısıtla birlikte, toplam oluşturulan rota sayısı ile ardışık rota sayısı arasındaki farkın kullanılan tesis sayısına eşit olması sağlanırken, (21) numaralı kısıt ile de toplam rota sayısı ile aynı araca atanan ardışık rota sayısı arasındaki farkın kullanılan araç sayısına eşit olması garantilenmektedir. (22) numaralı kısıt aynı araca atanmış olan ve kendisinden önce veya sonra herhangi bir rota olan iki müşterinin aynı araçta ardışık olmasını garantilemektedir. (23) numaralı kısıt ardışık iki rotadaki ilk ve son müşterilerin farklı tesislere atanmasını engellerken (24) numaralı kısıt aynı araçtaki ardışık iki rotadaki ilk ve son müşterilerin farklı araçlara atanmasını engellemektedir. (25) ve (26) numaralı kısıtlar ise herhangi bir tesise atama yapılıp yapılmadığını belirlemektedir.

(27) – (33) numaralı kısıtlar zaman kısıtlarıdır ve müşteri taleplerinin üretiminin bitiş zamanları ile servis başlangıç zamanlarını belirlemektedir. (27) numaralı kısıt aynı rotada bulunan ardışık iki düğüm için servis başlangıç zamanlarını gösterirken, (28) numaralı kısıt aynı araçtaki ardışık rotaların ilk müşterileri için servis başlangıç zamanlarını belirtmektedir. Servis başlangıç zamanı için alt sınır (29) numaralı kısıt ile belirlenmiştir.

Benzer şekilde (30) numaralı kısıt ile aynı rotada ardışık olarak bulunan iki düğümün üretim tamamlanma zamanları belirlenmektedir. (31) numaralı kısıt, ardışık rotalardaki son müşterilerin üretim tamamlanma zamanlarını vermektedir. Üretim tamamlanma zamanları için alt ve üst sınır değerleri ise (32) ve (33) numaralı kısıtlar ile belirlenmiştir. (34) numaralı kısıt ürün ömrünün aşılmasını garanti ederken, (35) numaralı kısıt ise planlama ufku sınırlamasının aşılmasını garantilemektedir. Her tesisin üretim kapasitesinin aşılmaması (36) numaralı kısıt ile sağlanmaktadır. (37) – (39) numaralı kısıtlar bir rotadaki müşterilerin toplam üretim sürelerini göstermektedir. (37) numaralı kısıt ile her müşteri için kendisi ve kendisinden sonraki müşterilerin toplam üretim süreleri belirlenirken, (38) ve (39) numaralı kısıtlar üretim sürelerinin alt ve üst sınırlarını göstermektedir. Son olarak (40) ve (41) numaralı kısıtlar ise işaret kısıtlarıdır.

Matematiksel modellerin performansını artırmak amacıyla literatürde sıklıkla kullanılan yöntemlerden birisi de geçerli eşitsizlik eklemeleridir. Bu eşitsizlikler, çözüm süresini kısaltmak için, matematiksel modellerdeki işaret kısıtları gevşetilerek doğrusal modelin çözülmesi ile elde edilen alt sınırı eniyi çözüme yaklaştırmak amacıyla kullanılmaktadır. Normal şartlar altında mevcut matematiksel modelin eniyi çözüme ulaşmasında herhangi bir etkisi olmayan bu kısıtlar, doğrusal gevşetme ile anlamlı hale

gelmektedir. Geçerli eşitsizlikler, geliştirilen kesin algoritmalarda bazı kesirli ve eniyi olmayan çözümlerin çözüm uzayından atılmasında oldukça etkin matematiksel ifadelerdir (Karaođlan, 2009). Aşağıdaki kısıtlar bu amaçla matematiksel modele eklenen geçerli eşitsizliklerdir.

$$\sum_{j \in N_c} \sum_{k \in N_0} x_{kj} \geq [(\sum_{j \in N_c} d_j)/Q] \quad (42)$$

$$\sum_{k \in N_0} a_{(v-1)k} \geq \sum_{k \in N_0} a_{vk} \quad \forall v \in V; v \neq 1 \quad (43)$$

$$\sum_{v \in V} a_{vk} \geq z_k \quad \forall k \in N_0 \quad (44)$$

$$\sum_{k \in N_0} PQ_k z_k \geq \sum_{i \in N_c} d_i \quad (45)$$

$$y_{ik} \leq \sum_{v \in V} a_{vk} \quad \forall i \in N_c, \forall k \in N_0 \quad (46)$$

$$\sum_{i \in N_c} g_{iv} \geq \sum_{k \in N_0} a_{vk} \quad \forall v \in V \quad (47)$$

4.3. Memetik Algoritma (MA)

MA kavramı ilk olarak Moscato (1989) tarafından yapılmış olan bir çalışmada kullanılmıştır. Sonraki yıllarda oldukça popüler bir araştırma alanı haline gelen MA, genetik algoritma ile yerel arama algoritmasının birlikte ele alınması sonucu ortaya çıkmıştır. Memetik algoritmalarda popülasyonun her bir bireyinin uygunluđunu iyileştirmek için yerel arama algoritması kullanılır. Mevcut popülasyondaki yüksek kalitedeki bireyler seçilerek, kullanılan algoritmanın temel adımları bu seçilen bireylere uygulanır ve böylece bir sonraki jenerasyon için yeni bir popülasyon elde edilmiş olur.

Sezgisel algoritmaların başarısı çeşitlendirme (*diversification*) ve yoğunlaşma (*intensification*) olarak adlandırılan iki temel stratejiye dayanmaktadır. GA ile arama uzayının büyük bir kısmı taranırken, yerel arama yaklaşımı ile de taranan bölgelerde komşu çözümler derinlemesine incelenerek daha kaliteli bireylerin elde edilmesi sağlanmaktadır. Bu iki temel stratejiyi de bünyesinde barındıran MA, polinom zamanda çözülemeyen pek çok problemin çözümünde sıklıkla kullanılmaktadır. Genetik algoritmada çaprazlama ve mutasyon ile üretilen yeni birey çözüm uzayının çok farklı bir noktasında ve yerel optimalden uzak olabilmektedir. MA ile ise yerel arama algoritması sayesinde bu yeni üretilen bireylerin tekrar yerel optimal çözüm uzayına girmesi sağlanmaktadır (Radcliffe ve Surry, 1994). Yerel arama algoritması MA'nın en önemli özelliklerinden birisidir. Burada aday çözümler üzerinde yapılan küçük deđişiklikler ile her aday çözümün etrafındaki daha iyi çözümler aranır. Yerel arama algoritmaları MA

çözüm sürecinin değişik aşamalarında yer alabilir. Bazen çaprazlama veya mutasyon operatöründen sonra eklenebilirken, bazen de sadece yeniden üretim aşamasından sonra eklenebilir (Moscato ve ark., 2004).

Memetik algoritmada popülasyondaki her bir birey, ilgili probleme ait bir çözümü temsil eder. Hayatta kalarak genlerini bir sonraki nesillere aktaracak bireyler probleme özgü bir uygunluk değeri ile belirlenir. Uygunluk değeri problemin amaç fonksiyonu değerine eşdeğerdir. Uygunluk değeri yüksek olan bireyler hayatta kalmakta ve genlerini sonraki nesillere aktarabilmektedir. Uygunluk değeri düşük olan bireylerin ise sonraki nesillere genlerinin aktarılması olasılığı çok daha düşüktür. Dolayısıyla, uygunluk değeri yüksek olan bireylerin genleri sonraki kuşaklarda (jenerasyon) daha fazla bireye dağılacaktır. Farklı bireylerden gelen iyi özelliklerin birleşimi ile bazen her bir bireye göre uygunluk değeri çok daha yüksek bireylerin elde edilmesi söz konusu olabilmektedir. Bu durumda bulunduğu çevre için giderek daha uygun olan türler elde edilmektedir. Uygunluk değeri bakımından iyi olduğuna karar verilen bazı bireyler, bir takım genetik operatörler vasıtasıyla yeni bireyler meydana getirirler. Bahsi geçen genetik operatörler çaprazlama ve mutasyon olarak adlandırılır. Çaprazlama, iki bireyin sahip olduğu genlerin belirli bir kısmının aktarılmasıyla yeni bireylerin üretilmesi, mutasyon ise, bir bireyin gen yapısında bir takım küçük değişiklikler yapılarak yeni bireyler üretilmesi işlemidir. Bir popülasyonda yer alan ve herhangi bir genetik operatör işlemine tabi tutulmamış bireylere ebeveyn adı verilirken, ebeveynlere genetik operatörler uygulandıktan sonra elde edilen bireylere ise çocuk adı verilir. Genetik operatörlerin uygulanmasından sonra, ebeveyn ve çocuk bireylerin tümü uygunluk fonksiyonuna göre değerlendirilir ve bunlardan bir kısmı seçilerek yeni popülasyon oluşturulur. Bu işlemler belirli sayıda tekrar edildikten sonra, algoritma, probleme ilişkin optimal veya optimale yakın çözümü temsil edebilecek bir bireyi elde eder (Gen ve Cheng, 1999).

MA'nın uygulama adımlarının daha iyi anlaşılabilmesi için evrimsel algoritmada kullanılan temel kavramlardan bahsedilmiştir. Gen, belirli bir fiziksel özellikten sorumlu olan ve diğer genlerle birlikte yorumlanması gereken kromozom üzerindeki en küçük parçadır. Kromozom ise bir bireyin tüm genetik bilgilerini içeren bir dizidir. Bu tanımlardan yola çıkarak gen probleme ait anlamlı bilgi taşıyan en küçük genetik birim olarak tanımlanabilir. Kromozom ise incelenen problem için aday çözümleri ifade etmektedir. Evrimsel algoritmalarda, yeniden üretim sürecinde oluşan bireyler ebeveynlerinin sadece fiziksel özelliklerini (fenotip) değil aynı zamanda fiziksel özelliklerle ilgili bilgilerini (genotip) de miras olarak alır. Bu nedenle genetik operatörler

genotip üzerinde çalışırken, bireylerin değerlendirilmesi ise fenotip üzerinden gerçekleştirilir. Kromozomun içerdiği bilgileri yani kod yapısını çözüm için anlamlı bilgiye dönüştürme süreci genetik kodun çözümü (*decoding*) olarak adlandırılırken, tam tersi işlem kodlama işlemi (*encoding*) olarak adlandırılmaktadır (Yağmur, 2021). MA'nın genel adımları Çizelge 4.3'te gösterilmiştir.

Çizelge 4.3. MA'nın Genel Adımları

Prosedür:	MA Prosedürü
Girdi:	S_0 : Problem Bilgileri
Çıktı:	S: MA Sonrası Elde Edilen Çözüm
Adım 1:	Başlangıç popülasyonunu oluştur.
Adım 2:	Popülasyondaki her bir bireyin uygunluk değerini hesapla.
Adım 3:	Jenerasyon Sayısı kadar aşağıdaki adımları tekrarla.
Adım 3.1:	Ebeveynleri seç.
Adım 3.2:	Seçilen ebeveynlerden çaprazlama operatörünü kullanarak iki yeni birey oluştur.
Adım 3.3:	Yeni oluşturulan bireylere belirli bir olasılıkla mutasyon operatörünü uygula.
Adım 3.4:	Her iki yeni birey için <i>Yerel Arama Prosedürünü</i> uygula.
Adım 3.5:	Yeni bireyleri değerlendir.
	<ul style="list-style-type: none"> - <i>eğer</i> her iki yeni bireyin de uygunluk değeri popülasyondaki en kötü iki uygunluk değerinden daha iyiyse iki bireyi de kabul ederek, en kötü iki bireyi popülasyondan çıkart. - <i>eğer</i> yeni bireylerden birisinin uygunluk değeri popülasyondaki en kötü uygunluk değerinden daha iyiyse o bireyi kabul ederek, en kötü bireyi popülasyondan çıkart. - <i>değilse</i> Adım 3'e dön.
Adım 4:	En iyi uygunluk değerine sahip bireyi (çözümü) belirle.

MA prosedürü daha önceden belirlenen parametreler ışığında başlangıç popülasyonunun oluşturulması ile başlamaktadır (Adım 1). Daha sonra oluşturulan her bireyin uygunluk değeri hesaplanmaktadır (Adım 2). Popülasyon içerisinde 2 ebeveyn seçilerek (Adım 3.1) bu ebeveynlerin çaprazlanması sonucu iki yeni çocuk oluşturulur (Adım 3.2). Yeni oluşturulan çocuklar belirli bir olasılıkla mutasyona tabi tutulur (Adım 3.3). Yeni oluşturulan çocuklara yerel arama algoritması uygulanarak elde edilen çözümün daha da iyileştirilmesi amaçlanır (Adım 3.4). Elde edilen yeni bireylerin uygunluk değerleri popülasyondaki diğer bireylerin uygunluk değerleri ile karşılaştırılarak popülasyonda güncellemeler yapılır (Adım 3.5). Jenerasyon sona erdiğinde o zamana kadar elde edilmiş en iyi birey raporlanır (Adım 4).

4.3.1. ÇT_ÇA_BÜDÇ problemi için önerilen memetik algoritma (MA)

Bu bölümde ÇT_ÇA_BÜDÇ problemi için geliştirilmiş olan MA tüm ayrıntılarıyla açıklanmıştır. MA algoritmasının genel adımları Çizelge 4.4'te ayrıntılı olarak verilmiştir.

Çizelge 4.4. ÇT_ÇA_BÜDÇ Problemi İçin Önerilen MA'nın Adımları

Prosedür:	MA Prosedürü
Girdi:	S_0 : ÇT_ÇA_BÜDÇ Problem Bilgileri
Çıktı:	S : MA Sonrası Elde Edilen Çözüm
Adım 1:	Popülasyon büyüklüğü kadar yeni birey oluştur. Popülasyondaki ilk bireyi Çizelge 4.5'teki algoritmaya göre, kalan tüm bireyleri rassal olarak oluştur.
Adım 2:	Popülasyondaki her bir birey için aşağıdaki adımları tekrarla.
Adım 2.1:	<i>Prins Algoritması</i> nı kullanarak mevcut çözümü depolara ve alt turlara böl.
Adım 2.2:	Mevcut çözümün uygunluk değerini hesapla.
Adım 3:	Jenerasyon Sayısı kadar aşağıdaki adımları tekrarla.
Adım 3.1:	İkili Turnuva Seçim Mekanizması ile ebeveynleri seç.
Adım 3.2:	Seçilen ebeveynlere ikili çaprazlama operatörünü uygulayarak iki yeni birey oluştur.
Adım 3.3:	Yeni oluşturulan bireylere mutasyon operatörünü uygula.
Adım 3.4:	Yeni oluşturulan her iki bireye de <i>Prins Algoritması</i> nı uygulayarak çözümleri depolara ve alt turlara böl.
Adım 3.5:	Her iki yeni bireyin de uygunluk değerlerini hesapla.
Adım 3.6:	Her iki yeni birey için de <i>Yerel Arama Prosedürü</i> nü uygula.
Adım 3.7:	Yeni bireyleri değerlendir. <ul style="list-style-type: none"> - Eğer her iki yeni bireyin de uygunluk değeri popülasyondaki en kötü iki uygunluk değerinden daha iyiyse iki bireyi de kabul ederek, en kötü iki bireyi popülasyondan çıkart. - Eğer yeni bireylerden birisinin uygunluk değeri popülasyondaki en kötü uygunluk değerinden daha iyiyse o bireyi kabul ederek, en kötü bireyi popülasyondan çıkart. - Değilse Adım 3'e dön.
Adım 4:	En iyi uygunluk değerine sahip bireyi (çözümü) belirle ve elinde tut.

Önerilen MA'ya ilk olarak popülasyon büyüklüğü kadar yeni birey oluşturularak başlanmaktadır. Popülasyondaki ilk birey Çizelge 4.5'te ayrıntılı şekilde anlatılacak olan prosedüre göre geliştirilirken, kalan bireyler tamamen rassal olarak oluşturulmaktadır (Adım 1). Algoritmadaki her bir birey (çözüm) müşterilere ve tesislere ait genel bir sıralamayı ifade eden permütasyon kodlama ile gösterilmektedir. Bu durumda her birey sonsuz kapasiteli araç rotalama problemine karşılık gelmektedir. Ancak ÇT_ÇA_BÜDÇ probleminde hem birden çok tesis hem de mevcut araçların ve tesislerin belirli kapasiteleri mevcuttur. Dolayısıyla tek bir turda tüm müşterilere aynı tesisten ve aynı araçla hizmet etmek mümkün değildir. Bu nedenle mevcut sıradan hangi müşterilerin hangi tesise ve o tesiste hangi rotaya atanacağını belirlemek gerekmektedir. Bu tesise ve rotaya ayırma işlemleri için Prins (2004) tarafından geliştirilen algoritmadan yararlanılmıştır. Prins algoritmasında sadece mesafe enküçüklenmesi göz önüne alındığı

için ÇT_ÇA_BÜDÇ problemine uyarlanarak kullanılmıştır (Adım 2.1). Prins Algoritmasının probleme uyarlanmış şekli ayrıntılı şekilde Çizelge 4.6'da anlatılmaktadır. Müşterilerin tesislere ve rotalara ayrılması işleminden sonra her bir tesis için uygunluk değeri hesaplanmakta ve tüm tesislerin uygunluk değeri toplanarak mevcut bireyin uygunluk değeri belirlenmektedir (Adım 2.2). Uygunluk değeri hesaplanırken mevcut bireyin tesis atamaları gerçekleştirilirken tesis kapasitesinin aşılması ya da oluşturulan rotalarda ürün ömrünün aşılması gibi durumlar söz konusu ise uygunluk değerine ceza maliyetleri de eklenmektedir. Bu durumda uygunluk değeri çok yüksek olan bu çözümlerin uygun çözüm olarak seçilmemesi sağlanmaktadır. Başlangıç popülasyonu oluşturulduktan ve uygunluk değerleri hesaplandıktan sonra mevcut popülasyondan ikili turnuva seçimi ile ebeveynler seçilmektedir (Adım 3.1). Burada ilk ebeveyni belirlemek için 2 adet birey rassal olarak seçilir ve daha sonra bu bireylerden uygunluk değeri en yüksek olan birey birinci ebeveyn olarak seçilir. İkinci ebeveyn de benzer şekilde seçilmektedir. Ebeveynler belirlendikten sonra iki noktalı çaprazlama operatörü ile iki farklı birey oluşturulur. İkili çaprazlama operatöründe ebeveyn dizisinden rassal olarak iki farklı kesme noktası seçilerek, bu ebeveynlerden 2 farklı birey oluşturulmaktadır. Çaprazlama operatöründen sonra mutasyon operatörü ile çaprazlama sonucunda elde edilen çözümler üzerindeki bir veya birkaç değer rassal olarak değiştirilerek yığında yeni dizilerin (yani, arama uzayında yeni çözüm noktalarının) elde edilmesini sağlar. Mutasyon operatörü olarak *Ekleme* (0,0) operatörü kullanılmıştır. Çaprazlama operatörü ile, farklı çözümler arasında bilgi değişimini sağlayarak arama uzayının benzer ancak araştırılmamış bölgelerine ulaşmayı amaçlanırken, mutasyon operatörü ile de mevcut dizilerin bir kısmında rassal değişimi sağlayarak çözüm uzayında yeni noktalar elde edilmesi amaçlanmaktadır. Yeni elde edilen her iki birey de Prins algoritması ile çözüme dönüştürülerek (Adım 3.4), uygunluk değerleri hesaplanmaktadır (Adım 3.5).

Yerel arama sürecinde 5 farklı komşu arama operatörü kullanılmaktadır (Adım 3.6). Bunlar; *Swap*(0,0) (Aynı rota içerisinde yer değiştirme), *Swap*(1,0) (Aynı tesis farklı rota içerisinde yer değiştirme), *Insert* (0,0) (Aynı rota içerisine ekleme), *Insert* (1,0) (Aynı tesis farklı rota içerisine ekleme) ve *RotaTaşıma* (Aynı tesis içerisinde rotanın yerini değiştirme)'dir. Bu operatörler bir önceki bölümde geliştirilmiş DKA algoritmasında da kullanılan komşu arama operatörleridir. Yerel arama sürecinde bahsedilen operatörler kullanılırken araç kapasitesi, tesis kapasitesi ve ürün ömrü kısıtları göz önünde bulundurulmakta böylece uygun çözüm durumu korunmaktadır. Bu şekilde

üretileen komşu çözümler içerisinde en iyi uygunluk değerine sahip olan çözüm yeni çözüm olarak kabul edilir. Yeni çözümlerin uygunluk değerleri mevcut bireylerin uygunluk değerleri ile karşılaştırılır. Mevcut bireyler içerisindeki en kötü uygunluk değerine sahip 2 birey ile yeni bireylerin uygunluk değerleri karşılaştırılır. Eğer yeni elde edilen bireylerin her ikisinin uygunluk değeri de bu çözümlerin uygunluk değerinden daha iyi ise her iki çözümde popülasyona eklenmekte ve en kötü iki çözüm popülasyondan çıkarılmaktadır. Benzer şekilde yeni çözümlerden yalnızca bir tanesinin uygunluk değeri iyiyse o çözüm popülasyona eklenerek, uygunluk değeri en kötü olan çözüm popülasyondan çıkarılır. Eğer yeni üretilen her iki çözümde daha kötü uygunluk değerine sahip ise mevcut popülasyon değiştirilmeden bir sonraki jenerasyona geçilir (Adım 3.7). En iyi uygunluk değerine sahip olan çözüm ve uygunluk değeri hafızada tutularak her jenerasyon sonunda bu çözümle karşılaştırılır ve en iyi çözüm elde edilir (Adım 4).

Başlangıç popülasyonu oluşturulurken en temel ve basit yöntem tüm popülasyonu rassal oluşturmaktır. Ancak tüm popülasyonun rassal olarak oluşturulması iyi çözümlerden uzak çözümler elde edilmesine neden olabilir. Bu nedenle popülasyon oluşturulurken ilk birey temelde ARP için geliştirilmiş olan Clarke ve Wright (1964) tasarruf algoritmasının ÇT_ÇA_BÜDÇ problemine uyarlanmış hali kullanılarak oluşturulmuştur. Bu algoritmada müşteriler, kendileri için belirlenmiş ceza maliyetleri (pc_i) göz önünde tutularak depolara atanır ve her müşteriye tek bir rota üzerinde hizmet verilir. Daha sonra ardışık iyileştirme yöntemleri ile daha iyi çözümler elde edilmeye çalışılır. Bu algoritma Çizelge 4.5'te ayrıntılı olarak anlatılmaktadır.

Çizelge 4.5. ÇT_ÇA_BÜDÇ Problemi İçin İlk Bireyin Oluşturulması

Prosedür:	Başlangıç Popülasyonundaki İlk Bireyin Oluşturulması
Girdi:	S_0 : ÇT_ÇA_BÜDÇ Problem Bilgileri
Çıktı:	S : Popülasyondaki İlk Birey
Adım 1:	Bütün Müşteriler İçin Ceza Maliyetlerini (pc_i) Hesapla. $pc_i = d_{ik} - d_{il}$ (k, l : i müşterisine en yakın ve ikinci en yakın tesisler)
Adım 2:	Bütün müşteriler için pc_i değerlerini azalan sırada sırala.
Adım 3:	Bütün müşteriler için pc_i değeri en yüksek olan müşteriden başlamak üzere tüm müşterileri sırasıyla tesislere ata. - Eğer $CQ_k < PQ_k$ ise müşteriyi k tesisine ata. - Değilse müşteriyi l tesisine ata.
Adım 4:	Tesise atama işlemleri sonrasında eldeki sıralamayı popülasyondaki ilk birey olarak kabul et.

ÇT_ÇA_BÜDÇ probleminde birden çok tesis ve her tesisin de belirli bir kapasitesi vardır. Dolayısıyla klasik Clarke ve Wright (C&W) algoritmasındaki gibi her müşterinin

kendisine en yakın olan tesise atanması sonucunda tesis kapasitesi aşılabileceği için uygun çözüm elde edilemeyebilir. Dolayısıyla her müşterinin kendisine en yakın ve ikinci en yakın tesise olan uzaklığının belirlenip çeşitliliğin artırılması çözüm kalitesi açısından önemlidir (Adım 1). Tüm müşteriler için pc_i ceza maliyetleri belirlendikten sonra bu değerler büyükten küçüğe olacak şekilde sıralanacaktır (Adım 2). Müşterilerin tesislere atanması bu sıraya göre gerçekleştirilecektir. Bu durumda pc_i değerinin büyük olması müşterinin ilk tesise atanması ile ikinci tesise atanması arasındaki uzaklık farkının çok olduğu anlamına gelmektedir. Dolayısıyla müşteriler mümkün olduğu kadar kendisine en yakın tesise atanacaktır. Tüm müşterilerin tesislere atama işlemi tesis kapasiteleri de göz önüne alınarak gerçekleştirilmektedir (Adım 3). Eğer mevcut kapasite (CQ_k) tesis kapasitesi (PQ_k)'ni aşmıyor ise mevcut müşteri o tesise atanır ve bir sonraki müşteri ile algoritma devam eder. Tüm müşteriler tesise atandıktan sonra ilk tesisten başlamak üzere atanan müşterilerin sırası başlangıç popülasyonundaki ilk birey olacak şekilde kodlanır.

Her bir bireyi gerçek bir çözüme dönüştürmek için kullanılan ÇT_ÇA_BÜDÇ problemine uyarlanan Prins algoritmasının adımları Çizelge 4.6'da ayrıntılı bir şekilde açıklanmıştır.

Çizelge 4.6'da ayrıntılı olarak anlatılan Prins Algoritması, başlangıç değerlerinin atanmasıyla başlamaktadır (Adım 1). Standart Prins algoritmasında yalnızca tek tesis mevcut olduğu için tesis atama işlemlerine ihtiyaç duyulmamaktadır. Ancak bu problem için birden çok tesisin var olması tesis atama sürecine de ihtiyaç oluşturmuştur. Bu durumda bireyin ilk elemanından başlamak üzere atanan elemanların müşteri ya da tesis indisi olduğunun kontrol edilmesi gerekmektedir. Permütasyon kodlama kullanılarak oluşturulan bu bireyler $Müşteri Sayısı + Tesis Sayısı - 1$ elemandan oluşmaktadır. Müşterileri tesise atama işlemleri de tesis indislerine göre belirlenmektedir. Sırasıyla seçilen her eleman için tesis indisine kadar olan tüm elemanlar indisi en küçük olan tesise atanmaktadır. Tesis indisi gelmesi halinde ise o ana kadar atama yapılan tesisten bir sonraki indisli tesise atama yapılmaya başlanacaktır. Bir sonraki eleman ile algoritmaya devam edilir (Adım 2).

Çizelge 4.6. Prins Algoritması Adımları

Prosedür:	Prins Algoritması
Girdi:	S_0 : Popülasyondaki Bireylerin Bilgileri
Çıktı:	S : Popülasyondaki, Bireylerin Çözüme Dönüşmüş Hali
Adım 1:	V_i , i müşterisine giden hattın maliyeti olmak üzere, tüm tesisler için $V_k = 0$ ve tüm müşteriler için $V_i = \infty$ atamasını yap.
Adım 2:	Tüm müşteriler bir depoya atanana kadar aşağıdaki adımları tekrarla. <ul style="list-style-type: none"> - Eğer diziden seçilen eleman müşteri indisine karşılık geliyor ise elemanı ilk tesise ata ve bir sonraki indise geç. - Eğer diziden seçilen eleman tesis indisine karşılık geliyorsa bir sonraki indise geç ve bir sonraki tesise ata.
Adım 3:	Her tesis için aşağıdaki adımları uygula. <p>Adım 3.1: İlgili tesise atanan tüm müşterilerin müşteri taleplerini topla.</p> <p>Adım 3.2: Tesis kapasitesi (PQ_k) ile toplam müşteri taleplerini (CQ_k) karşılaştır. <ul style="list-style-type: none"> - Eğer $CQ_k < PQ_k$ bir sonraki adıma geç. - Değilse uygunluk değerine $(CQ_k - PQ_k) * M_t$ değerini ekle ve bir sonraki adıma geç. </p>
Adım 4:	Araç kapasitesi kısıtı sağlandığı sürece tüm alternatif hatlar için aşağıdaki adımları uygula. <p>Adım 4.1: Hattın maliyetini ($c_{i,i+1}$) hesapla.</p> <p>Adım 4.2: <ul style="list-style-type: none"> - Eğer $c_{i,i+1} - d_{i+1,k} < B$ ise - Eğer $V_{i-1} + c_{i,i+1} < V_j$ ise V_j maliyetini güncelle. - Değilse bir sonraki seçenek ile devam et. - Eğer rotada tek eleman varsa ve $c_{i,i+1} - d_{i+1,k} > B$ ise uygunluk değerine $c_{i,i+1} - B) * M_b$ değerini ekle. - Değilse bir sonraki seçenek ile devam et. </p>
Adım 5:	En düşük V_i değerini veren rotaları sondan geriye doğru belirleyerek rotalara ayırma işlemini tamamla ve uygunluk değeri hesaplamaya geç.

Tesise atama işlemleri tamamlandıktan sonra bir diğer önemli nokta tesis kapasitesinin müşteri taleplerinin hepsini karşılayıp karşılamayacağını belirlenmesidir (Adım 3). İlgili tesise atanan tüm müşterilerin talepleri toplamının tesis kapasitesini geçmemesi çözümün uygunluğu açısından oldukça önemlidir (Adım 3.1). Yani tesis kapasitesinin aşılması halinde elde edilen çözüm uygun bir çözüm olmayacaktır. Bu nedenle tesis kapasitesinin aşılması halinde çözümün uygunluk değerine kapasitenin aşılan miktarı ile orantılı olacak şekilde hesaplanan ceza maliyeti eklenmekte ve çözümün uygun çözüm olarak seçilmesi engellenmektedir (Adım 3.2).

Tesis kapasitesi kontrolünden sonra ise rotalara ayırma süreci başlamaktadır. Bireyin ilk elemanından başlamak üzere sırasıyla mümkün tüm olasılıklar denenerek en düşük maliyetli rotalar belirlenmeye çalışılmaktadır (Adım 4). Hat maliyetleri tesisten ilk elemana ve ilk elemandan da tesise olacak rota alternatifi ile başlamaktadır (Adım 4.1). Eğer hatta tek bir eleman olmasına rağmen ürün ömrü kısıtı aşıyorsa bu çözüm uygun bir çözüm olmadığı anlamına gelmektedir ve uygunluk değerine ürün ömrünün aşılan miktarı ile orantılı olacak şekilde elde edilen ceza maliyeti eklenmektedir. Hatta birden fazla eleman olması durumunda ise o rota alternatifi devre dışı bırakılarak farklı

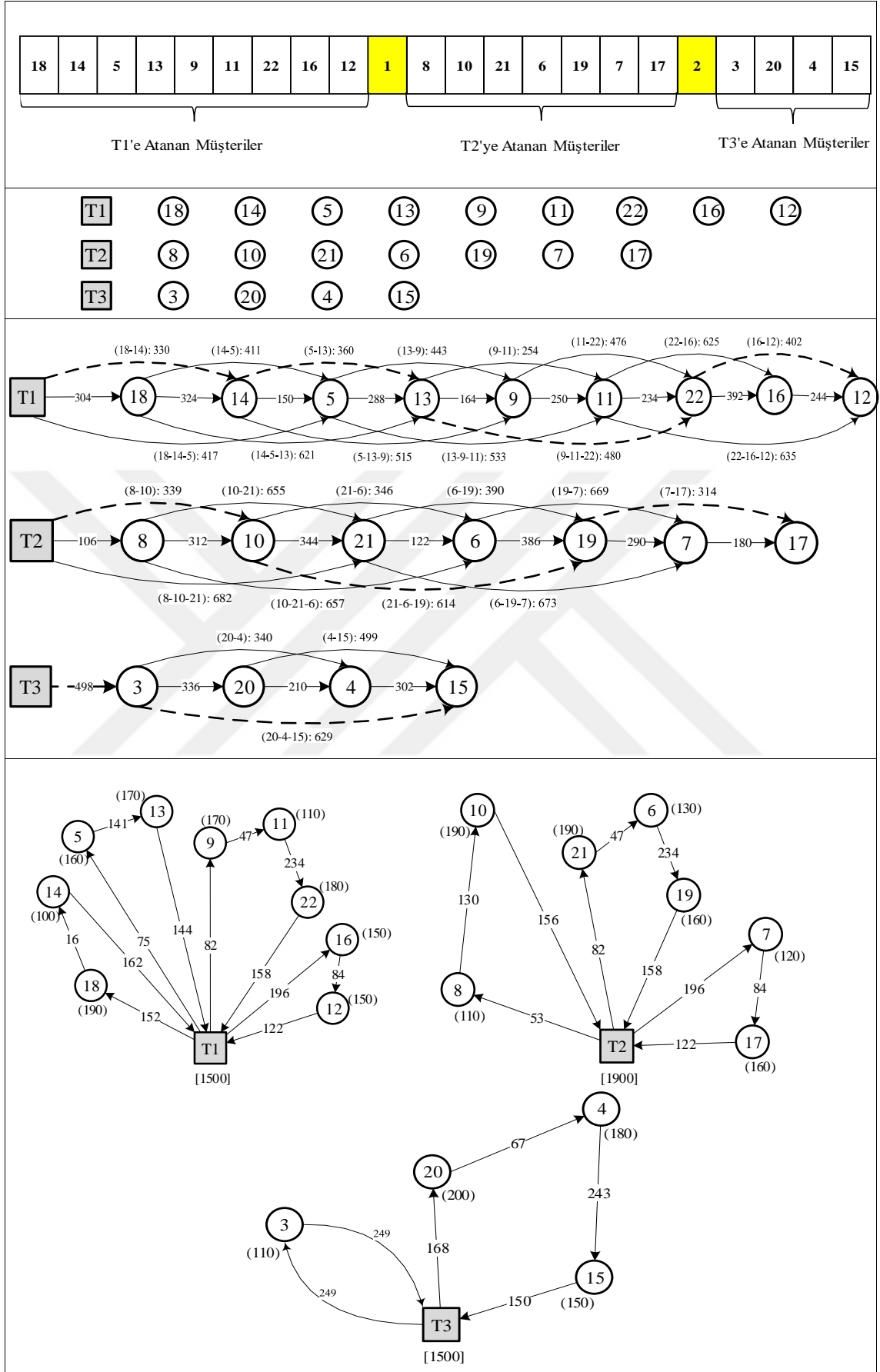
alternatiflerin değerlendirilmesi ile devam edilir. Tüm bu adımlardan sonra bütün elemanlar için en iyi V_i değerleri belirlenmiş olacaktır (Adım 4.2). En iyi V_i değerleri belirlendikten sonra ise rotalar sondan geriye doğru olacak şekilde hangi müşteriye hangi indisten geldiği ve sonrasında nereye gidileceği de belirlenmiş olacaktır (Adım 5).

4.3.2. MA'nın örnek probleme uygulanması

Bu bölümde daha önce matematiksel model ile çözümü örnek olarak verilen test probleminin önerilen MA ile çözümü ele alınmıştır. MA için popülasyon büyüklüğü $min(MüşSay * 4; 100)$ olarak kabul edilmiştir. Bu durumda 10 müşterilik bir problem için başlangıç popülasyonunun büyüklüğü 40 olacaktır. Önerilen MA yaklaşımında çaprazlama operatörü olarak iki noktalı çaprazlama operatörü kullanılmıştır. Jenerasyon sayısı olarak da yine müşteri sayısına bağlı olarak $min(MüşSay * 40; 1000)$ hesaplaması kullanılmıştır.

Buna göre algoritma ilk olarak başlangıç popülasyonunun oluşturulması ile başlayacaktır. Bu aşamada ele alınan problem için 40 adet birey oluşturulacaktır. Bunlardan ilk birey C&W algoritmasının bu problem için revize edilen hali ile oluşturulurken geri kalan 39 birey rassal olarak oluşturulacaktır. Popülasyondaki bireylerin kromozom sayısı müşteri sayısı ve tesis sayısı toplamından 1 eksik olacak şekilde planlanmıştır. Örnek bir çözüm gösterimi Şekil 4.3'te gösterilmiştir.

Şekil 4.3'te 20 müşteri ve 3 tesisin mevcut olduğu bir problem için kromozom gösterim örneği verilmiştir. Popülasyondaki her bireyin kromozom sayısı $20 + 3 - 1 = 22$ olarak belirlenmiştir. Tüm çözüm tek bir permütasyon kodlama ile temsil edilmektedir. Burada asıl önemli olan nokta müşterilerin hangi tesise atanacağıdır. Bunu belirleyebilmek için $TesisSayısı - 1$ adet ayraç kullanılmıştır. Bu ayraçlar hangi müşterinin hangi tesisten hizmet alacağını göstermektedir. Ayraç için müşteri indisi kullanıldığından müşteri indisleri 1'den değil tesis sayısından başlamaktadır. Algoritma bireyin ilk kromozomundan başlamak üzere sırasıyla ilerlemektedir. İlk ayraça gelene kadar olan tüm müşteriler T1'e atanmakta, sonraki ayraça gelene kadar olan tüm müşteriler T2'ye atanmaktadır. Bu şekilde müşteriler ve tesisler bitene kadar bu işlemlere devam edilir. $TesisSayısı - 1$ sayısına ulaşıldığı zaman kalan tüm müşteriler ise son tesise (T3) atanmaktadır.



Şekil 4.3. Örnek Kromozom Gösterimi ve Çözümüne Dönüştürülmesi

Şekil 4.3'e göre 1 numaralı ayrıca kadar olan 18-14-5-13-9-11-22-16 ve 12 indis numaralı müşteriler T1'e, 8-10-21-6-19-7-17 indis numaralı müşteriler T2 ve 3-20-4-15 indis numaralı müşteriler ise T3'e atanmıştır. Müşterilerin atanacağı tesisler belirlendikten sonra tesis kapasitesinin aşıp aşılmadığı kontrol edilmektedir. Tesislerin kapasitesi "[X]" şeklinde belirtilmiştir. Bu örnek için her üç tesiste de kapasite kısıtı sağlanmaktadır. Daha sonraki adımda Prins Algoritması yardımıyla tesislere atanan müşterilerin rotalara bölünme işlemi gerçekleştirilir. Araç kapasitesi ve ürün ömrünün 600 birim olarak ele alındığı bu problem için her tesis bazında birleştirme alternatifleri gösterilmiştir. Buna göre T1 tesisinde 14 numaralı müşteriye gidebilmek için iki farklı alternatif söz konusudur. Bunlardan ilki T1-18-T1 ve T1-14-T1 rotalarını takip etmektir. Bu durumda toplam kat edilen mesafe 628 birim olacaktır. İkinci alternatifte ise T1-18-14-T1 rotası olacaktır. Bu durumda araç kapasitesi ve ürün ömrü kısıtlarının sağlanıp sağlanmadığı kontrol edilmelidir. Kısıtlar sağlanıyorsa mevcut rota alternatif bir çözüm olarak değerlendirmeye alınır. T1-18-14-T1 rotası için toplam mesafe 330 birim olacaktır. Tüm müşteriler için o müşteriye gelinebilecek tüm rota alternatifleri değerlendirildikten sonra T1 için elde edilen rotalar T1-18-14-T1 / T1-5-13-T1 / T1-9-11-22-T1 ve T1-16-12-T1 olarak belirlenmiştir. Benzer şekilde T2 tesisi için elde edilen rotalar T2-8-10-T2 / T2-21-6-19 / T2-7-17-T2 şeklinde oluşturulmuştur. T3 tesisi için 3 numaralı müşteriden sonra gelen 20 numaralı müşteriye gidebilmek için tek alternatif T3-3-T3 rotasından sonradır. 3 numaralı ve 20 numaralı müşterilerin aynı rotada birleştirilebilmesi araç kapasitesi kısıtını sağlamasına rağmen ürün ömrü kısıtını sağlamadığından dolayı gerçekleştirilemez. Dolayısıyla ilk olarak 3 numaralı müşteri ziyaret edilecek daha sonraki rota 20 numaralı müşteriyle başlayacaktır. T3 tesisine atanan 20-4 ve 15 numaralı müşteriler ise tek bir rotada birleştirilerek en uygun maliyetli çözüm elde edilmiştir.

Benzer gösterim ile açıklayıcı örnek olarak seçilen problemimiz için de 40 adet birey oluşturulmuştur. Bu bireylerden 5 adet birey örnek olarak Şekil 4.4'te gösterilmektedir.

1. Birey	2	10	5	4	6	1	3	7	8	9	11
2. Birey	1	10	3	2	11	7	5	4	6	9	8
3. Birey	5	9	6	10	1	7	3	4	11	2	8
4. Birey	9	5	10	1	4	11	3	2	6	8	7
5. Birey	11	4	7	6	9	3	10	8	5	1	2

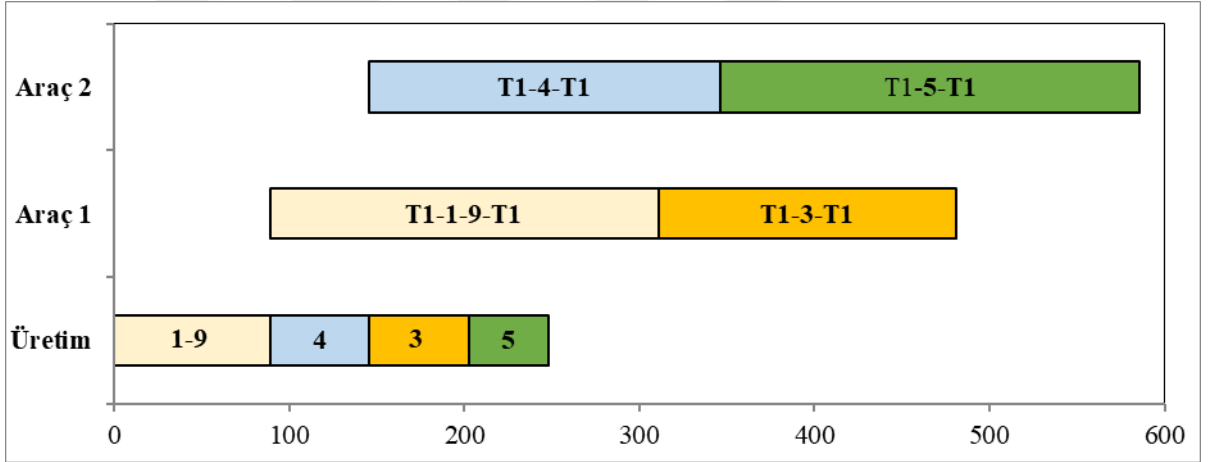
Şekil 4.4. Popülasyondaki Bireylerin Gösterimi

Şekil 4.4'te popülasyondaki bazı bireyler gösterilmiştir. Buradaki ilk birey C&W algoritması ile elde edilen çözümü temsil ederken, kalan bireyler rassal olarak oluşturulmuş çözümlerdir. Örnekte 2 adet tesis mevcut olduğu için bir tane araç kullanılmıştır. İlk bireyde 2-4-5-6 ve 10 numaralı müşteriler T1'e atanırken, 3-7-8-9 ve 11 numaralı müşteriler ise T2'ye atanmıştır. Tesis indisi (araç) 1 ile gösterildiği için müşteri indisleri 2'den başlamaktadır. Dolayısıyla 2-4-5-6 ve 10 numaralı indisler sırasıyla 1-3-4-5 ve 9 numaralı müşterileri temsil etmektedir. Burada herhangi bir rotalama söz konusu değildir. Rotalama işlemleri ayrıca Prins Algoritması kullanılarak gerçekleştirilecektir. 2. Bireye bakıldığında ilk kromozomun ayrıca denk geldiği görülmektedir. Bu durumda T1 tesisine herhangi bir atama yapılmaz. Tüm müşteriler T2'den hizmet almaktadır.

Tesis atamaları belirlendikten sonraki aşamada elde edilen çözümlerin uygunluğu kontrol edilerek, bireylerdeki kromozom sırası korunacak şekilde Prins algoritması ile her tesis için ayrı olarak rotalama işlemleri gerçekleştirilmektedir. 1. Birey için kromozom gösterimin çözüme dönüştürülmesi tesislere atama işleminden sonra tesis kapasitesinin uygunluğunun kontrolü ile devam etmektedir. Buna göre T1 tesisine atanan 2-10-5-4 ve 6 numaralı müşterilerin talepleri sırasıyla 150,130, 180,180 ve 140 olmak üzere toplam 780 birimdir. T1 tesisinin kapasitesi ise 1200 birim olduğu için kapasite aşılması durumu söz konusu değildir. Kapasitenin aşılması halinde bu çözüm uygun bir çözüm olmayacak ve ceza maliyetleri ile çözümün amaç fonksiyonu değeri yükseltilecek bu çözümün çaprazlama ve mutasyon işlemlerinde elenmesi sağlanacaktır.

Kapasite kontrolü gerçekleştirildikten sonraki adımda Prins algoritması ile 2-5-10-4 ve 6 numaralı müşterilerin atanacakları rotaların belirlenmesine geçilmektedir. 2 numaralı müşteriden başlamak üzere her mümkün alternatif değerlendirilerek en düşük maliyetli rotalar oluşturulmaktadır. Rotalar birleştirilirken tüm alternatifler için araç kapasitesi ve ürün ömrü kontrolü yapılmaktadır. Örneğin, 2 ve 5 numaralı müşterilerin aynı rotada birleştirilmesinde herhangi bir engel yokken, 5 ve 10 numaralı müşterilerin aynı rotada birleştirilmesi araç kapasitesi kısıtını sağlamadığından mümkün değildir. Tüm mümkün olasılıklar değerlendirildikten sonra T1 tesisi için elde edilen sıralamalar T1-2-10-T1 / T1-5-T1 / T1-4-T1 / T1-6-T1 şeklinde elde edilmiştir. Rotalar belirlendikten sonra amaç fonksiyonunun hesaplanabilmesi için araç sayısının belirlenmesi gerekmektedir. Bu aşamada ilk olarak tesiste teorik araç sayısı belirlenmekte ve buna göre Gantt şeması oluşturulmaktadır. Teorik araç sayısı o tesise atanan tüm müşterilerin taleplerinin üretim süresi ve o tesise atanan son rotanın dağıtım süresinin toplamının planlama ufkuna

bölünmesi ile elde edilmektedir. Sistemde tek makine olduğundan ürün ömrü kısıtını sağlamak için herhangi bir öteleme olmadığı varsayılarak üretimin biteceği zaman belirlidir ve daha kısa sürede tamamlanma imkânı bulunmamaktadır. Dağıtım süreleri için bu durum göz önüne alındığında ise son rotanın üretimi tamamlandığı zaman aracın tesiste bulunduğu varsayılırsa son dağıtımın gerçekleştirilip aracın tesise döneceği zaman o rotanın uzunluğuna bağlıdır. Bu elde edilen değerler ürün ömrü kısıtı ya da aracın tesise dönmemesi gibi durumlardan uzayabilir ancak teorik değerden daha kısa bir değer elde edilemez. Bu nedenle tesise atanabilecek minimum araç sayısı bu hesaplamayla elde edilen sayıdan daha az olamayacaktır. Teorik araç sayısı belirlendikten sonra tesisteki müşterilerin taleplerinin ne zaman üretileceği ve ne zaman dağıtıma çıkarılacağı hesaplanmaktadır. Bu problem için teorik araç sayısı 2 olarak hesaplanmıştır. Dolayısıyla üretim ve dağıtım süreleri hesaplanırken sistemde 2 araç olduğu varsayılmaktadır. Şekil 4.5'te T1 tesisinin C&W sonrası elde edilen Gantt şeması mevcuttur.

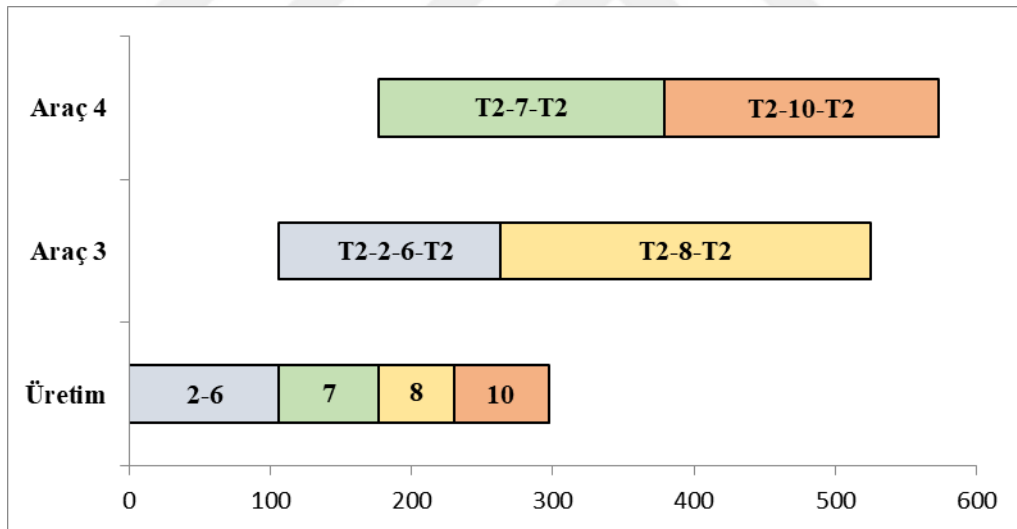


Şekil 4.5. T1 Tesisinin C&W Sonrası Elde Edilen Gantt Şeması

T1 tesisinin C&W sonuçlarına göre elde edilen Gantt şeması Şekil 4.5'te verilmiştir. Şekilde görüldüğü üzere her rotanın üretim süreleri kısa, dağıtım süreleri uzundur. Bu nedenle araç herhangi bir müşterinin ürün talebini dağıtıp dönene kadar diğer bir üretim tamamlanmaktadır. Bu durumda eğer tesiste boş bekleyen herhangi bir araç mevcut ise dağıtım için o araç kullanılırken, tesiste araç olmaması halinde ise tesise ilk dönen araç kullanılmaktadır. T1-2-10-T1 / T1-5-T1 / T1-4-T1 / T1-6-T1 ile gösterilen sıralamadaki indisler sırasıyla 1,9,4,3 ve 5 numaralı müşterileri temsil etmektedir. Dolayısıyla Gantt şemasında da asıl müşteriler belirtilmiştir. Şekil 4.5'te verilen Gantt şemasındaki müşteriler ile matematiksel model ile elde edilen sonuçları gösteren Şekil

4.1'deki müşteriler aynı müşterilerdir. Rota atamaları MA için T1-1-9-T1 / T1-4-T1 / T1-3-T1 / T1-5-T1 iken matematiksel modelde bu sıra T1-1-9-T1 / T1-3-T1 / T1-4-T1 / T1-5-T1 şeklindedir. Planlama ufkunun 800 olduğu bu problem için T1 tesisinin MA ile elde edilen D_{max} değeri 586 iken matematiksel model ile bu sonuç 556 olarak hesaplanmıştır. Her iki çözüm için D_{max} değerleri farklı olsa bile amaç fonksiyonu değerleri birbirine eşittir. MA ile elde edilen çözümde T1 tesisindeki üretim maliyeti 3354 para birimi, dağıtım maliyeti 832 para birimi ve araç kullanım maliyeti de 200 olarak hesaplanmıştır. T1 tesisinin toplam maliyeti 4386 para birimidir.

T1 için çözüm oluşturulduktan sonra aynı adımlar T2 için de uygulanacaktır. T2'ye atanan müşterilerin taleplerinin tesis kapasitesini geçip geçmediği kontrol edilerek başlanır. Toplam talep 840 birimken tesis kapasitesi 1300 birimdir. Dolayısıyla tesis kapasitesinin aşılmama koşulu sağlanmaktadır. T2'ye atanan müşteriler T2-2-6-T2 / T2-7-T2 / T2-8-T2 ve T2-10-T2 şeklinde rotalara ayrılmıştır. Teorik araç sayısı 2 olarak hesaplanmıştır. Buna göre T2 tesisindeki rota atamalarına göre elde edilen Gantt şeması Şekil 4.6'da verilmiştir.



Şekil 4.6. T2 Tesisinin C&W Sonrası Elde Edilen Gantt Şeması

T2 tesisine atanmış olan tüm müşterilerin sırasıyla taleplerinin üretilmesi ve dağıtılması işlemleri Şekil 4.6'da gösterilmiştir. Özellikle üretimin kısa dağıtımın uzun olduğu bu tarz problemlerde tesislerde birden çok araç kullanılması önemli hale gelmektedir. Çok kısa sürelerde üretilen bu ürünler dağıtım için çok fazla beklemekte ve birçok durumda ürün ömrünün aşılmaması koşulunu yerine getirememektedir. Bu nedenle özellikle planlanan süreler içerisinde müşterilere hizmet etmek, planlama ufku

ve ürün ömrü koşullarını sağlayabilmek için birden fazla araç kullanımını mecburi hale gelmektedir. T2 tesisinde tüm müşterilerin talebini üretmek için gereken zaman 297 zaman birimi iken, en son müşteri olan 10 numaralı müşterinin talebinin dağıtım işlemi tamamlanıp aracın tesise geri döndüğü zaman 573 zaman birimidir. Bu da T2 tesisi için D_{max} değerine eşittir. T2 tesisinin üretim maliyeti, dağıtım maliyeti ve araç kullanım maliyeti sırasıyla 3528,815 ve 200 para birimidir. T2 tesisinin toplam maliyeti 4543 para birimidir. T1 ve T2 tesislerinin toplam maliyeti 8929 para birimidir. Bu değer popülasyondaki 1.Bireyin amaç fonksiyonu değeri (AFD)'dir. Örnekte verilen 5 birey için amaç fonksiyonu değerleri Şekil 4.7'de gösterilmektedir.

	Atamalar											AFD
1. Birey	2	10	5	4	6	1	3	7	8	9	11	8929
2. Birey	1	10	3	2	11	7	5	4	6	9	8	3922424
3. Birey	5	9	6	10	1	7	3	4	11	2	8	9785
4. Birey	9	5	10	1	4	11	3	2	6	8	7	9972
5. Birey	11	4	7	6	9	3	10	8	5	1	2	3783031

Şekil 4.7. Başlangıç Popülasyonundaki Bireylerin Amaç Fonksiyonu Değerleri

Şekil 4.7'de görüldüğü üzere farklı başlangıç çözümleri için farklı AFD'ler elde edilmiştir. Özellikle 2 ve 5. bireylerin tesis kapasitesi kısıtını sağlamaması nedeniyle uygun çözümler olmadığı açıkça görülmektedir. Bu nedenle AFD hesaplanırken kapasiteyi aşan miktarın 1000 katı ceza maliyeti de eklenmiştir. Dolayısıyla bu çözümler çaprazlama ve mutasyon aşamalarında popülasyondan ilk olarak elenecek çözümler olarak değerlendirilebilir. AFD belirlendikten sonra çaprazlama operatörüne geçilmektedir. Çaprazlama operatörü için ebeveynler belirlenirken ikili turnuva seçiminden faydalanılmıştır. Bu aşamada rassal olarak 2 birey seçilir ve AFD'leri karşılaştırılır. AFD'si en küçük olan birey Ebeveyn1 olarak seçilir. Aynı işlem Ebeveyn2 seçimi için de tekrar edilir. Ebeveynler belirlendikten sonra iki noktalı çaprazlama ile 2 farklı çocuk oluşturulur. İki noktalı çaprazlama için kromozomlar üzerinden 2 farklı nokta rassal olarak seçilmekte ve ebeveynlerdeki bu iki nokta arasındaki kromozomlar çocuklara aynen aktarılmaktadır. Ardından diğer ebeveynde yer alan ve çocuklarda bulunmayan kromozomlar sırasıyla çocuklara aktarılmaktadır. Ebeveyn1'i belirlemek için rassal olarak seçilen 2 birey 1 ve 2 numaralı bireyler olsun. Bu durumda AFD'si en küçük olan 1 numaralı birey Ebeveyn1 olarak atanacaktır. Yine rassal olarak seçilen 5 ve

2 numaralı bireylerden 5 numaralı birey Ebeveyn2 olarak belirlenmiştir. Çaprazlama ve mutasyon örneği Şekil 4.8’de gösterilmiştir.

Ebeveyn1	2	10	5	4	6	1	3	7	8	9	11
Ebeveyn2	11	4	7	6	9	3	10	8	5	1	2
Çocuk1	11	7	5	4	6	1	3	8	9	10	2
Çocuk2	2	5	7	6	9	3	10	4	8	1	11
Çocuk2	2	5	6	7	9	3	10	4	8	1	11

Şekil 4.8. Çaprazlama ve Mutasyon Örneği

Şekil 4.8’de seçilen 1 ve 5 numaralı bireylerden elde edilen yeni çocuklar gösterilmiştir. 1 numaralı birey Ebeveyn1 ve 5 numaralı birey Ebeveyn2 olarak belirlenmiştir. Bu iki bireyin çaprazlama noktaları 2 ve 7. kromozomlar arasındaki noktalar olarak rassal seçilmiştir. Çocuk1 için bu noktalar arasındaki kromozomlar Ebeveyn1’den, Çocuk2 için ise Ebeveyn2’den aynen aktarılmıştır. Çocuk1’in kalan kromozomları Ebeveyn2’den sırasıyla aktarılmaya başlanır. Ebeveyn2’nin ilk kromozomu olan 11, Çocuk1’e aynen aktarılmıştır. Ancak Ebeveyn2’nin ikinci kromozomu olan 4, Çocuk1’de zaten mevcuttur. Bu nedenle Ebeveyn2’nin sıradaki aktarılmamış kromozomlarına bakılarak devam edilir. Benzer şekilde Ebeveyn1’in kromozomları Çocuk2’ye aktarılacaktır. Çocuk1 ve Çocuk2 için çaprazlama işlemleri sona erince, mutasyon operatörüne geçilir. Mutasyon operatörü belli bir yüzde ile uygulanmaktadır. Bu oran %0.1’den %0.50’ye kadar jenerasyon sayısına bağlı olarak kademeli olarak artırılmaktadır. Çocuk1 için mutasyon gerçekleştirilmezken Çocuk2 için mutasyon gerçekleştirilmiştir. Mutasyon işlemi olarak *YerineEkleme* operatörü kullanılmıştır. Buna göre rassal seçilen bir nokta, yine rassal seçilen başka bir noktanın önüne eklenmektedir. 4 numaralı kromozom olan 6, 3 numaralı kromozom olan 7’nin önüne eklenmiş ve yeni bir birey elde edilmiştir. Çocuk1 için T1’e atanan müşteriler sırasıyla T1-11-T1 / T1-7-T1 / T1-5-T1 / T1-4-T1 ve T1-6-T1 şeklinde rotalar oluşturmuştur. T1 için toplam maliyet 5329 para birimidir. Çaprazlama ve mutasyon operatörlerinden sonra AFD hesaplanmakta, hesaplanan AFD yerel arama algoritması ile iyileştirilmeye çalışılmaktadır. Çocuk1’deki ilk tesis için herhangi bir iyileştirme söz konusu değildir. Çocuk1’in ikinci tesisinde ilk rotalar T2-3-T2 / T2-8-T2 / T2-9-T2 / T2-10-T2 ve T2-2-T2 şeklinde oluşturulmuştur. T2 için toplam maliyet 1916026 zaman birimi olarak hesaplanmıştır. Maliyetin bu kadar yüksek çıkması planlama ufkunun

aşılması sonucu katlanılan ceza maliyeti kaynaklıdır. T2 için uygulanan yerel arama algoritmasında T2-8-T2 / T2-3-T2 / T2-9-T2 / T2-10-T2 ve T2-2-T2 şeklinde rotalar güncellenmiş ve bu güncelleme sonrasında planlama ufku kısıtı sağlandığı için AFD 5126 olarak yeniden hesaplanmıştır. Çocuk1 için toplam AFD değeri 10455 para birimi olarak belirlenmiştir.

Benzer şekilde Çocuk2 için hesaplamalar yapıldığında T1 için tüm müşteriler ayrı rotalar oluşturmuş ve AFD 4321427 olarak hesaplanmıştır. Bu maliyette hem tesis kapasitesi aşımı hem de planlama ufkunun aşılmasından kaynaklı cezalar mevcuttur. Yerel arama algoritması ile T1-2-T1 / T1-8-T1 / T1-6-T1 / T1-7-T1 / T1-9-T1 / T1-3-T1 / T1-10-T1 / T1-4-T1 / T1-5-T1 şeklinde rotalar güncellenerek AFD 2331427 zaman birimine düşürülmüştür. T2 tesisindeki tek müşteri 11 numaralı müşteridir. T2 tesisi için AFD 1092 olarak hesaplanmıştır. Çocuk2'nin toplam AFD'si ise 2332519 olarak hesaplanmıştır.

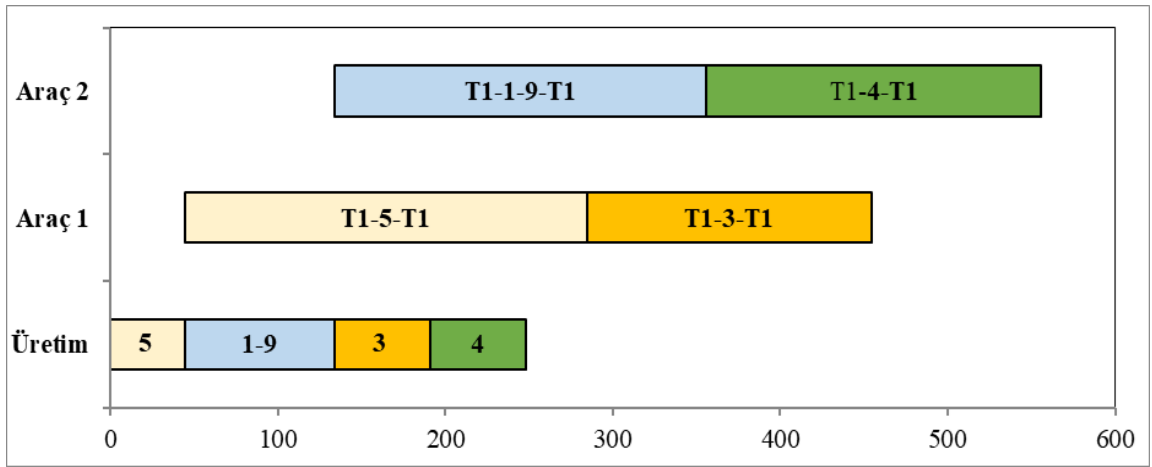
Her iki çocuk için de AFD hesaplandıktan sonra popülasyondaki mevcut bireylerle AFD'ler karşılaştırılarak, popülasyondaki en kötü AFD'ye sahip iki birey sistemden çıkarılarak bunların yerine çocuklar eklenir. Çocukların kötü olması durumunda ise popülasyonda herhangi bir değişiklik yapılmamaktadır. Buna göre Şekil 4.9'da popülasyondaki bireylerin ve yeni iki çocuğun AFD değerleri küçükten büyüğe sıralanarak verilmiştir.

	Atamalar											AFD
1. Birey	2	10	5	4	6	1	3	7	8	9	11	8929
3. Birey	5	9	6	10	1	7	3	4	11	2	8	9785
4. Birey	9	5	10	1	4	11	3	2	6	8	7	9972
Çocuk1	11	7	5	4	6	1	3	8	9	10	2	10455
Çocuk2	2	5	6	7	9	3	10	4	8	1	11	2332519
5. Birey	11	4	7	6	9	3	10	8	5	1	2	3783031
2. Birey	1	10	3	2	11	7	5	4	6	9	8	3922424

Şekil 4.9. Popülasyondaki Bireylerin AFD'ye Göre Sıralanmış Hali

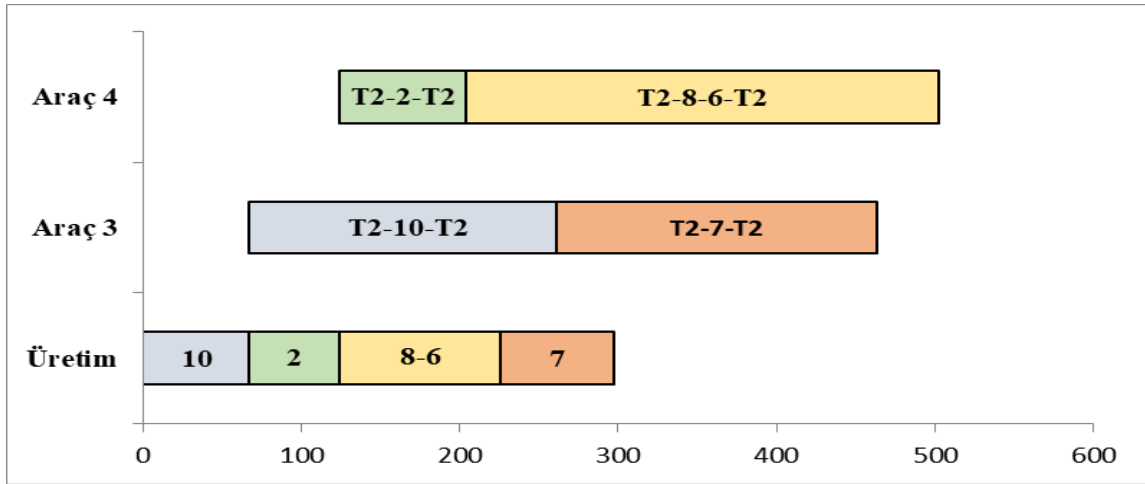
Şekil 4.9'da görüldüğü üzere AFD bakımından en iyi birey C&W algoritması ile oluşturulmuş olan 1.Bireydir. AFD değeri bakımından en kötü sonucu veren iki birey 2 ve 5 numaralı bireylerdir. Popülasyon büyüklüğünü korumak ve yeni iyileştirilmiş çözümlerle ilerleyebilmek için bu iki çözüm sistemden çıkarılacaktır. Kötü çözümlerin sistemden çıkarılması ile daha iyi çözümlerin bulunma olasılığı da artmaktadır. Yeni jenerasyonda Çocuk1, 2.Birey ve Çocuk2'de 5.Birey olacak şekilde algoritmaya devam

edilir. Tüm jenerasyon sonunda elde edilen en iyi çözüm T1 ve T2 için sırasıyla Şekil 4.10 ve Şekil 4.11’de gösterilmiştir.



Şekil 4.10. Problemin En İyi Çözümünde T1 Tesisinin Gantt Şeması

Şekil 4.10’da görüldüğü üzere T1 tesisine atanan müşteriler 1,3,4,5 ve 9 numaralı müşterilerdir. Bunlardan 1 ve 9 numaralı müşteriler tek rotada birleştirilmiş diğerleri ayrı rotalar olarak ele alınmıştır. Tüm müşterilerin taleplerinin üretimi için gereken süre 248 zaman birimidir. İlk olarak 5 numaralı müşterinin talebi üretilmiş ve araç tesiste hazır olarak beklediği için üretimi biter bitmez dağıtıma çıkmıştır. Aynı şekilde sonraki rota olan 1 ve 9 numaralı müşterilerin talepleri üretilmiş bu sırada ilk araç tesiste olmamasına rağmen ikinci araç tesiste hazır beklemektedir ve üretim biter bitmez dağıtım için yola çıkmıştır. Üçüncü rota olan 3 numaralı müşterinin talebinin üretimi tamamlandığı zaman tesise dönmüş herhangi bir araç yoktur. Bu nedenle araçların dönmesi için beklemek zorundadır. İlk olarak Araç1 tesise dönmektedir ve 3 numaralı müşterinin talebini teslim için yola çıkacaktır. Tüm bu işlemler için gerekli olan tamamlanma zamanı 556 zaman birimidir. T1 tesisinin AFD değeri 4386 para birimidir.



Şekil 4.11. Problemin En İyi Çözümünde T2 Tesisinin Gantt Şeması

Şekil 4.11’de görüldüğü üzere T2 tesisine atanan müşteriler 2, 6, 7, 8 ve 10 numaralı müşterilerdir. Bunlardan 6 ve 8 numaralı müşteriler tek rotada birleştirilmiş diğerleri ayrı rotalar olarak ele alınmıştır. Tüm müşterilerin taleplerinin üretimi için gereken süre 297 zaman birimidir. İlk olarak 10 numaralı müşterinin talebi üretilmiş ve araçlar tesiste hazır olarak beklediği için üretimi biter bitmez Araç 3 dağıtımına çıkmıştır. Aynı şekilde sonraki rota olan 2 numaralı müşterinin talebi üretilmiştir. Bu sırada araç 3 tesiste olmamasına rağmen araç 4 tesiste hazır beklemektedir ve üretim biter bitmez dağıtım için yola çıkmıştır. Üçüncü rota olan 8 ve 6 numaralı müşterilerin toplam taleplerinin üretimi tamamlandığı zaman Araç 4 tesise dönmüş durumdadır. Bu nedenle Araç 4 ile, 8 ve 6 müşterilerinin taleplerinin dağıtımını gerçekleştirilecektir. Tüm bu işlemler için gerekli olan tamamlanma zamanı 502 zaman birimidir. T2 tesisinin AFD değeri ise 4502 para birimidir. Problemin toplam AFD’si ise 8888 para birimidir. Bu değer matematiksel modelle elde edilen optimal değerdir.

4.4. ÇT_ÇA_BÜDÇ Problemi İçin Deney Tasarımı

Çalışmanın ilk aşamasında olduğu gibi, ÇT_ÇA_BÜDÇ probleminin çözümü için geliştirilen yöntemleri etkinliğini araştırmak amacıyla da Geismar ve ark. tarafından 2008 yılında üretilmiş olan test problemlerindeki yapıya benzer bir yapı kullanılarak test problemleri üretilmiştir. Bu test problemlerinde 2 farklı araç kapasitesi mevcuttur ($Q = 300, 600$). Ayrıca ürün ömrü için de yine $B = 300$ ve 600 olmak üzere 2 farklı durum söz konusudur. Üretim hızı $r = 1, 2, 3$ olmak üzere 3 farklı şekilde ele alınmıştır ancak her tesisin üretim hızı farklı olduğu için üretim hızları tesisler için farklılaştırılmıştır.

Örneğin $r = 1$ iken tesislerin hızları $r_k = 1 \pm 0.5$ olarak rassal hesaplanmakta ve her tesisin birbirinden farklı üretim hızına sahip olmaları sağlanmaktadır.

Müşteri taleplerinin (100 – 200) ve (100 – 300) arasında rassal oluşturulduğu 2 farklı durum mevcuttur. Müşterilerin konumları belirlenirken 3 farklı koordinat düzlemi kullanılmıştır. Bunlar; (-100×100) , (-150×150) ve (-200×200) koordinat düzlemidir.

Tesislerin belirli kapasiteleri vardır ve bu kapasiteler toplamı tüm müşteri taleplerinin toplamının 1.5 katına eşittir. Toplam tesis kapasitesinin yarısı tesis sayısına bölünerek elde edilen sayı tüm tesislere eşit olarak dağıtıldıktan sonra kalan talep miktarı ise rassal olarak dağıtılmaktadır. Bu durum birbirine yakın tesis kapasitelerinin elde edilmesi için önemli ve gereklidir.

Tesisler heterojen olduğu ve üretim hızları birbirinden farklı olduğu için her tesisdeki üretim maliyetleri de farklı olarak ele alınmıştır. Tüm müşterilerin kendisine en yakın tesise atandığı varsayımı ile toplam uzaklık belirlenmiş ve bu uzaklığın 8 katının toplam talep miktarına bölünmesi ile de üretim hızı en yavaş olan tesisdeki bir birim ürün üretme maliyeti belirlenmiştir. Tesis hızları arttıkça üretim maliyetinin de arttığı varsayımı ile elde edilen maliyete 0.1 birim eklenerek tüm tesislerin birim üretim maliyetleri belirlenmiştir.

Planlama ufku belirlenirken tüm müşterilerin en yakın tesise atandığı uzaklıklar dikkate alınarak ortalama uzaklık belirlenmiştir. Benzer şekilde toplam talebin depolara eşit dağıtıldığı ve üretildiği varsayımı ile taleplerin üretim zamanı belirlenmiş ve bu iki değer 2 katı planlama ufku olarak belirlenmiştir.

Araçların belirli bir kullanım maliyeti mevcuttur. Tesise atanan ve bir kez kullanımda ortaya çıkan bu maliyet aracın kaç kez kullanıldığından bağımsız olarak 100 birim olarak belirlenmiştir.

Müşteri sayıları küçük boyutlu problemler için 5 ve 10; orta boyutlu problemler için 15 ve 20, büyük boyutlu problemler içinse 50 ve 100 olarak belirlenmiştir. Küçük boyutlu problemlerde tesis sayısı 2, orta boyutlu problemlerde tesis sayısı 2 ve 3, büyük boyutlu problemler içinse tesis sayısı 3, 4 ve 5 olarak belirlenmiştir.

Her parametre kombinasyonu için 3 farklı test problemi üretilmiştir. Toplamda 5 müşterilik problemler için $1 \times 2 \times 2 \times 3 \times 2 \times 3 \times 3 = 216$ adet test problemi üretilmiştir. Orta boyutlu problemlerin her biri için 432 adet test problemi oluşturulmuştur. Orta boyutlu problemler toplamda $2 \times 432 = 864$ adet iken büyük boyutlu problemler içinse bu sayı 1296 adettir. Toplamda 2592 adet test problemi üretilmiştir.

4.5. ÇT_ÇA_BÜDÇ Problemi İçin Deneysel Karşılaştırmalar

Bu bölümde ÇT_ÇA_BÜDÇ probleminin çözümü için geliştirilmiş olan matematiksel modelin ve MA'nın sonuçları ele alınmış, elde edilen sonuçlar birbiriyle kıyaslanmıştır.

4.5.1. Önerilen matematiksel modelin performansı

Önerilen matematiksel model "GAMS" ara yüzünde kodlanmış ve matematiksel model çözücü olarak "CPLEX 20.1" kullanılmıştır. Bütün koşumlarda çözücünün varsayılan parametre seviyeleri kullanılmıştır. Her bir koşul "Intel Xeon (R) E-2236 (12 Core) 3.40 GHz" hızında "32 GB" ara belleğe sahip, "Windows 10" işletim sistemi ile çalışan bilgisayarda gerçekleştirilmiştir. Bütün koşullar 2 saat (7200 saniye) çözüm süresi ile sınırlandırılmıştır. Çizelge 4.7'de müşteri taleplerinin durumuna göre elde edilen sonuçlar verilmiştir.

Çizelge 4.7. Farklı Talepler İçin Deneysel Çalışmalar

	Çözüm Durumu	Adet	Ortalama Çözüm Süresi
D (100-200)	Optimal	423	476
	Uygun Çözüm	218	7200
	Çözüm	655	7200
	Bulunamayan		
	Toplam(Ortalama)	1296	(5005)
D (100-300)	Optimal	464	596.40
	Uygun Çözüm	178	7200
	Çözüm	654	7200
	Bulunamayan		
	Toplam(Ortalama)	1296	(4835)

ÇT_ÇA_BÜDÇ probleminin çözümü için geliştirilen matematiksel modelin çözümü için geliştirilen test problemlerindeki 2 farklı talep durumu için deney sonuçlarına bakıldığında taleplerin 100 ile 200 arasında olduğu durumda 423 adet optimal sonuç elde edilmiştir. Aynı şekilde taleplerin 100 ile 300 arasında olduğu durumda 464 adet optimal sonuç elde edilmiştir. D(100-300) için çözüm uzayının daha dar olması optimal çözüm bulunmasını da kolaylaştırmaktadır. D(100-200) için uygun çözüm bulunan problem sayısı 218 iken D(100-300) için bu sayı 178 adettir. Çözüm bulunamayan problem sayıları ise sırasıyla 655 ve 654 adettir. Ortalama çözüm süreleri

D(100-200) için 5005 saniye olarak hesaplanmışken, D(100-300) için bu süre 4835 saniyedir. Her iki talep durumu için genel süre ortalaması 4920 saniye olarak hesaplanmıştır.

Çizelge 4.8’de çözüm durumlarının müşteri sayılarına göre dağılımı gösterilmiştir. Buna göre küçük boyutlu problemlerin hepsinde optimal sonuçlar elde edilmiştir. 20 müşterilik problemlerden itibaren de çözüm bulunamayan problemler ortaya çıkmaktadır. 2592 adet test probleminin 887 adedinde optimal sonuçlar, 396 adet problemde ise uygun çözümler elde edilmiştir. Büyük boyutlu problemler içinse 7200 saniye süre sınırı içerisinde uygun çözüm bulunabilen 5 adet problem mevcuttur. Büyük boyutlu 1291 adet test probleminde ise herhangi bir uygun çözüm elde edilememiştir.

Çizelge 4.8. Müşteri Sayılarına Göre Deneysel Çalışmalar

Müşteri Sayısı	Çözüm Durumu			Toplam
	Optimal	Uygun Çözüm	Çözüm Bulunamayan	
5	216	-	-	216
10	216	-	-	216
15	289	143	-	432
20	166	248	18	432
50	-	5	643	648
100	-	-	648	648
Toplam	887	396	1309	2592

ÇT_ÇA_BÜDÇ probleminin etkinliğinin belirlenmesinde de ÇT_TA_BÜDÇ probleminde olduğu gibi YSD değeri kullanılmıştır. YSD değeri ÇT_TA_BÜDÇ problemi için kullanılan eşitliğin aynısı kullanılarak hesaplanmaktadır.

$$YSD = \left(\frac{Z^{\text{ÜS}} - Z^{\text{AS}}}{Z^{\text{ÜS}}} \right) * 100$$

Bu değer “0”a yakınlığı çözümün de en iyi çözüme o kadar yakın olduğu anlamına gelmektedir. Optimal veya uygun çözüm bulunan problemler için müşteri sayılarına ve müşteri taleplerine göre elde edilen YSD değerleri Çizelge 4.9’da verilmiştir.

Çizelge 4.9’da görüldüğü üzere her iki problem tipi için de müşteri sayısı arttıkça YSD değeri de artmaktadır. Özellikle büyük boyutlu problemlerde uygun çözüm bulabilmek oldukça zordur. YSD değeri müşteri talebinin 100-200 olduğu durumda

ortalama %0.60 iken 100-300 olduğu durumda ise %0.35 olarak belirlenmiştir. Bu da her iki problem için de çözüm elde edilebilen problemlerin optimal sonuca çok yakın sonuçlar elde ettiğini göstermektedir. Müşteri talebinin 100-300 arasında olduğu durumda araç kapasitesi ve ürün ömrü kısıtlarından dolayı çözüm uzayı daralmakta ve bu da daha kolay çözüm elde edilmesine neden olmaktadır.

Çizelge 4.9. Müşteri Taleplerine Göre YSD

	Müşteri Sayısı	YSD*	Ortalama Çözüm Süresi
D (100-200)	5	0.00	0.11
	10	0.00	31
	15	0.45	3129
	20	1.15	5174
	50	11.03	7200
	100	-	7200
	Ortalama	0.60	5005
D (100-300)	5	0.00	0.08
	10	0.00	17
	15	0.22	2518
	20	0.84	4821
	50	-	7200
	100	-	7200
	Ortalama	0.35	4835

*: Ortalama değerler üst sınır değerinin elde edildiği test problemleri kullanılarak elde edilmiştir.

Test problemlerindeki diğer parametrelerin değişimine göre elde edilen sonuçlar aşağıdaki çizelgelerde ayrıntılı şekilde ele alınmıştır. Buna göre ilk olarak tesis sayısının değişiminin sonuçlar üzerindeki etkisi Çizelge 4.10'da gösterilmektedir.

İlk problemde olduğu gibi üretilen test problemlerinde tesis sayısı için 4 farklı durum söz konusudur. Küçük boyutlu problemler için 2 adet tesis mevcutken, orta boyutlu problemler için 2 ve 3 adet tesis mevcuttur. Büyük boyutlu problemler için ise 3-4 ve 5 adet tesisin bulunduğu durumlar incelenmiştir.

Çizelge 4.10. Tesis Sayısına Göre Deneysel Sonuçlar

Tesis Sayısı	Çözüm Durumu			Toplam	Ortalama YSD*	Ortalama Çözüm Süresi	
	Optimal	Uygun Çözüm	Çözüm Bulunamayan				
D (100-200)	2	352	79	109	540	0.26	1594
	3	71	139	222	432	1.30	5160
	4	-	-	216	216	-	7200
	5	-	-	108	108	-	7200
	Toplam	423	218	655	1296	0.60	5005
D (100-300)	2	340	90	110	540	0.20	2928
	3	124	88	220	432	0.65	5446
	4	-	-	216	216	-	7200
	5	-	-	108	108	-	7200
	Toplam	464	178	654	1296	0.35	4835

*: Ortalama değerler üst sınır değerinin elde edildiği test problemleri kullanılarak elde edilmiştir.

Çizelge 4.10'da da görüldüğü üzere tesis sayısının 2 adet olduğu 1080 adet test probleminin 692 adedinde optimal sonuç elde edilmiştir. Yine tesis sayısının 2 adet olduğu problemler için optimal sonuca ulaşamayan problemler için ortalama YSD değeri %0.23 olarak hesaplanmıştır. Bu da optimal sonuca çok yakın sonuçlar elde edildiğini göstermektedir. Ortalama çözüm süresi ise 1714 saniye olarak hesaplanmıştır. Tesis sayısının 3 adet olduğu toplam 864 adet test probleminden ise 195 adedinde optimal sonuçlar elde edilmiştir. Bu 195 adet problemin 71 adedi D(100-200) olduğu talep durumunda elde edilmişken, 124 adedi ise D(100-200) talep durumunda elde edilmiştir. Tesis sayısının 4 ve 5 olduğu hiçbir durumda 7200 saniye süre sınırı içerisinde optimal ya da uygun çözüm bulunamamıştır. Tüm çözüm elde edilen problemler için elde edilen YSD değeri 0.48 iken, tüm problemlerin çözümü için gereken süre 4920 saniye olarak hesaplanmıştır.

Üretim hızının değişmesi de problemlerin performansını etkilemektedir. ÇT_TA_BÜDÇ problemi için geliştirilen test problemlerinde 3 farklı hız belirlenmiş ve sonuçlar raporlanmıştır. ÇT_ÇA_BÜDÇ probleminde ise her tesisin üretim hızı birbirinden farklı olduğu için problemler için belirlenen üretim hızlarının da tesisler için farklılaştırılması gerekmektedir. Bu nedenle üretim hızları yine 1,2 ve 3 olarak belirlenmiş fakat tesisler için bu üretim hızları ± 0.5 ile çeşitlendirilmiştir. Yani üretim hızının 1 olduğu test problemlerinde tesislerin üretim hızları 0.5 ile 1.5 arasında rassal olarak ele alınmıştır. Buna göre her tesisteki üretim süresi farklılık göstermektedir. Üretim hızı arttıkça müşteri taleplerinin üretimi için gerekli olan süre azalmaktadır.

Çizelge 4.11’de test problemlerinin sonuçlarının üretim hızına bağlı değişimi gösterilmiştir.

Çizelge 4.11’de görüldüğü üzere taleplerin D(100-300) olduğu durumda elde edilen optimal çözüm sayısı daha fazladır. D(100-300) için 464 adet problemde optimal çözüm elde edilirken bu sayı D(100-200) için 423 olarak hesaplanmıştır. Tesis hızının 1 olduğu durum iki talep durumu için toplam 864 adet problemin 321 adedinde optimal sonuç elde edilirken, tesis hızının 2 olduğu durum için bu sayı 323 adettir.

Çizelge 4.11. Üretim Hızına Göre Deneysel Sonuçlar

Üretim Hızı	Çözüm Durumu			Toplam	Ortalama YSD*	Ortalama Çözüm Süresi	
	Optimal	Uygun Çözüm	Çözüm Bulunamayan				
D (100-200)	1	153	61	218	432	0.59	4823
	2	154	62	216	432	0.53	4789
	3	116	95	221	432	0.69	5402
	Toplam	423	218	655	1296	0.60	5005
D (100-300)	1	168	45	219	432	0.35	4636
	2	169	45	218	432	0.27	4618
	3	127	88	217	432	0.43	5252
	Toplam	464	178	654	1296	0.35	4835

*: Ortalama değerler üst sınır değerinin elde edildiği test problemleri kullanılarak elde edilmiştir.

Tesis hızının 3 olduğu durumda ise 243 adet test probleminde optimal sonuç elde edilmiştir. Tesis hızının 3 olduğu problemler üretim süresinin en az olduğu problemlerdir. Üretim süresi kısa iken bu problem türünde dağıtım süreleri dolayısıyla koordinat düzlemi bağlayıcı rol oynamaktadır. Tesis hızı ve koordinat düzlemi arasındaki ilişki Çizelge 4.12’de gösterilmiştir.

Çizelge 4.12’de K1 müşteri koordinatlarının (-100,100) arasında rassal oluşturulduğu koordinat düzlemini, K2 müşteri koordinatlarının (-150,150) arasında rassal oluşturulduğu koordinat düzlemini ve K3 müşteri koordinatlarının (-200,200) arasında rassal oluşturulduğu koordinat düzlemini temsil etmektedir. Bu durumda müşterilerin birbirlerine en uzak konumlandırıldığı problem türü K3’tür. Üretim hızının fazla ve müşterilerin birbirine en uzak olduğu (K3) durumda elde edilen optimal çözüm sayısı 62’dir. Bu da koordinat düzleminin de çözüm üzerindeki etkisini açıkça göstermektedir. Üretim ne kadar hızlı olursa olsun aracın dağıtım için aracın dağıtımını gerçekleştirmesi için gereken zaman mesafeyle beraber arttığı için ürün ömrünün bağlayıcılığı da artmaktadır. Üretim hızı 1 iken ortalama YSD değeri %0.47, 2 iken %0.40

ve 3 iken %0.56 olarak hesaplanmıştır. Elde edilen ortalama YSD değerlerinde çok büyük farklar olmamasına rağmen tüm üretim hızları için K2 durumunda elde edilen ortalama YSD değerleri daha küçüktür.

Çizelge 4.12. Üretim Hızı ve Koordinat Düzlemine Göre Deneysel Sonuçlar

Üretim Hızı	Koordinat Düzlemi	Çözüm Durumu			Toplam	Ortalama YSD*	Ortalama Çözüm Süresi
		Optimal	Uygun Çözüm	Çözüm Bulunamayan			
1	K1	105	39	144	288	0.58	4753
	K2	109	34	145	288	0.41	4688
	K3	107	33	148	288	0.42	4749
	Toplam	321	106	437	864	0.47	4730
2	K1	118	28	142	288	0.47	4469
	K2	116	28	144	288	0.28	4569
	K3	89	51	148	288	0.45	5073
	Toplam	323	107	434	864	0.40	4704
3	K1	99	47	142	288	0.50	4928
	K2	82	60	146	288	0.45	5354
	K3	62	76	150	288	0.74	5699
	Toplam	243	183	438	864	0.56	5327

*: Ortalama değerler üst sınır değerinin elde edildiği test problemleri kullanılarak elde edilmiştir.

Araç kapasitesi ve talep durumu değişiminin test problemleri üzerindeki etkisi Çizelge 4.13'te gösterilmektedir.

Çizelge 4.13. Araç Kapasitesine Göre Deneysel Sonuçlar

Araç Kapasitesi	D	Çözüm Durumu			Toplam	Ortalama YSD*	Ortalama Çözüm Süresi
		Optimal	Uygun Çözüm	Çözüm Bulunamayan			
D (100-200)	300	238	82	328	648	0.18	4644
	600	185	136	327	648	1.02	5366
	Toplam	423	218	655	1296	0.60	5005
D (100-300)	300	254	68	326	648	0.17	453
	600	210	110	328	648	0.53	5140
	Toplam	464	178	654	1296	0.35	4835

*: Ortalama değerler üst sınır değerinin elde edildiği test problemleri kullanılarak elde edilmiştir.

Her iki talep durumu için de araç kapasitesinin 300 birim olduğu durumda 600 birim olduğu duruma göre daha fazla optimal sonuç elde edilmiştir. Talebin D(100-200) olduğu durumda araç kapasitesinin 300 birim ve 600 birim olduğu durumda optimal çözüm sayıları sırasıyla 238 ve 185 adettir. Araç kapasitesi 300 iken çözüm uzayının daha küçük

olması hem optimal çözüm bulmayı kolaylaştırmakta hem de daha iyi sonuçlar elde edilmesini sağlamaktadır. Benzer şekilde talebin D(100-300) olduğu durumda araç kapasitesinin 300 birim ve 600 birim olduğu durumda optimal çözüm sayıları sırasıyla 254 ve 210 adettir.

Çizelge 4.14. Ürün Ömrüne Göre Deneysel Sonuçlar

Ürün Ömrü	Çözüm Durumu			Toplam	Ortalama YSD*	Ortalama Çözüm Süresi	
	Optimal	Uygun Çözüm	Çözüm Bulunamayan				
D (100-200)	300	214	101	333	648	0.61	4999
	600	209	117	322	648	0.60	5010
	Toplam	423	218	655	1296	0.60	5005
D (100-300)	300	228	91	329	648	0.37	4926
	600	236	87	325	648	0.33	4745
	Toplam	464	178	654	1296	0.35	4835

*: Ortalama değerler üst sınır değerinin elde edildiği test problemleri kullanılarak elde edilmiştir.

Çizelge 4.14'te ürün ömrünün test problemleri üzerindeki etkisi gösterilmektedir. Tüm test problemlerinin ortalama çözüm süresi 4920 saniye olarak hesaplanırken, uygun çözüm elde edilebilen 1283 adet problem için ortalama YSD değeri %0.48'dir. Ürün ömrünün 300 birim olduğu problemler için toplam 442 adet problemde optimal sonuç elde edilmişken, ürün ömrünün 600 birim olduğu problemler için 445 adet test probleminde optimal sonuçlar elde edilmiştir. Ortalama YSD değerleri ve ortalama çözüm süreleri de her iki ürün ömrü alternatifi için de birbirlerine yakındır. 300 ve 600 birimlik ürün ömrü alternatifleri ve D(100-200) talep durumu ve için ortalama YSD alternatifleri sırasıyla %0.61 ve %0.60 olarak hesaplanmışken, benzer şekilde 300 ve 600 birimlik ürün ömrü alternatifleri ve D(100-300) talep durumu ve için ortalama YSD alternatifleri sırasıyla %0.37 ve %0.33 olarak hesaplanmıştır.

Test problemlerinin sonuçlarının koordinat düzlemine göre değerlendirilmesi Çizelge 4.15'te gösterilmektedir. Ürün ömrünün etkisine benzer şekilde koordinat düzleminin değişik durumları için de birbirine yakın sonuçlar elde edilmiştir. Koordinat düzlemi genişledikçe çözüm uzayı da genişlemekteyken ürün ömrü kısıtından dolayı aynı rotada müşterileri birleştirmek zorlaşmaktadır. Benzer şekilde koordinat düzlemi daraldıkça rotada birleşebilecek müşteri sayıları artarken bu sefer araç kapasitesi bağlayıcı kısıt haline gelmektedir.

Çizelge 4.15. Koordinat Düzlemine Göre Deneysel Sonuçlar

Koordinat Düzlemi	Çözüm Durumu			Toplam	Ortalama YSD*	Ortalama Çözüm Süresi	
	Optimal	Uygun Çözüm	Çözüm Bulunamayan				
D (100-200)	1	149	71	212	432	0.72	4844
	2	147	68	217	432	0.45	4994
	3	127	79	226	432	0.64	5176
	Toplam	423	218	655	1296	0.60	5005
D (100-300)	1	173	43	216	432	0.31	4589
	2	160	54	218	432	0.31	4747
	3	131	81	220	432	0.43	5170
	Toplam	464	178	654	1296	0.35	4835

*: Ortalama değerler üst sınır değerinin elde edildiği test problemleri kullanılarak elde edilmiştir.

Çizelge 4.16'da görüldüğü üzere küçük ve orta boyutlu problemler için matematiksel model ile optimal ya da optimal sonuca yakın çözümler elde edilmiştir.

Çizelge 4.16. Geçerli Eşitsizliğin Matematiksel Model Üzerindeki Etkisi

Parametre	Seviye	MP							rMP + GE		rMP	
		Optimal Çözüm	Uygun Çözüm	Çözüm Bulunamayan	Alt Sınır	GAP (1283/2592)	Çözüm Süresi	Ç. Süresi (opt)	Alt Sınır	GAP (1283/2592)	Alt Sınır	GAP (1283/2592)
Müşteri Sayısı	5	216	0	0	3834.79	0.00	0.09	0.09	3714.19	3.20	3648.31	5.08
	10	216	0	0	7373.93	0.00	24.59	24.59	7162.46	2.95	7011.54	5.02
	15	289	143	0	10650.65	0.34	2823.91	658.58	10286.11	3.77	10054.31	6.02
	20	166	248	18	14560.80	0.99	5087.07	1701.29	14149.99	3.82	13803.31	6.26
	50	0	5	643	35210.19	11.03	7200	-	34683.47	11.18	33776.46	13.15
	100	0	0	648	-	-	7200	-	-	-	-	-
Talep Aralığı	(100-200)	423	218	655	18353.88	0.60	5005.36	475.99	17972.57	3.86	17459.78	6.15
	(100-300)	464	178	654	18815.50	0.35	4835.75	596.40	18427.27	3.30	18030.11	5.44
Araç Kapasitesi	300	492	150	654	19341.96	0.18	4587.91	319.37	18820.07	3.11	18342.85	5.51
	600	395	246	655	17827.42	0.78	5253.20	812.51	17579.76	4.05	17147.04	6.08
Ürün Ömrü	300	442	192	662	18585.39	0.49	4963.09	641.11	18202.03	3.61	17746.33	5.84
	600	445	204	647	18583.99	0.47	4878.01	437.54	18197.81	3.55	17743.56	5.76
Uzay Boyutu	(-100 x 100)	322	114	428	13104.55	0.52	4716.92	537.34	12810.34	3.89	12452.99	6.41
	(-150 x 150)	307	122	435	19365.50	0.38	4870.94	645.26	18979.46	3.30	18493.98	5.44
	(-200 x 200)	258	160	446	23284.02	0.53	5173.79	414.56	22809.95	3.55	22287.85	5.52
Üretim Hızı	1	321	106	437	18581.16	0.47	4730.19	552.28	18202.14	3.48	17746.32	5.70
	2	323	107	434	18576.72	0.40	4704.13	523.74	18197.22	3.45	17742.21	5.66
	3	243	183	438	18596.19	0.56	5327.34	541.66	18200.39	3.82	17746.30	6.03
Toplam (Ortalama)		887	396	1309	18584.69	0.48	4920.55	538.98	18199.92	3.58	17744.94	5.80

Küçük boyutlu tüm problemler için optimal sonuçlar elde edilirken, orta boyutlu problemlerin 391 adedinde 7200 saniye süre kısıtı içerisinde herhangi uygun bir çözüm bulunamamıştır. Toplam 2592 adet test probleminin 887 adedinde optimal sonuç, 396 adet test probleminde ise 7200 saniye süre kısıtı içerisinde uygun bir çözüm elde edilmiştir. Toplam 2592 adet test probleminin ortalama çözüm süresi 4920.55 saniye iken, optimal çözümlerin elde edilme süresi 538.98 saniyedir. Uygun çözüm bulunan

1283 adet problem ile hesaplanan ortalama YSD değeri 0.48'dir. Bu durumda optimal çözüme çok yakın sonuçlar elde edildiği açıkça söylenebilir. Matematiksel modelin 7200 saniye içerisinde elde ettiği alt sınır değeri ortalama 18584.69'dur. Bu değer geçerli eşitsizliklerin olmadığı durumda (rMIP) gevşetilmiş olarak çözüm elde edildiği zaman 17744.94 olarak hesaplanmıştır. Matematiksel modele geçerli eşitsizliklerin eklenmesiyle (rMIP + GE) elde edilen alt sınır 18199.92'dir. Çizelgede de açıkça görüldüğü gibi matematiksel modeli geçerli eşitsizlikler ile kuvvetlendirmek çok daha iyi alt sınırlar elde edilmesini sağlamış bu da çözüm süresi ve kalitesinin etkilemiştir. Modele geçerli eşitsizlik eklenmesiyle elde edilen ortalama alt sınır geçerli eşitsizlik eklenmeden elde edilen alt sınırdan %2.30 daha iyidir.

4.5.2. Matematiksel model ve MA ile elde edilen çözümlerin karşılaştırılması

Bu bölümde geliştirilmiş olan matematiksel model ile memetik algoritmanın sonuçları karşılaştırmalı olarak verilmiştir. MA için, farklı başlangıç değerleri kullanılarak 5 farklı çözüm elde edilmiş ve bu elde edilen çözümler içerisinde en küçük, en büyük ve ortalamalar raporlanmıştır. Buna göre çok kısa süreler içerisinde tüm problemler için uygun çözümler elde edilmiştir. Çizelge 4.17'de matematiksel model ile MA ile elde edilen çözümlerin değişik parametrelere göre sonuçları verilmiştir.

Çizelge 4.17. Matematiksel Model ve MA'nın Deneysel Karşılaştırılması

Parametre	Seviye	MIP						MA			
		Optimal Çözüm	Uygun Çözüm	Çözüm Bulunamayan	YSD (1283/2592)	Çözüm Süresi	Ç. Süresi (Opt)	YSD (EK)	YSD (ORT)	YSD (EB)	Çözüm Süresi
Müşteri Sayısı	5	216	0	0	0.00	0.09	0.09	0.00	0.03	0.11	0.02
	10	216	0	0	0.00	24.59	24.59	0.00	0.07	0.17	0.04
	15	289	143	0	0.34	2823.91	658.38	0.00	0.11	0.28	0.09
	20	166	248	18	0.99	5087.07	1701.29	-0.16	-0.06	0.09	0.21
	50	0	5	643	11.03	7200	-	-7.37	-7.17	-6.89	2.28
	100	0	0	648	-	7200	-	-	-	-	9.52
Talep Aralığı	(100-200)	423	218	655	0.60	5005.36	475.98719	-0.10	0.00	0.17	3.15
	(100-300)	464	178	654	0.35	4835.75	596.40	-0.06	0.02	0.11	2.87
Araç Kapasitesi	300	492	150	654	0.18	4587.91	319.37	-0.02	0.03	0.11	3.17
	600	395	246	655	0.78	5253.20	812.51	-0.14	-0.02	0.17	2.84
Ürün Ömrü	300	442	192	662	0.49	4963.09	641.11	-0.10	-0.02	0.12	3.18
	600	445	204	647	0.47	4878.01	437.54	-0.06	0.03	0.17	2.83
Koordinat Düzlemi	(-100 x 100)	322	114	428	0.52	4716.92	537.34	-0.12	-0.01	0.15	2.55
	(-150 x 150)	307	122	435	0.38	4870.94	645.26	-0.04	0.02	0.11	3.04
	(-200 x 200)	258	160	446	0.53	5173.79	414.56	-0.08	0.02	0.17	3.44
Üretim Hızı	1	321	106	437	0.47	4730.19	552.28	-0.11	0.00	0.16	2.36
	2	323	107	434	0.40	4704.13	523.74	-0.06	0.00	0.11	3.06
	3	243	183	438	0.56	5327.34	541.66	-0.06	0.02	0.15	3.60
Toplam (Ortalama)		887	396	1309	0.48	4920.55	538.98	-0.08	0.01	0.14	3.01

Deneysel çalışmalar farklı parametre seviyeleri için ele alınmış ve sonuçlar raporlanmıştır. 2592 adet test probleminin 1283 adedinde optimal ya da uygun çözümler elde edilmiştir. Dolayısıyla YSD değerleri alt sınır elde edilen bu problemler için hesaplanmıştır. Matematiksel modele göre elde edilen YSD değeri %0.48 olarak hesaplanmıştır. YSD değerinin 0'a çok yakın olması optimal sonuca çok yakın sonuçlar elde edildiğini göstermektedir. Bu sonuçların elde edilebilmesi için geçen ortalama süre 4920.55 saniyedir. Bu da yaklaşık 82 dakika etmektedir. Optimal çözümler ise ortalama 538.98 saniyede elde edilmiştir. Bu süre de yaklaşık olarak 9 dakikadır. MA ile bu çözümlerin elde edilmesi için gereken ortalama süre 3 saniyedir. Bu da algoritmanın ne kadar hızlı ve doğru sonuçlar verdiğini göstermektedir. Problemlerin MA ile çözümünden elde edilen en küçük değerlerin matematiksel model ile aralarındaki fark YSD (EK) ile gösterilmektedir. Buradaki sonuçlara göre Matematiksel model ve MA birbirine çok yakın sonuçlar vermektedir. Elde edilen en küçük değerler göz önüne alındığında MA %0.08 daha iyi sonuç vermektedir. 5 farklı çözümün ortalaması göz önüne alındığında ise matematiksel model çözümlerinin ve MA çözümlerinden %0.01 daha iyi olduğu söylenebilir. Aynı ya da çok yakın sonuçlara MA ile çok kısa sürelerde ulaşılması mümkündür. MA ile çözümü için gereken süre en küçük 0.01 saniye iken en uzun sürede

çözülen problem 26 saniyede çözülmüştür. Ayrıca MA ile tüm problemler için uygun bir çözüm elde edilmişken matematiksel model ile çözüm elde edilemeyen 1309 adet problem mevcuttur. Bu nedenle MA algoritmasının matematiksel modele göre çok daha başarılı olduğu söylenebilir. Müşteri sayısı arttıkça matematiksel model ile uygun çözüm elde etmek zorlaşırken MA ile matematiksel modelden çok daha iyi sonuçlar elde edilmiştir. Matematiksel model ile 50 müşterilik 5 problem için uygun bir çözüm elde edilmiş bu problemlerin YSD değeri %11.03 olarak hesaplanmıştır. MA ile bu 5 problem için %7.17 daha iyi sonuçlar elde edilmiştir. Diğer tüm parametreler için MA ile elde edilen en küçük değerler matematiksel model ile elde edilen değerlerden daha iyidir.

MA'nın performansı ise Çizelge 4.18'de ayrıntılı olarak ele alınmıştır. Çizelge 4.18'de görüldüğü üzere ilk iki sütun problemin performansını değerlendirebileceğimiz değişik parametreleri ve bu parametrelerin aldığı değerleri göstermektedir. Sonraki üç sütun için EK ile belirtilen sütunda 2592 adet test probleminin MA ile çözümünden elde edilen sonuçlar içerisindeki en küçük değerlerin ortalamasını göstermektedir. Benzer şekilde ORT sütunu ortalama değerleri ve EB sütunu ise test problemlerinin çözümünden elde edilen en büyük sonuçları göstermektedir. YSD ile gösterilen sütunda ise MA'dan elde edilen sonuçlardan EB ile EK arasındaki farkın % değeri hesaplanmıştır. Son sütun ise bu işlemler için gereken ortalama süreyi göstermektedir.

Çizelge 4.18. MA'nın Performansı

Parametre	Değer	MA				
		EK	ORT	EB	YSD	Çözüm Süresi
Müşteri Sayısı	5	3834.85	3836.01	3838.80	0.11	0.02
	10	7374.30	7378.90	7386.56	0.17	0.04
	15	10690.10	10701.84	10720.56	0.27	0.09
	20	14693.67	14709.54	14732.04	0.26	0.21
	50	36422.81	36558.59	36699.53	0.72	2.28
	100	71134.93	71904.28	72630.99	1.97	9.52
Talep Aralığı	(100-200)	32000.24	32260.52	32513.60	0.89	3.15
	(100-300)	32108.07	32310.53	32506.75	0.68	2.87
Araç Kapasitesi	300	33178.94	33420.70	33655.04	0.74	3.17
	600	30929.38	31150.34	31365.31	0.83	2.84
Ürün Ömrü	300	32164.04	32467.74	32748.27	0.98	3.18
	600	31944.28	32103.30	32272.08	0.60	2.83
Koordinat Düzlemi	(-100 x 100)	22733.04	22829.51	22932.39	0.57	2.55
	(-150 x 150)	33497.59	33713.91	33934.84	0.72	3.04
	(-200 x 200)	39931.85	40313.15	40663.30	1.07	3.44
Üretim Hızı	1	31986.92	32153.73	32313.06	0.61	2.36
	2	32087.19	32314.07	32533.59	0.75	3.06
	3	32088.36	32388.76	32683.87	1.00	3.60
Ortalama		32054.16	32285.52	32510.18	0.79	3.01

Buna göre MA'nın 5 farklı rassal deęer ile özölmesinden elde edilen en küçük deęerler ortalama 32054.16, ortalama 32285.52 ve en büyük 32510.18 olarak hesaplanmıřtır. Elde edilen deęerlerden en büyük ile en küçük deęerler arasında %0.79'luk bir fark ortaya çıkmaktadır. Bu durumda en büyük ve en küçük deęerler arasında ok büyük farklar olmadıęı ve özömlerin birbirlerine yakın olduęu söylenebilir. Müřteri sayıları arttıa YSD deęerleri de artmaktadır. Elde edilen en büyük YSD deęeri 100 müřterili problemlerin özümünden elde edilen %1.97'lik deęerdir. 100 müřterilik bu problemlerin özümü için gereken süre ise ortalama yaklaşık 10 saniyedir. ok kısa sürelerde elde edilen bu özömler MA'nın matematiksel modele göre hızlı ve etkin bir algoritma olduęunu göstermektedir.



5. SONUÇLAR VE ÖNERİLER

Bu tez çalışmasında ilk olarak sistemde birden fazla tesisin bulunduğu ve müşterilere her tesiste bulunan tek araçla hizmet edilen ÇT_TA_BÜDÇ problemi ele alınmıştır. Belirli varsayımlar altında ÇT_TA_BÜDÇ probleminde amacımız müşteri siparişlerinin üretilip dağıtımının gerçekleştirileceği en kısa zaman içerisinde, hangi müşterinin hangi depodan hizmet göreceğini, hangi müşteri siparişlerinin hangi parti içerisinde olacağını, her bir partinin üretimine ne zaman başlanıp bitirileceğini ve her bir tur içerisinde müşterilerin hangi sırada ziyaret edileceğini tespit etmektir. Bu amaç doğrultusunda her müşteriye yalnızca bir kere ziyaret gerçekleştirilmeli, araca yüklenen ürünlerin toplam alan, hacim, ağırlık vb. gereksinimi araç kapasitesini geçmemeli, her rota bir tesiste başlamalı ve aynı tesiste sonlanmalı ve bir partinin üretimi tamamlandıktan sonra o parti içerisinde bulunan tüm ürünler (siparişler) ilgili müşterilere B birimlik zaman içerisinde teslim edilmelidir. Problemin çözümü için öncelikle bir matematiksel model geliştirilmiştir. Hem makine çizelgeleme hem de araç rotalama problemi NP-zor problem sınıfında yer aldığı için BÜDÇ probleminin de NP-zor yapıda olduğu bilinmektedir. Bu nedenle büyük boyutlu problemler için daha kısa sürelerde optimal ya da optimale yakın çözümler elde edilebilmesi amacıyla DKA Algoritması geliştirilmiştir.

Geliştirilen matematiksel model ve DKA algoritması 6 farklı parametre üzerinden değerlendirilmiştir. Geliştirilen 2592 adet test probleminin matematiksel model ile çözülmesi sonucunda 7200 sn süre sınırı içerisinde 833 adet test probleminde çözüm elde edilememişken DKA algoritması tüm problemler için uygun bir çözüm bulunmuştur.

Matematiksel model ile 74 adet test probleminde optimal sonuçlar elde edilmiştir ve bu problemlerin hepsi 10 müşterili problemlerdir. Optimal sonuçların elde edildiği problemler için ortalama çözüm süresi 3085 saniyedir. DKA algoritması ile matematiksel model ile elde edilen sonuçlardan daha iyi ya da o sonuçlara yakın sonuçlar 1 saniyenin altında bir sürede elde edilmiştir. Matematiksel model ile DKA algoritmasının sonuçları karşılaştırıldığı zaman elde edilen YSD değerleri arasındaki fark ortalama %15.16 olarak hesaplanmıştır. Matematiksel modelin çözümü için gereken süre ortalama 7083 saniye iken bu süre DKA algoritmasında yaklaşık 70 saniyedir. DKA algoritmasının hem süre hem de performans açısından çok daha iyi sonuçlar sağlamaktadır.

Değişken komşu arama algoritması ile matematiksel modelle elde edilen optimal çözümlerin hepsi çok daha kısa sürelerde elde edilmiştir. Ayrıca DKA algoritması kullanılarak çözülen tüm problemlerde uygun bir çözüm elde edilmiştir. DKA

algoritmasındaki çözüm sürelerinin özellikle küçük ve orta boyutlu problemler için çok kısa olduğu da açıkça görülmektedir. Çözüm süresinin 10 dakikanın üzerinde olduğu problem sayısı D(100-200) için 37 adet ve D(100-300) için 34 adettir. Müşteri sayısı arttıkça çözüm süresinin arttığı da açıkça görülmektedir.

Matematiksel modelde elde edilen tüm optimal sonuçların araç kapasitesinin 300 birim olduğu durumlar için olduğu bilinmektedir. Araç kapasitesinin 600 olduğu hiçbir problemde optimal çözüme ulaşılamamıştır. Matematiksel modelde optimal çözümlere ulaşmak için gereken ortalama süre 1 saate yakınken bu sonuçlar DKA algoritması ile milisaniye ile ifade edilebilecek kadar kısa sürelerde bulunabilmektedir.

Matematiksel modelde D(100-200) olduğu test problemleri için harcanan süre ortalama 7099.63 saniye iken D(100-300) olduğu problemler için bu süre 6959.35 saniyedir. DKA algoritmasında ortalama sürelerin yaklaşık 1 dakika civarında olduğu göz önüne alınacak olursa algoritmanın hızlı ve doğru sonuçlar verdiği açıkça söylenebilir.

Tez çalışmasında çalışılan bir diğer problem ise yine sistemde çok tesis bulunduğu fakat tesislerde birden çok aracın mevcut olduğu durumu göz önüne alındığı (ÇT_ÇA_BÜDÇ) problemidir. Belirli varsayımlar altında bu problemde amacımız, müşteri siparişlerinin üretilip dağıtımının gerçekleştirileceği en az maliyet ile, hangi müşterinin hangi tesisten ve hangi araçla hizmet göreceğini, hangi müşteri siparişlerinin hangi parti içerisinde olacağını, her bir partinin üretimine ne zaman başlanıp bitirileceğini, sistemde her tesis için kaç araç kullanılacağını ve her bir tur içerisinde müşterilerin hangi sırada ziyaret edileceğini tespit etmektir. Bu amaç doğrultusunda her müşteriye yalnızca bir kere ziyaret gerçekleştirilmeli, Araçlara yüklenen ürünlerin toplam alan, hacim, ağırlık vb. gereksinimi araç kapasitesini aşmamalı, her rota bir tesiste başlamalı ve aynı tesiste sonlanmalı, bir partinin üretimi tamamlandıktan sonra o parti içerisinde bulunan tüm ürünler ilgili müşterilere B birimlik zaman içerisinde teslim edilmeli, bir tesise atanan müşterilerin talepleri o tesisin kapasitesini geçmemeli ve tüm üretim ve dağıtım işlemleri planlama ufkundan önce bitirilmelidir. Problemin çözümü için öncelikle bir matematiksel model geliştirilmiştir. Hem makine çizelgeleme hem de araç rotalama problemi NP-zor problem sınıfında yer aldığı için BÜDÇ probleminin de NP-zor yapıda olduğu bilinmektedir. Bu nedenle büyük boyutlu problemler için daha kısa sürelerde optimal ya da optimale yakın çözümler elde edilebilmesi amacıyla Memetik Algoritma (MA) geliştirilmiştir.

ÇT_ÇA_BÜDÇ için gerçekleştirilen deneysel çalışmalar farklı parametre seviyeleri için ele alınmış ve sonuçlar raporlanmıştır. 2592 adet test probleminin 1283

adedinde optimal ya da uygun çözümler elde edilmiştir. Matematiksel modele göre elde edilen YSD değeri %0.48 olarak hesaplanmıştır. YSD değerinin 0'a çok yakın olması optimal sonuca çok yakın sonuçlar elde edildiğini göstermektedir. Bu sonuçların elde edilebilmesi için geçen ortalama süre 4920.55 saniyedir. Bu da yaklaşık 82 dakika etmektedir. Optimal çözümler ise ortalama 538.98 saniyede elde edilmiştir. Bu süre de yaklaşık olarak 9 dakikadır. MA ile bu çözümlerin elde edilmesi için gereken ortalama süre 3 saniyedir. Bu da algoritmanın ne kadar hızlı ve doğru sonuçlar verdiğini göstermektedir. Problemlerin MA ile çözümünden elde edilen en küçük değerlerin matematiksel model ile aralarındaki fark YSD (EK) ile gösterilmektedir. Buradaki sonuçlara göre Matematiksel model ve MA birbirine çok yakın sonuçlar vermektedir. Elde edilen en küçük değerler göz önüne alındığında MA %0.08 daha iyi sonuç vermektedir. 5 farklı çözümün ortalaması göz önüne alındığında ise matematiksel model çözümlerinin ve MA çözümlerinden %0.01 daha iyi olduğu söylenebilir. Aynı ya da çok yakın sonuçlara MA ile çok kısa sürelerde ulaşılması mümkündür. MA algoritmasının çözümü için gereken süre en küçük 0.01 saniye iken en uzun sürede çözülen problem 26 saniyede çözülmüştür. Ayrıca MA ile tüm problemler için uygun bir çözüm elde edilmişken matematiksel model ile çözüm elde edilemeyen 1309 adet problem mevcuttur. Bu nedenle MA algoritmasının matematiksel modele göre çok daha başarılı olduğu söylenebilir.

Müşteri sayısı arttıkça matematiksel model ile uygun çözüm elde etmek zorlaşırken MA ile matematiksel modelden çok daha iyi sonuçlar elde edilmiştir. Matematiksel model ile 50 müşterilik 5 problem için uygun bir çözüm elde edilmiş bu problemlerin YSD değeri %11.03 olarak hesaplanmıştır. MA ile bu 5 problem için %7.17 daha iyi sonuçlar elde edilmiştir. Diğer tüm parametreler için MA ile elde edilen en küçük değerler matematiksel model ile elde edilen değerlerden daha iyidir.

Tez kapsamında ele alınan BÜDÇ Problemleri ile ilgili ilerleyen çalışmalarda aşağıdaki hususlar dikkate alınabilir:

- (i) BÜDÇ problemlerinin çok tesisli olarak ele alındığı durumda üretim ortamları çeşitlendirilebilir. Çok tesisli ve paralel makineli üretim ortamı, çok tesisli akış tipi üretim ortamı ve çok tesisli atölye tipi üretim ortamına sahip problemler çalışılabilir.
- (ii) Yine benzer şekilde dağıtımın çeşitlendirilmesi de söz konusudur. Zaman pencereleri, zamana bağımlı dağıtım süreleri, toplama ve dağıtımın aynı

anda gerçekleştirildiği vs. durumlar göz önüne alınarak dağıtım ortamı çeşitlendirilebilir.

- (iii) ÇT_TA_BÜDÇ problemi için geliştirilen DKA algoritması ve ÇT_ÇA_BÜDÇ problemi için geliştirilen MA algoritmasından farklı olarak yine kısa sürelerde optimal ya da optimale yakın sonuçlar verecek metasezgisel algoritmalar geliştirilebilir.
- (iv) BÜDÇ problemleri için, elde edilen çözümleri iyileştirmek ve en iyi çözümleri elde edebilmek amacıyla, önerilen matematiksel modele, sezgisel yöntem ve geçerli eşitsizlikler (valid inequalities) eklenerek bir kesin algoritma geliştirilebilir.
- (v) Tez kapsamında incelenen problemlerde tüm parametreler belirli ve önceden bilinmektedir. Ancak böyle bir durum günlük hayatın olağan akışına pek uygun değildir. Bu nedenle bazı parametrelerin stokastik olduğu durumlar göz önüne alınabilir.

KAYNAKLAR

- Almada-Lobo, B., Oliveira, J. F. ve Carravilla, M. A., 2008, Production planning and scheduling in the glass container industry: A VNS approach, *International Journal of Production Economics*, 114 (1), 363-375.
- Amorim, P., Belo-Filho, M., Toledo, F., Almeder, C. ve Almada-Lobo, B., 2013, Lot sizing versus batching in the production and distribution planning of perishable goods, *International Journal of Production Economics*, 146 (1), 208-218.
- Armstrong, R., Gao, S. ve Lei, L., 2008, A zero-inventory production and distribution problem with a fixed customer sequence, *Annals of Operations Research*, 159 (1), 395-414.
- Belo-Filho, M., Amorim, P. ve Almada-Lobo, B., 2015, An adaptive large neighbourhood search for the operational integrated production and distribution problem of perishable products, *International Journal of Production Research*, 53 (20), 6040-6058.
- Bilgen, B. ve Ozkarahan, I., 2004, Strategic tactical and operational production-distribution models: a review, *International Journal of Technology Management*, 28 (2), 151-171.
- Camargo, V. C., Toledo, F. M. ve Almada-Lobo, B., 2014, HOPS–Hamming-Oriented Partition Search for production planning in the spinning industry, *European journal of operational research*, 234 (1), 266-277.
- Chang, Y.-C. ve Lee, C.-Y., 2004, Machine scheduling with job delivery coordination, *European Journal of Operational Research*, 158 (2), 470-487.
- Chang, Y.-C., Li, V. C. ve Chiang, C.-J., 2014, An ant colony optimization heuristic for an integrated production and distribution scheduling problem, *Engineering Optimization*, 46 (4), 503-520.
- Chen, H.-K., Hsueh, C.-F. ve Chang, M.-S., 2009, Production scheduling and vehicle routing with time windows for perishable food products, *Computers & operations research*, 36 (7), 2311-2319.
- Chen, Z.-L., 2004, Integrated production and distribution operations, *Handbook of quantitative supply chain analysis*, 711-745.
- Chen, Z.-L. ve Vairaktarakis, G. L., 2005, Integrated scheduling of production and distribution operations, *Management Science*, 51 (4), 614-628.
- Chen, Z.-L., 2010, Integrated production and outbound distribution scheduling: review and extensions, *Operations research*, 58 (1), 130-148.
- Cheref, A., Artigues, C. ve Billaut, J.-C., 2016, Online recoverable robustness based on groups of permutable jobs for integrated production scheduling and delivery routing.
- Chiang, W.-C., Russell, R., Xu, X. ve Zepeda, D., 2009, A simulation/metaheuristic approach to newspaper production and distribution supply chain problems, *International Journal of Production Economics*, 121 (2), 752-767.
- Clarke, G. ve Wright, J. W., 1964, Scheduling of vehicles from a central depot to a number of delivery points, *Operations research*, 12 (4), 568-581.
- Condotta, A., Knust, S., Meier, D. ve Shakhlevich, N. V., 2013, Tabu search and lower bounds for a combined production–transportation problem, *Computers & Operations Research*, 40 (3), 886-900.
- Devapriya, P., Ferrell, W. ve Geismar, N., 2006, Optimal fleet size of an integrated production and distribution scheduling problem for a perishable product, *IIE Annual Conference. Proceedings*, 1.

- Devapriya, P., Ferrell, W. ve Geismar, N., 2017, Integrated production and distribution scheduling with a perishable product, *European Journal of Operational Research*, 259 (3), 906-916.
- Erengüç, Ş. S., Simpson, N. C. ve Vakharia, A. J., 1999, Integrated production/distribution planning in supply chains: An invited review, *European journal of operational research*, 115 (2), 219-236.
- Fu, L.-L., Aloulou, M. A. ve Triki, C., 2017, Integrated production scheduling and vehicle routing problem with job splitting and delivery time windows, *International Journal of Production Research*, 55 (20), 5942-5957.
- Ganji, M., Kazemipoor, H., Molana, S. M. H. ve Sajadi, S. M., 2020, A green multi-objective integrated scheduling of production and distribution with heterogeneous fleet vehicle routing and time windows, *Journal of Cleaner Production*, 259, 120824.
- Garcia, J., Lozano, S. ve Canca, D., 2004, Coordinated scheduling of production and delivery from multiple plants, *Robotics and Computer-Integrated Manufacturing*, 20 (3), 191-198.
- García, J. M. ve Lozano, S., 2004, Production and vehicle scheduling for ready-mix operations, *Computers & Industrial Engineering*, 46 (4), 803-816.
- García, J. M. ve Lozano, S., 2005, Production and delivery scheduling problem with time windows, *Computers & Industrial Engineering*, 48 (4), 733-742.
- Geismar, H. N., Laporte, G., Lei, L. ve Sriskandarajah, C., 2008, The integrated production and transportation scheduling problem for a product with a short lifespan, *INFORMS Journal on Computing*, 20 (1), 21-33.
- Geismar, H. N., Dawande, M. ve Sriskandarajah, C., 2011, Pool-point distribution of zero-inventory products, *Production and Operations Management*, 20 (5), 737-753.
- Gen, M. ve Cheng, R., 1999, Genetic algorithms and engineering optimization, John Wiley & Sons, p.
- Gharaei, A. ve Jolai, F., 2018, A multi-agent approach to the integrated production scheduling and distribution problem in multi-factory supply chain, *Applied Soft Computing*, 65, 577-589.
- Hajiaghaei-Keshteli, M., Aminnayeri, M. ve Ghomi, S. F., 2014, Integrated scheduling of production and rail transportation, *Computers & Industrial Engineering*, 74, 240-256.
- Hansen, P. ve Mladenović, N., 2001, Variable neighborhood search: Principles and applications, *European journal of operational research*, 130 (3), 449-467.
- He, P., Li, K. ve Kumar, P. R., 2021, An enhanced branch-and-price algorithm for the integrated production and transportation scheduling problem, *International Journal of Production Research*, 1-16.
- Hurter, A. P. ve Van Buer, M. G., 1996, The newspaper production/distribution problem, *Journal of Business Logistics*, 17 (1), 85.
- Jamili, N., Ranjbar, M. ve Salari, M., 2016, A bi-objective model for integrated scheduling of production and distribution in a supply chain with order release date restrictions, *Journal of Manufacturing Systems*, 40, 105-118.
- Kang, H.-Y., Pearn, W., Chung, I.-P. ve Lee, A. H., 2016, An enhanced model for the integrated production and transportation problem in a multiple vehicles environment, *Soft Computing*, 20 (4), 1415-1435.
- Karaođlan, İ., 2009, Dağıtım Ağları Tasarımında Yer Seçimi ve Eşzamanlı Topla-Dağıt Araç Rotalama Problemi, *Doktora Tezi, Gazi Üniversitesi, Fen Bilimleri Enstitüsü.*

- Karaođlan, İ. ve Kesen, S. E., 2017, The coordinated production and transportation scheduling problem with a time-sensitive product: a branch-and-cut algorithm, *International Journal of Production Research*, 55 (2), 536-557.
- Kergosien, Y., Gendreau, M. ve Billaut, J. C., 2017, A Benders decomposition-based heuristic for a production and outbound distribution scheduling problem with strict delivery constraints, *European journal of operational research*, 262 (1), 287-298.
- Kesen, S. E. ve Bektaş, T., 2019, Integrated production scheduling and distribution planning with time windows, In: *Lean and Green Supply Chain Management*, Eds: Springer, p. 231-252.
- Lacomme, P., Moukrim, A., Quilliot, A. ve Vinot, M., 2018, Supply chain optimisation with both production and transportation integration: multiple vehicles for a single perishable product, *International Journal of Production Research*, 1-24.
- Lee, J., Kim, B.-I., Johnson, A. L. ve Lee, K., 2014, The nuclear medicine production and delivery problem, *European Journal of Operational Research*, 236 (2), 461-472.
- Li, C.-L., Vairaktarakis, G. ve Lee, C.-Y., 2005, Machine scheduling with deliveries to multiple customer locations, *European Journal of Operational Research*, 164 (1), 39-51.
- Li, C.-L. ve Vairaktarakis, G., 2007, Coordinating production and distribution of jobs with bundling operations, *IIE transactions*, 39 (2), 203-215.
- Li, K., Zhou, C., Leung, J. Y. ve Ma, Y., 2016, Integrated production and delivery with single machine and multiple vehicles, *Expert Systems with Applications*, 57, 12-20.
- Li, W. ve Ferrell, W. G., 2011, Integrated production and distribution schedule problem with a perishable product, *IIE Annual Conference. Proceedings*, 1.
- Li, W. ve Ling, Z., 2015, A Heuristic method for the integrated production distribution problem, *IIE Annual Conference. Proceedings*, 3027.
- Liu, L., Li, W., Li, K. ve Zou, X., 2020, A coordinated production and transportation scheduling problem with minimum sum of order delivery times, *Journal of Heuristics*, 26 (1), 33-58.
- Low, C., Li, R.-K. ve Chang, C.-M., 2013, Integrated scheduling of production and delivery with time windows, *International Journal of Production Research*, 51 (3), 897-909.
- Low, C., Chang, C.-M., Li, R.-K. ve Huang, C.-L., 2014, Coordination of production scheduling and delivery problems with heterogeneous fleet, *International Journal of Production Economics*, 153, 139-148.
- Low, C., Chang, C.-M. ve Gao, B.-Y., 2017, Integration of production scheduling and delivery in two echelon supply chain, *International Journal of Systems Science: Operations & Logistics*, 4 (2), 122-134.
- Marandi, F. ve Zegordi, S., 2017, Integrated production and distribution scheduling for perishable products, *Scientia Iranica*, 24 (4), 2105-2118.
- Meinecke, C. ve Scholz-Reiter, B., 2014, A heuristic for the integrated production and distribution scheduling problem, *International Science Index*, 8 (2), 290-297.
- Mladenović, N., Hansen, P. J. C. ve research, o., 1997, Variable neighborhood search, 24 (11), 1097-1100.
- Mohammadi, S., Al-e-Hashem, S. M. ve Rezik, Y., 2020, An integrated production scheduling and delivery route planning with multi-purpose machines: A case study from a furniture manufacturing company, *International Journal of Production Economics*, 219, 347-359.

- Moons, S., Ramaekers, K., Caris, A. ve Arda, Y., 2017, Integrating production scheduling and vehicle routing decisions at the operational decision level: a review and discussion, *Computers & Industrial Engineering*, 104, 224-245.
- Moscato, P., 1989, On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms, *Caltech concurrent computation program, C3P Report*, 826, 1989.
- Moscato, P., Cotta, C. ve Mendes, A., 2004, Memetic algorithms, In: *New optimization techniques in engineering*, Eds: Springer, p. 53-85.
- Naso, D., Surico, M., Turchiano, B. ve Kaymak, U., 2007, Genetic algorithms for supply-chain scheduling: A case study in the distribution of ready-mixed concrete, *European Journal of Operational Research*, 177 (3), 2069-2099.
- Park, Y.-B. ve Hong, S.-C., 2009, Integrated production and distribution planning for single-period inventory products, *International Journal of Computer Integrated Manufacturing*, 22 (5), 443-457.
- Potts, C. N., 1980, Analysis of a heuristic for one machine sequencing with release dates and delivery times, *Operations Research*, 28 (6), 1436-1441.
- Prins, C., 2004, A simple and effective evolutionary algorithm for the vehicle routing problem, *Computers & operations research*, 31 (12), 1985-2002.
- Pundoor, G. ve Chen, Z. L., 2005, Scheduling a production–distribution system to optimize the tradeoff between delivery tardiness and distribution cost, *Naval Research Logistics (NRL)*, 52 (6), 571-589.
- Radcliffe, N. J. ve Surry, P. D., 1994, Formal memetic algorithms, *AISB workshop on evolutionary computing*, 1-16.
- Ramezani, R., Mohammadi, S. ve Cheraghali, A., 2017, Toward an integrated modeling approach for production and delivery operations in flow shop system: trade-off between direct and routing delivery methods, *Journal of Manufacturing Systems*, 44, 79-92.
- Rohmer, S. ve Billaut, J.-C., 2015, Production and outbound distribution scheduling: a two-agent approach, *Industrial Engineering and Systems Management (IESM), 2015 International Conference on*, 135-144.
- Russell, R., Chiang, W.-C. ve Zepeda, D., 2008, Integrating multi-product production and distribution in newspaper logistics, *Computers & Operations Research*, 35 (5), 1576-1588.
- Sarmiento, A. M. ve Nagi, R., 1999, A review of integrated analysis of production-distribution systems, *IIE transactions*, 31 (11), 1061-1074.
- Scholz-Reiter, B., Makuschewitz, T., Novaes, A. G., Frazzon, E. M. ve Lima Jr, O. F., 2011, An approach for the sustainable integration of production and transportation scheduling, *International Journal of Logistics Systems and Management*, 10 (2), 158-179.
- Talbi, E.-G., 2009, *Metaheuristics: from design to implementation*, John Wiley & Sons, p.
- Tavares-Neto, R. F. ve Nagano, M. S., 2018, An Iterated Greedy approach to integrate production by multiple parallel machines and distribution by a single capacitated vehicle, *Swarm and Evolutionary Computation*.
- Ullrich, C. A., 2013, Integrated machine scheduling and vehicle routing with time windows, *European Journal of Operational Research*, 227 (1), 152-165.
- Van Buer, M. G., Woodruff, D. L. ve Olson, R. T., 1999, Solving the medium newspaper production/distribution problem, *European Journal of Operational Research*, 115 (2), 237-253.

- Viergutz, C. ve Knust, S., 2014, Integrated production and distribution scheduling with lifespan constraints, *Annals of Operations Research*, 213 (1), 293-318.
- Wang, J., Yao, S., Sheng, J. ve Yang, H., 2019a, Minimizing total carbon emissions in an integrated machine scheduling and vehicle routing problem, *Journal of Cleaner Production*, 229, 1004-1017.
- Wang, S., Wu, R., Chu, F. ve Yu, J., 2019b, Variable neighborhood search-based methods for integrated hybrid flow shop scheduling with distribution, *Soft Computing*, 1-20.
- Yağmur, E. ve Kesen, S. E., 2020, A memetic algorithm for joint production and distribution scheduling with due dates, *Computers & Industrial Engineering*, 142, 106342.
- Yağmur, E. ve Kesen, S. E., 2021, Multi-trip heterogeneous vehicle routing problem coordinated with production scheduling: Memetic algorithm and simulated annealing approaches, *Computers & Industrial Engineering*, 107649.
- Yağmur, E. ve Kesen, S. E., 2022, Bi-objective coordinated production and transportation scheduling problem with sustainability: formulation and solution approaches, *International Journal of Production Research*, 1-22.
- Yağmur, E. Ç., 2021, Bütünleşik üretim ve dağıtım çizelgeleme problemleri için çözüm yaklaşımları, *Doktora Tezi, Konya Teknik Üniversitesi, Lisansüstü Eğitim Enstitüsü*.
- Yan, R., Li, M. ve Wei, W., 2018, INTEGRATED PRODUCTION SCHEDULING AND DISTRIBUTION PLANNING WITH A TWO-STAGE SEMI-CONTINUOUS FLOW SHOP ENVIRONMENT, *International Journal of Simulation Modelling (IJSIMM)*, 17 (3).
- Zhong, X. ve Jiang, D., 2015, Parallel machine scheduling with batch delivery to two customers, *Mathematical Problems in Engineering*, 2015.
- Zou, X., Liu, L., Li, K. ve Li, W., 2018, A coordinated algorithm for integrated production scheduling and vehicle routing problem, *International Journal of Production Research*, 56 (15), 5005-5024.
- Zu, L., Li, W. ve Kurz, M. E., 2014, Integrated Production and distribution problem with pickup and delivery and multiple trips, *IIE Annual Conference. Proceedings*, 1587.