T.C.
GEBZE TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

SIRA BAĞIMLI HAZIRLIK SÜRESİ VE MAKİNE KISITLARI İLE İLİŞKİSİZ PARALEL MAKİNE ÇİZELGELEME PROBLEMİNE DEĞİŞKEN KOMŞULUK ARAMASI YAKLAŞIMI

EMİNE DUYSAK

YÜKSEK LİSANS TEZİ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

DANIŞMAN: DR. ÖĞR. ÜYESİ A. BURAK PAÇ

TEMMUZ 2024

T.C.
GEBZE TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

SIRA BAĞIMLI HAZIRLIK SÜRESİ VE MAKİNE KISITLARI İLE İLİŞKİSİZ PARALEL MAKİNE ÇİZELGELEME PROBLEMİNE DEĞİŞKEN KOMŞULUK ARAMASI YAKLAŞIMI

EMİNE DUYSAK

YÜKSEK LİSANS TEZİ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

DANIŞMAN: DR. ÖĞR. ÜYESİ A. BURAK PAÇ

TEMMUZ 2024

T.R.

GEBZE TECHNICAL UNIVERSITY

GRADUATE SCHOOL

VARIABLE NEIGHBORHOOD SEARCH APPROACH ON THE UNRELATED PARALLEL MACHINE SCHEDULING PROBLEM WITH SEQUENCE-DEPENDENT SETUP TIMES AND MACHINE CONSTRAINTS

EMİNE DUYSAK

A THESIS OF MASTER OF SCIENCE

DEPARTMENT OF INDUSTRIAL ENGINEERING

ADVISOR: DR. ASSIST. PROF. A. BURAK PAÇ

JULY 2024

YÜKSEK LİSANS JÜRİ ONAY FORMU

GTÜ Lisansüstü Eğitim Enstitüsü Yönetim Kurulunun 01/07/2024 tarih ve 2024/33 sayılı kararıyla oluşturulan jüri tarafından 16/07/2024 tarihinde tez savunma sınavı yapılan Emine Duysak'ın tez çalışması Endüstri Mühendisliği Anabilim Dalında YÜKSEK LİSANS tezi olarak kabul edilmiştir.

JÜRİ

ÜYE
(TEZ DANIŞMANI) : Dr. Öğr. Üyesi A. Burak Paç

ÜYE : Dr. Öğr. Üyesi Suat Genç

ÜYE : Dr. Öğr. Üyesi Amine Gizem Tiniç

ONAY

Gebze Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü Yönetim Kurulunun

……/……./...…. tarih ve …..…/…..... sayılı kararı.

İMZA/MÜHÜR

ÖZET

Bu çalışmada, sıra bağımlı hazırlık süreli ilişkisiz paralel makine çizelgeleme problemi ele alınmıştır. İşlerin birden fazla ilişkisiz paralel makinelerde işlenebileceği ve her işin işlenmeye başlamadan önce bir hazırlık süresine ihtiyaç duyduğu bir üretim ortamı modellenmektedir. Hazırlık süresi, makineye ve işlerin sırasına bağlı olarak değişkenlik göstermektedir. Problem, makinelerin planlı duruşlar nedeniyle belirli bir süre boyunca kullanılamayacağı varsayımı altında incelenmektedir. Çalışmanın amacı, toplam gecikme ve erken teslim sürelerini en aza indirmektir.

NP-zor olarak sınıflandırılan problemin uzun çözüm süreleri ve kesin çözüme ulaşmanın zorluğu nedeniyle problemin çözümünde metasezgisel algoritmalar tercih edilmiştir. Çalışmada ele alınan sıra bağımlı hazırlık süreli ilişkisiz paralel makine çizelgeleme problemi için yeni bir Değişken Komşuluk Arama (DKA) algoritması önerilmiştir. Önerilen algoritma, çeşitli başlangıç çözümlerden başlayarak çözüm uzayının farklı alanlarını keşfetmek üzere tasarlanmış üç farklı komşuluk yapısı ve Değişken Komşuluk İniş (DKİ) algoritmasını kullanılmaktadır. Komşuluk dereceleri ile giderek büyüyen bir çözüm uzayı keşfedilmekte olup, mevcut duruma göre adaptif bir şekilde komşuluk yapısı seçimi yapılmaktadır. Bu yaklaşım, algoritmanın arama verimliliğini artırmayı ve global optimuma ulaşma süresini hızlandırmayı amaçlamaktadır.

Sentetik veri oluşturma algoritmasıyla üretilen ve gerçek dünya verilerini iyi bir şekilde temsil eden test problemleri kullanılarak, önerilen DKA algoritması matematiksel programlama çözümleri ile karşılaştırıldı. Elde edilen sonuçlar, önerilen algoritmanın problemin çeşitli boyutları ve karmaşıklık düzeyleri için matematiksel programlama çözümlerine kıyasla çözüm süresi ve en iyi çözüme yakınsama yeteneğine göre daha etkin ve verimli bir yöntem olduğunu göstermektedir.

Anahtar Kelimeler: İlişkisiz Paralel Makine Çizelgeleme, Sıra Bağımlı Hazırlık Süreli, Değişken Komşuluk Arama, Metasezgisel

ABSTRACT

In this study, the unrelated parallel machine scheduling problem with sequence-dependent setup times is addressed. A production environment is modeled where jobs can be processed on multiple unrelated parallel machines, and each job requires a setup time before processing. The setup time varies depending on the machine and the job sequence. The problem is examined under the assumption that machines may be unavailable for certain periods due to scheduled downtimes. The objective of the study is to minimize total tardiness and early completion times.

Due to the NP-hard classification of the problem, which results in prolonged solution times and the difficulty of achieving exact solutions, metaheuristic algorithms have been employed. This study proposes a new Variable Neighborhood Search (VNS) algorithm for the unrelated parallel machine scheduling problem with sequence-dependent setup times. The proposed algorithm leverages three distinct neighborhood structures and the Variable Neighborhood Descent (VND) algorithm, designed to explore different regions of the solution space from various initial solutions. The neighborhood degrees facilitate the exploration of an increasingly expansive solution space, while the adaptive selection of the neighborhood structure is performed based on the current state of the solution. This approach aims to improve the efficiency of the search process and expedite the attainment of the global optimum.

The proposed VNS algorithm was compared with mathematical programming solutions using test problems generated by a synthetic data generation algorithm, which well represents real-world data. The obtained results demonstrate that the proposed algorithm is a more effective and efficient method in terms of solution time and convergence ability to the best solution compared to mathematical programming solutions for various problem sizes and complexity levels.

Keywords: Unrelated Parallel Machine Scheduling, Sequence-Dependent Setup Times, Variable Neighborhood Search, Metaheuristic

TEŞEKKÜR

Tezimin hazırlanması sürecinde desteğini ve yardımlarını esirgemeyen danışman hocam Dr. Öğr. Üyesi Ahmed Burak Paç'a, her zaman yanımda olan ve beni cesaretlendiren aileme ve sevdiklerime en içten teşekkürlerimi sunarım.

İÇİNDEKİLER

	Sayfa
ÖZET	vi
ABSTRACT	vii
TEŞEKKÜR	viii
İÇİNDEKİLER	ix
SİMGELER ve KISALTMALAR DİZİNİ	x
ŞEKİLLER DİZİNİ	xi
TABLOLAR/ÇİZELGELER DİZİNİ	xii
1. GİRİŞ	1
2. ÇİZELGELEME PROBLEMİ	4
2.1. Çizelgeleme Probleminin Sınıflandırılması	4
2.2. Sıra Bağımlı Hazırlık Süreli İlişkisiz Paralel Makine Çizelgeleme Problemi	6
2.2.1. Matematiksel Model	6
2.3. Literatür Araştırması	9
3. YÖNTEM	14
3.1. Önerilen Değişken Komşuluk Arama Algoritması	14
3.1.1. Değişken Komşuluk İniş Algoritması	17
3.1.2. Komşuluk Yapıları	20
3.1.3. Yöntemin Problem Türevlerinde Uygulanabilirliği ve Esnekliği	22
4. HESAPLAMALI SONUÇLAR	24
4.1. Test Problemleri	24
4.1.1. Önerilen Sentetik Veri Oluşturma Prosedürü	24
4.1.2. Sentetik Veri ile Matematiksel Model Çözümü	26
4.2. DKA Algoritması için Deneysel Sonuçlar	27
5. SONUÇLAR	32
KAYNAKLAR	33
ÖZGEÇMİŞ	38
EKLER	39

SİMGELER ve KISALTMALAR DİZİNİ

AA	:	Ateşböceği Algoritması
AGDKA	:	Akıllı Genel Değişken Komşuluk Arama
AURA	:	Açgözlü Rasgele Uyarlanabilir Arama Algoritması
BA	:	Kombinatoryal Benders Ayrıştırması
BO	:	Balina Optimizasyon
BT	:	Benzetilmiş Tavlama
DE	:	Diferansiyel Evrim
DKA	:	Değişken Komşuluk Arama
DKİ	:	Değişken Komşuluk İniş
EDKA	:	Evrimsel Değişken Komşuluk Arama
EKY	:	ε-Kısıt Yöntemi
ERA	:	Emperyalist Rekabetçi Algoritma
GA	:	Genetik Algoritma
HÇAÖÖTO	:	Hibrit Çok Amaçlı Öğretme-Öğrenme Temelli Optimizasyon
İPMÇP	:	İlişkisiz Paralel Makine Çizelgeleme Problemi
JIT	:	Tam Zamanında Üretim
KA	:	Komşuluk Arama
KP	:	Kısıt Programlama
KS	:	Kurucu Sezgisel
KKS	:	Karışık Kurbağa Sıçrama Algoritması
MTBA	:	Mantık Tabanlı Benders Ayrıştırması
SBHSİPMÇP	:	Sıra Bağımlı Hazırlık Süreli İlişkisiz Paralel Makine Çizelgeleme Problemi
SK	:	Sinüs-Kosinüs Algoritması
SKA	:	Sabit Küme Araması
SO	:	Solucan Optimizasyon Algoritması
SOA	:	Simbiyotik Organizma Arama Algoritması
SS	:	Salp Sürüsü Algoritması
TA	:	Tabu Arama
UBKA	:	Uyarlanabilir Büyük Komşuluk Arama
YAK	:	Yapay Arı Kolonisi Algoritması
YBS	:	Yapay Bağışıklık Sisteminden Esinlenen Algoritma
YYA	:	Yinelenen Yerel Arama

ŞEKİLLER DİZİNİ

Sayfa

Şekil 3.1: DKA'da çözüm uzayının taranması. 15

Şekil 4.1: M=3, N=[10, 15, 20, 25] olan problemlerin çözüm süreleri. 27

TABLOLAR/ÇİZELGELER DİZİNİ

		Sayfa
Tablo 2.1:	Matematiksel modele ait notasyonlar.	7
Tablo 2.2:	SBHSİPMÇP'nin matematiksel modeli.	7
Tablo 2.3:	SBHSİPMÇP’ni ele alan çalışmalar.	11
Tablo 2.4:	SBHSİPMÇP’ni kullanılmayan zaman dilimleri varsayımı altında ele alan çalışmalar.	13
Tablo 3.1:	Değişken komşuluk arama algoritmasının parametreleri.	16
Tablo 3.2:	Değişken komşuluk arama algoritması.	16
Tablo 3.3:	Değişken komşuluk inişi algoritmasının parametreleri.	18
Tablo 3.4:	Değişken komşuluk inişi algoritması.	19
Tablo 4.1:	Sentetik veri oluşturma parametreleri.	24
Tablo 4.2:	Sentetik veri oluşturma parametreleri ve dağılımları.	25
Tablo 4.3:	Komşuluk yapısı seçimde kullanılan olasılık fonksiyonunun komşuluk derecelerine göre ağırlık değerleri.	28
Tablo 4.4:	DKA ve MP yöntemlerinin deneysel sonuçları.	29

1. GİRİŞ

Endüstriyel alanlarda, üretim süreçlerinin verimli bir şekilde planlanması ve yönetilmesi kritik bir öneme sahiptir. İşletmeler, rekabet avantajı elde edebilmek için mevcut kaynakları etkin bir şekilde kullanarak üretim miktarlarını arttırmayı amaçlar. Doğru ve etkin bir çizelgeleme, işletmelerin verimliliğini arttırma ve kaynakları etkin kullanma gibi hedeflerine doğrudan katkı sağlamaktadır.

Çizelgeleme, istenilen çıktıları istenilen zamanda üretmek için gerekli tüm faaliyetleri gerçekleştirmek üzere kaynak kullanımını organize etme, seçme ve zamanlama sürecidir. Bu süreç, faaliyetler ve kaynaklar arasındaki çok sayıda zaman ve ilişki kısıtlamalarını karşılamak zorundadır [1]. Genel olarak, çizelgeleme hem işlerin sıralanmasını hem de kaynakların tahsisini içerir. Tek bir makinenin bulunduğu durumlarda, çizelgeleme işlemi yalnızca işlerin sıralanmasıyla sınırlıdır [2]. Birden fazla paralel makine içeren problemlerin iki temel aşaması vardır: İşlerin hangi makinede üretileceğini belirlemek ve her bir makinedeki işlerin hangi sırayla gerçekleştireceğini belirlemek. Dolayısıyla, paralel makine çizelgeleme problemleri tek makine çizelgeleme problemlerine göre daha karmaşık ve zordur.

Çizelgeleme problemleri, makine, işgücü ve malzeme gibi kaynakların bulunduğu üretim ortamına, işlem süresi, kurulum süresi ve öncelik kuralları gibi işlem özelliklerine ve kapasite, zaman ve bütçe gibi kısıtlamalara bağlı olarak çeşitlilik göstermektedir. Çizelgeleme problemlerinde üretim sisteminin tüm özelliklerinin doğru bir şekilde belirlenmesi son derece önemlidir. Bu özellikler çözüm yönteminin seçiminde etkili olmaktadır. Örneğin, tek makineli üretim sistemleri için, nispeten basit kesin çözüm yöntemleri genellikle yeterli olmaktadır. Ancak, birden fazla paralel makineye sahip karmaşık üretim sistemleri, daha zorlayıcı çizelgeleme problemlerine yol açmaktadır. Bu gibi durumlarda, kesin çözüm yöntemleri hesaplama açısından pratik değildir. Bu nedenle, sezgisel veya metasezgisel çözüm yöntemleri tercih edilmektedir.

Çizelgeleme problemlerinde, işlem özellikleri arasında önemli kavramlardan biri hazırlık süresidir. Hazırlık süresi, bir işin üretime veya işleme başlamadan önce gerçekleştirilmesi gereken işlemleri ifade eder. Bu süreç genellikle; makine ayarı,

makine temizliği, hammadde ve ekipman temini, işin makineye bağlanması gibi faaliyetlerden oluşur. Hazırlık işlemleri literatürde genellikle iki şekilde ele alınmıştır. Hazırlıkların sadece işlem görecek işe bağlı olduğu durum sıra bağımsız, hem işe hem de işin sırasına bağlı olduğu durum sıra bağımlı olarak ifade edilir [3].

Bu çalışma, karmaşık üretim ortamlarında yaygın olarak karşılaşılan bir problem olan sıra bağımlı hazırlık süreli ilişkisiz paralel makine çizelgeleme problemine (SBHSİPMÇP) odaklanmaktadır. Literatürde bu problemi ele alan çalışmaların çoğunda makinelerin sürekli olarak kullanılabilir durumda olduğu varsayılır. Oysa, gerçek hayatta üretim önleyici bakımlar, molalar, izinler ve diğer planlı duruşlar nedeniyle kesintiye uğramaktadır. Gerçekçi çizelgeler oluşturmak için önceden planlanan kullanılamayan zaman dilimlerinin dikkate alınması oldukça önemlidir. Bu sayede, kaynakların daha etkin kullanımı, zaman yönetiminin daha iyi sağlanması ve planlanan hedeflere daha kolay ulaşılması mümkün hale gelir.

Bu çalışmada, makine uygunluk kısıtlarını ve kullanılmayan zaman dilimlerini dikkate alan sıra bağımlı hazırlık süreli ilişkisiz paralel makine çizelgeleme problemi araştırılmıştır. Çalışmanın amacı, toplam gecikme ve erken teslim sürelerini en aza indirmektir. Tam zamanında üretim (JIT) süreçlerinde verimliliği artırması nedeniyle günümüzde oldukça popülerdir. JIT, tercih edilen amaç fonksiyonuyla müşterilerin talep ettiği teslimat tarihlerini karşılamayı ve erken üretimi engelleyerek stok maliyetlerini azaltmayı hedeflemektedir. Ayrıca, ürünlerin raf ömrü veya bekleme koşulları gibi kısıtlar içeren üretim sistemlerinde, toplam gecikme ve erken teslim sürelerinin en aza indirilmesi büyük önem taşımaktadır.

Problemin çözümü için geliştirilen sentetik veri algoritması kullanılarak, gerçek dünya koşullarını iyi bir şekilde yansıtan test problemleri oluşturulmuştur. Matematiksel modelin sağlamlığı test edilerek NP-zor olduğu doğrulanmıştır. Bu nedenle, problemin çözümünde metasezgisel yöntemler tercih edilmiştir. Ele alınan problem için değişken komşuluk arama algoritması önerilmiştir. Önerilen algoritma, çözüm uzayının farklı bölgelerini keşfetmek için çeşitli başlangıç çözümlerinden yola çıkarak üç farklı komşuluk yapısı ve adaptif bir komşuluk yapısı seçim yapısı ve DKİ algortiması kullanmaktadır. Komşuluk yapıları, komşuluk dereceleri ile giderek genişleyen bir çözüm uzayını keşfetmek üzere tasarlanmıştır.

Çalışmanın izleyen bölümünde, çizelgeleme problemlerinden bahsedilmiştir. Bu bölümde çizelgeleme probleminin sınıflandırılması, sıra bağımlı hazırlık süreli ilişkisiz paralel makine çizelgeleme probleminin matematiksel modeli ve literatür araştırmasına yer verilmiştir. Üçüncü bölümde önerilen değişken komşuluk arama algoritması açıklanmıştır. DKİ algoritması ve komşuluk yapıları detaylı bir şekilde ele alınmıştır. Dördüncü bölüm, deneysel sonuçlara odaklanmaktadır. Öncelikle, test problemlerini üretmek için geliştirilen sentetik veri algoritması açıklanmıştır. Ardından, matematiksel model ve önerilen değişken komşuluk algoritması test edilmiştir. Çalışmada ele alınan test problemleri üzerinde, önerilen DKA algoritması ile Matematiksel Programlama (MP) yöntemlerinin sonuçları karşılaştırılmış ve sonuçlar tablo ve grafiklerle sunulmuştur. Elde edilen sonuçlar, DKA algoritmasının çözüm süresi ve en iyi çözüme yakınsama kapasitesi açısından MP yöntemine göre daha etkili ve verimli olduğunu göstermektedir.

2. ÇİZELGELEME PROBLEMİ

Çizelgeleme, üretim ve hizmet endüstrilerinde yaygın olarak kullanılan kritik bir karar verme sürecidir. Bu süreç, belirli kaynakların belirli görevlere atanmasıyla ilgilidir, bu kaynaklar genellikle makinelerden oluşurken, görevler üretim süreçlerindeki işleri temsil eder. Çizelgelemenin temel amacı, makine-iş atamalarını yaparken bir veya daha fazla hedefi optimize etmektir. Bu hedefler arasında, üretim süresini minimize etmek, işlerin gecikme sürelerini azaltmak gibi çeşitli amaçlar yer alabilir. Bu nedenle, çizelgeleme süreci, üretim sistemlerinde önemli bir rol oynar ve operasyonel verimliliği artırmaya yönelik kritik bir araç olarak değerlendirilir [4].

Çizelgeleme, her bir işlemin ne zaman başlayacağını ve ne zaman biteceğini belirleme görevidir. Her işlem, zaman ve kapasite gibi kısıtlı kaynaklar için diğer işlemlerle potansiyel rekabet içindedir. [5].

Çizelgeleme problemleri kombinatoryal eniyileme problemlerinin önemli bir bölümünü oluşturmaktadır. Kombinatoryal eniyileme problemleri, amaç fonksiyonunu eniyileyen kesikli karar değişkenlerinin değerlerini bulmayı hedefler. Çizelgeleme problemleri genellikle P veya NP problemleri olarak sınıflandırılır. P tipi problemler, polinom zamanlı algoritmalarla etkin bir şekilde çözülebilirken, NP tipi problemler için polinom zamanlı bir algoritmanın bulunması mümkün görülmez ve bu tür problemler genellikle üstel zamanlı algoritmalarla çözülebilirler [6].

Çizelgeleme problemleri, gerçek üretim sorunlarını temsil ettiğinden, genellikle büyük boyutlu ve çeşitli kısıtlarla dolu kompleks bir yapıya sahiptirler. Hesaplama zorluğundan dolayı, NP-zor problem olarak sınıflandırılırlar. Üretim sisteminin yapısı, problemin zorluğunu doğrudan etkiler. Dolayısıyla, işlerin ve makinelerin özelliklerini ve modelin kısıtlamalarını doğru bir şekilde tanımlamak oldukça önemlidir.

2.1. Çizelgeleme probleminin sınıflandırılması

Çizelgeleme problemi, kaynak yapılandırması ve işlerin yapısı gibi temel özelliklere göre sınıflandırabilir. Modeller, tek bir makine veya birden fazla makine içerebilir. Tek bir makine içeren modelde gelen işler tek bir makine tarafından birer birer işlenir. Tek makine durumu, tüm olası makine ortamlarının en temel ve basit biçimidir ve

diğer daha karmaşık makine düzenlemelerinin özel bir versiyonunu oluşturur [4]. Birden fazla makine içeren modeller paralel makine, akış tipi ve atölye tipi olmak üzere üç ana kategoriye ayrılır. Paralel makine çizelgeleme problemleri makine özelliklerine bağlı olarak üçe ayrılmaktadır: özdeş paralel makinalar (identical parallel machines), farklı hızlara sahip paralel makinalar (uniform parallel machines) ve ilişkisiz paralel makinalar (unrelated parallel machines). Özdeş paralel makinalar aynı işlevi görebilen ve birbirine eşit olan makinaları ifade eder. Farklı hızlara sahip paralel makinalar ise aynı tipte olup farklı işleme hızlarına sahiptir. İlişkisiz paralel makinalar ise farklı tiplerde olup benzer işlemleri gerçekleştirebilen makinaları kapsar [7]. Akış tipindeki bir iş, farklı görevlere bölünmüş operasyonlardan oluşur ve her operasyon farklı bir makinede gerçekleştirilir. Bu durumda, bir iş, belirli bir öncelik yapısına sahip operasyonların bir araya gelmiş halidir. [8]. Atölye tipi modelde ise, her iş, akış tipi modeldeki gibi öncelik yapısına sahip birkaç operasyondan oluşur fakat akış yönü tek yönlü değildir.

İşlerin sisteme giriş şekline göre statik ve dinamik; parametrelerin belirliliğine göre deterministik ve stokastik olarak sınıflandırılmaktadır. Çizelgelenecek işler kümesi zaman içerisinde değişmiyorsa sisteme statik denilmektedir. Buna karşın, zaman içerisinde yeni işlerin ortaya çıktığı durumlarda ise sistem dinamik olarak adlandırılır. Statik modellerin çözülebilirlik açısından dinamik modellere kıyasla daha elverişli olduğu kanıtlanmış olup, bu tür modeller üzerine daha kapsamlı çalışmalar yapılmıştır. Dinamik modellerin pratik uygulamalarda daha önemli olduğu düşünülse de, statik modeller genellikle dinamik sistemlerin temel özelliklerini doğru bir şekilde yansıtabilir. Statik problemlerin analizi, dinamik durumlar için değerli bilgilerin ve sağlam sezgisel ilkelerin geliştirilmesine katkı sağlar. Son olarak, koşulların kesin olarak bilindiği varsayıldığında modele deterministik denir. Diğer taraftan, belirsizliğin açıkça olasılık dağılımları ile kabul edildiği durumlarda ise modele stokastik denilmektedir [8]. Bir diğer deyişle, tüm veriler önceden biliniyor ve sabit ise deterministik problem, verilerden en az biri kesin olarak bilinmiyor ve olasılık dağılımına göre değer alıyor ise stokastik problem olarak adlandırılır.

Çizelgeleme alanındaki çoğu çalışma, makinelerin her zaman kullanılabilir olduğunu varsayar. Ancak, gerçek endüstriyel ortamlarda bu varsayım her zaman geçerli olmayabilir; yani makineler her zaman kullanılabilir durumda olmayabilir. Örneğin, makineler çizelgeleme dönemi boyunca makine arızası (stokastik) veya önleyici bakım

(deterministik) nedeniyle kullanılamayabilir [9]. Önleyici bakım faaliyetleri, bakım sürelerinin önceden bilindiği ve sabit tutulduğu deterministik durumdur. Makine arıza ve onarım süreçleri önceden bilinmediği ve rastgelelik içerdiği için stokastik durumdur.

Literatürde, çizelgeleme problemlerinde işlerin sürdürülebilirliği iki şekilde değerlendirilir. Sürdürülebilir iş, eğer bir iş makinenin arıza başlangıcından önce tamamlanamazsa, makine tekrar çalışır hale geldikten sonra işin kaldığı yerden devam edebilmesidir. Öte yandan, sürdürülemez işler, makine arızasından sonra devam etmek yerine yeniden başlatılması gereken işlerdir [9].

2.2. Sıra Bağımlı Hazırlık Süreli İlişkisiz Paralel Makine Çizelgeleme Problemi

Sıra bağımlı hazırlık süreli ilişkisiz paralel makine çizelgeleme problemi, makine çizelgelemenin özel ve karmaşık bir alt konusudur. Bu problemde işler, farklı özellikteki paralel makinelerde işlenir ve her bir işin her bir makinede tamamlanma süreleri farklıdır. Her iş işlenmeye başlamadan önce bir hazırlık süresine ihtiyaç duymaktadır. Hazırlık süreleri, işlerin bir makinede işlenen son işe veya o makinede ilk işlenme durumuna bağlı olarak farklılık gösterebilir [10]. Hazırlık süresi, ilişkisiz paralel makine ortamında makineye bağlı olarak da değişmektedir.

2.2.1. Matematiksel Model

Bu bölümde sıra bağımlı hazırlık süreli ilişkisiz paralel makine çizelgeleme probleminin matematiksel tam sayılı doğrusal programlama modeline ait varsayımlar, gösterimler ve formüller aşağıda verilmiştir. Bu formüller Furugi [11] tarafından geliştirilmiştir.

Bu problemde, N adet işin M adet ilişkisiz paralel makine üzerinde çizelgelenmesi gerekmektedir. İşler birbirinden bağımsızdır ve öncelik kısıtlaması bulunmamaktadır. Her iş sadece bir makinede bölünme veya kesinti olmaksızın işlenebilir. Her makine aynı anda birden fazla işi yapamaz ve her biri farklı hazırlık sürelerine sahiptir, bu süreler işin sırası ve makine tipine bağlı olarak değişiklik gösterir. Makinelerin hızları ve özellikleri birbirinden farklıdır. Bu nedenle her iş her makine de üretilemez.

Makineler belirli işleri yapabilecek kapasitede olup, bakım nedeniyle belirli zaman aralıklarında müsait olmayabilirler. Bir işin tamamlanması, makinenin müsait olduğu zaman aralığı ile çakışıyorsa, iş, makine tekrar müsait olduğunda devam ettirilecek şekilde çizelgelenir. İşlerin herhangi bir önceden belirlenmiş teslim zamanında tamamlanması gerekmektedir.

Modele ait gösterimler Tablo 2.1'de, amaç fonksiyonu ve kısıtlar Tablo 2.2'de verilmiştir.

Tablo 2.1: Matematiksel modele ait notasyonlar.

Gösterim	Gösterimin Açıklaması
p_{ji}	j. işin i. makinesindeki işlem süresi
d_j	j. işin termini
S_{jki}	eğer j. işten sonra gelen k. işinin i. makinedeki hazırlık süresi
S_{0ji}	eğer j. iş i. makinede birinci sırada yapılıyor ise, ilgili makinedeki hazırlık süresi
u_{ji}	eğer j. iş i. makinede yapılabiliyor ise 1, yapılamıyor 0
x_j	j işinin başlayabileceği en erken zaman
C_j	j. işin tamamlanma zamanı
E_j	j. işin erken teslim süresi
T_j	j. işin gecikme süresi
A_{ji}	eğer i. makineye j. iş atanır ise 1, aksi halde 0
Z_{jki}	eğer i. makinede j. işten sonra k. iş yapılıyor ise 1, aksi halde 0
Z_{0ji}	eğer i. makinede j. iş birinci sırada yapılıyor ise 1, aksi halde 0
y_j	eğer j. iş herhangi makinenin ikinci müsait olmadığı zaman aralığından sonra işlem görüyor ise 1, aksi halde 0
$[0, R_i]$	i. makinenin başlangıçta müsait olmadığı zaman aralığı
$[SM_i, FM_i]$	i. makinenin müsait olmadığı zaman aralığı
BigM	büyük sayı

Tablo 2.2: SBHSİPMÇP'nin matematiksel modeli.

$\min \sum_{j=1}^{n} (E_j + T_j)$		(1)
$\sum_{i=1}^{m} u_{ji} A_{ji} = 1$	$\forall j$	(2)
$A_{ji} \leq u_{ji}$	$\forall i, j$	(3)
$\sum_{j=1, j\neq k}^{n} Z_{jki} + Z_{0ki} = A_{ki}$	$\forall i, k$	(4)
$\sum_{j=1, j\neq k}^{n} Z_{kji} \leq A_{ki}$	$\forall i, k$	(5)

$\sum_{j=1}^{n} Z_{0ji} \leq 1$	$\forall\, i$	(6)
$Z_{jki} \leq A_{ji}$	$\forall\, i,j,k; j \neq k$	(7)
$Z_{jki} \leq A_{ki}$	$\forall\, i,j,k; j \neq k$	(8)
$C_j \geq x_j + \sum_{i=1}^{m} \sum_{k=1,j\neq k}^{n} Z_{kji}\, S_{kji} + \sum_{i=1}^{m} Z_{0ji}\, S_{0ji} + \sum_{i=1}^{m} A_{ji}\, p_{ji}$	$\forall\, j$	(9)
$x_j \geq \sum_{i=1}^{m} A_{ji}\, R_i$	$\forall\, j$	(10)
$C_j \leq \sum_{i=1}^{m} A_{ji} SM_i + bigM\, y_j$	$\forall\, j$	(11)
$x_j \geq \sum_{i=1}^{m} A_{ji} FM_i - bigM\,(1 - y_j)$	$\forall\, j$	(12)
$C_j \leq x_k + bigM\left(1 - \sum_{i=1}^{m} Z_{jki}\right)$	$\forall\, j,k; j \neq k$	(13)
$E_j \geq d_j - C_j$	$\forall\, j$	(14)
$T_j \geq C_j - d_j$	$\forall\, j$	(15)
$x_j, C_j, E_j, T_j \geq 0$	$\forall\, j$	(16)
$A_{ji}, Z_{jki}, Z_{0ji}, y_j \in \{0,1\}$	$\forall\, i,j,k; j \neq k$	(17)

Tablo 2.2'deki 1. eşitsizlikte amaç fonksiyonu belirtilmiştir. Amaç fonksiyonu toplam gecikme ve erken teslim sürelerinin en küçüklenmesidir. 2. eşitsizlik, her işin yalnızca üretilebileceği makineler arasından bir tanesinde gerçekleştirilmesini sağlar. 3. eşitsizlik, bir makinenin yapamayacağı işlerin o makineye atanmasını önler. 4. eşitsizlik, her bir işin atandığı makinede mutlaka bir öncülü olması sağlar. 5. eşitsizlik, her bir işin atandığı makinede en fazla bir ardılı olabileceğini belirtir. 6. eşitsizlik, her makinede en fazla bir işin birinci sıradan yapılmasını sağlar. 7 ve 8. eşitsizlikler doğrusallaştırma kısıtlarıdır. 9. eşitsizlik, her işin tamamlanma süresini hesaplar. 10-12. eşitsizlikler, her işin sadece makinelerin uygun olduğu zaman dilimlerinde başlayıp bitirilmesini sağlar. 13. eşitsizlik, her makineye atanan işlerin sırasını belirler. 14 ve 15. eşitsizlikler, işlerin gecikme veya erken teslim sürelerini hesaplar. 16 ve 17. eşitsizlikler ise pozitif değişkenleri ve ikili değişkenleri tanımlar.

2.3. Literatür Araştırması

İlişkisiz paralel makine çizelgeleme problemleri (İPMÇP), çeşitli kısıtlar ve farklı üretim ortamları altında incelenmiş ve birçok araştırmacının çalışma konusu olmuştur. Cheng ve Sin [12], Chen vd. [13] ve Mokotoff [14] ilişkisiz paralel çizelgeleme problemine odaklanan öncü çalışmalara imza atmışlardır. İPMÇP farklı üretim ortamlarına göre farklı kısıtlamalar içerebilmektedir. Çoğu çalışmada modeli basitleştirmek için bu kısıtlamalar göz ardı edilir; ancak, gerçek üretimde doğru çizelgeler oluşturmak için bu kısıtlamaların modele dahil edilmesi gerekmektedir.

Bu çalışmada, sıra bağımlı hazırlık süreli, kullanılmayan zaman dilimleri ve makine uygunluk kısıtlamaları içeren İPMÇP'i ele alınmıştır. Literatürde, kullanılmayan zaman dilimlerini dikkate alan ve toplam gecikme ve erken teslim süresini en aza indirmeyi amaçlayan SBHSİPMÇP'ini ele alan çalışmaların az olması bu çalışmanın motivasyon kaynağı olmuştur. Tablo 2.3 ve 2.4'te, son on yılda SBHSİPMÇP konusunu ele alan çalışmalar, kullanılan amaç fonksiyonları ve çözüm yöntemleri ile birlikte özetlenmiştir. Tablo 2.3, SBHSİPMÇP'ni ele alan çalışmaları, Tablo 2.4 ise makinelerin belirli zaman dilimlerinde kullanılmadığı varsayımı altında SBHSİPMÇP'ni elen alan çalışmaları içermektedir. Çalışmalarda çözüm yöntemi olarak genellikle iki veya daha fazla algoritmayı birleştiren hibrit metasezgisel algoritmalar, amaç fonksiyonu olarak da en büyük tamamlanma zamanının (C_{max}) en aza indirilmesi tercih edilmiştir.

Literatürde, sıra bağımlı hazırlık süreli ilişkisiz paralel makine çizelgeleme problemi için çeşitli çözüm yöntemleri geliştirilmiştir. Bu yöntemler kesin, sezgisel, metasezgisel ve hibrit yaklaşımları içermektedir.

Gedik vd. [15] özelleştirilmiş dallanma stratejisi kullanan bir kısıt programlama (KP) modeli, Gomes ve Mateus [16] kombinatoryal benders ayrıştırmasının (BA) yakınsama süresini hızlandırmak için üç prosedür, Tran vd. [17] mantık tabanlı benders ayrıştırmasına (MTBA) ve dallanma ve kontrole dayalı olarak iki ayrıştırma tabanlı yöntemler önerilmişlerdir. Naderi vd. [18] çok amaçlı karma tam sayılı doğrusal programlama ile ε-kısıt yöntemini (EKY) kullanarak küçük boyutlu problemleri ele almışlardır. Ayrıca, hibrit çok amaçlı öğretme-öğrenme temelli optimizasyon (HÇAÖÖTO) algoritması önermişlerdir. Saraç ve Özçelik [19], mat-sezgisel bir algoritma geliştirerek genişletilmiş ε-kısıt yöntemiyle sonuçları

karşılaştırmışlardır. Moonsri vd. [20] küçük boyutlu problemler için bir matematik model, büyük boyutlu problemler için diferansiyel evrim (DE) algoritmasını önermişlerdir. Stefanello vd. [21], kesin ve metasezgisel bir algoritmaya dayanan hibrit bir komşuluk arama (KA) yaklaşımı önermişlerdir. Rego ve Souza [22], kesin ve sezgisel optimizasyon stratejilerini birleştirerek çözüm uzayını araştıran geliştirilmiş akıllı genel değişken komşuluk arama (AGDKA) algoritmasını önermişlerdir. Perez-Gonzalez vd. [23], yeni onarım ve iyileştirme aşamalarına sahip çeşitli kurucu sezgisel yöntemler (KS) önermişlerdir. Liao vd. [24], üç aşamalı sezgisel bir yöntem ve üç farklı yerel arama yöntemi kullanan genetik algoritma önermişlerdir. Ji vd. [25], çeşitli yok etme ve onarma operatörleri ile gelişmiş uyarlanabilir büyük komşuluk arama (UBKA) algoritması önermişlerdir. Fang vd. [26], öğrenme otomatlarıyla UBKA ve tabu arama (TA) yöntemlerini birleştiren bir hibrit metasezgisel yaklaşım önermişlerdir. Çözümü geliştirmek ve algoritmanın yerel optimumdan kaçmasına yardımcı olmak için dinamik bir pertürbasyon şeması önerilmiştir. Iori vd. [27] gerçek dünya problemini çözmek için çeşitli yerel arama prosedürlerini kullanan açgözlü rastgele uyarlamalı arama yöntemi (ARUA) geliştirmişlerdir. Jovanovic ve Voß [28], ARUA'ya öğrenme mekanizması ekleyen yeni bir popülasyon tabanlı sabit küme araması (SKA) geliştirmişlerdir. Arnaout [29] yeni bir solucan optimizasyon (SO) algoritması, Ezugwu ve Akutsah [30] yerel arama ile iyileştirilmiş ateş böceği algoritması (AA), Huang vd. [31] kur yapma öğrenimi ile güçlendirilmiş AA geliştirmişlerdir. Al-qaness vd. [32] AA algoritmasının operatörlerini kullanarak balina optimizasyon (BO) algoritmasının arama yeteneğini geliştiren hibrit bir algoritma, Ewees vd. [33] ateş böceği algoritması tabanlı geliştirilmiş bir salp sürüsü (SS) algoritması geliştirmişlerdir. Moser vd. [34] problem çözümü için benzetilmiş tavlama (BT), Ezugwu [35] simbiyotik organizma arama algoritması (SOA) ve BT yöntemlerini geliştirip aynı zamanda SOA ve BT yöntemlerini birleştirerek hibrit bir yaklaşım önermişlerdir. Zeidi ve MohammadHosseini [36], yeni bir matematiksel model sunmuşlar ve problemi çözmek için genetik algoritma (GA) ve BT yöntemlerini entegre eden bir algoritma önermişlerdir. Canıyılmaz vd. [37], önerdikleri yeni komşuluk yaklaşımını yapay arı kolonisi (YAK) algoritması ve GA ile entegre ederek algoritmaları kıyaslamışlardır. Jouhari vd. [38] BT ve sinüs kosinüs (SK) algoritmalarını birleştiren hibrit bir yöntem, Lin ve Ying [39] hibrit bir YAK algoritması, Adan [40] hibrit bir GA önermişlerdir. Diana vd. [41], değişken komşuluk iniş (DKİ) yöntemi tabanlı yapay bağışıklık

sisteminden (YBS) esinlenen bir algoritma önerilmişlerdir. Avalos-Rosales vd. [42], problem için yeni bir maksimum tamamlanma süresi doğrusallaştırması ve birkaç karışık tamsayı formülasyonu geliştirmişlerdir. Problem çözümü için birden fazla başlangıç noktalı DKİ algoritması önermişlerdir. Haddad vd. [43] ve Diana vd. [44], yinelemeli yerel arama (YYA) ve DKİ yöntemlerinin birleştirildiği bir hibrit metasezgisel önermişlerdir. Abdullah vd. [45], iki aşamalı bir evrimsel değişken komşuluk arama (EDKA) algoritmasını önermişlerdir. Kılıç [10], yeni bir DKA algoritması önermiştir. Yeni önerilen algoritma, komşuluk yapılarına 2 farklı parametre seçimi eklemektedir. Tozzo vd. [46], GA ve DKA algoritmalarının performanslarını değerlendirmişlerdir. DKA'nın çözüm kalitesi ve çözüm süresi açısından GA'dan daha iyi performans gösterdiğini sonucuna varmışlardır. Al-Harkan ve Qamhan [47], problemin çözümü için yeni bir karma tam sayılı doğrusal programlama modeli sunmuşlardır. Ayrıca, DKA ve BT temelli iki aşamalı bir hibrit metasezgisel önermişlerdir.

[11],[48]-[52] çalışmalarında belirli zaman dilimlerinde kullanılmadığı varsayımı altında SBHSİPMÇP'ini incelemişlerdir. Problemin çözümü için Avalos-Rosales vd. [48] çoklu başlangıç stratejisine dayalı etkili bir metasezgisel algoritma, Wang ve Pan [49] emperyalist rekabetçi (ER) algoritma, Lei ve Yang [50] YAK, Furugi [11] TA ve Özçelik ve Saraç [51] GA önermişlerdir. Lei ve Yi [52], kötüleşen önleyici bakım varsayımı altında problemi ele almış ve problemi çözmek için yeni bir farklılaştırılmış karışık kurbağa sıçrama (KKS) algoritması geliştirmişlerdir.

Tablo 2.3: SBHSİPMÇP'ni ele alan çalışmalar.

Kaynak	C_{max}	$\sum T_j$	$\sum w_j T_j$	$\sum E_j + T_j$	T_{max}	E_{max}	$\sum S_{jki}$	Yöntem
Gedik vd. [15]	x							KP
Gomes ve Mateus [16]	x							BA
Tran vd. [17]	x							MTBA
Naderi vd. [18]	x				x	x		EKY, HÇAÖÖTO
Saraç ve Özçelik [19]	x	x						Mat-Sezgisel
Moonsri vd. [20]	x							DE
Stefanello vd. [21]	x							KA
Rego ve Souza [22]	x							AGDKA

Perez-Gonzalez vd. [23]		x				KS
Liao vd. [24]	x					GA
Ji vd. [25]	x					UBKA
Fang vd. [26]	x					UBKA -TA
Iori vd. [27]			x		x	AURA
Jovanovic ve Voß [28]	x					ARUA-SKA
Arnaout [29]	x					SO
Ezugwu ve Akutsah [30]						AA
Huang vd. [31]	x					AA
Al-qaness vd. [32]	x					BO-AA
Ewees vd. [33]	x					SS-AA
Moser vd. [34]	x	x				BT
Ezugwu [35]	x					SOA-BT
Zeidi ve MohammadHosseini [36]				x		GA-BT
Canıyılmaz vd. [37]	x	x				YAK-GA
Jouhari vd. [38]	x					BT-SK
Lin ve Ying [39]	x					YAK
Adan [40]	x					GA
Diana vd. [41]	x					YBS
Avalos-Rosales vd. [42]	x					DKİ
Haddad vd. [43]	x					DKİ -YYA
Diana vd. [44]			x			DKİ-YYA
Abdullah vd. [45]	x					EDKA
Kılıç [10]	x					DKA
Tozzo vd. [46]	x					DKA, GA
Al-Harkan ve Qamhan [47]	x					DKA-BT

*C_{max}:en büyük tamamlanma zamanı, $\sum T_j$:toplam gecikme süresi, $\sum w_j T_j$:toplam ağırlıklı gecikme süresi, $\sum E_j + T_j$: toplam gecikme ve erken teslim süresi, T_{max}:en büyük gecikme süresi, E_{max}: en büyük erken teslim süresi, $\sum S_{jki}$: toplam hazırlık süresi

Tablo 2.4: SBHSİPMÇP'ni kullanılmayan zaman dilimleri varsayımı altında ele alan çalışmalar.

Kaynak	C_{max}	$\sum T_j$	$\sum E_j + T_i$	Yöntem
Avalos-Rosales vd. [48]	x			Metasezgisel
Wang ve Pan [49]	x	x		ER
Lei ve Yang [50]	x	x		YAK
Furugi [11]			x	TA
Özçelik ve Saraç [51]	x			GA
Lei ve Yi [52]	x			KKS

*C_{max}:en büyük tamamlanma zamanı, $\sum T_j$:toplam gecikme süresi, $\sum E_j + T_j$: toplam gecikme ve erken teslim süresi

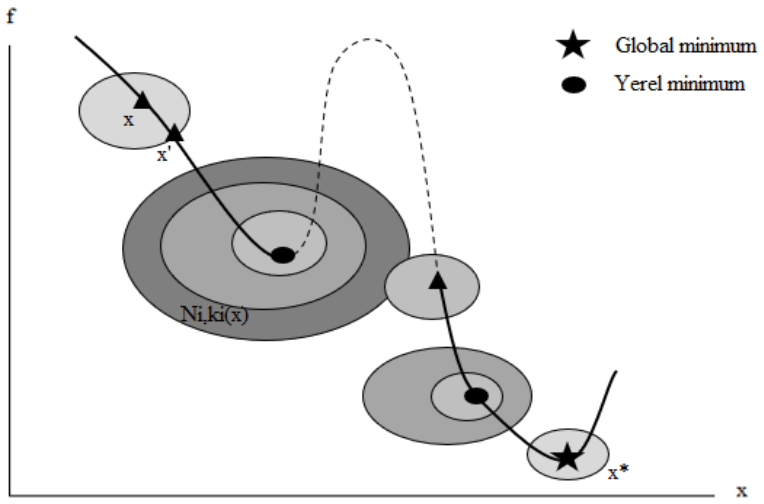
3. YÖNTEM

Sıra bağımlı hazırlık süreli ilişkisiz paralel makine çizelgeleme problemi endüstriyel alanlarda sıkça karşılaşılan NP-zor bir problemdir. Küçük boyutlu problemlerde kesin çözüm yöntemleri kullanılabilir. Problem çözümünde kullanılan matematiksel model, kesin çözümler elde etmesine rağmen, çözüm süresi açısından yeterli bulunmamıştır. Problemin boyutu, problemin kombinatoryal yapısından dolayı üstel bir şekilde artış göstermekte ve 20 boyutlu bazı problemlerde 7200sn içerisinde en iyi çözümü bulmadıkları gözlemlenmiştir. Bu nedenle, özellikle büyük boyutlu problemlerde çözüm yöntemi olarak daha kısa sürede sonuç veren metasezgisel yöntemler önerilmiştir. Bu çalışmada da SBHSİPMÇP çözüm yöntemi olarak metasezgisel algoritma olan DKA algoritması önerilmiştir.

3.1. Önerilen Değişken Komşuluk Arama Algoritması

DKA, yerel arama ile komşuluk yapılarını sistematik bir şekilde keşfetme fikrine dayanan bir metasezgiseldir [53]. DKA, temel yerel arama yönteminin bir varyasyonunu önerir; her bir arama iterasyonunda farklı karmaşıklık düzeylerine sahip bir dizi komşuluğu keşfetmek, mevcut çözümü iyileştirmek, her iyileştirmeden sonra en basit komşuluk düzeyinden başlayarak daha iyi çözüm arayışını devam ettirmek üzere işler [54]. DKA'nın DKİ olarak adlandırılan her bir iterasyonu bu adımları içerir, bulabildiği en iyi çözümün tüm komşuluk düzeylerindeki komşularından hiçbiri daha iyi hedef değere sahip olmadığında sonlanır. DKA temel olarak iki bileşenin yinelemelerinden oluşur: başlangıç çözümü üretimi, ve bu çözümden başlanılarak DKİ'nin uygulanması. Yeni başlangıç çözümü üretimi çözüm uzayının daha verimli keşfi için önceki başlangıç ve DKİ çıktı çözümlerinin oluşturduğu geçmiş veriden yararlanabilir. Bir yinelemede DKİ çıktısı DKA ile o yinelemeye kadar elde edilmiş en iyi çözümden daha iyi hedef değere sahipse, en iyi hedef değer ve en iyi çözüm güncellenir.

Şekil 3.1'de Hansen vd. [55]'nin çalışmasından yararlanılarak, değişken komşuluk arama algoritması ile farklı komşuluk yapıları kullanılarak global optimuma ulaşırken çözüm uzayının nasıl tarandığı gösterilmiştir.



Şekil 3.1: DKA'da çözüm uzayının taranması.

Önerilen değişken komşuluk arama algoritması, çözüm uzayında çeşitli başlangıç çözümleri ile yinelemeler başlatılarak ve bu başlangıç çözümleri DKİ ile daha iyi çözüm aramasına tabi tutularak çalışmaktadır. Başlangıç çözümü elde etmek için önce her iş bu işi işleyebilen geçerli makinelerden birine rastgele olarak atanmakta, daha sonra her makineye atanmış işler kendi içlerinde termin sürelerine göre sıralanmaktadır. Eldeki en iyi çözümün makine atamalarını içeren kısmı çözüm uzayında yeni başlangıç çözümlerinin bulunması için saklanmaktadır. Yeni bir başlangıç çözümünde her bir iş P ihtimalle işlenebileceği diğer makinelerden birine rastgele olarak atanmaktadır. Algoritmanın süre sınırı dışındaki durma koşulu T DKİ yinelemesinde eldeki en iyi çözümün iyileştirilememesidir. DKA algoritmasının parametreleri Tablo 3.1'te sunulmaktadır.

Tablo 3.1: Değişken komşuluk arama algoritmasının parametreleri.

Parametre	Açıklama
T	Daha iyi çözüm bulunmaksızın sürdürülecek DKİ yineleme sayısı
P	Eldeki en iyi çözümün makine atamalarına göre bir işin makine atamasının değiştirilme olasılığı

Tablo 3.2'de N adet iş ve M adet makine için önerilen değişken komşuluk arama algoritması verilmiştir. Algoritma, DKİ ile eniyilenmiş herhangi bir başlangıç çözümü olmadan, x* değişkenine boş küme atanarak başlamaktadır (satır 1). T, değişken komşuluk iniş yinelemelerinde iyileştirme sağlanmayan art arda iterasyon sayısı, 0

olacak şekilde başlatılır (satır 2). DKA üst üste T DKİ yinelemesinde hedef değerde geliştirme sağlayamadığı bir durum ortaya çıkıncaya kadar yinelemelere devam eder (satır 3). Henüz bir DKİ yinelemesi yapılmadıysa (satır 5) tüm işler işlenebildikleri makinelerden birine rastgele atanır (satır 6). Mevcut eniyi çözüm olduğunda (satır 7) bu çözümdeki her işin makine ataması P olasılıkla işlenebildiği bir diğer makineye rastgele seçime göre değiştirilerek (satır 9) yeni bir DKİ başlangıç çözümünün makine atamaları elde edilir (satırlar 8-13). Her makineye atanan işler kendi içlerinde termin tarihlerine göre sıralanır (satır 14). Atamalar (A) ve iş sıralarıyla (Z) tam olarak belirlenmiş olan yeni çözüm x = (A, Z) DKİ'ye başlangıç çözümü olarak verilir ve DKİ'nin çıktısı x′ olarak kaydedilir (satır 16). Eğer x′ ilk DKİ koşumunun sonucu ise (satır 17) en iyi çözüm x* olarak x′ kaydedilir (satır 18). Eğer daha önce DKİ koşumu gerçekleştiyse (satır 19) ve x′ hedef değerde (belli bir çözüm x için toplam gecikme ve erken teslim süresi olan hedef değer f(x) ile gösterilmiştir) mevcut en iyi çözüm x*'a göre iyileştirme sağlıyorsa (satır 20), x* x′ olarak güncellenir (satır 21), hedef değer iyileşmesinin sağlanmadığı yineleme sayacı T sıfırlanır (satır 22); aksi durumda (satır 23) T bir artırılır (satır 24). T yineleme boyunca yeni başlangıç çözümleriyle DKİ uygulanır ve bu yinelemelerde hedef değer iyileşmezse algoritma sonlanır. DKİ iterasyonlarından elde edilen en iyi çözüm x* raporlanır (satır 25). Algoritma, global optimuma ulaşmayı hedefleyen ve çözüm uzayında geniş bir arama yapmayı sağlayan esnek bir çerçeve sunar ve T ve P parametrelerinin ince ayarı ile başarımı artırılabilir.

Tablo 3.2: Değişken komşuluk arama algoritması.

1: İlk başlangıç çözümü: x* ← ∅
2: t ← 0
3: t ≤ T olduğu sürece: ▷ İyileşme sağlanmayan yineleme T'den az
4: $A_{j,i} = 0$, i=1,…,M, j=1,…,N
5: Eğer x* = ∅: ▷ İlk iterasyonda yeni çözüm üretilir
6: j=1,…,N: $\hat{\imath}$ ← rastgele seçim ({i: $u_{j,i}$ = 1}), $A_{j,\hat{\imath}} = 1$
7: Değilse: ▷ Mevcut en iyi çözüm $x^* = (A^*, Z^*)$
8: j = 1'den N'ye kadar:
9: Eğer rastgele[0,1] < P:
10: $\hat{\imath}$ ← rastgele seçim({i: $u_{j,i}$ = 1, $A^*_{j,i} \neq 1$})
11: $A_{j,\hat{\imath}} = 1$
12: Değilse:
13: $A_{j,i} = A^*_{j,i}, i: A^*_{j,i} = 1$
14: $Z_i = (j_1, \dots, j_{n_i});\ l = 1, \dots, n_i\ için\ A_{ij_l} = 1\ ise;$
$d_{j_l} \leq d_{j_{l+1}}, l = 1, \dots, n_i - 1, i = 1, \dots, m$
15: x = (A, Z)

```
    x′ = DKİ(x)
    Eğer x* = ∅
                x* = x′
    Değilse:
        Eğer f(x) < f(x*)  ise:
            x* = x′
            t ← 0
        Değilse:
            t ← t + 1
Çıktı x*
```

3.1.1. Değişken Komşuluk İniş Algoritması

DKİ yöntemi, komşuluk değişimlerinin deterministik bir şekilde gerçekleştirildiği bir yöntemdir [55]. DKİ, önerilen DKA algoritmasının temel eniyileme adımı olarak DKA ile entegre bir şekilde çalışmaktadır.

Önerilen DKİ algoritmasında, Bölüm 3.1.2'de ayrıntılı açıklanacağı üzere, farklı d_1, d_2, d_3 değerleri ile karmaşıklık dereceleri gösterilen üç farklı komşuluk yapısı olan $N_{1,d_1}, N_{2,d_2}, N_{3,d_3}$ taranmaktadır. İlk komşuluk yapısı belli bir makineye atanmış işler arasından iki veya daha fazlasının seçilerek kendi aralarında sıra değişimine tabi tutulmasını öngörür. İkinci komşuluk yapısında makinesi aynı veya farklı olan işlerden iki veya daha fazlasının seçilerek kendi aralarında sıra değişimine tabi tutulmasını öngörülür. Bu durumda işlerin atandığı makineler de değişebilir ve elde edilen yeni çözüm ancak makineler üzerlerine yeni atanan işleri işleyebiliyorsa bir komşu çözüm olarak kabul edilir. Üçüncü komşuluk yapısında bir veya daha fazla iş seçilir ve atanabileceği farklı makineler arasından birine atanır. Üzerlerine yeni işlerin atandığı makinelerde eklenen işler hedef değeri en azaltacak şekilde mevcut iş sırasının başına, sonuna veya mevcut işlerin arasına yerleştirilir. Üç komşuluk yapısı için dereceler, yani değişim için seçilen işlerin sayısı, sırasıyla, $2\text{-}D_1$, $2\text{-}D_2$ ve $1\text{-}D_3$, arasında değer alır. Birinci ve ikinci komşuluk türünde yer değişimi türü bir işlem olduğu için en düşük komşuluk derecesi 2, üçüncü komşuluk yapısında tek bir iş de farklı bir makineye atanabildiği için en düşük komşuluk derecesi 1'dir. Bir makine atamasının iyi bir tercih olduğu makine içi sıralarda hedef değeri azaltan bir düzen elde edildiğinde ortaya çıkar. Bununla birlikte, makine ataması en iyiden çok uzak olduğunda makine içindeki sıranın eniyilenmesi için hesaplama gücünün sarfı verimsiz olur. Böylelikle, üç komşuluk yapısından birinin içerdiği komşu çözümlerin

keşfi bir diğerininkine göre kesin olarak öncelikli değildir ve DKİ belli bir çözümün daha üst dereceli komşuluklarını keşfederek arama yaparken her seferinde üç komşuluk yapısından birini rastgele tercih ederek onun keşfedilmemiş bir üst derecesini ele alır. Şu söylenebilir ki DKİ başlangıçta makine atamalarındaki değişikliklere daha fazla odaklanırken bu atamaların makine içi sıralamaya kıyasla önemi çözüm iyileştikçe azalır. Buna uygun uyarlanabilen bir seçim sağlamak için, bir sonraki adımda bir üst derecesi keşfedilecek olan komşuluğun birinci, ikinci veya üçüncü komşuluk yapılarından gelme olasılıkları olan p_1, p_2 ve p_3, keşfedilmiş olan en yüksek dereceler d_1, d_2 ve d_3'e bağlı olarak değer alan fonksiyonlardır:

$$p_i(d_1, d_2, d_3) \geq 0, i = 1,2,3 \tag{3.1}$$

$$\sum_{i=1}^{3} p_i(d_1, d_2, d_3) = 1 \tag{3.2}$$

$$d_1 \in \{2, \dots, D_1\}, d_2 \in \{2, \dots, D_2\}, d_3 \in \{1, \dots, D_3\} \tag{3.3}$$

Bu seçim biçimi hangi komşunun önce keşfedileceğine rassal bir nitelik kazandırmakla birlikte, eldeki çözümden daha iyi bir çözüm bulunana kadar her üç komşuluk tipinde d_1, d_2 ve d_3'ün ilgili aralıkları olan 2-D1, 2-D2 ve 1-D3 artan sırayla tamamen taranacaktır. Tablo 3.3'te DKİ algoritmasının parametrelerini özetlemektedir.

Tablo 3.3: Değişken komşuluk inişi algoritmasının parametreleri.

Parametre	Açıklama
D_i	i. komşuluk yapısında taranacak en yüksek derece. $i = 1,2,3$
$p_i(d_1, d_2, d_3)$	Üç komşuluk yapısında d_1, d_2, d_3 derecelerine kadar komşuluklar taranmışken bir sonraki taranacak komşuluğun i. komşuluk yapısından gelme olasılığı. $i = 1,2,3$, $d_1 \in \{2, \dots, D_1\}, d_2 \in \{2, \dots, D_2\}, d_3 \in \{1, \dots, D_3\}$.

Seçilen komşuluk yapısında tüm komşu çözümler taranır ve en iyi komşu çözüm mevcut çözüm ile kıyaslanır. Eğer bulunan komşu çözüm mevcut çözümden daha iyi ise komşuluk dereceleri sıfırlanır ve yeni bir komşuluk yapısı seçimi yapılarak yeni çözümün komşuları taranır. Seçilen komşuluk yapısında mevcut derecede iyileştirme sağlanamazsa, mevcut komşuluk yapısının derecesi bir arttırılır ve keşfedilecek yeni komşuluk için tekrar kura çekilir. Tüm komşuluk dereceleri üst sınıra ulaştığında

döngüden çıkılarak DKİ tamamlanır. Bu şekilde özetlenebilecek olan DKİ algoritması Algoritma 3.4'te açıklanmaktadır. DKA'dan DKİ ile iyileştirilmesi için beslenen bir x çözümü ve komşuluk dereceleri (d_1, d_2, d_3)'ün en alt düzeyleri ile olan (2, 2, 1) ile başlanmaktadır (satırlar 0-1). Üç komşuluk yapısının en üst derecesine kadar taranmamış olanları arasından (satırlar 3-4) p_1, p_2, p_3 ağırlıklarına göre bir rastgele seçim yapılmaktadır (satır 2). Seçilen komşuluk yapısının keşfedilmemiş en üst derecesindeki komşu çözümler, $N_{i,d_i}(x)$, daha düşük hedef değere sahip çözüm bulunana kadar taranmakta (satırlar 5-6), eğer böyle bir çözüm bulunursa x'in yerini almakta (satır 7) ve arama yeni x değerinin en düşük dereceli komşuları taranarak devam etmektedir (satırlar 8-9). Taranan komşulukta daha iyi bir çözüm bulunamamış ve henüz (D_1, D_2, D_3) değerlerine kadar tüm komşuluklar taranmamışsa (satır 10) en son taranan komşuluk yapısı için daha sonra taranacak komşuluk derecesi bir artırılır (satır 13) ve komşuluk tarama süreci devam eder (satır 14). Bir x çözümü için (D_1, D_2, D_3) değerlerine kadar tüm komşuluklar taranmışsa (satır 10) ve x'ten daha iyi hedef değere sahip bir komşu bulunamamışsa, DKİ x çözümünün çıktı olarak sunulmasıyla tamamlanır.

Tablo 3.4: Değişken komşuluk iniş algoritması.

0: Girdi: Başlangıç çözümü x.
1: $(d_1, d_2, d_3) \leftarrow (2, 2, 1)$
2: i ← rastgele seçim({1, 2, 3}; $p_1(d_1, d_2, d_3)$, $p_2(d_1, d_2, d_3), p_3(d_1, d_2, d_3)$)
3: Eğer $d_i > D_i$ ise:
4: 2'ye dön
5: Değilse:
6: Her $\bar{x} \in N_{i,d_i}(x)$ için:
7: f($\bar{x}$) < f(x) ise:
8: x ← $\bar{x}$
9 $(d_1, d_2, d_3) \leftarrow (2, 2, 1)$
10: 3'e dön
11: Eğer $(d_1, d_2, d_3) \geq (D_1, D_2, D_3)$ ise:
12: 15'e git
13: Değilse:
14: $d_i \leftarrow d_i + 1$
15: 2'ye dön
16: Çıktı x

3.1.2. Komşuluk Yapıları

Çalışmada $N_{1,d_1}, N_{2,d_2}, N_{3,d_3}$ olarak tanımlanan 3 farklı komşuluk yapısı tasarlanmıştır. Bu komşuluk yapılarında kullanılan d_1, d_2, d_3 komşuluk dereceleri, algoritmanın başlangıçta daha küçük komşuluklarla arama yapmasını, ardından giderek büyüyen komşuluk yapılarıyla daha geniş bir çözüm uzayını keşfetmesini sağlamaktadır. Her bir komşuluk derecesinde, ilgili komşuluk yapısına göre üretilen tüm alternatif komşu çözümler taranır. Bu, algoritmanın daha derinlemesine bir keşif yapabilmesini ve daha iyi çözümler bulma olasılığını artırır.

N_{1,d_1} - Aynı makinede iş değişimleri:

Her bir makine için, eğer o makinede en az d_1 adet iş var ise o makinedeki işlerin $\binom{\sum_{j=1}^{n} A_{j,i}}{d_1}$ adet d_1'li kombinasyonlarının tümünü ele alır. Her bir kombinasyon için, işlerin makinedeki pozisyonları arasında yer değişimleri yaparak mevcut sıralamadan farklı tüm permütasyonlar $(d_1! - 1)$ denenir.

Örneğin, d_1=3 komşuluk derecesi için, i. makinede seçilen kombinasyondaki işler j_1 , j_2 ve j_3 sırasıyla s_1 , s_2 ve s_3 pozisyonlarında işleniyor olsun. Bu işlerin (2-1-3) permütasyonuna göre yer değiştirmesi durumunda j_1 işi s_2 pozisyonunda, j_2 işi s_1 pozisyonunda ve j_3 işi mevcut konumunda işlenir. Yeni sıralama ile i. makinede toplam gecikme ve erken teslim süresi değerlendirilir. Yeni durumda mevcuttan düşük bir gecikme ve erken teslim süresi söz konusu ise hedef değeri geliştiren bir komşu elde edilmiştir.

N_{2,d_2} – Makineler arası iş değişimleri:

N işin, makine bakmaksızın $\binom{N}{d_2}$ adet d_2'li kombinasyonların tümü ele alınır. Her bir kombinasyonda seçili işlerin mevcut makinelerindeki pozisyonları arasında tüm farklı permütasyonlar denenir.

Örneğin $d_2 = 3$ komşuluk derecesi için, seçilen kombinasyondaki işler j_1 , j_2 ve j_3 olsun. Bu işler sırasıyla i_1 , i_2 ve i_3 makinelerinde s_1 , s_2 ve s_3 sıralarında işleniyor olsun. (1-3-2) permütasyonuna göre bu işlerin yer değiştirmesi durumda, j_1 mevcut atama ve konumunda kalır, j_2 i_3 makinesinde s_3 sırasında işlenir ve j_3 ise i_2

makinesinde s_2 sırasında işlenir. Yani işlerin işlem yerleri (1-3-2) permütasyonuna göre birbirleriyle yer değiştirir.

Ancak bu şekilde elde edilen her permütasyon geçerli bir komşu çözüm olarak tanımlanmayabilir; işlerin uygun olmayan bir makineye atanmasını içeren permütasyonlar geçersiz kabul edilir. Eğer yeni pozisyonlarında tüm işler onları işleyebilen bir makineye atandıysa, yani olurlu bir komşu çözüm elde edildiyse, ve yeni durumda mevcuttan düşük bir toplam gecikme ve erken teslim süresi söz konusu ise hedef değeri geliştiren bir komşu elde edilmiştir.

N_{3,d_3} – Karmaşık yeniden atamalar:

Bu komşuluk, hesaplama açısından en maliyetli olandır.

N adet işin, $\binom{N}{d_3}$ adet d_3'lü kombinasyonlarından her biri ele alınır. Kombinasyondaki her bir işin atanabileceği alternatif makinelere atandıkları tüm durumlar değerlendirilir. $\prod_{l=1}^{d_3} \left(\sum_{i=1}^{m} u_{j_l,i} - 1 \right)$ atama alternatifi söz konusudur. İş eksilen fakat yeni iş atanmayan makineler için, makine içindeki toplam gecikme ve erken teslim süreleri güncellenir. Yeni işler atanan makineler için ise, bu yeni işlerin makinedeki tüm pozisyonlara yerleşimi ve bu pozisyonlardaki tüm permütasyonları denenir, gecikme ve erken teslimi en azaltan yerleşim kabul edilir.

Örneğin; d_3=2 komşuluk derecesi için, seçilen kombinasyondaki işler j_1 ve j_2 sırasıyla i_1 ve i_2 makinelerinde işleniyor olsun. İşlerin işlenebileceği alternatif makineler j_1 için i_3, j_2 için i_1 ve i_3 olsun. İşlerin işlenebileceği alternatif makine atama kombinasyonlarından seçilen durum, j_1 için i_3, j_2 içinde i_3 olsun. Yeni durumda i_1 ve i_2 makinelerinde iş eksilmiş ve yeni iş atanmamış olduğu için bu makinelerin toplam gecikme ve erken teslim süreleri güncellenir. i_3 makinesine j_1 ve j_2 işleri atanmış olduğu için, yeni atanan bu işlerin i_3 makinesinde yeni pozisyonları için tüm permütasyonlar denenir ve en düşük toplam gecikme ve erken teslim süresini sağlayan sıralama seçilir. Değişikliğe uğrayan makineler üzerinden toplam gecikme ve erken teslim güncellendiğinde mevcut çözümden daha düşük bir hedef değer bulunuyorsa hedef değeri geliştiren bir komşu elde edilmiştir.

3.1.3. Yöntemin Problem Türevlerinde Uygulanabilirliği ve Esnekliği

Yerel arama sezgisel yöntemleri, bir başlangıç çözümü seçmek, bu çözümden iniş yönü bulmak ve bu yön boyunca komşuluk içinde hareket etmek gibi adımları içerir [55]. Bu çalışmada kullanılan yerel arama yöntemleri, geniş ve kapsamlı aramalar gerçekleştirir. Düşük komşuluk derecelerinden başlanarak, belirlenen olasılıklarla rastgele seçilen komşuluk yapısı üzerinden çözümler keşfedilir. Yeni bir iyileştirme sağlandığında, bu çözüm mevcut en iyi çözüm olarak kabul edilir ve komşuluk dereceleri sıfırlanır. İyileştirme sağlanamazsa, komşuluk derecesi artırılır, böylece algoritma her iterasyonda daha geniş bir çözüm kümesi üzerinde arama yapar.

Ele alınan problemde özel durum ve ayrıntıların çoğu tanıma dahil edildiği için, literatürde ele alınan deterministik paralel makine çizelgeleme problemlerinin çoğu, geliştirilen DKA algoritması ile çözülebilmektedir. Özellikle, makinelerin başlangıçta çalışmaya hazır olduğu durumlar, ara bakım yapılmadığı durumlar veya makinelerin tüm işleri işleyebilecek kapasitede olduğu durumlar ilgili parametreler tekdüze olarak ilgili duruma göre 0 veya 1 değerlerine ayarlandığında DKA ile doğrudan çözülür. Makineler veya işler özdeş olduğunda işlem süreleri ve kurulum süreleri özdeşliğe uygun şekilde eşit ayarlanabilir. Benzer şekilde, kurulumlar sıra bağımlı olmadığında, kurulum süreleri buna uygun şekilde ayarlanarak önerilen DKA algoritması problem üzerinde doğrudan koşulabilir.

Farklı hedef değerlerin ele alındığı problem türevlerinde, DKA'nın komşuluk yapıları ve yerel arama metotlarıyla doğrudan uyumluluk söz konusudur. Bu durumda, hedef fonksiyon “f(x)” değişiklik gösterse de kısıtlar aynı kaldığı için mevcut komşuluk yapıları geçerli olmaya devam eder. Yeni hedef fonksiyon eklenip yönteme entegre edildiğinde önerilen DKA yeni hedef fonksiyona sahip problem için de verimli bir yaklaşım olarak önerilebilir.

Ek kısıtlar içeren problem türevlerinde, örneğin rijit termin tarihlerinin, öncelik kısıtlamalarının veya işlerin birbirine bağımlılığının söz konusu olduğu durumlarda, DKA'nın komşuluk yapıları yeni kısıtlarla uyumlu hale getirilmelidir. Rassal uzantılarda, örneğin işlem sürelerinin rassal olduğu ve bu rassallığa bağımlı olarak işlem sürelerinin ara süreye denk gelme riskinin kısıtlandığı durumlarda, bu yeni kısıtlar algoritmaya entegre edilebilir. Genellikle, komşuluk yapılarındaki

kombinatoryal tarama yinelemeleri değişmez; ancak önerilen komşuluk yapılarındaki aday çözümlerin bir alt kümesi olurlu komşular olarak kabul edilir. Bu olurlu çözümleri içeren komşuluklar yönteme bir ekleme ile entegre edilebilir: Komşuluk taramasında belli bir aday çözümün yeni kısıtları sağlayıp sağlamadığı bir koşul yapısıyla kontrol edilir. Yeni kısıt eklenmesini gerektiren bu tür durumlarda da önerilen DKA komşuluk yapıları verimli arama için kolaylıkla uyarlanabilecek bir çözüm yöntemidir.

Böylelikle, bu çalışmada önerilen DKA algoritması, paralel makine çizelgeleme problem türevlerinin çeşitliliğine uyum sağlayabilen, esnek ve geniş uygulama alanlarına sahip bir yöntem olarak değerlendirilmektedir.

4. HESAPLAMALI SONUÇLAR

Önerilen DKA algoritmasının performansını değerlendirmek için kullanılan test problemleri önerilen sentetik veri oluşturma ile elde edilmiştir. Test problemleri ve sentetik veri oluşturma prosedürü Bölüm 4.1 altında açıklanmıştır. Bölüm 4.1.2'de test problemleri matematiksel model üzerinde uygulanmış ve problemin NP-Zor problem olduğu kanıtlanmıştır. Bölüm 4.2'de önerilen değişken komşuluk arama algoritması ve matematiksel programlama yöntemleri kıyaslanmıştır ve önerilen algoritmanın başarısı gösterilmiştir.

4.1. Test Problemleri

Önerilen yöntemin performansını test etmek için sentetik veri oluşturma prosedürü ile elde edilen gerçek dünya verilerini iyi bir şekilde temsil eden test problemleri kullanılmıştır. Sentetik veri kullanımı, farklı senaryoları, çeşitli veri dağılımlarını ve nadir durumları simüle ederek yöntemin performansını test etmek için geçerli bir yaklaşımdır.

4.1.1. Önerilen Sentetik Veri Oluşturma Prosedürü

Sentetik veri, kontrollü olarak seçilmiş parametreler ile sistematik bir yaklaşım kullanılarak üretilmektedir. Tablo 4.1 ve 4.2'de sentetik veri oluşturmada kullanılan parametreler ve dağılımlar sunulmuştur. Bu parametreler ve dağılımlar, sentetik veri oluşturma prosedürü içinde SBHSİPMÇP'ne uygun test problemleri üretmek için kullanılmaktadır. Her bir parametre ve dağılım, oluşturulan test problemlerinin gerçek dünya durumlarına benzerliğini veya spesifik senaryoları simüle etme yeteneği kazandırmaktadır.

Tablo 4.1: Sentetik veri oluşturma parametreleri.

Parametre	Değer	Açıklama
M	{3,5,7,9}	Makine sayısı
N	{10,15,20,25}	İş sayısı
T	1000	Planlama ufku
α	0.7	Makine uygunluk katsayısı
β	0.35	İşlem süresi katsayısı

η	0.03	İlk işin hazırlık süresi katsayısı
γ	0.08	Ardışık işler arası hazırlık süresi katsayısı
ζ	0.05	Makinelerin başlangıç duruş süresi katsayısı
δ	0.06	Makinelerin duruş süresi katsayısı
θ	0.15	Teslimat için boşluk
CV_{JM}	0.1	Bir işin farklı makinelerde işlem sürelerindeki değişkenliğin katsayısı
CV_M	0.15	Makineler içinde işler için işlem sürelerindeki değişkenliğin katsayısı
CV_P	0.18	İşlem sürelerinin değişim katsayısı
CV_{S_0}	0.3	İlk hazırlık sürelerinin değişim katsayısı
CV_S	0.3	Ardışık işler arası hazırlık sürelerinin değişim katsayısı
CV_R	0.2	Makinelerin başlangıçtaki duruş sürelerinin değişim katsayısı
CV_D	0.15	Makinelerin duruş sürelerinin değişim katsayısı

Tablo 4.1'de tanımlanan parametre değerleri altında rastgele veri oluşturmak için bazı olasılık dağılımlarından yararlanılmıştır. Örneğin, her bir işin belirli bir makineye atanıp atanmayacağını belirleyen u_{ji} matrisinin oluşturulmasında Bernoulli dağılımı kullanılmıştır. Üretilen rasgele bir sayının α değerinden küçük olması durumunda u_{ji} parametresinin 1 değerini, diğer durumda 0 değerini almasını sağlamaktadır. Her bir işin belirli bir makinedeki işlem süresini temsil eden p_{ji} parametresi değerlerinin üretilmesinde Log-Normal dağılım kullanılmıştır. $(M * T / N)$ ifadesi planlama süresince işlem sürelerinin alabileceği en büyük ortalama değeri ifade ederken, işlem süresinin hesaplanmasında bu değerin β parametresi ile kontrol altına alınması sağlanmıştır. β parametresi, işlem süresi büyüklüğünü belirlenmesini sağlarken, işlem süreleri arasındaki değişikliği CV_P varyasyon katsayısı ile sağlanmaktadır. Ayrıca belirlenen işlem sürelerinin farklı makinelerde işlenmesi sonucunda değişkenlik göstermesi için Log-Normal dağılım ve $\mu = p_{ji}/MC_i$, $\sigma = p_{ji} * CV_{jm}$ dağılım parametreleri ile sağlanmaktadır.

Tablo 4.2: Sentetik veri oluşturma parametreleri ve dağılımları.

Parametre	Dağılım	Ortalama (μ)	Standart Sapma (σ)
u_{ji}	Bernoulli dağılımı	α	$\sqrt{\alpha * (1-\alpha)}$
p_{ji}	Log-Normal dağılım	$(M * T / N) * \beta$	$\mu * CV_P$

S_{jki}	Log-Normal dağılım	$\frac{\mathrm{T} * \mathrm{M} * \gamma}{N - M}$	$\mu * CV_S$
S_{0ki}	Log-Normal dağılım	$\frac{T * M * \eta}{M}$	$\mu * CV_{S_0}$
R_i	Log-Normal dağılım	$T * \zeta$	$\mu * CV_R$
$*DM_i$	Log-Normal dağılım	$\min\left(T * \delta, \frac{T}{2}\right)$	$\mu * CV_D$
MC_i	Log-Normal dağılım	1	CV_M

*(DM_i=$FM_i - SM_i$, MC_i: i. makinenin kapasitesi)

Bu sistem, işleri rastgele bir sıraya yerleştirir ve her bir işi, atanabilecek uygun makinelerden birine göre yönlendirir. Her makinede işlerin atanma sırasına göre makinenin ilk duruş süresi, ilk hazırlık süresi, işlem süresi, ara hazırlık süresi ve işlem süresi art arda gelecek şekilde sıralanır. Her makine için belirlenen duruş süreleri ilk işin hazırlık süresinden önce veya bir makinenin işinin tamamlanmasından sonra gelen boşluklardan birine rastgeleme yoluyla yerleştirilir. İşler arasında oluşan zaman aralıklarını ve beklemeleri temsil eden boşluklar tanımlanmıştır. Bu boşluklar, aşağıda formülü verilen boşluk katsayısına göre belirlenmektedir.

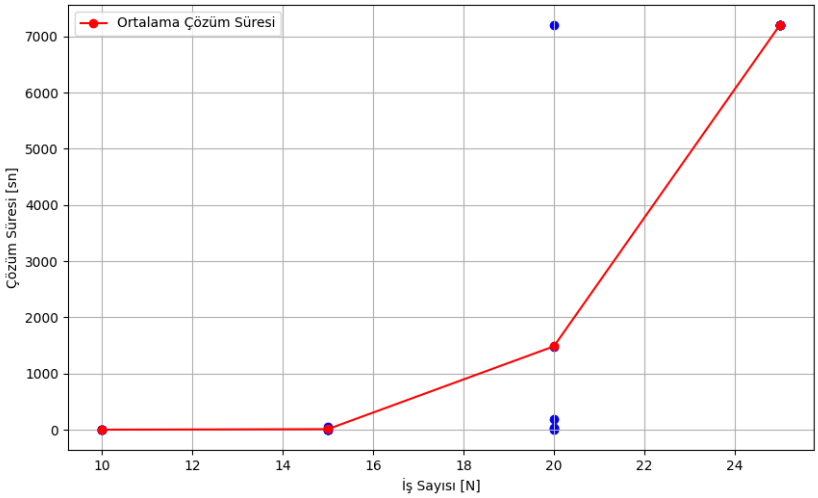
$$Boşluk\ katsayısı = \max(0\ , 1 - alpha - beta - eta - gamma - delta - zeta) \tag{4.1}$$

4.1.2. Sentetik Veri ile Matematiksel Model Çözümü

Bölüm 2.2.1'de verilen matematiksel model, sentetik veri oluşturma prosedürü ile makine sayısı M'nin [3, 5, 7, 9] ve iş sayısı N'nin [10, 15, 20, 25] değerleriyle oluşturulan kombinasyonlar kullanılarak 6 çeşit ve toplamda 96 farklı test problemi ile çözülmüştür. Her bir problemin çözüm süresi, en iyi çözüme olan uzaklığı ve düğüm sayısı gibi veriler analiz edilip matematiksel modelin sağlamlığı test edilmiştir.

Elde edilen sonuçlara göre, problemde iş sayısının artması, modelin karmaşıklığını doğrudan etkilemektedir. Bunu düğüm sayısı ve çözüm sürelerinin artışından gözlemlemek mümkündür. Ancak problemde makine sayısının artması, çözüm sürelerinde düzenli bir artış veya azalışa neden olmamıştır. Şekil 4.1'de, makine sayısı 3 olarak sabit tutulurken iş sayısı 10, 15, 20 ve 25 değerlerini alan toplam 24 farklı veri setinin çözüm süreleri grafik üzerinde gösterilmiştir. Modelin durma kriteri,

çözüm süresinin 7200 saniyeye ulaşması olarak belirlendiği için çözüm süresi değeri 7200 saniye olan veri setlerinin en iyi çözüme ulaşmadıkları bilinmektedir. Dolayısıyla, bu veri setlerinin çözüm süreleri, en iyi çözüme ulaşmak için gereken süreyi yansıtmayabilir.



Şekil 4.1: M=3, N=[10, 15, 20, 25] olan problemlerin çözüm süreleri.

96 farklı test probleminde, 7200 saniye çözüm süresi içerisinde sadece 18 tanesi en iyi çözüme ulaşılamamıştır. Bu 18 problemin iş sayıları 20 ve 25 olduğu belirlenmiştir. Bu bulgular, iş sayısı 20 ve üstü olan sıra bağımlı hazırlık süreli ilişkisiz paralel makine problemleri için uzun çözüm sürelerine ihtiyaç duyulduğu göstermektedir. Dolayısıyla, bu tür problemlerin NP-zor problem sınıfında yer aldığı ve kesin çözüm yöntemlerinin bu problemleri etkili bir şekilde çözmekte yetersiz kalabileceği sonucuna varılabilir. Bu durum, sezgisel ve metasezgisel yöntemlerin bu tür problemlerde daha iyi performans sağlayabileceğini ve pratikte daha verimli çözümler sunabileceğini ortaya koymaktadır.

4.2. DKA Algoritması için Deneysel Sonuçlar

Bu bölümde, önerilen DKA algoritmasının performansını test etmek için yapılan deneylerin sonuçları sunulmaktadır. Elde edilen sonuçlar, matematiksel programlama (MP) model çözüm sonuçları ile kıyaslanmıştır. Çalışmada, makine sayısı M = {3, 5, 10, 15 ve 20}, iş sayısı N = {5, 10, 20, 30 ve 60} ve örneklem büyüklüğü n = 6 olmak üzere toplamda 150 farklı problemden oluşan bir sentetik veri seti kullanılmıştır. DKA

ve MP modeli çözümü arasındaki hesaplamalı testler Python 3.10 sürümü üzerinde çalıştırılmış, MP çözümü Gurobi Optimizer 11.0.0 sürümü ile gerçekleştirilmiştir. Donanım olarak, 24 fiziksel çekirdek üzerinde 48 mantıksal işlemci sunan AMD Ryzen Threadripper 3960X işlemci kullanılmıştır. DKA algoritmasında eşzamanlılık işlenmediğinden, Gurobi yazılımı karşılaştırılabilirliği sağlamak adına tek bir mantıksal işlemci ile sınırlandırılmıştır. MP çözücüsü için yakınsama toleransı %0.01 olarak ayarlanmıştır. Her iki yaklaşım için de maksimum çözüm zaman sınırı 5000 saniye olarak belirlenmiştir.

DKA komşuluk yapılarının derece üst limitleri aşağıdaki gibi ayarlanmıştır.

N_{1,d_1}(aynı makinede iş değişimleri) komşuluk yapısı için $D_1 = 5$

N_{2,d_2}(makineler arası iş değişimleri) komşuluk yapısı için $D_2 = 5$

N_{3,d_3}(karmaşık yeniden atamalar) komşuluk yapısı için $D_3 = 4$

Komşuluk yapısı seçiminde kullanılan olasılık fonksiyonu, uygulanacak olan dereceye göre adaptif olarak değişmektedir. Bu olasılık fonksiyonu, denklem (4.2)'de ve ağırlık değerleri Tablo 4.3'te verilmiştir.

$$p_i(d_1, d_2, d_3) = w_i\ (d_i)/(\ w_1\ (d_1) + w_2\ (d_2) + w_3\ (d_3)),\ \text{i=1,2,3} \qquad (4.2)$$

Tablo 4.3: Komşuluk yapısı seçimde kullanılan olasılık fonksiyonunun komşuluk derecelerine göre ağırlık değerleri.

w_i	d_1	d_2	d_3
1	-	-	12
2	12	12	6
3	12	12	0.25
4	12	6	0.1
5	12	1	-

Denklem (4.2)'deki ifade her bir komşuluk yapısının seçilme olasılığını hesaplarken, ilgili komşuluk yapısının derecesinin diğer komşuluk yapılarının derecelerinin toplamına bölünmesiyle elde edilir. Örneğin komşuluk dereceleri d_1=3, d_2=4 ve d_3=1 iken yeni seçilecek olan komşuluk yapılarından $N_{3,2}$ komşuluk yapısının seçim olasılığı $p_3(3, 4, 1) = 6$ / (12+6+12) = 0,2 olmaktadır.

Bu ağırlık değerleri, komşuluk yapısının ve derecesinin keşfettiği çözüm uzayı büyüklüğüne göre belirlenmiştir. N_{3,d_3} komşuluk yapısı diğer komşuluk yapılarından hesaplama yükü açısından daha maliyetlidir ve yüksek komşuluk derecelerinde bu maliyet hızla artmaktadır. Bu nedenle maliyetli komşuluk yapılarında komşuluk dereceleri arttıkça komşuluk yapısının seçim olasılıkları azalmaktadır.

Önerilen DKA algoritması ve MP yöntemlerinin en iyi çözüme yakınsama değerleri Tablo 4.4'de verilmiştir. Belirli bir çözüm süresi sonunda iki yönteminde benzer değerlere yakınsadığı ancak DKA algoritmasının genel olarak daha hızlı yakınsama eğilimi gösterdiği gözlemlenmiştir.

Tablo 4.4: DKA ve MP yöntemlerinin deneysel sonuçları.

M	N	MP					DKA				
		ORT	SS	SH	GAA	GAÜ	ORT	SS	SH	GAA	GAÜ
3	5	794	510	208	259	1329	698	302	123	381	1015
3	10	1215	255	104	947	1483	1232	253	103	966	1498
3	20	2402	526	215	1849	2954	2440	566	231	1847	3034
3	30	2906	596	243	2281	3532	2938	522	213	2390	3486
3	60	6789	1094	446	5641	7937	6364	974	398	5342	7386
5	10	1718	263	107	1441	1994	1748	258	105	1478	2018
5	20	2182	278	114	1890	2474	2329	285	116	2029	2628
5	30	3504	490	200	2990	4019	3556	561	229	2967	4145
5	60	7216	880	359	6293	8139	6752	718	293	5998	7505
10	20	2471	612	250	1830	3113	2705	699	285	1971	3439
10	30	3188	694	283	2460	3916	3253	712	291	2506	4001
10	60	6740	953	389	5740	7740	6445	1088	444	5303	7586
15	20	2217	461	188	1733	2701	2394	388	158	1987	2800
15	30	3170	941	384	2183	4157	3468	855	349	2571	4366
15	60	6959	531	217	6403	7516	6654	569	232	6057	7251
20	30	3560	535	218	2999	4121	3569	858	350	2669	4469
20	60	7437	641	262	6764	8110	7015	507	207	6483	7547

*M: Makine sayısı; N: iş sayısı; MP: matematik program çözümü; DKA: değişken komşuluk arama ile çözüm; her problem boyutunda 6 sentetik veri seti (örneklem büyüklüğü n=6) ile çözümde hedef değerlerin ORT: ortalaması, SS: standart sapması, SH: standart hatası, GAA: güven aralığı alt değeri, GAÜ: güven aralığı üst değeri.

Problem boyutlarına göre iki yöntemin çözüm süresi ve en iyi çözüme yakınsama yeteneğine göre detaylı karşılaştırılması şu şekildedir:

(M = 3, N = 20): DKA, MP'ye göre daha hızlı yakınsama eğilimi göstermektedir (Şekil Ek A.4.1). Örneğin, 5. ve 6. denemelerde, DKA tarafından 20 saniyede ulaşılan hedef değerler, MP tarafından ancak 40 saniyede yakalanabilmiştir. MP, DKA'nın ulaştığı

değerlere yakınsamak için daha uzun süre gerektirmektedir. Deneme 2 ve 3'te, DKA’nın üst seviyelerin kısıtlanması nedeniyle hızlı bir şekilde sonuca ulaşırken, MP iyileşmeye devam ederek DKA’nın sonlandığı hedef değerlerden %5’e kadar daha iyi değerler elde edebilmektedir.

(M = 3, N = 30): Gözlemlenen desenler önceki duruma benzerlik göstermektedir. Ancak, deneme 1 ve 6 gibi durumlarda, MP, DKA’nın ulaştığı hedef değerlere ulaşamamaktadır (Şekil Ek A.4.2). Bu, DKA’nın erken yakınsama açısından MP'ye kıyasla tutarlı bir şekilde daha iyi performans gösterdiğini vurgulamaktadır.

(M = 5, N = 20) ve (M = 5, N = 30): DKA’nın daha erken yakınsama eğilimi devam etmektedir (Şekil Ek A.4.3). Bu senaryolarda, MP bazen DKA’nın performansına yetişmekte zorlanmaktadır, bu da DKA’nın daha hızlı sonuçlar elde etme verimliliğini yeniden teyit etmektedir (Şekil Ek A.4.4).

(M = 5, N = 60): DKA tutarlı bir şekilde daha erken hedef değerlere ulaşmakta ve bu avantajını 5000 saniyelik hesaplama süresi boyunca sürdürmektedir. Bu fark, (M = 5, N = 60) durumu için daha da belirgin hale gelmektedir (Şekil Ek A.4-5).

(M = 10, N = 60): Makine sayısı arttıkça, DKA, önceki durumlara kıyasla hedef değerlerde daha erken yakınsamalar gösterdiği ve daha iyi sonuçlara ulaştığı gözlemlenmiştir. Ancak, MP, bu senaryoda önceki durumlara kıyasla farkı kapatma konusunda daha etkili olabilmektedir, ancak yine de DKA üstünlüğünü korumaya devam etmektedir (Şekil Ek A.4.6).

(M = 15, N = 30): İş ve makine sayısının neredeyse eşit olduğu durumda, MP, DKA'dan daha hızlı bir yakınsama göstererek problemi 50-100 saniye içinde çözmektedir (Şekil Ek A.4.7). Ancak, bu durum, analiz edilen diğer senaryolara kıyasla daha az rastlanmaktadır.

(M = 15, N = 60) ve (M = 20, N = 60): İşlerin makinelerin sayısına oranının daha yüksek olduğu durumlarda, DKA’nın erken yakınsama gösterdiği daha belirgin hale gelmektedir (Şekil Ek A.4.8 ve Şekil Ek A.4.9). MP, sonunda benzer performansa ulaşsa da, DKA ile aynı hedef değerlere ulaşması önemli ölçüde daha uzun sürmektedir.

Genel olarak, DKA tutarlı bir şekilde erken yakınsama göstererek MP'ye kıyasla daha üstün performans sergilemektedir, özellikle iş/makine oranı yüksek olduğunda. MP, zamanla iyileşmeler gösterip bazen nihai hedef değerler açısından DKA'yı geçse de,

bunu yapmak için genellikle daha fazla hesaplama süresine ihtiyaç duymaktadır. Bu analiz, çeşitli senaryolarda, özellikle iş/makine oranı yüksek olan durumlarda, DKA'nın verimliliğini ve etkinliğini vurgulamaktadır.

DKA'nın buluduğu en iyi hedef değerin %1 fazlasına yakınsama süreleri dikkate alındığında, M = 3 ve N = 10 için, MP çözümü iki örnekte %58 ve %80 daha hızlıdır. Diğer durumlarda, yakınsama süresi %24 ile %1266 arasında değişerek daha uzun sürmektedir. M = 3 ve N = 20 olduğunda, bu aralık dramatik bir şekilde artmakta ve MP'nin yakınsama süresi %114 ile %7792 arasında uzamaktadır, bu da DKA'nın tüm durumlar için daha hızlı yakınsandığını göstermektedir. M = 3 ve N = 30 için, MP için süre %31 ile %3568 arasında daha uzun sürmektedir (Tablo Ek A.4.1).

M = 3, N = 60 ve M = 5, N = 60 için, böyle bir karşılaştırma mevcut değildir çünkü MP çözümü zaman sınırları içinde karşılaştırılabilir hedef değerlere ulaşamamaktadır. M = 10 ve N = 20 için yaklaşımlar daha karşılaştırılabilir hale gelmekte ve dengeler hafifçe MP çözümünün lehine kaymaktadır. Bu eğilim, makine sayısının iş sayısına göre nispeten yüksek olduğu durumlarda, M = 15 ve N = 30 ile M = 20 ve N = 30'da görüldüğü gibi devam etmektedir (Tablo Ek A.4.2).

Ancak, göreceli olarak daha fazla iş olduğunda, örneğin M = 15 ve N = 60'da, DKA MP çözümüne kıyasla çok daha hızlı yakınsamaktadır. M = 20 ve N = 60 için, MP çözümü DKA tarafından ulaşılan hedef değerin %5 daha kötüsüne ulaşamadığından genellikle bir karşılaştırma mümkün değildir. Göreceli olarak daha büyük durumlar olan M = 15 ve 20, N = 60 için, DKA'nın en iyi hedef değerlerinden %10 daha kötüye ilk ulaşıldığında, karşılaştırılabilecek beş örnekte (biri hedef değere ulaşamadığı için karşılaştırılamaz), MP'nin yakınsama süresi sırasıyla %829 ve %445 daha uzundur (Tablo Ek B-2).

Özetle, özellikle iş sayısı makine sayısına göre arttıkça DKA çok daha hızlı yakınsamaktadır. MP çözümleri, daha küçük problem örneklerinde veya makine sayısı iş sayısına yakınken daha hızlı olabilmektedir, ancak genellikle daha büyük, apaçık çözüme sahip olmayan karmaşık senaryolarda DKA çok daha hızlı yakınsamaktadır.

5. SONUÇLAR

Sıra bağımlı hazırlık süreli ilişkisiz paralel makine problemi, endüstriyel alanlarda sıkça karşılaşılan ve literatürde üzerinde yoğun çalışmalar yapılmış önemli bir NP-zor problemdir. Bu çalışmada, N adet işin M adet makineye toplam gecikme ve erken teslim sürelerini en aza indirgeyecek şekilde atanması problemine odaklanılmıştır. Problem, makinelerin kullanılmayan zaman dilimleri olduğu varsayımı altında ele alınmış ve önerilen sentetik veri oluşturma prosedürü ile üretilen test problemleri üzerinde deneyler yapılmıştır.

Çalışmada önerilen DKA algoritması, işlerin makineler arasında optimal bir şekilde atanması ve sıralanması amacıyla etkili bir çözüm sunmuştur. Deneysel sonuçlar, DKA'nın matematiksel programlama ile karşılaştırıldığında, çözüm süresi ve en iyi çözüme yakınsama kapasitesi açısından MP çözümüne göre verimliliği ortaya çıkmıştır. Özellikle, iş sayısının makine sayısına göre fazla olduğu fakat göreceli olarak küçük, M = 3 ve N = 20 boyutlu problemlerde, DKA'nın ulaştığı hedef değerlere MP'nin iki katı sürede yaklaşabildiği; M = 3 ve N = 30 için, MP'nin %31 ile %3568 arasında daha uzun sürede yakınsadığı; ve M = 15, N = 60 ve M = 20, N = 60 gibi problem boyutunun büyüdüğü durumlarda DKA'nın sırasıyla %829 ve %445 gibi yüksek oranlarda daha hızlı sonuçlar elde ettiği belirlenmiştir.

Sonuç olarak, DKA algoritması, özellikle iş/makine oranının yüksek olduğu karmaşık ve büyük ölçekli problemlerde, geniş ve esnek çözüm uzayında arama yapabilme yeteneği sayesinde, mevcut matematiksel programlama yöntemlerine kıyasla önemli bir avantaj sağlamaktadır. Bu çalışma, algoritmanın pratik uygulamaları için potansiyelini ve problemlerin çözümünde verimlilik kazanımı sunabileceğini vurgulamaktadır.

KAYNAKÇA

[1] Morton T. E., ve Pentico D. W., (1993), “Heuristic Scheduling Systems”, John Wiley & Sons, New York, USA.

[2] Baker K. R., Trietsch D., (2019), “Principles of Sequencing and Scheduling”, Second Adition, John Wiley & Sons.

[3] Özçelik F., Saraç T., (2011), “Sıra bağımlı hazırlık süreli iki ölçütlü tek makine çizelgeleme problemi için sezgisel bir çözüm yöntemi”, Endüstri Mühendisliği Dergisi, 22(4), 48-57.

[4] Pinedo, M. L., (2018), “Scheduling: Theory, Algorithms, and Systems”, Springer.

[5] Alharkan I. M., (2005), “Algorithms for Sequencing and Scheduling”, Industrial Engineering Department, King Saud University, Riyadh, Saudi Arabia.

[6] Eren T., Güner E., (2002), “Tek ve Paralel Makinalı Problemlerde Çok Ölçütlü Çizelgeleme Problemleri İçin Bir Literatür Taraması”, Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi, 17(4).

[7] Pinedo M. L., (2002), “Scheduling: Theory, Algorithms, and Systems”, Second Edition, Prentice Hall, New Jersey, USA.

[8] Baker K. R., Trietsch D., (2009), Principles of Sequencing and Scheduling”, First Edition, John Wiley & Sons.

[9] Lee C. Y., Lei L., Pinedo M. L, (1997), “Current trends in deterministic scheduling”, Annals of Operations Research, 70, 1-41.

[10] Kılıç G., (2023), “Sıra bağımlı ilişkisiz paralel makine çizelgeleme problemi için yeni bir sezgisel algoritma önerisi”, Yayımlanmamış doktora tezi, Pamukkale Üniversitesi, Denizli.

[11] Furugi A., (2021), “Makine uygunluk kısıtlaması ve sıra bağımlı kurulum süresi ile özdeş olmayan paralel makine çizelgeleme problemi için tabu arama algoritması”, Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, 36(3), 1539-1550.

[12] Cheng T., Sin C., (1990), “A state-of-the-art review of parallel-machine scheduling research” Eur J Oper Res, 47(3), 271–292.

[13] Chen B., Potts C. N., Woeginger G. J., (1998). “A review of machine scheduling: complexity, algorithms and approximability”, In: Handbook of combinatorial optimization, Springer, 1493–1641.

[14] Mokotof E., (2001), “Parallel machine scheduling problems: a survey” Asia-Pac J Oper Res, 18.

[15] Gedik R., Kalathia D., Egilmez G., Kirac E., (2018), “A constraint programming approach for solving unrelated parallel machine scheduling problem”, Computers & Industrial Engineering, 121, 139-149.

[16] Gomes F. R. A., Mateus G. R., (2017), “Improved Combinatorial Benders Decomposition for a Scheduling Problem with Unrelated Parallel Machines”, Journal of Applied Mathematics, 2017, 1-10.

[17] Tran T. T., Araujo A., Beck J. C., (2016), “Decomposition Methods for the Parallel Machine Scheduling Problem with Setups” INFORMS Journal on Computing, 28(1), 83-95.

[18] Naderi B., Tavakkoli-Moghaddam R., Sadati A., Mohammadi M., (2017), “Solving a New Multi-objective Unrelated Parallel Machines Scheduling Problem by Hybrid Teaching-learning Based Optimization”, International Journal of Engineering, 30 (2), 224-233.

[19] Saraç T., Özçelik F., (2023), “A matheuristic algorithm for multi-objective unrelated parallel machine scheduling problem Çok amaçli ilişkisiz paralel makine çizelgeleme problemi için bir matsezgisel algoritma”, Journal of the Faculty of Engineering and Architecture of Gazi University, 38 (3), 1953-1966.

[20] Moonsri K., Sethanan K., Sangsawang C., (2015), “Metaheuristics for Scheduling Unrelated Parallel Machines with Sequence-Dependent Setup Time and Machine Eligibility”, Chiang Mai Univ. J. Nat. Sci., 14(4), 431-446.

[21] Stefanello F., Araújo O., Muller F. M., Zanetti M., (2015), “MIP–Based Neighborhood Search for the Unrelated Parallel Machine Scheduling Problem With Sequence and Machine-Dependent Setup Times,” International Journal of Advanced Manufacturing Technology, 7(3), 506-523.

[22] Rego M. F., Souza M. J. F., (2020), “A Hybrid Algorithm for the Unrelated Parallel Machine Scheduling Problem”, Enterprise Information Systems, ICEIS 2019, vol 378, 37-56.

[23] Perez-Gonzalez P., Fernandez-Viagas V., García M. Z., Framinan J. M., (2019), “Constructive heuristics for the unrelated parallel machines scheduling problem with machine eligibility and setup times”, Computers & Industrial Engineering, 131, 131-145.

[24] Liao C. J., Lee C. H., Tsai H. T., (2016), “Scheduling with multi-attribute set-up times on unrelated parallel machines”, International Journal of Production Research, 54(16), 4839-4853.

[25] Ji B., Zhang S., Yu S. S., Zhang D., (2023), "An Enhanced Adaptive Large Neighborhood Search for Unrelated Parallel Machine Scheduling With Sequence Dependent Setup Times," in IEEE Access, 11, 6735-16748.

[26] Fang W., Zhu H., Mei Y., (2022), “Hybrid meta-heuristics for the unrelated parallel machine scheduling problem with setup times”, Knowledge-Based Systems, 241.

[27] Iori M., Locatelli A., Locatelli M., (2023), “A GRASP for a real-world scheduling problem with unrelated parallel print machines and sequence-dependent setup times” International Journal of Production Research, 61(21), 7367-7385.

[28] Jovanovic R., Voß S., (2021), “Implementation of fixed set search to minimize build time on unrelated parallel machines with sequence-dependent setup times”, Appl Softw Comput, 110, 107521.

[29] Arnaout J. P., (2020), “A worm optimization algorithm to minimize the makespan on unrelated parallel machines with sequence-dependent setup times”, Annals of Operations Research, 285 (1), 273-293.

[30] Ezugwu A. E., Akutsah F., (2018), “An improved firefly algorithm for unrelated parallel machine scheduling problem with sequence-dependent setup times” IEEE Access, 6, 54459-54478.

[31] Huang X., Chen L., Zhang Y., Su S., Lin Y., Cao X., (2022), “Improved firefly algorithm with courtship learning for unrelated parallel machine scheduling problem with sequence-dependent setup times”, Journal of Cloud Computing, 11 (9).

[32] Al-qaness M. A. A., Ewees A. A., Abd Elaziz M., (2021), Modified whale optimization algorithm for solving unrelated parallel machine scheduling problems”, Soft Comput, 25, 9545-9557.

[33] Ahmed A. Ewees, Mohammed A.A. Al-qaness, Mohamed Abd Elaziz, Enhanced salp swarm algorithm based on firefly algorithm for unrelated parallel machine scheduling with setup times,Applied Mathematical Modelling,Volume 94,2021,Pages 285-305.

[34] Moser M., Musliu N., Schaerf A., Winter F., (2022), “Exact and metaheuristic approaches for unrelated parallel machine scheduling”, Journal of Scheduling, 25, 507-534.

[35] Ezugwu A. E., (2019), “Enhanced symbiotic organisms search algorithm for unrelated parallel machines manufacturing scheduling with setup times”, Know.-Based Syst., 172, 15-32.

[36] Zeidi J. R., MohammadHosseini S., (2015), “Scheduling unrelated parallel machines with sequence-dependent setup times”, Int J Adv Manuf Technol, 81, 1487-1496.

[37] Caniyilmaz E., Benli B., İlkay M. S., (2014), “An artificial bee colony algorithm approach for parallel machine scheduling unrelated to process set constraints, job sequence-dependent setup times, and deadline”, Int J Adv Manuf Technol, 77 (9–12), 2105-2115.

[38] Jouhari H., Lei D., Al-qaness, M. A. A., Abd Elaziz M., Ewees A. A., Farouk O., (2019), “Sine-Cosine Algorithm to Enhance Simulated Annealing for Unrelated Parallel Machine Scheduling with Setup Times” Mathematics, 7, 1120.

[39] Lin S. W., Ying K. C., (2014), “ABC-based manufacturing scheduling for unrelated parallel machines with machine-dependent and job sequence-dependent setup times”, Computers & Operations Research, 51, 172-181.

[40] Adan J., (2022), A hybrid genetic algorithm for parallel machine scheduling with setup times: A comparative study of metaheuristics on large problem instances”, Journal of Intelligent Manufacturing, 33 (7), 2059-2073.

[41] Diana R. O. M., de França Filho M. F. , de Souza S. R. , de Almeida Vitor J. F., (2015), “An immune-inspired algorithm for an unrelated parallel machines’ scheduling problem with sequence and machine dependent setup-times for makespan minimisation”, Neurocomputing, 163, 94–105.

[42] Avalos-Rosales O., Ángel-Bello F., Alvarez A., (2015), “Efficient metaheuristic algorithm and re-formulations for the unrelated parallel machine scheduling problem with sequence and machine-dependent setup times”, The International Journal of Advanced Manufacturing Technology, 76 (2), 1705-1718.

[43] Haddad M. N., Cota L. P., Souza M. J. F., Maculan N., (2014), “AIV: A heuristic algorithm based on iterated local search and variable neighborhood descent for solving the unrelated parallel machine scheduling problem with setup times”, In ICEIS (1), 376-383.

[44] Diana, R. O., de Souza, S. R., Moacir Filho, F. F., (2018), “A Variable Neighborhood Descent as ILS local search to the minimization of the total weighted tardiness on unrelated parallel machines and sequence dependent setup times”, Electronic Notes in Discrete Mathematics, 66, 191-198.

[45] Abdullah S., Turky A., Nazri M. Z. A., Sabar N. R., (2021), “An Evolutionary Variable Neighbourhood Search for the Unrelated Parallel Machine Scheduling Problem”, in IEEE Access, 9, 42857-42867.

[46] Tozzo, E., Cotrim, S. L., Galdamez, E. V. C., Leal, G. C. L., (2018), “A genetic algorithm and variable neighborhood search for the unrelated parallel machine scheduling problem with sequence dependent setup time”, Acta Scienriarum Technology, 40 (1), 1-13.

[47] Al-Harkan I. M., Qamhan A. A., (2019), “Optimize Unrelated Parallel Machines Scheduling Problems With Multiple Limited Additional Resources, Sequence-Dependent Setup Times and Release Date Constraints”, IEEE Access, 7, 171533-171547.

[48] Avalos-Rosales O., Ángel-Bello F., Alvarez A., Cardona-Valdés Y., (2018), “Including preventive maintenance activities in an unrelated parallel machine

environment with dependent setup times", Computers & Industrial Engineering, 123, 364-377.

[49] Wang M., Pan G., (2019), "A Novel Imperialist Competitive Algorithm With Multi-Elite Individuals Guidance for Multi-Object Unrelated Parallel Machine Scheduling Problem", IEEE Access, 7, 121223-121235.

[50] Lei D., Yang H., (2022), "Scheduling unrelated parallel machines with preventive maintenance and setup time: Multi-sub-colony artificial bee colony", Applied Soft Computing, 125, 109154.

[51] Özçelik F., Saraç T., (2022), "Parallel Machine Scheduling Problem with Unavailable Time Periods and Sequence Dependent Setup Times", Gazi Üniversitesi Fen Bilimleri Dergisi Part C: Tasarım ve Teknoloji, 10 (3), 588-600.

[52] Lei D., Yi T., (2021), "A Novel Shuffled Frog-Leaping Algorithm for Unrelated Parallel Machine Scheduling with Deteriorating Maintenance and Setup Time" Symmetry, 13 (9), 1574.

[53] Hansen P., Mladenovic N., Perez J. A. M., (2010), "Variable neighbourhood search: methods and applications", 4OR, 6, 319-360.

[54] Maniezzo V., Stützle T., Voß S., (2021), "Matheuristics: Algorithms and implementations", EURO Advanced Tutorialson Operational Research, Cham: Springer.

[55] Hansen P., Mladenovic N., Moreno-Perez J. A., (2008), "Variable neighborhood search: methods and applications", 4OR, 6, 319-360.

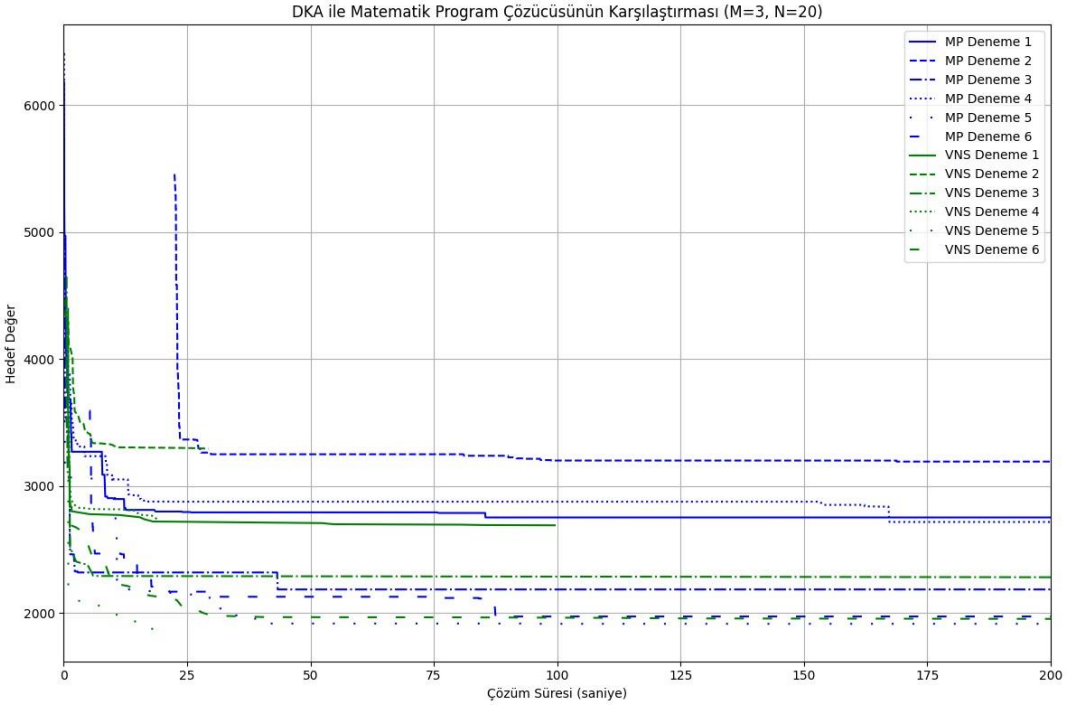
ÖZGEÇMİŞ

Adım Emine Duysak, endüstri mühendisliği dalında lisans eğitimimi 2016-2020 yılları arasında Eskişehir Osmangazi Üniversitesi'nde tamamladım. 2021 yılında başladığım Gebze Teknik Üniversitesi'nde endüstri mühendisliği dalında yüksek lisans çalışmalarımı sürdürüyorum. Araştırma alanlarım arasında yöneylem araştırması, modelleme ve optimizasyon bulunmaktadır.

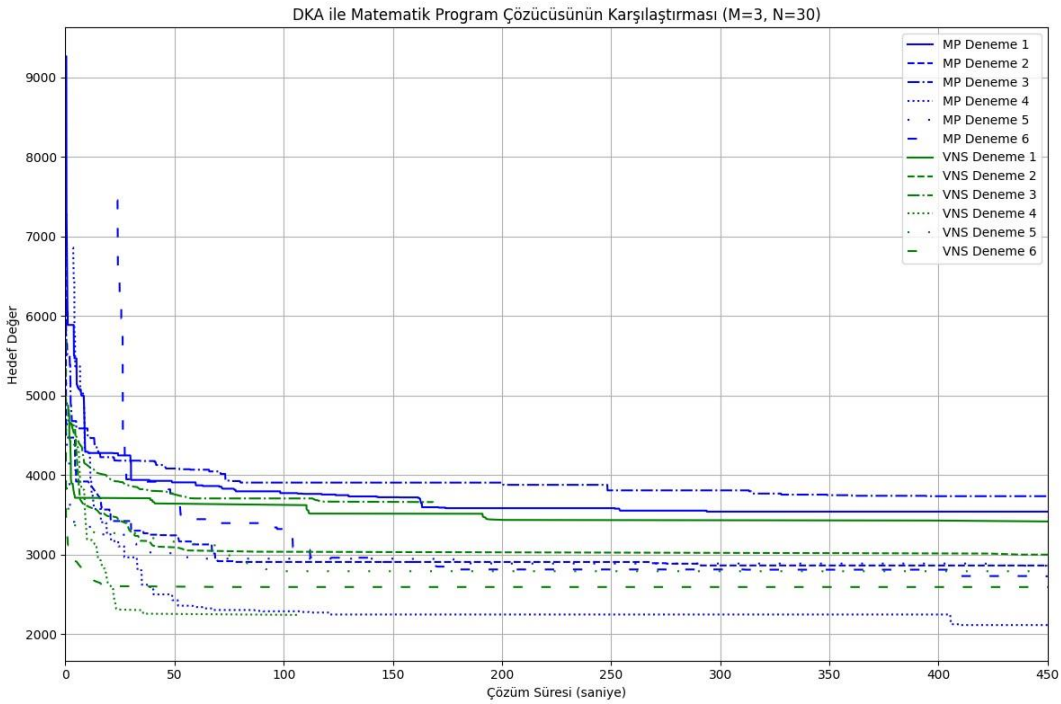
Akademik geçmişim boyunca farklı sektörlerde deneyim kazandım. 2020-2022 yılları arasında Mageba Yapı Sanayi ve Ticaret A.Ş.'de satın alma uzmanı olarak görev yaptım ve 2022 yılından beri Şişecam Camiş Ambalaj Sanayi A.Ş.'de üretim planlama mühendisi olarak çalışmaktayım. 2022 yılında yayınladığım "Teslim Zamanlarının Dikkate Alındığı 1,5 Boyutlu Kesme ve Ana Malzeme Seçimi Problemi için Matematiksel Algoritma" başlıklı bir makale bulunmaktadır.

EKLER

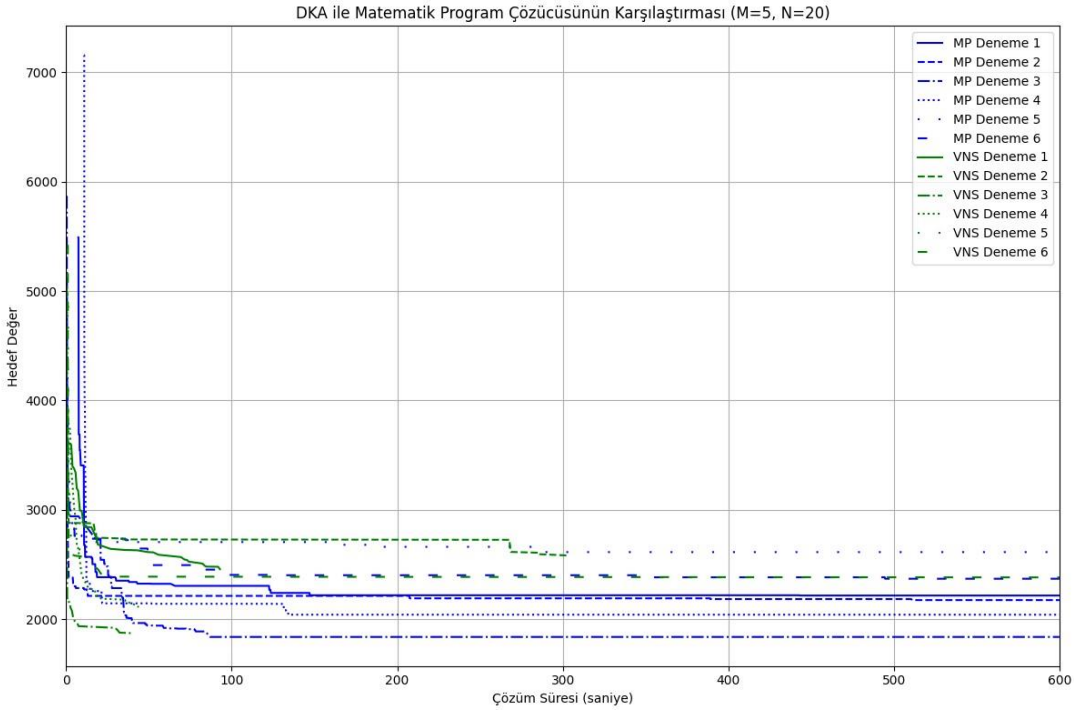
Ek-A: Problem boyutuna göre DKA ve MP yöntemlerinin deneysel sonuçlarının grafikleri ve tabloları



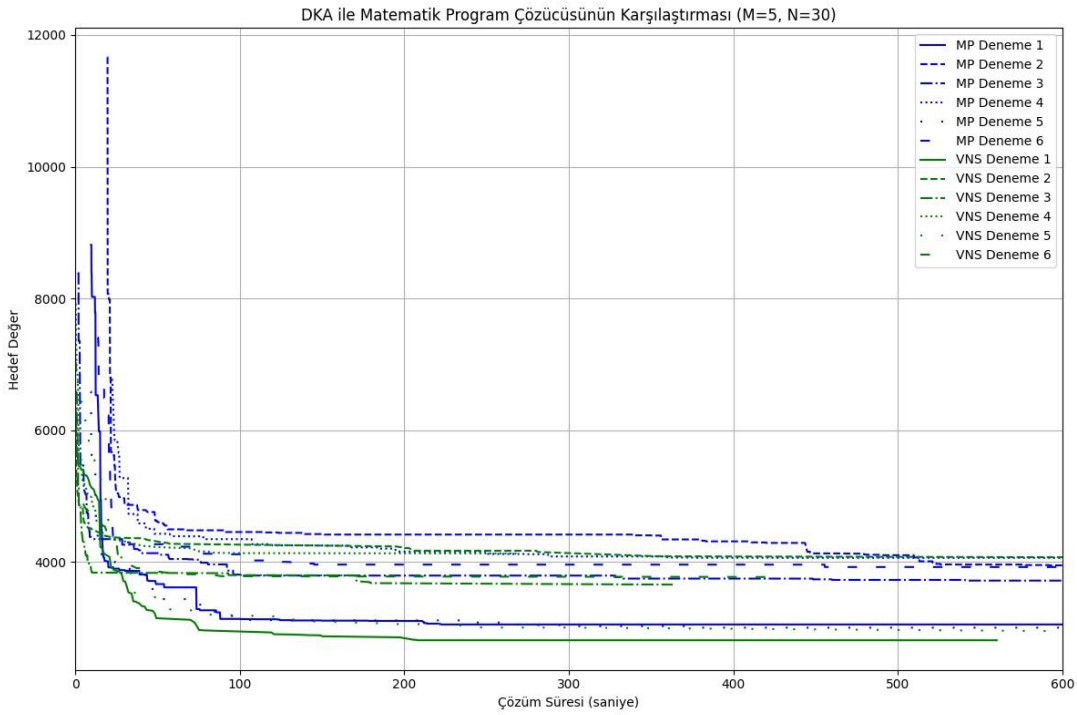
Şekil Ek A.4.1: 3 makine (M = 3), 20 iş (N = 20) ile DKA (yeşil) ve MP (mavi) için 6 hesaplamalı deneyde (n=6) hedef değerde çözüm süresine göre iyileşme.



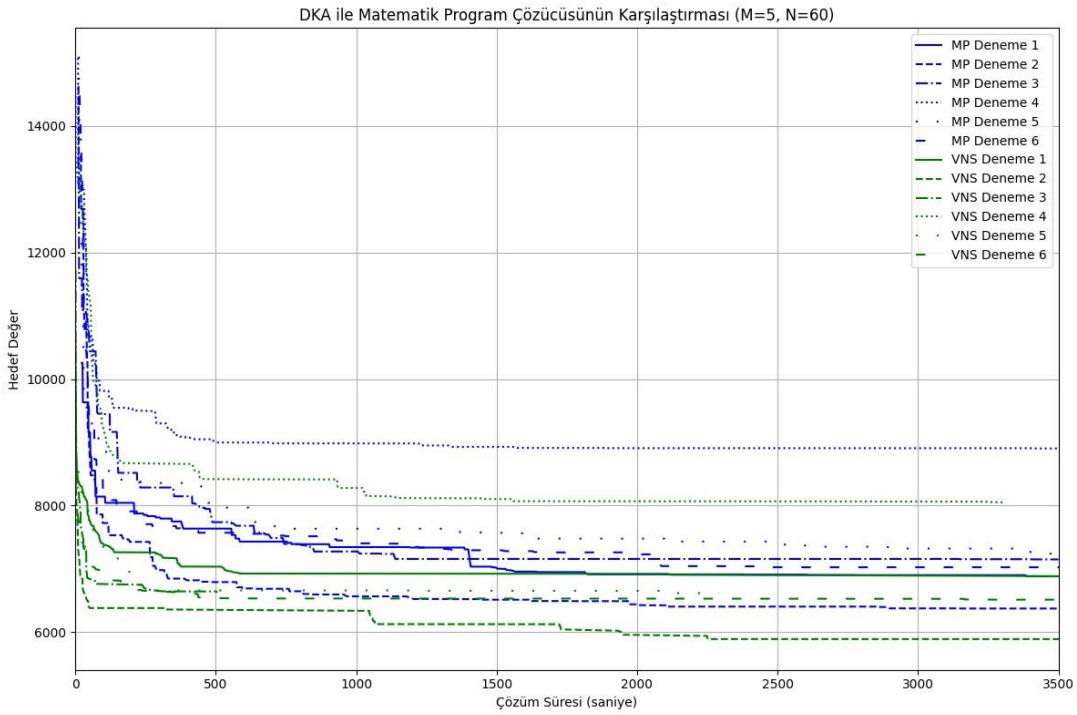
Şekil Ek A.4.2: 3 makine (M = 3), 30 iş (N = 30) ile DKA (yeşil) ve MP (mavi) için 6 hesaplamalı deneyde (n=6) hedef değerde çözüm süresine göre iyileşme.



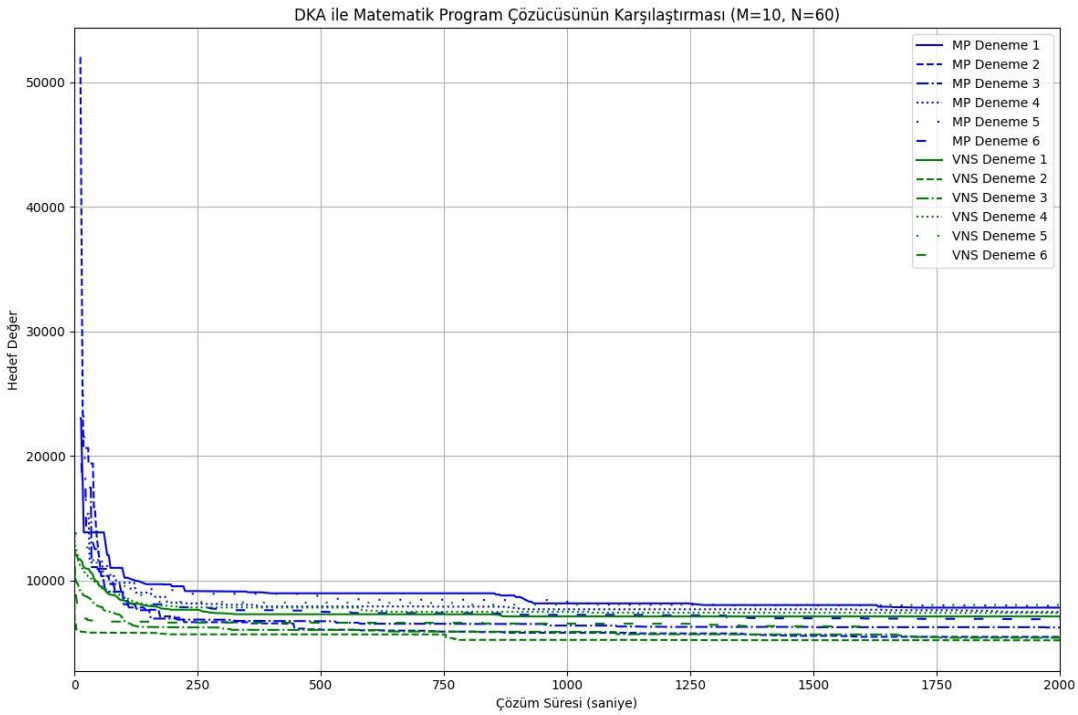
Şekil Ek A.4.3: 5 makine (M = 5), 20 iş (N = 20) ile DKA (yeşil) ve MP (mavi) için 6 hesaplamalı deneyde (n=6) hedef değerde çözüm süresine göre iyileşme.



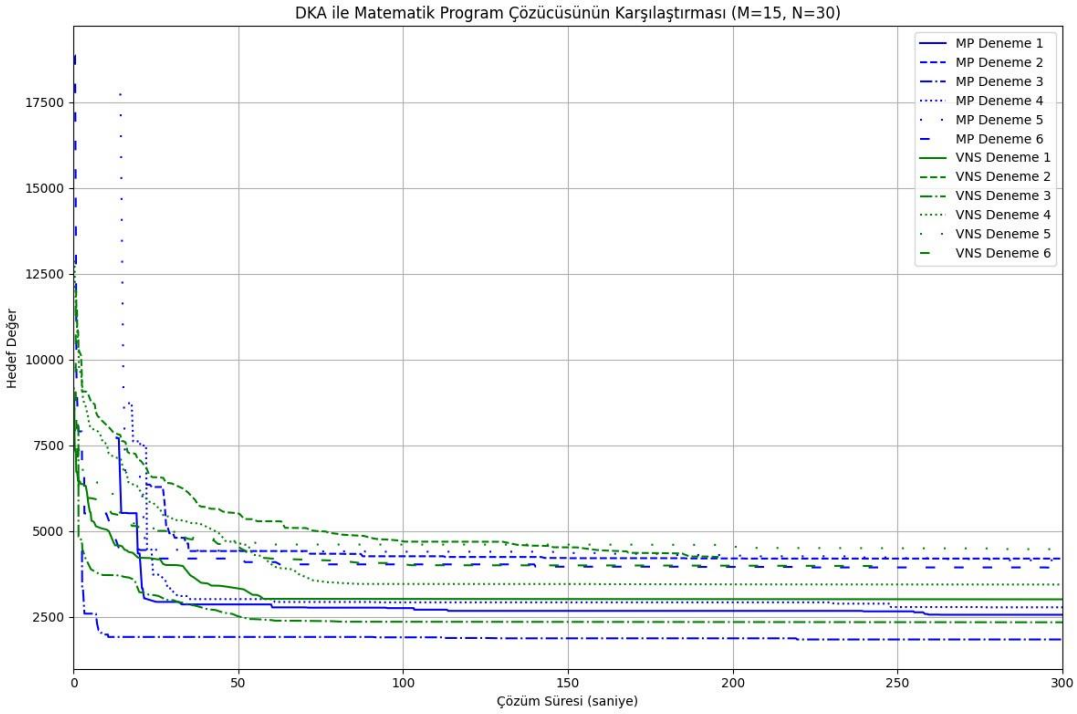
Şekil Ek A.4.4: 5 makine (M = 5), 30 iş (N = 30) ile DKA (yeşil) ve MP (mavi) için 6 hesaplamalı deneyde (n=6) hedef değerde çözüm süresine göre iyileşme.



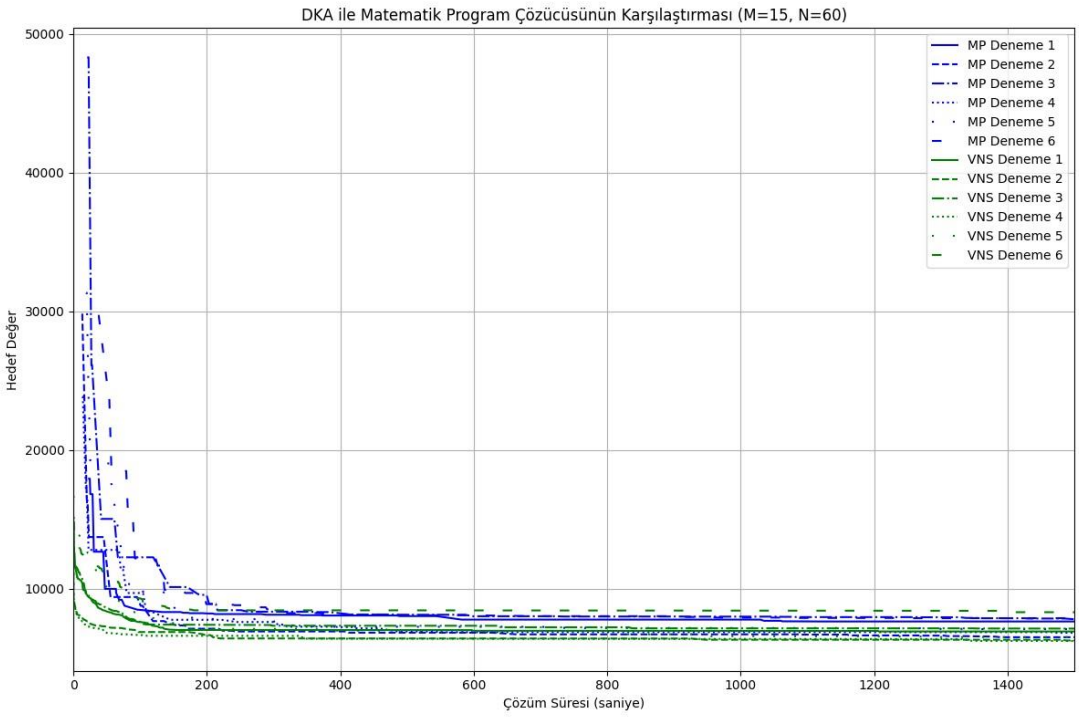
Şekil Ek A.4.5: 5 makine (M = 5), 60 iş (N = 60) ile DKA (yeşil) ve MP (mavi) için 6 hesaplamalı deneyde (n=6) hedef değerde çözüm süresine göre iyileşme.



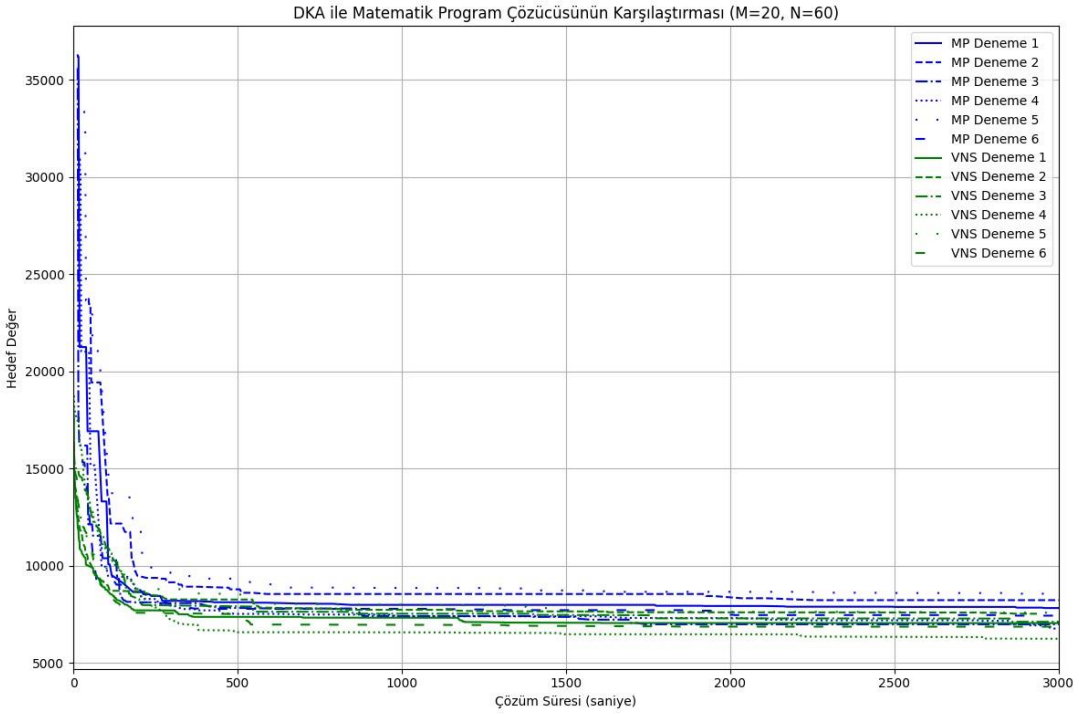
Şekil Ek A.4.6: 10 makine (M = 10), 60 iş (N = 60) ile DKA (yeşil) ve MP (mavi) için 6 hesaplamalı deneyde (n=6) hedef değerde çözüm süresine göre iyileşme.



Şekil Ek A.4.7: 15 makine (M = 15), 30 iş (N = 30) ile DKA (yeşil) ve MP (mavi) için 6 hesaplamalı deneyde (n=6) hedef değerde çözüm süresine göre iyileşme.



Şekil Ek A.4.8: 15 makine (M = 15), 60 iş (N = 60) ile DKA (yeşil) ve MP (mavi) için 6 hesaplamalı deneyde (n=6) hedef değerde çözüm süresine göre iyileşme.



Şekil Ek A.4.9: 20 makine (M = 20), 60 iş (N = 60) ile DKA (yeşil) ve MP (mavi) için 6 hesaplamalı deneyde (n=6) hedef değerde çözüm süresine göre iyileşme.

Tablo Ek A.4.1: DKA'nın ulaştığı en iyi hedef değerin %1 fazlasına DKA ve MP'nin ilk ulaşma süreleri

M	N	Örnek #	Hedef Değer (+1%)	DKA Süre	MP Süre	% Fark
3	5	0	836	1.0		
3	5	1	362	36.0	21.7	-40
3	5	2	830	0.7	22.7	3329
3	5	3	850	0.8	0.2	-79
3	5	4	293	0.8	0.2	-68
3	5	5	1058	19.5	0.1	-100
3	10	0	1510	5.3	1.5	-71
3	10	1	1343	169.6	0.9	-99
3	10	2	1423	2.5	27.8	1012
3	10	3	1088	163.1	9.3	-94
3	10	4	1292	48.7	5.0	-90
3	10	5	810	124.9	16.7	-87
3	20	0	2716	52.4	398.1	659
3	20	1	3326	9.8	27.3	179
3	20	2	2211	1141.6	43.4	-96
3	20	3	2765	18.1	167.3	823
3	20	4	1869	18.6	1912.8	10209
3	20	5	1901	555.9	943.1	70
3	30	0	3438	201.2		
3	30	1	2981	753.3	68.6	-91
3	30	2	3692	114.3	2813.2	2361
3	30	3	2262	35.6	120.9	240
3	30	4	2817	91.9	1604.4	1646
3	30	5	2615	17.2		
3	60	0	7755	2180.6		
3	60	1	6669	620.2		
3	60	2	6835	3554.4		
3	60	3	6725	2021.9		
3	60	4	5596	3083.5		
3	60	5	4988	2663.5		
5	10	0	1856	4.7	0.6	-88
5	10	1	1591	2503.5	26.6	-99
5	10	2	1535	0.9	0.5	-45
5	10	3	2170	41.8	0.5	-99
5	10	4	1522	398.1	0.6	-100
5	10	5	1919	2.4	2.2	-9
5	20	0	2481	84.6	17.7	-79
5	20	1	2597	285.3	1.1	-100
5	20	2	1889	32.2	78.2	143
5	20	3	2130	42.5	131.5	210
5	20	4	2622	17.4	285.1	1542
5	20	5	2392	22.9	346.8	1412
5	30	0	2839	204.3	2337.8	1044
5	30	1	4099	351.7	510.0	45

5	30	2	3692	179.2	1342.3	649
5	30	3	4157	81.6	250.9	208
5	30	4	2954	650.4		
5	30	5	3809	88.5	4087.3	4517
5	60	0	6916	2111.2	1825.2	-14
5	60	1	5923	2250.0		
5	60	2	6687	251.1		
5	60	3	8132	1140.2		
5	60	4	6682	239.7		
5	60	5	6576	438.3		
10	20	0	2153	1319.1	36.8	-97
10	20	1	2692	649.3	2.6	-100
10	20	2	3437	1.9	26.4	1257
10	20	3	1985	140.3	47.3	-66
10	20	4	3727	33.7	69.2	105
10	20	5	2398	80.8	13.2	-84
10	30	0	3751	208.6	112.5	-46
10	30	1	4339	3688.4	2062.5	-44
10	30	2	2312	586.7		
10	30	3	3433	735.7	3371.9	358
10	30	4	3098	3103.5	2890.6	-7
10	30	5	2782	290.1	827.6	185
10	60	0	7140	868.0		
10	60	1	5133	2140.2		
10	60	2	5399	1760.3		
10	60	3	7031	3330.7		
10	60	4	7997	2933.1	1333.1	-55
10	60	5	6354	1350.3		
15	20	0	2243	3254.0	6.6	-100
15	20	1	1990	2174.4	1.3	-100
15	20	2	2334	110.7	6.9	-94
15	20	3	2343	71.0	3.9	-95
15	20	4	3153	1617.6	18.7	-99
15	20	5	2441	42.8	18.5	-57
15	30	0	2750	4886.1	95.3	-98
15	30	1	4280	186.9	87.7	-53
15	30	2	2299	584.6	7.2	-99
15	30	3	3289	1397.2	29.6	-98
15	30	4	4418	389.6	57.4	-85
15	30	5	3981	239.0	140.2	-41
15	60	0	6651	3362.4		
15	60	1	6239	3250.3		
15	60	2	6852	3269.0	4822.4	48
15	60	3	6258	1342.6		
15	60	4	6532	1279.4		
15	60	5	7792	3628.6	2025.2	-44
20	30	0	2082	217.7		
20	30	1	3699	92.5	12.7	-86
20	30	2	3389	968.0	19.1	-98
20	30	3	3727	1413.1	39.6	-97

20	30	4	4047	60.7	44.8	-26
20	30	5	4685	1556.9	13.9	-99
20	60	0	7015	3378.2		
20	60	1	7606	2760.8		
20	60	2	7050	3700.2	1740.3	-53
20	60	3	6262	2773.0		
20	60	4	7659	1501.2		
20	60	5	6922	1387.3		

Tablo Ek A.4.2: 15 ve 20 makine (M = 15, 20), 60 iş (N = 60) için DKA'nın ulaştığı en iyi hedef değerin %10 fazlasına DKA ve MP'nin ilk ulaşma süreleri

M	N	Örnek #	Hedef Değer (+10%)	DKA Süre	MP Süre	% Fark
15	60	0	7244	127.2		
15	60	1	6795	184.5	664.5	260
15	60	2	7463	104.3	2691.0	2480
15	60	3	6815	49.8	1470.0	2849
15	60	4	7114	175.3	1280.5	630
15	60	5	8487	176.9	309.3	75
20	60	0	313	7640.3		
20	60	1	290	8283.3	2248.0	675
20	60	2	550	7677.7	881.0	60
20	60	3	379	6819.5	2967.8	683
20	60	4	548	8341.0	4192.5	665
20	60	5	487	7538.5	1998.3	310