

**KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

**ÇOK AMAÇLI DEMONTAJ HATTI Dengeleme Problemleri ve Çözüm
Yaklaşımları: Çok Çalışanlı İstasyonların Heterojen Çalışanlarla
Değerlendirilmesi**

YÜKSEK LİSANS TEZİ

Büşra YAZICI

**AĞUSTOS 2020
TRABZON**



KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

**ÇOK AMAÇLI DEMONTAJ HATTI Dengeleme Problemleri ve Çözüm
Yaklaşımları: Çok Çalışanlı İstasyonların Heterojen Çalışanlarla
Değerlendirilmesi**

Büşra YAZICI

ORCID : 0000 -0002 - 7814 - 0439

**Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsünde
"ENDÜSTRİ YÜKSEK MÜHENDİSİ"
Unvanı Verilmesi İçin Kabul Edilen Tezdir.**

Tezin Enstitüye Verildiği Tarih : 30 / 07 / 2020

Tezin Savunma Tarihi : 31 / 08 / 2020

Tez Danışmanı : Dr. Öğr. Üyesi Ömer Faruk YILMAZ
ORCID : 0000 -0002 - 6202 - 4011

Trabzon 2020

ÖNSÖZ

Gelişen teknoloji ile üretim hızı artmakta ve üretilen birçok ürün kullanım ömrünün sonuna gelmeden atıl hale gelmektedir. Kullanım ömrü biten ve atık adı verilen bu ürünler çevre, toplum ve üreticiler için olumsuz etkilere neden olmaktadır. Bu olumsuz etkilerin giderilmesinde atık yönetiminin önemi büyüktür ve atık yönetiminde en önemli rolü geri dönüşüm üstlenmektedir.

Ucuz, yenilenebilir ve çevreye duyarlı bir kaynak kapısı olarak geri dönüşüm, üreticiler için bir fırsat olarak değerlendirilebilmektedir. Özellikle elektronik atık ürünlerde geri dönüşümün bir fırsata dönüştürülebilmesi için etkili bir geri dönüşüm yöntemi olan demontaj faaliyetlerinin iyi yönetilmesi gerekmektedir. Bu çalışma kapsamında demontaj faaliyetlerinin en iyi şekilde yönetimi için çok amaçlı demontaj hattı dengeleme problemleri ve çözüm yaklaşımları ele alınmaktadır.

Yapılan tez çalışmasının hazırlanması sürecinde yardım ve desteğini yedi gün 24 saat hiç esirgemeyen, her daim sabırla yol gösteren danışman hocam Dr. Öğr. Üyesi Ömer Faruk YILMAZ 'a teşekkürlerimi sunarım. Ayrıca yüksek lisans eğitimi boyunca desteklerini hiç esirgemeyen tüm hocalarıma teşekkürü bir borç bilirim.

Tez çalışmasında yardımlarından ve manevi desteklerinden dolayı başta Tuğçe YAVUZ ve ablam Kübra YAZICI olmak üzere tüm arkadaşlarıma ve aileme teşekkür ederim.

Hayatıma ve karakterime yön veren anneannem Bahriye (Yeter) ÖZTAŞ anısına...

Büşra YAZICI

Trabzon 2020

TEZ ETİK BEYANNAMESİ

Yüksek Lisans Tezi olarak sunduğum “Çok Amaçlı Demontaj Hattı Dengeleme Problemleri ve Çözüm Yaklaşımları: Çok Çalışanlı İstasyonların Heterojen Çalışanlarla Değerlendirilmesi” başlıklı bu çalışmayı baştan sona kadar danışmanım Dr. Öğr. Üyesi Ömer Faruk Yılmaz'ın sorumluluğunda tamamladığımı, verileri/örnekleri kendim topladığımı, deneyleri/analizleri ilgili laboratuvarlarda yaptığımı/yaptırdığımı, başka kaynaklardan aldığım bilgileri metinde ve kaynakçada eksiksiz olarak gösterdiğimi, çalışma sürecinde bilimsel araştırma ve etik kurallara uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim. 31/08/2020

Büşra YAZICI

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ.....	III
TEZ ETİK BEYANNAMESİ.....	IV
İÇİNDEKİLER.....	V
ÖZET	VI
SUMMARY	VII
ŞEKİLLER DİZİNİ	VIII
TABLolar (ÇİZELGELER) DİZİNİ.....	IX
SEMBOLLER DİZİNİ	X
1. GİRİŞ.....	1
2. YAYIN TARAMASI	4
3. PROBLEMİN TANIMI VE OPTİMİZASYON MODELİ.....	10
3.1. Problemin Tanımı	10
3.2. Optimizasyon Modeli	12
4. UYGULAMA.....	19
4.1. Deneysel Veri	19
4.2. Değerlendirme Metrikleri	23
4.3. Artırılmış Epsilon Kısıt Yöntemi 2	25
4.4. Baskın Olmayan Sıralamalı Genetik Algoritma-II.....	29
4.4.1. Kodlama ve Çözümleme Prosedürleri.....	30
4.4.2. İlk Popülasyonun Üretimi.....	37
4.4.3. Operatörler.....	41
4.5. Parametre Değerlerinin Hesaplanması	45
4.6. Karşılaştırma Analizi.....	47
5. SONUÇLAR VE ÖNERİLER.....	59
6. KAYNAKLAR.....	61
7. EKLER	67
ÖZGEÇMİŞ	

Yüksek Lisans Tezi

ÖZET

ÇOK AMAÇLI DEMONTAJ HATTI Dengeleme Problemleri ve Çözüm
Yaklaşımları: Çok Çalışanlı İstasyonların Heterojen
Çalışanlarla Değerlendirilmesi

Büşra YAZICI

Karadeniz Teknik Üniversitesi
Fen Bilimleri Enstitüsü
Endüstri Mühendisliği Anabilim Dalı
Danışman: Dr. Öğr. Üyesi Ömer Faruk YILMAZ
2020, 66 Sayfa, 43 Sayfa Ek

Geri dönüşüm, çevresel sürdürülebilirlik açısından kritik bir rol oynar ve ekonomik faydalar sağlar. Geri dönüşüm yöntemlerinden biri olan demontaj, bir ürünü kendisini oluşturan parçalarına, alt gruplarına veya elde edilmesi istenen parçalara ayırmak için sistematik bir yöntem olarak tanımlanabilir. Bu çalışmada çok amaçlı demontaj hattı dengeleme problemlerinde, zaman ve kaynak kullanımında etkili bir yöntem olan takım çalışması ve çalışanların heterojenliği dikkate alınmaktadır. Odaklanılan problemler için maliyet, çevrim süresi ve iş gücü dengeleme amaçları gözetilerek bir optimizasyon modeli geliştirilmiştir. Bu kapsamda üretilen küçük boyutlu problemlerin çözümü için artırılmış epsilon kısıt yöntemi 2 kullanılırken, büyük boyutlu problemler için baskın olmayan sıralamalı genetik algoritma-II (NSGA-II) kullanılmıştır. Elde edilen karşılaştırmalı sonuçlara göre takım çalışması esaslı istasyon stratejisi her zaman daha iyi sonuçlara ulaşmayı sağlamaktadır. Çalışanların beceri seviyelerinin dikkate alınması ulaşılan yargılarda farklılığa neden olmamıştır. Çalışan yeterlilik seviyelerinin daha geniş aralıkta dağılması ile istatistiksel olarak anlamlı olmasa da daha iyi sonuçlar saptanmıştır.

Anahtar Kelimeler: Çok Amaçlı Demontaj Hattı Dengeleme, Takım Çalışması Esaslı İstasyonlar, AUGMECON2, NSGA-II

Master Thesis

SUMMARY

MULTI-OBJECTIVE LINE BALANCING PROBLEMS AND SOLUTION
APPROACHES: CONSIDERING MULTI-MANNED STATIONS WITH
HETEROGENOUS WORKERS

Büşra YAZICI

Karadeniz Technical University
The Graduate School of Natural and Applied Sciences
Industrial Engineering Graduate Program
Supervisor: Asst. Prof. Ömer Faruk YILMAZ
2020, 66 Pages, 43 Pages Appendix

Recycling plays a critical role in environmental sustainability and provides economic benefits. Disassembly, which is one of the recycling methods, can be defined as a systematic method to separate a product into its constituent parts, subgroups, or desired parts. In this study, the multi-objective disassembly line balancing problems are addressed by considering the workers' heterogeneity and group-based worker assignment strategy, which is an effective method in using time and resources. A novel optimization model is proposed to formulate the problems with the overall cost, cycle time, and workload imbalance objectives. In this context, while the improved version of the augmented ϵ -constrained (AUGMECON2) method is employed to solve small-sized problems, the non-dominated sorting genetic algorithm-II (NSGA-II) is adapted for large-sized problems. According to the computational results, high-quality solutions are achieved when the group-based assignment strategy is realized. From the results, it is observed that the skill levels of the workers do not cause any difference in the solutions. Besides, although it is not found as the statistically significant, the heterogeneous workers provide slightly better solutions compared to homogeneous workers.

Key Words: Multi-objective disassembly line balancing; Multi-manned stations; AUGMECON2; NSGA-II

ŞEKİLLER DİZİNİ

Sayfa No

Şekil 1. Öncelik ilişki türleri: (a) VE öncelik ilişkisi, (b) VEYA öncelik ilişkisi, (c) VEYA ardıl ilişkisi	11
Şekil 2. Problem özellikleri ve çözüm yaklaşımları	20
Şekil 3. Küçük boyutlu problemler için elde edilen Pareto optimal sonuçlar	27
Şekil 4. Kodlama ve çözümlenme prosedürü (a) klasik istasyon stratejisi için, (b) takım çalışması esaslı istasyon stratejisi için.....	32
Şekil 5. (a) Kromozom yapısı, (b) VEYA öncelik ilişkisine örnek, (c) tek noktalı çaprazlama prosedürü, (d) iki noktalı çaprazlama prosedürü.....	40
Şekil 6. Mutasyon prosedürü	44
Şekil 7. İki farklı çaprazlama operatörü ile elde edilen Pareto optimal sonuçların C metriğine göre değerleri	48
Şekil 8. İki farklı çaprazlama operatörü ile elde edilen Pareto optimal sonuçların DI_R metriğine göre değerleri	48
Şekil 9. İki farklı çaprazlama operatörü ile elde edilen Pareto optimal sonuçların OS metriğine göre değerleri	49
Şekil 10. İki farklı çaprazlama operatörü ile elde edilen Pareto optimal sonuçların baskılanamayan çözüm sayıları.....	50
Şekil 11. Teorik senaryolar için elde edilen Pareto optimal sonuçların C metriğine göre karşılaştırmalı sonuçları	51
Şekil 12. Pratik senaryolar için elde edilen Pareto optimal sonuçların C metriğine göre karşılaştırmalı sonuçları	51
Şekil 13. Teorik senaryolar için elde edilen Pareto optimal sonuçların DI_R metriğine göre karşılaştırmalı sonuçları	53
Şekil 14. Pratik senaryolar için elde edilen Pareto optimal sonuçların DI_R metriğine göre karşılaştırmalı sonuçları	53
Şekil 15. Teorik senaryolar için elde edilen Pareto optimal sonuçların OS metriğine göre karşılaştırmalı sonuçları	54
Şekil 16. Pratik senaryolar için elde edilen Pareto optimal sonuçların OS metriğine göre karşılaştırmalı sonuçları	54
Şekil 17. Teorik senaryolar için elde edilen Pareto optimal sonuçların Nnd metriğine göre karşılaştırmalı sonuçları	55
Şekil 18. Teorik senaryolar için elde edilen Pareto optimal sonuçların Nnd metriğine göre karşılaştırmalı sonuçları	55
Şekil 19. Çalışan beceri ve yeterlilik seviyeleri için karşılaştırmalı sonuçlar	57

TABLolar (ÇİZELGELER) DİZİNİ

Sayfa No

Tablo 1. Literatür Matrisi	8
Tablo 2. Optimizasyon modelinde mevcut olan kısıt sayısı	18
Tablo 3. Optimizasyon modelinde mevcut olan değişken sayısı.....	18
Tablo 4. Küçük boyutlu problemlere ait veriler	21
Tablo 5. Büyük boyutlu problemlere ait veriler	22
Tablo 6. Küçük boyutlu problemlere ait sınır değerleri tablosu.....	25
Tablo 7. Küçük boyutlu problemlerin karşılaştırmalı sonuçları.....	28
Tablo 8. Küçük boyutlu problemlerden problem 5 için Pareto optimal sonuçlar	45
Tablo 9. Parametre optimizasyonu	46
Tablo 10. Çalışan yeterlilik seviyelerinin karşılaştırılması için varyans analizi tablosu.....	58
Ek Tablo 1. Küçük boyutlu problemler için AUGMECON2 yönteminin sonuçları	81
Ek Tablo 3. Tüm tekrarlar için parametre optimizasyonu değerleri.....	103
Ek Tablo 4. Senaryo 1 için çaprazlama operatörlerinin karşılaştırmalı sonuçları.....	104
Ek Tablo 5. Senaryo 2 için çaprazlama operatörlerinin karşılaştırmalı sonuçları.....	105
Ek Tablo 6. Senaryo 3 için çaprazlama operatörlerinin karşılaştırmalı sonuçları.....	106
Ek Tablo 7. Senaryo 4 için çaprazlama operatörlerinin karşılaştırmalı sonuçları.....	107
Ek Tablo 8. Tek noktalı çaprazlama operatörüne göre teorik stratejilerin karşılaştırmalı sonuçları.....	108
Ek Tablo 9. Tek noktalı çaprazlama operatörüne göre pratik stratejilerin karşılaştırmalı sonuçları.....	109

SEMBOLLER DİZİNİ

AUGMECON	Artırılmış Epsilon Kısıt Yöntemi
AUGMECON2	Artırılmış Epsilon Kısıt Yöntemi 2
CR1	Tek Noktalı Çaprazlama Operatörü
CR2	İki Noktalı Çaprazlama Operatörü
ÇADHDP	Çok Amaçlı Demontaj Hattı Dengeleme Problemi
DHDP	Demontaj Hattı Dengeleme Problemi
MHDP	Montaj Hattı Dengeleme Problemi
NCR1SC1	Senaryo 1 için Tek Noktalı Çaprazlama ile NSGA-II
NCR2SC1	Senaryo 1 için İki Noktalı Çaprazlama ile NSGA-II
NCR1SC2	Senaryo 2 için Tek Noktalı Çaprazlama ile NSGA-II
NCR2SC2	Senaryo 2 için İki Noktalı Çaprazlama ile NSGA-II
NCR1SC3	Senaryo 3 için Tek Noktalı Çaprazlama ile NSGA-II
NCR2SC3	Senaryo 3 için İki Noktalı Çaprazlama ile NSGA-II
NCR1SC4	Senaryo 4 için Tek Noktalı Çaprazlama ile NSGA-II
NCR2SC4	Senaryo 4 için İki Noktalı Çaprazlama ile NSGA-II
NSGA-II	Baskın Olmayan Genetik Algoritma-II
SC1	Klasik İstasyon Stratejisi İçin Çalışan Yeterlilik Seviyelerini Ele Alan Senaryo
SC2	Klasik İstasyon Stratejisi İçin Çalışan Yeterlilik Ve Beceri Seviyelerini Ele Alan Senaryo
SC3	Takım Çalışması Esaslı İstasyon Stratejisi İçin Çalışan Yeterlilik Seviyelerini Ele Alan Senaryo
SC4	Takım Çalışması Esaslı İstasyon Stratejisi İçin Çalışan Yeterlilik Ve Beceri Seviyelerini Ele Alan Senaryo
ST1	Klasik İstasyon Stratejisi
ST2	Takım Çalışması Esaslı İstasyon Stratejisi

1. GİRİŞ

Üretilen birçok ürünün kullanım ömrünün sonuna gelmeden atıl hale gelmesi ve mevzuatta yayınlanan birçok yönetmelik ile geri dönüşümün önemi oldukça artmıştır. Geri dönüşüm sürecinde, ürünün kendisini oluşturan alt parçalarının en iyi şekilde analiz edilebilmesini sağlayan ve geride en az miktarda parça kalmasının yolunu açan en etkili yöntemlerden birisi demontajdır (McGovern ve Gupta, 2007a). Demontaj, atık adı verilen geri dönüştürülebilir parçalardan oluşan ürünlerin belirli işlemler ile elde edilmek istenen alt parçalarına ayrılması yöntemidir (Cevikcan vd., 2019). Demontaj yöntemi ile atıklar kendisini oluşturan parçalara ya da yapılara ayrılarak basit onarım ya da yenilemeler sonucu üretime geri kazandırılmaktadır. Demontaj sonucu üretime geri kazandırılması mümkün olmayan ya da tehlike oluşturabilecek parçalar ise imha edilmektedir (Gungor ve Gupta, 2001).

Demontaj yönteminin akışının gerçekleşmesi demontaj hatları ile sağlanmaktadır. Demontaj yönteminde istenen parçaları elde etmek için yapılması gereken sökme görevleri demontaj hatlarında sıralanır. Bu sıralamada görevler arasındaki öncelik ilişkileri dikkate alınmalıdır. Demontaj hattında hangi görevlerin, hangi sıraya göre, hangi çalışan tarafından, hangi istasyonda, ne zaman yapılması gerektiği gibi sorular demontaj hattı dengeleme problemlerini (DHDP) oluşturmaktadır.

İşletmeler tarafından demontaj yönteminin uygulanması için maliyetinin düşük, elde edilen parçalara talebin yüksek olması beklenir (Avikal ve Mishra, 2012). Bu durumda demontaj yönteminin uygulanabilmesi için talebi olan parçaların en düşük maliyetle, en az kaynakla ve en kısa zamanda sökülmesi beklenir. Bu nedenle demontaj yönteminde tüm görevlerin yerine getirilmesi yerine sadece talebi olan parçaların elde edilebileceği görevlerin yapılması tercih edilmektedir (Bentaha vd., 2014). Diğer yandan demontaj yönteminin uygulanabilmesi, bu metod için kullanılan kaynaklara da bağlıdır. Bu nedenle açılan istasyon sayısı, kullanılan ürün miktarı, bu işlem için atanan çalışan sayısı gibi maliyet oluşturacak kalemlerin de minimize edilmesi gerekir. Aynı zamanda demontaj işletme açısından ek zaman ayrılması gereken bir yöntem olduğu için işlemler en kısa zamanda yapılmalıdır. Ayrıca işletmenin sosyal etkinliklerini iyileştirmede ve sürdürülebilirliği sağlamada çalışanlar arasında adalet ve hakkaniyetin sağlanması da önem arz etmektedir (Lian vd., 2018).

Tüm bu durumlar demontaj hatları dengelenirken birden fazla amacın aynı anda göz önüne alınmasının önemini ön plana çıkarmaktadır. Birbiri cinsinden ifade edilemeyen amaçlar, çok amaçlı problemleri oluşturmaktadır. Çok amaçlı problemlerde en iyi çözüm tek bir noktada toplanmak yerine, Pareto optimal sonuçlar olarak adlandırılan birden fazla en iyi noktada toplanır (Deb vd., 2002).

Montaj ve demontaj hatlarında, zamanın ve alan kapasitesinin etkili kullanılmasını sağlayan yöntemlerden bir tanesi takım çalışmasıdır. Takım çalışmasında çalışanlar, öncelik ilişkilerini göz önüne almak koşulu ile farklı görevleri aynı anda yapabilmektedir. Takım çalışması; teslim süresi ve yarı ürün stoğu, kalite, hat uzunluğu ve taşıma, esneklik ve iş memnuniyeti gibi konularda avantaj sağlamaktadır (Cevikcan vd., 2019).

İşletmelerin sahip olduğu çalışan havuzu ele alındığında her bir çalışanın farklı görevleri yerine getirme ya da getirebilme düzeyleri heterojenlik göstermektedir. Bu durum işlem zamanını, çıktı oranını, kaliteyi etkileyebilmektedir (Lian vd., 2018). Bu nedenle işletmelerin farklı yeterlilik/ beceri seviyelerine sahip çalışanları ilgili görevlere etkili bir şekilde ataması ile elde edebileceği faydalar demontaj hatları için de geçerlidir.

Bu çalışmada çalışanların heterojenlikleri göz önüne alınarak çok amaçlı demontaj hattı dengeleme problemi (ÇADHDP), klasik istasyon ve takım çalışması esaslı istasyon olmak üzere iki strateji altında analiz edilmiştir. Problem için geliştirilen optimizasyon modelinde toplam maliyetin, çevrim süresinin ve çalışanlar arasındaki dengesizliğin en aza indirgenmesi amaçlanmıştır. Geliştirilen modelde amaçların en iyi değerlerinin bulunması ile birlikte hangi görevlerin yapılacağı, çevrim süresinin ne olacağı, talebin karşılanması için kaç adet ürünün demonte edilmesi gerektiği, hangi görevin hangi çalışan tarafından hangi istasyonda ne zaman başlayacağı gibi kararlar da belirlenmektedir.

Çalışmanın özgün yönlerine aşağıda yer verilmektedir.

- Takım çalışması esaslı istasyonları içeren ÇADHDP'ni toplam maliyet, çevrim süresi ve iş gücü dengeleme amaçları ile birlikte ele almaktadır.
- Demontaj ilişki türlerinden VEYA ardıl ilişki türünü dikkate alan ve görevler arasındaki ilişkileri gösteren teknolojik öncelik diyagramlarını kullanan bir optimizasyon modeli geliştirilmiştir.
- Klasik istasyon ile takım çalışması esaslı istasyon içeren ÇADHDP stratejilerini taktiksel seviyede karşılaştırmaktadır.
- Pratikte var olup, teoride çok fazla kullanılmayan çalışan beceri seviyelerini operasyonel seviyede senaryo üreterek incelemektedir.

- Çalışanların yeterlilik seviyelerindeki heterojenlik miktarlarını analiz etmektedir.
- Küçük boyutlu ÇADHDP için artırılmış epsilon kısıt yöntemi 2 (AUGMECON2) yöntemini uygularken, büyük boyutlu problemler için iki farklı çaprazlama operatörü ile baskın olmayan sıralamalı genetik algoritma-II (NSGA-II) tabanlı algoritma geliştirmektedir.
- Elde edilen karşılaştırmalı sonuçlar ile taktiksel seviye stratejileri ve operasyonel seviye senaryoları arasında çeşitli yönetimsel görüşler ve çıkarımlar sunmaktadır.

Çalışma kapsamında kullanılan veriler, DHDP ile ilgili çalışmalarda bulunan ve arasında gerçek üretim işletmelerinden alınan verilerin de bulunduğu problem verilerine dayanılarak üretilmiştir. Problem verilerine Yılmaz (2020)'de yer verilmektedir. Bu kapsamda bu çalışma geri dönüşüm ile ilgilenen tesislerin yönetimi ile ilgilidir. Çalışmada ele alınan noktalarla birlikte gerçekleştirilecek çalışmalara ışık tutulması ve ilgili işletmelere yol haritası sunulması amaçlanmaktadır.

Bu tez çalışmasının diğer kısımları şu şekilde tasarlanmıştır: İkinci başlık altında kapsamlı bir yayın taramasına yer verilmiştir. Üçüncü başlıkta ele alınan problem detaylıca açıklanarak problem için bir optimizasyon modeli önerisi getirilmiştir. Dördüncü başlıkta ÇADHDP için kullanılan yöntem ve materyaller anlatılmış, deneysel hesaplamalara yer verilmiştir. Son olarak beşinci başlıkta çalışmadan çıkarılan yargılara yer verilmiş ve öneriler getirilmiştir.

2. YAYIN TARAMASI

Bu bölümde demontaj hattı dengeleme problemi (DHDP) ve takım çalışması ile ilgili akademik literatürde yer alan çalışmalara yer verilmiştir.

DHDP için yapılan ilk çalışmalar incelendiğinde Gupta ve Taleb (1994)'in, demontaj hatlarında çizelgelemeyi ele alarak, demontaj hattının montaj hattının tersi gibi görüldüğü halde daha karmaşık yapıya sahip olduğu tanısını ortaya koyarak literatüre yeni bir araştırma alanı kazandırdıkları görülmektedir. Gungor ve Gupta (2001), demontaj hattında görev hatalarının olması durumunu ele almış ve *VEYA* ardılının önemini vurgulamıştır. Çalışmada bir veya birden fazla görevin yerine getirilemediği durumda öncelik ilişkileri nedeniyle devamında gelen görevlerin yapılamayacağına vurgu yapmış ve bu durumda oluşabilecek komplikasyonlar tartışılmıştır.

Devam eden süreç içerisinde McGovern ve Gupta (2007a; 2007b), hesaplama açısından çok zor olduğunu ileri sürdükleri çok amaçlı DHDP için genetik algoritma, karınca koloni algoritması, açgözlü arama algoritması, tepe tırmanma algoritması metotlarını içeren meta-sezgisel, sezgisel ve hibrit yaklaşımlar önermiştir. Agrawal ve Tiwari (2008), DHDP'yi karma model ve görev sürelerinin belirsizliği altında inceleyerek U-tipi demontaj hattı önermiştir. Problemin çözümü için işbirlikçi karınca kolonisi optimizasyonunu kullanmıştır. Lambert (2007), temel DHDP'ni ele alarak iş istasyonu sayısını minimize etmek için sezgisel üretmiştir. Altekin vd. (2008), bütün görevlerin yapılma zorunluluğunun olmadığı varsayımı altında görev-parça ilişkilerini inceleyerek kar maksimizasyonu için optimizasyon modeli önermiştir. Koc vd. (2009), *VE/VEYA* grafikleri ile DHDP için iki kesin formülasyon geliştirerek yöntemlerinin etkinliklerini kanıtlamıştır.

DHDP için yapılan çalışmaların hız kazandığı süre içerisinde Ding vd. (2010), çok amaçlı DHDP'ye yoğunlaşarak meta-sezgisel karınca kolonisi algoritmasını kullanarak Pareto optimal çözümler ile ilgilenmiştir. Avikal ve Mishra (2012), önemine göre sıraladığı amaçlar doğrultusunda problem için U-tipi hat önererek çözüm için sezgisel yöntem kullanmıştır. Avikal vd. (2013), *VE/VEYA* öncelik kısıtlamalarını dikkate alarak U-tipi hat ile DHDP için bir sezgisel algoritma geliştirmiştir.

Aydemir-Karadag ve Turkbey (2013), görev sürelerinin belirsizliği altında çok amaçlı DHDP için paralel istasyon yapısını önermiştir. Ele aldığı iki farklı amacı Pareto optimal sonuçlar ile değerlendirerek çözüm için genetik algoritmayı kullanmıştır. Algoritmada iki farklı kondisyon değerlendirme yaklaşımı, onarım algoritması kullanılmış ve çeşitlendirme stratejisine başvurmuştur. Öncelik ilişkilerini ifade etmek için *VE/ VEYA* grafiklerini ele almıştır. Paksoy vd. (2013), çok amaçlı karma modellenli DHDP için kesin sonuç veren optimizasyon modeli sunmuştur. Kalayci ve Gupta (2013), görevlerin yapılma sıralamalarına göre görev sürelerinde değişim olabileceğini ileri sürerek çözüm için karınca kolonisi optimizasyonunu kullanmıştır. Kalayci vd. (2014a) belirsiz görev sürelerini bulanık sayılar ile birlikte ele alarak çözüm için ayrık yapay arı kolonisi algoritmasını kullanmıştır. Ele aldığı çok amaçlı problemi Pareto optimal sonuçlar ile birlikte analiz etmiştir. Kalayci vd. (2014b), yine görev sıralamasına bağlı olarak değişen görev sürelerini ele alan çok amaçlı DHDP'yi farklı bir komşu arama algoritması ile hibrit ettiği genetik algoritma yöntemi ile çözmüştür. Bentaha vd. (2014), tehlikeli parçaların ve belirsiz görev sürelerinin varlığında demontaj ürünlerinin kısmi sökme yaparak dengelenmesini ele almış, maksimum kar amacı ile Monte Carlo simülasyonuna dayanan kesin sonuç veren bir çözüm önerisi getirmiştir. Hezer ve Kara (2014), çok modellenli DHDP için paralel hatta en kısa rotayı bulmayı hedeflemiş ve bunun için de teknolojik öncelik diyagramlarını detaylıca ele almıştır.

DHDP'yi farklı açılardan ele alan çalışmalardan Özceylan ve Paksoy (2013), problemi ters tedarik zinciri açısından inceleyerek müşteri, toplama merkezi, demontaj merkezi ve tesisler arasında dağıtımını, maliyeti, iş istasyonlarını, çevrim süresini ve görev atamalarını en iyilemiştir. Özceylan vd. (2014), nakliye, satın alma, yenileme ve iş istasyonlarının işletim maliyetlerini en aza indirmek amacıyla ileri ve geri tedarik zincirini en iyilemiştir. Kalaycılar vd. (2015), aynı tür üründen farklı parçaların ayrılması durumunu sabit istasyon sayısı ve sınırlı ürün girişi varsayımları altında maksimum gelir amacı ile incelemiştir. Riggs vd. (2015), farklı hasar görmüş aynı tür ürünler üzerinden değişen görevleri ve belirsiz görev sürelerini ortak teknolojik diyagram üzerinden değerlendirmiştir.

Yakın zamanda yapılan çalışmalar incelendiğinde Kannan vd. (2016), karma modellenli DHDP'de ters tedarik zincirini üçüncü taraf lojistik sağlayıcıları ile birlikte değerlendirerek maliyeti minimize etmiştir. Mete vd. (2016a) problemi robot, ekipman, malzeme, özel makine, kalifiye personel gibi kısıtlı kaynak varsayımı altında ele almıştır. Mete vd. (2016b), DHDP için farklı bir meta-sezgisel yöntem olan ışın arama algoritmasını önermiştir. Altekin (2017), belirsiz görev süreleri ile DHDP için daha hızlı ve kesin çözüm bulmak amacıyla

ikinci dereceden koni programlama ile parçalı karma tam sayılı doğrusal programlama modellerini ele almıştır. Zhang vd. (2017), belirsiz görev süreleri ile birlikte ele aldığı problemi çözmek için yapay balık sürüsü algoritmasını kullanmıştır. Ren vd. (2018a), sökme sırasına bağlı gecikmeleri azaltmak için paralel hat önermiştir ve problemi çözmek için genetik algoritmayı kullanmıştır. Ren vd. (2018b), bulanık çok kriterli karar verme yöntemleri ile hibrit edilen iki sezgisel kullanılarak DHDP'ye çözüm önermiştir. Li vd. (2019), dal-sınır algoritmasını kullanarak istasyon sayısının minimize edilmesi için DHDP'ye kesin çözüm getiren bir algoritma önermiştir. Edis vd. (2019), probleme 30 görev sayısına kadar kesin çözüm veren bir optimizasyon modeli önermiştir. Wang vd. (2019), çok amaçlı DHDP için paralel hat önererek çözüm için ayrık çiçek tozlaşma algoritmasını Pareto optimal sonuçlar ile birlikte kullanmıştır. Liu vd. (2019), DHDP'de belirsizlikleri gürbüz optimizasyon ile ele alarak çözüm için karma tamsayı ikinci dereceden koni programlama modeli önermiştir. Bentaha vd. (2020), parça kalitelerini ele alarak elde edilen kara göre karar mekanizması geliştirmiştir. Liu vd. (2020), görevler arasındaki geçiş sürelerinin farklı olduğu varsayımı altında robotik DHDP için ayrık arı algoritmasını kullanmıştır.

Son zamanlarda bu alanda yapılan çalışmaların artması ile araştırmacılara yol göstermek adına Deniz ve Ozcelik (2019) ile Özceylan vd. (2018) DHDP üzerine ayrıntılı yayın taraması yapmıştır.

Takım çalışmasını ele alan çalışmalardan Chen (2017), otomotiv endüstrisinde iş istasyonlarının sayısını en aza indirmek ve maliyeti minimize etmek için ele aldığı montaj hattı dengeleme problemini takım çalışması ile birlikte değerlendirmiştir. Çalışmada çevrim süresinin ve bir istasyona atanabilecek maksimum çalışan sayısının sonuçlar üzerindeki etkileri analiz edilmiştir. Chen vd. (2018), kaynak kısıtlı montaj hattı dengeleme problemini takım çalışması ile ele almıştır. Çalışmada iş istasyonu, operatör ve kaynakların sayısının minimize edilmesi amaçlanmıştır. Şahin ve Kellegöz (2019), istasyon sayısını ve kaynakları minimize ederek çok çalışanlı paralel montaj hatlarının maliyetini azaltmaya çalışmıştır. Lopes vd. (2019), çok çalışanlı montaj hatları için esnek istasyon sınırlamalarına dikkat çekmiştir. Demontaj hattında takım çalışmasını ele alan çalışmalardan Fang vd. (2018), karma modellenmiş DHDP için otomasyonu önermiş ve robotları takım çalışması ile incelemiştir. VE/VEYA grafiklerine ve paralel hat yapısına bağlı olarak çözüm için genetik algoritmayı kullanmıştır. Çevikcan vd. (2019), takım çalışmasının önemini vurgulayarak bir optimizasyon modeli önerisi getirmiş ve iki sezgisel ile birlikte çözüme gitmiştir. Takım

çalışması kapsamında yapılan tüm çalışmalarda, hattın etkinliğinin artırılmasında takım çalışmasının önemine vurgu yapılmıştır.

Yayın taramasında incelenen çalışmalar Tablo 1’de verilen literatür matrisi kapsamında özetlenmektedir. Görüldüğü üzere DHDP için takım çalışmasını ele alan çalışma sayısı oldukça sınırlı olmakla beraber bu çalışmalarda çalışanların heterojenliği ele alınmamıştır. Aynı zamanda VEYA ardıl ilişkisini ele alan çalışmalar genellikle VE/VEYA grafikleri ile birlikte kullanmakta, bu durum da benzer görevlerin tekrarlanmasından dolayı grafiklerde karmaşıklığa neden olmaktadır. VEYA ardıl ilişkisini görevler arasındaki ilişkiyi esas alan teknolojik öncelik diyagramları ile birlikte kullanan çalışma sayısı çok az sayıdadır. Çalışmalarda talep ile çevrim süresi ilişkilendirilmemekte, talebin hat dengeleme üzerindeki etkisi göz ardı edilmektedir. Bu çalışma kapsamında ele alınan amaçlar hiçbir çalışmada birlikte değerlendirilmemiş olup, çözüm yöntemi olarak artırılmış epsilon kısıt yöntemi 2 (AUGMECON2) ile baskın olmayan sıralamalı genetik algoritma-II (NSGA-II) metotları kullanılmamıştır. Literatürdeki bu eksiklikler göz önüne alındığında yapılan çalışma; incelenen problem, kullanılan yöntem, ele alınan stratejiler ve senaryolar ile birlikte değerlendirildiğinde literatüre katkı sağlayacağı düşünülmektedir.

Tablo 1. Literatür Matrisi

Çalışma	Amaç	Hat tipi	Ürün tipi	Çözüm Yöntemi	Çalışan tipi	Takım Çalışması
Güngör ve Gupta (2001)	İS	I-tipi	Tek	Sezgisel	Homojen	Yok
McGovern ve Gupta (2007a)	İS, HD, TPÖ, YTPÖ, RD	I-tipi	Tek	Meta-sezgisel (GA, KKO), Sezgisel (AG, HK), Hibrit yaklaşımlar	Homojen	Yok
McGovern ve Gupta (2007b)	İS, HD, TPÖ, YTPÖ, RD	I-tipi	Tek	Meta-sezgisel (GA)	Homojen	Yok
Lambert (2007)	İS	I-tipi	Tek	Sezgisel	Homojen	Yok
Agrawal ve Tiwari (2008)	KVK, HV	U-tipi	Karma	Meta-sezgisel (KKO)	Homojen	Yok
Altekin vd. (2008)	TK	I-tipi	Tek	Sezgisel	Homojen	Yok
Koc vd. (2009)	İS	I-tipi	Tek	Kesin çözüm	Homojen	Yok
Ding vd. (2010)	İS, HD	I-tipi	Tek	Meta-sezgisel (ÇAKKO)	Homojen	Yok
Avikal ve Mishra (2012)	İS	U-tipi	Tek	Sezgisel	Homojen	Yok
Avikal vd. (2013)	HD	U-tipi	Tek	Sezgisel	Homojen	Yok
Aydemir-Karadag ve Turkbey (2013)	HD, TM	I-tipi	Tek	Meta-sezgisel (GA)	Homojen	Yok
Kalayci ve Gupta (2013)	İS, HD, TPÖ, YTPÖ	I-tipi	Tek	Meta-sezgisel (KKO)	Homojen	Yok
Paksoy vd. (2013)	CT, İS, HD	I-tipi	Karma	BÇAP	Homojen	Yok
Özceylan ve Paksoy (2013)	TM, İS	I-tipi	Tek	Kesin çözüm	Homojen	Yok
Bentaha vd. (2014)	TK	I-tipi	Tek	Kesin çözüm	Homojen	Yok
Hezer ve Kara (2014)	EKR	Paralel	Tek	Kesin çözüm	Heterojen	Yok
Kalayci vd. (2014a)	İS, HD, TPÖ, YTPÖ	I-tipi	Tek	Meta-sezgisel (HAYAKA)	Homojen	Yok
Kalayci vd. (2014b)	İS, HD, TPÖ, YTPÖ	I-tipi	Tek	Meta-sezgisel (HGA)	Homojen	Yok
Özceylan vd. (2014)	TM	I-tipi	Tek	Kesin çözüm	Homojen	Yok
Kalaycılar vd. (2015)	TG	I-tipi	Tek	Sezgisel	Homojen	Yok

Tablo 1'in devamı

Çalışma	Amaç	Hat tipi	Ürün tipi	Çözüm Yöntemi	Çalışan tipi	Takım Çalışması
Riggs vd. (2015)	HD	I-tipi	Tek	Kesin çözüm	Homojen	Yok
Kannan vd. (2016)	TM	I-tipi	Karma	Kesin çözüm	Homojen	Yok
Mete vd. (2016a)	TKM	I-tipi	Tek	Kesin çözüm	Homojen	Yok
Mete vd. (2016b)	İS	I-tipi	Tek	Meta-sezgisel (IAA)	Homojen	Yok
Altekin vd. (2017)	İS	I-tipi	Tek	Kesin çözüm	Homojen	Yok
Zhang vd. (2017)	İS, HD, TM, RD	I-tipi	Tek	Meta-sezgisel (YBSA)	Homojen	Yok
Ren vd. (2018a)	CT, TE, TKM	Paralel	Tek	Meta-sezgisel (GA)	Homojen	Yok
Ren vd. (2018b)	İS, HD, TPÖ, YTPÖ	I-tipi	Tek	Hibrit yaklaşım	Homojen	Yok
Fang vd. (2018)	CT, TE, TKM	I-tipi	Karma	Meta-sezgisel (GA)	Homojen	Var
Cevikcan vd. (2019)	İS, ÇS	I-tipi	Tek	Sezgisel	Homojen	Var
Edis et al. (2019)	İS, HD, TPÖ, YTPÖ	I-tipi	Tek	Kesin çözüm	Homojen	Yok
Li vd. (2019)	İS	I-tipi	Tek	Kesin çözüm	Homojen	Yok
Liu vd. (2019)	TM	I-tipi	Tek	Kesin Çözüm	Homojen	Yok
Wang vd. (2019)	İS, HD, TE, TK	Paralel	Tek	Meta-sezgisel (AÇTA)	Homojen	Yok
Bentaha vd. (2020)	TK	I-tipi	Tek	Kesin çözüm	Homojen	Yok
Liu vd. (2020)	İS, HD, ÇZ, CT	I-tipi	Tek	Meta-sezgisel (AAA)	Homojen	Yok
Bu çalışma	TM, CT, ÇD	I-tipi	Tek	Meta-sezgisel (NSGA-II)	Heterojen	Var

İS: İstasyon sayısı, HD: Hat dengeleme/ istasyonların boşa kalan zamanı, TPÖ: Tehlikeli parçalara öncelik, YTPÖ: Yüksek talepli parçalara öncelik, RD: Rota değişimi, KVK: Kaynakların verimli kullanımı, HV: Hat verimliliği, TK: Toplam kar, TM: Toplam maliyet, ÇD: Çalışan dengeleme, CT: Çevrim süresi, EKR: En kısa rota, TG: Toplam gelir, TKM: Toplam kaynak miktarı, TE: Toplam enerji, ÇS: Çalışan sayısı, ÇZ: Çalışma zamanı; GA: Genetik algoritma, KKO: Karınca kolonisi optimizasyonu, AG: Açgözlü arama (Greedy) algoritması, HK: Hunter-killers sezgiseli, ÇAKKO: Çok amaçlı karınca kolonisi optimizasyonu, BÇAP: Bulanık çok amaçlı programlama, HAYAKA: Hibrit ayrık yapay arı kolonisi algoritması, HGA: Hibrit genetik algoritma, IAA: Işın arama algoritması, YBSA: Yapay balık sürüsü algoritması, AÇTA: Ayrık çiçek tozlaşma algoritması, AAA: Ayrık arı algoritması

3. PROBLEMİN TANIMI VE OPTİMİZASYON MODELİ

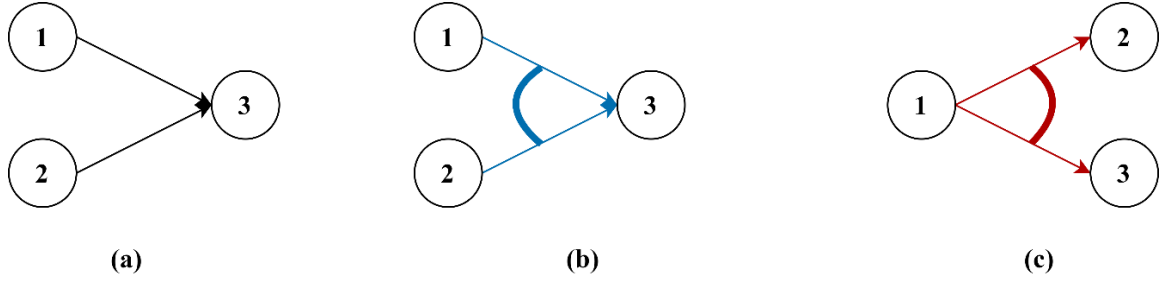
Bu bölümde problem ayrıntılı bir şekilde tanıtılmış ve problem için geliştirilen optimizasyon modeline yer verilmiştir.

3.1. Problemin Tanımı

Demontaj hattı dengeleme problemi (DHDP), geri dönüşüm sürecinde talep edilen parçaların atık adı verilen ürünlerden sökülmesi aşamasında en etkili demontaj yönteminin ne olduğu ile ilgilenir. Demontaj yöntemi montaj yönteminin tersi gibi görünse de DHDP; görevler arasındaki ilişki türleri, sökülecek ürünlerin içerdiği tehlike oluşturabilecek parçalar, tüm görevlerin yapılma zorunluluğunun olmaması gibi nedenlerden dolayı montaj hattı dengeleme problemlerine (MHDP) göre daha kapsamlı konumdadır (Gungor ve Gupta, 2001). Bu durum DHDP'ye MHDP'ye göre ek kısıtlamalar getirmektedir.

Demontaj yönteminin uygulanmasında herhangi bir görevin yapılması üründe hasar oluşumuna neden olabileceğinden bu durum diğer görevlerin yapılmasına engel teşkil edebilmektedir. Bu nedenden dolayı demontaj işlemlerinde tüm görevlerin yapılması şartı aranmamaktadır (Altekin vd., 2008). Bununla beraber bazı durumlarda bir görevin yapılabilmesi için önceliğinde gelen farklı görevlerden en az bir tanesinin yapılması yeterli olabilmektedir. Tüm bu durumları göz önüne alabilmek için demontaj hatlarında sökülecek ürüne ait görevler arasında Şekil 1'de de görüldüğü gibi üç farklı ilişki türü belirlenmiştir.

- *VE* öncelik ilişkisi: Bu ilişki türünde ilgili görevin yapılabilmesi için öncesinde gelen *VE* öncelikli görevlerin yapılması şarttır (Şekil 1 (a)).
- *VEYA* öncelik ilişkisi: Bu ilişki türünde ilgili görevin yapılabilmesi için öncesinde gelen *VEYA* öncelikli görevlerden en az bir tanesinin yapılması şarttır (Şekil 1 (b)).
- *VEYA* ardıl ilişkisi: Bu ilişki türünde ilgili görevden sonra gelen *VEYA* ardıl ilişkisine sahip görevlerden en fazla bir tanesi yapılabilir (Şekil 1 (c)).



Şekil 1. Öncelik ilişki türleri: (a) *VE* öncelik ilişkisi, (b) *VEYA* öncelik ilişkisi, (c) *VEYA* ardıl ilişkisi

Bu ilişki türleri göz önüne alınarak ürünün demontaj adımlarını ifade etmek için iki farklı teknolojik öncelik diyagramı kullanılmaktadır. Bunlardan biri olan parça tabanlı teknolojik öncelik diyagramları, üründen parçaların ayrılma sıralamasına göre oluşturulmaktadır. Bu diyagram türü, görev sürelerini analiz etmede yetersiz kaldığı için çok fazla kullanılmamaktadır. Diğer diyagram türü olan görev tabanlı teknolojik öncelik diyagramları ise sökme görevleri arasındaki ilişkilere göre oluşturulmaktadır. DHDP’de parçaların hasar görmesine neden olan görevleri daha iyi analiz etmek ve görevler arasında rota oluşturmak amacıyla bu diyagram geliştirilerek *VE/VEYA* grafikleri oluşturulmuştur (Hezer ve Kara, 2014). Ancak *VE/VEYA* grafikleri benzer görevlerin farklı rotalarda tekrar gösterilmesine yol açtığı için yapılan çalışmada karmaşıklığı azaltmak adına görev tabanlı teknolojik öncelik diyagramı ele alınmıştır. Bu çalışmada ele alınan çok amaçlı demontaj hattı dengeleme problemi (ÇADHDP) aşağıdaki gibi tanımlanmaktadır.

Tek tür ürünlerin demonte edilmesi ile K adet parçaya olan taleplerin karşılanması istenmektedir. K adet parçaya olan taleplerin karşılanması şartıyla, ürünlerin demontajı için belirlenen I adet görevin tümünün yapılması zorunlu değildir. Demontaj görevlerinin yapılması için W adet çalışan mevcuttur. Hangi görevden hangi parçanın/ parçaların kaç adet ayrıldığı, hangi görevde hangi parçanın kullanıldığı bilinmektedir. Her görevin standart işlem süreleri ve hangi görevin hangi çalışan tarafından ne kadarlık bir yeterlilik seviyesi ile yapıldığı ya da yapılmadığı öngörülmektedir. Görevler arasında *VE*, *VEYA* öncelik ile *VEYA* ardıl ilişki türleri bulunmaktadır. Bir görevin yapılabilmesi için bu ilişki türlerinin yerine getirilmiş olması şarttır. Aralarında öncelik ilişkisi bulunan iki görevden biri bitmeden diğer görev başlayamaz ve aynı zamanda bir çalışan bir görevi bitirmeden diğer göreve başlayamaz. Sisteme girecek ürünlerin demonte edilmesi için bu işlemlere ayrılabilir toplam demontaj süresi $Time$ kadardır. İstasyonların işlem süreleri, bu süre dahilinde hesaplanan takt süresini aşamazlar. Takım çalışması kapsamında bir istasyona en

fazla W_{MAX} adet çalışan atanabilmektedir (Eğer problem klasik istasyon için değerlendirilecekse $W_{MAX} = 1$ alınır). Bir çalışan birden fazla istasyona atanamaz.

Çalışma kapsamında göz önüne alınan varsayımlar aşağıdaki şekildedir.

- Problem tek tür ürünler için ele alınmaktadır.
- Ürün adet sınırı bulunmamaktadır.
- Her görev birbirinden bağımsız olarak yürütülür.
- Görevler bölünemez ya da terk edilemez.
- Paralel iş veya istasyona izin verilmez.
- İş istasyonlarının kapasiteleri sadece takt zamanı ve W_{MAX} ile sınırlandırılır.

Bu şartlar doğrultusunda demontaj hattının minimum toplam maliyet, minimum çevrim süresi ve çalışanlar arasında eşit görev dağılımı amaçları ile dengelenmesi istenilmektedir.

3.2. Optimizasyon Modeli

Aşağıda bu çalışma kapsamında ele alınan ÇADHDP'nin çözümü için önerilen optimizasyon modeline yer verilmiştir.

İndisler ve Kümeler;

i, s sökme görev indisi

j, h istasyon indisi

k sökülen parça indisi

w çalışan indisi

b sökülen parça sayısını tanımlayan indis

TS görev kümesi $TS = \{1, \dots, I\}$

ST istasyon kümesi $ST = \{1, \dots, /TS/\}$

SW çalışan kümesi $SW = \{1, \dots, W\}$

SK sökülen parça kümesi $SK = \{1, \dots, K\}$

SB sökülen parça sayısını içeren küme $SB = \{1, \dots, B\}$

P^A_i i . görev için VE öncelik ilişki kümesi

P^O_i i . görev için VEYA öncelik ilişki kümesi

S^O_i i . görev için VEYA ardıl ilişki kümesi

Parametreler;

W_{MAX}	bir istasyona atanabilecek maksimum çalışan sayısı
$Time$	demontaj için tüm işlemlere ayrılacak toplam süre
FC	istasyon açma maliyeti
SC	sökülen ürünün elde bulundurma maliyeti
WC_w	w . çalışanın ücreti
C	çevrim süresinin hesaplanmasında kullanılan yardımcı parametre
L	pozitif büyük bir sayı
t_i	i . görevin standart işlem süresi
f_{iw}	i . görevin w . çalışan tarafından yeterlilik/beceri seviyesi
d_k	k . parçaya olan talep
m_{ik}	$m_{ik} = \begin{cases} m_{ik} \geq 1; & k. \text{ parça } i. \text{ görev sonucu } m_{ik} \text{ adet oluşuyorsa} \\ m_{ik} = -1; & k. \text{ parça } i. \text{ görevde kullanılıyorsa} \\ m_{ik} = 0; & \text{diğer durumlarda} \end{cases}$

Değişkenler;

$$x_{ijw} = \begin{cases} 1; & i. \text{ görev } j. \text{ istasyonda } w. \text{ çalışan tarafından yapılıyorsa} \\ 0; & \text{diğer durumlarda} \end{cases}$$

$$\forall i \in TS; \forall j \in ST; \forall w \in SW$$

$$y_{is} = \begin{cases} 1; & i. \text{ görev } s. \text{ görevden önce yapılıyorsa} \\ 0; & \text{diğer durumlarda} \end{cases}$$

$$\forall i \in TS; \forall s \in TS$$

$$st_j = \begin{cases} 1; & j. \text{ istasyon açıldıysa} \\ 0; & \text{diğer durumlarda} \end{cases}$$

$$\forall j \in ST$$

$$ws_{jw} = \begin{cases} 1; & w. \text{ çalışan } j. \text{ istasyona atandıysa} \\ 0; & \text{diğer durumlarda} \end{cases}$$

$$\forall j \in ST; \forall w \in SW$$

$$epor_{is} = \begin{cases} 1; & s. \text{ görev } i. \text{ görev için VEYA önceliğini yerine getiriyorsa} \\ 0; & \text{diğer durumlarda} \end{cases}$$

$$\forall i \in \{g: g \in TS \wedge P_g^{OR} \neq \emptyset\}; \forall s \in P_i^{OR}$$

$$pr_b = \begin{cases} 1; & b \text{ indisi sökülmesi gereken parça sayısına eşit ya da azsa} \\ 0; & \text{diğer durumlarda} \end{cases}$$

$$\forall b \in SB$$

$$CT \quad \text{çevrim süresi}$$

$$strt_{ij} \quad i. \text{ görevin } j. \text{ istasyonda başlama süresi}$$

$$\forall i \in TS; \forall j \in ST$$

$$p_i \quad i. \text{ görev için sökülmesi gereken ürün sayısı}$$

$$\forall i \in TS$$

$$Product \quad \text{sisteme girmesi gereken ürün sayısı}$$

MKW çalışanların maksimum işlem süresi
 MNW çalışanların minimum işlem süresi

Amaç Fonksiyonları;

$$\text{Min } f1 = FC * \sum_{j \in ST} st_j + SC * Product + \sum_{w \in SW} WC_w * \left(\sum_{j \in ST} wS_{jw} \right) \quad (1)$$

$$\text{Min } f2 = CT \quad (2)$$

$$\text{Min } f3 = MKW - MNW \quad (3)$$

Kısıtlar;

$$\sum_{j \in ST} \sum_{w \in SW} x_{ijw} \leq 1 \quad \forall i \in TS \quad (4)$$

$$\sum_{i \in TS} x_{ijw} - |TS| * wS_{jw} \leq 0 \quad \forall j \in ST; \forall w \in SW \quad (5)$$

$$\sum_{w \in SW} wS_{jw} - W_{MAX} * st_j \leq 0 \quad \forall j \in ST \quad (6)$$

$$\sum_{j \in ST} wS_{jw} \leq 1 \quad \forall w \in SW \quad (7)$$

$$p_i \leq L * \left(\sum_{j \in ST} \sum_{w \in SW} x_{ijw} \right) \quad \forall i \in TS \quad (8)$$

$$d_k \leq \sum_{i \in TAD} m_{ik} * p_i \quad \forall k \in SK \quad (9)$$

$$Product \geq p_i \quad \forall i \in TS \quad (10)$$

$$C * \sum_{b \in SB} pr_b \geq Product \quad (11)$$

$$CT * C * b \leq Time + L * (1 - pr_b) \quad \forall b \in SB \quad (12)$$

$$\sum_{w \in SW} x_{ijw} - \sum_{h=1}^j \sum_{w \in SW} x_{shw} \leq 0$$

$$\forall i \in \{g: g \in TS \wedge P_g^A \neq \emptyset\}; \forall s \in P_i^A; \forall j \in ST \quad (13)$$

$$\sum_{w \in SW} x_{ijw} - \sum_{h=1}^j \sum_{w \in SW} x_{shw} \leq L * (1 - epor_{is})$$

$$\forall i \in \{g: g \in TS \wedge P_g^O \neq \emptyset\}; \forall s \in P_i^O; \forall j \in ST \quad (14)$$

$$\sum_{s \in POR_i} epor_{is} \geq 1 \quad \forall i \in \{g: g \in TS \wedge P_g^O \neq \emptyset\} \quad (15)$$

$$\sum_{j \in ST} \sum_{w \in SW} x_{ijw} - \sum_{s \in SOR} \sum_{j \in ST} \sum_{w \in SW} x_{sjw} \geq 0 \quad \forall i \in \{g: g \in TS \wedge S_g^O \neq \emptyset\} \quad (16)$$

$$strt_{ij} + \sum_{w \in SW} x_{ijw} * t_i * f_{iw} \leq CT \quad \forall i \in TS, \forall j \in ST \quad (17)$$

$$strt_{ij} - strt_{sj} + L * \left(1 - \sum_{w \in SW} x_{sjw}\right) + L * \left(1 - \sum_{w \in SW} x_{ijw}\right)$$

$$\geq \sum_{w \in SW} x_{sjw} * t_s * f_{sw} \quad \forall i \in \{g: g \in TS \wedge P_g^A \neq \emptyset\}; \forall s \in P_i^A; \forall j \in ST \quad (18)$$

$$\begin{aligned}
& str_{t_{ij}} - str_{t_{sj}} + L * \left(1 - \sum_{w \in SW} x_{sjw}\right) + L * \left(1 - \sum_{w \in SW} x_{ijw}\right) \\
& \geq \sum_{w \in SW} x_{sjw} * t_s * f_{sw} - L * (1 - epor_{is}) \\
& \forall i \in \{g: g \in TS \wedge P_g^0 \neq \emptyset\}; \forall s \in P_i^0; \forall j \in ST
\end{aligned} \tag{19}$$

$$\begin{aligned}
& str_{t_{ij}} - str_{t_{sj}} + L * (1 - x_{sjw}) + L * (1 - x_{ijw}) + L * (y_{is}) \\
& \geq \sum_{w \in SW} x_{sjw} * t_s * f_{sw} \quad \forall i \in TS; \forall s \in TS; \forall j \in ST; \forall w \in SW
\end{aligned} \tag{20}$$

$$\begin{aligned}
& str_{t_{sj}} - str_{t_{ij}} + L * (1 - x_{sjw}) + L * (1 - x_{ijw}) + L * (1 - y_{is}) \\
& \geq \sum_{w \in SW} x_{ijw} * t_s * f_{sw} \quad \forall i \in TS; \forall s \in TS; \forall j \in ST; \forall w \in SW
\end{aligned} \tag{21}$$

$$MKW \geq \sum_{i \in TS} \sum_{j \in ST} x_{ijw} * t_i * f_{iw} \quad \forall w \in SW \tag{22}$$

$$MNW \leq \sum_{i \in TS} \sum_{j \in ST} x_{ijw} * t_i * f_{iw} + L * (1 - \sum_{j \in ST} ws_{jw}) \quad \forall w \in SW \tag{23}$$

$$\begin{aligned}
& str_{t_{ij}} \geq 0, CT \geq 0, MKW \geq 0, MNW \geq 0 \quad \forall i \in TS; \forall j \in ST \\
& p_i \geq 0 \text{ ve tamsayı, Product} \geq 0 \text{ ve tamsayı} \quad \forall i \in TS \\
& x_{ijw} \in \{0,1\}, st_j \in \{0,1\}, ws_{jw} \in \{0,1\} \quad \forall i \in TS; \forall j \in ST; \forall w \in SW \\
& y_{is} \in \{0,1\}, pr_b \in \{0,1\} \quad \forall i \in TS; \forall s \in TS; \forall b \in SB \\
& epor_{is} \in \{0,1\} \quad \forall i \in \{g: g \in TS \wedge P_g^0 \neq \emptyset\} \forall s \in P_i^0
\end{aligned} \tag{24}$$

Denklem seti (1)-(2)-(3), amaç fonksiyonlarını göstermektedir. Birinci amacı ifade eden denklem (1)'de, istasyon açma, elde bulundurma ve çalışan maliyetleri dikkate alınarak maliyet minimize edilir. İkinci amacı ifade eden denklem (2)'de, minimum çevrim süresi hesaplanır. Üçüncü amacı ifade eden denklem (3)'de ise, çalışanlar arasındaki işgücünü dengeleme amacı gösterilmektedir.

Denklem seti (4)-(5)-(6)-(7), modelin temel kısıtlarıdır. Bu kısıtlar ile değişkenlerin tanımlanması, birbirleri ile ilişki kurmaları, takım çalışmasının temel özelliklerinin yerine getirilmesi sağlanır. Denklem (4)'de, her görevin yapılma zorunluluğunun olmadığı ve bir görevin en fazla bir istasyonda bir çalışan tarafından yapılabileceği ifade edilir. Denklem (5)'de, görev atama değişkeni ile çalışan atama değişkeni arasında ilişki kurulur. Denklem (6)'da istasyon açma değişkeni ile çalışan atama değişkeni arasında ilişki kurulur, aynı zamanda bir istasyona en fazla W_{MAX} adet çalışanın atanabilmesine izin verilir. Denklem (7)'de ise bir çalışanın en fazla bir istasyonda çalışmasına izin verilir.

Denklem seti (8)-(9)-(10) ile parçalara olan taleplerin karşılanması sağlanır. Denklem (8)'de talebin karşılanması için parçanın üretildiği görevin yapılması şart koşulur. Denklem (9), parça-görev ilişkilerine göre görevlerin en az talebin karşılanacağı kadar yapılması koşullandır. Denklem (10), tüm talebin karşılanması için sisteme girmesi gereken en az ürün miktarının belirlenmesinde kullanılır.

Denklem seti (11)-(12), çevrim süresini takt zamanı ile sınırlayan kısıtlardır. Denklem (11)'de, demonte edilmesi gereken minimum ürün sayısı $C*b$ 'ye eşit oluncaya kadar pr_b değişkenine "1" değeri atanır. Böylece takt zamanı denklem (12)'de doğrusal bir kısıt ile hesaplanmış olur.

Denklem seti (13)-(14)-(15)-(16), öncelik ilişkilerinin kurulmasında kullanılır. Denklem (13), bir görevin yapılabilmesi için VE öncelik kümesindeki önceliklerinin yapılmış olmasını garantiler. Denklem (14) ve (15)'de, $VEYA$ öncelikleri olan bir görevin yapılabilmesi için $VEYA$ önceliklerinden en az birinin yapılmış olması koşullandırken, denklem (16)'da bir görevin $VEYA$ ardıllarından en fazla birinin yapılabileceği koşullandır.

Denklem seti (17)-(18)-(19), görevlerin istasyonlara atanmaları için zaman sıralamasının yapılmasında kullanılır. Denklem (17)'de, her görevin çevrim süresinden önce bitmesi sağlanır. Denklem (18) ve (19)'de ise, görevlerin başlama zamanlarında sırasıyla VE ve $VEYA$ önceliklerinin bitmiş olması dikkate alınır.

Denklem seti (20)-(21)'de, bir çalışanın bir görevi bitirmeden diğer görevlere başlayamaması sağlanır. Denklem seti (22)-(23), üçüncü amacın belirlenmesi için kullanılır ve bu kısıtlarda sırasıyla çalışanların en fazla ve en az işlem süreleri belirlenir. Son olarak kısıt (24) ise, değişkenlere getirilen işaret kısıtlamalarını ifade etmektedir.

Önerilen optimizasyon modeli klasik istasyon tabanlı ÇADHDP'nin çözümü için uygun yapıdadır. Bunun için $W_{MAX} = 1$ alınmalıdır. Modelde beceri seviyelerini dikkate almak için ise yeterlilik seviyelerine büyük bir sayı atanması yeterlidir.

Tablo 2. Optimizasyon modelinde mevcut olan kısıt sayısı

Kısıt	Miktar	Kısıt	Miktar
4	I	15	$ P^O_i *I$
5	$I*W$	16	$ S^O_i $
6	I	17	$I*I$
7	W	18	$ P^A_i *I$
8	I	19	$ P^O_i *I$
9	K	20	$I*I*I*W$
10	I	21	$I*I*I*W$
11	1	22	W
12	B	23	W
13	$ P^A_i *I$	24	$I*I*W + 2*I + I*W + B$
14	$ P^O_i *I$		$+ P^O_i + 2*I*I + 4$

Toplam kısıt sayısı: $6*I + 3*W + K + 2*B + 2*I*W + 3*I*I + I*I*W + 2*|P^A_i|*I + 3*|P^O_i|*I + |P^O_i| + |S^O_i| + 2*I*I*I*W + 5$

Modelde mevcut olan kısıt ve değişken sayıları sırasıyla Tablo 2’de ve Tablo 3’te verilmiştir. Veriler incelendiğinde özellikle görev sayısının artması ile problemin boyutunun oldukça arttığı görülmektedir. Artırılmış epsilon kısıt yöntemi 2 (AUGMECON2) ile GAMS® 23.5/CPLEX 12.2 çözücüsü kullanılarak önerilen model analiz edildiğinde, problem parametrelerinin ancak sekiz görev, dört parça ve dört çalışandan az olması durumunda makul sürede sonuç verdiği belirlenmiştir.

Tablo 3. Optimizasyon modelinde mevcut olan değişken sayısı

Değişken	Miktar	Değişken	Miktar
x_{ijw}	$I*I*W$	p_i	I
st_j	I	$strt_{ij}$	$I*I$
ws_{jw}	$I*W$	CT	1
y_{is}	$I*I$	MKW	1
pr_b	B	MNW	1
$epor_{isj}$	$ P^O_i $	$Product$	1

Toplam değişken sayısı: $I*I*W + 2*I + I*W + B + |P^O_i| + 2*I*I + 4$

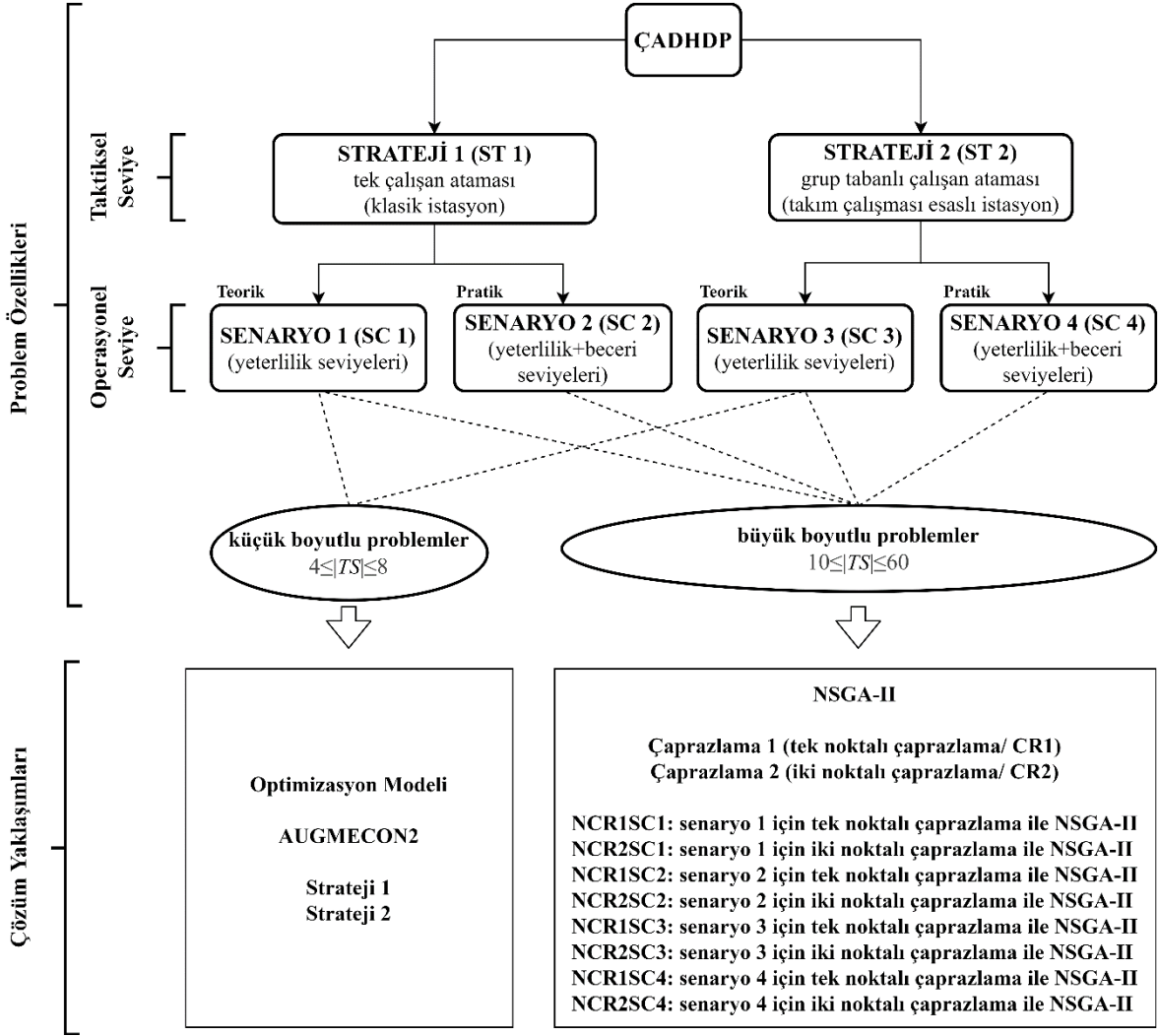
4. UYGULAMA

4.1. Deneysel Veri

Geri dönüşümün işletme açısından avantajlı olması için, demontaj hatlarının dengelemesinde alan ve zaman kapasitesinin etkili kullanılması önemlidir. Takım çalışması, öncelik ilişkileri dikkate alınmak koşuluyla birden fazla çalışanın farklı demontaj görevlerini aynı anda yapması olanağını sunar. Hiçbir öncelik ilişkisinin olmadığı varsayıldığında demontaj için ayrılan zamanın takım çalışmasındaki çalışan sayısı ile ters orantılı olacak şekilde azaldığı görülmektedir. Aynı zamanda birden fazla çalışanın aynı istasyona atanması bu işlem için gerekli alanın da azalmasına katkı sağlamaktadır. Takım çalışmasının etkilerini daha iyi analiz etmek için bu çalışmada tek çalışan ataması içeren klasik istasyon (ST1) ve grup tabanlı çalışan ataması içeren takım çalışması esaslı istasyon (ST2) olmak üzere iki farklı strateji taktiksel açıdan ele alınmıştır.

Teoride daha etkili analizler yapabilmek adına bütün çalışanların tüm görevleri yapabildiği varsayılabilirken, pratikte iş istasyonlarında bütün çalışanlar tüm görevleri yerine getirememektedir. Bu durum beceri seviyeleri ile ifade edilerek, yapılan çalışmada operasyonel seviye olarak ele alınmıştır. Çalışan beceri seviyelerinin problemde dikkate alınıp alınmadığı, iki farklı strateji ile birlikte değerlendirildiğinde çalışmada analiz edilen başlıca dört senaryo belirlenmiştir:

- i. Tek çalışan atamasına dayanan klasik istasyonda ÇADHDP'nin sadece yeterlilik seviyeleri ile ele alındığı senaryo 1 (SC1),
- ii. Tek çalışan atamasına dayanan klasik istasyonda ÇADHDP'nin yeterlilik ve beceri seviyeleri ile ele alındığı senaryo 2 (SC2),
- iii. Grup tabanlı çalışan atamasına dayanan takım çalışması esaslı istasyonda ÇADHDP'nin sadece yeterlilik seviyeleri ile ele alındığı senaryo 3 (SC3),
- iv. Grup tabanlı çalışan atamasına dayanan takım çalışması esaslı istasyonda ÇADHDP'nin yeterlilik ve beceri seviyeleri ile ele alındığı senaryo 4 (SC4).



Şekil 2. Problem özellikleri ve çözüm yaklaşımları

Çalışanların yeterlilik seviyelerinin farklı olmasından kaynaklanan görevlerin tamamlanma zamanlarındaki farklılıklar, bu çalışmada üç farklı yeterlilik seviyesi aralığı dikkate alınarak değerlendirilmiştir. Bu kapsamda görevlerin standart işlem süreleri ve çalışanların görevlere göre yeterlilik seviyelerinin bilindiği varsayılmıştır. Çalışanların yeterlilik seviyelerinin tekdüze (düzgün) dağıldığı varsayılarak üç farklı çalışan tipi belirlenmiştir:

- i. Çalışan yeterlilik seviyelerinin $[0,85 - 1,15]$ arasında düzgün dağıldığı heterojen çalışanlar,
- ii. Çalışan yeterlilik seviyelerinin $[0,9 - 1,1]$ arasında düzgün dağıldığı az heterojen çalışanlar,
- iii. Çalışan yeterlilik seviyelerinin aynı (yani $[1]$) olduğu homojen çalışanlar.

Çalışma kapsamında ele alınan ÇADHDP'nin özellikleri Şekil 2'de özetlenmiştir. Optimizasyon Modeli bölümünde ÇADHDP için geliştirilen modelin, artırılmış epsilon kısıt yöntemi 2 (AUGMECON2) ve GAMS® 23.5/CPLEX 12.2 çözücüsü kullanılarak problem parametrelerinin ancak sekiz görev, dört parça ve dört çalışandan az olması durumunda makul sürede sonuç verdiği belirtilmişti. Bu durum göz önüne alınarak yapılan çalışma kapsamında beş adet küçük boyutlu problem üretilerek AUGMECON2 yöntemi ile çözülmüştür. Küçük boyutlu problemlerin çözümü için SC1 ile SC3 senaryoları dikkate alınırken, çalışan yeterlilik seviyesi olarak az heterojen çalışan tipi kullanılmıştır. Üretilen küçük boyutlu problemlere ait veriler Tablo 4'te özetlenmektedir. Problemlerde bir istasyona atanabilecek maksimum çalışan sayısı (W_{MAX}) SC1'de bir, SC3'de iki alınmıştır. Parça talepleri, standart işlem süreleri, çalışan yeterlilik seviyeleri ve çalışan ücretlerinin düzgün (uniform) dağıldığı varsayılmıştır. Aynı zamanda çalışan ücretleri, çalışanların yetenek seviyeleri ortalamaları ile ters orantılı olacak şekilde üretilmiştir. Küçük boyutlu problemlere ilişkin teknolojik öncelik diyagramları Yılmaz (2020)'de bulunmaktadır.

Tablo 4. Küçük boyutlu problemlere ait veriler

Problem	Görev Sayısı	Parça Sayısı	Çalışan Sayısı	VEYA öncelik sayısı	VEYA ardıl sayısı	Talep	Görev Süreleri (saniyede)	Çalışan Ücretleri
P-1	4	4	4	1	0	U[50,100]	U[10,80]	U[200,400]
P-2	5	4	4	1	0	U[50,100]	U[10,80]	U[200,400]
P-3	6	4	4	1	1	U[50,100]	U[10,80]	U[200,400]
P-4	7	4	4	1	1	U[50,100]	U[10,80]	U[200,400]
P-5	8	4	4	2	0	U[50,100]	U[10,80]	U[200,400]

Erişilebilir demontaj süresi ($Time$)=10000; istasyon açma maliyeti(FC)=1000; sökülen ürünün elde bulundurma maliyeti (SC)=200; parti büyüklüğü (C)=10; pozitif büyük bir sayı(L)=100000; bir istasyona atanabilecek maksimum çalışan sayısı (W_{MAX})=2; ; yeterlilik seviyesi (f_{iw})=[0.9, 1.10]

Şekil 2'de çözüm yaklaşımlarında da görüldüğü üzere ÇADHDP için tüm senaryolar iki farklı çaprazlama operatörü ile baskın olmayan sıralamalı genetik algoritma-II (NSGA-II) yöntemi kullanılarak analiz edilmiştir. Çalışma kapsamında ele alınan altı adet büyük boyutlu problem, Gümüşkaya (2013) ve Aslan (2019) çalışmalarında bulunan teknolojik

diyagramlar üzerinden türetilmiştir. Üretilen büyük boyutlu problemlere rastgele VEYA ardıl ilişkisinin eklenmesi ile birlikte problemlere ait veriler Tablo 5’te özetlenmektedir. Problemlerde bir istasyona atanabilecek maksimum çalışan sayısı (W_{MAX}) SC1 ve SC2’de bir alınmıştır. Parça talepleri, standart işlem süreleri, SC3 ve SC4 için bir istasyona atanabilecek maksimum çalışan sayısı, çalışan yeterlilik seviyeleri ve çalışan ücretleri parametrelerinin düzgün (uniform) dağıldığı varsayılmıştır. Aynı zamanda çalışan ücretleri, çalışanların yetenek seviyeleri ortalamaları ile ters orantılı olacak şekilde üretilmiştir. Büyük boyutlu problemlere ilişkin teknolojik öncelik diyagramları Yılmaz (2020)’de bulunmaktadır.

Tablo 5. Büyük boyutlu problemlere ait veriler

Problem	Görev Sayısı	Parça Sayısı	Çalışan Sayısı	VEYA öncelik sayısı	VEYA ardıl sayısı	W_{MAX} (SC3 ve SC4 için)	Parça Talepleri	İşlem Süreleri (Saniye)
P-1	10	9	5	1	1	U[2,4]	U[50,100]	U[10,80]
P-2	20	18	10	2	1	U[2,4]	U[50,100]	U[10,80]
P-3	30	27	15	3	2	U[2,4]	U[50,100]	U[10,80]
P-4	40	35	20	5	2	U[2,4]	U[50,100]	U[10,80]
P-5	50	44	30	2	3	U[2,4]	U[50,100]	U[10,80]
P-6	60	55	40	7	4	U[2,4]	U[50,100]	U[10,80]

Erişilebilir üretim süresi (Time)=10000; istasyon açma maliyeti (FC)=1000; sökülen ürünün elde bulundurma maliyeti (SC)=200; parti büyüklüğü (C)=1

Üretilen büyük boyutlu problemlerin her biri belirlenen dört farklı senaryo ve üç farklı çalışan tipi ile çözülmüştür. Aynı zamanda bu problemler üzerinden standart işlem süreleri, parça talepleri ve bir istasyona atanabilecek maksimum çalışan sayısı parametreleri kullanılarak beşer farklı problem verisi üretilmiş ve üretilen her bir problem verisi geliştirilen algoritma ile beşer kez çözülmüştür. Bu durumda; altı farklı problem, dört farklı senaryo, iki farklı çaprazlama yöntemi, üç farklı yeterlilik seviyesi aralığı, parametreler üzerinden türetilen beşer farklı problem verisi ve beşer tekrar olmak üzere toplam 3600 koşum gerçekleştirilmiştir. Gerçekleştirilen koşumlara ait tüm veriler Yılmaz (2020)’de bulunmaktadır.

Çözüm yaklaşımlarının özeti Şekil 2’de mevcuttur. Çalışmanın ilerleyen bölümlerinde ele alınan yaklaşım, şekilde verilen kısaltmalar ile birlikte kullanılacaktır. Buna örnek olarak NCR1SC1 kısaltması, çözüm yöntemi olarak geliştirilen NSGA-II algoritmasında tek noktalı çaprazlama operatörünü ele alan beceri seviyelerinin ihmal edildiği klasik istasyon stratejisine dayanan ÇADHDP’ni ifade etmektedir.

4.2. Değerlendirme Metrikleri

Çalışma kapsamında analiz edilen problemlerde, iki farklı strateji ile iki farklı çaprazlama operatörü sonucu elde edilen Pareto optimal sonuçları karşılaştırmak ve analiz etmek için literatürde kullanılan C , DI_R , OS ve baskılanamayan çözüm sayısını ifade eden Nnd karşılaştırma metrikleri kullanılmıştır (Yılmaz, 2019).

A ve B iki farklı Pareto optimal sonuç olsun. A kümesindeki çözümlerin, B kümesinin ne kadarını baskıladığı $C(A,B)$ metriği ile ifade edilebilmektedir. İki kümeyi çözümlerin baskılanma durumuna göre karşılaştıran ve $[0,1]$ aralığında değer alan $C(A,B)$ oranı denklem (25) kullanılarak elde edilmektedir. Bu durumda A kümesinin B ’ye tercih edilebilmesi için, $C(A,B)$ oranının bire $C(B,A)$ oranından daha yakın olması beklenir. C metriği ile elde edilen oranların simetrik olduğu düşünülemez ($C(A,B) + C(B,A) \neq 1$). Denklemden $a < pb$, a çözümünün b çözümüne baskın olduğu anlamına gelmektedir. (Yılmaz, 2019).

$$C(A,B) = \frac{|\{b \in B | \exists a \in A : a < pb\}|}{|B|} \quad (25)$$

İkinci karşılaştırma metriği olan DI_R metriği ise bir A kümesinin, referans bir S kümesine uzaklığını ölçmektedir. Referans S kümesi, en iyi Pareto optimal sonucu ya da en iyiye yakın bir Pareto optimal sonucu ifade eder. Eğer en iyi S kümesi bilinmiyorsa, karşılaştırma yapılacak Pareto optimal sonuçlar birleştirilerek en iyiye yakın S^* Pareto optimal sonucu elde edilir. DI_R metriği denklem (26) ile hesaplanmaktadır. DI_R metriğinin daha küçük sonuç vermesi, en iyi çözüme daha yakın olduğunu ve daha iyi sonuç verdiğini göstermektedir (Yılmaz, 2019).

$$D1_R(A) = \frac{1}{|S^*|} \sum_{y \in S^*} \min\{d_{xy} : x \in A\} \quad ((26))$$

Denklemden d_{xy} , amaç fonksiyonlarının normalize edildiği uzayda x çözümü ile referans y çözümü arasındaki uzaklığı ifade etmektedir. p adet amaç fonksiyonunun olduğu bir problemde, x ile y çözümleri arasındaki d_{xy} uzaklığı denklem (27) ile hesaplanmaktadır.

$$d_{xy} = \sqrt{(f_1^*(y) - f_1^*(x))^2 + \dots + (f_p^*(y) - f_p^*(x))^2} \quad ((27))$$

Denklemden f_p^* , p . amaç fonksiyonunun normalize edilmiş değerini ifade etmektedir. Bir f_p^* değerinin S^* kümesi kullanılarak normalize edilme işlemi ise denklem (28) kullanılarak yapılmaktadır.

$$f_i^*(\cdot) = \frac{f_i(\cdot) - f_{\min_i}(S^*)}{f_{\max_i}(S^*) - f_{\min_i}(S^*)} \quad ((28))$$

$f_{\min_i}(S^*)$ ve $f_{\max_i}(S^*)$ değerleri sırasıyla S^* kümesindeki minimum ve maksimum değerleri göstermektedir.

Bir diğer karşılaştırma metriği olan OS ise iki Pareto optimal sonucun arasındaki yayılımı ölçmektedir. Bir A kümesi ile B kümesini yayılım açısından karşılaştırmak için $OS(A,B)$ metriği denklem (29) kullanılarak hesaplanmaktadır (Yılmaz, 2019).

$$OS(A,B) = \frac{\prod_{i=1}^p \max_{x \in A}(f_i(x)) - \min_{x \in A}(f_i(x)) + 0.5}{\prod_{i=1}^p \max_{x \in B}(f_i(x)) - \min_{x \in B}(f_i(x)) + 0.5} \quad ((29))$$

Denklemden Pareto optimal sonuçlarda herhangi bir amacın en küçük ve en büyük değerlerinin birbirine eşit olma durumu göz önüne alınarak, pay ve paydaya 0.5 değeri eklenmiştir. OS değerinin bir değerinden büyük olması A Pareto optimal sonucunun B Pareto optimal sonucuna göre daha iyi yayılım gösterdiğini ifade eder.

Pareto optimal sonuçlardaki baskılanamayan çözüm sayısını veren Nnd ise bir diğer karşılaştırma metriği olarak kullanılmıştır.

4.3. Artırılmış Epsilon Kısıt Yöntemi 2

Artırılmış epsilon kısıt yöntemi 2 (AUGMECON2) Mavrotas ve Florios (2013) tarafından çok amaçlı problemlerin Pareto optimal sonuçlarını belirlemek için geliştirilen bir yaklaşımdır. Yaklaşım temel olarak, amaçlardan bir tanesinin amaç fonksiyonu diğerlerinin kısıt olarak ifade edilmesi ile çözüme gidilen ε -kısıtlama yöntemine dayanmaktadır. ε -kısıtlama yönteminin tüm amaçlara tek tek uygulanması ile elde edilen sınır değerleri tablosunun içerdiği aralıkları daraltarak daha kısa zamanda Pareto optimal sonuçlara ulaşmak için artırılmış epsilon kısıt yöntemi (AUGMECON) geliştirilmiştir.

Tablo 6. Küçük boyutlu problemlere ait sınır değerleri tablosu

		Strateji 1			Strateji 2		
		<i>f1</i>	<i>f2</i>	<i>f3</i>	<i>f1</i>	<i>f2</i>	<i>f3</i>
Problem 1	Min <i>f1</i>	23000	80.42	6.13	22000	80.42	6.13
	Min <i>f2</i>	24390	74.14	32.96	23390	74.14	32.96
	Min <i>f3</i>	23030	80.42	4.33	22030	80.42	4.33
Problem 2	Min <i>f1</i>	23370	84.11	14.64	23370	84.11	14.64
	Min <i>f2</i>	24770	66.95	28.27	23770	66.95	28.27
	Min <i>f3</i>	23530	90.75	12.10	23530	90.75	12.10
Problem 3	Min <i>f1</i>	21910	95.80	3.69	21910	95.80	3.69
	Min <i>f2</i>	23140	66.70	5.46	22140	66.70	5.46
	Min <i>f3</i>	23200	68.88	2.28	22200	68.88	2.28
Problem 4	Min <i>f1</i>	21340	88.67	9.86	20340	91.13	15.99
	Min <i>f2</i>	22740	55.79	0.91	22050	55.79	18.55
	Min <i>f3</i>	21460	89.20	0.53	21740	71.33	0
Problem 5	Min <i>f1</i>	22080	72.74	8.57	21080	84.55	8.57
	Min <i>f2</i>	23440	55.98	9.13	22440	55.98	9.13
	Min <i>f3</i>	22150	72.74	6.33	21150	86.67	6.33

AUGMECON yönteminde sınır değerleri tablosunu elde etmek için belirlenen herhangi bir amacın en iyi değeri ε -kısıtlama yöntemine göre hesaplandıktan sonra, bulunan değer amaç fonksiyonuna eşitlenerek diğer amaçlar için ε -kısıtlama yöntemi uygulanır. Böylece belirlenen ilk amacın en iyi değerinde diğer amaçların sırasıyla aldığı en iyi değerler hesaplanmış olur. Bu yöntem her bir amaç için tekrar edilir. Böylece AUGMECON yöntemine göre sınır değerleri tablosu hesaplanmış olur (Mavrotas, 2009). Aynı yöntem AUGMECON2 için de geçerlidir. Çözülen küçük boyutlu ÇADHDP için GAMS® 23.5/CPLEX 12.2 çözücüsünde sınır değerleri tablosunun bulunması için kullanılan kodlar Ek 1’de, klasik istasyon ve takım çalışması esaslı istasyon stratejileri kapsamında problemlerin çözümü için elde edilen sınır değerleri tabloları Tablo 6’da verilmiştir.

AUGMECON2 yönteminin devam eden sürecinde sınır değerleri tablosunda hesaplanan aralıklar kullanılarak belirli kurallar çerçevesinde model, bir ana döngüye tabi tutulmaktadır. Aşağıda üç amaçlı minimizasyon problemi için ana döngüde amaç fonksiyonlarını gösteren bir optimizasyon modeline yer verilmiştir.

Amaç fonksiyonu;

$$\text{Min } f_1(x) - \text{eps} * \left(\frac{S_2}{f \max_2 - f \min_2} + 10^{-1} \frac{S_3}{f \max_3 - f \min_3} \right) \quad (30)$$

Kısıtlar;

$$f_2(x) + S_2 = f \min_2 + t_2 * (f \max_2 - f \min_2) / q_2 \quad (31)$$

$$f_3(x) + S_3 = f \min_3 + t_3 * (f \max_3 - f \min_3) / q_3 \quad (32)$$

$$x \in S \text{ ve } S_i \in R^+ \quad (33)$$

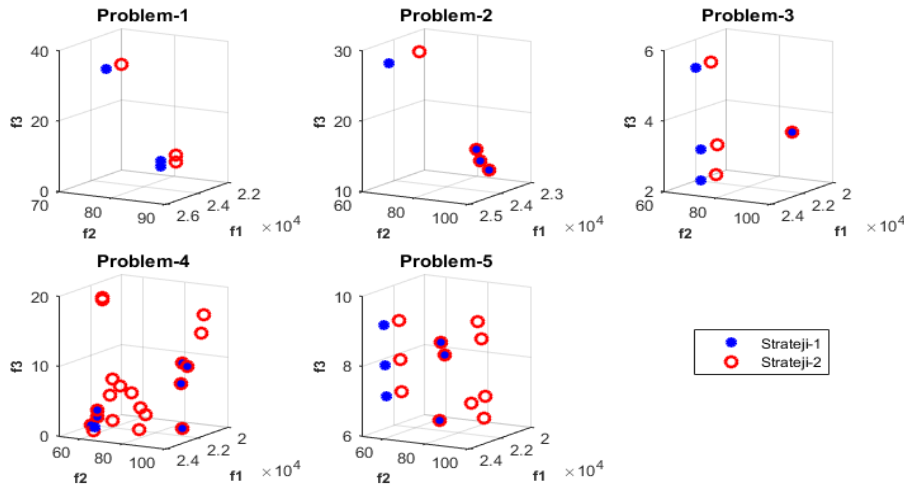
Maliyet, çevrim süresi ve iş yükü dengesizliği amaç fonksiyonlarının sırasıyla f_1 , f_2 ve f_3 ile ifade edildiği optimizasyon modelinde; S_2 ve S_3 aylak değişkenleri, q_2 ve q_3 sınır değerleri tablosundaki değişim aralıklarının bölünme sayısını, t_2 ve t_3 ise o anki iterasyonu ifade etmektedir. eps de genellikle $[10^{-6}, 10^{-3}]$ aralığında çok küçük bir sayıyı temsil etmektedir.

Optimizasyon modelinde de görüldüğü üzere AUGMECON2 yönteminde Pareto optimal sonuçları elde etmek için ilk amaç denklem (30) de görüldüğü gibi amaç fonksiyonu

olarak gösterilirken, diğer amaçlar denklem (31) ve (32)'de olduğu gibi kısıt olarak ifade edilir. Problemin çözümü sırasında kısıtlarda yer alan amaçların da en iyilenmesi için kısıtlara ait aylak değişkenler değişim aralıklarına bölünerek amaç fonksiyonuna eklenir. Böylece tüm amaçlar aynı anda en iyilenmiş olur. AUGMECON yöntemine göre önerilen optimizasyon modeli ile problem $(q_2+1)*(q_3+1)$ iterasyon boyunca çözülmektedir. AUGMECON2 yönteminde ise denklem (34)'de verilen atlama sayısı, ikinci amacın her iterasyonunda mevcut iterasyon sayısına eklenerek bu iterasyon sayısı azaltılmaktadır. Denklemde b parametresi, en iç döngüde bulunan ikinci amaç için iterasyon atlama miktarını ifade etmektedir. Denklem sonucu bulunan değer aşağı yuvarlanır (Mavrotas ve Florios, 2013).

$$b = tms \left(\frac{S_2 * q_2}{f_{max_2} - f_{min_2}} \right) \quad (34)$$

İterasyonlar sonucu elde edilen amaçların değerleri ile Pareto optimal sonuçlar elde edilmiş olur. Pareto optimal sonuçların bulunması için GAMS programında kullanılan kodlar Ek 2'de verilmiştir. Belirlenen sınır değerleri tablosuna göre AUGMECON2 yöntemi kullanılarak elde edilen Pareto optimal sonuçlar ise Ek 3'te ve Şekil 3'te verilmiştir.



Şekil 3. Küçük boyutlu problemler için elde edilen Pareto optimal sonuçlar

Ele alınan beş problemin iki farklı strateji için belirlenen dört metriğe göre karşılaştırmalı sonuçları ise Tablo 7'de verilmiştir.

Tablo 7. Küçük boyutlu problemlerin karşılaştırmalı sonuçları

Problem	C (ST1,ST2)	C (ST2,ST1)	DI_R (ST1)	DI_R (ST2)	OS (ST1,ST2)	Nnd (ST1)	Nnd (ST2)	CPU (saniye)
P1	0	1	1.09	1.00	1	3	3	172.61
P2	0.75	1	0.74	0.74	3.50	4	4	2054.2
P3	0.25	1	0.99	0.24	4.44	4	4	22044
P4	0.35	1	0.95	0.58	0.40	8	20	27310
P5	0.27	1	0.71	0.64	0.55	6	11	52375

Elde edilen karşılaştırmalı sonuçlara bakıldığında C metriğine göre tek çalışan atama stratejisi (ST1) ile grup tabanlı çalışan atama stratejisi (ST2) karşılaştırıldığında, ST2 stratejisinin ST1 stratejisine göre mutlak baskın olduğu $C(ST2,ST1)$ için verilen sonuçlardan açıkça görülmektedir. Tek çalışan atama stratejisi ile çözülen problemlerden elde edilen sonuçlar aynı zamanda grup tabanlı çalışan atama stratejisiyle de elde edilebileceği için bu beklenen bir durumdur. Ancak ele alınan problemlerde paralel yapılabilecek görevlerin kısıtlı olması ve bir istasyona atanabilecek maksimum çalışan sayısının az olması nedeni ile takım çalışması, özellikle çevrim süresi ve iş yükü dengeleme amaçları üzerinde etkisini çok fazla gösterememektedir.

İkinci bir karşılaştırma metriği olan DI_R metriğinin hesaplanmasında referans Pareto optimal küme normalize edilmiş değerler üzerinden (0, 0, 0) noktası olarak alınmıştır. DI_R metriği incelendiğinde ise C metriği ile benzer sonuçlar verdiği görülmektedir. Buna göre ST2 stratejisinin ST1 stratesine göre referans noktaya daha yakın sonuçlara ulaştığı sonucuna varılmaktadır.

İki farklı strateji OS metriği ile değerlendirildiğinde ise görev sayısının daha az, baskılanamayan çözüm sayısının eşit olduğu problemlerde ST1 stratejisi daha geniş aralıklarda yayılım göstermiştir. Görev sayısının biraz daha fazla ve baskılanmayan çözüm sayısının takım çalışması esaslı istasyon stratejisinde daha fazla olduğu durumda ise ST2 stratejisinin daha iyi yayılım gösterdiği görülmektedir. Bu farklılık, sınır değerleri tablosunda da görüleceği üzere ilk durumda ST2 strateji ile elde edilen Pareto optimal sonuçlarda, takım çalışmasının sadece birinci amaç üzerinde daha etkili sonuç vermesinden

kaynaklanmaktadır. Aynı zamanda birinci amaca ait en iyiye en uzak değerlerin iki strateji için eşit olması, ST1 stratejisi için *OS* metriğinin daha iyi sonuç vermesine neden olmuştur. Bu nedenle Pareto optimal sonuçları karşılaştırmada sadece bir metriğe odaklanmak doğru olmamaktadır.

Son olarak iki strateji baskılanamayan çözüm sayılarına göre karşılaştırıldığında ise ilk problemlerde eşit sonuçlar verirken, görev sayısının artması ile ST2 stratejisinin daha fazla sonuç verdiği görülmüştür. Bu durum takım çalışmasının alternatif çözüm miktarlarını arttırdığını göstermektedir. Karşılaştırma metrikleri ile elde edilen sonuçlar genel olarak değerlendirildiğinde ST2 stratejisinin hem çeşitlilik hem de en iyiye yakınlık olarak daha iyi sonuçlar verdiği sonucuna varılmaktadır.

4.4. Baskın Olmayan Sıralamalı Genetik Algoritma-II

Çalışma kapsamında ele alınan ÇADHDP'nin çözümü için baskın olmayan sıralamalı genetik algoritma-II (NSGA-II) metodundan yararlanılmaktadır. NSGA-II, özellikle en iyi çözüme ulaşmanın zor olduğu büyük boyutlu çok amaçlı problemlerde en iyi çözüme yakın sonuçlar elde etmek için kullanılan Pareto optimal sonuçları ele alan bir meta-sezgisel algoritmadır. Tek amaçlı problemlerin çözümü için kullanılan genetik algorithmada olduğu gibi bu algorithmada da en iyi çözüme ulaşmak için seçim (yeni bireyleri oluşturacak ebeveyn bireylerin rassal ya da belirli önceliklere göre belirlendiği operatör), çaprazlama (çözüm adaylarına yakın noktalarda en iyi çözümü aramak için kullanılan operatör) ve mutasyon (çözüm adaylarına uzak noktalarda en iyi çözümü aramak için kullanılan operatör) operatörleri kullanılmaktadır. Genetik algorithmadan farklı olarak NSGA-II'de elitizm (en iyi çözüm adaylarının değişmeden bir sonraki nesle aktarılması) için farklı bir yol izlenir. NSGA-II'de yeni çözüm adayları bir önceki nesil ile birleştirilerek baskılanma durumuna ve yığılma uzaklıklarına göre sıralanır (Deb vd., 2002). Sıralanan çözüm adayları popülasyon boyutuna göre seçilerek yeni jenerasyon belirlenmiş olur. Böylece her yeni jenerasyonda en iyi çözümler elde tutulmuş olur. Çalışma kapsamında yeni jenerasyonun belirlenmesi için MATLAB 2015a programında geliştirilen algoritma için kodlar Ek 4'te verilmiştir. Bu yöntem NSGA-II'nin diğer çok amaçlı meta-sezgisel algorithmalara göre en iyi çözüme daha hızlı yakınsamasını sağlar. Yapılan farklı alanlardaki birçok çalışmada, bu algoritma ile en iyi çözüme yakın çözümler elde edildiği görülmüştür (Oksuz vd., 2017; Yılmaz, 2019; Lian vd., 2018).

Yapılan çalışmada geliştirilen NSGA-II algoritmasında, kullanılan operatörler açısından ana döngü için sözde kodlar aşağıda ifade edilmektedir.

girdi: ps (popülasyonun büyüklüğü), mg (maksimum jenerasyon sayısı), cp (çaprazlama olasılığı), mp (mutasyon olasılığı)
çikti: Pareto-optimal sonuçlar
başla
 ps bireyden oluşan başlangıç popülasyonunu oluştur
baskılanamayan çözümlere ve yığılma uzaklıklarına göre sırala
her $k \in mg$ için
[Seçilim] (İkili turnuva yöntemi ile bireylerin seçimi)
[Çaprazlama] (cp 'ye göre tek noktalı (CR1) ya da iki noktalı (CR2) çaprazlama)
[Mutasyon] (mp 'ye göre değişim mutasyonu)
bir önceki jenerasyon ile yeni jenerasyonu birleştir
baskılanamayan çözümlere ve yığılma uzaklıklarına göre sırala
 ps kadarını seç ve yeni popülasyonu belirle
döndür k
mevcut çözümleri kaydet
bitir

Ana döngü için MATLAB 2015a programında kullanılan kodlar Ek 5'te verilmiştir. Bölümün devamında çalışma kapsamında ele alınan ÇADHDP için geliştirilen NSGA-II yönteminde kullanılan yapılara ve algoritmalara yer verilmiştir.

4.4.1. Kodlama ve Çözümleme Prosedürleri

NSGA-II yönteminde ilgili problemin en iyi çözümünü bulabilmek için birey adı verilen yapılar kullanılmaktadır. Yöntemde her bir birey çözüm adaydır ve probleme ait tüm kısıtları yerine getirmelidir. Bu çalışma kapsamında geliştirilen algorithmada problemin çözüm adaylarını temsil edecek birey yapısı Şekil 4'te her bir strateji için (klasik istasyon ve takım çalışması esaslı istasyon) sağ üstte gösterilmiştir. Önerilen birey yapısı üç satırdan ve I adet sütundan oluşmaktadır. Şekilde de görüleceği üzere bireyin ilk satırı görevleri, ikinci satırı ilk satırdaki görevleri yerine getiren çalışanları, üçüncü satırı ise çalışanların ve görevlerin atandığı istasyonları göstermektedir. Birey, bir istasyona en fazla W_{MAX} adet farklı çalışanın atanabildiği I adet görev ve W adet çalışan içeren bir demontaj sisteminde optimizasyon modelinde yer alan x_{ijw} değişkenine hitap etmektedir. Bu nedenle görevler, her bir strateji için şeklin sol üst köşesinde yer alan teknolojik diyagramdaki öncelik ilişkilerine göre sıralanmakta, bir çalışan iki farklı istasyona atanmamakta, bir istasyona en fazla W_{MAX}

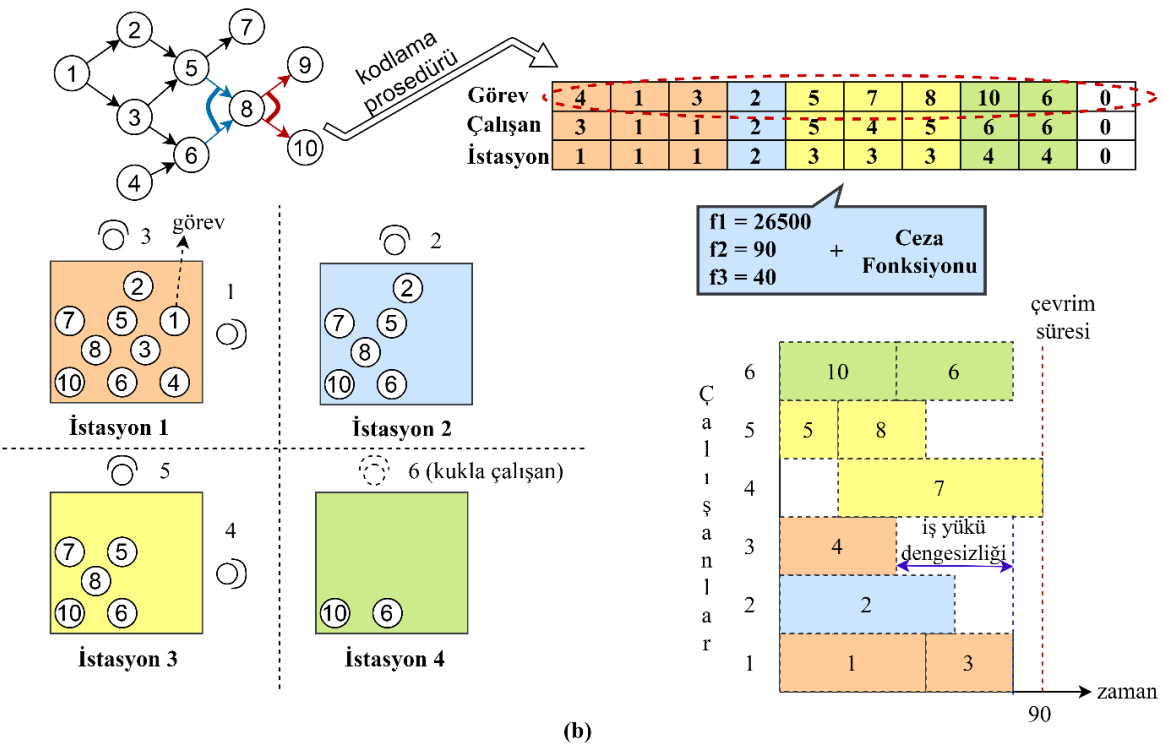
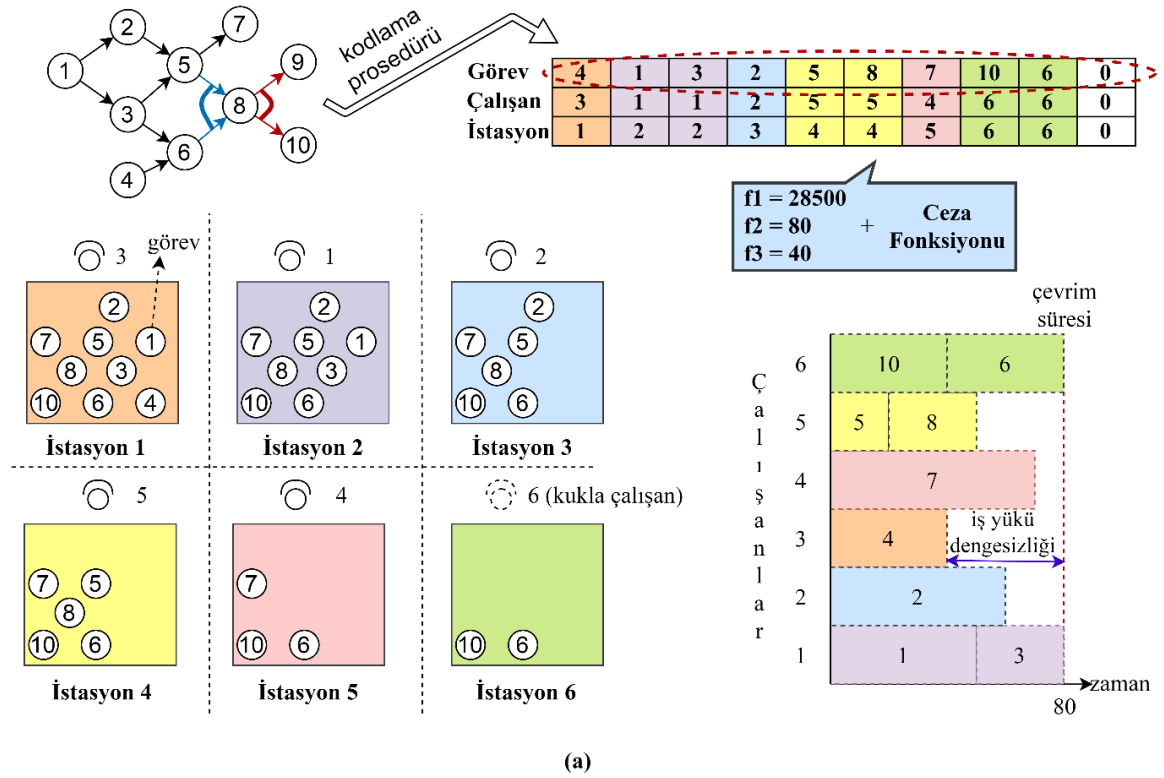
adet çalışan atanabilmekte ve istasyon süreleri en fazla ilgili birey için rastgele belirlenen çevrim süresi üst sınırı ile kısıtlanmaktadır. Bireyde *VEYA* ardıl ilişkisi ya da gereksiz olduğu için yapılması engellenen görevler, bireyin son sütunlarında “0” olarak görülmektedir.

Üretilen birey ile ifade edilen demontaj hattındaki istasyonlar, her bir strateji için Şekil 4’te sol alt kısımda simüle edilmiştir. Burada her renk kare bireydeki aynı renkte gösterilen istasyona gelen ürünleri, karenin içinde bulunan daire içindeki sayılar ise ürünlerde ilgili bireye göre yapılacak görevleri ifade etmektedir. Karenin etrafındaki daireler ve sayılar ise o istasyona atanan çalışanları sembolize etmektedir. Şekil 4 (a) ‘da klasik istasyon ele alındığı için her bir istasyonda tek çalışan bulunurken, Şekil 4 (b) ‘de takım çalışması esaslı istasyon ele alındığı için istasyonlarda bir ya da iki çalışan bulunabilmektedir.

Şekilde gerçek sistemlerde bulunmayan çalışanları gösteren kukla çalışanlar, kesik çizgili daireler ile ifade edilmektedir. Bireyde kukla çalışan bulunması istenmeyen bir durumdur ve bu nedenle bu çalışanlar en kötü yeterlilik/ beceri seviyelerine sahiptir. Kukla çalışan kullanılmadaki amaç ise problemin çözümünde çalışanları aynı çalışan havuzu içinden seçmektir. İlk popülasyonun üretiminde çözüm adayları tamamen rastgele üretildikleri için çözüm adayları en iyi çözüme uzak olmakta, bu da kukla çalışan görülme olasılığını arttırmaktadır. İlerleyen popülasyonlarda ise kukla çalışanların en kötü yeterlilik/beceri seviyesine sahip olması, operatörlerde geliştirilen algoritmaların bu çalışanları eleyecek şekilde tasarlanması ve kukla çalışan içeren bireylere ceza fonksiyonunun uygulanması nedeni ile kukla çalışanlar silinmektedir.

Bireylere ait çalışan-görev çizelgesi Şekil 4’te her bir strateji için sağ alt kısımda görülmektedir. Bu çizelge ile bireye ait çevrim süresi ve iş yükü dengesizlikleri görülebilmektedir.

Üretilen bireylerin en iyi çözüme yakınlığını ölçmek için uygunluk fonksiyonları kullanılır. Bu çalışmada kullanılan uygunluk fonksiyonu denklem (1)-(2)-(3) de verilen amaç fonksiyonlarına ceza fonksiyonunun eklenmesi ile elde edilmiştir.



Şekil 4. Kodlama ve çözümlene prosedürü (a) klasik istasyon stratejisi için, (b) takım çalışması esaslı istasyon stratejisi için

Çalışmada ceza fonksiyonu iki farklı durum için kullanılmaktadır: Kukla çalışanlar ve beceri seviyelerinde (SS_{iw}) engellenen görev-çalışan eşleşmeleri. Kukla çalışanlar beceri seviyelerinde tüm görevler için engellenmiş olarak gösterilir. Böylece ceza fonksiyonu tüm durumlar için sadece SS_{iw} kümesi üzerinden değerlendirilir. Ceza fonksiyonu denklem (35) kullanılarak hesaplanmaktadır. Denklemde mg değişkeni o anki iterasyon numarasını göstermektedir. Burada amaç geliştirilen algoritmanın ilk iterasyonlarında rasgele üretilen bireylerin çeşitliliğini korumaktır. İlerleyen iterasyonlarda ceza fonksiyonunun artması ile ceza maliyetine maruz kalan bireylerin elenmesi hızlanmaktadır.

$$\text{Ceza Fonksiyonu} = \left(\sum_{i \in TS} \sum_{w \in SW} SS_{iw} \right) * mg \quad (35)$$

Bireyin çözümlenmesini içeren sözde kodlar aşağıda yer almaktadır. Kullanılan parametrelerde optimizasyon modelinden farklı olarak PL_{iw} parametresi her bir çalışanın her bir göreve göre yeterlilik seviyesini, SS_{iw} parametresi ise beceri seviyesini ifade etmekte olup, f_{iw} parametresine hitap etmektedirler.

başla

her $b \in ps$ **için**

b bireyinin ceza fonksiyonunu hesapla

uygunluk fonksiyonu 1:

bireyin istasyon satırından açılan istasyon sayısını bul

denklem kümesi (8-10) ve bireyin görev satırındaki görevleri kullanarak b . birey için sisteme girmesi gereken en az ürün miktarını ($Product$ değişkeni) hesapla

bireyin çalışan satırından atanan çalışanları bul

denklem (1)'i ve ceza fonksiyonunu kullanarak uygunluk fonksiyonu 1'i hesapla

uygunluk fonksiyonu 2:

denklem kümesi (17-21) ve bireyin görev ve çalışan satırını kullanarak her bir görevin başlama ve bitiş zamanını hesapla

denklem (2)'yi ve ceza fonksiyonunu kullanarak uygunluk fonksiyonu 2'yi hesapla

uygunluk fonksiyonu 3:

denklem kümesi (22-23) ile bireyin görev ve çalışan satırını kullanarak çalışanların maksimum ve minimum işlem sürelerini hesapla

denklem (3)'ü ve ceza fonksiyonunu kullanarak uygunluk fonksiyonu 3'ü hesapla

döndür b

bitir

Çalışma kapsamında bireyin çözümlenmesi için MATLAB 2015a programında kullanılan kodlar Ek 6’te verilmiştir.

Kromozom üzerindeki klasik istasyon ile takım çalışması esaslı istasyon stratejileri arasındaki fark, klasik istasyonda W_{MAX} parametresinin “1” değerini almasıyla çözülmüştür. Önerilen kromozom yapısını ve çözümlemesini daha iyi ifade etmek için aşağıda her bir strateji için Şekil 2’de verilen veriler üzerinden birer sayısal örnek sunulmuştur.

Her iki strateji için de problemlere ait ortak parametreler şu şekildedir:

- Sayısal örneğe ilişkin teknolojik diyagram Şekil 2’de iki strateji için de aynı olacak şekilde verilmiştir.
- Teknolojik diyagramda yer alan 10 görevin standart işlem süreleri şu şekildedir: $t_1 = 50$; $t_2 = 60$; $t_3 = 30$; $t_4 = 40$; $t_5 = 20$; $t_6 = 40$; $t_7 = 70$; $t_8 = 30$; $t_9 = 50$; $t_{10} = 40$
- Sunulan sayısal örnekte VEYA ardıl ilişkisi olan görevlerden aynı, diğer görevden farklı birer adet parça söküldüğü varsayılarak $m_{1,1}=1$; $m_{2,2}=1$; $m_{3,3}=1$; $m_{4,4}=1$; $m_{5,5}=1$; $m_{6,6}=1$; $m_{7,7}=1$; $m_{8,8}=1$; $m_{9,9}=1$; $m_{10,9}=1$ alınmıştır (m parametresine ait verilmeyen değerler ‘0’ kabul edilmiştir).
- Örnekte yer alan dokuz farklı parçaya talepler şu şekildedir: $d_1 = 100$; $d_2 = 100$; $d_3 = 100$; $d_4 = 100$; $d_5 = 100$; $d_6 = 100$; $d_7 = 100$; $d_8 = 100$; $d_9 = 100$
- Örnekte görevleri yerine getirmek üzere beş çalışan belirlenmiştir. Belirlenen çalışanların tüm görevleri yapabildiği varsayılarak çalışanların yeterlilik seviyelerinin (PL_{iw}) tümü “1” kabul edilmiştir. Bu verilere göre de tüm çalışanların ücreti de şu şekilde belirlenmiştir: $WC_1=500$; $WC_2=500$; $WC_3=500$; $WC_4=500$; $WC_5=500$
- Diğer parametreler şu şekildedir: $FC=1000$; $SC=200$

İki strateji arasında fark oluşturan parametre olarak, klasik istasyon stratejisi için $W_{MAX}=1$; takım çalışması esaslı istasyon stratejisi için $W_{MAX}=2$ alınmıştır.

Verilen sayısal örnek için Şekil 2 (a) ‘da klasik istasyon stratejisi için önerilen kromozomu incelediğimizde: istasyon {1}’e çalışan {3} ve görev {4} atanmıştır. Aynı şekilde istasyon {2} ‘ye çalışan {1} ve sırasıyla görev {1,3} atanmış ve bu şekilde devam etmektedir. Görüldüğü üzere artan sırada istasyonlara atanan görevler öncelik ilişkilerini yerine getirmektedir. Kromozomda taleplerin karşılanması için {1, 2, 3, 4, 5, 6, 7, 8, 10} görevlerinin yapılması belirlenmiş, {9} görevinin yapılması VEYA ardıl ilişkisinden kaynaklanarak engellenmiş ve bu görev kromozomun son sütununda “0” değerlerinin oluşmasına neden olmuştur. Aynı zamanda belirlenen bu dokuz görevin yerine getirilmesi

için problemde yer alan beş çalışan yetersiz kaldığı için istasyon {6} da görev {10, 6} ya kukla çalışan {6} atanmıştır.

Klasik istasyon için elde edilen bu kromozomun çözümlediğimizde şu yargılara varmaktayız:

- Kromozomdan şu değişken değerleri elde edilir: $x_{431} = x_{112} = x_{312} = x_{223} = x_{554} = x_{854} = x_{745} = x_{10-6} = x_{6-6} = 1$; $ws_{31} = ws_{12} = ws_{23} = ws_{54} = ws_{45} = 1$; $st_1 = st_2 = st_3 = st_4 = st_5 = st_6 = 1$. Buradan toplam altı istasyonun açıldığı ve demontaj işleminin yapılması için çalışan {1, 2, 3, 4, 5} 'in hatta atandığı sonuçları elde edilmektedir.
- Kromozomun görev satırına bakıldığında görev {1, 2, 3, 4, 5, 6, 7, 8, 10} yapıldığı görülmektedir. m parametresinden bu görevlerden birer adet {1, 2, 3, 4, 5, 6, 7, 8, 9} parçalarının ayrıldığı bilinmektedir. Buradan tüm taleplerin karşılanması için sisteme en az $\max(100/1; 100/1; 100/1; 100/1; 100/1; 100/1; 100/1; 100/1; 100/1) = 100$ adet ürünün girmesinin zorunlu olduğu sonucu çıkmaktadır. Buraya kadar elde edilen verilerden denklem (1) kullanılarak birinci uygunluk fonksiyonu $6 \times 1000 + 200 \times 100 + 5 \times 500 = 28500$ olarak hesaplanır.
- Klasik istasyon stratejisinde her bir istasyonda tek çalışan olduğu ve görevler öncelik ilişkilerine göre sıralandığı için yapılan görevlerin işlem sürelerini toplamak istasyonların sürelerini hesaplamak için yeterli olacaktır. Her bir çalışanın her bir görevi yapma yeterliliği "1" olduğu için süreler standart işlem süreleri üzerinden değerlendirilebilir. Bu durumda istasyon süreleri; istasyon {1} için $t_4 = 40$, istasyon {2} için $t_1 + t_3 = 50 + 30 = 80$, istasyon {3} için $t_2 = 60$, istasyon {4} için $t_5 + t_8 = 20 + 30 = 50$, istasyon {5} için $t_7 = 70$; istasyon {6} için $t_{10} + t_6 = 40 + 40 = 80$ olur. Çevrim süresi maksimum istasyon süresi olarak değerlendirilirse ikinci uygunluk fonksiyonu 80 değerini alır.
- Yine çalışan becerilerinin "1" olduğu göz önüne alınarak her bir çalışanın yaptığı işlem süreleri hesaplanırsa; çalışan {1}'in işlem süresi $t_1 + t_3 = 50 + 30 = 80$; çalışan {2}'nin $t_2 = 60$; çalışan {3}'ün $t_4 = 40$; çalışan {4}'ün $t_7 = 70$; çalışan {5}'in $t_5 + t_8 = 20 + 30 = 50$ olarak bulunur. Buradan üçüncü uygunluk fonksiyonu denklem (3) kullanılarak hesaplanırsa $80 - 40 = 40$ olarak bulunmuş olur.
- Beceri seviyelerinin "0" olmasına rağmen kromozomun iki adet kukla çalışan içermesi, kromozoma $2 * mg$ (o anki iterasyon sayısı) kadar ceza fonksiyonu eklenmesine neden olur. Kromozomun onuncu jenerasyondaki bireye ait olduğu kabul edilirse bu durumda $2 * 10 = 20$ ceza fonksiyonu tüm uygunluk

fonksiyonlarına eklenir. Böylece uygunluk fonksiyonlarının son değerleri $f1=28520, f2=100$ ve $f3=60$ olarak hesaplanır.

Verilen sayısal örnekte Şekil 2 (b) 'de takım çalışması esaslı istasyon stratejisi için önerilen kromozomu incelendiğinde ise klasik istasyondan farklı olarak istasyon {1}'e çalışan {3, 1}'in atandığı görülür. Böylece çalışan {3}'e görev {4} atanırken çalışan {1}'e görev {1, 3} atanarak üç görevin aynı istasyona atanabilmesine izin verilmiştir. Ancak takım çalışması esaslı istasyon stratejisinde görevlerin başlama süreleri belirlenirken optimizasyon modelinde denklem seti (17-19)'da yer alan öncelik ilişkilerini göz önüne alınması ve denklem seti (20-21)'de yer alan çalışanların boşta olma durumunun göz önüne alınması durumlarına dikkat edilir. Şekil 2 (b)'deki grafikten de görüleceği üzere bu stratejide görev {3} ün başlayabilmesi için hem öncelik ilişkisinden hem de aynı çalışana atanan iki farklı görev olmaları nedeni ile görev {1}'in bitmiş olması gerekmektedir. Benzer şekilde istasyon {3}'e atanan görev {7}'nin başlayabilmesi için farklı çalışanlara atanan iki görev olmalarına rağmen öncelik ilişkisi nedeni ile görev {5}'in bitmiş olması gerekmektedir. Bu durum dışında kromozom yapısı klasik istasyonda anlatılanlar ile benzerlik göstermektedir.

Takım çalışması esaslı istasyon stratejini ele alan kromozom yapısı çözümlendiğinde ise şu yargılara ulaşılmaktadır:

- Kromozomdan şu değişken değerleri elde edilir: $x_{431} = x_{111} = x_{311} = x_{222} = x_{553} = x_{853} = x_{743} = x_{10-4} = x_{6-4} = 1$; $ws_{31} = ws_{11} = ws_{22} = ws_{53} = ws_{43} = 1$; $st_1 = st_2 = st_3 = st_4 = 1$. Buradan toplam açılan istasyon sayısının dört ve hatta atanan çalışanların {1, 2, 3, 4, 5} olduğu görülmektedir.
- Kromozomun görev satırına bakıldığında görev {1, 2, 3, 4, 5, 6, 7, 8, 10} yapıldığı görülmektedir. m parametresinden bu görevlerden birer adet {1, 2, 3, 4, 5, 6, 7, 8, 9} parçalarının ayrıldığı bilinmektedir. Buradan tüm taleplerin karşılanması için sisteme en az $\max(100/1; 100/1; 100/1; 100/1; 100/1; 100/1; 100/1; 100/1; 100/1)$ = 100 adet ürünün girmesinin zorunlu olduğu sonucu çıkmaktadır. Buraya kadar elde edilen verilerden denklem (1) kullanılarak birinci uygunluk fonksiyonu $4 \times 1000 + 200 \times 100 + 5 \times 500 = 26500$ olarak hesaplanır.
- Takım çalışması esaslı istasyon stratejisinde istasyonların işlem süreleri belirlenirken yukarıda da belirtildiği gibi öncelik ilişkileri ve çalışanların boşta olma durumları dikkate alınmaktadır. Tüm çalışanların yeterlilik seviyeleri "1" kabul edildiği için bu durumda istasyon {3} için istasyon işlem süresi Şekil 2 (b)'deki grafikten hareketle $t_5 + t_7 = 20 + 70 = 90$ olarak hesaplanır. Aynı şekilde diğer

istasyonların işlem süreleri de hesaplandığında tüm istasyonların işlem süreleri sırasıyla {80, 60, 90, 80} olmaktadır. Sonuç olarak çevrim süresi maksimum istasyon süresi olarak değerlendirildiğinde ikinci uygunluk fonksiyonu 90 olarak hesaplanır.

- Yine çalışan becerilerinin “1” olduğu göz önüne alınarak her bir çalışanın yaptığı işlem süreleri hesaplanırsa; çalışan {1}'in $t_1 + t_3 = 50+30=80$; çalışan {2}'nin $t_2 = 60$; çalışan {3}'ün $t_4 = 40$; çalışan {4}'ün $t_7 = 70$; çalışan {5}'in $t_5 + t_8 = 20+30=50$ olarak hesaplanır. Buradan üçüncü uygunluk fonksiyonu denklem (3) kullanılarak hesaplandığında $80-40=40$ olarak bulunmuş olur.
- Beceri seviyelerinin “0” olmasına rağmen kromozomun iki adet kukla çalışan içermesi, kromozoma $2*mg$ (o anki iterasyon sayısı) kadar ceza fonksiyonu eklenmesine neden olur. Kromozomun onuncu jenerasyondaki bireye ait olduğu kabul edilirse bu durumda $2*10 = 20$ ceza fonksiyonu tüm uygunluk fonksiyonlarına eklenir. Böylece uygunluk fonksiyonlarının son değerleri $f1=26520$, $f2=110$ ve $f3=60$ olarak hesaplanır.

4.4.2. İlk Popülasyonun Üretimi

Problemin çözüm adaylarını ifade eden kromozom yapısı bir önceki bölümde ayrıntılı olarak anlatılmıştır. Bu bölümde önerilen kromozom yapısına sahip bireylerin üretilmesi için geliştirilen algoritmaya ve dikkat edilen noktalara değinilecektir.

Sırasıyla görev, çalışan ve istasyon satırlarından oluşan kromozom yapısını gösteren diğer bir örnek Şekil 5 (a)' da görülmektedir. Şekildeki kromozom yapısına sahip bireylerden oluşan ilk popülasyonun üretiminde dikkat edilen noktalardan bir tanesi VEYA öncelik ilişkileridir. Şekil 5 (b)'deki VEYA öncelik ilişkisi ele alınsın. VEYA öncelik ilişkisi gereği görev {11} in yapılabilmesi için {7, 8} görevlerinden en az birinin yapılması şarttır. Bunun için ilk popülasyonun üretiminde her bir birey için bu görevler arasında bir seçim yapılarak VEYA öncelik ilişkileri VE öncelik ilişkilerine dönüştürülür. İlerleyen popülasyonlarda bu ilişkiler bireyden bireye farklılık göstereceği için VEYA öncelik ilişkilerinin korunması adına, bu ilişki türünde ebeveyn bireydeki sıralama korunmaktadır. Şekil 5'te bu ilişki türüne sahip genler, kromozomlarda sarı renkte gösterilmiştir.

İlk popülasyonun üretiminde dikkat edilen diğer bir nokta ise talebi karşılayacak parçaların söküldüğü görevlerin yerine getirilmesidir. Bunun için öncelikle her bir talebi

olan parça için o parçanın söküldüğü görevler belirlenir. Bir parçanın iki farklı görevden üretilmesi durumunda, iki görev arasında rasgele olarak seçim yapılır. Bu durum her bir bireyde yerine getirilen görevler arasında farklılıklara neden olmaktadır. İlerleyen jenerasyonlarda talebi olan parçaların karşılanmasında oluşabilecek sorunları önlemek için, yeni popülasyonun oluşturulmasında ebeveyn bireydeki görevler kullanılmaktadır. Böylece görev değişimi engellenerek talebi karşılayan görevler korunmuş olur.

Son olarak dikkat edilen noktalardan bir diğeri gereksiz olan görevlerin silinmesidir. Bu aşamada herhangi bir talebi olan parçayı sökmeyen ya da bu parçaların sökülmesi için yapılması gereken görevlere öncelik teşkil etmeyen görevlerin kromozoma atanması engellenmektedir. Bu aşama aynı zamanda VEYA ardıl ilişkileri bakımından öncelik ilişkilerinin sağlanmasına katkı sağlar. Ancak VEYA ardıl ilişkileri için asıl eleme işlemi görev sıralamasının yapıldığı aşamada gerçekleşmektedir.

Bireylerde çevrim süresi amacı kapsamında çeşitlilik oluşturmak adına her bir birey için takt zamanı ile çevrim süresi alt sınırı arasında rastgele bir çevrim süresi üst sınırı belirlenerek, istasyonlar bu çevrim süresi üst sınırına göre sınırlandırılmıştır. Bu kapsamda takt zamanı denklem seti (11-12) ile hesaplanmaktadır. Çevrim süresi alt sınırı ise çalışanların yeterlilik seviyeleri dikkate alınarak mevcut görevlerin maksimum işlem süresi hesaplanarak belirlenmektedir. Oluşturulan yeni jenerasyonlarda bireylerin çevrim süresi üst sınırı, ebeveyn bireyin çevrim süresi (f_2) hesaplanarak belirlenmektedir.

Bu kısıtlamalar altında ilk popülasyonun üretimi için geliştirilen algoritma aşağıdaki sözde kodlar ile ifade edilmektedir.

başla

her $b \in ps$ için

VEYA öncelik kümesinde ilişkisi olan görevler için öncelik teşkil edecek görevi b . birey için rasgele olarak belirle

taleplerin karşılandığı parçaların söküleceği görevi/ görevleri b . birey için rasgele olarak belirle

b . birey için gereksiz olan görevlerin atanmasını engelle
görevleri bireyin birinci satırına belirlenen öncelik ilişkilerini dikkate alarak rasgele sırala

denklem kümesi (8-10) ve bireyin görev satırındaki görevleri kullanarak b . birey için sisteme girmesi gereken en az ürün miktarını (*Product* değişkeni) hesapla

takt zamanını hesapla

sıralanan görevlerin her birinin alabileceği maksimum işlem süresini hesaplayarak çevrim süresi için bir alt sınır belirle

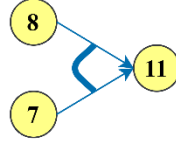
b . birey için takt zamanı ile çevrim süresinin alt sınırı arasında rasgele bir çevrim süresi üst sınırı belirle

çalışanlardan rasgele W_{MAX} adet seç
istasyon numarasını bire eşitle
her $i \in$ (atanan görevler) **için**
bireyin ikinci satırında i . göreve belirlenen çalışanlardan birini rasgele ata
bireyin üçüncü satırında i . göreve mevcut istasyonu ata
takım çalışması esasına göre i . görevin başlangıç ve bitiş sürelerini hesapla
eğer bitiş süresi, çevrim süresi üst sınırından büyük **ise**
çalışan kümesini ve istasyon numarasını güncelle, çalışan
kalmamışsa kukla çalışan ata
mevcut bireye yeni çalışan ve istasyon numarası ata
son
döndür i
bireyi popülasyonda tut
döndür b
oluşturulan bireyleri uygunluk fonksiyonları doğrultusunda baskılanamayan
çözümlere ve yığılma uzaklıklarına göre sırala (Deb vd., 2002)
bitir

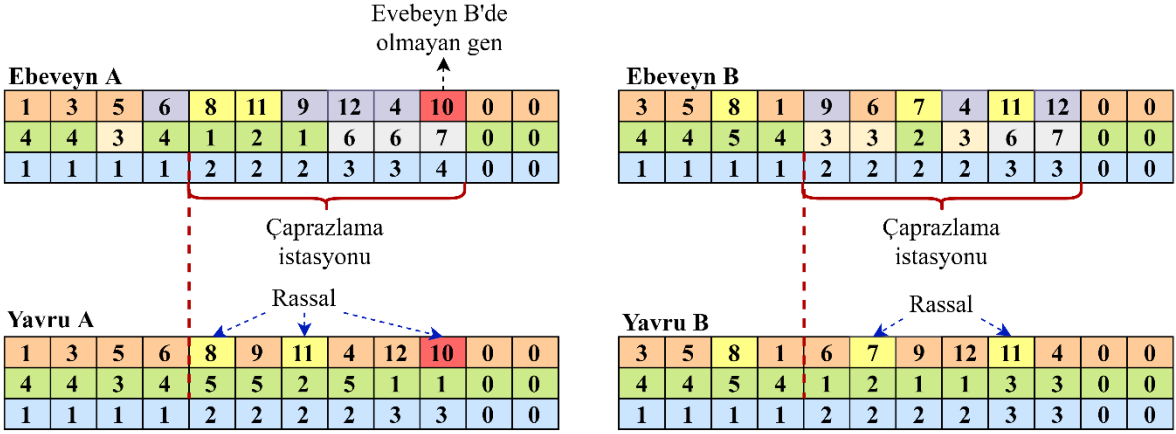
Sözde kodlar ile ifade edilen açıklamalar için MATLAB 2015a programında yazılan kodlar Ek 7'de verilmektedir. Algoritma ile üretilen ilk popülasyon, NSGA-II yöntemi kapsamında seçim, çaprazlama, mutasyon operatörleri ve bir sonraki jenerasyona aktarılacak bireylerin belirlenmesi işlemlerine durdurma koşulu sağlanıncaya kadar tabi tutulmaktadır.

Görevler	1	3	5	9	8	11	6	12	4	10	0	0
Çalışanlar	4	4	5	4	1	2	1	3	6	7	0	0
İstasyonlar	1	1	1	1	2	2	2	3	4	4	0	0

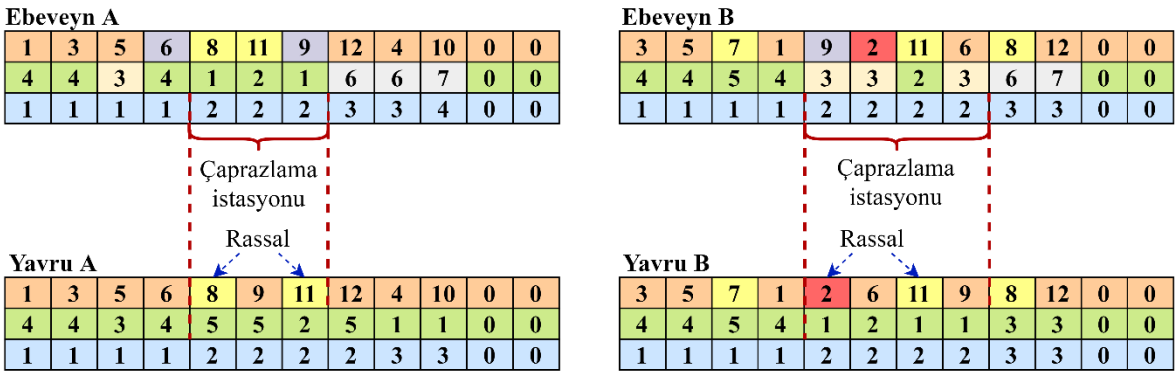
(a)



(b)



(c)



(d)

Şekil 5. (a) Kromozom yapısı, (b) VEYA öncelik ilişkisine örnek, (c) tek noktalı çaprazlama prosedürü, (d) iki noktalı çaprazlama prosedürü

4.4.3. Operatörler

Geliştirilen NSGA-II yönteminde ikili turnuva yöntemi ile seçim operatörü, tek noktalı ve iki noktalı olmak üzere iki farklı çaprazlama operatörü ve değişim mutasyonu operatörü kullanılmıştır. Kullanılan iki farklı çaprazlama operatöründen elde edilen Pareto optimal sonuçların belirli metrikler ile karşılaştırılması sonucunda, en iyi sonuçlara ulaşan çaprazlama operatörü üzerinden analizlere devam edilmiştir.

İkili turnuva yöntemi ile seçim operatöründe, mevcut popülasyondan seçilen rastgele iki bireyden baskınlık ve yığılma olarak daha iyi olan bireyin ebeveyn bireylere aktarılması ile çaprazlama operatöründe kullanılan ebeveyn bireyler oluşturulur. ps bireyden oluşan bir popülasyonda bu işlem $ps/2$ kez yapılarak ebeveyn A ve tekrar $ps/2$ kez yapılarak ebeveyn B ara popülasyonları oluşturulur.

Seçim operatörünün ardından kullanılan her iki çaprazlama operatöründe ortak olarak cp olasılığına göre ebeveyn bireyler çaprazlama işlemine tabi tutularak yavru bireyler oluşturulur. Çaprazlama işlemine tabi tutulan ebeveyn bireyler için her iki bireyde de açılan rastgele bir istasyon belirlenir. Şekil 5 (c)'de görüldüğü üzere tek noktalı çaprazlama işleminde belirlenen istasyonun başlangıç noktası kesme noktası olarak belirlenirken, Şekil 5 (d)'de görüldüğü üzere iki noktalı çaprazlama operatöründe belirlenen istasyonun başlangıç ve bitiş noktaları kesme noktaları olarak belirlenir. Kullanılan iki farklı çaprazlama yöntemi arasındaki ana fark budur. Çaprazlama işlemine tabi tutulmayan ebeveyn bireyler ise bütün genlerini aynı şekilde yavru bireye aktarır.

Tek noktalı çaprazlama (CR1) operatöründe ilk olarak Şekil 5 (c)'de görüldüğü gibi ebeveyn bireyin kesme noktasına kadar olan görev dizilimi aynı şekilde yavru bireye aktarılır. Kesme noktasından sonraki görev genlerinden *VEYA* öncelik ilişkisine sahip olan ve karşı ebeveynde olmayan görev genleri çıkarılır. Kalan görev genleri karşı ebeveyndeki sıralamalarına göre sıralandıktan sonra çıkarılan genler (*VEYA* öncelik ilişkisine sahip genlerin ebeveyndeki sıralamasının bozulmaması şartıyla) öncelik ilişkilerini dikkate alacak şekilde yeni sıralamaya rasgele olarak yerleştirilir. Bütün bu işlemler, sıralamalar arasında öncelik ilişkileri kurulup görevlerin yeni öncelik ilişkileri ile ilk iterasyonda olduğu gibi rastgele sıralanması ile gerçekleşmektedir. İki noktalı çaprazlama (CR2) operatöründe tüm bu işlemler iki kesme noktası arasındaki görevler için yapılmaktadır.

Bir sonraki adımda görev satırları oluşan yavru bireyler için ebeveynlerinin çevrim süreleri hesaplanarak çevrim süresi üst sınırları belirlenir. CR1 operatöründe karşı ebeveyn

bireyin kesme noktasından sonraki istasyonlara atanan çalışanlar, sıralamaları ile birlikte değerlendirilerek çevrim süresi üst sınırına göre yeni istasyonlar belirlenir. Karşı ebeveynden alınan çalışanlar arasında kesme noktasından önce kullanılan çalışanlar varsa bunlar yavru bireyde olmayan çalışanlarla ya da kukla çalışanlarla değiştirilir. Yavru bireye atanmamış çalışan olmasına rağmen, yavru bireyin kukla çalışan içermesi durumunda kukla çalışanlar atanmayan çalışanlar ile sıralama bozulmadan değiş tokuş yapılır. Yavru bireyde yeni istasyon açılması durumunda da atanmayan çalışanlar ya da kukla çalışanlar kullanılarak yeni sıralama oluşturulur. CR2 operatöründe CR1 operatöründen farklı olarak tüm bu işlemlerin sadece belirlenen istasyon için yapılmasıdır. Ancak yavru bireyde belirlenen istasyonda daha fazla ya da daha az görev yapılması durumunda çalışanlara atanan görevler kayma gösterebilmektedir.

Çaprazlama operatörü uygulanması için geliştirilen algoritmaya ait sözde kodlar aşağıdaki gibidir.

başla

her $b \in ps$ **için**

[0-1] aralığında rasgele bir sayı seç

eğer sayı cp 'den küçük **ise**

eğer CR1 operatörü seçildi **ise**

her iki ebeveynde (ebeveyn A ve ebeveyn B) de bulunan istasyonlardan rasgele birini çaprazlama istasyonu olarak belirle

ebeveyn A ve B'nin çaprazlama istasyonuna kadar olan genlerini sırasıyla yavru A ve B'ye aktar

ebeveyn A'nın çaprazlama istasyonu ve sonrasında gelen görev genlerinde, ebeveyn B'de bulunup VEYA öncelik ilişkisi bulunmayan görevleri belirle, aynı işlemi ebeveyn B için tekrarla

ebeveyn A'da belirlenen görevler arasında ebeveyn B'deki sıralamalarına göre SA adlı bir kümede öncelik ilişkisi kur, aynı işlemi ebeveyn B için SB kümesinde tekrarla

ebeveyn A'nın çaprazlama istasyonu ve sonrasındaki görev genlerinde, VEYA öncelik ilişkisi için yapılan ilk görevi belirle ve SA kümesine VE öncelik şeklinde aktar, aynı işlemi SB kümesi için tekrarla

SA ve SB ile VE öncelik kümesini birleştir

yavru A ve B'nin çaprazlama istasyonu ve sonraki görev dizilimini bu kümelere göre oluştur

yavru A ve B için çevrim süresi üst sınırını hesapla (Sırasıyla ebeveyn A ve B'nin çevrim süresi)

ebeveyn A ve B'nin çaprazlama istasyonu ve sonraki genlerindeki çalışan genlerini sıralamaları ile birlikte istasyonlara göre belirle

ebeveyn B'de istasyonlara göre belirlenen çalışan genlerinden yavru A'ya önceden atanmış olan çalışanları, atanmayan çalışanlar ile ya da kukla çalışanlar ile (tüm çalışanlar atanmış ise) değiş, aynı işlemi yavru B için yap

belirlenen ebeveyn B'nin çalışanlarına göre, yavru A'nın çaprazlama istasyonu ve sonraki çalışan ve istasyon genlerini oluştur, aynı işlemi yavru B için tekrarla

eğer CR2 operatörü seçildi **ise**

her iki ebeveynde (ebeveyn A ve ebeveyn B) de bulunan istasyonlardan rasgele birini çaprazlama istasyonu olarak belirle

ebeveyn A ve B'nin çaprazlama istasyonuna kadar olan genlerini sırasıyla yavru A ve B'ye aktar

ebeveyn A'nın çaprazlama istasyonundaki görev genlerinde, ebeveyn B'de bulunup VEYA öncelik ilişkisi bulunmayan görevleri belirle, aynı işlemi ebeveyn B için tekrarla

ebeveyn A'da belirlenen görevler arasında ebeveyn B'deki sıralamalarına göre SA adlı bir kümede öncelik ilişkisi kur, aynı işlemi ebeveyn B için SB kümesinde tekrarla

ebeveyn A'nın çaprazlama istasyonundaki görev genlerinde, VEYA öncelik ilişkisi için yapılan ilk görevi belirle ve SA kümesine VE öncelik şeklinde aktar, aynı işlemi SB kümesi için tekrarla

SA ve SB ile VE öncelik kümesini birleştir

yavru A ve B'nin çaprazlama istasyonundaki görev dizilimini bu kümelere göre oluştur

ebeveyn A ve B deki çaprazlama istasyonundan sonraki görev genlerini aynı sıra ile yavru bireylere aktar

yavru A ve B için çevrim süresi üst sınırını hesapla (Sırasıyla ebeveyn A ve B'nin çevrim süresi)

ebeveyn A ve B'nin çaprazlama istasyonundaki çalışan genlerini sıralamaları ile birlikte belirle

ebeveyn B'de belirlenen çalışan genlerinden ebeveyn A'da çaprazlama istasyonu dışındaki çalışan genlerinde olanları, atanmayan çalışanlar ya da kukla çalışanlar ile (tüm çalışanlar atanmış ise) değiş, aynı işlemi yavru B için yap

belirlenen ebeveyn B'nin çalışanlarına göre, yavru A'nın çaprazlama istasyonu çalışan ve istasyon genlerini oluştur, aynı işlemi yavru B için tekrarla

yavru bireylerin kalan genlerini ebeveyn bireylerin genlerini kullanarak tamamla

son

son

döndür *b*

yavru bireyleri yeni ara popülasyona aktar

bitir

Seçilim ve iki farklı çaprazlama operatörü için MATLAB 2015a programında yazılan kodlar Ek 8 ve 9'da verilmiştir. Çaprazlama operatöründen sonra oluşan ara popülasyon mutasyon operatörüne aktarılır. Mutasyon operatöründe ebeveyn bireyler *mp* olasılığına göre değişim mutasyonuna uğrayarak yeni popülasyonu oluştururlar. Değişim mutasyonuna uğrayan bireyler için mutasyon prosedürü Şekil 6'da görülmektedir.

Ebeveyn											
1	3	8	9	5	11	6	12	4	10	0	0
4	4	5	4	1	2	1	2	6	7	0	0
1	1	1	1	2	2	2	2	3	3	0	0

Yavru											
1	3	8	9	12	5	11	6	4	10	0	0
4	4	1	4	5	2	5	2	2	3	0	0
1	1	1	1	2	2	2	2	2	3	0	0

Şekil 6. Mutasyon prosedürü

Şekil 6’da görüldüğü üzere bu operatörde ilk olarak ebeveyn bireyde bulunan istasyonlardan rasgele bir tanesi seçilir. Seçilen istasyondaki görev genleri öncelik ilişkisi dikkate alınarak rasgele tekrar sıralanır. Daha sonra bireydeki çalışan genlerinden farklı iki çalışan seçilerek çalışanların görevleri birbirleri ile değiştirilir. Eğer ebeveyn birey kukla çalışan içeriyorsa ve çalışan havuzunda bireye atanmamış çalışan varsa kukla çalışanlar değiştirilir. Mutasyon operatörünün son aşamasında ise istasyonlar yeniden belirlenerek yeni birey oluşturulur.

Mutasyon operatörü uygulanması için geliştirilen algoritmaya ait sözde kodlar aşağıdaki gibidir.

başla
her $b \in ps$ **için**
 [0-1] aralığında rasgele bir sayı seç
eğer sayı mp 'den küçük **ise**
 birey için rasgele bir istasyon belirle
 istasyondaki görev genlerinin yerlerini öncelik ilişkilerini dikkate alarak rasgele değiştir
 rasgele seçilen iki çalışanın görevlerini değiştir
 birey kukla çalışan içeriyorsa ve atanmamış çalışan varsa, kukla çalışanı atanmayan çalışan ile deęiş
 ebeveyn bireyin çevrim süresini hesapla
 bireyin çalışan ve istasyon atamalarını güncelle
son
döndür b
bitir

Mutasyon operatörü için MATLAB 2015a programında yazılan kodlar Ek 10’da verilmiştir. Bu operatörden sonra oluşan yeni popülasyon eski popülasyon ile birleştirilir. Elde edilen popülasyonda $2*ps$ adet birey önce baskılanamama durumuna göre, daha sonra

baskılanamayan her bir küme kendi içinde yığılma uzaklıklarına göre sıralanır. Sıralanan bireylerden ps adet birey seçilerek yeni popülasyon oluşturulmuş olur.

Yapılan çalışmada geliştirilen NSGA-II algoritması, AUGMECON2 yöntemi ile çözülen küçük boyutlu problemlerden problem 5 ile test edilerek iki yöntemin sonuçları Tablo 8’de toplanmıştır. Tablodan da görüleceği üzere geliştirilen NSGA-II yöntemi problem 5 için Pareto optimal sonuçları bulmayı başarmıştır.

Tablo 8. Küçük boyutlu problemlerden problem 5 için Pareto optimal sonuçlar

Strateji-1						Strateji-2					
AUGMECON-2			NSGA-II			AUGMECON-2			NSGA-II		
$f1$	$f2$	$f3$	$f1$	$f2$	$f3$	$f1$	$f2$	$f3$	$f1$	$f2$	$f3$
22150	72.74	6.33	22150	72.74	6.33	22150	72.74	6.33	22150	72.74	6.33
23440	57.21	7.10	23440	57.21	7.10	21150	86.67	6.33	21150	86.67	6.33
23440	56.16	8	23440	56.16	8	21440	82.84	6.78	21440	82.84	6.78
22080	74.64	8.22	22080	74.64	8.22	21080	86.73	6.96	21080	86.73	6.96
22080	72.74	8.57	22080	72.74	8.57	22440	57.21	7.1	22440	57.21	7.1
23440	55.98	9.13	23440	55.98	9.13	22440	56.16	8	22440	56.16	8
						22080	74.64	8.22	22080	74.64	8.22
						22080	72.74	8.57	22080	72.74	8.57
						21080	84.55	8.57	21080	84.55	8.57
						22440	55.98	9.13	22440	55.98	9.13
						21150	83.51	9.07	21150	83.51	9.07

4.5. Parametre Değerlerinin Hesaplanması

Çalışmanın bu kısmında geliştirilen NSGA-II algoritmasında kullanılan operatörler için en iyi çözümleri veren girdi parametreleri belirlenmiştir.

Meta-sezgisel algoritmalarda kullanılan çaprazlama olasılığı, mutasyon olasılığı, popülasyon büyüklüğü ve iterasyon sayısı gibi parametreler algoritmanın verimliliğini oldukça etkilemektedir (Wu ve Che, 2019). Bu nedenle çözüme gidilmeden önce ele alınacak problem için en etkili parametreleri bulmak oldukça önemlidir.

Bu çalışmada parametre optimizasyonu kapsamında analiz edilmesi için görev sayıları 30 ve 60 olan iki farklı problem, iki noktalı çaprazlama operatörü ve beceri

seviyelerinin ihmal edildiği takım çalışması esaslı istasyon stratejisi yaklaşımı (NCR2SC3) ile test edilmiştir. Optimizasyon kapsamında popülasyon büyüklüğü (ps) 30 ve 30'dan az görev içeren problemler için 100, 30'dan fazla olan problemler için 150 olarak belirlenmiştir. Çaprazlama olasılığı (cp) ve mutasyon olasılığı (mp) için sırasıyla {0.7, 0.8, 0.9} ve {0.01, 0.05, 0.1} değerleri göz önüne alınmıştır. Literatürde bu aşamada kullanılmış olan DI_R metriğinin değerinin artarda 50 iterasyon boyunca değişim göstermemesi ya da iterasyon sayısının 500'e ulaşması durdurma kriteri olarak belirlenmiştir (Wang ve Liu, 2015). Algoritmanın sonlandığı iterasyonlar her bir kombinasyon için maksimum iterasyon sayısını (mg) belirlemiştir. Her bir kombinasyon kendi içinde beşer tekrar koşturularak ortalamaları alınmıştır. Her bir koşum için detaylı sonuçlar Ek 11'de verilmektedir. Ortalama değerlerin verildiği Tablo 9'da seçilen parametrelerin bulunduğu kombinasyon koyu renkte gösterilmiştir.

Tablo 9. Parametre optimizasyonu

Görev Sayısı	ps	cp	mp	mg	$DI_R(\text{NCR2SC3})$
$p \leq 30$ {10, 20, 30}	100	0.7	0.01	210	0.87
	100	0.7	0.05	240	0.71
	100	0.7	0.1	211	0.74
	100	0.8	0.01	133	1.05
	100	0.8	0.05	198	0.79
	100	0.8	0.1	144	0.87
	100	0.9	0.01	105	1.07
	100	0.9	0.05	191	0.87
	100	0.9	0.1	210	0.74
	$30 < p \leq 60$ {40, 50, 60}	150	0.7	0.01	166
150		0.7	0.05	176	0.58
150		0.7	0.1	321	0.47
150		0.8	0.01	149	0.64
150		0.8	0.05	186	0.57
150		0.8	0.1	192	0.63
150		0.9	0.01	135	0.75
150		0.9	0.05	152	0.67
150	0.9	0.1	199	0.56	

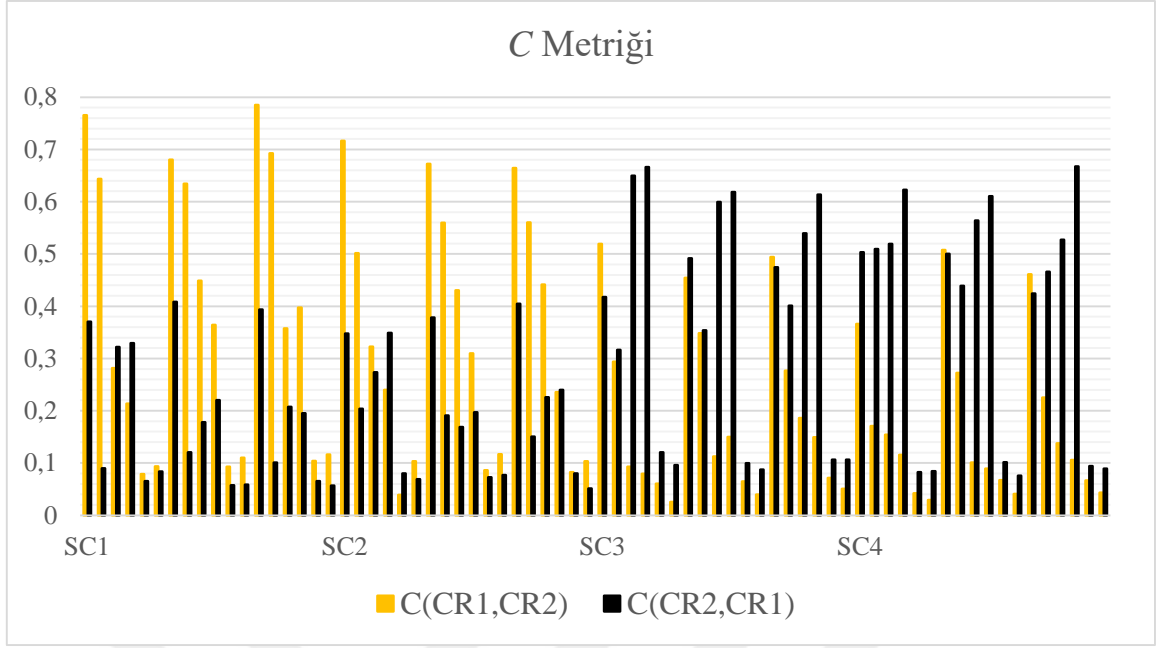
Yapılan parametre optimizasyonunda iki farklı problem grubu için kendi aralarında DI_R metriğinin son iterasyondaki değerinin en düşük olduğu kombinasyon seçilmiştir. Buna göre ele alınan büyük boyutlu problemlerin geliştirilen NSGA-II algoritması ile çözülmesi aşamasında 30 ve daha az görev içeren problemler için popülasyon büyüklüğü 100, çaprazlama olasılığı 0.7, mutasyon olasılığı 0.05, iterasyon sayısı 240; 31 ile 60 arasında görev içeren problemler için popülasyon büyüklüğü 150, çaprazlama olasılığı 0.7, mutasyon olasılığı 0.1, iterasyon sayısı 330 olarak belirlenmiştir.

4.6. Karşılaştırma Analizi

Parametre optimizasyonu sonucu bulunan parametre değerleri ve geliştirilen NSGA-II algoritması kullanılarak Deneysel Veri bölümünde belirlenen altı büyük boyutlu problem Intel® Core™ i7-8750H CPU 2.20 GHz ve 16 GB ram özelliklerine sahip bir bilgisayarda toplam 3600 kere koşturulmuştur. Koşumlara ait detaylı problem özellikleri yine Deneysel Veri bölümünde mevcuttur. Koşumlardan elde edilen Pareto optimal sonuçlar Yılmaz (2020)'de bulunmaktadır.

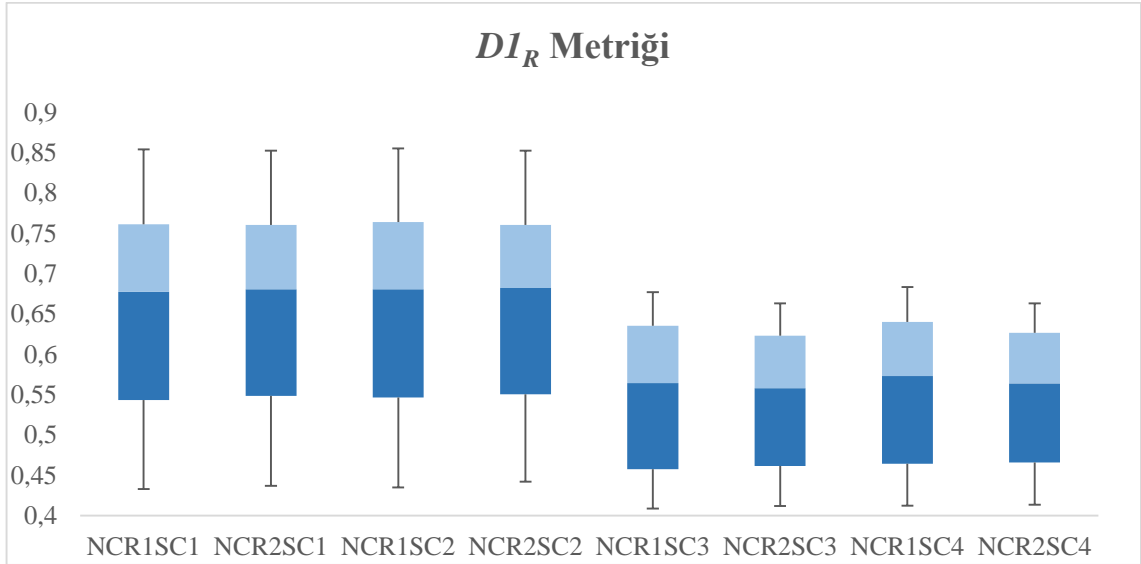
Çalışma kapsamında elde edilen Pareto optimal sonuçları analiz etmek için ilk olarak tüm Pareto optimal sonuçlar iki farklı çaprazlama operatörüne göre karşılaştırılmıştır. C , DIR , OS ve Nnd metrikleri kullanılarak karşılaştırılan Pareto optimal sonuçlardan elde edilen değerler Ek 12'de verilmiştir.

Şekil 7'de iki farklı çaprazlama operatörü için C metriğine ait karşılaştırmalı sonuçlara yer verilmiştir. Grafikten de görüleceği üzere klasik istasyon içeren senaryolarda tek noktali çaprazlama operatörü (CR1) ile elde edilen sonuçlar daha baskın sonuçlara ulaşabilirken, takım çalışması esaslı istasyon içeren senaryolarda iki noktali çaprazlama operatörü ile elde edilen sonuçlar daha baskın sonuçlara ulaşmıştır. Bu nedenle C metriğine bakarak iki çaprazlama operatörü arasında seçim yapmak çok zordur.

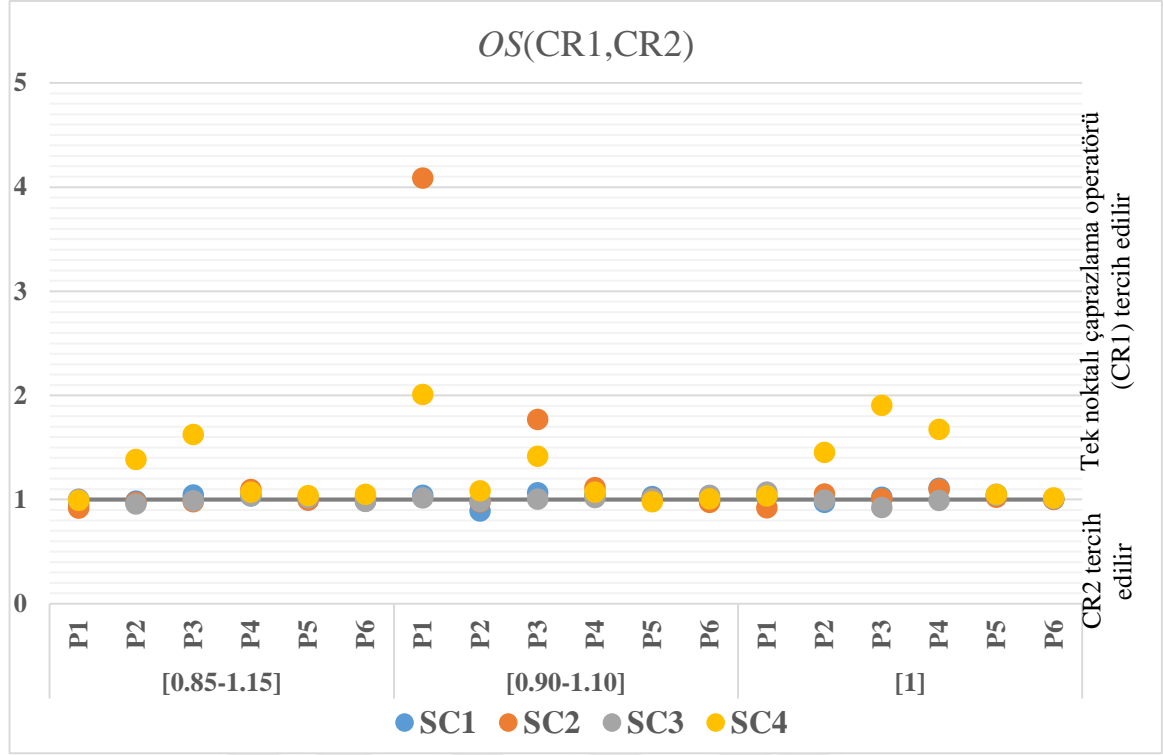


Şekil 7. İki farklı çaprazlama operatörü ile elde edilen Pareto optimal sonuçların C metriğine göre değerleri

Şekil 8’de verilen DI_R metriğine baktığımızda ise iki operatör ile elde edilen uzaklıklar arasında seçim yapılamayacak kadar küçük farklılıkların olduğu görülmektedir.



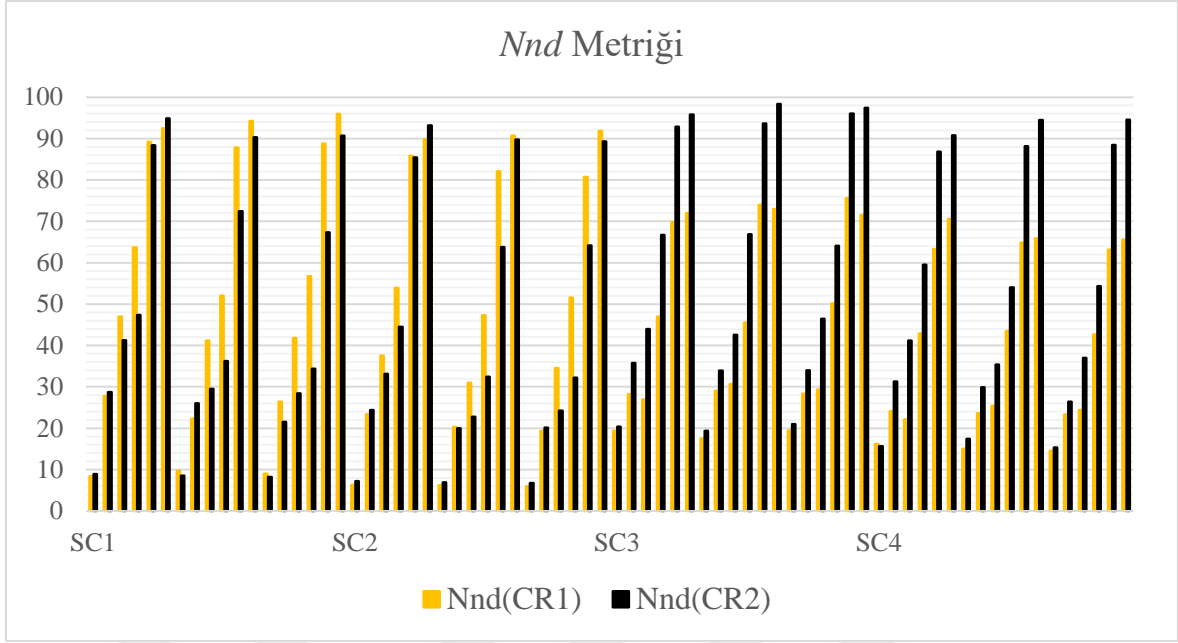
Şekil 8. İki farklı çaprazlama operatörü ile elde edilen Pareto optimal sonuçların DI_R metriğine göre değerleri



Şekil 9. İki farklı çaprazlama operatörü ile elde edilen Pareto optimal sonuçların OS metriğine göre değerleri

Bir diğer karşılaştırma metriği olarak kullanılan OS metriği için elde edilen değerler Şekil 9'da görülmektedir. Grafikte verilerin daha iyi analiz edilebilmesi için hesaplanan değerlerin amaç sayısına göre kökü (yani küp kökü) alınmıştır. OS metriğinin birden büyük olması ilgili yaklaşımın daha iyi yayıldığını ifade etmektedir. Buna göre Şekil 9'da da görüldüğü üzere CR1 operatörü ile elde edilen sonuçlar CR2 operatörüne göre daha iyi yayılım göstermiştir.

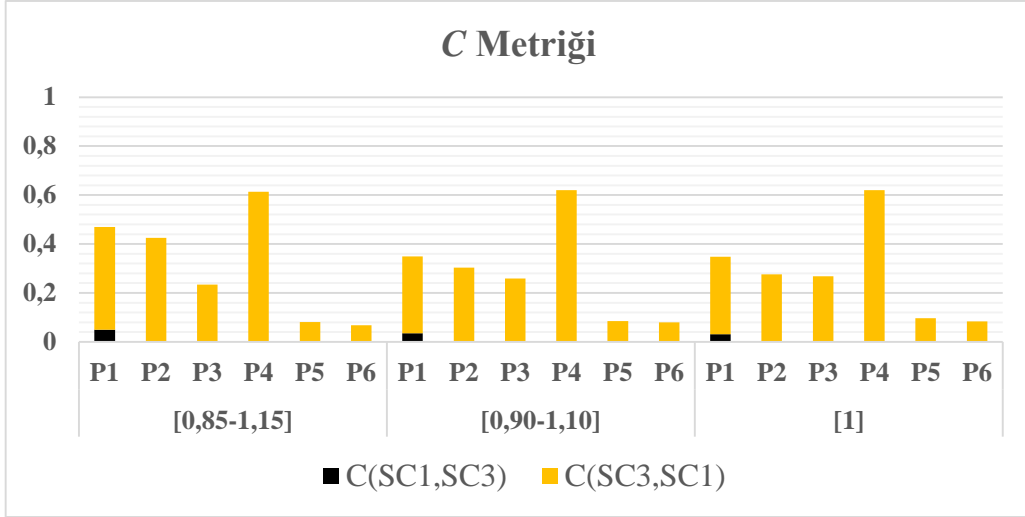
Son olarak Şekil 10'da baskılanamayan çözüm sayısını gösteren Nnd metriğine bakıldığında da C metriğinden elde edilen sonuçlara yakın bir durum söz konusudur. Buna göre SC1 ve SC2 senaryolarında genel olarak CR1 operatörü ile daha fazla çözüm üretilmişken, SC3 ve SC4 senaryolarında CR2 operatörü ile daha fazla çözüm elde edilmiştir.



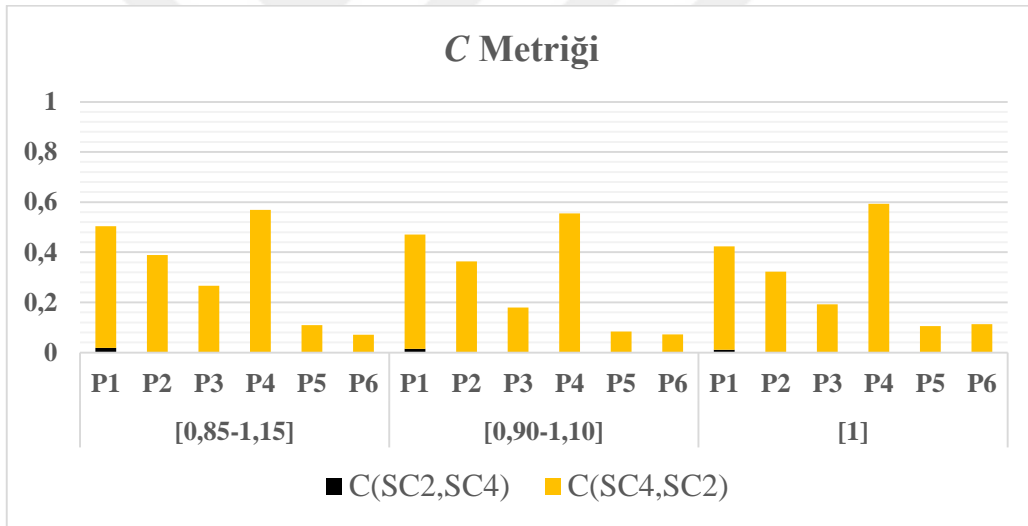
Şekil 10. İki farklı çaprazlama operatörü ile elde edilen Pareto optimal sonuçların baskılanamayan çözüm sayıları

Çaprazlama operatörleri için yapılan karşılaştırmalar sonucunda en belirgin sonuç *OS* metriğinde çözümlerin yayılımı ile elde edilmiştir. Bu nedenle çalışmanın kalan kısmında tek noktalı çaprazlama operatörü ile elde edilen çözümler dikkate alınarak devam edilmiştir.

Bir sonraki adım olarak ÇADHDP için ele alınan klasik istasyon stratejisi ile takım çalışması esaslı istasyon stratejisi aralarında karşılaştırılmış, sonuçlar analiz edilmiştir.. Beceri seviyelerinin ele alınıp alınmamasına göre teorik ve pratik senaryo olarak ele alınan yaklaşımlar birbirleriyle karşılaştırılmayacağı için bu kapsamda NCR1SC1 ile NCR1SC3 yaklaşımları kendi içinde, NCR1SC2 ile NCR1SC4 yaklaşımları kendi içinde analiz edilmiştir. Her bir problem iki farklı stratejiye göre karşılaştırılırken benzer çalışan heterojenlikleri göz önüne alınmıştır. Çalışan atama stratejilerinin önceden belirlenmiş metrikler açısından karşılaştırmalı sonuçları Ek 13'te verilmiş olup demontaj hattının performansı üzerindeki etkileri aşağıda detaylı olarak açıklanmıştır.



Şekil 11. Teorik senaryolar için elde edilen Pareto optimal sonuçların C metriğine göre karşılaştırmalı sonuçları

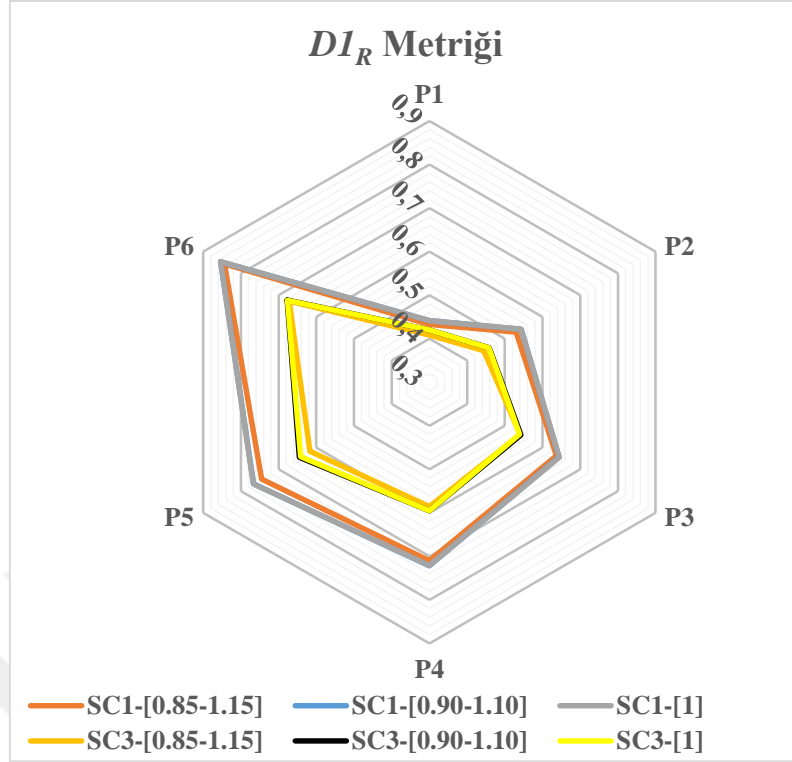


Şekil 12. Pratik senaryolar için elde edilen Pareto optimal sonuçların C metriğine göre karşılaştırmalı sonuçları

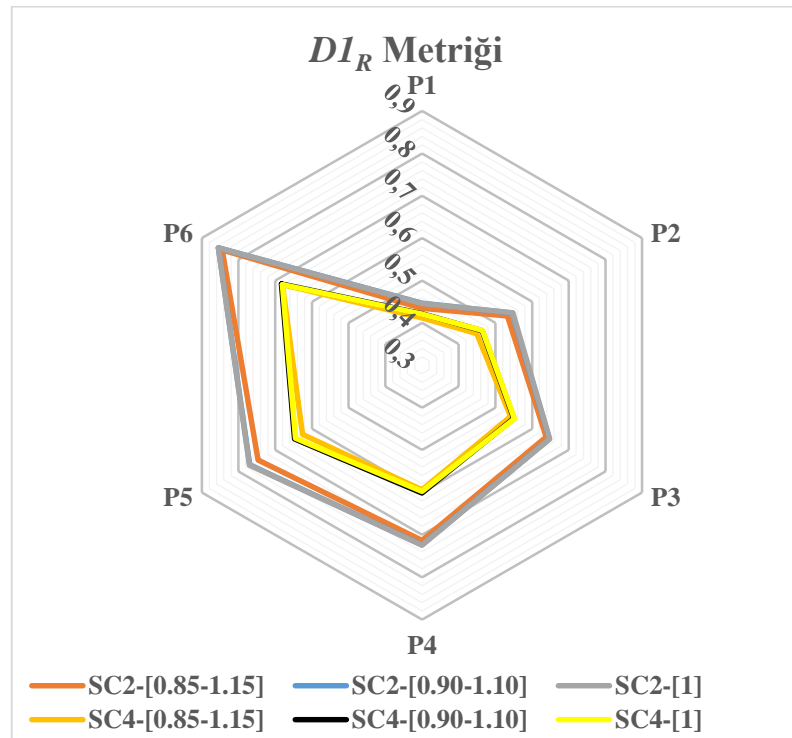
Karşı Pareto optimal sonuçlardaki çözümlere baskınlık oranının ifade edildiği C metriği ele alındığında teorik senaryolar için Şekil 11’de görüldüğü üzere takım çalışması esaslı istasyon stratejisi (ST2) ile elde edilen Pareto optimal sonuçların, klasik istasyon stratejisine (ST1) göre daha en iyi sonuçların elde edildiği görülmektedir. Benzer sonuçlar Şekil 12’de görüldüğü üzere pratik senaryolar için de geçerli durumdadır. Görev sayısının daha az olduğu problem 1’de ST1 stratejisi için her iki senaryoda ST2 stratejisine göre baskılanamayan sonuçların var olması paralel yapılacak görevlerin kısıtlı olmasından ve

takım çalışmasının avantajlarını küçük boyutlu problemlerde çok etkili şekilde gösterememesinden kaynaklanmaktadır. Önceki bölümlerde AUGMECON2 yöntemi ile elde edilen daha az görev sayısına sahip problemlerin çözümleri incelendiğinde benzer durum görülmektedir. Oysaki görev sayısının artması ile Şekil 11 ve 12’de de görüldüğü üzere ST1 stratejisinin ST2 stratejisine göre baskın çözüm sayısı kalmamaktadır. Elde edilen çıkarımlar özellikle paralel görev yapımının elverişli olduğu büyük boyutlu problemlerde her iki senaryo için maliyet, çevrim süresi ve iş gücü dengeleme amaçları bakımından takım çalışmasının daha avantajlı olduğunu doğrular. Bu da istasyonlarda takım çalışmasına izin vererek zaman ve kapasitenin daha etkili kullanılması ile sağlanmaktadır.

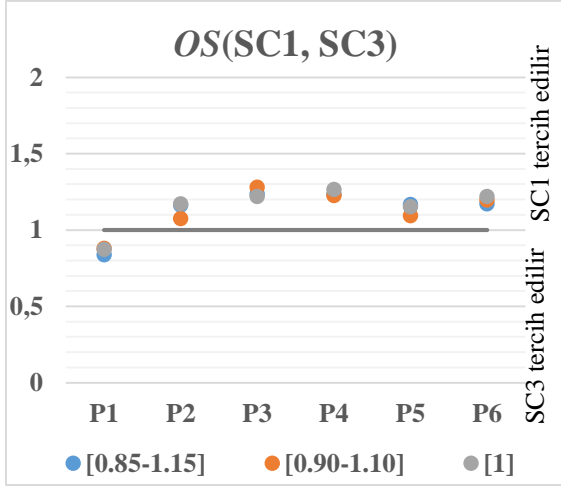
Referans Pareto optimal sonuçlara mesafeyi ele alan DI_R metriği aynı zamanda iki farklı Pareto optimal sonuçların karşılaştırılması için de kullanılabilir. Bu kapsamda elde edilen DI_R metrik değerleri referans (0, 0, 0) noktasına göre hesaplanmıştır. Metrikte elde edilen değerlerin küçük olması daha iyi sonuçların elde edildiğini göstermektedir. Karşılaştırılan ST1 ile ST2 stratejileri bu metriğe göre analiz edildiğinde Şekil 13 ve 14’te sırasıyla teorik ve pratik senaryolarda görüldüğü üzere ST2 stratejisi olan takım çalışması esaslı istasyon stratejisinin referans noktaya daha yakın çözümler elde ettiği görülmektedir. Bu çıkarım C metriği ile elde edilen yargıları destekler niteliktedir. Buna göre grafiklerde de görüldüğü üzere özellikle problem boyutunun artması ile takım çalışması esaslı istasyon stratejisi; maliyet, çevrim süresi ve iş yükü dengeleme amaçları açısından en iyiye daha yakın sonuçlara ulaşmıştır. Problem boyutunun artması ile takım çalışmasının etkileri daha iyi gözlemlenebilmektedir.



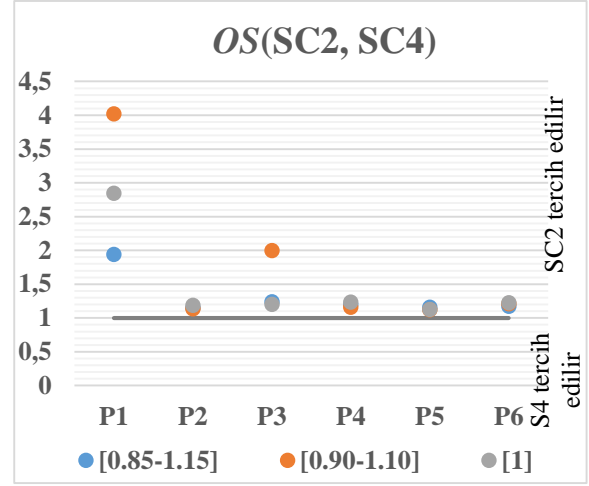
Şekil 13. Teorik senaryolar için elde edilen Pareto optimal sonuçların DI_R metriğine göre karşılaştırmalı sonuçları



Şekil 14. Pratik senaryolar için elde edilen Pareto optimal sonuçların DI_R metriğine göre karşılaştırmalı sonuçları



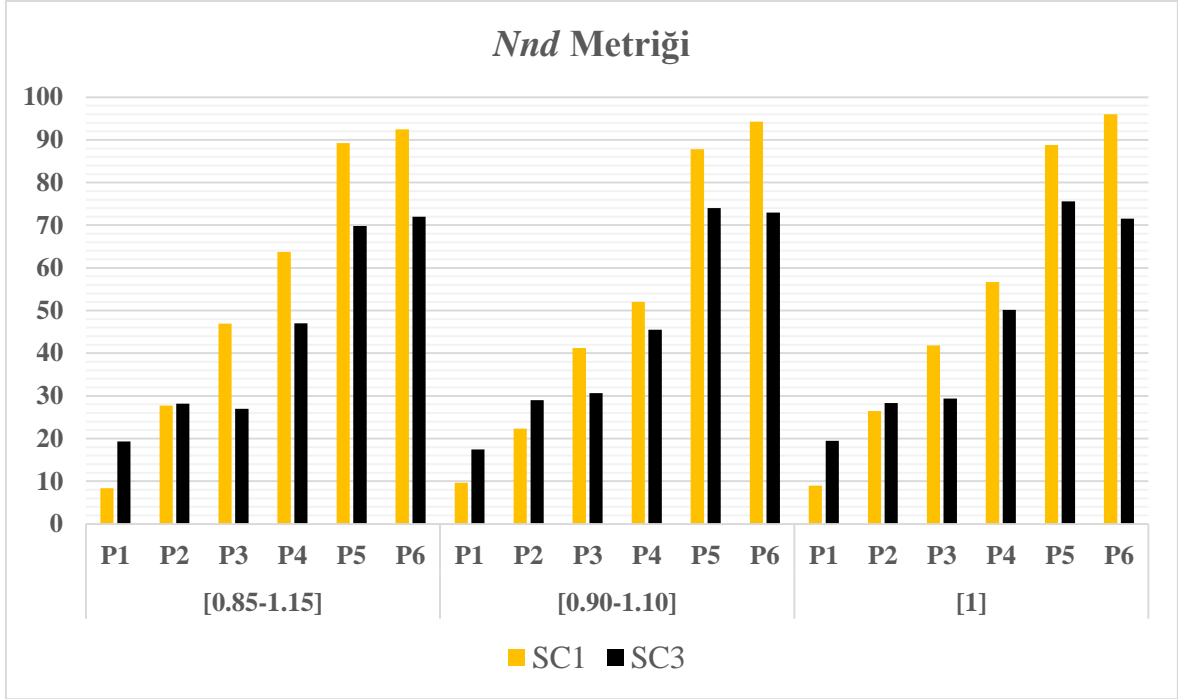
Şekil 15. Teorik senaryolar için elde edilen Pareto optimal sonuçların OS metriğine göre karşılaştırmalı sonuçları



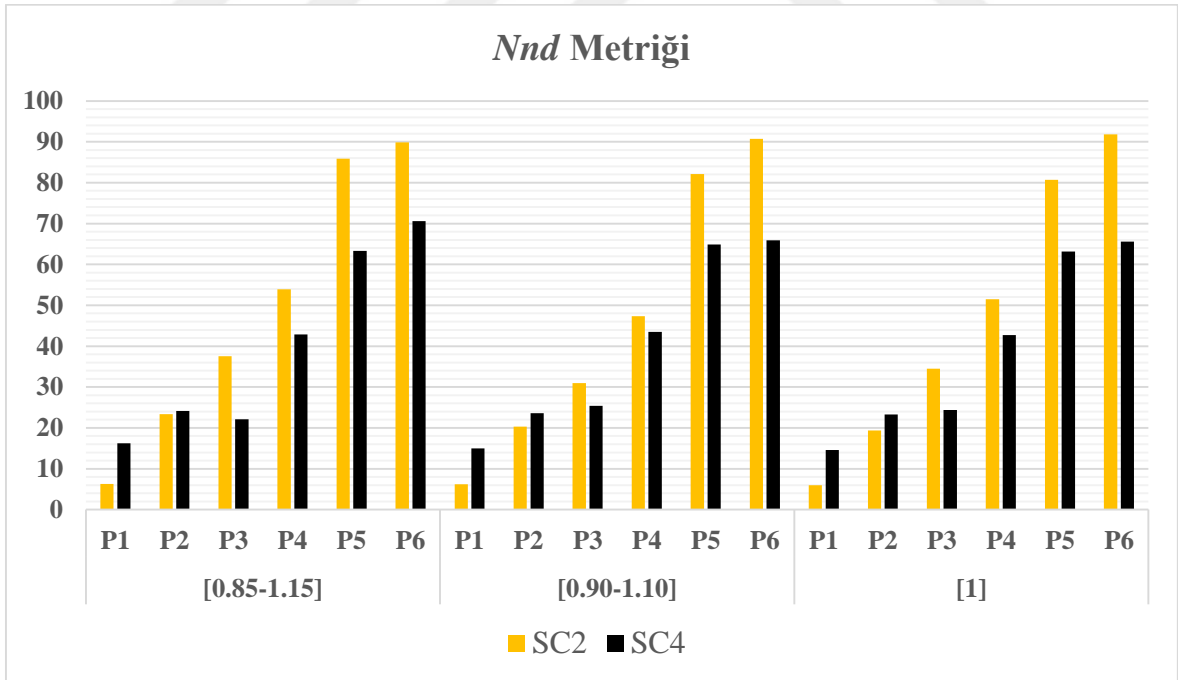
Şekil 16. Pratik senaryolar için elde edilen Pareto optimal sonuçların OS metriğine göre karşılaştırmalı sonuçları

Amaçların yayılımını ele alan OS metriği için elde edilen değerler teorik ve pratik senaryolar için Şekil 15 ve 16'da sırasıyla verilmiştir. Grafikte verilerin daha iyi analiz edilebilmesi için hesaplanan değerlerin amaç sayısına göre kökü (yani küp kökü) alınmıştır. OS metriğinin birden büyük olması, ilgili yaklaşımın daha iyi yayıldığını ifade etmektedir. Grafikler incelendiğinde ST1 stratejisi için her iki senaryonun da daha iyi yayılım gösterdiği görülmektedir. Buradan şu çıkarım yapılmaktadır: ST2 stratejisi genel olarak ST1 stratejisinden daha iyi sonuçlar elde edebilirken yayılım olarak ST1 stratejisi daha etkilidir.

Son olarak iki stratejiyi karşılaştırmak için Pareto optimal sonuçlardaki baskılanamayan çözüm sayısı değerlendirildiğinde, teorik ve pratik senaryolar için elde edilen veriler Şekil 17 ve 18'de gösterilmiştir. Veriler değerlendirildiğinde görev sayısının artması ile birlikte baskılanamayan çözüm sayısının da arttığı gözlenmiştir. Bunun yanı sıra ST1 stratejisinde ST2 stratejisine göre daha çok baskılanamayan çözümün elde edildiği görülmektedir. *Nnd* metriğinin daha fazla olması, ilgili stratejinin seçilmesi için bir öneri oluşturmaz. Ancak diğer metriklerin değerlendirilmesinde bir katkı sağlayabilmektedir. Bu nedenle tek metrik yerine yaklaşımların birden fazla metrik ile değerlendirilmesi doğru yargılara ulaşma aşamasında önemlidir.



Şekil 17. Teorik senaryolar için elde edilen Pareto optimal sonuçların *Nnd* metriğine göre karşılaştırmalı sonuçları

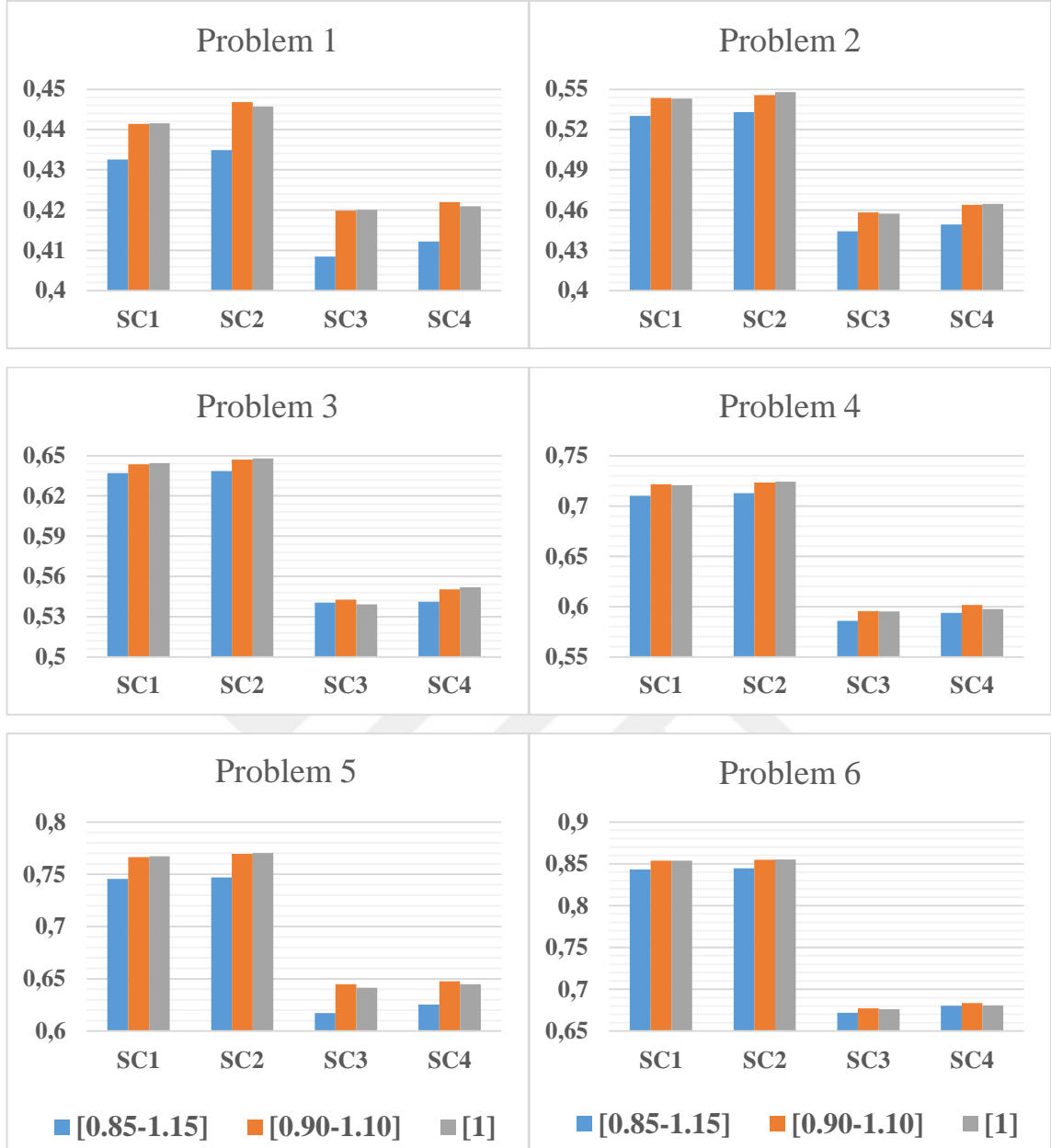


Şekil 18. Teorik senaryolar için elde edilen Pareto optimal sonuçların *Nnd* metriğine göre karşılaştırmalı sonuçları

Sonuç olarak önerilen metrikler birlikte değerlendirildiğinde baskılanamayan çözüm sayısının ST2 stratejisinde daha az olmasına rağmen, elde edilen çözümlerin ST1 stratejisindeki tüm çözümleri genel olarak baskıladığı görülmüştür. Referans alınan (0, 0, 0) noktasına ST2 stratejisi ile daha yakın çözümler elde edilmesi nedeniyle takım çalışması esaslı istasyon stratejisinin klasik istasyon stratesine göre maliyet, çevrim süresi ve iş gücü dengeleme amaçları bakımından daha avantajlı olduğu belirlenmiştir. Buna rağmen ST1 stratejisi ST2 stratejisine göre daha iyi yayılım göstermektedir.

Teoride, tüm çalışanların tüm görevleri yerine getirebildiği varsayılırken gerçek üretim işletmelerinde tüm çalışanların tüm görevleri yapma becerileri olmadığı için bu çalışmada bu durum pratik senaryo olarak ele alınmıştır. Tüm çalışanların tüm görevleri yapabildiği teorik senaryo ile pratik senaryo yukarıda verilen grafikler ve veriler kapsamında değerlendirildiğinde iki farklı senaryo arasında anlamlı bir farkın olmadığı görülmektedir. Bu nedenle çalışma kapsamında teorik senaryo ile elde edilen tüm yargıların pratik senaryo için de geçerli olduğu sonucuna varılmaktadır.

Çalışma kapsamında analiz edilen bir diğer yaklaşım ise çalışan yeterlilik seviyelerindeki heterojenliklerdir. Yapılan çalışmada çalışan yeterlilik seviyesi aralıkları [0.85-1.15], [0.90-1.10] ve [1] olmak üzere sırasıyla heterojen, az heterojen ve homojen çalışan tipleri ele alınmıştır. Çalışan yeterlilik seviyelerinin karşılaştırılması aşamasında referans noktaya uzaklıkları ifade eden DI_R metriği ile elde edilen grafikler Şekil 19'da görülmektedir. Grafiklerden de anlaşılacağı üzere çalışan yeterlilik seviyelerindeki farklılıklar demontaj hattının performansı üzerinde çok fazla bir etkiye neden olmasa bile, hat dengelemede çalışan yeterlilik seviyelerinin göz önüne alınması küçük de olsa farklılıklara neden olmaktadır. Buna göre heterojenlik seviyesinin fazla olduğu hatlarda görevlerin yetkin kişilere atanması, çalışanların yeterlilik seviyelerinin ortalamaları ile çalışan ücretleri arasında ters orantı olmasına rağmen, sistemin performansını az da olsa iyileştirmektedir. Bu nedenle sistemlerde çalışanların farklı alanlarda uzmanlaşmasının sisteme katkı sağlayacağı düşünülmektedir.



Şekil 19. Çalışan beceri ve yeterlilik seviyeleri için karşılaştırmalı sonuçlar

Varyans analizi ön koşulları sağlamasa dahi gürbüz bir yaklaşımdır ve bu nedenle kullanışlıdır (Yılmaz ve Durmuşoğlu, 2018). Çalışan yeterlilik seviyelerinin istatistiksel açıdan karşılaştırmasını içeren varyans analizi tablosu Tablo 10'da verilmektedir. Elde edilen varyans analizi tablosuna göre önem derecelerinin (sig.) 0.05 değerinden büyük çıkması ile tüm senaryolar için çalışanların yeterlilik seviyeleri arasında anlamlı bir fark olmadığı görülmektedir.

Tablo 10. Çalışan yeterlilik seviyelerinin karşılaştırılması için varyans analizi tablosu

	Kareler Toplamı	Serbestlik Derecesi	Kareler Ortalaması	F Değeri	Anlamlılık
Senaryo 1					
Gruplar Arası	.001	2	.000	.013	.987
Grup İçi	.340	15	.023		
Genel Toplam	.341	17			
Senaryo 2					
Gruplar Arası	.001	2	.000	.015	.985
Grup İçi	.337	15	.022		
Genel Toplam	.337	17			
Senaryo 3					
Gruplar Arası	.000	2	.000	.023	.977
Grup İçi	.156	15	.010		
Genel Toplam	.156	17			
Senaryo 4					
Gruplar Arası	.000	2	.000	.021	.980
Grup İçi	.158	15	.011		
Genel Toplam	.159	17			

Deneysel Hesaplamalar bölümünde yapılan hesaplamalı analiz ile elde edilen yargılar aşağıdaki şekilde özetlenmektedir.

- Geliştirilen NSGA-II yönteminde tek noktalı çaprazlama operatörü ile daha etkili sonuçlar elde edilmektedir.
- Takım çalışması esaslı istasyon stratejisi maliyet, çevrim süresi ve iş gücü dengeleme amaçları dikkate alındığında sistemin performansını önemli derecede arttırmaktadır.
- Takım çalışmasının avantajları, paralel görev yapımı elverişli olduğu sürece görev sayısı arttıkça artar.
- Çalışan beceri seviyelerinin dikkate alınması ile elde edilen sonuçlar, beceri seviyelerinin dikkate alınmaması durumunda elde edilen sonuçları destekler.
- Çalışan yeterlilik seviyelerinin heterojenliğinin artması hattın etkili dengelenmesi durumunda sisteme az da olsa katkı sağlar.
- İki farklı yaklaşımın karşılaştırılmasında farklı metriklerin beraber değerlendirilmesi sonuçların doğru yorumlanması için önemlidir.

5. SONUÇLAR VE ÖNERİLER

Bu tez kapsamında minimum maliyet, minimum çevrim süresi ve minimum iş gücü dengesizliği amaçları dikkate alınarak çok amaçlı demontaj hattı dengeleme problemlerine yer verilmiştir. Ele alınan problemler için optimizasyon modeli geliştirilmiştir. Takım çalışması ve çalışanların heterojenliği göz önüne alınarak üretilen beş küçük boyutlu problem AUGMECON2, altı büyük boyutlu problem NSGA-II çözüm yaklaşımları kullanılarak analiz edilmiştir. Elde edilen Pareto optimal çözümler baskınlık, en iyi çözüme yakınlık, yayılım ve çeşitliliklerine göre karşılaştırılmıştır.

Problemler, klasik istasyon-takım çalışması esaslı istasyon stratejileri ve çalışan beceri seviyelerine göre teorik-pratik senaryolar altında analiz edilmiştir. Buna ek olarak üç farklı aralıkta çalışan yeterlilik seviyesi ele alınarak, demontaj hattının performansı üzerine etkisi incelenmiştir. Çözüm yaklaşımı olarak geliştirilen NSGA-II algoritmasında iki farklı geçiş operatörü kullanılarak elde edilen tüm sonuçlar karşılaştırmalı olarak analiz edilmiştir.

İncelenen sonuçlardan yapılan çıkarımlar doğrultusunda takım çalışması esaslı istasyon stratejisinin maliyet, çevrim süresi ve iş gücü dengeleme amaçları dikkate alındığında sistemin performansını arttırdığı gözlenmiştir. Bu kapsamda takım çalışmasının sistemin performansı üzerindeki etkisi, bir istasyona atanan çalışan sayısına ve ürünün özelliklerine göre değişiklik göstermektedir. Pratik senaryolar ile elde edilen sonuçlar, teorik senaryo ile elde edilen çıkarımları desteklemektedir. Bu durum yapılacak çalışmalarda çalışan beceri seviyelerinin göz ardı edilmesi ile ulaşılan sonuçların, çalışan becerilerinin dikkate alınması durumunda da geçerli olacağını göstermektedir. Çalışan yeterlilik seviyelerinin analizinde ise çalışan heterojenliklerinin fazla olması istatistiksel olarak fark oluşturmaya da grafiklerde az da olsa daha iyi sonuçlar göstermiştir. Bu nedenle sistemin performansını arttırmak için çalışanların eğitilmesi önem arz etmektedir.

Çalışmanın yenilikleri aşağıdaki şekilde özetlenmektedir.

- Maliyet, çevrim süresi ve iş gücü dengeleme amaçları kapsamında takım çalışması esaslı istasyonların demontaj sisteminin performansı üzerindeki etkisini inceleyen ÇADHDP'ni ele almaktadır.

- DHDP için görev tabanlı teknolojik öncelik diyagramları ile VEYA ardıl ilişki türünü dikkate alan bir çalışmadır.
- Çalışan yeterlilik ve beceri seviyelerinin demontaj sisteminin performansına etkisini analiz etmektedir.
- Küçük boyutlu ÇADHDP'nin çözümü için AUGMECON2, büyük boyutlu ÇADHDP'nin çözümü için iki farklı çaprazlama operatörü ile NSGA-II yöntemlerini kullanmaktadır.

Gelecek çalışmalar için önerilere aşağıda yer verilmektedir:

- Ele alınan problemin çözümü için farklı meta-sezgisel yöntemler kullanılabilir.
- Önerilen model için talep ve görev sürelerindeki belirsizlikler ele alınabilir.
- Demonte edilmek üzere hata alınan ürünlerin farklı olması ya da farklı hasarlar almış olması göz önüne alınabilir.
- Görevler arasındaki geçiş sürelerinin farklı olmasına dikkat edilebilir.
- Sistem, ürünlerin tedarigi açısından ele alınabilir.

6. KAYNAKLAR

- Agrawal, S., ve Tiwari, M., 2008. A Collaborative Ant Colony Algorithm To Stochastic Mixed-Model U-Shaped Disassembly Line Balancing And Sequencing Problem, International Journal of Production Research, 46, 6, 1405-1429, DOI:10.1080/00207540600943985.
- Altekin, F. T., 2017. A Comparison Of Piecewise Linear Programming Formulations For Stochastic Disassembly Line Balancing, International Journal of Production Research, DOI:10.1080/00207543.2017.1351639.
- Altekin, F. T., Kandiller, L., ve Ozdemirel, N. E., 2008. Profit-Oriented Disassembly-Line Balancing, International Journal of Production Research, 46, 10 , 2675-2693, DOI:10.1080/00207540601137207.
- Aslan, D., 2019. Takım Çalışması Esaslı Demontaj Hattı İçin Optimizasyon Yaklaşımı: Beyaz Eşya Endüstrisinde Bir Uygulama, Doktora Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.
- Avikal, S., ve Mishra, P. K., 2012. A New U-Shaped Heuristic for Disassembly Line Balancing Problems, Pratibha: International Journal Of Science, Spirituality, Business And Technology (IJSSBT), 1, 1.
- Avikal, S., Jain, R., ve Mishra, P. K., 2013. A Heuristic for U-Shaped Disassembly Line Balancing Problems, MIT International Journal of Mechanical Engineering, 3, 1, 51-56.
- Avikal, S., Mishra, P. K., ve Jain, R., 2014. A Fuzzy AHP and PROMETHEE Method-Based Heuristic For Disassembly Line Balancing Problems, International Journal of Production Research, 52, 5, 1306-1317, DOI:10.1080/00207543.2013.831999.
- Aydemir-Karadag, A., ve Turkbey, O., 2013. Multi-Objective Optimization Of Stochastic Disassembly Line Balancing With Station Paralleling, Computers & Industrial Engineering, 65, 413-425, DOI:10.1016/j.cie.2013.03.014.
- Bentaha, M. L., Battaia, O., ve Dolgui, A., 2014. Disassembly Line Balancing and Sequencing under Uncertainty, 21st CIRP Conference on Life Cycle Engineering (Procedia CIRP), 15, s. 239-244, DOI:10.1016/j.procir.2014.06.016.
- Bentaha, M. L., Voisin, A., ve Marange, P., 2020. A Decision Tool For Disassembly Process Planning Under End-Of-Life Product Quality, International Journal of Production Economics, 219, 386-401, DOI:10.1016/j.ijpe.2019.07.015.

- Cevikcan, E., Aslan, D., ve Yeni, F. B., 2019. Disassembly Line Design With Multi-Manned Workstations: A Novel Heuristic Optimisation Approach, International Journal of Production Research, DOI:10.1080/00207543.2019.1587190.
- Chen, Y. Y., 2017. A Hybrid Algorithm For Allocating Tasks, Operators, And Workstations in Multi-Manned Assembly Lines, Journal of Manufacturing Systems, 42, 196-209.
- Chen, Y. Y., Cheng, C.Y., ve Li, J. Y., 2018. Resource-Constrained Assembly Line Balancing Problems With Multi-Manned Workstations, Journal of Manufacturing Systems, 48, 107-119.
- Deb, K., Pratap, A., Agarwal, S., ve Meyarivan, T., 2002. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II, IEEE Transactions On Evolutionary Computation, 6, 2, 182-197.
- Deniz, N., ve Ozcelik, F., 2019. An Extended Review On Disassembly Line Balancing With Bibliometric & Social Network And Future Study Realization Analysis, Journal of Cleaner Production, 225, 697-715, DOI:10.1016/j.jclepro.2019.03.188.
- Ding, L.-P., Feng, Y.-X., Tan, J.-R., ve Gao, Y.-C., 2010. A New Multi-Objective Ant Colony Algorithm For Solving The Disassembly Line Balancing Problem, Int J Adv Manuf Technol, 48, 761-771, DOI:10.1007/s00170-009-2303-5.
- Edis, E. B., Ilgin, M. A., ve Edis, R. S., 2019. Disassembly Line Balancing With Sequencing Decisions: A Mixed Integer Linear Programming Model And Extensions, Journal of Cleaner Production, 238, DOI:10.1016/j.jclepro.2019.117826.
- Fang, Y., Liu, Q., Li, M., Laili, Y., ve Pham, D. T., 2018. Evolutionary Many-Objective Optimization For Mixed-Model Disassembly Line Balancing With Multi-Robotic Workstations, European Journal of Operational Research, DOI:10.1016/j.ejor.2018.12.035.
- Gungor, A., ve Gupta, S. M., 2001. A Solution Approach To The Disassembly Line Balancing Problem In The Presence Of Task Failures, International Journal of Production Research, 7, 39, 1427-1467, DOI:10.1080/00207540110052157.
- Gupta, S. M., ve Taleb, K. N., 1994. Scheduling Disassembly, International Journal of Production Research, 32, 8, 1857-1866, DOI:10.1080/00207549408957046.
- Gümüşkaya, V., 2013. Değişken İş Zamanlı Kâr Amaçlı Hibrit Hatlarda Demontaj Hat Dengelenmesi, Yüksek Lisans Tezi, Orta Doğu Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Ankara.
- Hezer, S., ve Kara, Y., 2014. A Network-Based Shortest Route Model For Parallel Disassembly Line Balancing Problem, International Journal of Production Research, DOI:10.1080/00207543.2014.965348.

- Kalayci, C. B., ve Gupta, S. M., 2013. Ant Colony Optimization For Sequence-Dependent Disassembly Line Balancing Problem, Journal of Manufacturing Technology Management, 24, 3, 413-427, DOI:10.1108/17410381311318909.
- Kalayci, C. B., Hancilar, A., Gungor, A., ve Gupta, S. M., 2014. Multi-Objective Fuzzy Disassembly Line Balancing Using A Hybrid Discrete Artificial Bee Colony Algorithm, Journal of Manufacturing Systems, DOI:10.1016/j.jmsy.2014.11.015.
- Kalayci, C. B., Polat, O., ve Gupta, S. M., 2014. A Hybrid Genetic Algorithm For Sequence-Dependent Disassembly Line Balancing Problem, Annals of Operations Research, 242, 321-354, DOI:10.1007/s10479-014-1641-3.
- Kalaycilar, E. G., Azizoğlu, M., ve Yeralan, S., 2015. A Disassembly Line Balancing Problem With Fixed Number Of Workstations, European Journal of Operational Research, 1-13, DOI:10.1016/j.ejor.2015.09.004.
- Kannan, D., Garg, K., Jha, P., ve Diabat, A., 2016. Integrating Disassembly Line Balancing In The Planning Of A Reverse Logistics Network From The Perspective Of A Third Party Provider, Annals of Operations Research, 253, 353-376, DOI:10.1007/s10479-016-2272-7.
- Koc, A., Sabuncuoglu, I., ve Erel, E., 2009. Two Exact Formulations For Disassembly Line Balancing Problems With Task Precedence Diagram Construction Using An And/Or Graph, IIE Transactions, 41, 10, 866-881, DOI:10.1080/07408170802510390.
- Lambert, A. J., 2007. A Heuristic For Assembly And Disassembly Line Balancing, IFAC Proceedings Volumes, 40, 2, 69-74.
- Li, Z., Çil, Z. A., Mete, S., ve Kucukkoc, I., 2019. A Fast Branch, Bound And Remember Algorithm For Disassembly Line Balancing Problem, International Journal of Production Research, DOI:10.1080/00207543.2019.1630774.
- Lian, J., Liu, C., Li, W., ve Yin, Y., 2018. A Multi-Skilled Worker Assignment Problem In Seru Production Systems Considering The Worker Heterogeneity, Computers & Industrial Engineering, 118, 366-382, DOI:10.1016/j.cie.2018.02.035.
- Liu, J., Zhou, Z., Pham, D. T., Xu, W., Ji, C., ve Liu, Q., 2020. Collaborative Optimization Of Robotic Disassembly Sequence Planning And Robotic Disassembly Line Balancing Problem Using Improved Discrete Bees Algorithm In Remanufacturing, Robotics and Computer Integrated Manufacturing, 61, 101829, DOI:10.1016/j.rcim.2019.101829.
- Liu, M., Liu, X., Chu, F., Zheng, F., ve Chu, C., 2019. Robust Disassembly Line Balancing With Ambiguous Task Processing Times, International Journal of Production Research, DOI:10.1080/00207543.2019.1659520.
- Lopes, T. C., Pastre, G. V., Michels, A. S., ve Magatao, L., 2019. Flexible multi-manned assembly line balancing problem: Model, heuristic procedure, and lower bounds for line length minimization, Omega, 9,29.

- Lu, Q., Ren, Y., Jin, H., Meng, L., Li, L., Zhang, C., ve Sutherland, J. W., 2020. A Hybrid Metaheuristic Algorithm For A Profit-Oriented And Energy-Efficient Disassembly Sequencing Problem, Robotics and Computer Integrated Manufacturing, 61, DOI:10.1016/j.rcim.2019.101828.
- Mavrotas, G., 2009. Effective Implementation Of The E-Constraint Method In Multi-Objective Mathematical Programming Problems, Applied Mathematics and Computation, 213, 455-465, DOI:10.1016/j.amc.2009.03.037.
- Mavrotas, G., ve Florios, K., 2013. An Improved Version Of The Augmented E-Constraint Method (AUGMECON2) For Finding The Exact Pareto Set In Multi-Objective Integer Programming Problems, Applied Mathematics and Computation, 219, 9652-9669, DOI:10.1016/j.amc.2013.03.002.
- McGovern, S. M., ve Gupta, S. M., 2007. A Balancing Method And Genetic Algorithm For Disassembly Line Balancing, European Journal of Operational Research, 179, 692-708, DOI:10.1016/j.ejor.2005.03.055.
- McGovern, S., ve Gupta, S., 2007. Combinatorial Optimization Analysis Of The Unary NP-Complete Disassembly Line Balancing Problem, International Journal of Production Research, 45, 18-19, 4485-4511, DOI:10.1080/00207540701476281.
- Mete, S., Çil, Z. A., Ağpak, K., Özceylan, E., ve Dolgui, A., 2016b. A Solution Approach Based On Beam Search Algorithm For Disassembly Line Balancing Problem, Journal of Manufacturing Systems, 41, 188-200, DOI:10.1016/j.jmsy.2016.09.002.
- Mete, S., Çil, Z. A., Özceylan, E., ve Ağpak, K., 2016, Resource Constrained Disassembly Line Balancing Problem, IFAC-PapersOnLine, Elsevier Ltd, 921-925.
- Oksuz, M. K., Buyukozkan, K., ve Satoglu, S. I., 2017. U-Shaped Assembly Line Worker Assignment And Balancing Problem: A Mathematical Model And Two Meta-Heuristics, Computers & Industrial Engineering, 112, 246-263, DOI:10.1016/j.cie.2017.08.030.
- Özceylan, E., ve Paksoy, T., 2013. Reverse Supply Chain Optimisation With Disassembly Line Balancing, International Journal of Production Research, 51, 20, 5985-6001, DOI:10.1080/00207543.2013.784405.
- Özceylan, E., Kalayci, C. B., Güngör, A., ve Gupta, S. M., 2018. Disassembly Line Balancing Problem: A Review Of The State Of The Art And Future Directions, International Journal of Production Research, DOI:10.1080/00207543.2018.1428775.
- Özceylan, E., Paksoy, T., ve Bektaş, T., 2014. Modeling And Optimizing The Integrated Problem Of Closed-Loop Supply Chain Network Design And Disassembly Line Balancing, Transportation Research Part E, 61, 142-164, DOI:10.1016/j.tre.2013.11.001.

- Paksoy, T., Güngör, A., Özceylan, E., ve Hancılar, A., 2013. Mixed Model Disassembly Line Balancing Problem With Fuzzy Goals, International Journal of Production Research, 51, 20, 6082-6096, DOI:10.1080/00207543.2013.795251.
- Ren, Y., Zhang, C., Zhao, F., Tian, G., Lin, W., Meng, L., ve Li, H., 2018. Disassembly Line Balancing Problem Using Interdependent Weights-Based Multi-Criteria Decision Making And 2-Optimal Algorithm, Journal of Cleaner Production, 174, 1475-1486, DOI:10.1016/j.jclepro.2017.10.308.
- Ren, Y., Zhang, C., Zhao, F., Xiao, H., ve Tian, G., 2018a. An Asynchronous Parallel Disassembly Planning Based On Genetic Algorithm, European Journal of Operational Research, 1-14, DOI:10.1016/j.ejor.2018.01.055.
- Riggs, R. J., Battaia, O., ve Hu, S. J., 2015. Disassembly Line Balancing Under High Variety Of End Of Life States Using A Joint Precedence Graph Approach, Journal of Manufacturing Systems, DOI:10.1016/j.jmsy.2014.11.002.
- Şahin, M., ve Kellegöz, T., 2019. A new mixed-integer linear programming formulation and particle swarm optimization based hybrid heuristic for the problem of resource investment and balancing of the assembly line with multi-manned workstations, Computers & Industrial Engineering, 133, 107-120.
- Wang, K., Li, X., ve Gao, L., 2019. A Multi-Objective Discrete Flower Pollination Algorithm For Stochastic Two-Sided Partial Disassembly Line Balancing Problem, Computers & Industrial Engineering, 130, 634-649, DOI:10.1016/j.cie.2019.03.017.
- Wang, S., ve Liu, M., 2015. Multi-Objective Optimization Of Parallel Machine Scheduling Integrated With Multi-Resources Preventive Maintenance Planning, Journal of Manufacturing Systems, 37, 182-192.
- Wu, X., ve Che, A., 2019. Energy-Efficient No-Wait Permutation Flow Shop Scheduling By Adaptive Multi-Objective Variable Neighborhood Search, Omega.
- Yılmaz, Ö.F. ve Durmuşoğlu M.B., 2018. A Performance Comparison And Evaluation Of Metaheuristics For A Batch Scheduling Problem In A Multi-Hybrid Cell Manufacturing System With Skilled Workforce Assignment, Journal Of Industrial And Management Optimization, 14, 1219-1249.
- Yılmaz, Ö. F., 2019. Operational Strategies For Seru Production System: A Bi-Objective Optimisation Model And Solution Methods, International Journal of Production Research, DOI:10.1080/00207543.2019.1669841.
- Yılmaz, Ö. F., 2020. MODALBP-Problem data and computational results, GitHub-omerfarukktu repository:
<https://github.com/omerfarukktu/MODALBP-Problem-data-and-results>

Zhang, Z., Wang, K., Zhu, L., ve Wang, Y., 2017. A Pareto Improved Artificial Fish Swarm Algorithm For Solving A Multi-Objective Fuzzy Disassembly Line Balancing Problem, Expert Systems With Applications, 86, 165-176, DOI:10.1016/j.eswa.2017.05.053.



7. EKLER

Ek 1. Sınır Değerleri Tablosu Kodlar (AUGMECON2)

```
model Ornek1;
sets
i görevler /1*7/
k parça /1*4/
w çalışan /1*4/
b çevrim süresi katsayısı /1*10/
iterasyon/1*3/
CS/1*2/
;

alias(i,s,j,h);

Scalar
L büyük bir sayı/1000000/
C çevrim süresi katsayısı/10/
Time müsait zaman /10000/
wmax hücreye atanabilecek max çalışan/1/
FC sabit maliyet/1000/
SC hurda maliyeti/200/
d1 amaç1/10000000/
d2 amaç2/10000000/
d3 amaç3/10000000/
takip/1/
starttime
elapsed
;

sets
PAND(i,s) görev i nin ve öncülleri
*Problem1
*/2.1, 3.1/
*Problem2
*/2.1, 3.1, 5.3/
*Problem3
*/2.1, 3.1, 5.4, 6.4/
*Problem4
*/2.1, 3.1, 4.1, 6.4, 7.4/
*Problem5
*/2.1, 3.1, 4.1, 6.2, 6.3, 6.5, 7.3, 7.4/
```

Ek 1'in devamı

POR_i(i)

*Problem1, Problem2, Problem3

*/4/

*Problem4

*/5/

*Problem5

*/5, 8/

POR(i,s) görev i nin veya öncülleri

*Problem1, Problem2, Problem3

*/4.2, 4.3/

*Problem4

*/5.2, 5.3/

*Problem5

*/5.2, 5.3, 8.6, 8.7/

SOR_i(i)

*Problem3, Problem4

*/4/

SOR(i,s) görev i nin veya öncülleri

*Problem3

*/4.5, 4.6/

*Problem4

*/4.5, 4.6/

;

Parameters

t(i) görev süreleri

*Problem 1

*\$call GDXXRW demontaj.xlsx par=t rng=t!a2:b5 rdim=1 cdim=0

*Problem 2

*\$call GDXXRW demontaj.xlsx par=t rng=t!d2:e6 rdim=1 cdim=0

*Problem 3

*\$call GDXXRW demontaj.xlsx par=t rng=t!g2:h7 rdim=1 cdim=0

*Problem 4

*\$call GDXXRW demontaj.xlsx par=t rng=t!j2:k8 rdim=1 cdim=0

*Problem 5

*\$call GDXXRW demontaj.xlsx par=t rng=t!m2:n9 rdim=1 cdim=0

\$GDXIN demontaj.gdx

\$LOAD t

\$GDXIN

f(i,w) işçi becerileri

*Problem 1

*\$call GDXXRW demontaj.xlsx par=f rng=f!a1:e5 rdim=1 cdim=1

*Problem 2

Ek 1'in devamı

```
*$call GDXXRW demontaj.xlsx par=f rng=f!a7:e12 rdim=1 cdim=1
*Problem 3
*$call GDXXRW demontaj.xlsx par=f rng=f!a14:e20 rdim=1 cdim=1
*Problem 4
*$call GDXXRW demontaj.xlsx par=f rng=f!i1:m8 rdim=1 cdim=1
*Problem 5
*$call GDXXRW demontaj.xlsx par=f rng=f!i10:m18 rdim=1 cdim=1
$GDXIN demontaj.gdx
$LOAD f
$GDXIN
```

d(k) talep

```
*Problem 1
*$call GDXXRW demontaj.xlsx par=d rng=d!a2:b5 rdim=1 cdim=0
*Problem 2
*$call GDXXRW demontaj.xlsx par=d rng=d!d2:e5 rdim=1 cdim=0
*Problem 3
*$call GDXXRW demontaj.xlsx par=d rng=d!g2:h5 rdim=1 cdim=0
*Problem 4
*$call GDXXRW demontaj.xlsx par=d rng=d!j2:k5 rdim=1 cdim=0
*Problem 5
*$call GDXXRW demontaj.xlsx par=d rng=d!m2:n5 rdim=1 cdim=0
$GDXIN demontaj.gdx
$LOAD d
$GDXIN
```

m(i,k) görev*parça ilişkisi

```
*Problem 1
*$call GDXXRW demontaj.xlsx par=m rng=m!a1:e5 rdim=1 cdim=1
*Problem 2
*$call GDXXRW demontaj.xlsx par=m rng=m!a7:e12 rdim=1 cdim=1
*Problem 3
*$call GDXXRW demontaj.xlsx par=m rng=m!a14:e20 rdim=1 cdim=1
*Problem 4
*$call GDXXRW demontaj.xlsx par=m rng=m!i1:m8 rdim=1 cdim=1
*Problem 5
*$call GDXXRW demontaj.xlsx par=m rng=m!i10:m18 rdim=1 cdim=1
$GDXIN demontaj.gdx
$LOAD m
$GDXIN
```

WC(w) talep

```
*Problem 1
*$call GDXXRW demontaj.xlsx par=wc rng=wc!a2:b5 rdim=1 cdim=0
*Problem 2
*$call GDXXRW demontaj.xlsx par=wc rng=wc!d2:e5 rdim=1 cdim=0
*Problem 3
```

Ek 1'in devamı

```
*$call GDXXRW demontaj.xlsx par=wc rng=wc!g2:h5 rdim=1 cdim=0
*Problem 4
*$call GDXXRW demontaj.xlsx par=wc rng=wc!j2:k5 rdim=1 cdim=0
*Problem 5
*$call GDXXRW demontaj.xlsx par=wc rng=wc!m2:n5 rdim=1 cdim=0
$GDXIN demontaj.gdx
$LOAD wc
$GDXIN
```

```
goal1(CS,iterasyon)
goal2(CS,iterasyon)
goal3(CS,iterasyon)
istasyon(CS,iterasyon)
urun(CS,iterasyon)
calisan(CS,iterasyon)
```

Binary Variables

```
x(i,j,w) i görevi j istasyonuna w çalışanına atandıysa
y(i,s)
st(j) j. isyasyon açıldıysa
ws(j,w) w. çalışan j isyasyonundaysa
epor(i,s)
Pr(b)
;
```

Integer Variables

```
p(i)
Product
toplamst
toplamcl
;
*p.up(i)=10000;
*Product.up=10000;
```

Positive Variables

```
CT
strt(i,j)
MKC
MNC
G32
;
```

Variables

```
G1 amaç1
G2 amaç2
G3 amaç3
;
```


Ek 1'in devamı

Equations

amac1 maliyetin minimize edilmesi

amac2 çevrim süresi

amac3 çalışanlar arasında dengesizlik

amac32

Kisit4 her görevin 1 ist. 1 çalışan tarafından yerine getirilebilmesi

Kisit5 görev atamaları ile çalışan atamaları arasında ilişki

Kisit6 çalışan atamaları ile istasyon açma arasında ilişki

Kisit7 bir çalışan bir istasyona atanabilir

Kisit8 görev atama ile görev yapma miktarı arasında ilişki

Kisit9 sökülen parça sayısının talebi karşılaması

Kisit10 Demonte edilen ürün sayısının bulunması

Kisit11 Çevrim süresi indisinin hesaplanması

Kisit12 Çevrim süresinin hesaplanması

Kisit13 VE öncülü

Kisit14 VEYA öncülü

Kisit15 VEYA öncülünün bulunması

Kisit16 VEYA ardılı

Kisit17 bir görevin başlangıç zamanı ve süresinin toplamının çevrim süresini

geçmemesi

Kisit18 VE öncülü işlem süresi

Kisit19 VEYA öncülü işlem süresi

Kisit20 çalışanların yaptığı görevlerin zamanlarının çakışmasının engellenmesi

Kisit21 çalışanların yaptığı görevlerin zamanlarının çakışmasının engellenmesi

Kisit22 Maksimum çalışma zamanı

Kisit23 Minimum çalışma zamanı

Kisit24 Toplam istasyon

Kisit25 Toplam çalışan

Kisit26

Kisit27

Kisit28

;

amac1..

$G1=e=FC*\sum(j,st(j))+SC*Product+\sum(w,WC(w)*\sum(j,ws(j,w)));$

amac2.. $G2=e=CT;$

amac3.. $G32=e=MKC-MNC;$

amac32.. $G3=e=G32;$

Kisit4(i).. $\sum((j,w),x(i,j,w))=l=1;$

Kisit5(j,w).. $\sum(i,x(i,j,w))=l=card(i)*ws(j,w);$

Kisit6(j).. $\sum(w,ws(j,w))=l=wmax*st(j);$

Kisit7(w).. $\sum(j,ws(j,w))=l=1;$

Kisit8(i).. $p(i)=l=\sum((j,w),x(i,j,w))*L;$

Kisit9(k).. $d(k)=l=\sum(i,p(i)*m(i,k));$

Kisit10(i).. $Product=g=p(i);$

Kisit11.. $C*\sum(b,Pr(b))=g=Product;$

Kisit12(b).. $CT*C*ord(b)=l=Time+L*(1-Pr(b));$

Ek 1'in devamı

```

Kisit13(PAND(i,s),j)..          sum(w,x(i,j,w))=l=sum(h$(ord(h) le
ord(j)),sum(w,x(s,h,w)));
Kisit14(POR(i,s),j)..          sum(w,x(i,j,w))=l=sum(h$(ord(h) le
ord(j)),sum(w,x(s,h,w)))+L*(1-epor(i,s));
Kisit15(PORi(i)..             sum((POR(i,s)),epor(i,s))=g=1;
Kisit16(SORi(i)..
sum((j,w),x(i,j,w))=g=sum(SOR(i,s),sum((j,w),x(s,j,w)));
Kisit17(i,j)..                 strt(i,j)+sum(w,t(i)*f(i,w)*x(i,j,w))=l=CT;
Kisit18(PAND(i,s),j)..        strt(i,j)-strt(s,j)+L*(1-sum(w,x(s,j,w)))+L*(1-
sum(w,x(i,j,w)))=g=sum(w,x(s,j,w)*t(s)*f(s,w));
Kisit19(POR(i,s),j)..        strt(i,j)-strt(s,j)+L*(1-sum(w,x(s,j,w)))+L*(1-
sum(w,x(i,j,w)))=g=sum(w,x(s,j,w)*t(s)*f(s,w))-L*(1-epor(i,s));
Kisit20(i,s,j,w)$ (ord(s) gt ord(i))..   strt(i,j)-strt(s,j)+L*(1-x(s,j,w))+L*(1-
x(i,j,w))+L*y(i,s)=g=x(s,j,w)*t(s)*f(s,w);
Kisit21(i,s,j,w)$ (ord(s) gt ord(i))..   strt(s,j)-strt(i,j)+L*(1-x(s,j,w))+L*(1-
x(i,j,w))+L*(1-y(i,s))=g=x(i,j,w)*t(i)*f(i,w);
Kisit22(w)..                   MKC=g=sum((i,j),x(i,j,w)*t(i)*f(i,w));
Kisit23(w)..                   MNC=l=sum((i,j),x(i,j,w)*t(i)*f(i,w))+L*(1-
sum(j,ws(j,w)));
Kisit24..                       toplamst=e=sum(j,st(j));
Kisit25..                       toplamcl=e=sum((j,w),ws(j,w));
Kisit26..                       G1=l=d1;
Kisit27..                       G2=l=d2;
Kisit28..                       G3=l=d3;

```

```

model Ornek1 /all/;
*OPTION reslim =1000000;
*OPTION ITERLIM = 90000000;
*option intVarUp = 0;
*option optcr=0;
*option threads = -1;
*starttime=jnow;
*solve Ornek1 using mip minimizing G1;
*elapsed=(jnow-starttime)*24*3600;
*display G1.l, G2.l, G3.l, MKC.l, MNC.l, toplamst.l, toplamcl.l, Product.l, elapsed,
x.l, strt.l, ws.l, epor.l, st.l;

```

file prb4/C:\Users\ASUS\Desktop\prb4.txt/;

put prb4/;

```

loop (CS,
loop (iterasyon,
if (takip = 1,
starttime=jnow;
solve Ornek1 using mip minimizing G1;
d1=G1.l;
solve Ornek1 using mip minimizing G2;
d2=G2.l;

```

Ek 1'in devamı

```

        solve Ornek1 using mip minimizing G3;
        elapsed=(jnow-starttime)*24*3600;
    );
    if (takip = 2,
        starttime=jnow;
        solve Ornek1 using mip minimizing G2;
        d2=G2.l;
        solve Ornek1 using mip minimizing G1;
        d1=G1.l;
        solve Ornek1 using mip minimizing G3;
        elapsed=(jnow-starttime)*24*3600;
    );
    if (takip = 3,
        starttime=jnow;
        solve Ornek1 using mip minimizing G3;
        d3=G3.l;
        solve Ornek1 using mip minimizing G1;
        d1=G1.l;
        solve Ornek1 using mip minimizing G2;
        elapsed=(jnow-starttime)*24*3600;
    );
    goal1(CS,iterasyon)= G1.l;
    goal2(CS,iterasyon)= G2.l;
    goal3(CS,iterasyon)= G3.l;
    istasyon(CS,iterasyon)= toplamst.l;
    calisan(CS,iterasyon)=toplaml.l;
    urun(CS,iterasyon)=Product.l;

    put    wmax    takip    goal1(CS,iterasyon)    goal2(CS,iterasyon)
goal3(CS,iterasyon)    istasyon(CS,iterasyon)    calisan(CS,iterasyon)    urun(CS,iterasyon)
elapsed/;

    takip=takip+1;
    d1=100000000;
    d2=100000000;
    d3=100000000;
);
wmax=2;
takip=1;
);
display goal1,goal2,goal3,calisan, istasyon, urun;
putclose prb4;

```

Ek 2. Ana Döngü Kodlar (AUGMECON2)

model Ornek1;

sets

i görevler /1*4/

k parça /1*4/

w çalışan /1*4/

b çevrim süresi katsayısı /1*10/

cycle/0*13/

balance/0*58/

CS/1*2/

;

alias(i,s,j,h);

Scalar

L büyük bir sayı/1000000/

C çevrim süresi katsayısı/10/

Time müsait zaman /10000/

wmax hücreye atanabilecek max çalışan/1/

FC sabit maliyet/1000/

SC hurda maliyeti/200/

r2/6.28/

r3/28.63/

cevrim/74/

mincyc/74/

dengeleme/4/

minbal/4/

step/0.5/

starttime

elapsed

;

sets

PAND(i,s) görev i nin ve öncülleri

*Problem1

/2.1, 3.1/

*Problem2

*/2.1, 3.1, 5.3/

*Problem3

*/2.1, 3.1, 5.4, 6.4/

*Problem4

*/2.1, 3.1, 4.1, 6.4, 7.4/

*Problem5

*/2.1, 3.1, 4.1, 6.2, 6.3, 6.5, 7.3, 7.4/

Ek 2'nin devamı

PORi(i)

*Problem1, Problem2, Problem3

/4/

*Problem4

*/5/

*Problem5

*/5, 8/

POR(i,s) görev i nin veya öncülleri

*Problem1, Problem2, Problem3

/4.2, 4.3/

*Problem4

*/5.2, 5.3/

*Problem5

*/5.2, 5.3, 8.6, 8.7/

SORi(i)

*Problem3, Problem4

*/4/

SOR(i,s) görev i nin veya öncülleri

*Problem3

*/4.5, 4.6/

*Problem4

*/4.5, 4.6/

;

Parameters

t(i) görev süreleri

*Problem 1

\$call GDXXRW demontaj.xlsx par=t rng=t!a2:b5 rdim=1 cdim=0

*Problem 2

*\$call GDXXRW demontaj.xlsx par=t rng=t!d2:e6 rdim=1 cdim=0

*Problem 3

*\$call GDXXRW demontaj.xlsx par=t rng=t!g2:h7 rdim=1 cdim=0

*Problem 4

*\$call GDXXRW demontaj.xlsx par=t rng=t!j2:k8 rdim=1 cdim=0

*Problem 5

*\$call GDXXRW demontaj.xlsx par=t rng=t!m2:n9 rdim=1 cdim=0

\$GDXIN demontaj.gdx

\$LOAD t

\$GDXIN

f(i,w) işçi becerileri

*Problem 1

\$call GDXXRW demontaj.xlsx par=f rng=f!a1:e5 rdim=1 cdim=1

*Problem 2

Ek 2'nin devamı

```
*$call GDXXRW demontaj.xlsx par=f rng=f!a7:e12 rdim=1 cdim=1
*Problem 3
*$call GDXXRW demontaj.xlsx par=f rng=f!a14:e20 rdim=1 cdim=1
*Problem 4
*$call GDXXRW demontaj.xlsx par=f rng=f!i1:m8 rdim=1 cdim=1
*Problem 5
*$call GDXXRW demontaj.xlsx par=f rng=f!i10:m18 rdim=1 cdim=1
$GDXIN demontaj.gdx
$LOAD f
$GDXIN
```

d(k) talep

```
*Problem 1
$call GDXXRW demontaj.xlsx par=d rng=d!a2:b5 rdim=1 cdim=0
*Problem 2
*$call GDXXRW demontaj.xlsx par=d rng=d!d2:e5 rdim=1 cdim=0
*Problem 3
*$call GDXXRW demontaj.xlsx par=d rng=d!g2:h5 rdim=1 cdim=0
*Problem 4
*$call GDXXRW demontaj.xlsx par=d rng=d!j2:k5 rdim=1 cdim=0
*Problem 5
*$call GDXXRW demontaj.xlsx par=d rng=d!m2:n5 rdim=1 cdim=0
$GDXIN demontaj.gdx
$LOAD d
$GDXIN
```

m(i,k) görev*parça ilişkisi

```
*Problem 1
$call GDXXRW demontaj.xlsx par=m rng=m!a1:e5 rdim=1 cdim=1
*Problem 2
*$call GDXXRW demontaj.xlsx par=m rng=m!a7:e12 rdim=1 cdim=1
*Problem 3
*$call GDXXRW demontaj.xlsx par=m rng=m!a14:e20 rdim=1 cdim=1
*Problem 4
*$call GDXXRW demontaj.xlsx par=m rng=m!i1:m8 rdim=1 cdim=1
*Problem 5
*$call GDXXRW demontaj.xlsx par=m rng=m!i10:m18 rdim=1 cdim=1
$GDXIN demontaj.gdx
$LOAD m
$GDXIN
```

WC(w) talep

```
*Problem 1
$call GDXXRW demontaj.xlsx par=wc rng=wc!a2:b5 rdim=1 cdim=0
*Problem 2
*$call GDXXRW demontaj.xlsx par=wc rng=wc!d2:e5 rdim=1 cdim=0
*Problem 3
```

Ek 2'nin devamı

```
*$call GDXXRW demontaj.xlsx par=wc rng=wc!g2:h5 rdim=1 cdim=0
*Problem 4
*$call GDXXRW demontaj.xlsx par=wc rng=wc!j2:k5 rdim=1 cdim=0
*Problem 5
*$call GDXXRW demontaj.xlsx par=wc rng=wc!m2:n5 rdim=1 cdim=0
$GDXIN demontaj.gdx
$LOAD wc
$GDXIN
```

```
goal1(CS,balance,cycle)
goal2(CS,balance,cycle)
goal3(CS,balance,cycle)
istasyon(CS,balance,cycle)
urun(CS,balance,cycle)
calisan(CS,balance,cycle)
```

Binary Variables

```
x(i,j,w) i görevi j istasyonuna w çalışanına atandıysa
y(i,s)
st(j) j. isyasyon açıldıysa
ws(j,w) w. çalışan j isyasyonundaysa
epor(i,s)
Pr(b)
;
```

Integer Variables

```
p(i)
Product
toplamst
toplamcl
;
*p.up(i)=10000;
*Product.up=10000;
```

Positive Variables

```
CT
strt(i,j)
MKC
MNC
G32
S2
S3
;
```

Variables

```
G1 amaç1
G2 amaç2
```

Ek 2'nin devamı

G3 amaç3

;

Equations

amac1 maliyetin minimize edilmesi

amac2 çevrim süresi

amac3 çalışanlar arasında dengesizlik

amac32

Kisit4 her görevin 1 ist. 1 çalışan tarafından yerine getirilebilmesi

Kisit5 görev atamaları ile çalışan atamaları arasında ilişki

Kisit6 çalışan atamaları ile istasyon açma arasında ilişki

Kisit7 bir çalışan bir istasyona atanabilir

Kisit8 görev atama ile görev yapma miktarı arasında ilişki

Kisit9 sökülen parça sayısının talebi karşılama

Kisit10 Demonte edilen ürün sayısının bulunması

Kisit11 Çevrim süresi indisinin hesaplanması

Kisit12 Çevrim süresinin hesaplanması

Kisit13 VE öncülü

Kisit14 VEYA öncülü

Kisit15 VEYA öncülünün bulunması

*Kisit16 VEYA ardılı

Kisit17 bir görevin başlangıç zamanı ve süresinin toplamının çevrim süresini geçmemesi

Kisit18 VE öncülü işlem süresi

Kisit19 VEYA öncülü işlem süresi

Kisit20 çalışanların yaptığı görevlerin zamanlarının çakışmasının engellenmesi

Kisit21 çalışanların yaptığı görevlerin zamanlarının çakışmasının engellenmesi

Kisit22 Maksimum çalışma zamanı

Kisit23 Minimum çalışma zamanı

Kisit24 Toplam istasyon

Kisit25 Toplam çalışan

Kisit26

Kisit27

;

amac1..

$G1=e=FC*\sum(j,st(j))+SC*Product+\sum(w,WC(w)*\sum(j,ws(j,w)))-0.001*((S2/r2)+0.1*(S3/r3));$

amac2.. $G2=e=CT;$

amac3.. $G32=e=MKC-MNC;$

amac32.. $G3=e=G32;$

Kisit4(i).. $\sum((j,w),x(i,j,w))=l=1;$

Kisit5(j,w).. $\sum(i,x(i,j,w))=l=card(i)*ws(j,w);$

Kisit6(j).. $\sum(w,ws(j,w))=l=wmax*st(j);$

Kisit7(w).. $\sum(j,ws(j,w))=l=1;$

Kisit8(i).. $p(i)=l=\sum((j,w),x(i,j,w))*L;$

Kisit9(k).. $d(k)=l=\sum(i,p(i)*m(i,k));$

Ek 2'nin devamı

```

Kisit10(i)..          Product=g=p(i);
Kisit11..            C*sum(b,Pr(b))=g=Product;
Kisit12(b)..         CT*C*ord(b)=l=Time+L*(1-Pr(b));
Kisit13(PAND(i,s),j)..          sum(w,x(i,j,w))=l=sum(h$(ord(h) le
ord(j)),sum(w,x(s,h,w)));
Kisit14(POR(i,s),j)..          sum(w,x(i,j,w))=l=sum(h$(ord(h) le
ord(j)),sum(w,x(s,h,w)))+L*(1-epor(i,s));
Kisit15(PORi(i))..          sum((POR(i,s)),epor(i,s))=g=1;
*Kisit16(SORi(i))..
sum((j,w),x(i,j,w))=g=sum(SOR(i,s),sum((j,w),x(s,j,w)));
Kisit17(i,j)..          strt(i,j)+sum(w,t(i)*f(i,w)*x(i,j,w))=l=CT;
Kisit18(PAND(i,s),j)..          strt(i,j)-strt(s,j)+L*(1-sum(w,x(s,j,w)))+L*(1-
sum(w,x(i,j,w)))=g=sum(w,x(s,j,w)*t(s)*f(s,w));
Kisit19(POR(i,s),j)..          strt(i,j)-strt(s,j)+L*(1-sum(w,x(s,j,w)))+L*(1-
sum(w,x(i,j,w)))=g=sum(w,x(s,j,w)*t(s)*f(s,w))-L*(1-epor(i,s));
Kisit20(i,s,j,w)$ (ord(s) gt ord(i))..          strt(i,j)-strt(s,j)+L*(1-x(s,j,w))+L*(1-
x(i,j,w))+L*y(i,s)=g=x(s,j,w)*t(s)*f(s,w);
Kisit21(i,s,j,w)$ (ord(s) gt ord(i))..          strt(s,j)-strt(i,j)+L*(1-x(s,j,w))+L*(1-
x(i,j,w))+L*(1-y(i,s))=g=x(i,j,w)*t(i)*f(i,w);
Kisit22(w)..          MKC=g=sum((i,j),x(i,j,w)*t(i)*f(i,w));
Kisit23(w)..          MNC=l=sum((i,j),x(i,j,w)*t(i)*f(i,w))+L*(1-
sum(j,ws(j,w)));
Kisit24..          toplamst=e=sum(j,st(j));
Kisit25..          toplamcl=e=sum((j,w),ws(j,w));
Kisit26..          G2+S2=e=cevrim;
Kisit27..          G3+S3=e=dengeleme;

```

```

model Ornek1 /all/;
OPTION reslim =1000000;
OPTION ITERLIM = 90000000;
*option intVarUp = 0;
option optcr=0;
*option threads = -1;
*starttime=jnow;
*solve Ornek1 using mip minimizing G1;
*elapsed=(jnow-starttime)*24*3600;
*display G1.l, G2.l, G3.l, MKC.l, MNC.l, toplamst.l, toplamcl.l, Product.l, elapsed,
x.l, strt.l, ws.l, epor.l, st.l;

```

```
file problem1/C:\Users\ASUS\Desktop\problem1.txt/;
```

```
put problem1/;
```

```
loop (CS,
```

```
loop (balance,
```

```
loop (cycle,
```

```
starttime=jnow;
```

```
solve Ornek1 using mip minimizing G1;
```

```
elapsed=(jnow-starttime)*24*3600;
```

Ek 2'nin devamı

```

goal1(CS,balance,cycle)= G1.l;
goal2(CS,balance,cycle)= G2.l;
goal3(CS,balance,cycle)= G3.l;
istasyon(CS,balance,cycle)= toplamst.l;
calisan(CS,balance,cycle)=toplaml.l;
urun(CS,balance,cycle)=Product.l
if(Ornek1.modelStat = %modelStat.optimal%,
    put wmax cevrim dengeleme goal1(CS,balance,cycle)
goal2(CS,balance,cycle) goal3(CS,balance,cycle) istasyon(CS,balance,cycle)
calisan(CS,balance,cycle) urun(CS,balance,cycle) elapsed/;
    );
    cevrim=cevrim+step;
    );
    dengeleme=dengeleme+step;
    cevrim=mincyc;
    );
    wmax=2;
    dengeleme=minbal;
    );
display goal1,goal2,goal3,calisan, istasyon, urun;
putclose problem1;

```

Ek 3. Sınır Değerleri Tablosu

Ek Tablo 1. Küçük boyutlu problemler için AUGMECON2 yönteminin sonuçları

Problem-1/ Strateji-1			Problem-4/ Strateji-1			Problem-5/ Strateji-1		
<i>f1</i>	<i>f2</i>	<i>f3</i>	<i>f1</i>	<i>f2</i>	<i>f3</i>	<i>f1</i>	<i>f2</i>	<i>f3</i>
23030	80.43	4.33	22740	55.79	0.91	22150	72.74	6.33
23000	80.43	6.13	22790	57.67	0.56	23440	57.21	7.10
24390	74.14	32.97	21460	89.20	0.53	23440	56.16	8
Problem-1/ Strateji-2			22650	57.88	2.08	22080	74.64	8.22
<i>f1</i>	<i>f2</i>	<i>f3</i>	22650	57.67	3.04	22080	72.74	8.57
22030	80.43	4.33	21370	88.14	6.84	23440	55.98	9.13
22000	80.43	6.13	21340	90.66	9.36			
23390	74.14	32.97	21340	88.67	9.86			
Problem-2/ Strateji-1			Problem-4/ Strateji-2			Problem-5/ Strateji-2		
<i>f1</i>	<i>f2</i>	<i>f3</i>	<i>f1</i>	<i>f2</i>	<i>f3</i>	<i>f1</i>	<i>f2</i>	<i>f3</i>
23530	90.75	12.10	22740	55.79	0.91	22150	72.74	6.33
23530	87.20	13.37	22740	56.80	0	21150	86.67	6.33
23370	84.11	14.64	21740	58.63	0.91	21440	82.84	6.78
24770	66.95	28.27	21740	71.33	0	21080	86.73	6.96
Problem-2/ Strateji-2			21460	89.20	0.53	22440	57.21	7.10
<i>f1</i>	<i>f2</i>	<i>f3</i>	22650	57.88	2.08	22440	56.16	8
23530	90.75	12.10	21650	73.27	2.08	22080	74.64	8.22
23530	87.20	13.37	22650	57.67	3.04	22080	72.74	8.57
23370	84.11	14.64	21650	71.03	3.04	21080	84.55	8.57
23770	66.95	28.27	21740	57.55	4.39	22440	55.98	9.13
Problem-3/ Strateji-1			21650	66.69	5.02	21150	83.51	9.07
<i>f1</i>	<i>f2</i>	<i>f3</i>	21650	61.62	5.82			
23200	68.88	2.28	21650	57.55	6.65			
23140	68.88	3.13	21370	88.14	6.84			
21910	95.80	3.69	21340	90.66	9.36			
23140	66.70	5.46	21340	88.67	9.86			
Problem-3/ Strateji-2			20510	91.23	13.48			
<i>f1</i>	<i>f2</i>	<i>f3</i>	20340	91.13	15.99			
22200	68.88	2.28	22050	55.91	18.19			
22140	68.88	3.13	22050	55.79	18.55			
21910	95.80	3.69						
22140	66.70	5.46						

Ek 4. Pareto Optimal Sıralama Kodlar (NSGA-II)

```

function [populasyon] =
Pareto(bireysayisi,populasyon,arapop,SC,FC,WC2,d,m,t,PL,PL2,skill,WMAX,mg)
kromozom=zeros(size(populasyon,1),size(populasyon,2),size(populasyon,3)*2);
kromozom(:,1:size(populasyon,3))=populasyon;
kromozom(:,size(populasyon,3)+1:end)=arapop;
populasyon=kromozom;
[ R ] = Objectives( populasyon,SC,FC,WC2,d,m,t,PL,PL2,skill,WMAX,mg);
P=zeros(size(R,1),3);
i=1;
while size(R,1)>0
    essiz=unique(R(:,1:3),'rows','stable');
    R(:,4)=sum((bsxfun(@ge,R(:,1),essiz(:,1))).*bsxfun(@ge,R(:,2),essiz(:,2))).*bsxfun(
    @ge,R(:,3),essiz(:,3))),2);
    uzakliklar=[inf*ones(numel(find(R(:,4)==1)),3) R(find(R(:,4)==1),:)];
    [siralama1,sira1]=sort(uzakliklar(:,4));
    [siralama2,sira2]=sort(uzakliklar(:,5));
    [siralama3,sira3]=sort(uzakliklar(:,6));
    if numel(siralama1)>=3
        for j=2:(numel(siralama1)-1)
            uzakliklar(sira1(j),1)=(siralama1(j+1)-siralama1(j-1))/(siralama1(end)-
siralama1(1)+eps);
            uzakliklar(sira2(j),2)=(siralama2(j+1)-siralama2(j-1))/(siralama2(end)-
siralama2(1)+eps);
            uzakliklar(sira3(j),3)=(siralama3(j+1)-siralama3(j-1))/(siralama3(end)-
siralama3(1)+eps);
        end
    end
    P(R(find(R(:,4)==1),5),1)=i;
    P(R(find(R(:,4)==1),5),2)=sum(uzakliklar(:,1:3),2);
    P(R(find(R(:,4)==1),5),3)=R(find(R(:,4)==1),5);
    R(find(R(:,4)==1),:)=[];
    i=i+1;
end
P(:,2)=-1*P(:,2);
H=unique(P,'rows');
populasyon=populasyon(:,H(1:bireysayisi,3));
end

```

Ek 5. Ana Döngü Kodlar (NSGA-II)

```

function [ Pareto,sure] =
MainLoop(wtime,SC,FC,PAND,POR,SOR,m,PL,WC,skill,iterasyon,t,d,WMAX,bireysayi
si,caporan,mutoran,point )
    basla=clock;
    PL2=[PL max(PL,[],2)*ones(1,WMAX)];
    WC2=[WC;max(WC)*ones(WMAX,1)];
    mg=0;
    [ populasyon ] = Population(
WMAX,PL,PL2,skill,d,t,wtime,SC,FC,WC2,PAND,POR,SOR,m,bireysayisi,mg);
    for mg=1:iterasyon
        if point==1
            [ arapop ] = Crossover1( populasyon,caporan,PAND,POR,PL,PL2,t,WMAX );
        elseif point==2
            [ arapop ] = Crossover2( populasyon,caporan,PAND,POR,PL,PL2,t,WMAX );
        end
        [ arapop ] = Mutation( arapop,mutoran,PAND,POR,PL,PL2,t,WMAX);
        [populasyon] =
Pareto(bireysayisi,populasyon,arapop,SC,FC,WC2,d,m,t,PL,PL2,skill,WMAX,mg);
    end
    sure=etime(clock,basla);
    [ R ] = Objectives( populasyon,SC,FC,WC2,d,m,t,PL,PL2,skill,WMAX,mg);
    Pareto=unique(R(:,1:3),'rows','stable');
    Pareto(:,4)=sum((bsxfun(@ge,Pareto(:,1),Pareto(:,1')).*bsxfun(@ge,Pareto(:,2),Paret
o(:,2'))).*bsxfun(@ge,Pareto(:,3),Pareto(:,3'))),2);
    Pareto=Pareto(find(Pareto(:,4)==1),1:3);
end

```

Ek 6. Kromozomun Çözümlemesi Kodlar (NSGA-II)

```

function [ R ] = Objectives(
populasyon,SC,FC,WC2,d,m,t,PL,PL2,skill,WMAX,mg)
    bsayisi=size(populasyon,3);
    maliyet = zeros(bsayisi,1);
    dengeleme = zeros(bsayisi,1);
    CT=zeros(bsayisi,1);
    skill2=[skill ones(size(skill,1),WMAX)];
    for
    b=1:bsayisiceza=sum(diag(skill2(populasyon(1,find(populasyon(1, :,b)>0),b),populasyon(2,find(populasyon(1, :,b)>0),b))))*mg;
        maliyet(b)=FC*max(populasyon(3, :,b))+SC*max(ceil(d'./sum(m(populasyon(1,find(populasyon(1, :,b)>0),b), :,1)))+sum(WC2.*sign(sum(bsxfun(@eq,(1:size(PL2,2))',populasyon(2, :,b)),2)),1)+ceza;
        CT(b)=max(populasyon(5, :,b))+ceza;
        worker=sum(bsxfun(@eq,(1:size(PL,2))',populasyon(2,find(populasyon(2, :,b)>0),b)).*(ones(size(PL,2),1)*t(populasyon(1,find(populasyon(2, :,b)>0),b))'.*PL(populasyon(1,find(populasyon(2, :,b)>0),b), :)',2));
        worker(find(worker==0))=[];
        dengeleme(b)=max(worker)-min(worker)+ceza;
    end
    R=[maliyet CT dengeleme zeros(bsayisi,1) (1:bsayisi)'];
end

```

Ek 7. İlk Popülasyonun Oluşturulması Kodlar (NSGA-II)

```

function [ populasyon ] = Population(
WMAX,PL,PL2,skill,d,t,wtime,SC,FC,WC2,PAND,POR,SOR,m,bireysayisi,mg)
boyut=size(t,1);
populasyon=zeros(5,boyut,bireysayisi);
for b=1:bireysayisi
    kPOR=zeros(size(POR));
    km=m;
    oncelikliler=find(sum(POR,2)>0);
    while numel(oncekliler)>0
        onculler=find(POR(oncekliler(1,:),:)>0);
        rs=randperm(numel(onculler));
        kPOR(oncekliler(1),onculler(rs(1:randi(numel(onculler),1))))=1;
        oncekliler(1)=[];
    end
    kPOR=kPOR+bsxfun(@ne,kPOR,POR)';
    while numel(find(sum(km,1)>1))>0
        parcalar=find(sum(km,1)>1);
        gorevler=find(km(:,parcalar(1))>0);
        gorevler=gorevler(randperm(numel(gorevler)));
        km(gorevler(2:end),parcalar(1))=0;
    end
    P=PAND+kPOR;
    gereksiz=find(sum(bsxfun(@eq,(1:boyut)',[(find(sum(abs(km(:,find(d>0))),2)>0))'
    find(sum(P(find(sum(abs(km(:,find(d>0))),2)>0),:),1)>0]),2)==0);
    kSOR=zeros(size(SOR));
    kSOR(gereksiz,:)=1;
    kPS=PAND+kPOR+kSOR;
    for i=1:boyut
        atanabilecek=find(sum(kPS,2)==0);
        if numel(atanabilecek)==0
            break
        end
        gen=atanabilecek(randi(numel(atanabilecek),1));
        populasyon(1,i,b)= gen;
        kPS(gen,:)=1;
        kPS(:,gen)=0;
        if sum(SOR(:,gen),1)>0
            kPS(find(SOR(find(SOR(:,gen)==1),:)==1),:)=1;
        end
    end
    end
    gorevler=[populasyon(1,(populasyon(1,:,b)>0),b)
    find(sum(bsxfun(@eq,populasyon(1,(populasyon(1,:,b)>0),b)',1:size(populasyon,2)),1)==0
   )]; sPOR=bsxfun(@gt,(1:size(POR,1))',1:size(POR,1)).*POR(gorevler,gorevler);
    epor=zeros(size(POR));
    epor(gorevler,gorevler)=bsxfun(@(x,y) sPOR(x,y).*(1-sign(sum(sPOR(x,1:(y-
    1)),2))),(1:size(POR,1))',1:size(POR,1));

```

Ek 7'nin devamı

```

P=epor+PAND;
CTU=randi([ceil(max(t)*max(max(PL,[],2))),floor(wtime/max(ceil(d'./sum(m(populasyon(
1,find(populasyon(1, :,b)>0),b),:),1)))]),1);
calisan=randperm(size(PL,2));
atama=calisan(1:WMAX);
calisan(1:WMAX)=[];
istasyon=1;
for i=1:boyut
    if populasyon(1,i,b)==0
        break
    end
    populasyon(2,i,b)=atama(randi(numel(atama),1));
    populasyon(3,i,b)=istasyon;
populasyon(4,i,b)=max(max(populasyon(5,find(populasyon(3, :,b)==istasyon),b).*P(popula
syon(1,i,b),populasyon(1,find(populasyon(3, :,b)==istasyon),b))),max(populasyon(5,(find(p
opulasyon(3, :,b)==istasyon,1)-
1+find(populasyon(2,find(populasyon(3, :,b)==istasyon,1):i,b)==populasyon(2,i,b))),b));
populasyon(5,i,b)=populasyon(4,i,b)+t(populasyon(1,i,b))*PL2(populasyon(1,i,b),populas
yon(2,i,b));
    atama2=atama;
    while populasyon(5,i,b)>CTU
        atama2(find(populasyon(2,i,b)==atama2))=[];
        populasyon(2:5,i,b)=0;
        if numel(atama2)==0
            istasyon=istasyon+1;
            if numel(calisan)>=WMAX
                atama=calisan(1:WMAX);
                calisan(1:WMAX)=[];
            elseif
numel(find(sum(bsxfun(@eq,populasyon(2, :,b)',1:size(PL,2)),1)==0))>0
                aylaklar=find(sum(bsxfun(@eq,populasyon(2, :,b)',1:size(PL,2)),1)==0);
                aylaklar=[aylaklar(randperm(numel(aylaklar)))
(size(PL,2)+1:size(PL,2))];
                atama=aylaklar(1:WMAX);
            else
                atama=(size(PL,2)+1):(size(PL,2)+WMAX);
            end
            populasyon(2,i,b)=atama(randi(numel(atama),1));
            populasyon(3,i,b)=istasyon;
            populasyon(4,i,b)=0;
populasyon(5,i,b)=t(populasyon(1,i,b))*PL2(populasyon(1,i,b),populasyon(2,i,b));
        else
            populasyon(2,i,b)=atama2(randi(numel(atama2),1));
            populasyon(3,i,b)=istasyon;

```


Ek 7'nin devamı

```

    populasyon(4,i,b)=max(max(populasyon(5,find(populasyon(3,:,b)==istasyon),b).*P(
    populasyon(1,i,b),populasyon(1,find(populasyon(3,:,b)==istasyon),b))),max(populasyon(5,
    (find(populasyon(3,:,b)==istasyon,1)-
    1+find(populasyon(2,find(populasyon(3,:,b)==istasyon,1):i,b)==populasyon(2,i,b))),b));
    populasyon(5,i,b)=populasyon(4,i,b)+t(populasyon(1,i,b))*PL2(populasyon(1,i,b),populas
    yon(2,i,b));
        end
    end
end
end
[ R ] = Objectives( populasyon,SC,FC,WC2,d,m,t,PL,PL2,skill,WMAX,mg);
P=zeros(size(R,1),3);
i=1;
while size(R,1)>0
    essiz=unique(R(:,1:3),'rows','stable');
R(:,4)=sum((bsxfun(@ge,R(:,1),essiz(:,1)).*bsxfun(@ge,R(:,2),essiz(:,2)).*bsxfun(@ge,R
(:,3),essiz(:,3))),2);
    uzakliklar=[inf*ones(numel(find(R(:,4)==1)),3) R(find(R(:,4)==1),:)];
    [siralama1,sira1]=sort(uzakliklar(:,4));
    [siralama2,sira2]=sort(uzakliklar(:,5));
    [siralama3,sira3]=sort(uzakliklar(:,6));
    if numel(siralama1)>=3
        for j=2:(numel(siralama1)-1)
            uzakliklar(sira1(j),1)=(siralama1(j+1)-siralama1(j-1))/(siralama1(end)-
siralama1(1)+eps);
            uzakliklar(sira2(j),2)=(siralama2(j+1)-siralama2(j-1))/(siralama2(end)-
siralama2(1)+eps);
            uzakliklar(sira3(j),3)=(siralama3(j+1)-siralama3(j-1))/(siralama3(end)-
siralama3(1)+eps);
        end
    end
    P(R(find(R(:,4)==1),5),1)=i;
    P(R(find(R(:,4)==1),5),2)=sum(uzakliklar(:,1:3),2);
    P(R(find(R(:,4)==1),5),3)=R(find(R(:,4)==1),5);
    R(find(R(:,4)==1),:)=[];
    i=i+1;
end
P(:,2)=-1*P(:,2);
H=unique(P,'rows');
populasyon=populasyon(:,H(1:bireysayisi,3));
end

```

Ek 8. Seçilim ve Tek Noktalı Çaprazlama Operatörü Kodlar (NSGA-II)

```

function [ arapop ] = Crossover1( populasyon,caporan,PAND,POR,PL,PL2,t,WMAX
)

%Seçilim
bsayisi=size(populasyon,3);
ParentA=populasyon(:,min(randi(bsayisi,2,ceil(bsayisi/2))))';
ParentB=populasyon(:,min(randi(bsayisi,2,ceil(bsayisi/2))))';
%Çaprazlama
ciftler=[randperm(ceil(bsayisi/2)) randperm(ceil(bsayisi/2))];
arapop=zeros(size(populasyon));
for b=1:round(bsayisi/2)
    adayanne = ParentA(:,ciftler(b,1));
    adaybaba = ParentB(:,ciftler(b,2));
    offspringA=adayanne;
    offspringB=adaybaba;
    capolasilik=unifrnd(0,1);
    if (capolasilik<caporan)
        gorevlerA=[adayanne(1,(adayanne(1,:)>0))
find(sum(bsxfun(@eq,adayanne(1,(adayanne(1,:)>0))',1:size(adayanne,2)),1)==0)];

sPORA=bsxfun(@gt,(1:size(POR,1))',1:size(POR,1)).*POR(gorevlerA,gorevlerA);
    eporA=zeros(size(POR,1),size(POR,2));
    eporA(gorevlerA,gorevlerA)=bsxfun(@(x,y) sPORA(x,y).*(1-
sign(sum(sPORA(x,1:(y-1)),2)),(1:size(POR,1))',1:size(POR,1)));
    gorevlerB=[adaybaba(1,(adaybaba(1,:)>0))
find(sum(bsxfun(@eq,adaybaba(1,(adaybaba(1,:)>0))',1:size(adaybaba,2)),1)==0)];

sPORB=bsxfun(@gt,(1:size(POR,1))',1:size(POR,1)).*POR(gorevlerB,gorevlerB);
    eporB=zeros(size(POR,1),size(POR,2));
    eporB(gorevlerB,gorevlerB)=bsxfun(@(x,y) sPORB(x,y).*(1-
sign(sum(sPORB(x,1:(y-1)),2)),(1:size(POR,1))',1:size(POR,1)));
    PA=sign(PAND+eporA);
    PB=sign(PAND+eporB);
    capist=randi(min(max(adayanne(3,:)),max(adaybaba(3,:))),1);
    CPA=find(adayanne(3,:)==capist,1):max(find(adayanne(3,:)>0));
    CPB=find(adaybaba(3,:)==capist,1):max(find(adaybaba(3,:)>0));
    gorevler=adayanne(1,CPA);
gorevler(find(sum(bsxfun(@eq,find((sum(POR,1)+sum(POR,2))>0)',gorevler),1)>0))=[];
siralama=sum(bsxfun(@eq,gorevler',adaybaba(1,:)),1).*(1:size(adaybaba,2));
    siralama(find(siralama==0))=[];
    PAcap=zeros(size(PAND));
    PAcap(adaybaba(1,siralama),adaybaba(1,siralama))=bsxfun(@gt,(1:numel(siralama))',1:nu
mel(siralama));
    PAK=sign(PA+PAcap);

```

Ek 8'in devamı

```

gorevler=adaybaba(1,CPB);
gorevler(find(sum(bsxfun(@eq,find((sum(POR,1)+sum(POR,2))>0)',gorevler),1)>0))=[];
siralama=sum(bsxfun(@eq,gorevler',adayanne(1,:),1),1).*(1:size(adayanne,2));
siralama(find(siralama==0))=[];
PBcap=zeros(size(PAND));
PBcap(adayanne(1,siralama),adayanne(1,siralama))=bsxfun(@gt,(1:numel(siralama))',1:nu
mel(siralama));
PBK=sign(PB+PBcap);
gorevlerA=sort(adayanne(1,CPA));
katalizorA=[];
gorevlerB=sort(adaybaba(1,CPB));
katalizorB=[];
while numel(gorevlerA)>0 || numel(gorevlerB)>0
    if numel(gorevlerA)>0
        if numel(gorevlerA)>0
            numel(gorevlerA(find(sum(PAK(gorevlerA,gorevlerA),2)==0)))==0
            PAK=PA;
        end
        atanabilecek=gorevlerA(find(sum(PAK(gorevlerA,gorevlerA),2)==0));
        gen=atanabilecek(randi(numel(atanabilecek),1));
        katalizorA=[katalizorA gen];
        gorevlerA(find(gorevlerA==gen,1))=[];
    end
    if numel(gorevlerB)>0
        if numel(gorevlerB)>0
            numel(gorevlerB(find(sum(PBK(gorevlerB,gorevlerB),2)==0)))==0
            PBK=PB;
        end
        atanabilecek=gorevlerB(find(sum(PBK(gorevlerB,gorevlerB),2)==0));
        gen=atanabilecek(randi(numel(atanabilecek),1));
        katalizorB=[katalizorB gen];
        gorevlerB(find(gorevlerB==gen,1))=[];
    end
end
offspringA(1,CPA)=katalizorA;
offspringB(1,CPB)=katalizorB;
offspringA(2:5,CPA)=0;
offspringB(2:5,CPB)=0;
gorevlerA=[offspringA(1,(offspringA(1,:)>0))
find(sum(bsxfun(@eq,offspringA(1,(offspringA(1,:)>0))',1:size(offspringA,2)),1)==0)];
sPORA=bsxfun(@gt,(1:size(POR,1))',1:size(POR,1)).*POR(gorevlerA,gorevlerA);
eporA=zeros(size(POR,1),size(POR,2));
eporA(gorevlerA,gorevlerA)=bsxfun(@(x,y) sPORA(x,y).*(1-
sign(sum(sPORA(x,1:(y-1)),2))),1:size(POR,1))',1:size(POR,1));
gorevlerB=[offspringB(1,(offspringB(1,:)>0))
find(sum(bsxfun(@eq,offspringB(1,(offspringB(1,:)>0))',1:size(offspringB,2)),1)==0)];

```

Ek 8'in devamı

```

sPORB=bsxfun(@gt,(1:size(POR,1))',1:size(POR,1)).*POR(gorevlerB,gorevlerB);

    eporB=zeros(size(POR,1),size(POR,2));
    eporB(gorevlerB,gorevlerB)=bsxfun(@x,y)          sPORB(x,y).*(1-
sign(sum(sPORB(x,1:(y-1)),2))), (1:size(POR,1))',1:size(POR,1));
    PA=sign(PAND+eporA);
    PB=sign(PAND+eporB);
    atamaA=adaybaba(2,find(adaybaba(3,:)==capist));
    wanne=find(sum(bsxfun(@eq,[offspringA(2,:)
adaybaba(2,(max(find(adaybaba(3,:)==capist))+1):end)]',1:size(PL,2)),1)>0);
    if      numel(find(sum(bsxfun(@eq,wanne',atamaA),1)>0))>0           ||
numel(find(atamaA>size(PL,2)))>0
        aylaklar=find(sum(bsxfun(@eq,wanne',1:size(PL,2)),1)==0);
        if      numel(find(sum(bsxfun(@eq,wanne',atamaA),1)>0))>0       &&
numel(find(atamaA>size(PL,2)))>0
            degistir=[find(sum(bsxfun(@eq,atamaA(find(sum(bsxfun(@eq,wanne',atamaA),1)>0
))',1:size(PL,2)),1)>0)
size(PL,2)+find(sum(bsxfun(@eq,atamaA',(size(PL,2)+1):size(PL2,2)),1)>0)];
            elseif numel(find(sum(bsxfun(@eq,wanne',atamaA),1)>0))>0
                degistir=find(sum(bsxfun(@eq,atamaA(find(sum(bsxfun(@eq,wanne',atamaA),1)>0)
)',1:size(PL,2)),1)>0);
            elseif numel(find(atamaA>size(PL,2)))>0
                degistir=size(PL,2)+find(sum(bsxfun(@eq,atamaA',(size(PL,2)+1):size(PL2,2)),1)>0
);
        end
        aylaklar=[aylaklar(randperm(numel(aylaklar))) (size(PL,2)+1):size(PL2,2)];

atamaA=((sum(bsxfun(@eq,degistir',atamaA),1)~=1).*atamaA)+aylaklar(1:numel(degistir
))*bsxfun(@eq,degistir',atamaA);
    end
    atamaB=adayanne(2,find(adayanne(3,:)==capist));
    wbaba=find(sum(bsxfun(@eq,[offspringB(2,:)
adayanne(2,(max(find(adayanne(3,:)==capist))+1):end)]',1:size(PL,2)),1)>0);           if
numel(find(sum(bsxfun(@eq,wbaba',atamaB),1)>0))>0           ||
numel(find(atamaB>size(PL,2)))>0
        aylaklar=find(sum(bsxfun(@eq,wbaba',1:size(PL,2)),1)==0);
        if      numel(find(sum(bsxfun(@eq,wbaba',atamaB),1)>0))>0       &&
numel(find(atamaB>size(PL,2)))>0
            degistir=[find(sum(bsxfun(@eq,atamaB(find(sum(bsxfun(@eq,wbaba',atamaB),1)>0
))',1:size(PL,2)),1)>0)
size(PL,2)+find(sum(bsxfun(@eq,atamaB',(size(PL,2)+1):size(PL2,2)),1)>0)];
            elseif numel(find(sum(bsxfun(@eq,wbaba',atamaB),1)>0))>0
                degistir=find(sum(bsxfun(@eq,atamaB(find(sum(bsxfun(@eq,wbaba',atamaB),1)>0)
)',1:size(PL,2)),1)>0);
            elseif numel(find(atamaB>size(PL,2)))>0
                degistir=size(PL,2)+find(sum(bsxfun(@eq,atamaB',(size(PL,2)+1):size(PL2,2)),1)>0
);
    end

```

Ek 8'in devamı

```

end
    aylaklar=[aylaklar(randperm(numel(aylaklar))) (size(PL,2)+1):size(PL2,2)];
atamaB=((sum(bsxfun(@eq,degistir',atamaB),1)~=1).*atamaB)+aylaklar(1:numel(degistir)
)*bsxfun(@eq,degistir',atamaB);
end

CTUA=max(ceil(max(t)*max(max(PL,[],2))),max(adayanne(5,:)));
CTUB=max(ceil(max(t)*max(max(PL,[],2))),max(adaybaba(5,:)));
istasyonA=capist;
istasyonB=capist;
kA=0;
kB=0;
for
i=min(find(adayanne(3,:)==capist,1),find(adaybaba(3,:)==capist,1)):size(populasyon,2)
    if offspringA(1,i)==0 && offspringB(1,i)==0
        break
    end
    if offspringA(1,i)>0 && offspringA(2,i)==0
        kA=kA+1;
        offspringA(2,i)=atamaA(mod((kA-1),numel(atamaA))+1);
        offspringA(3,i)=istasyonA;
        offspringA(4,i)=max(max(offspringA(5,find(offspringA(3,:)==istasyonA)).*PA(offspring
A(1,i),offspringA(1,find(offspringA(3,:)==istasyonA))),max(offspringA(5,(find(offspring
A(3,:)==istasyonA,1)-
1+find(offspringA(2,find(offspringA(3,:)==istasyonA,1):i)==offspringA(2,i))))));
        offspringA(5,i)=offspringA(4,i)+t(offspringA(1,i))*PL2(offspringA(1,i),offspringA(2,i));
        atama2A=atamaA;
        while offspringA(5,i)>CTUA
            atama2A(find(offspringA(2,i)==atama2A))=[];
            offspringA(2:5,i)=0;
            if numel(atama2A)==0
                istasyonA=istasyonA+1;
                kA=1;
                if numel(find(adaybaba(3,:)==istasyonA))>0
                    atamaA=adaybaba(2,find(adaybaba(3,:)==istasyonA));
                    wanne=find(sum(bsxfun(@eq,[offspringA(2,:)
adaybaba(2,(max(find(adaybaba(3,:)==istasyonA))+1:end)],1:size(PL,2)),1)>0);
                    if numel(find(sum(bsxfun(@eq,wanne',atamaA),1)>0))>0 ||
numel(find(atamaA>size(PL,2)))>0
                        aylaklar=find(sum(bsxfun(@eq,wanne',1:size(PL,2)),1)==0);
                        if numel(find(sum(bsxfun(@eq,wanne',atamaA),1)>0))>0 &&
numel(find(atamaA>size(PL,2)))>0
                            degistir=[find(sum(bsxfun(@eq,atamaA(find(sum(bsxfun(@eq,wanne',atamaA),1)>0))',1:s
ize(PL,2)),1)>0)
size(PL,2)+find(sum(bsxfun(@eq,atamaA',(size(PL,2)+1):size(PL2,2)),1)>0)];

```

Ek 8'in devamı

```

        elseif numel(find(sum(bsxfun(@eq,wanne',atamaA),1)>0))>0
            degistir=find(sum(bsxfun(@eq,atamaA(find(sum(bsxfun(@eq,wanne',atamaA),1)>0)
)',1:size(PL,2)),1)>0);
            elseif numel(find(atamaA>size(PL,2)))>0
                degistir=size(PL,2)+find(sum(bsxfun(@eq,atamaA',(size(PL,2)+1):size(PL2,2)),1)>0
);
            end
            aylaklar=[aylaklar(randperm(numel(aylaklar)))                (size(PL,2)+1):size(PL2,2)];
            atamaA=((sum(bsxfun(@eq,degistir',atamaA),1)~=1).*atamaA)+aylaklar(1:numel(degistir
))*bsxfun(@eq,degistir',atamaA);
            end
            if
                numel(find(sum(bsxfun(@eq,atamaA',1:size(PL,2)),1)>0))<WMAX                &&
                numel(find(sum(bsxfun(@eq,[atamaA  offspringA(2,:)]',1:size(PL,2)),1)==0))>=(WMAX-
                numel(find(sum(bsxfun(@eq,atamaA',1:size(PL,2)),1)>0)))
                    aylaklar=find(sum(bsxfun(@eq,[atamaA
                    offspringA(2,:)]',1:size(PL,2)),1)==0);
                    atamaA=[atamaA  aylaklar(randi(numel(aylaklar),1,(WMAX-
                    numel(find(sum(bsxfun(@eq,atamaA',1:size(PL,2)),1)>0)))));
                    atamaA=atamaA(randperm(numel(atamaA)));
            end
            elseif
                numel(find(sum(bsxfun(@eq,offspringA(2,:) ',1:size(PL,2)),1)==0))>0
                    calisanlar=find(sum(bsxfun(@eq,offspringA(2,:) ',1:size(PL,2)),1)==0);
                    calisanlar=[calisanlar(randperm(numel(calisanlar))) size(PL,2)+1:size(PL2,2)];
                    worker=calisanlar(1:WMAX);
                    atamaA=worker(randi(WMAX,1,size(PL,1)));
            else
                worker=(size(PL,2)+1):size(PL2,2);
                atamaA=worker(randi(WMAX,1,size(PL,1)));
            end
            offspringA(2,i)=atamaA(mod((kA-1),numel(atamaA))+1);
            offspringA(3,i)=istasyonA;
            offspringA(4,i)=0;

            offspringA(5,i)=t(offspringA(1,i))*PL2(offspringA(1,i),offspringA(2,i));
            else
                offspringA(2,i)=atama2A(mod((kA-1),numel(atama2A))+1);
                offspringA(3,i)=istasyonA;

            offspringA(4,i)=max(max(offspringA(5,find(offspringA(3,:)==istasyonA)).*PA(offspring
            A(1,i),offspringA(1,find(offspringA(3,:)==istasyonA))))),max(offspringA(5,(find(offspring
            A(3,:)==istasyonA,1)-
            1+find(offspringA(2,find(offspringA(3,:)==istasyonA,1):i)==offspringA(2,i)))));

            offspringA(5,i)=offspringA(4,i)+t(offspringA(1,i))*PL2(offspringA(1,i),offspringA(2,i));
            end

```

Ek 8'in devamı

```

        end
    end
    if offspringB(1,i)>0 && offspringB(2,i)==0
        kB=kB+1;
        offspringB(2,i)=atamaB(mod((kB-1),numel(atamaB))+1);
        offspringB(3,i)=istasyonB;
        offspringB(4,i)=max(max(offspringB(5,find(offspringB(3,:)==istasyonB)).*PB(offspringB
(1,i),offspringB(1,find(offspringB(3,:)==istasyonB))))),max(offspringB(5,(find(offspringB(
3,:)==istasyonB,1)-
1+find(offspringB(2,find(offspringB(3,:)==istasyonB,1):i)==offspringB(2,i))))));
        offspringB(5,i)=offspringB(4,i)+t(offspringB(1,i))*PL2(offspringB(1,i),offspringB(2,i));
        atama2B=atamaB;
        while offspringB(5,i)>CTUB
            atama2B(find(offspringB(2,i)==atama2B))=[];
            offspringB(2:5,i)=0;
            if numel(atama2B)==0
                istasyonB=istasyonB+1;
                kB=1;
                if numel(find(adayanne(3,:)==istasyonB))>0
                    atamaB=adayanne(2,find(adayanne(3,:)==istasyonB));
                    wbaba=find(sum(bsxfun(@eq,[offspringB(2,:)
adayanne(2,(max(find(adayanne(3,:)==istasyonB))+1):end)],1:size(PL,2)),1)>0);
                    if numel(find(sum(bsxfun(@eq,wbaba',atamaB),1)>0))>0    ||
numel(find(atamaB>size(PL,2)))>0
                        aylaklar=find(sum(bsxfun(@eq,wbaba',1:size(PL,2)),1)==0);
                        if numel(find(sum(bsxfun(@eq,wbaba',atamaB),1)>0))>0 &&
numel(find(atamaB>size(PL,2)))>0
                            degistir=[find(sum(bsxfun(@eq,atamaB(find(sum(bsxfun(@eq,wbaba',atamaB),1)>0
)),1:size(PL,2)),1)>0)
size(PL,2)+find(sum(bsxfun(@eq,atamaB',(size(PL,2)+1):size(PL,2)),1)>0)];
                            elseif numel(find(sum(bsxfun(@eq,wbaba',atamaB),1)>0))>0
                                degistir=find(sum(bsxfun(@eq,atamaB(find(sum(bsxfun(@eq,wbaba',atamaB),1)>0)
)',1:size(PL,2)),1)>0);
                                elseif numel(find(atamaB>size(PL,2)))>0
                                    degistir=size(PL,2)+find(sum(bsxfun(@eq,atamaB',(size(PL,2)+1):size(PL,2)),1)>0)
);
                            end
                            aylaklar=[aylaklar(randperm(numel(aylaklar)))          (size(PL,2)+1):size(PL,2)];
                            atamaB=((sum(bsxfun(@eq,degistir',atamaB),1)~=1).*atamaB)+aylaklar(1:numel(degistir)
)*bsxfun(@eq,degistir',atamaB);
                        end
                    end
                if
numel(find(sum(bsxfun(@eq,atamaB',1:size(PL,2)),1)>0))<WMAX          &&
numel(find(sum(bsxfun(@eq,[atamaB offspringB(2,:)]',1:size(PL,2)),1)==0))>=(WMAX-
numel(find(sum(bsxfun(@eq,atamaB',1:size(PL,2)),1)>0)))

```

Ek 8'in devamı

```

        aylaklar=find(sum(bsxfun(@eq,[atamaB
offspringB(2,:)'],1:size(PL,2)),1)==0);
        atamaB=[atamaB    aylaklar(randi(numel(aylaklar),1,(WMAX-
numel(find(sum(bsxfun(@eq,atamaB',1:size(PL,2)),1)>0))))));
        atamaB=atamaB(randperm(numel(atamaB)));
    end
    elseif
numel(find(sum(bsxfun(@eq,offspringB(2,:)'],1:size(PL,2)),1)==0)>0
    calisanlar=find(sum(bsxfun(@eq,offspringB(2,:)'],1:size(PL,2)),1)==0);
    calisanlar=[calisanlar(randperm(numel(calisanlar))) size(PL,2)+1:size(PL2,2)];
    worker=calisanlar(1:WMAX);
    atamaB=worker(randi(WMAX,1,size(PL,1)));
    else
    worker=(size(PL,2)+1):size(PL2,2);
    atamaB=worker(randi(WMAX,1,size(PL,1)));
    end
    offspringB(2,i)=atamaB(mod((kB-1),numel(atamaB))+1);
    offspringB(3,i)=istasyonB;
    offspringB(4,i)=0;

offspringB(5,i)=t(offspringB(1,i))*PL2(offspringB(1,i),offspringB(2,i));
    else
    offspringB(2,i)=atama2B(mod((kB-1),numel(atama2B))+1);
    offspringB(3,i)=istasyonB;

offspringB(4,i)=max(max(offspringB(5,find(offspringB(3,:)==istasyonB)).*PB(offspringB
(1,i),offspringB(1,find(offspringB(3,:)==istasyonB))))),max(offspringB(5,(find(offspringB(
3,:)==istasyonB,1)-
1+find(offspringB(2,find(offspringB(3,:)==istasyonB,1):i)==offspringB(2,i)))));

offspringB(5,i)=offspringB(4,i)+t(offspringB(1,i))*PL2(offspringB(1,i),offspringB(2,i));
    end
    end
    end
    end
    end
    arapop(:,:(2*b-1))= offspringA;
    arapop(:,:(2*b))= offspringB;
end
end

```


Ek 9. Seçilim ve İki Noktalı Çaprazlama Operatörü Kodlar (NSGA-II)

```
[ arapop ] = Crossover2( populasyon,caporan,PAND,POR,PL,PL2,t,WMAX )

%Seçilim
bsayisi=size(populasyon,3);
ParentA=populasyon(:,min(randi(bsayisi,2,ceil(bsayisi/2)))));
ParentB=populasyon(:,min(randi(bsayisi,2,ceil(bsayisi/2)))));
%Çaprazlama
ciftler=[randperm(ceil(bsayisi/2)) randperm(ceil(bsayisi/2))];
arapop=zeros(size(populasyon));
for b=1:round(bsayisi/2)
    adyanne = ParentA(:,ciftler(b,1));
    adaybaba = ParentB(:,ciftler(b,2));
    offspringA=adyanne;
    offspringB=adaybaba;
    capolasilik=unifrnd(0,1);
    if (capolasilik<caporan)
        gorevlerA=[adyanne(1,(adyanne(1,:)>0))
        find(sum(bsxfun(@eq,adyanne(1,(adyanne(1,:)>0))',1:size(adyanne,2)),1)==0)];
        sPORA=bsxfun(@gt,(1:size(POR,1))',1:size(POR,1)).*POR(gorevlerA,gorevlerA);
        eporA=zeros(size(POR,1),size(POR,2));
        eporA(gorevlerA,gorevlerA)=bsxfun(@(x,y) sPORA(x,y).*(1-
        sign(sum(sPORA(x,1:(y-1)),2))), (1:size(POR,1))',1:size(POR,1));
        gorevlerB=[adaybaba(1,(adaybaba(1,:)>0))
        find(sum(bsxfun(@eq,adaybaba(1,(adaybaba(1,:)>0))',1:size(adaybaba,2)),1)==0)];

        sPORB=bsxfun(@gt,(1:size(POR,1))',1:size(POR,1)).*POR(gorevlerB,gorevlerB);
        eporB=zeros(size(POR,1),size(POR,2));
        eporB(gorevlerB,gorevlerB)=bsxfun(@(x,y) sPORB(x,y).*(1-
        sign(sum(sPORB(x,1:(y-1)),2))), (1:size(POR,1))',1:size(POR,1));
        PA=sign(PAND+eporA);
        PB=sign(PAND+eporB);
        capist=randi(min(max(adyanne(3,:)),max(adaybaba(3,:))),1);
        gorevler=adyanne(1,find(adyanne(3,:)==capist));
        gorevler(find(sum(bsxfun(@eq,find((sum(POR,1)+sum(POR,2))>0)',gorevler),1)>0))=[];
        siralama=sum(bsxfun(@eq,gorevler',adaybaba(1,:)),1).*(1:size(adaybaba,2));
        siralama(find(siralama==0))=[];
        PAcap=zeros(size(PAND));

        PAcap(adaybaba(1,siralama),adaybaba(1,siralama))=bsxfun(@gt,(1:numel(siralama))',1:nu
        mel(siralama));
        PAK=sign(PA+PAcap);
        gorevler=adaybaba(1,find(adaybaba(3,:)==capist));
        gorevler(find(sum(bsxfun(@eq,find((sum(POR,1)+sum(POR,2))>0)',gorevler),1)>0))=[];
        siralama=sum(bsxfun(@eq,gorevler',adyanne(1,:)),1).*(1:size(adyanne,2));
        siralama(find(siralama==0))=[];
```

Ek 9'un devamı

```

PBcap=zeros(size(PAND));
PBcap(adayanne(1,siralama),adayanne(1,siralama))=bsxfun(@gt,(1:numel(siralama))',1:numel(siralama));
PBK=sign(PB+PBcap);
gorevlerA=sort(adayanne(1,find(adayanne(3,:)==capist)));
katalizorA=[];
gorevlerB=sort(adaybaba(1,find(adaybaba(3,:)==capist)));
katalizorB=[];
while numel(gorevlerA)>0 || numel(gorevlerB)>0
    if numel(gorevlerA)>0
        if numel(gorevlerA)>0 &&
numel(gorevlerA(find(sum(PAK(gorevlerA,gorevlerA),2)==0)))==0
            PAK=PA;
        end
        atanabilecek=gorevlerA(find(sum(PAK(gorevlerA,gorevlerA),2)==0));
        gen=atanabilecek(randi(numel(atanabilecek),1));
        katalizorA=[katalizorA gen];
        gorevlerA(find(gorevlerA==gen,1))=[];
    end
    if numel(gorevlerB)>0
        if numel(gorevlerB)>0 &&
numel(gorevlerB(find(sum(PBK(gorevlerB,gorevlerB),2)==0)))==0
            PBK=PB;
        end
        atanabilecek=gorevlerB(find(sum(PBK(gorevlerB,gorevlerB),2)==0));
        gen=atanabilecek(randi(numel(atanabilecek),1));
        katalizorB=[katalizorB gen];
        gorevlerB(find(gorevlerB==gen,1))=[];
    end
end
offspringA(1,find(adayanne(3,:)==capist))=katalizorA;
offspringB(1,find(adaybaba(3,:)==capist))=katalizorB;
offspringA(2:5,find(adayanne(3,:)==capist,1):end)=0;
offspringB(2:5,find(adaybaba(3,:)==capist,1):end)=0;
gorevlerA=[offspringA(1,(offspringA(1,:)>0))
find(sum(bsxfun(@eq,offspringA(1,(offspringA(1,:)>0))',1:size(offspringA,2)),1)==0)];
sPORA=bsxfun(@gt,(1:size(POR,1))',1:size(POR,1)).*POR(gorevlerA,gorevlerA);
eporA=zeros(size(POR,1),size(POR,2));
eporA(gorevlerA,gorevlerA)=bsxfun(@(x,y) sPORA(x,y).*(1-
sign(sum(sPORA(x,1:(y-1)),2))), (1:size(POR,1))',1:size(POR,1));
gorevlerB=[offspringB(1,(offspringB(1,:)>0))
find(sum(bsxfun(@eq,offspringB(1,(offspringB(1,:)>0))',1:size(offspringB,2)),1)==0)];
sPORB=bsxfun(@gt,(1:size(POR,1))',1:size(POR,1)).*POR(gorevlerB,gorevlerB);
eporB=zeros(size(POR,1),size(POR,2));
eporB(gorevlerB,gorevlerB)=bsxfun(@(x,y) sPORB(x,y).*(1-
sign(sum(sPORB(x,1:(y-1)),2))), (1:size(POR,1))',1:size(POR,1));
PA=sign(PAND+eporA);

```

Ek 9'un devamı

```

PB=sign(PAND+eporB);
atamaA=adaybaba(2,find(adaybaba(3,:)==capist));
wanne=find(sum(bsxfun(@eq,adayanne(2,:),1:size(PL,2)),1)>0);
wanne(find(sum(bsxfun(@eq,adayanne(2,find(adayanne(3,:)==capist))',wanne),1)>0))=[];
if numel(find(sum(bsxfun(@eq,wanne',atamaA),1)>0))>0 ||
numel(find(atamaA>size(PL,2)))>0
    aylaklar=find(sum(bsxfun(@eq,wanne',1:size(PL,2)),1)==0);
    if numel(find(sum(bsxfun(@eq,wanne',atamaA),1)>0))>0 &&
numel(find(atamaA>size(PL,2)))>0
        degistir=[find(sum(bsxfun(@eq,atamaA(find(sum(bsxfun(@eq,wanne',atamaA),1)>0
))',1:size(PL,2)),1)>0
size(PL,2)+find(sum(bsxfun(@eq,atamaA',(size(PL,2)+1):size(PL2,2)),1)>0)];
        elseif numel(find(sum(bsxfun(@eq,wanne',atamaA),1)>0))>0
            degistir=find(sum(bsxfun(@eq,atamaA(find(sum(bsxfun(@eq,wanne',atamaA),1)>0
))',1:size(PL,2)),1)>0);
            elseif numel(find(atamaA>size(PL,2)))>0
                degistir=size(PL,2)+find(sum(bsxfun(@eq,atamaA',(size(PL,2)+1):size(PL2,2)),1)>0
);
        end
        aylaklar=[aylaklar(randperm(numel(aylaklar))) (size(PL,2)+1):size(PL2,2)];
atamaA=((sum(bsxfun(@eq,degistir',atamaA),1)~=1).*atamaA)+aylaklar(1:numel(degistir
))*bsxfun(@eq,degistir',atamaA);
    end
    atamaB=adayanne(2,find(adayanne(3,:)==capist));
    wbaba=find(sum(bsxfun(@eq,adaybaba(2,:),1:size(PL,2)),1)>0);
wbaba(find(sum(bsxfun(@eq,adaybaba(2,find(adaybaba(3,:)==capist))',wbaba),1)>0))=[];
if numel(find(sum(bsxfun(@eq,wbaba',atamaB),1)>0))>0 ||
numel(find(atamaB>size(PL,2)))>0
    aylaklar=find(sum(bsxfun(@eq,wbaba',1:size(PL,2)),1)==0);
    if numel(find(sum(bsxfun(@eq,wbaba',atamaB),1)>0))>0 &&
numel(find(atamaB>size(PL,2)))>0
        degistir=[find(sum(bsxfun(@eq,atamaB(find(sum(bsxfun(@eq,wbaba',atamaB),1)>0
))',1:size(PL,2)),1)>0
size(PL,2)+find(sum(bsxfun(@eq,atamaB',(size(PL,2)+1):size(PL2,2)),1)>0)];
        elseif numel(find(sum(bsxfun(@eq,wbaba',atamaB),1)>0))>0
            degistir=find(sum(bsxfun(@eq,atamaB(find(sum(bsxfun(@eq,wbaba',atamaB),1)>0
))',1:size(PL,2)),1)>0);
            elseif numel(find(atamaB>size(PL,2)))>0
                degistir=size(PL,2)+find(sum(bsxfun(@eq,atamaB',(size(PL,2)+1):size(PL2,2)),1)>0
);
        end
        aylaklar=[aylaklar(randperm(numel(aylaklar))) (size(PL,2)+1):size(PL2,2)];
atamaB=((sum(bsxfun(@eq,degistir',atamaB),1)~=1).*atamaB)+aylaklar(1:numel(degistir
))*bsxfun(@eq,degistir',atamaB);
    end
    CTUA=max(ceil(max(t)*max(max(PL,[],2))),max(adayanne(5,:)));
    CTUB=max(ceil(max(t)*max(max(PL,[],2))),max(adaybaba(5,:)));

```

Ek 9'un devamı

```

    istasyonA=capist;
    istasyonB=capist;
    kA=0;
    kB=0;
    for
i=min(find(adayanne(3,:)==capist,1),find(adaybaba(3,:)==capist,1)):size(populasyon,2)
        if offspringA(1,i)==0 && offspringB(1,i)==0
            break
        end
        if offspringA(1,i)>0 && offspringA(2,i)==0
            kA=kA+1;
            offspringA(2,i)=atamaA(mod((kA-1),numel(atamaA))+1);
            offspringA(3,i)=istasyonA;
            offspringA(4,i)=max(max(offspringA(5,find(offspringA(3,:)==istasyonA)).*PA(offspring
            A(1,i),offspringA(1,find(offspringA(3,:)==istasyonA))))),max(offspringA(5,(find(offspring
            A(3,:)==istasyonA,1)-
            1+find(offspringA(2,find(offspringA(3,:)==istasyonA,1):i)==offspringA(2,i)))));
            offspringA(5,i)=offspringA(4,i)+t(offspringA(1,i))*PL2(offspringA(1,i),offspringA(2,i));
            while offspringA(5,i)>CTUA
                atamaA(find(offspringA(2,i)==atamaA))=[];
                offspringA(2:5,i)=0;
                if numel(atamaA)==0
                    istasyonA=istasyonA+1;
                    kA=1;
                    if numel(find(adayanne(3,:)==istasyonA))>0
                        atamaA=adayanne(2,find(adayanne(3,:)==istasyonA));
                        if numel(find(atamaA>size(PL,2)))>0 &&
                        numel(find(sum(bsxfun(@eq,[adayanne(2,:) offspringA(2,:)]',1:size(PL,2)),1)==0))>0
                            aylaklar=find(sum(bsxfun(@eq,[adayanne(2,)
                            offspringA(2,:)]',1:size(PL,2)),1)==0);
                            aylaklar=[aylaklar(randperm(numel(aylaklar)) (size(PL,2)+1):size(PL2,2))];
                            degistir=size(PL,2)+find(sum(bsxfun(@eq,atamaA',(size(PL,2)+1):size(PL2,2)),1)>0);
                            atamaA=((sum(bsxfun(@eq,degistir',atamaA),1)~1).*atamaA)+aylaklar(1:numel(degistir
                            ))*bsxfun(@eq,degistir',atamaA);
                        end
                    elseif
                        numel(find(sum(bsxfun(@eq,offspringA(2,:)]',1:size(PL,2)),1)==0))>0
                            calisanlar=find(sum(bsxfun(@eq,offspringA(2,:)]',1:size(PL,2)),1)==0);
                            calisanlar=[calisanlar(randperm(numel(calisanlar))) size(PL,2)+1:size(PL2,2)];
                            worker=calisanlar(1:WMAX);
                            atamaA=worker(randi(WMAX,1,size(PL,1)));
                        else
                            worker=(size(PL,2)+1):size(PL2,2);
                            atamaA=worker(randi(WMAX,1,size(PL,1)));
                        end
                    offspringA(2,i)=atamaA(mod((kA-1),numel(atamaA))+1);
                    offspringA(3,i)=istasyonA;

```

Ek 9'un devamı

```

    offspringA(4,i)=0;

offspringA(5,i)=t(offspringA(1,i))*PL2(offspringA(1,i),offspringA(2,i));
    else
        offspringA(2,i)=atamaA(mod((kA-1),numel(atamaA))+1);
        offspringA(3,i)=istasyonA;
offspringA(4,i)=max(max(offspringA(5,find(offspringA(3,:)==istasyonA)).*PA(offspring
A(1,i),offspringA(1,find(offspringA(3,:)==istasyonA))))),max(offspringA(5,(find(offspring
A(3,:)==istasyonA,1)-
1+find(offspringA(2,find(offspringA(3,:)==istasyonA,1):i)==offspringA(2,i)))));
offspringA(5,i)=offspringA(4,i)+t(offspringA(1,i))*PL2(offspringA(1,i),offspringA(2,i));
    end
end
end
    if offspringB(1,i)>0 && offspringB(2,i)==0
        kB=kB+1;
        offspringB(2,i)=atamaB(mod((kB-1),numel(atamaB))+1);
        offspringB(3,i)=istasyonB;
offspringB(4,i)=max(max(offspringB(5,find(offspringB(3,:)==istasyonB)).*PB(offspringB
(1,i),offspringB(1,find(offspringB(3,:)==istasyonB))))),max(offspringB(5,(find(offspringB(
3,:)==istasyonB,1)-
1+find(offspringB(2,find(offspringB(3,:)==istasyonB,1):i)==offspringB(2,i)))));
offspringB(5,i)=offspringB(4,i)+t(offspringB(1,i))*PL2(offspringB(1,i),offspringB(2,i));
        while offspringB(5,i)>CTUB
            atamaB(find(offspringB(2,i)==atamaB))=[];
            offspringB(2:5,i)=0;
            if numel(atamaB)==0
                istasyonB=istasyonB+1;
                kB=1;
                if numel(find(adaybaba(3,:)==istasyonB))>0
                    atamaB=adaybaba(2,find(adaybaba(3,:)==istasyonB));
                    if numel(find(atamaB>size(PL,2)))>0 &&
numel(find(sum(bsxfun(@eq,[adaybaba(2,:) offspringB(2,:)]',1:size(PL,2)),1)==0))>0
                        aylaklar=find(sum(bsxfun(@eq,[adaybaba(2,:)
offspringB(2,:)]',1:size(PL,2)),1)==0);
                        aylaklar=[aylaklar(randperm(numel(aylaklar))) (size(PL,2)+1):size(PL2,2)];

degistir=size(PL,2)+find(sum(bsxfun(@eq,atamaB',(size(PL,2)+1):size(PL2,2)),1)>0);
atamaB=((sum(bsxfun(@eq,degistir,atamaB),1)~=1).*atamaB)+aylaklar(1:numel(degistir)
).*bsxfun(@eq,degistir,atamaB);
            end
        elseif
numel(find(sum(bsxfun(@eq,offspringB(2,:)]',1:size(PL,2)),1)==0))>0
            calisanlar=find(sum(bsxfun(@eq,offspringB(2,:)]',1:size(PL,2)),1)==0);
            calisanlar=[calisanlar(randperm(numel(calisanlar))) size(PL,2)+1:size(PL2,2)];
            worker=calisanlar(1:WMAX);
            atamaB=worker(randi(WMAX,1,size(PL,1)));

```

Ek 9'un devamı

```

else
    worker=(size(PL,2)+1):size(PL2,2);
    atamaB=worker(randi(WMAX,1,size(PL,1)));
end
offspringB(2,i)=atamaB(mod((kB-1),numel(atamaB))+1);
offspringB(3,i)=istasyonB;
offspringB(4,i)=0;

offspringB(5,i)=t(offspringB(1,i))*PL2(offspringB(1,i),offspringB(2,i));
else
    offspringB(2,i)=atamaB(mod((kB-1),numel(atamaB))+1);
    offspringB(3,i)=istasyonB;
offspringB(4,i)=max(max(offspringB(5,find(offspringB(3,:)==istasyonB)).*PB(offspringB(1,i),offspringB(1,find(offspringB(3,:)==istasyonB))))),max(offspringB(5,(find(offspringB(3,:)==istasyonB,1)-1+find(offspringB(2,find(offspringB(3,:)==istasyonB,1):i)==offspringB(2,i)))));
offspringB(5,i)=offspringB(4,i)+t(offspringB(1,i))*PL2(offspringB(1,i),offspringB(2,i));
end
end
end
end
arapop(:,:(2*b-1))= offspringA;
arapop(:,:(2*b))= offspringB;
end

end

```

Ek 10. Mutasyon Operatörü Kodlar (NSGA-II)

```

[ arapop ] = Mutation( arapop,mutoran,PAND,POR,PL,PL2,t,WMAX)
bsayisi=size(arapop,3);
for b=1:bsayisi
    mutolasilik=unifrnd(0,1);
    if (mutolasilik<mutoran)
        birey=arapop(:,b);
        gorevler=[birey(1,(birey(1,:)>0))
find(sum(bsxfun(@eq,birey(1,(birey(1,:)>0))',1:size(birey,2)),1)==0)];
sPOR=bsxfun(@gt,(1:size(POR,1))',1:size(POR,1)).*POR(gorevler,gorevler);
epor=zeros(size(POR));
epor(gorevler,gorevler)=bsxfun(@(x,y) sPOR(x,y).*(1-sign(sum(sPOR(x,1:(y-
1)),2))),1:size(POR,1))',1:size(POR,1));
P=sign(PAND+epor);
CTU=max(ceil(max(t)*max(max(PL,[],2))),max(birey(5,:)));
capist=randi(max(birey(3,:),1),1);
gorevler=sort(birey(1,find(birey(3,:)==capist)));
while numel(gorevler)>0
    atanabilecek=gorevler(find(sum(P(gorevler,gorevler),2)==0));
    gen=atanabilecek(randi(numel(atanabilecek),1));
    katalizor=[katalizor gen];
    gorevler(find(gorevler==gen,1))=[];
end
birey(1,find(birey(3,:)==capist))=katalizor;
calisan=birey(2:3,find(birey(3,:)>0));
worker=find(sum(bsxfun(@eq,calisan(1,:),1:size(PL,2)),1)>0);
worker=worker(randperm(numel(worker)));
dummy=find(calisan(1,:)==worker(2));
calisan(1,find(calisan(1,:)==worker(1)))=worker(2);
calisan(1,dummy)=worker(1);
birey(2:5,:)=0;
gorevler=[birey(1,(birey(1,:)>0))
find(sum(bsxfun(@eq,birey(1,(birey(1,:)>0))',1:size(birey,2)),1)==0)];
sPOR=bsxfun(@gt,(1:size(POR,1))',1:size(POR,1)).*POR(gorevler,gorevler);
epor=zeros(size(POR));
epor(gorevler,gorevler)=bsxfun(@(x,y) sPOR(x,y).*(1-sign(sum(sPOR(x,1:(y-
1)),2))),1:size(POR,1))',1:size(POR,1));
P=sign(PAND+epor);
istasyon=1;
atama=calisan(1,find(calisan(2,:)==istasyon));
k=0;
for i=1:numel(find(birey(1,:)>0))
    k=k+1;
    birey(2,i)=atama(mod((k-1),numel(atama))+1);
    birey(3,i)=istasyon;

```

Ek 10'un devamı

```

    birey(4,i)=max(max(birey(5,find(birey(3,:)==istasyon)).*P(birey(1,i),birey(1,find(birey(3,:)==istasyon))))),max(birey(5,(find(birey(3,:)==istasyon,1)-1+find(birey(2,find(birey(3,:)==istasyon,1):i)==birey(2,i)))));
    birey(5,i)=birey(4,i)+t(birey(1,i))*PL2(birey(1,i),birey(2,i));
        atama2=atama;
        while birey(5,i)>CTU
            atama2(find(birey(2,i)==atama2))=[];
            birey(2:5,i)=0;
            if numel(atama2)==0
                istasyon=istasyon+1;
                k=1;
                if numel(find(calisan(2,:)==istasyon))>0
                    atama=calisan(1,find(calisan(2,:)==istasyon));
                else
                    calisanlar=find(sum(bsxfun(@eq,birey(2,:)',1:size(PL,2)),1)==0);
                    calisanlar=[calisanlar(randperm(numel(calisanlar))) size(PL,2)+1:size(PL2,2)];
                    worker=calisanlar(1:WMAX);
                    atama=worker(randi(WMAX,1,size(PL,1)));
                end
                birey(2,i)=atama(mod((k-1),numel(atama))+1);
                birey(3,i)=istasyon;
                birey(4,i)=0;
                birey(5,i)=t(birey(1,i))*PL2(birey(1,i),birey(2,i));
            else
                birey(2,i)=atama2(mod((k-1),numel(atama2))+1);
                birey(3,i)=istasyon;
            end
            birey(4,i)=max(max(birey(5,find(birey(3,:)==istasyon)).*P(birey(1,i),birey(1,find(birey(3,:)==istasyon))))),max(birey(5,(find(birey(3,:)==istasyon,1)-1+find(birey(2,find(birey(3,:)==istasyon,1):i)==birey(2,i)))));
            birey(5,i)=birey(4,i)+t(birey(1,i))*PL2(birey(1,i),birey(2,i));
        end
    end
end
    arapop(:,b)=birey;
end
end
end

```


Ek 11. Parametre Optimizasyonu

Ek Tablo 2. Tüm tekrarlar için parametre optimizasyonu değerleri

Görev Sayısı	1. Koşum				2. Koşum				3. Koşum				4. Koşum				5. Koşum			
	<i>ps</i>	<i>cp</i>	<i>mp</i>	<i>mg</i>	$DI_R(\text{NCR2SC3})$	<i>mg</i>	$DI_R(\text{NCR2SC3})$	<i>mg</i>	$DI_R(\text{NCR2SC3})$	<i>mg</i>	$DI_R(\text{NCR2SC3})$	<i>mg</i>	$DI_R(\text{NCR2SC3})$	<i>mg</i>	$DI_R(\text{NCR2SC3})$	<i>mg</i>	$DI_R(\text{NCR2SC3})$			
$p \leq 30$	100	0.7	0.01	243	1.00	254	0.79	306	0.64	135	0.83	111	1.11							
	100	0.7	0.05	169	0.79	259	0.66	356	0.75	304	0.62	112	0.73							
	100	0.7	0.1	126	0.99	237	0.65	137	0.55	386	0.73	166	0.79							
	100	0.8	0.01	160	1.08	192	0.65	164	1.20	61	1.00	84	1.34							
	100	0.8	0.05	148	0.78	135	0.59	284	1.02	306	0.65	115	0.90							
	100	0.8	0.1	99	1.13	167	0.75	122	0.64	152	1.26	180	0.58							
	100	0.9	0.01	146	1.51	140	0.63	61	1.17	65	1.22	110	0.81							
	100	0.9	0.05	173	1.00	127	0.81	224	0.90	321	0.85	110	0.80							
	100	0.9	0.1	108	0.85	268	0.57	116	1.11	241	0.65	315	0.52							
$30 < p \leq 60$	150	0.7	0.01	88	0.63	91	0.80	272	0.71	267	0.70	110	0.80							
	150	0.7	0.05	149	0.46	331	0.50	136	0.51	138	0.91	125	0.52							
	150	0.7	0.1	325	0.37	474	0.54	361	0.48	254	0.38	190	0.60							
	150	0.8	0.01	99	0.66	83	0.63	232	0.72	245	0.46	83	0.74							
	150	0.8	0.05	312	0.57	117	0.53	86	0.75	311	0.40	100	0.59							
	150	0.8	0.1	198	0.58	228	0.68	97	0.77	338	0.40	97	0.72							
	150	0.9	0.01	90	0.69	83	0.69	119	0.96	203	0.66	180	0.74							
	150	0.9	0.05	161	0.75	111	0.69	223	0.62	182	0.63	79	0.69							
	150	0.9	0.1	220	0.36	150	0.83	184	0.49	174	0.58	266	0.51							

Ek 12. Çaprazlama Operatörleri İçin Karşılaştırmalı Sonuçlar

Ek Tablo 3. Senaryo 1 için çaprazlama operatörlerinin karşılaştırmalı sonuçları

		NCR1SC1/ NCR2SC1								
		<i>C</i> (CR1,CR2)	<i>C</i> (CR2,CR1)	<i>DIR</i> (CR1)	<i>DIR</i> (CR2)	<i>OS</i> (CR1,CR2)	<i>Nnd</i> (CR1)	<i>Nnd</i> (CR2)	<i>CPU</i> (CR1)	<i>CPU</i> (CR2)
[0.85-1.15]	P1	0.77	0.37	0.43	0.44	0.88	8.40	8.92	17.83	15.23
	P2	0.64	0.09	0.53	0.53	0.96	27.76	28.76	22.74	17.51
	P3	0.28	0.32	0.64	0.64	1.14	46.96	41.28	25.10	18.04
	P4	0.21	0.33	0.71	0.71	1.11	63.72	47.40	97.29	71.23
	P5	0.08	0.07	0.75	0.74	0.99	89.24	88.36	107.37	74.24
	P6	0.09	0.08	0.84	0.83	0.96	92.48	94.88	181.89	124.62
[0.90-1.10]	P1	0.68	0.41	0.44	0.44	1.13	9.68	8.52	17.77	15.04
	P2	0.63	0.12	0.54	0.55	0.71	22.32	26.04	23.99	20.46
	P3	0.45	0.18	0.64	0.65	1.21	41.20	29.52	24.13	17.67
	P4	0.36	0.22	0.72	0.72	1.15	52.00	36.24	69.74	51.40
	P5	0.09	0.06	0.77	0.77	1.09	87.84	72.40	100.49	73.70
	P6	0.11	0.06	0.85	0.85	0.96	94.24	90.28	185.49	126.16
[1]	P1	0.79	0.39	0.44	0.44	1.06	8.96	8.24	8.34	7.13
	P2	0.69	0.10	0.54	0.55	0.92	26.44	21.56	26.29	20.48
	P3	0.36	0.21	0.64	0.65	1.07	41.80	28.40	33.47	24.59
	P4	0.40	0.20	0.72	0.73	1.36	56.72	34.40	78.63	58.39
	P5	0.10	0.07	0.77	0.77	1.16	88.76	67.32	122.72	81.49
	P6	0.12	0.06	0.85	0.85	1.01	95.96	90.68	189.65	129.65

Ek 12'nin devamı

Ek Tablo 4. Senaryo 2 için çaprazlama operatörlerinin karşılaştırmalı sonuçları

		NCR1SC2/ NCR2SC2								
		C (CR1,CR2)	C (CR2,CR1)	D1R (CR1)	D1R (CR2)	OS (CR1,CR2)	Nnd (CR1)	Nnd (CR2)	CPU (CR1)	CPU (CR2)
[0.85-1.15]	P1	0.72	0.35	0.43	0.44	0.78	6.32	7.24	17.68	15.18
	P2	0.50	0.20	0.53	0.54	0.92	23.40	24.40	22.47	17.39
	P3	0.32	0.27	0.64	0.64	0.94	37.56	33.16	24.22	18.13
	P4	0.24	0.35	0.71	0.71	1.32	53.88	44.48	95.38	71.27
	P5	0.04	0.08	0.75	0.74	0.98	85.84	85.44	107.37	74.01
	P6	0.10	0.07	0.84	0.83	1.07	89.84	93.16	179.65	123.93
[0.90-1.10]	P1	0.67	0.38	0.45	0.45	68.32	6.20	6.92	17.72	15.09
	P2	0.56	0.19	0.55	0.55	0.94	20.32	19.96	24.80	20.17
	P3	0.43	0.17	0.65	0.65	5.53	31.00	22.80	23.71	17.42
	P4	0.31	0.20	0.72	0.73	1.39	47.32	32.40	71.25	53.32
	P5	0.09	0.07	0.77	0.77	1.03	82.08	63.80	100.63	74.15
	P6	0.12	0.08	0.85	0.85	0.93	90.72	89.76	182.26	126.74
[1]	P1	0.66	0.40	0.45	0.45	0.79	5.96	6.76	8.23	6.69
	P2	0.56	0.15	0.55	0.55	1.17	19.40	20.12	26.08	20.43
	P3	0.44	0.23	0.65	0.65	1.02	34.52	24.20	33.41	23.31
	P4	0.24	0.24	0.72	0.73	1.33	51.52	32.24	81.78	57.08
	P5	0.08	0.08	0.77	0.77	1.07	80.72	64.16	120.46	83.50
	P6	0.10	0.05	0.86	0.85	1.01	91.84	89.32	182.95	131.92

Ek 12'nin devamı

Ek Tablo 5. Senaryo 3 için çaprazlama operatörlerinin karşılaştırmalı sonuçları

		NCR1SC3/ NCR2SC3								
		C (CR1,CR2)	C (CR2,CR1)	D1R (CR1)	D1R (CR2)	OS (CR1,CR2)	Nnd (CR1)	Nnd (CR2)	CPU (CR1)	CPU (CR2)
[0.85-1.15]	P1	0.52	0.42	0.41	0.41	1.01	19.36	20.40	18.99	16.82
	P2	0.29	0.32	0.44	0.45	0.88	28.16	35.80	23.77	19.75
	P3	0.09	0.65	0.54	0.53	0.97	26.96	43.96	26.04	20.73
	P4	0.08	0.67	0.59	0.58	1.12	47.00	66.68	99.73	79.70
	P5	0.06	0.12	0.62	0.60	1.05	69.80	92.84	111.38	81.51
	P6	0.03	0.10	0.67	0.65	0.95	71.96	95.80	186.28	139.29
[0.90-1.10]	P1	0.45	0.49	0.42	0.42	1.06	17.48	19.40	18.63	16.66
	P2	0.35	0.35	0.46	0.46	0.95	29.00	33.88	25.43	23.36
	P3	0.11	0.60	0.54	0.53	1.02	30.68	42.56	24.95	20.34
	P4	0.15	0.62	0.60	0.59	1.06	45.52	66.84	69.95	58.90
	P5	0.06	0.10	0.64	0.63	1.03	74.00	93.60	104.30	81.21
	P6	0.04	0.09	0.68	0.66	1.13	73.00	98.36	185.04	141.77
[1]	P1	0.49	0.47	0.42	0.42	1.23	19.44	21.00	8.71	7.56
	P2	0.28	0.40	0.46	0.46	0.99	28.32	34.00	27.44	23.61
	P3	0.19	0.54	0.54	0.54	0.79	29.36	46.40	34.90	196.92
	P4	0.15	0.61	0.60	0.59	0.98	50.12	64.04	80.69	62.26
	P5	0.07	0.11	0.64	0.63	1.13	75.60	96.00	126.93	89.55
	P6	0.05	0.11	0.68	0.66	1.02	71.52	97.40	188.40	144.07

Ek 12'nin devamı

Ek Tablo 6. Senaryo 4 için çaprazlama operatörlerinin karşılaştırmalı sonuçları

		NCR1SC4/ NCR2SC4								
		C (CR1,CR2)	C (CR2,CR1)	D1R (CR1)	D1R (CR2)	OS (CR1,CR2)	Nnd (CR1)	Nnd (CR2)	CPU (CR1)	CPU (CR2)
[0.85-1.15]	P1	0.37	0.50	0.41	0.41	0.98	16.20	15.64	18.44	16.60
	P2	0.17	0.51	0.45	0.45	2.67	24.12	31.28	23.49	20.23
	P3	0.15	0.52	0.54	0.53	4.30	22.12	41.16	25.39	20.37
	P4	0.12	0.62	0.59	0.58	1.24	42.88	59.48	99.26	79.28
	P5	0.04	0.08	0.63	0.61	1.12	63.32	86.80	107.93	80.74
	P6	0.03	0.08	0.68	0.66	1.16	70.56	90.76	183.84	139.61
[0.90-1.10]	P1	0.51	0.50	0.42	0.42	8.13	15.00	17.40	18.53	16.45
	P2	0.27	0.44	0.46	0.47	1.27	23.60	29.92	26.21	21.94
	P3	0.10	0.56	0.55	0.54	2.83	25.40	35.40	24.92	20.03
	P4	0.09	0.61	0.60	0.59	1.23	43.52	54.04	72.35	58.28
	P5	0.07	0.10	0.65	0.63	0.95	64.84	88.16	101.85	78.93
	P6	0.04	0.08	0.68	0.66	1.02	65.88	94.44	183.98	141.09
[1]	P1	0.46	0.42	0.42	0.42	1.11	14.60	15.32	8.66	7.47
	P2	0.23	0.47	0.46	0.47	3.07	23.32	26.44	27.13	23.27
	P3	0.14	0.53	0.55	0.54	6.92	24.40	37.00	35.02	27.58
	P4	0.11	0.67	0.60	0.59	4.69	42.68	54.36	82.53	63.07
	P5	0.07	0.09	0.64	0.63	1.16	63.16	88.44	116.62	91.87
	P6	0.04	0.09	0.68	0.66	1.05	65.56	94.52	187.89	145.91

Ek 13. Stratejiler İçin Karşılaştırmalı Sonuçlar

Ek Tablo 7. Tek noktalı çaprazlama operatörüne göre teorik stratejilerin karşılaştırmalı sonuçları

		NCR1SC1/ NCR1SC3								
		<i>C</i> (SC1,SC3)	<i>C</i> (SC3,SC1)	<i>DIR</i> (SC1)	<i>DIR</i> (SC3)	<i>OS</i> (SC1,SC3)	<i>Nnd</i> (SC1)	<i>Nnd</i> (SC3)	CPU (SC1)	CPU (SC3)
[0.85-1.15]	P1	0.05	0.42	0.43	0.41	0.59	8.40	19.36	17.83	18.99
	P2	0.00	0.43	0.53	0.44	1.57	27.76	28.16	22.74	23.77
	P3	0.00	0.23	0.64	0.54	1.89	46.96	26.96	25.10	26.04
	P4	0.00	0.61	0.71	0.59	1.88	63.72	47.00	97.29	99.73
	P5	0.00	0.08	0.75	0.62	1.59	89.24	69.80	107.37	111.38
	P6	0.00	0.07	0.84	0.67	1.61	92.48	71.96	181.89	186.28
[0.90-1.10]	P1	0.04	0.31	0.44	0.42	0.68	9.68	17.48	17.77	18.63
	P2	0.00	0.30	0.54	0.46	1.24	22.32	29.00	23.99	25.43
	P3	0.00	0.26	0.64	0.54	2.09	41.20	30.68	24.13	24.95
	P4	0.00	0.62	0.72	0.60	1.84	52.00	45.52	69.74	69.95
	P5	0.00	0.09	0.77	0.64	1.31	87.84	74.00	100.49	104.30
	P6	0.00	0.08	0.85	0.68	1.71	94.24	73.00	185.49	185.04
[1]	P1	0.03	0.32	0.44	0.42	0.67	8.96	19.44	8.34	8.71
	P2	0.00	0.28	0.54	0.46	1.61	26.44	28.32	26.29	27.44
	P3	0.00	0.27	0.64	0.54	1.82	41.80	29.36	33.47	34.90
	P4	0.00	0.62	0.72	0.60	2.02	56.72	50.12	78.63	80.69
	P5	0.00	0.10	0.77	0.64	1.53	88.76	75.60	122.72	126.93
	P6	0.00	0.08	0.85	0.68	1.81	95.96	71.52	189.65	188.40

Ek 13'ün devamı

Ek Tablo 8. Tek noktalı çaprazlama operatörüne göre pratik stratejilerin karşılaştırmalı sonuçları

		NCR1SC2/ NCR1SC4								
		<i>C</i> (SC2,SC4)	<i>C</i> (SC4,SC2)	<i>DIR</i> (SC2)	<i>DIR</i> (SC4)	<i>OS</i> (SC2,SC4)	<i>Nnd</i> (SC2)	<i>Nnd</i> (SC4)	<i>CPU</i> (SC2)	<i>CPU</i> (SC4)
[0.85-1.15]	P1	0.05	0.42	0.43	0.41	0.59	8.40	19.36	17.83	18.99
	P2	0.00	0.43	0.53	0.44	1.57	27.76	28.16	22.74	23.77
	P3	0.00	0.23	0.64	0.54	1.89	46.96	26.96	25.10	26.04
	P4	0.00	0.61	0.71	0.59	1.88	63.72	47.00	97.29	99.73
	P5	0.00	0.08	0.75	0.62	1.59	89.24	69.80	107.37	111.38
	P6	0.00	0.07	0.84	0.67	1.61	92.48	71.96	181.89	186.28
[0.90-1.10]	P1	0.04	0.31	0.44	0.42	0.68	9.68	17.48	17.77	18.63
	P2	0.00	0.30	0.54	0.46	1.24	22.32	29.00	23.99	25.43
	P3	0.00	0.26	0.64	0.54	2.09	41.20	30.68	24.13	24.95
	P4	0.00	0.62	0.72	0.60	1.84	52.00	45.52	69.74	69.95
	P5	0.00	0.09	0.77	0.64	1.31	87.84	74.00	100.49	104.30
	P6	0.00	0.08	0.85	0.68	1.71	94.24	73.00	185.49	185.04
[1]	P1	0.03	0.32	0.44	0.42	0.67	8.96	19.44	8.34	8.71
	P2	0.00	0.28	0.54	0.46	1.61	26.44	28.32	26.29	27.44
	P3	0.00	0.27	0.64	0.54	1.82	41.80	29.36	33.47	34.90
	P4	0.00	0.62	0.72	0.60	2.02	56.72	50.12	78.63	80.69
	P5	0.00	0.10	0.77	0.64	1.53	88.76	75.60	122.72	126.93
	P6	0.00	0.08	0.85	0.68	1.81	95.96	71.52	189.65	188.40

ÖZGEÇMİŞ

Büşra YAZICI, 1994 yılında Trabzon’da doğdu. Lise öğrenimini 2012 yılında Ordu Fen Lisesi’nde tamamladı. 2017 yılında Karadeniz Teknik Üniversitesi, Endüstri Mühendisliği bölümünden, 2018 yılında Anadolu Üniversitesi, Bankacılık ve Sigortacılık bölümünden mezun oldu. 2020 yılında Karadeniz Teknik Üniversitesi, Endüstri Mühendisliği Anabilim Dalı’nda tezli yüksek lisans eğitimini tamamladı. 2020 yılından itibaren Trabzon Ulaşım A.Ş. ‘nde çalışan Büşra Yazıcı, iyi derecede İngilizce bilmektedir.

AKADEMİK YAYINLAR

Ay H., Yavuz T., Yazıcı B., Yeni F. B., Baban P., 2017. Resource Planning in Small Medium Enterprises (Smes): Application in a Factory, Computational and Statistical Methods in Applied Sciences, COSTAS'2017, Samsun, Türkiye, 9 – 11 Kasım 2017, 14-15.