



T.C.

SELÇUK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

BULANIK PARALEL ÇOK İŞLEMCİLİ MAKİNA PROBLEMLERİNİN ÇÖZÜMÜNDE HARMONİ ARAMA ALGORİTMASI

Mehmet Cabir AKKOYUNLU

YÜKSEK LİSANS TEZİ

Endüstri Mühendisliği Anabilim Dalı

Ocak-2013

KONYA

Her Hakkı Saklıdır

ÖZET

YÜKSEK LİSANS TEZİ

BULANIK PARALEL ÇOK İŞLEMCİLİ MAKİNA PROBLEMLERİNİN ÇÖZÜMÜNDE HARMONİ ARAMA ALGORİTMASI

Mehmet Cabir AKKOYUNLU

Selçuk Üniversitesi Fen Bilimleri Enstitüsü
Endüstri Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Orhan ENGİN

2013, 58 Sayfa

Jüri
Doç. Dr. Orhan ENGİN
Yrd. Doç. Dr. M. Emin BAYSAL
Yrd. Doç. Dr. Erdal CANIYILMAZ

İş sıralama ve çizelgeleme problemleri araştırmacıların üzerinde en çok çalıştığı konulardan birisidir. Araştırmacıların sıklıkla karşılaştığı bu problem türleri için çok çeşitli çözüm yöntemleri önerilmiştir. Ancak problemin NP-zor olması nedeniyle uzmanların optimum sonuçlara ulaşması güçtür. Bu nedenle katlanılabilir süreler dâhilinde başarılı çözümlere ulaşabilen meta sezgisel algoritmaların önerildiği çalışmalar yaygındır. Bu çalışmada Bulanık Paralel Çok İşlemcili Makine Problemlerinin Çözümü için Harmoni Arama Algoritması önerilmiştir.

Önerilen algoritma Oğuz'un (2005) geliştirdiği ve Gözen'in (2007) bulanıklaştırdığı test problemleri ile denenmiştir. Elde edilen sonuçlar Gözen'in (2007) ve Yılmaz'ın (2008) çalışmalarıyla kıyaslanmıştır. Kıyaslamalar sonucunda önerilen algoritmanın Bulanık Paralel Çok İşlemcili Problemlerin Çözümü için etkili ve rekabetçi bir yöntem olduğu belirlenmiştir.

Anahtar Kelimeler: Bulanık Paralel Makine Problemleri, Harmoni Arama Algoritması, İş çizelgeleme, Optimizasyon.

ABSTRACT

MS THESIS

SOLVING FUZZY PARALLEL MULTIPROCESSOR MACHINE PROBLEMS WITH HARMONY SEARCH ALGORITHM

Mehmet Cabir AKKOYUNLU

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF SELÇUK UNIVERSITY
THE DEGREE OF MASTER OF SCIENCE
IN INDUSTRIAL ENGINEERING

Advisor: Assoc. Prof. Dr. Orhan ENGİN

2013, 58 Pages

Jury
Assoc. Prof. Dr. Orhan ENGİN
Asst. Prof. Dr. M. Emin BAYSAL
Asst. Prof. Dr. Erdal CANIYILMAZ

Job sequencing and scheduling problems are very popular research field. Researchers have been tried extensively to solve these problems by using various ways. To get the optimal solution is very difficult due to NP-hard structure of the problems. Therefore, the meta-heuristic algorithms emplay to solve the problems in an affordable time more than optimal algorithms. In this study harmony search algorithm is used for solving Fuzzy Parallel Multiprocessor Machine Problems.

The proposed algorithm is tested with using problems developed by Oğuz (2005) and fuzzified by Gözen (2007). The results are compared with studies of Gözen (2007) and Yılmaz (2008). As a result of the comparisons it is indicated that the proposed algorithm is efficient and competent for solving Fuzzy Parallel Multiprocessor Machine Problems.

Keywords: Fuzzy Parallel Machine Problems, Harmony Search Algorithms, Job Scheduling, Optimisation.

ÖNSÖZ

Önceki çalışmalar ışığında doğayı keşfetme çabasını sürdüren bilim adamları, bulgularını çeşitli metotlar geliştirerek optimizasyon problemlerine de uygulamaya çalışmışlardır. Genetik Algoritma, Tabu Arama Algoritması, Tavlama Benzetimi, Karınca Kolonisi Algoritması bu çalışmalar içerisinde en çok bilinenleridir. Doğanın sırları optimizasyon problemlerinin modellenmesinde ilham kaynağı olmuştur. Aynı şekilde bir müzisyenin eserini oluşturması sırasındaki süreçten esinlenilerek geliştirilen Harmoni Arama Algoritması da bu tür optimizasyon çalışmaları arasına son yıllarda dahil olmuş bir metottur.

Bu çalışmada, Bulanık Paralel Çok İşlemcili Makine Problemlerinin Harmoni Arama Algoritmasıyla Çözümü yapılmıştır. Çalışmalarım sırasında benden desteğini esirgemeyen danışmanım Sayın Doç. Dr. Orhan ENGİN'e teşekkür ederim.

Mehmet Cabir AKKOYUNLU
KONYA-2013

İÇİNDEKİLER

ÖZET ..... **iv**

ABSTRACT ..... **v**

ÖNSÖZ ..... **vi**

İÇİNDEKİLER ..... **vii**

SİMGELER VE KISALTMALAR ..... **viii**

1. GİRİŞ **1**

2. KAYNAK ARAŞTIRMASI **3**

2.1. Harmoni Arama Algoritması için Kaynak Araştırması 3
2.2. Bulanık Çizelgeleme Problemleri için Kaynak Araştırması 8

3. MATERYAL VE YÖNTEM **12**

3.1. Materyal 12
3.1.1. Paralel Makine Problemleri 12
3.1.2. Esnek Akış Tipi Problemler 13
3.1.3. Çok İşlemcili Esnek Akış Tipi Problemler 14
3.1.4. Bulanık Mantık 15
3.1.4.1. Bulanık Sistemlerin Gelişimi 16
3.1.4.1. Bulanık Küme Teorisi 17
3.2. Yöntem 21
3.2.1. Harmoni Arama Algoritması 22
3.2.1.1. Problemin kurulması ve algoritma parametrelerinin belirlenmesi 24
3.2.1.2. Harmoni belleğinin oluşturulması 25
3.2.1.3. Yeni harmoni oluşturulması 25
3.2.1.4. Harmoni belleğinin güncellenmesi 26
3.2.1.5. Durma koşulunun kontrolü 26
3.2.2. Kesikli Harmoni Arama Algoritması 26
3.2.3. Bulanık paralel makine problemleri için parametre analizi 29
3.2.4.Paralel makine problemlerinin bulanıklaştırılması 31

4. ARAŞTIRMA SONUÇLARI **36**

5. SONUÇLAR VE ÖNERİLER **51**

5.1 Sonuçlar 51
5.2 Öneriler 51

KAYNAKLAR **52**

ÖZGEÇMİŞ **58**

SİMGELER VE KISALTMALAR

AF	: Amaç Fonksiyonu
Bg	: Bant genişliği
BNS	: Başlangıç Nota Sayısı
EBPD	: En Büyük Pozisyon Değeri
GA	: Genetik Arama
HAA	: Harmoni Arama Algoritması
HH	: Harmoni Hafızası
HHDAO	: Harmoni Hafızası Dikkate Alma Oranı
HHB	: Harmoni Hafızası Boyutu
İS	: İterasyon Sayısı
KHA	: Kesikli Harmoni Arama
NEH	: Nawaz-Enscore-Han
NP	: Polinomiyal olmayan
OUİ	: Ortalama Uyumlaştırma İndeksi
AAO	: Akort Ayar Oranı
Rst	: Rastgele
TB	: Tavlama Benzetimi
UI	: Uyumlaştırma İndeksi
YBA	: Yapay Bağışıklık Algoritması

1. GİRİŞ

Optimizasyon, hemen her alanda kullanılan bir kavram olup, kazancı maksimum ve maliyeti minimum yapmayı amaçlamaktadır. Optimizasyon için bir çok yöntem kullanılabilir. Eğer kullanılan yöntem, belirli bir probleme her uygulanışında aynı sonucu veriyorsa bunlara deterministik yöntemler denir. Deterministik yöntemler, genellikle en iyi bir tek çözümü bulmak için kullanılırlar. Deterministik yöntemler, bazı problemlerin çözümü için mümkün olmayabilmekte veya çok zaman ve maliyet gerektirebilmektedir. Bu tür problemlerde deterministik olmayan yöntemlerin kullanımı başarılı olabilir.

Deterministik olmayan yaklaşımlar, aynı başlangıç durumu için farklı sonuçlar verebilmekte, ancak en iyi olmasa da kabul edilebilir bir sonuca belirli bir süre sonunda ulaşabilmektedirler. Metasezgisel yöntemler, deterministik olmayan yöntemlerdir ve en iyi çözümü garanti etmemekle birlikte, denenmesi gereken olasılıkların çok fazla olduğu durumlarda daha az deneme ile kabul edilebilir seviyede iyi bir çözüm bulmaktadırlar.

Bir sistemin birden fazla mümkün çözümleri olabilir. Bunlardan bazıları diğerlerinden daha iyi olabilir. Bu nedenle, bu alternatif tasarımları karşılaştıracak bir kriter olmalıdır.

Pek çok optimizasyon algoritması bulunmaktadır. Ancak bu metotların çoğu ancak belirli bir tür problemin çözümü için geliştirilmiş yöntemlerdir. Bu nedenle, optimizasyon probleminin çözümünde gerekli metodun seçimi için, optimizasyon probleminin türünün belirlenmesi önemlidir.

Gerçek hayatta karşılaşılan problemlerin birçoğunda, problemin çözüm uzayı sonsuzdur veya tüm çözümlerin değerlendirilemeyeceği kadar geniştir. Sezgisel metotlar bu sorunun ortadan kaldırılması, kabul edilebilir bir zaman diliminde iyi bir çözümün bulunması amacı ile geliştirilmiş tekniklerdir. Böyle problemler için kabul edilebilir bir sürede çözümlerin değerlendirilmesiyle; aslında tüm çözüm uzayında bazı çözümlerin değerlendirilmesi aynı anlama gelmektedir. Bazı çözümlerin neye göre ve nasıl seçileceği ise sezgisel tekniğe göre değişiklik göstermektedir (Cura, 2008).

Sezgisel modeller çoğunlukla "NP-zor" olarak bilinen "Zor" problemlerin çözümü için geliştirilmiş optimizasyon teknikleridir. Bu modellerin kullanımı son yıllarda giderek artmakta ve mevcut tekniklere yenileri eklenmektedir. Bu modeller, çözüm uzayının tamamını aramadıkları için optimum çözümü garanti

edememektedirler, bununla birlikte kabul edilebilir iyilikte bir çözümü, kısa zamanda bulmaktadırlar.

Literatürde çokça çalışılan sezgisel tekniklerden bazıları; Yapay Sinir Ağları, Genetik Algoritma (GA), Karınca Kolonisi Optimizasyonu, Tavlama Benzetimi (TB), Tabu Arama, Parçacık Sürü Optimizasyonu olarak sıralanabilir. Bunlara nazaran daha yeni veya daha az çalışılan metasezgisellerden bazıları da; Arı Kolonisi Algoritması, Harmoni Arama, Kanguru Algoritması olarak sıralanabilir.

Bu araştırmada, son dönemlerde gelişim gösteren sezgisel bir model olan, Harmoni Arama Algoritması (HAA) modelinin bulanık paralel çok işlemcili makine problemlerine uygulanabilirliği araştırılmıştır. Yapılan çalışma Gözen (2007)'in Genetik Arama ve Tavlama Benzetimi, Yılmaz (2008)'ın Yapay Bağışıklık Algoritması (YBA) ile kıyaslanmıştır. Bu modelle ilgili olarak daha geniş bilgi "Materyal ve Metot" kısmında sunulmuştur.

2. KAYNAK ARAŞTIRMASI

Literatürde Harmoni arama algoritması ve bulanık paralel makine problemleri ile ilgili olarak birçok çalışma yapılmıştır. Çoğu çalışmada yeni geliştirilen metotlar anlatılmasına rağmen aynı zamanda literatür taraması şeklinde yapılan çalışmalar da mevcuttur. Kaynak araştırması "Harmoni arama algoritması için kaynak araştırması" ve "Bulanık paralel çok işlemcili makine problemleri için kaynak araştırması" olmak üzere iki kısımdan oluşmaktadır.

2.1. Harmoni Arama Algoritması için Kaynak Araştırması

Optimizasyon alanındaki birçok yayına rağmen, harmoni arama ve kesikli harmoni arama metotlarıyla ilgili olarak yapılan çalışma sınırlı sayıdadır. Harmoni arama ile ilgili yapılan literatür çalışmaları kronolojik sırayla aşağıda sunulmaktadır.

Geem ve ark. (2001) müzik geliştirme üzerine temellenen bir harmoni arama mantığını optimizasyon problemleri üzerine uygulamışlardır. Temel olarak Harmoni Arama Algoritması (HAA), harmoni vektörlerinden rastgele bir biçimde başlangıç popülâsyonu oluşturulması ve bu hafıza vektörünün geliştirilmesi mantığıyla çalıştırılmıştır.

Lee ve Geem (2004) Harmoni arama algoritmasına dayanan bir optimizasyon yöntemi kullanmışlardır. Harmoni arama algoritmasında, müziğin birimleri arasındaki uyumun, arama sürecine uygulanması kavramsallaştırılmıştır. Bu yöntemde ilk başta bir başlangıç değerine gerek duyulmamıştır, ardışık arama yerine, rastgele arama kullanılmış ve türetilen bilgilerin gereksiz olduğu söylenmiştir.

Mahdavi ve ark. (2007), optimizasyon problemlerinin çözümü için geliştirilmiş bir harmoni arama yöntemi önermişlerdir. Bu metot, harmoni arama algoritmasının yaklaşma oranı ve düzgünlüğünün artırılmasıyla türetilmiştir. Çalışmada, harmoni arama algoritması için seçilen parametreler tartışılmış ve bu parametreler için stratejiler sunulmuştur. Önerilen metodun, standart optimizasyon problemlerindeki başarısı ve çeşitli kıyaslamaları çalışmada sunulmuştur.

Ayvaz (2007), yer altı Piezometrik gözlemlerin verilen bir kümesine dayanarak, parametrelerin bölge yapısının ve akifer parametrelerinin tanımlandığı bir ters çözüm

algoritması önermiştir. Bölge yapısı tanımlama probleminde bulanık ortalama kümeleme yöntemi kullanmıştır. Optimizasyon tekniği olarak harmoni arama yöntemi kullanılarak problemi çözmüştür.

Omran ve Mahdavi (2008) küresel en iyi harmoni arama adlı araştırmada, parçacık mantığından esinlenerek harmoni aramayı geliştirmeye çalışmışlardır. Önerdikleri metodu 10 farklı örnek çözümü ile diğer harmoni arama örnekleriyle kıyaslamışlar ve genellikle daha başarılı sonuçlar elde etmişlerdir. Küresel en iyi harmoni arama parametrelerini analiz etmişler ve tam sayılı programlama deneme problemlerinde test etmişlerdir. Önerdikleri metodun harmoni arama problemleri için etkin bir alternatif olduğunu sunmuşlardır.

Fesanghary ve ark. (2008) mühendislik optimizasyon problemlerini çözmede, sürekli dizayn değişkenlerinin kullanıldığı bir hibrid harmoni arama algoritması kullanmışlardır. Harmoni arama algoritmalarının başarılı sonuçlar bulmalarına rağmen yerel arama konusunda hala yetersiz olduğunu vurgulayan araştırmacılar, bu konuda Sıralı Dörtlü Programlama kullanarak yerel arama adımını hızlandırmışlardır. Yapılan örnek çalışmada ise önerdikleri yöntemin sonuçlarını sunmuşlardır.

Geem (2008), öncelikle harmoni arama çalışmalarında genellikle sürekli değişkenlerin kullanıldığını, ancak günlük yaşamda verilerin daha çok kesikli olduğunu vurgulamış ve gerçekçi tasarım durumlarını hesaba katmak için "kesikli tasarım değişkenleri için yeni türetilmiş harmoni arama algoritması" geliştirmiştir.

Ceylan ve ark. (2008) harmoni arama kullanarak ulaştırma enerjisi talebi hesaplayan yeni bir metot önermişlerdir. Nüfus, yurt içi üretim ve araç kilometrelerini girdi olarak almışlardır. Oluşturulan modelin sonuçlarını, Enerji ve Tabi Kaynaklar Bakanlığı'nın verileriyle karşılaştırmışlardır. Sonuçlar, önerilen Harmoni Arama metodunun, enerji modellemede kullanılabileceğini göstermiştir.

Saka (2009) BS 5950 için performans sınırları ve davranışlarını kullanarak optimum dizayn algoritması geliştirmiştir. Kiriş ve kolonlar için hesaba katılan güç sınırları, yanal burulma burkulma olarak dikkate alınmıştır. Uygun malzemenin tasarımının optimum ağırlıkta seçimine karar vermede harmoni arama algoritması kullanılmıştır.

Coelho ve Bernert (2009) farklı ilk şartlardaki, iki özdeş kesikli kaotik sistemin senkronizasyonu için, geliştirilmiş harmoni arama algoritmasına dayanan, Orantılı Bütünleşik Türetilmiş Kontrolcüler konusunu araştırmışlardır.

Ayvaz (2009) çalışmasında simülasyon modeli olarak üç boyutlu farklı yer altı kaynakları akış modeli geliştirmiştir. Sonra bu modeli harmoni arama algoritması ile bütünleştirmiştir. Sonuçları çeşitli yer altı kaynakları yönetimi problemleri ile test etmiştir.

Alataş (2010) "Kaotik harmoni arama algoritmaları" çalışmasında karakteristiklere yaklaşmayı geliştirmek için parametrelerin adaptasyonunda, kaotik haritalandırmayı kullanmıştır. Bu yöntemin sürecin yerel arama noktalarında sıkışıp kalmasını önlediğini savunmuştur. Bunun için kaotik numara üreticisi kullanmıştır. Yapılan kıyaslamalı çalışmalar sonucunda bazı metotların çözüm kalitesini, bazılarının ise küresel arama kabiliyetini artırdığını gözlemlemiştir.

Wang ve ark (2010) akış tipi çizelgelemede, toplam akış zamanını minimize etmede, hibrid arama algoritması, hibrid küresel en iyi arama algoritması ve hibrid değiştirilmiş küresel en iyi arama algoritması olmak üzere üç hibrid harmoni arama algoritması geliştirmişlerdir. Yaptıkları uygulamalar sonucunda hibrid değiştirilmiş küresel en iyi arama algoritmasının, iş çizelgelemede toplam akış zamanı kriterine bağlı problemlerin çözümünde etkinliğini sunmuşlardır.

Geem ve Sim (2010) "Parametre ayarı gerektirmeyen harmoni arama algoritması" adlı çalışmalarında Genetik Algoritma, Tabu Araması, Karınca Koloni Optimizasyonu ve Parçacıklı Sürü Optimizasyonu gibi yöntemlerde gerekli olan parametre ihtiyacına değinmişlerdir. Çalışmada daha başarılı sonuçlar için bu parametrelerin seçimine özen göstermek gerektiğinden bahsetmişlerdir. Deneyim gerektiren bu parametreli sistem yerine, harmoni hafızasında her değişken için operasyon tipi içeren ek bir matris önerisi sunmuşlardır.

Jaberipour ve Khorram (2010) sürekli tasarım değişkenleri için iki farklı yeni harmoni arama yöntemi önermişlerdir. İlk algoritma yeni bir aralık genişliği tanımlayan harmoni aramadır. İkincisi, ilk algoritmanın yaklaşma oranını ve başarı düzgünlüğünü geliştirmiş bir harmoni arama algoritmasıdır. Bazı matematiksel fonksiyon minimizasyon problemleri ve bazı mühendislik optimizasyon problemlerinin çözümü bu algoritmalarının etkinliğini ve yeterliliğini göstermiştir.

Zou ve ark. (2010a) parçacıklı sürü optimizasyon yönteminin sürü zekâsından ilham alarak, güvenilirlik problemlerinin çözümü için yeni bir harmoni arama algoritması önermişlerdir. Önerilen metot pozisyon güncelleme ve küçük olasılıklarla genetik mutasyon olmak üzere iki önemli aşamadan oluşturulmuştur. İteratif olarak harmoni hafızası bir sonraki küresel en iyi sonuçla optimum değeri bulmaktadır.

Zou ve ark. (2010b) iş atama problemlerinde ara işlemci iletişiminin toplamını ve bazı kısıtlı kaynaklar için işlem maliyetlerini minimize etmek için iş atama problemlerinde yeni bir küresel harmoni arama algoritması kullanmışlardır.

Pan ve ark. (2010a) sürekli optimizasyon problemlerinin çözümü için kendinden adaptif küresel en iyi harmoni arama algoritması önermişlerdir. Önerilen algoritmada elde var olan verilerden yeni harmoniler üretmek için yeni bir geliştirme şeması sunmuşlardır. Harmoni hafızası durumu oranı ve aşama ayar oranı dinamik olarak öğrenen mekanizmalar yardımıyla değiştirilmiştir. Mesafe genişliği ilk aşamalarda iyi sonuç arama ile düzenlenmiş ve son aşamalarda arama süreçlerinden yararlanılmıştır.

Han ve ark. (2010) akış tipi çizelgeleme problemlerinde tamamlanma zamanı minimizasyonu amacı için Kesikli Harmoni Arama (KHA) algoritması kullanmışlardır. Önerilen hibrid KHA algoritması, kesikli iş permutasyonlarını, harmonileri göstermekte kullanmıştır ve yeni harmoniler üretmek için iş permutasyonuna dayalı geliştirme tablosuna başvurmuştur. Hesaplamalar ve simülasyonlar sonucu elde edilen sonuçlar önerilen algoritmanın sınırlı iş akış çizelgeleme problemlerinin çözümünde tamamlanma zamanı kriteri için etkili ve verimli olduğunu göstermiştir.

Pan ve ark. (2010b) yeni bir KHA algoritması kullanarak, sıralı bağımlı kurulum zamanları için yüklemelerin bölündüğü akış tipi çizelgelemelerde tamamlanma zamanı minimizasyonu yapmışlardır. Geleneksel Harmoni Arama algoritmasından farklı olarak iş permutasyonlarını harmonileri göstermekte ve iş permutasyonlarına dayalı geliştirilmiş harmoniler üretmekte kullanmışlardır. Algoritmanın arama kabiliyetini yükseltmek için yeni performans yükseltmeli sezgisel (NEH metodu) üzerine temellenen etkili bir başlangıç şeması oluşturmuşlardır.

Gao ve ark. (2011) beklemesiz akış tipi çizelgeleme problemlerinde, toplam akış zamanını minimize eden kesikli harmoni arama algoritması geliştirmişlerdir. Kesikli iş permutasyonlarıyla kurulan bir harmoni kurulmuş, NEH metodu ile harmoni geliştirilmiştir. Yeni aşama ayarı kuralı uygulanılarak yeni harmoni üretimi geliştirilmiştir. Algoritmanın yerel faydalanma yeteneğini geliştirmek için yerel arama kullanılmıştır. Yapılan hesaplamalar ve kıyaslamalar metodun başarısını vurgulamıştır.

Wang ve ark. (2011) sınırlı permutasyon akış tipi çizelgeleme problemlerinde, tamamlanma zamanı kriteri için, hibrid modifiye küresel en iyi harmoni arama algoritması kullanmışlardır. Öncelikle En Büyük Pozisyon Değeri (EBPD) kuralı sürekli harmoni vektörlerini, iş permutasyonlarına dönüştürmekte kullanılmıştır. Etkili başlama tablosu NEH metodu uygulanarak oluşturulmuştur. Yeni keşifler uygulayarak

harmoni hafızasındaki harmoni vektörleri geliştirilmiştir. Ayrıca yeni bir aşama ayarı kuralı, küresel en iyi harmoni vektöründen temel alınarak geliştirilmiştir.

Pan ve ark. (2011) eş boyutlardaki kesikli akış durumunda, akış tipi çizelgeleme problemlerinde dinamik alt harmoni hafızası algoritması ve yerel en iyi arama algoritmasını kullanarak, toplam ağırlıklandırılmış erken bitme ve gecikme hatalarını minimize etmeyi önermişlerdir. Öncelikle problemi, Harmoni Arama algoritmasına uygun hale getirmek için değer sıralaması kuralı uygulayarak sürekli harmoni vektörlerini kesikli iş sıralarına dönüştürmüşlerdir. Daha sonra NEH algoritması kullanarak etkili başlangıç şemasını oluşturmuşlardır. Sonraki aşamada alt harmoni hafızalarını birçok alt gruba bölerek çeşitliliği artırmışlardır. Alt harmoni hafızaları vektörlerinden yeni bir gelişim şeması oluşturmuşlardır. Harmoni vektörlerindeki değerlerin üretilmesinde kaotik sıra kullanarak çeşitliliği artırmışlardır. Yerel arama kabiliyetini geliştirmek için de yerel arama algoritmasını kullanmışlardır. Hesaplamalı deneyler ve kıyaslamalar Hibrid Genetik Algoritma ve Hibrid Kesikli Parçacıklı Sürü Algoritmasından daha iyi sonuçlar vermiştir.

Zou ve ark. (2011a) sırt çantası (0-1) problemleri için yeni bir küresel harmoni arama algoritması önermişlerdir. Önerilen algoritma konum güncelleme ve küçük olasılıklarla genetik mutasyon olmak üzere iki önemli aşama içermektedir. Her bir iterasyonda kötü harmoniden iyi bir harmoniye hareket, küresel en iyi harmoni algoritması ile sağlanmıştır.

Zou ve ark. (2011b) bir gaz türbininin aşırı hız koruma sistemi için, Küresel Harmoni Arama Algoritması uygulamışlardır. Karmaşık (köprü) Sistem Optimizasyon Problemleri ve Güvenilirlik–Fazlalık Optimizasyon Problemleri olmak üzere iki çeşit problem tipi olduğunu açıklamışlardır. Genel olarak iki problemin de Karışık tamsayılı lineer olmayan programlama kapsamında olduğunu belirtmiştirler. Çalışmalarında etkili bir küresel harmoni arama algoritmasını Parçacıklı sürü optimizasyonu algoritması ile birleştirerek optimizasyon problemlerini çözmüşlerdir. Yapılan hesaplamalar ve kıyaslamalar ile önerdikleri metodun başarısını sunmuşlardır.

Geem (2012) harmoni hafızası boyutundan daha fazla başlangıç çözümünün üretip daha sonra harmoni hafızası boyutunca sınırlandırılan bir ilk hafıza boyutu yaklaşımı önermiştir.

2.2. Bulanık Çizelgeleme Problemleri için Kaynak Araştırması

Han ve ark. (1994) bulanık teslim zamanı ile tek makine çizelgeleme problemlerini araştırmıştır.

Ishıbuchi ve ark. (1994) bulanık işlem zamanı ile iki akış tipi çizelgeleme problemi geliştirmiştir. Birinci problem tipinde verilen işler için minimum memnuniyet derecesinin maksimizasyonunu, ikinci problem tipinde ise toplam memnuniyet derecesinin maksimizasyonunu amaç olarak belirtmişlerdir.

Kuroda ve Wang (1996) bulanık atölye tipi çizelgeleme problemleri üzerine bir analiz çalışması yapmışlardır. Çalışmaları, klasik gönderme kuralını direkt olarak uygulayan bulanıklaştırma, bulanık sıralama ile ilgili metotları kullanma ve sezgisel metotlarla optimal çizelgelemeye karar verme şeklinde üç yaklaşım olarak raporlamıştır. Literatürde sınırlı sayıda Bulanık Paralel Makine Çizelgeleme Problemi yer aldığını ifade etmişlerdir.

Wang (1999) kesin olmayan bilgiler içeren ürün projelerini çizelgeleyebilmek için bir bulanık küme yaklaşımı geliştirmiştir. Projeler bulanık hazırlık zamanları, bulanık gecikme ve bulanık sürelere sahip aktivitelerden oluşturmuştur. Bulanıklık ikizkenar yamuk bulanık sayılardan oluşmuştur. Araştırmada amaç, bulanık hazırlık zamanının, gecikmenin, üstünlük ve kaynak kısıtlarının tatmin edildiği her aktivite için başlangıç zamanlarına karar vermek olarak belirtilmiştir.

Hapke ve Slowinski (2000) bulanık olarak kaynak kısıtlı proje çizelgeleme modelini çalışmışlardır. Bulanık parametrelerle seri ve paralel çizelgeleme problemlerine öncelik kuralları geliştirmişlerdir. Çalışmada bulanık kaynak kısıtlı çizelgeleme problemlerinin, çok amaçlı çözümünde tavlama benzetimi önermişlerdir.

Chanas ve Kasperski (2001) bulanık işlem zamanı ve bulanık teslim zamanı ile tek makine çizelgeleme problemi geliştirmişlerdir. Çalışmada ilk sorunun bulanık gecikmenin maksimum değerini minimize etmek, ikinci sorunun ise maksimum bulanık gecikmenin beklenen değerini minimize etmek olduğunu ifade etmişlerdir. Problemler için çözümü Lawler'in polinomial algoritmasını kullanarak geliştirmişlerdir.

Litoiu ve Tadei (2001) Gerçek Zamanlı Sistemlerdeki uygulama için bulanık çizelgelemeyi araştırmışlardır. Çalışmada bulanık işlem zamanı ve bulanık teslim süresi ile genel periyodik çizelgeleme modellerini sunmuşlardır. Çizelgelemenin üç karardan oluştuğunu ve bunların:

"Adım1) Çizelgeleme için iş seç,

Adım 2) Seçilen işin her bir operasyonu için makine seç,

Adım 3) Adım 2'de belirlenen makinenin kuyruktaki işi için uygun siparişi bul." olduğunu belirtmişlerdir. Bu kararları gerçekleştirilirken bulanık kuralları içeren hiyerarşik bir çizelgeleme kullanmışlardır. Modelin çözümünde memnuniyet olarak adlandırılan bir maliyet fonksiyonu tanımlamışlardır. Örnek olarak monotonik çizelgeleme teorisinin bir uzantısını sunmuşlardır.

Yuen ve ark. (2002) bulanık iş süreci zamanlaması ile zaman serisi çizelgelemeyi incelemişlerdir. İlk olarak bulanık süreç zamanlamasının üçgensel bulanık bir sayı ile gösterilebildiğini varsaymışlar ve sonrasında sıralı işlerin tamamlamanın olasılığının yeni bir matematiksel ifadeye sahip olması için, iş üyeliği fonksiyonlarından iş bitirme olasılık profili oluşturmada kısıtlı olasılık programlama yöntemi kullanmışlardır. Çalışmada bazı özel durumlar altındaki çizelgeleme modeli için optimal çözüm geliştirilerek, farklı teslim süreleri için optimizasyon yapmışlardır.

Yao ve Lin (2002) akış tipi çizelgeleme problemlerinde bulanık kümeleri istatistik ile birleştirerek yeni bir yaklaşım geliştirmişlerdir. Çalışmada kesin olarak bilinmeyen değerler yerine bazı örnek verilerin kullanılabileceğini belirtmişlerdir. İstatistik ve bulanık kümelerin kombinasyonu bu problemin çözümünde başarılı bir yöntem olmuştur.

Wang (2004) genetik algoritma kullanarak maksimum çizelgeleme performansına ulaşmayı amaçlayan bir yöntem geliştirmiştir.

Temiz ve Erol (2004) bulanık işlem zamanları ve bulanık teslim süreleri kullanılarak çok makineli akış tipi çizelgeleme problemleri için bulanık tamamlanma zamanı ve bulanık maksimum gecikmeyi birlikte eniyilemeyi amaçlayan bir genetik algoritma önermişlerdir.

Akyol (2004), *n* iş ve *m* makineden oluşan akış tipi çizelgeleme problemlerinde tamamlanma zamanı minimizasyonu için altı farklı sezgisel algoritmayı, yapay sinir ağları kullanarak çözmüştür.

Li ve ark. (2004), esnek üretim sistemlerinde çizelgeleme için küçük veri setlerinin bilgi eğilimlerinin kestirimini mega bulanıklaştırma yolu ile yapan bir model geliştirmişlerdir.

Peng ve Liu (2004), paralel makine çizelgeleme problemlerinin bulanık işlem zamanları ile modellenmesi için bir metot önermişlerdir. Çalışmada bulanık çizelgeleme modellerinin üç yeni türü sunulmuş ve problemlerin çözümünde melez zeki algoritma kullanmışlardır.

Engin ve Döyen (2004) esnek akış tipi çizelgeleme problemlerinin Yapay Bağışıklık Algoritması (YBA) ile çözümü için bir algoritma önermişlerdir. Algoritmaları klonal seçim algoritmasını ve yakınlık geliştirme tekniklerini içermektedir. Ayrıca yapay bağışıklık algoritmasında kullanılan parametrelerin optimizasyonun önemli olduğuna değinmişler ve optimizasyon çalışmaları yapmışladır. Sonuç olarak önermiş oldukları yöntemleri literatürde yer alan problemlerde denemişler ve yararlı sonuçlar elde etmişlerdir.

Brar ve ark. (2005), çok amaçlı optimizasyon problemlerinde birden fazla kriteri aynı zamanda tatmin etmeye yönelik bir model önermişlerdir. Çalışmada çok amaçlı termal güç istasyonunda çizelgeleme probleminde operasyon maliyeti, kirlilik, emisyon vb. indeksler minimize edilmeye çalışılmıştır.

Majozi ve Zhu (2005), bütünlenmiş planlama ve çizelgeleme için bulanık küme teorisinin bir uygulamasını sunmuşlardır. Bulanık küme teorisinin sanayide karşılaşılan belirsiz ve şüpheli durumların modellenmesinde kolaylık sağladığını ifade etmişlerdir.

Herroelen ve Leus (2005), çalışmada deterministik ağ değerlendirme yapıları ile projelerin belirsizlik altında çizelgelenmesi için yaklaşımların potansiyelleri tartışılmıştır. Araştırmada reaktif çizelgeleme alanındaki geçmiş çalışmalar incelenmiştir. Belirsizlik altındaki çizelgeleme teknikleri sınıflandırılmıştır.

Anglani ve ark. (2005), hazırlık maliyeti bağımlı paralel makine çizelgeleme problemlerinin çözümü için güçlü bir yaklaşım sunmuşlardır. Literatürde bu konuda geliştirilmiş model ve çözüm yöntemleri mevcuttur. Ancak çalışmaların çoğu deterministik yollar kullanmaktadır. Bu çalışmada, talep memnuniyeti ve toplam hazırlık zamanındaki optimal çözümü sağlamak için işlem süresindeki belirsizliği hesaplayarak bulanık matematiksel model oluşturulmuştur.

Kasperski (2005), bulanık parametrelerle sıralama problemlerine olasılık bakış açısı önermiştir. Tam olarak bilinmeyen her bir parametre için olasılık dağılımları vermiştir. İşlerin sırasını hesaplayarak, işlerin gecikme olasılığını hesaplamayı amaçlamıştır.

Kim ve ark. (2005), kaynak kısıtlı çoklu proje çizelgeleme problemlerini çözmek için bulanık mantık kontrolörlü bir genetik algoritma önermişlerdir. Çalışmada amaçlar, toplam proje zamanı minimizasyonu ve toplam gecikme zamanı minimizasyonu olarak tanımlanmıştır.

Logendran ve ark. (2006), çalışmalarında sıra bağımlı kurulum zamanlı esnek akış tipi problemlere grup çizelgeleme yaklaşımını önermiştir.

Allahverdi ve ark. (2008), çizelgeleme problemlerini kümeli, kümeli olmayan ve sıra bağımlı, sıra bağımsız kurumlu zamanlı problemler şeklinde sınıflandıran bir çalışma yapmışlardır.

Shiau ve ark. (2008), çalışmalarında toplam ağırlıklı tamamlanma zamanının minimize edildiği orantılı esnek akış tipi çizelgeleme için yapısal genetik algoritmaya dayanan yeni bir hibrit etkin yaklaşım önermişlerdir.

Jungwattanakit ve ark. (2009) çalışmalarında en az bir aşaması ilişkisiz paralel makinelerden oluşan esnek akış tipi çizelgeleme problemlerini dikkate almışlardır. Problemin amacını, durgun esnek akış tipi çevre şartlarında, geciken iş sayısını ve tamamlanma zamanının konveks toplamını minimize eden bir çizelge bulmak olarak ifade etmişlerdir.

Ruiz ve Rodriguez (2010), çalışmalarında hibrit akış tipi çizelgeleme problemlerinin çözümü için önerilen sezgisel ve metasezgisel yöntemlerin incelendiği ayrıntılı bir literatür tarama çalışması yapmışlardır.

Khademi ve Fakhrzad (2011), esnek akış tipi problemlerin çözümüne bulanık yaklaşım ile etkili bir algoritma önermişlerdir. Amaçlarını toplam iş gecikmesini minimize etmek olarak belirlemişlerdir. Farklı işlem zamanlı paralel makineleri dikkate almışlardır.

Engin ve ark. (2011) çalışmalarında çok işlemcili görevler için hibrit akış tipi çizelgeleme problemlerinin çözümüne etkili bir genetik algoritma önermişlerdir. Çok sayıda iş, makine ve görevin olduğu problemlerde tamamlanma zamanı minimizasyonu yapmışlardır.

Kianfar ve ark. (2012), çalışmalarında deterministik olmayan ve dinamik olarak gelen sıra bağımlı kurulum zamanlı esnek akış tipi çizelgeleme problemlerini dikkate almışlardır. Problemin amacını minimum ortalama toplam gecikme zamanını veren çizelgeyi elde etmek olarak belirlemişlerdir.

Choi ve Wang (2012), çalışmalarında en kısa işlem zamanı ve genetik algoritmayı kombine ederek stokastik işlem zamanlarına sahip esnek akış tipi çizelgeleme için tamamlanma zamanı minimizasyonu yapmışlardır.

3. MATERYAL VE YÖNTEM

3.1. Materyal

Paralel makine problemleri NP-zor kapsamına girdiği için henüz optimum çözümleri kısa sürede verecek algoritmalar geliştirilememiştir. Bu tür problemlerde optimum çözüme ulaşabilmek için bir çok sezgisel yöntem geliştirilmiştir. Sezgisel yöntemlerde, çözüm kümesinin tüm elemanlarını dikkate almadan, katlanılabilir süreler içinde başarılı veya tatmin edici sonuçlara ulaşmak hedeflenir.

Bu çalışmada çok işlemcili bulanık paralel makine problemlerinin çözümü için harmoni arama algoritması önerilmiştir. Geliştirilen yöntem Oğuz (2005)'un oluşturduğu ve Gözen (2007)' in bulanıklaştırdığı test problemleri için sınanmıştır.

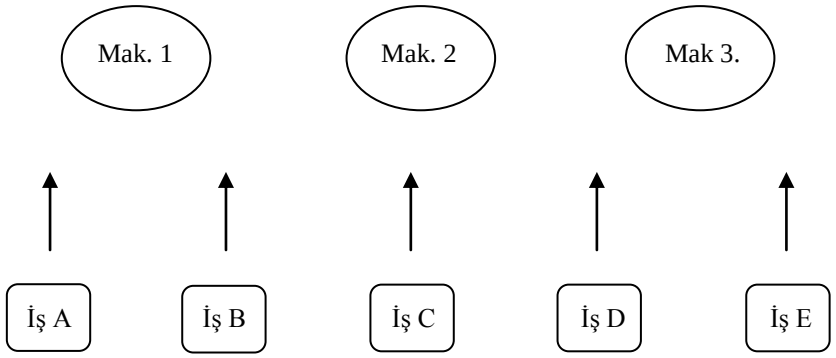
3.1.1. Paralel Makine Problemleri

Verilen amaçları optimum yapacak şekilde, *n* tane işin belirli bir zaman diliminde işlenmek üzere *m* adet makineye veya iş istasyonuna atanması problemlerine iş çizelgeleme problemleri denir. İşler operasyon denilen alt işlemlerden oluşmaktadır. Operasyonların işlem zamanları deterministik veya stokastik olabilir. İş çizelgeleme probleminde yer alan işlerin operasyon sıraları aynı ise "akış tipi çizelgeleme", farklı ise "atölye tipi çizelgeleme" olarak adlandırılmaktadır.

Bir çizelgeleme probleminde çizelgelenecek tüm işler çizelgeleme işleminin başında biliniyor ise "statik çizelgeleme" olarak adlandırılır. Eğer işler çizelgeleme esnasında değişiyorsa buna "dinamik çizelgeleme" denmektedir. Ayrıca iş çizelgelemede bütün parametrelerin kesin olarak bilindiği durumlar "deterministik çizelgeleme" olarak adlandırılırken en az bir parametrenin probabilistik olduğu durumlar ise "stokastik çizelgeleme" olarak adlandırılır.

Çizelgeleme problemlerinin temel amaçları verilen teslim tarihlerine uymak, makineleri maksimum verimlilikte kullanmak, hazırlık zamanlarını minimum tutmak vb. şekilde olabilir. Ayrıca bu amaçların bazılarının birlikte kullanıldığı çok amaçlı problemler de mevcuttur. Verilen bir problemde tüm amaçların hepsinin birden optimum yapılması imkansızdır. Çizelgeleme problemlerinde sistemin genel amacına göre tatmin edici iyi bir sonuç bulmak için çalışılmaktadır.

Paralel makine problemleri *m* tane özdeş (eş) makinenin paralel olarak yerleştirildiği sistemlerdir. Her iş yalnız bir operasyona sahiptir ve bu *m* makinenin herhangi birinde işlenebilmektedir. Genellikle bu tarz problemlerde makine sayısı iş sayısından az olmaktadır. Örnek bir paralel makine sistemi Şekil 3.1 'de sunulmuştur.

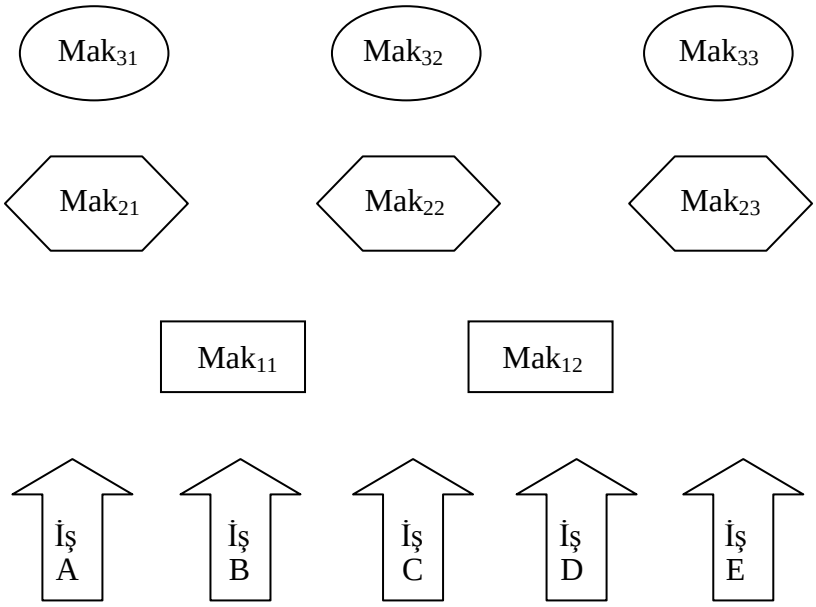


Şekil 3.1. Örnek bir paralel sistem modeli (Döyen (2004)).

3.1.2. Esnek Akış Tipi Problemler

Esnek akış tipi çizelgeleme problemi; akış tipi çizelgeleme problemleri ile paralel makine problemlerinin özelliklerini birleştirmektedir. Esnek akış tipi problemlerde, makineler *s* tane ardışık kademeye yerleştirilmişlerdir. $l = 1,2,...,s$ olmak üzere bir *l* kademesinde, m_1 tane eş makine bulunmaktadır. $j = 1,2,...,n$ olmak üzere bir *j* işi, her bir kademedeki makinelerden herhangi birinde yada daha fazlasında işlem görmelidir. Farklı kademelerde *j* işinin işlem süreleri $P_{1j}, P_{2j},...,P_{sj}$ ile gösterilir. İşlerin önceliği yoktur. Ancak, bir makinede bir operasyon başladıktan sonra başka bir operasyonun aynı makinede işlem görmeye başlayabilmesi için öncekinin mutlaka bitirilmesi gerekir.

Her makinede belli bir anda en fazla bir iş işlem görebilir. Her kademede işlem görmeyi bekleyen işler için ayrılan stok alanı kısıtsız kabul edilir. Amaç, genellikle en son işin sistemden ayrılma zamanını (C_{maks}), minimize etmektir. Esnek akış tipi problemler NP-zor'dur (Gupta, 1988). Şekil 3.2'de bir esnek akış tipi sistemin yapısı verilmiştir. Sistemde 5 iş, her birindeki makine sayıları sırasıyla 2, 3, 3 olan 3 kademede çizelgelenmektedir.



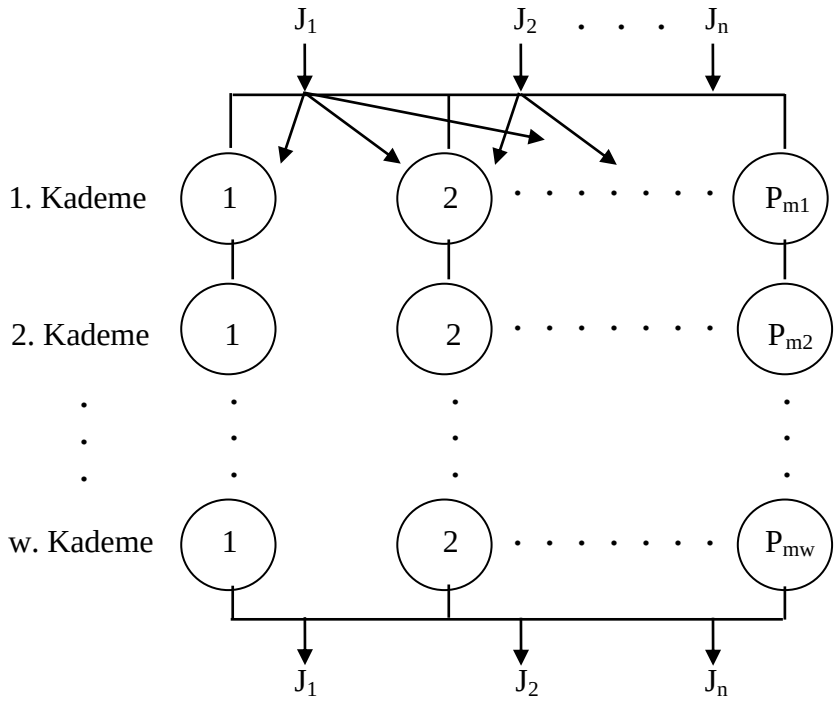
Şekil 3.2. Örnek bir esnek akış tipi sistem modeli (Döyen (2004)).

3.1.3. Çok İşlemcili Esnek Akış Tipi Problemler

Çok işlemcili esnek akış tipi çizelgeleme ile esnek akış tipi çizelgeleme arasındaki fark, herhangi bir işin birden çok işlemcide aynı anda işlenebilmesinin gerekli olmasıdır. Burada işlemci olarak ifade edilen nesne, insan veya makine olabilmektedir. Bazı üretim sistemlerinde birtakım operasyonların iki ya da daha fazla insan tarafından aynı anda yapılması gerekebilir.

Çok işlemcili esnek akış tipi çizelgeleme sisteminde, $j = \{1, 2 \ldots, n\}$ olmak üzere n tane iş, $i = \{1, 2, \ldots, w\}$ w tane aşamada sıralı bir şekilde işlem görmelidir. Her aşamada m_i adet eş paralel makine bulunur. Her iş tüm aşamalarda sırasıyla işlem görür. Her aşamada yalnız tek tip operasyon yapılmaktadır. Her bir operasyon sadece kendinden önceki operasyon tamamlandıktan sonra başlayabilir. *i.* aşamadaki *j.* işin operasyonunun gerçekleşebilmesi için ihtiyaç duyduğu makine sayısı $size_{ij}$, operasyon süresi P_{ij} olsun. Diğer bir deyişle *j.* işin *i.* operasyonu, m_i makinelerinde $size_{ij}$ kadar makinede aynı anda yapılmalıdır (Oğuz ve Ercan 2005, Yılmaz 2008).

Çok işlemcili esnek akış tipi çizelgeleme sisteminde tüm işlerin ve makinelerin çizelgeme sürecinde uygun olduğu varsayılır. Çok işlemcili esnek akış tipi çizelgeleme için bir örnek sistemin yapısı Şekil 3.3'de (Öztürk 2007) gösterilmiştir.



Şekil 3.3. Çok işlemcili bir sistemin yapısı (Öztürk 2007).

3.1.4. Bulanık Mantık

Bulanık mantık söylem olarak ilk kez 1965 yılında California Berkeley Üniversitesinden, Prof. Lotfi A. Zadeh'in bu konu üzerindeki ilk çalışmalarını literatürde yayınlamasıyla duyulmuştur. O tarihten sonra önemi gittikçe artan ve üzerinde birçok bilim insanının çalıştığı bulanık mantık, belirsizliklerin anlatımı ve belirsizliklerle çalışılabilmesi için kurulmuş katı bir matematik düzen olarak tanımlanabilir. Bulanık mantık, belirsizliklerin ifade edilmesi ve sistemlerin belirsizliklerle çalışılabilmesi için geliştirilmiş bir matematiksel formüller dizini olarak açıklanabilir (Yılmaz 2008).

Genel olarak bulanık mantık doğru/yanlış, evet/hayır, negatif/pozitif gibi geleneksel değerler arasında tanımlanabilen ara değerleri gösteren çoklu değerlemeli bir mantıktır. Ara değerler matematiksel formüller kullanılarak bulanıklaştırılabilir (Zadeh 1984).

Bulanık mantık teorisini anlamak için öncelikle bulanık küme teorisi anlaşılmalıdır. Örneğin "çok güçlü" kavramını inceleyecek olursak, bu kavram kişiden

kişiye değişiklik gösterebilir. Kesin sınırlar söz konusu olmadığı için bu kavramın matematiksel olarak ifade edilmesi de zordur. Ama genel olarak bazı kriterlerin kullanılması ile sayısal olarak güçlülük sınırları belirlenebilir. Böylece, kendi aitlik fonksiyonu ile temsil edilebilir. Aitlik fonksiyonu 0 ile 1 arasında her değeri alabilir. Böyle bir aitlik fonksiyonu ile "kesinlikle ait" veya "kesinlikle ait değil" arasında istenilen hassasiyette değerler elde edilebilmektedir (Sağıroğlu ve ark. 2003).

Bulanık mantık bilimsel olarak verilerin derecelendirilmesini sağladığı için "Çoklu değerlilik" olarak da adlandırılır. Bunun tam tersi olan kesinlik ise "İkili değerlilik" olarak bilinmektedir (Kosko 1993). Yani ikili değerlilik olarak güçlü veya güçsüz olarak yapılan derecelendirme, bulanık mantık kapsamında çok güçlü, çok güçsüz, az güçlü vb. şekillerde derecelendirilebilmektedir (Kaynak ve Armağan, 1992).

Bulanık mantık kullanılarak yapılan derecelendirme sistemi verilerin keskin sınırlar içerisinde ifade edilmesinin gerekliliğini azaltır. Böylece niteliksel verilere dayanan gerçek dünya problemlerinin matematiksel olarak tanımlanabilmesini sağlar. Bulanık mantığın genel özellikleri Zadeh tarafından aşağıdaki şekilde ifade edilmiştir (Gözen 2007);

- Kesin değerlere dayanan düşünme yerine, yaklaşık düşünme kullanılır,
- Bulanık mantıkta değerler [0,1] aralığında belirli bir derece ile gösterilir,
- Bulanık mantıkta bilgi güçlü, güçsüz, çok güçlü gibi niteliksel ifadeler şeklindedir,
- Bulanık çıkarım işlemi niteliksel ifadeler arasında tanımlanan kurallar ile yapılır,
- Her mantıksal sistemin bulanık olarak ifade edilebilirliği söz konusudur,
- Bulanık mantık, matematiksel modelin çok zor elde edildiği sistemlere kolay uygulanması açısından çok uygundur.

Bulanık mantık tam olarak bilinmeyen veya eksik girilen bilgilere göre işlem yapma fırsatı sunmaktadır.

3.1.4.1. Bulanık Sistemlerin Gelişimi

1965 yılında Zadeh tarafından geliştirilen bulanık mantığın bilinen ilk uygulaması, Mamdani tarafından 1974 yılında yapılan buhar makinesinin bulanık denetiminin gerçekleştirilmesi ile olmuştur. Daha sonraki yıllarda da bulanık denetleyici sistemler yaygın olarak kullanılmıştır (Elmas 2003).

3.1.4.1. Bulanık Küme Teorisi

Bulanık küme teorisinde, klasik kümelerdeki elemandır veya değildir ifadesi yerine ‘şu derecede elemanıdır’ ya da ‘şu derecede elemanı değildir’ ifadeleri yer almaktadır. Bir eleman için, eleman olma veya olmama durumu 0 veya 1 ile değil, 0 ile 1 arasındaki üyelik derecesi ile gösterilmektedir. Örneğin “Uzun boylu kime denir?” sorusunu her iki mantığa göre tanımlayalım. Klasik küme mantığına göre 160 cm. boyundaki bir kişi uzun boylu insanlar kümesi içinde değildir. Hatta 169 cm. boyundaki bir kişi uzun boylu insanlar kümesi içinde değildir. Oysa bulanık mantığa göre 160 cm. boyundaki kişiye kısa denilmez. Çünkü kısmen de olsa uzun boylu insanlar kümesi içindedir. Bulanık mantıkta 160 cm boyundaki biri 0,6 üyelik derecesiyle, 170 cm. boyundaki biri 0,7 üyelik derecesiyle, 180 cm. boyundaki biri de 1,0 üyelik derecesiyle uzun boylu olabilir.

Bulanık küme teorisi de klasik küme teorisinde var olduğu gibi kümeler üzerinde tanımlanmış birleşim, kesişim ve tümleme işlemlerini içinde barındırır. Tanımlanan bazı fonksiyonların kullanılmasıyla bu işlemler gerçekleştirilir. Bir fonksiyonun bulanık birleşim, bulanık kesişim veya bir bulanık tümleme fonksiyonu olabilmesi için bir takım aksiyomları yerine getirmesi gerekmektedir. Her işlem tanımlanmış bir aksiyometrik yapı ile ifade edilir ve her bir aksiyometrik yapı, mutlaka sağlanması gereken aksiyomlardan oluşur. Bu temel üç işlem için aksiyometrik yapıyı oluşturan aksiyomların yanı sıra tanımlı bir grup aksiyom vardır. Tanımlanmış bu aksiyom grubunun sağlanma zorunluluğu yoktur. Her üç işlem içinde tanımlı tüm aksiyomları sağlayan fonksiyonlar da varlığı söz konusudur. Bunlara örnek olarak bulanık küme teorisinin standart işlemleri verilebilir. Bunlar (Etaner 1993);

Bulanık Tümleme:

$$\mu_A - (x) = 1 - \mu_A(x) \tag{3.1}$$

Bulanık Birleşim:

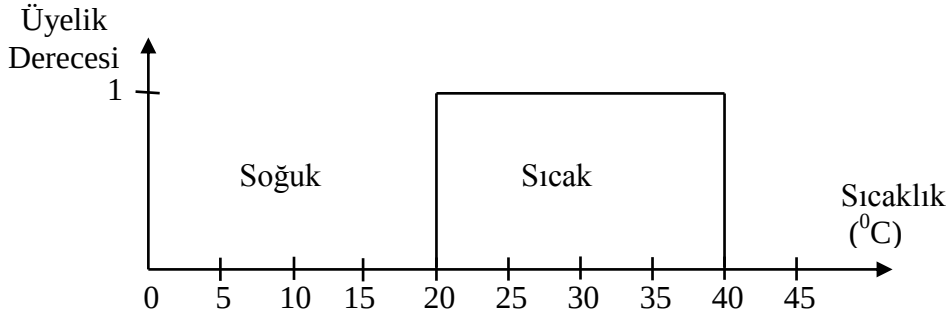
$$\mu_{AUB}(x) = Max\ (\mu_A(x), \mu_B(x)) \tag{3.2}$$

Bulanık Kesişim:

$$\mu_{A\cap B}(x) = Min\ (\mu_A(x), \mu_B(x)) \tag{3.3}$$

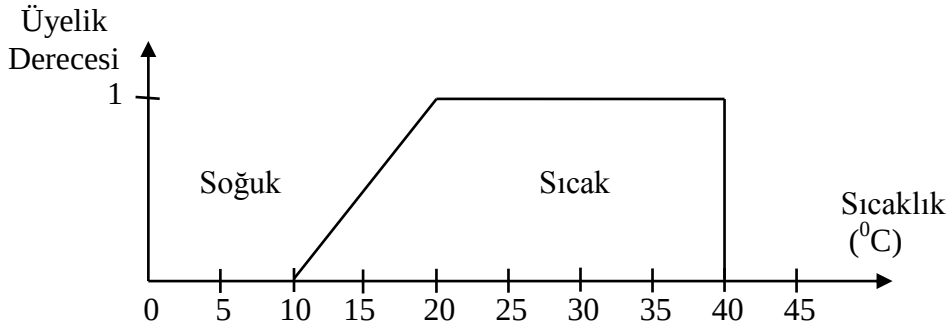
Şekil 3.4‘te klasik kümelere yönelik bir örnekleme verilmiştir. Bu örnekte sıcaklık 20^o C’ nin altına düşerse sıcak değildir. Yani hiçbir esnekliği içinde barındırmayan klasik mantık kuramına göre $19,5^o$ C sıcak değildir. Gerçek dünyada ise

sınırlar hiçbir zaman bu kadar keskin değildir. Şöyle ki, denetleyicilerin fiziksel kümeleri birbirlerinden çok net değerlerle ayrılacağı için,denetim çıktısı da her değişim için ani tepkiler gösterecektir (Özkan 1997).

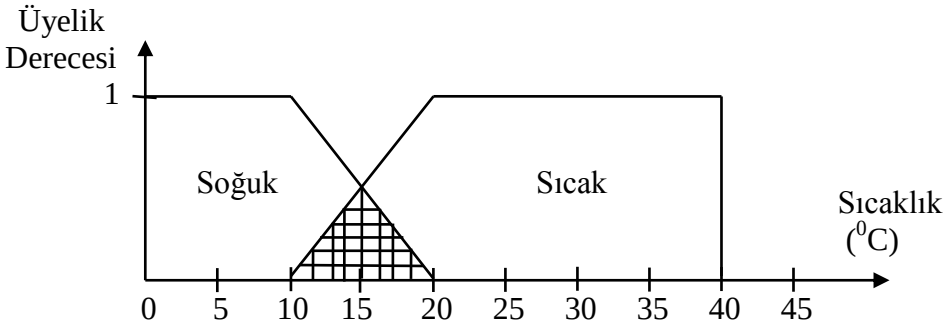


Şekil 3.4. Sıcaklık örneği için keskin küme (Yılmaz 2008).

Şekil 3.5'te ise bulanık kümeler için bir örnek verilmiştir. Burada 10-40^{o} C arasındaki değerler sıcak kümesine ait üyeleri göstermektedir. 20-40^{o} C arasında yer alan tüm değerlerin üyelik dereceleri 1'dir. 10-20^{o} C derece arasında yer alan sıcaklıkların üyelik dereceleri ise 0 ile 1 değerleri arasında değişecektir. Başka bir ifadeyle 11 ^{o} C az sıcak olarak ifade edilirken 17 ^{o} C biraz sıcak olarak değerlendirilecektir. 20 ^{o} C oda sıcaklığı kabulü ile soğuk bulanık kümesi Şekil 3.6' da görüldüğü gibi elde edilir (Etaner 1993).

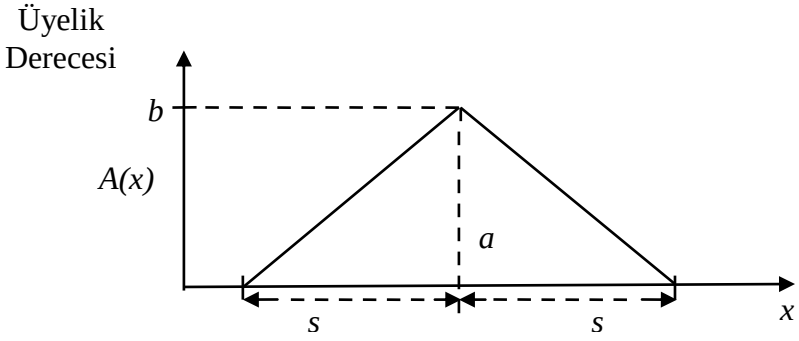


Şekil 3.5. Sıcaklık için bulanık küme örneği (Yılmaz 2008).



Şekil 3.6. Bulanık kümelerde örtüşüm (Yılmaz 2008).

Bulanık mantık sistemi temel olarak, üyelik işlevlerinden ortaya çıkarılan niteliksel değişkenlerin oluşturduğu girişlerin, karar verme sürecinde kullanmasını hedeflemektedir. Değişkenler "eğer-o halde" kuralların ön şartları sayesinde birbirleriyle eşleşirler. Her bir kuralın sonucu, girişlerin üyelik derecelerinden, durulaştırma metoduyla sayısal bir değer elde edinilmesi söz konusudur. Bulanık mantık sistemin kural listesi ve üyelik işlevi tasarımı için genellikle uzman işletmen kaynaklı sağlanmış bilgiler kullanılmalıdır. Üyelik işlevleri Şekil 3.7'de genel, simetrik ve üçgen üyelik fonksiyonu olarak görülmektedir (Klir ve ark. 1997).

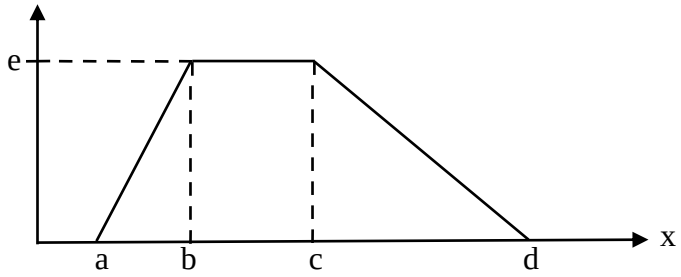


Şekil 3.7. Genel, simetrik ve üçgen üyelik fonksiyonu (Gözen 2007).

Simetrik ve üçgen şekilli üyelik fonksiyonları Şekil 3.7'de gösterilen *a*,*b* ve *s* parametreleri ile gösterilir. Genel gösterimi 3.4 teki ifade gibidir (Etaner 1993).

$$A(x) = \begin{cases} b(1 - \frac{|x-a|}{s}, & a - s \leq x \leq a + s \\ 0, & Aksi\ takdirde \end{cases} \qquad (3.4)$$

Üyelik fonksiyonu için bir diğer önemli sınıf da Şekil 3.8'de gösterilen yamuk üyelik fonksiyonudur.



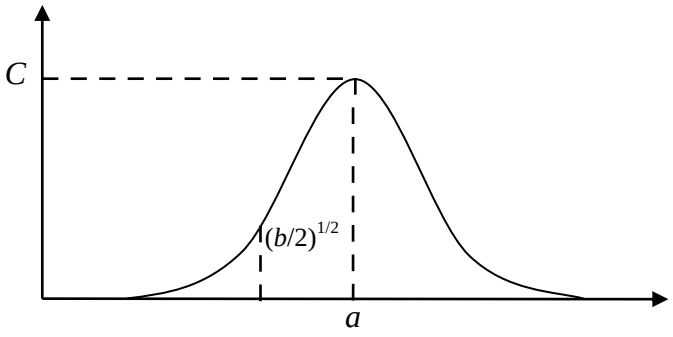
Şekil 3.8. Yamuk üyelik fonksiyonu (Gözen 2007).

Bu fonksiyon *a, b, c, d* ve *e* olmak üzere 5 parametreden oluşmakta ve genel olarak 3.5'teki gibi ifade edilmektedir (Etaner 1993).

$$A(x) = \begin{cases} \frac{(a-x)e}{a-b} & a \le x \le b \\ e & b \le x \le c \\ \frac{(d-x)e}{d-c} & c \le x \le d \\ 0 & Aksi\ takdirde \end{cases} \tag{3.5}$$

Parametreleri *a, b* ve *c* olarak ifade edilmiş bir çan eğrisi üyelik fonksiyonun (Şekil 3.9) formülü aşağıda verilmiştir (Ahmad 2004).

$$A(x) = ce^{\frac{(x-a)^2}{b}} \tag{3.6}$$



Şekil 3.9. Çan eğrisi üyelik fonksiyonu (Gözen 2007).

Bulanık kümelerde üyelik işlevleri Sayısal ve İşlevsel olmak üzere iki türe ayrılmaktadır. Bulanık kümenin üyelik işlevinin ve üyelik derecesinin sayılarından oluşturulan vektörle yapılan tanımlamaya sayısal tanımlama denilmektedir. Bu vektörün boyutu ayrıklaştırma seviyesine yani uzaydaki süreksiz elemanların sayısına

bağlıdır. İşlevsel tanımlamada ise bulanık kümenin üyelik işlevi, tanım uzayındaki her eleman için üyelik derecesi hesaplanabilen analitik deyimlerle tanımlanmaktadır. Üyelik işlevlerinde genel olarak [0,1] aralığında gerçek sayılar kullanılmaktadır (Elmas 2003).

Bulanık bir küme, üyelerinin net bir biçimde belirgin olmadığı ancak aday öğelerinin bu kümeye üyelik derecelerinin bilindiği bir kümedir. Bulanık bir kümenin gösterimi matematiksel olarak çeşitli şekillerde olabilir. Bu gösterimlerden bazıları aşağıda verilmiştir:

$$\forall x \in E: \mu_a(x) \in [0,1] \tag{3.7}$$

$$A = \{x, \mu_a(x) | x \in E\} \tag{3.8}$$

Bir bulanık küme, kümenin sahip olduğu elemanlar ve bu elemanların üyelik derecelerinin bilinmesi ile oluşturulabilir. A bir bulanık küme olmak üzere Eş. (3.9)' daki gibi tanımlanabilir;

$$A = \{u/, \mu_A(u) | u \in U\} \tag{3.9}$$

Burada *u*, *A* kümesinin bir elemanı, $\mu_A\ (u)$ üyelik işlevi, *U* ise *A* kümesinin tanımlandığı evreni ifade etmektedir. Üyelik işlevi bir kümenin elemanlarının o kümeye hangi üyelik derecesi ile ait olduğunu gösteren ve [0,1] arasında değer alabilen bir işlev olduğuna göre Eş. (3.10)' daki gibi tanımlanabilir;

$$\mu_A(u): U \rightarrow [0,1]: \mu_A(u): \in [0,1] \tag{3.10}$$

Burada *A* bir bulanık kümeyi, U ise *A* kümesinin üzerinde tanımlandığı evreni, $\mu_A\ (u)$ ise üyelik işlevini ifade etmektedir (Gözen 2007).

3.2. Yöntem

Bu çalışmada, bulanık paralel makina çizelgeleme problemlerinin harmoni arama algoritması ile çözümü C# programlama dili kullanılarak oluşturulan bilgisayar programı yardımıyla yapılmıştır. Harmoni arama algoritmasında optimal veya optimale yakın çözümler elde etmek için parametre optimizasyonu yapılmıştır.

3.2.1. Harmoni Arama Algoritması

Sezgisel ve metasezgisel yöntemler, optimizasyon problemlerinde sıklıkla kullanılmaktadır. Bu problemlerde uygulanan metodun performansı, minimum toplam tamamlanma zamanına, katlanılabilir hesaplama sürelerinde erişebilmesiyle değerlendirilir. Yani yalnızca toplam tamamlanma zamanının kısalığı değil, aynı zamanda hesaplanma süresinin kısalığı da istenmektedir.

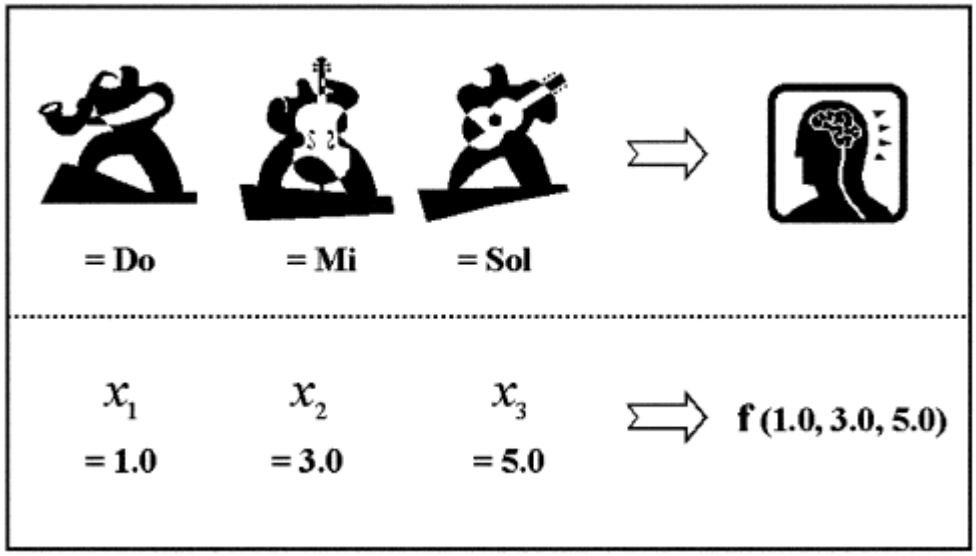
Optimizasyon tekniği olarak kullanılan ve ilk olarak Geem ve arkadaşları (2001) tarafından geliştirilen harmoni arama algoritması, bir orkestradaki müzisyenlerin çaldıkları notalar ile harmoni açısından en iyi melodinin elde edilmesi prensibine dayanmaktadır (Akkoyunlu ve Engin 2011).

Harmoni arama algoritmasının çeşitli mühendislik problemlerinde kullanımı ile ilgili olarak literatür taraması kısmında örnekler verilmiştir. Karar değişkenleri için özel bir başlangıç çözümü gerektirmemesi ve optimizasyon çözümüne birden fazla çözümle birçok yönde genel optimum çözüm arayarak ilerlemesi harmoni arama algoritmasının sağladığı avantajlardandır. Algoritma bu özelliği sayesinde yerel optimum çözümlerden kurtulmaktadır. Harmoni arama algoritmasının bir diğer avantajı ise optimizasyon işlemlerinde hem sürekli hem de kesikli değişkenler ile çalışma imkanı sunmasıdır (Mahdavi ve ark. 2007).

Harmoni arama algoritmasının hesaplama mantığı açısından genetik algoritma ile benzerlikleri olmasına rağmen, bu iki yöntem arasındaki en belirgin fark yeniden üretim aşamasındaki varsayımlardan kaynaklanmaktadır. Genetik algoritmada yeni bir karar değişkeni oluşturulurken toplum içerisinden alınan iki adet birey kullanılırken (çaprazlama), harmoni arama algoritmasında yeni birey toplum içerisindeki tüm bireylerin özelliklerini taşıyabilmektedir. Ayrıca harmoni arama algoritmasında bir iterasyonun tamamlanması genetik algoritmaya nispeten daha hızlıdır (Özyön ve ark. 2011).

Müzik geliştirme esnasında her bir sanatçı enstrümanının akordunu yaparak mümkün ses aralıkları dâhilinde icra yapmakta ve birlikte bir harmoni vektörü oluşturmaktadırlar. Eğer oluşturulan akortlar güzel bir harmoni sağlamışsa her sanatçının hafızasında bu harmoni depolanmaktadır (Şekil 3.10). Bunu mühendislik problemine uygularsak; başlangıçta her bir değişkenin başlangıç değeri mümkün aralıklar dahilinde seçilir, birlikte bir çözüm vektörü oluştururlar ve eğer iyi bir sonuç

elde edilmişse değişkenin hafızasında depo edilir. En kötü harmoni hafızadan çıkarılır. Ayrıca her bir adımda bulunan sonuç geliştirilmeye çalışılır (Lee ve Geem, 2004).



Şekil 3.10. Müzik geliştirme ve mühendislik optimizasyonu arasındaki bağlantı (Lee ve Geem, 2005).

Ayrıca bilindiği gibi müzik icrası sırasında müzisyenler bir takım notaları değiştirirler, eğer yapılan değişiklik daha güzel bir harmoni sağlıyorsa harmoni hafızasına alınır. Bu mantık optimizasyon problemine uyarlanırsa; atanan her bir değişken için değişiklikler söz konusudur, yapılan değişiklik başarı sağlıyorsa çözüm hafızasına alınır ve harmonideki en kötü çözüm de harmoni hafızasından çıkarılır. Böylece harmoni hafızası eklenen daha başarılı çözümler ile geliştirilir.

Ayrıca bir müzisyen yeni bir harmoni oluştururken aşağıda verilen şu üç kuralı uygular (Lee ve Geem, 2005):

1. Hafızasındaki bir akordu çalar.
2. Hafızasındaki akordun bir komşu akordunu kullanır.
3. Mümkün olan ses aralıkları dahilinde rastgele bir seçim yapar.

Bu sıralamayı aynı şekilde optimizasyon problemine uygulamak da mümkündür. Şöyle ki;

1. Hafıza karşılıkları da denilen Harmoni Hafızasından (HH) herhangi bir değer seçilebilir.
2. Mümkün olan aralıktaki bir değerin komşu değeri seçilebilir. (Bu adım Ayar düzeltmesi (Akort ayarı) olarak da adlandırılmaktadır.)
3. Mümkün olan rastgele bir seçim yapılabilir.

Temel olarak Harmoni Arama Algoritması (HAA), harmoni vektörlerinden rastgele bir biçimde başlangıç popülâsyonu oluşturur ve harmoni hafızasında (HH) depolar. Yeni harmoni; hafıza çözümleri, ayar düzeltmesi ve rastgele seçim gibi yöntemlerle harmoni hafızasındaki çözümlerden üretilir. Ardından harmoni hafızasından aday vektör ve harmonideki en kötü vektör ile güncelleme operatörü kullanılarak kıyaslanır. Aday vektör kıyaslamalarla güncellenir. Bu optimizasyon süreci belirlenmiş bitirme kriteri sağlanana kadar tekrar ettirilir. Basitliği ve uygulama kolaylığı nedeniyle Harmoni Arama Algoritması bir çok optimizasyon problemi için dikkat çekicidir (Geem ve ark., 2001).

Genel olarak harmoni arama algoritması 5 adımdan oluşur. Bu adımlar aşağıda verilmiştir (Özyön ve ark. 2011);

1. Problemin kurulması ve çözüm parametrelerinin belirlenmesi,
2. Harmoni belleğinin oluşturulması,
3. Yeni harmoni oluşturulması,
4. Harmoni belleğinin güncellenmesi,
5. Durma koşulunun kontrolü.

3.2.1.1. Problemin kurulması ve algoritma parametrelerinin belirlenmesi

Problemin bu adımında optimizasyonu yapılacak amaç fonksiyonu (AF) tanımlanmaktadır.

$$AF = Minimum\{f(x)\},\ x_i \in X_i\ , i = 1,2,\dots,N \tag{3.11}$$

Denklem 3.11'de f (x) minimize edilecek amaç fonksiyonunu, x_i her karar değişkeni için kullanılan çözüm uzayını, N ise toplam karar değişkeni sayısını göstermektedir. Harmoni arama algoritmasına ait çözüm parametreleri; Başlangıç Nota Sayısı (BNS), Harmoni Hafızası Boyutu (*HHB*), Harmoni Belleğini Dikkate Alma Oranı (*HHDAO*), İterasyon Sayısı (İS) ve Akort Ayarı Oranı (AAO) şeklinde belirtilebilir.

3.2.1.2. Harmoni belleğinin oluşturulması

Optimizasyon probleminin türüne bağlı olarak harmoni hafızası ya rastgele çözümlerle ya da başlangıç çözümlerinin üretilmesi ile oluşturulmaktadır (Geem ve ark. 2001).

$$HH = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_{N-1}^1 & x_N^1 \\ x_1^2 & x_2^2 & \dots & x_{N-1}^2 & x_N^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^{HHB-1} & x_2^{HHB-1} & \dots & x_{N-1}^{HHB-1} & x_N^{HHB-1} \\ x_1^{HHB} & x_2^{HHB} & \dots & x_{N-1}^{HHB} & x_N^{HHB} \end{bmatrix} \Rightarrow \begin{bmatrix} f(x^1) \\ f(x^2) \\ \vdots \\ f(x^{HHB-1}) \\ f(x^{HHB}) \end{bmatrix} \quad (3.12)$$

3.2.1.3. Yeni harmoni oluşturulması

Bu adımda, yeni harmoni vektörü $x^i=(x^i_1, x^i_2 , x^i_3 ,\dots,x^i_N)$, harmoni belleğindeki tonlara göre ve tamamen rastsal seçilen tonlara göre üretilmektedir. Harmoni belleğinde bulunan tonlara göre, yeni harmoni vektörüne ait ilk karar değişkeni x^i_1 mevcut harmoni belleği içerisindeki herhangi bir değerden rastsal olarak seçilmektedir. Diğer karar değişkenlerinin $(x^i_2,x^i_3 ,\dots,x^i_N)$ seçilmesi ise benzer şekilde yapılmaktadır. Değişkenlerin harmoni belleğinden seçilip seçilmeyeceğinin belirlenmesi, *HHDAO* değerine göre yapılmaktadır. Burada, *HHDAO* (*0* ile *1* arasında değişir) bir karar değişkeninin değerinin mevcut harmoni belleğinden seçilme olasılığını gösterirken, (*1- HHDAO*) oluşturulan yeni karar değişkeninin mevcut çözüm uzayı içerisinden rastsal olarak seçilmesine karşılık gelmektedir. Seçim işleminin yapılış biçimi denklem (3.13)'te gösterilmiştir.

$$x_i' = \begin{cases} HHDAO\ olasılığında, & x_i' \in \{x_i^1, x_i^2, x_i^3, \dots, x_i^{HHB}\} \\ 1 - HHDAO\ olasılığında, & x_i' \in X_i \end{cases} \quad (3.13)$$

Bundan sonra, ton ayarlama işleminin gerekli olup olmadığının belirlenmesi için her karar değişkeninin değerlendirilmesi yapılmaktadır. Akort ayarı denklem (3.14)'e göre yapılmaktadır.

$$x_i' = \begin{cases} AAO\ olasılığında, & x_i' \pm Rst\{0,1\} x\ Bg \\ 1 - AAO\ olasılığında, & x_i' \end{cases} \quad (3.14)$$

Denklem 3.14'te *Bg* rastsal seçilmiş bant genişliğini, *Rst*(*0,1*) ise *0* ile *1* arasında üretilmiş rastsal sayıyı göstermektedir. *AAO* olasılığında x^i_1 karar değişkeni değiştirilmekte, (*1-AAO)* durumunda ise hiçbir şey yapılmamaktadır. *HHDAO* ve AAO parametreleri algoritmanın sırasıyla genel ve yerel optimum çözümleri bulmasında yardımcı olmaktadır (Özyön ve ark. 2011).

3.2.1.4. Harmoni belleğinin güncellenmesi

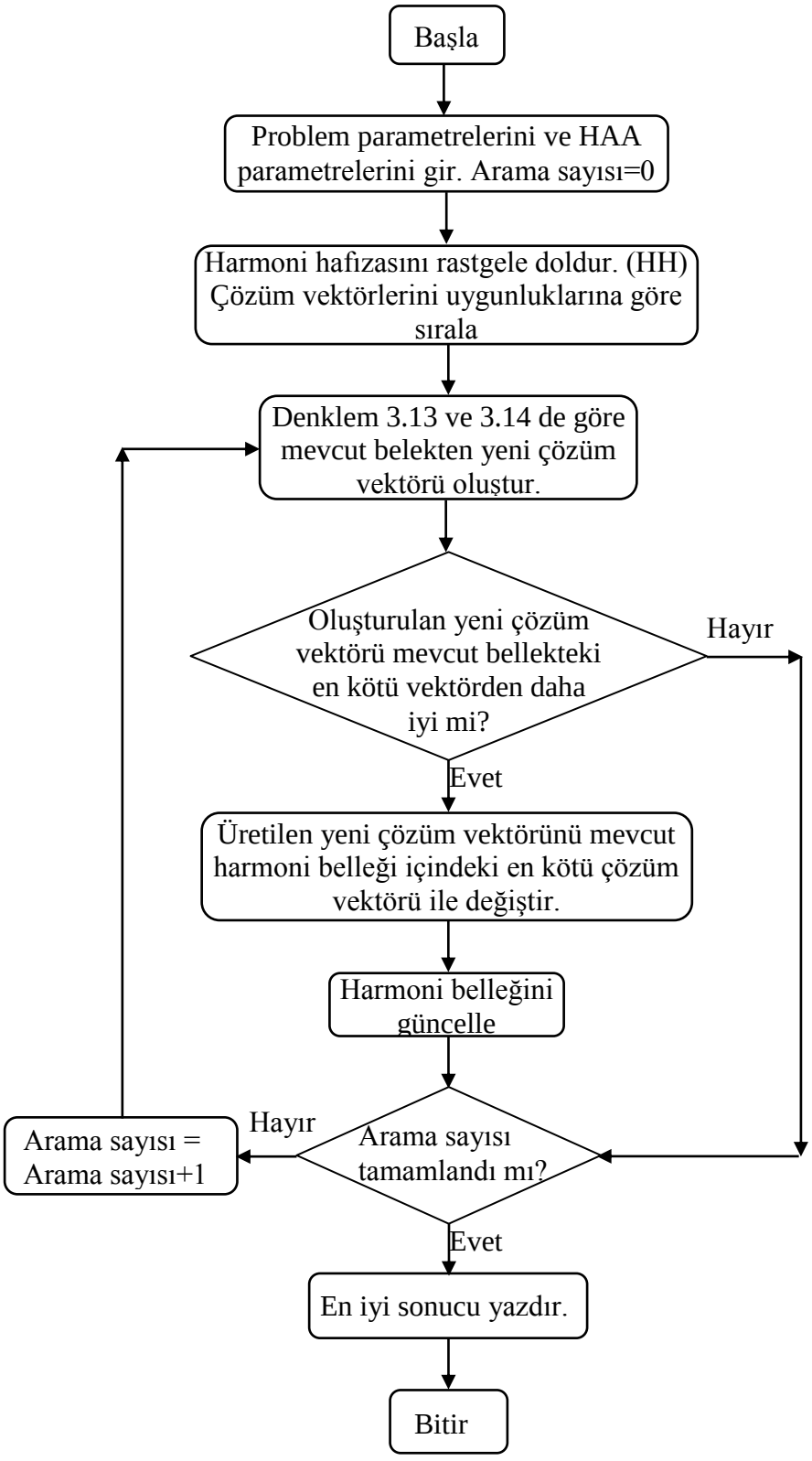
Bu adımda, amaç fonksiyonlarının değerleri bakımından yeni oluşturulan harmoni $x^i=(x^i_1, x^i_2 , x^i_3 ,...,x^i_N)$, ile bellekteki en kötü harmoni arasında karşılaştırma yapılmaktadır. Yeni oluşturulan harmoni vektörünün en kötü harmoniden daha iyi sonuç vermesi durumunda, en kötü harmoni vektörü bellekten çıkarılır ve yeni harmoni vektörü onun yerine atanır.

3.2.1.5. Durma koşulunun kontrolü

Bu adımda verilen durma koşulunun sağlanıp sağlanmadığı kontrol edilir. Koşulun sağlanmaması durumunda yeniden üretim, yeni harmoni oluşturulması, değişkenlerin harmoni belleğinden seçilip seçilmeyeceği (HHDAO), akort ayarlama (AAO) ve harmoni belleğinin güncellenmesi adımları tekrar edilir. Durma koşulu sağlandığında en iyi sonuç alınarak arama sonlandırılır. Bu adımları gösteren harmoni arama optimizasyon algoritmasının akış diyagramı Şekil 3.11'de verilmiştir (Özyön ve ark. 2011).

3.2.2. Kesikli Harmoni Arama Algoritması

Temel Harmoni Arama Algoritması sürekli yapıdadır. Kesikli çizelgeleme problemleri için doğrudan uygun iş permutasyonları üretmesi mümkün değildir. Bu nedenle Kesikli Harmoni Arama Algoritması geliştirilmiştir.



Şekil 3.11. Harmoni arama algoritması için akış şeması (Özyön ve ark. 2011).

Nawaz ve ark. (1983) NEH sezgisellerinin geleneksel iş çizelgeleme problemleri için en etkili sezgiselleden biri olduğunu ifade etmiştir. İş çizelgeleme problemlerinde

temel anlayış, toplam işlem süresi uzun olan işlerin, diğer kısa süreli işlere göre etkinlik açısından yüksek önceliğe sahip olduğu şeklindedir.

Han ve ark. (2010) kesikli problemler için başlangıç permutasyonlarının oluşturulması ve başlangıç prosedürünü aşağıdaki gibi açıklamıştır:

Adım 1: Yeni harmoni üretilir ve Harmoni hafızasına alınır.

Adım 2: Çözüm uzayında rastgele bir harmoni üretilir. Eğer üretilen harmoni, Harmoni Hafızasındaki var olan harmonilerden farklıysa hafızaya alınır; değilse dikkate alınmaz.

Adım 3: Harmoni Hafızası, harmoni hafızası boyutuna (HHB) sahip oluncaya kadar Adım 2 tekrar edilir.

Gelişim tablosu, önemsiz sırayla başlar ve hafıza hesaplama (karşılık) kuralı, ayar düzeltme kuralı ve kontrol parametreleri tarafından, X_{yeni} üretilir (Han ve ark., 2010).

$Q=\{1,2,...,n\}$ çizelgelenmemiş iş sırasını göstermektedir.

$\pi^k=\{\pi^k(1), \pi^k(2),..., \pi^k(n)\}$ karşılıklı iş kümesidir.

Bu bilgiler doğrultusunda geliştirme prosedürü aşağıdaki gibi oluşturulur (Han ve ark., 2010):

"*j = 1* den *n* ' ye kadar yap {

Eğer *Rst( ) < HHDAO* ise { // Hafıza Karşılığı

Eğer $X_a(j) \in Q$ ise // {1,2,…,HHB} kümesinden rastgele seçilen bir dizin anlamına gelir.

$X_{new}(j) = X_a(j)$

Ya da

$X_{new}(j) = \pi^k(1)$

else //*Q* dan rastgele bir iş seçilir ve $X_{new}(j)$ ye konulur.

Eğer

$X_{new}(j)$ işini *Q* ve π^k dan kaldır,

}

Eğer *Rst( )< AAO* o zaman { // ayar düzeltme

Eğer *Rst( )%2* o zaman

Operatörü değiştir

Ya da

Operatörü ekle"

Kesikli harmoni arama algoritmasında, gelişim yeteneğinin artırılabilmesi için yerel arama prosedürü gerekli bir şarttır.

$\pi^k = \{ \pi^k(1), \pi^k(2), \ldots, \pi^k(n) \}$ kümesi şimdiye kadar elde edilmiş en iyi iş sırası kümesini göstermiş olsun. Yerel arama prosedürü aşağıdaki gibi tanımlanır (Gao ve ark., 2011; Han ve ark., 2010);

Adım 1: $i=1$ varsayılır.

Adım 2: X yeni permutasyonunda $\pi^k{}_i$ işi bulunur ve çıkarılır.

Adım 3: $\pi^k{}_i$ işi X_{yeni} permutasyonunda farklı bir konuma yerleştirilir. Mümkün olan tüm yerleştirmeler hesaba katılır ve elde edilen en iyi π^* sırası yazılır. Eğer π^* sırası X_{yeni} den daha iyi ise $X_{yeni} = \pi^*$ kabul edilir.

Adım 4: $i=i+1$ olduğunda eğer $i \leq n$ ise Adım 2'ye gidilir.

Adım 5: X_{yeni} permutasyonu Adım 1'den Adım 4'e kadar geliştirildiğinde, Adım 1'e gidilir, yoksa çıktı şimdiki X_{yeni} permutasyonudur.

3.2.3. Bulanık paralel makine problemleri için parametre analizi

Parametre optimizasyonunda, Oğuz (2005)'un çalışmasında verilen tüm problemler çözülmüştür. Bulunan en iyi parametre setleri ile o gruptaki bütün problemler tekrar çözülmüştür. Oğuz (2005) çok prosesli esnek akış tipi kıyaslama problemlerini, zorluk derecesine göre P ve Q olmak üzere iki türe ayırmıştır. Her tipte de 2, 5 ve 8 aşama bulunmaktadır.

Optimizasyon sonucunda her bir örnek problem için en kısa sürede elde edilen en küçük FC_{max} değeri, başlangıç nota sayısı, Harmoni hafızası kapasitesi, iterasyon sayısı ve HMCR oranı bulunmuştur. *FCmax* değeri, bulanıklaştırılan problemlerdeki işlerin tamamlanma zamanıdır. Tablolardaki yer alan CPU zamanı saniye cinsinden verilmiştir.

Bu çalışmada kullanılan esnek akış tipi problemler, Oğuz (2005)'un kıyaslama problemleridir. Oğuz kıyaslama problemlerinde *n(iş) x s(aşama)* tipi problemlerdir. Örnek problem notasyonu "P20S5T08" şeklindedir. Burada P problem zorluk derecesini göstermekte olup P ve Q değerlerini almaktadır. Q tipi problemler daha zordur. 20 iş sayısını göstermektedir. Burada 100 iş H1 ile temsil edilmektedir. S5 aşama sayısını, T08 ise problem indisini göstermektedir. n=5, 10, 20, 50, 100 iş sayıları, m=2, 5, 8 değerleri ise aşama sayılarıdır. Her işlemci herhangi bir anda sadece bir işlemi

yürütebilmektedir. İşlemciler arıza yapmazlar. Bütün işler çizelgeleme başında hazırdırlar. İşler kesintisiz işlenir. Amaç, FC_{max}'ın en küçüklenmesidir.

Bu problemleri çözmek için geliştirilen Harmoni arama algoritmasında 4 parametre kullanılmaktadır. Her ayrı tip problem türü için gerekli parametreleri belirlemek amacıyla olası tüm kombinasyonlar test edilmiştir. Her faktörde birden fazla düzey bulunmaktadır. Bu düzeyler Çizelge 3.1' de verilmiştir.

Parametrelerin belirlenebilmesi için tüm problemler (300 problem) 300x4x3x3x3 = 32.400 defa çözülmüştür. Çözüm sonuçları için en önemli faktör FC_{max} değeridir. FC_{max} değerinden sonra dikkate alınan çözüm süresidir. Çözüm süreleri eşit ise iterasyon sayıları dikkate alınmaktadır.

Çizelge 3.1. Faktörler için Kullanılan Düzeyler.

Faktör	**Düzeyler**
Başlangıç Nota Sayısı	100-1000-5000-10000
Harmoni Hafızası Boyutu (HHB)	10-100-1000
İterasyon Sayısı	50-100-500
HHDAO Oranı (%)	30-50-80
AAO Oranı	1

Başlangıç nota sayısı CPU değerlerini aşırı derecede arttırmayacak şekilde problem setinde verilen maksimum iş sayısının (100) katları olarak belirlenmiştir (100, 100x10,100x50,100x100). Harmoni hafızası boyutu, oluşturulan başlangıç nota değerlerinden seçilecek en iyi grubun büyüklüğünü ifade etmektedir. Parametre düzeyleri problem boyutları dikkate alınarak belirlenmiştir (10x1,10x10,10x100). İterasyon sayısı algoritmanın kaç defa tekrar edileceğini ifade etmektedir ve CPU değeri üzerinde çok etkilidir. CPU zamanını ve problem boyutunu dikkate alarak parametre düzeyleri belirlenmiştir. HMCR oranı yeni bireyin oluşturulması esnasında eski notaların dikkate alınması oranıdır. Bu oranın çok az olduğu durumlarda çeşitlilik artmaktadır. Oranın yüksek olduğu durumlarda ise çözüm uzayının belli bir bölgesinde arama yapılmakta ve yerel optimum noktalara yakalanma riski artmaktadır. Bu mantığa göre ara değerler parametre düzeyi olarak belirlenmiştir. PAR oranı bulunan yeni harmoninin hafızaya alınma oranını belirlemektedir. Yapılan çalışmada başarılı bulunan

her sonucun hafızaya alınması istendiği için oran 1 kabul edilmiştir. Yani bulunan her iyi sonuç hafızaya alınmıştır.

3.2.4.Paralel makine problemlerinin bulanıklaştırılması

Gözen (2007) çalışmasında Oğuz (2005)'un çok işlemcili paralel makine problemlerini bulanıklaştırmıştır. Bu çalışmada Gözen (2007)'in bulanıklaştırdığı problem seti kullanılmıştır. Problemin amaçları aşağıdaki gibi belirlenmiştir (Gonzales ve ark. 2005).

1- Ortalama uyumlaştırma indeksinin maksimizasyonu,

2- Uyumlaştırma indeksinin maksimizasyonu,

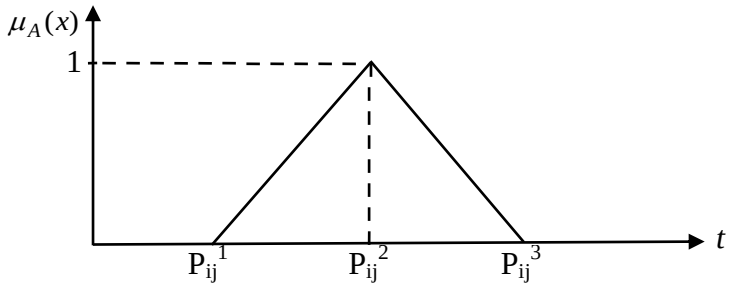
3- Bulanık tamamlanma zamanının minimizasyonu olarak belirlenmiştir.

Problemin bulanık kısmında aşağıdaki üyelik fonksiyonları kullanılmıştır (Gonzales ve ark. 2005).

1- Proses zamanı (Üçgen Üyelik Fonksiyonu)

2- Teslim süresi (Yamuk Üyelik Fonksiyonu)

Bulanık üçgensel üyelik fonksiyonu yardımıyla hesaplanan proses zamanları Şekil 3.12'de gösterilmiştir.



Şekil 3.12. Bulanık Proses Zamanı (Gonzales ve ark. 2005).

P_{ij}^1 =En kısa işlem süresi

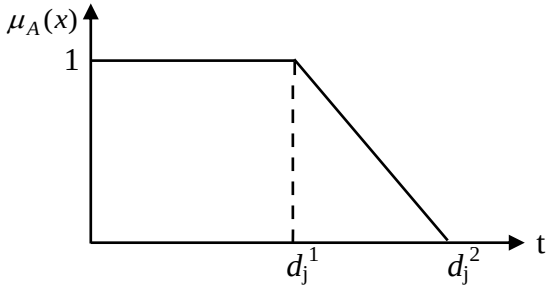
P_{ij}^2 =Orta işlem süresi

P_{ij}^3 =En uzun işlem süresi

$P_{ij}^1 \leq P_{ij}^2 \leq P_{ij}^3$

$$\mu_A(x) = \begin{cases} 0 & x < P_{ij}^1 \\ \frac{x - P_{ij}^1}{P_{ij}^2 - P_{ij}^1} & P_{ij}^1 \le x \le P_{ij}^2 \\ \frac{x - P_{ij}^3}{P_{ij}^2 - P_{ij}^3} & P_{ij}^2 \le x \le P_{ij}^3 \\ 0 & P_{ij}^3 < x \end{cases} \quad (3.15)$$

Bulanık yamuk üyelik fonksiyonu yardımıyla hesaplanan teslim zamanları Şekil 3.13'de gösterilmiştir (Gonzales ve ark. 2005).



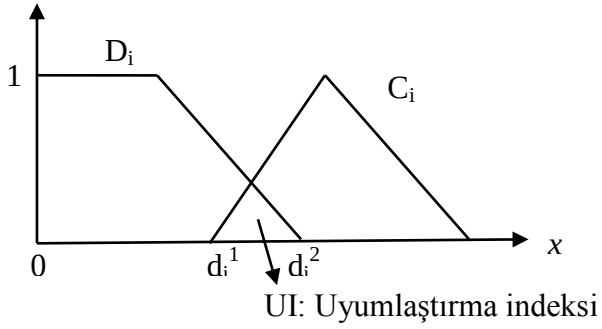
Şekil 3.13. Bulanık teslim tarihi (Gonzales ve ark. 2005).

d^1_j = Teslim tarihi

d^2_j = Üst sınır teslim tarihi

$$\mu_A(x) = \begin{cases} 0 & x < d_j^1 \\ \frac{x - d_j^1}{d_j^2 - d_j^1} & d_j^1 < x \le d_j^2 \\ 0 & x \ge d_j^1 \end{cases} \quad (3.16)$$

Problemin amaç fonksiyonlarından uyumlaştırma indeksi Şekil 3.14'te gösterilmiştir.



Şekil 3.14. Uyumlaştırma indeksi (Gözen 2007).

Di= Teslim süresi

Ci= İşlem süresi

$$UI = \frac{Alan(C_i \cap D_i)}{Alan(C_i)} \tag{3.17}$$

Ci= Her bir işin toplam tamamlanma zamanı

Amaç fonksiyonlarının formülleri (Gonzales ve ark. 2005):

1- *Ortalama Uyumlaştırma İndeksi maksimizasyonu:*

$$maksZ_1 = \frac{1}{n}\sum_{i=1}^{n} UI_i \tag{3.18}$$

2- *Uyumlaştırma İndeksi maksimizasyonu:*

$$maksZ_2 = UI_{min} = \min_{i=1,\ldots,n} UI_i \tag{3.19}$$

3- *Bulanık tamamlanma zamanı:*

$$minZ_3 = FC_{maks} = \frac{MinC_{maks} + 2.OrtC_{maks} + MaksC_{maks}}{4} \tag{3.20}$$

Bulanık paralel makina problemleri harmoni arama algoritması programıyla çözülmüştür. Yapılan testlerde harmoni arama algoritmasının çözüm kalitesi ve çözüme ulaşmada harcadığı süre (CPU zamanı) ölçülmüştür. Çözüm kalitesi, Gözen (2007) ve Yılmaz (2008)'ın sonuçlarıyla kıyaslanmıştır. Yüzde sapma, FC_{max} için (3.21)' deki ifadeye göre hesaplanır.

$$\% \text{ Sapma} = \frac{GA - HAA}{HAA} \times 100$$

$$\% \text{ Sapma} = \frac{TB - HAA}{HAA} \times 100 \tag{3.21}$$

$$\% \text{ Sapma} = \frac{YBA - HAA}{HAA} \times 100$$

Yüzde sapma ORT UI ve UI değerleri için (3.22)' deki ifadeye göre hesaplanır.

$$\% \text{ Sapma} = \frac{HAA - GA}{GA} \times 100$$

$$\% \text{ Sapma } = \frac{HAA - TB}{TB} \times 100 \qquad (3.22)$$

$$\% \text{ Sapma } = \frac{HAA - YBA}{YBA} \times 100$$

Örnek problemlerin bulunduğu veri setlerinin çözümü için harmoni arama algoritması aşağıda belirtildiği gibi 5 adımda uygulanır (Akkoyunlu ve Engin, 2011);

1. *Problemin kurulması ve çözüm parametrelerinin belirlenmesi:* Bu aşamada veri setinden okuma yapılarak değerleri önceki kısımlarda belirtildiği şekilde bulanıklaştırılmaktadır. Ardından parametre analizi kısmında ifade edildiği gibi parametreler seçilmektedir.
2. *Harmoni belleğinin oluşturulması:* Verilen problemin boyutları dikkate alınarak parametreler doğrultusunda başlangıç notası adedince rastgele çözümler oluşturulmaktadır. Bu çözümlerin her biri için FC_{maks} değerleri hesaplanarak en başarılı sonuçlar harmoni hafızası boyutu büyüklüğünde bir dizi olarak sıralanmaktadır.
3. *Yeni harmoninin oluşturulması:* Harmoni belleğindeki $x_1,x_2,...x_n$ değerler arasından parametreler doğrultusunda seçim yapılarak yeni harmoni oluşturulmaktadır. Yeni harmoni için FC_{maks} değeri hesaplanmaktadır. Elde edilen bu harmoni vektörü "swap" ve "insert" operatörleri kullanılarak geliştirilmeye çalışılmaktadır.

   SWAP Operatörü: *(1,2,3,4,5)* şeklinde örnek vektör için işlem şöyledir; program 2 adet rastgele sayı atar (örneğin 1 ve 4 seçilsin), atanan bu sayılar için vektördeki değerlerin konumlarını değiştirir. Böylece elimizdeki vektör *(4,2,3,1,5)* halini alır.

   INSERT Operaörü: *(1,2,3,4,5)* örnek vektörümüz için yine 2 rastgele sayı atama işlemi yapılır (Örneğin 1 ve 4 seçilsin), atanan bu sayılardan ilk değerdeki, ikinci değerdeki pozisyona taşınır. Böylece elimizdeki vektör *(2,3,4,1,5)* halini alır.
4. *Harmoni belleğinin güncellenmesi:* Geliştirilen yeni harmoni vektörü harmoni belleğine PAR oranı dikkate alınarak güncellenmektedir. Yani hafızadaki en kötü değer çıkarılırarak, elde edilen yeni harmoni belleğe eklenmektedir.

5. *Durma koşulunun kontrolü:* Algoritmanın ilk aşamasında belirtilen iterasyon sayısı kadar işlemler tekrarlanmaktadır. Problem iterasyon sayısı kadar tekrarın ardından sonlandırılmaktadır. Elde edilen sonuçlar çıktı dosyasına yazdırılmaktadır.

4. ARAŞTIRMA SONUÇLARI

Oğuz (2005)'un Gözen (2007) tarafından bulanıklaştırılan çok prosesli esnek akış tipi problemleri (300 adet problem), harmoni arama algoritması kullanılarak bilgisayar programı yardımıyla çözülmüştür. Algoritmaya ait çözümler Gözen (2007) ve Yılmaz (2008)'in sonuçları ile kıyaslanmıştır.

Bütün problem çözümleri Intel R Core 2 CPU T6400 2.00 GHz işlemcili, 2.99 GB Ram ve Microsoft Windows XP SP3 işletim sistemi özellikli bilgisayarda gerçekleştirilmiştir.

Çözülen 300 problem için HAA'nın kıyaslamaları Çizelge 4.1 ve 4.2'de, tüm çözümlerin genel sonuçları ise Çizelge 4.3'te verilmiştir.

P10S2 tipi problemlerin tümünde GA ve TB'den daha başarılı sonuçlar bulunmuştur. 1, 2, 3 ve 8 numaralı problemlerde YBA'dan daha başarılı sonuçlar; 4, 5, 6, 7, 9 ve 10 numaralı problemlerde ise YBA ile aynı sonuçlar bulunmuştur.

P10S5 tipi problemlerin tümünde GA ve TB'den daha başarılı sonuçlar bulunmuştur. Problemlerin 9 tanesinde YBA'dan daha başarılı sonuçlar bulunurken sadece 3 numaralı problemde YBA'dan daha kötü sonuç bulunmuştur.

P10S8 tipi problemlerin tümünde GA ve TB'den daha başarılı sonuçlar bulunmuştur. Ayrıca 2, 5 ve 10 numaralı problemlerde YBA'dan daha kötü sonuçlar bulunmasına rağmen diğer 7 problemin tamamında YBA 'dan daha başarılı sonuçlar elde edilmiştir.

P20S2 tipi problemlerin tümünde GA ve TB'den daha başarılı sonuçlar bulunmuştur. Ayrıca 1, 2, 8 ve 10 numaralı problemlerde YBA'dan daha kötü sonuçlar bulunmasına rağmen, 9 numaralı problemde aynı, diğer problemlerde ise daha başarılı sonuçlar bulunmuştur.

P20S5 tipi problemlerin tümünde YBA, GA ve TB'den daha başarılı sonuçlar bulunmuştur.

P20S8 tipi problemlerin tümünde YBA, GA ve TB'den daha başarılı sonuçlar bulunmuştur.

Çizelge 4.1. Kıyaslamalı sonuçlar.

Problem	HAA - YB (Fcmaks)	HAA - GA (Fcmaks)	HAA - TB (Fcmaks)	Problem	HAA - YB (Fcmaks)	HAA - GA (Fcmaks)	HAA - TB (Fcmaks)
P10S2T01	İyi	İyi	İyi	P20S5T03	İyi	İyi	İyi
P10S2T02	İyi	İyi	İyi	P20S5T04	İyi	İyi	İyi
P10S2T03	İyi	İyi	İyi	P20S5T05	İyi	İyi	İyi
P10S2T04	Eşit	İyi	İyi	P20S5T06	İyi	İyi	İyi
P10S2T05	Eşit	İyi	İyi	P20S5T07	İyi	İyi	İyi
P10S2T06	Eşit	İyi	İyi	P20S5T08	İyi	İyi	İyi
P10S2T07	Eşit	İyi	İyi	P20S5T09	İyi	İyi	İyi
P10S2T08	İyi	İyi	İyi	P20S5T10	İyi	İyi	İyi
P10S2T09	Eşit	İyi	İyi	P20S8T01	İyi	İyi	İyi
P10S2T10	Eşit	İyi	İyi	P20S8T02	İyi	İyi	İyi
P10S5T01	İyi	İyi	İyi	P20S8T03	İyi	İyi	İyi
P10S5T02	İyi	İyi	İyi	P20S8T04	İyi	İyi	İyi
P10S5T03	Kötü	İyi	İyi	P20S8T05	İyi	İyi	İyi
P10S5T04	İyi	İyi	İyi	P20S8T06	İyi	İyi	İyi
P10S5T05	İyi	İyi	İyi	P20S8T07	İyi	İyi	İyi
P10S5T06	İyi	İyi	İyi	P20S8T08	İyi	İyi	İyi
P10S5T07	İyi	İyi	İyi	P20S8T09	İyi	İyi	İyi
P10S5T08	İyi	İyi	İyi	P20S8T10	İyi	İyi	İyi
P10S5T09	İyi	İyi	İyi	P50S2T01	Kötü	İyi	İyi
P10S5T10	İyi	İyi	İyi	P50S2T02	Eşit	İyi	İyi
P10S8T01	İyi	İyi	İyi	P50S2T03	Kötü	İyi	İyi
P10S8T02	Kötü	İyi	İyi	P50S2T04	Kötü	İyi	İyi
P10S8T03	İyi	İyi	İyi	P50S2T05	Kötü	İyi	İyi
P10S8T04	İyi	İyi	İyi	P50S2T06	İyi	İyi	İyi
P10S8T05	Kötü	İyi	İyi	P50S2T07	Kötü	İyi	İyi
P10S8T06	İyi	İyi	İyi	P50S2T08	İyi	İyi	İyi
P10S8T07	İyi	İyi	İyi	P50S2T09	İyi	İyi	İyi
P10S8T08	İyi	İyi	İyi	P50S2T10	Kötü	İyi	İyi
P10S8T09	İyi	İyi	İyi	P50S5T01	İyi	İyi	İyi
P10S8T10	Kötü	İyi	İyi	P50S5T02	İyi	İyi	İyi
P20S2T01	Kötü	İyi	İyi	P50S5T03	İyi	İyi	İyi
P20S2T02	Kötü	İyi	İyi	P50S5T04	İyi	İyi	İyi
P20S2T03	İyi	İyi	İyi	P50S5T05	İyi	İyi	İyi
P20S2T04	İyi	İyi	İyi	P50S5T06	Kötü	İyi	İyi
P20S2T05	İyi	İyi	İyi	P50S5T07	İyi	İyi	İyi
P20S2T06	İyi	İyi	İyi	P50S5T08	İyi	İyi	İyi
P20S2T07	İyi	İyi	İyi	P50S5T09	İyi	İyi	İyi
P20S2T08	Kötü	İyi	İyi	P50S5T10	İyi	İyi	İyi
P20S2T09	Eşit	İyi	İyi	P50S8T01	İyi	İyi	İyi
P20S2T10	Kötü	İyi	İyi	P50S8T02	İyi	İyi	İyi
P20S5T01	İyi	İyi	İyi	P50S8T03	İyi	İyi	İyi
P20S5T02	İyi	İyi	İyi	P50S8T04	İyi	İyi	İyi

Çizelge 4.1. Kıyaslamalı sonuçlar (Devam).

Problem	HAA - YB (Fcmaks)	HAA - GA (Fcmaks)	HAA - TB (Fcmaks)	Problem	HAA - YB (Fcmaks)	HAA - GA (Fcmaks)	HAA - TB (Fcmaks)
P50S8T05	İyi	İyi	İyi	PH1S2T07	Kötü	Kötü	Kötü
P50S8T06	İyi	İyi	İyi	PH1S2T08	İyi	İyi	İyi
P50S8T07	İyi	İyi	İyi	PH1S2T09	Kötü	Kötü	Kötü
P50S8T08	İyi	İyi	İyi	PH1S2T10	Kötü	Kötü	Kötü
P50S8T09	İyi	İyi	İyi	PH1S5T01	İyi	İyi	İyi
P50S8T10	İyi	İyi	İyi	PH1S5T02	İyi	İyi	İyi
P5S2T01	Eşit	İyi	İyi	PH1S5T03	Kötü	İyi	İyi
P5S2T02	Eşit	İyi	İyi	PH1S5T04	Kötü	İyi	İyi
P5S2T03	Eşit	İyi	İyi	PH1S5T05	İyi	İyi	İyi
P5S2T04	Kötü	İyi	İyi	PH1S5T06	Kötü	İyi	İyi
P5S2T05	Kötü	İyi	İyi	PH1S5T07	İyi	İyi	İyi
P5S2T06	Eşit	İyi	İyi	PH1S5T08	İyi	İyi	İyi
P5S2T07	Kötü	İyi	İyi	PH1S5T09	İyi	İyi	İyi
P5S2T08	Eşit	İyi	İyi	PH1S5T10	Kötü	İyi	İyi
P5S2T09	Eşit	İyi	İyi	PH1S8T01	İyi	İyi	İyi
P5S2T10	Eşit	İyi	İyi	PH1S8T02	İyi	İyi	İyi
P5S5T01	Kötü	İyi	İyi	PH1S8T03	İyi	İyi	İyi
P5S5T02	Kötü	İyi	İyi	PH1S8T04	İyi	İyi	İyi
P5S5T03	Eşit	İyi	İyi	PH1S8T05	İyi	İyi	İyi
P5S5T04	Kötü	İyi	İyi	PH1S8T06	İyi	İyi	İyi
P5S5T05	Kötü	İyi	İyi	PH1S8T07	İyi	İyi	İyi
P5S5T06	Kötü	İyi	İyi	PH1S8T08	İyi	İyi	İyi
P5S5T07	Kötü	İyi	İyi	PH1S8T09	İyi	İyi	İyi
P5S5T08	Eşit	İyi	İyi	PH1S8T10	İyi	İyi	İyi
P5S5T09	Kötü	İyi	İyi	Q10S2T01	Eşit	İyi	İyi
P5S5T10	Kötü	İyi	İyi	Q10S2T02	İyi	İyi	İyi
P5S8T01	Kötü	İyi	İyi	Q10S2T03	Kötü	İyi	İyi
P5S8T02	Kötü	İyi	İyi	Q10S2T04	Kötü	İyi	İyi
P5S8T03	Kötü	İyi	İyi	Q10S2T05	İyi	İyi	İyi
P5S8T04	Kötü	İyi	İyi	Q10S2T06	Kötü	İyi	İyi
P5S8T05	Kötü	İyi	İyi	Q10S2T07	Kötü	İyi	İyi
P5S8T06	Eşit	İyi	İyi	Q10S2T08	Eşit	İyi	İyi
P5S8T07	Kötü	İyi	İyi	Q10S2T09	Kötü	İyi	İyi
P5S8T08	Kötü	İyi	İyi	Q10S2T10	İyi	İyi	İyi
P5S8T09	İyi	İyi	İyi	Q10S5T01	İyi	İyi	İyi
P5S8T10	Kötü	İyi	İyi	Q10S5T02	İyi	İyi	İyi
PH1S2T01	Kötü	Kötü	İyi	Q10S5T03	İyi	İyi	İyi
PH1S2T02	İyi	İyi	İyi	Q10S5T04	Kötü	İyi	İyi
PH1S2T03	Kötü	Kötü	Kötü	Q10S5T05	İyi	İyi	İyi
PH1S2T04	İyi	İyi	İyi	Q10S5T06	Kötü	İyi	İyi
PH1S2T05	İyi	İyi	İyi	Q10S5T07	İyi	İyi	İyi
PH1S2T06	İyi	İyi	İyi	Q10S5T08	İyi	İyi	İyi

Çizelge 4.1. Kıyaslamalı sonuçlar (Devam).

Problem	HAA - YB (Fcmaks)	HAA - GA (Fcmaks)	HAA - TB (Fcmaks)	Problem	HAA - YB (Fcmaks)	HAA - GA (Fcmaks)	HAA - TB (Fcmaks)
Q10S5T09	Kötü	İyi	İyi	Q50S2T01	Kötü	İyi	İyi
Q10S5T10	İyi	İyi	İyi	Q50S2T02	Kötü	Kötü	İyi
Q10S8T01	İyi	İyi	İyi	Q50S2T03	Kötü	İyi	İyi
Q10S8T02	İyi	İyi	İyi	Q50S2T04	Kötü	İyi	İyi
Q10S8T03	İyi	İyi	İyi	Q50S2T05	Kötü	İyi	İyi
Q10S8T04	İyi	İyi	İyi	Q50S2T06	Kötü	Kötü	İyi
Q10S8T05	İyi	İyi	İyi	Q50S2T07	Kötü	İyi	İyi
Q10S8T06	İyi	İyi	İyi	Q50S2T08	Kötü	İyi	İyi
Q10S8T07	İyi	İyi	İyi	Q50S2T09	Kötü	İyi	İyi
Q10S8T08	İyi	İyi	İyi	Q50S2T10	Kötü	İyi	İyi
Q10S8T09	İyi	İyi	İyi	Q50S5T01	İyi	İyi	İyi
Q10S8T10	İyi	İyi	İyi	Q50S5T02	Kötü	İyi	İyi
Q20S2T01	Kötü	İyi	İyi	Q50S5T03	Kötü	İyi	İyi
Q20S2T02	İyi	İyi	İyi	Q50S5T04	İyi	İyi	İyi
Q20S2T03	Kötü	İyi	İyi	Q50S5T05	Kötü	İyi	İyi
Q20S2T04	İyi	İyi	İyi	Q50S5T06	Eşit	İyi	İyi
Q20S2T05	Kötü	İyi	İyi	Q50S5T07	Kötü	İyi	İyi
Q20S2T06	Kötü	İyi	İyi	Q50S5T08	İyi	İyi	İyi
Q20S2T07	Kötü	İyi	İyi	Q50S5T09	Kötü	İyi	İyi
Q20S2T08	Kötü	İyi	İyi	Q50S5T10	Kötü	İyi	İyi
Q20S2T09	Eşit	İyi	İyi	Q50S8T01	Kötü	İyi	İyi
Q20S2T10	Kötü	İyi	İyi	Q50S8T02	İyi	İyi	İyi
Q20S5T01	İyi	İyi	İyi	Q50S8T03	İyi	İyi	İyi
Q20S5T02	İyi	İyi	İyi	Q50S8T04	İyi	İyi	İyi
Q20S5T03	İyi	İyi	İyi	Q50S8T05	İyi	İyi	İyi
Q20S5T04	Kötü	İyi	İyi	Q50S8T06	İyi	İyi	İyi
Q20S5T05	İyi	İyi	İyi	Q50S8T07	İyi	İyi	İyi
Q20S5T06	İyi	İyi	İyi	Q50S8T08	İyi	İyi	İyi
Q20S5T07	Kötü	İyi	İyi	Q50S8T09	Kötü	İyi	İyi
Q20S5T08	İyi	İyi	İyi	Q50S8T10	İyi	İyi	İyi
Q20S5T09	Kötü	İyi	İyi	Q5S2T01	Kötü	İyi	İyi
Q20S5T10	Kötü	İyi	İyi	Q5S2T02	Kötü	İyi	İyi
Q20S8T01	İyi	İyi	İyi	Q5S2T03	Eşit	İyi	İyi
Q20S8T02	İyi	İyi	İyi	Q5S2T04	Eşit	İyi	İyi
Q20S8T03	İyi	İyi	İyi	Q5S2T05	Kötü	İyi	İyi
Q20S8T04	İyi	İyi	İyi	Q5S2T06	Kötü	İyi	İyi
Q20S8T05	İyi	İyi	İyi	Q5S2T07	Kötü	İyi	İyi
Q20S8T06	İyi	İyi	İyi	Q5S2T08	Eşit	İyi	İyi
Q20S8T07	İyi	İyi	İyi	Q5S2T09	Kötü	İyi	İyi
Q20S8T08	İyi	İyi	İyi	Q5S2T10	Kötü	İyi	İyi
Q20S8T09	İyi	İyi	İyi	Q5S5T01	Kötü	İyi	İyi
Q20S8T10	İyi	İyi	İyi	Q5S5T02	Kötü	İyi	İyi

Çizelge 4.1. Kıyaslamalı sonuçlar (Devam).

Problem	HAA - YB (Fcmaks)	HAA - GA (Fcmaks)	HAA - TB (Fcmaks)	Problem	HAA - YB (Fcmaks)	HAA - GA (Fcmaks)	HAA - TB (Fcmaks)
Q5S5T03	Kötü	İyi	İyi	QH1S2T07	Kötü	Kötü	Kötü
Q5S5T04	Kötü	İyi	İyi	QH1S2T08	Kötü	Kötü	Kötü
Q5S5T05	Kötü	İyi	İyi	QH1S2T09	Kötü	Kötü	Kötü
Q5S5T06	Kötü	İyi	İyi	QH1S2T10	Kötü	Kötü	İyi
Q5S5T07	Kötü	İyi	İyi	QH1S5T01	Kötü	İyi	İyi
Q5S5T08	İyi	İyi	İyi	QH1S5T02	Kötü	İyi	İyi
Q5S5T09	Kötü	İyi	İyi	QH1S5T03	Kötü	İyi	İyi
Q5S5T10	Kötü	İyi	İyi	QH1S5T04	Kötü	İyi	İyi
Q5S8T01	Eşit	İyi	İyi	QH1S5T05	Kötü	İyi	İyi
Q5S8T02	Kötü	İyi	İyi	QH1S5T06	Kötü	İyi	İyi
Q5S8T03	Kötü	İyi	İyi	QH1S5T07	Kötü	İyi	İyi
Q5S8T04	Eşit	İyi	İyi	QH1S5T08	Kötü	İyi	İyi
Q5S8T05	Kötü	İyi	İyi	QH1S5T09	Kötü	İyi	İyi
Q5S8T06	İyi	İyi	İyi	QH1S5T10	Kötü	İyi	İyi
Q5S8T07	Kötü	İyi	İyi	QH1S8T01	İyi	İyi	İyi
Q5S8T08	Kötü	İyi	İyi	QH1S8T02	Kötü	İyi	İyi
Q5S8T09	Kötü	İyi	İyi	QH1S8T03	Kötü	İyi	İyi
Q5S8T10	Kötü	İyi	İyi	QH1S8T04	Kötü	İyi	İyi
QH1S2T01	Kötü	Kötü	Kötü	QH1S8T05	Kötü	İyi	İyi
QH1S2T02	Kötü	Kötü	Kötü	QH1S8T06	Kötü	İyi	İyi
QH1S2T03	Kötü	Kötü	Kötü	QH1S8T07	Kötü	İyi	İyi
QH1S2T04	Kötü	Kötü	Kötü	QH1S8T08	Kötü	İyi	İyi
QH1S2T05	Kötü	Kötü	Kötü	QH1S8T09	Kötü	İyi	İyi
QH1S2T06	Kötü	Kötü	Kötü	QH1S8T10	İyi	İyi	İyi

P50S2 tipi problemlerin tümünde GA ve TB‘den daha başarılı sonuçlar bulunmuştur. Ayrıca 6, 8 ve 9 numaralı problemlerde YBA‘dan daha başarılı, 2 numaralı problemde YBA ile aynı, diğer 6 problemde ise YBA’dan daha kötü sonuç bulunmuştur.

P50S5 tipi problemlerin tümünde GA ve TB‘den daha başarılı sonuçlar bulunmuştur. Ayrıca yalnızca 6 numaralı problemde YBA‘dan daha kötü sonuç bulunurken diğer tüm problemlerde YBA‘dan daha başarılı sonuçlar bulunmuştur.

P50S8 tipi problemlerin tümünde YBA, GA ve TB ‘den daha başarılı sonuçlar bulunmuştur.

P5S2 tipi problemlerin tümünde GA ve TB‘den daha başarılı sonuçlar bulunmuştur. Ayrıca 6 problemde YBA ile aynı sonuçlar, 3 problemde YBA‘dan daha

başarılı sonuçlar, 1 problemde ise YBA ‘dan daha kötü sonuç (2 numaralı problem) elde edilmiştir.

P5S5 tipi problemlerin tümünde GA ve TB‘den daha başarılı sonuçlar bulunmuştur. Ayrıca 3 ve 8 numaralı problemlerde YBA ile aynı sonuçlar bulunmasına rağmen kalan 8 problemde YBA‘dan kötü sonuçlar elde edilmiştir.

P5S8 tipi problemlerin tümünde GA ve TB‘den daha başarılı sonuçlar bulunmuştur. Ayrıca 6 numaralı problemde YBA ile aynı, 9 numaralı problemde YBA‘dan daha başarılı, kalan 8 problemde ise YBA‘dan daha kötü sonuçlar bulunmuştur.

PH1S2 tipi problemlerden 1 numaralı problemde YBA ve GA‘dan kötü sonuç bulunmasına rağmen TB ‘den daha iyi sonuç bulunmuştur. 2, 4, 5, 6 ve 8 numaralı problemlerde ise diğer algoritmalardan başarılı sonuçlar bulunmuştur. Kalan 4 problemde ise üç algoritmadan da daha başarısız sonuçlar elde edilmiştir.

PH1S5 tipi problemlerin tümünde GA ve TB‘den daha başarılı sonuçlar bulunmuştur. Ayrıca 1, 2 ve 5 numaralı problemlerde YBA‘dan daha başarılı, diğer 7 problemde ise YBA‘dan daha kötü sonuçlar bulunmuştur.

PH1S8 tipi problemlerin tümünde YBA, GA ve TB ‘den daha başarılı sonuçlar bulunmuştur.

Q10S2 tipi problemlerin tümünde GA ve TB ‘den daha başarılı sonuçlar bulunmuştur. Ayrıca 1ve 8 numaralı problemlerde YBA ile aynı sonuçlar, 2, 5 ve 10 numaralı problemlerde YBA‘dan daha başarılı sonuçlar, kalan 5 problemde ise YBA‘dan daha kötü sonuçlar bulunmuştur.

Q10S5 tipi problemlerin tümünde GA ve TB ‘den daha başarılı sonuçlar bulunmuştur. Ayrıca 4, 6 ve 9 numaralı problemlerde YBA ‘dan daha kötü sonuçlar bulunmasına rağmen kalan 7 problemde YBA’dan daha başarılı sonuçlar bulunmuştur.

Q10S8 tipi problemlerin tümünde YBA, GA ve TB‘den daha başarılı sonuçlar bulunmuştur.

Q20S2 tipi problemlerin tümünde GA ve TB‘den daha başarılı sonuçlar bulunmuştur. Ayrıca 2 ve 4 numaralı problemlerde YBA‘dan daha başarılı, 9 numaralı problemde YBA ile aynı, kalan 7 problemde ise YBA‘dan daha başarısız sonuçlar elde edilmiştir.

Q20S5 tipi problemlerin tümünde GA ve TB‘den daha başarılı sonuçlar bulunmuştur. Ayrıca 4, 7, 9 ve 10 numaralı problemlerde YBA‘dan daha kötü sonuçlar

elde edilmesine rağmen kalan 6 problemde YBA‘dan daha başarılı sonuçlar bulunmuştur.

Q20S8 tipi problemlerin tümünde YBA, GA ve TB‘den daha başarılı sonuçlar bulunmuştur.

Q50S2 tipi problemlerin tümünde YBA‘dan daha kötü sonuçlar bulunmuştur. 2 ve 7 numaralı problemlerde GA‘dan daha kötü sonuçlar bulunmasına rağmen kalan 8 problemde GA‘dan daha iyi sonuçlar bulunmuştur. Ayrıca Q50S2 problemlerinin tamamında TB‘den daha başarılı sonuçlar bulunmuştur.

Q50S5 tipi problemlerin tümünde GA ve TB‘den daha başarılı sonuçlar bulunmuştur. Ayrıca 1, 4 ve 8 numaralı problemlerde YBA‘dan daha başarılı, 6 numaralı problemde YBA ile aynı, kalan 6 problemde ise YBA‘dan daha kötü sonuçlar elde edilmiştir.

Q50S8 tipi problemlerin tümünde GA ve TB‘den daha başarılı sonuçlar bulunmuştur. Ayrıca 1 ve 9 numaralı problemlerde YBA‘dan daha kötü sonuçlar bulunmasına rağmen kalan 8 problemde YBA’dan daha başarılı sonuçlar elde edilmiştir.

Q5S2 tipi problemlerin tümünde GA ve TB ‘den daha başarılı sonuçlar bulunmuştur. Ayrıca 3, 4 ve 8 numaralı problemlerde YBA ile aynı sonuçlar bulunmasına rağmen kalan 7 problemde YBA‘dan daha kötü sonuçlar elde edilmiştir.

Q5S5 tipi problemlerin tümünde GA ve TB ‘den daha başarılı sonuçlar bulunmuştur. Ayrıca 8 numaralı problemde YBA ile aynı sonuç bulunmasına rağmen kalan 9 problemde YBA ‘dan daha kötü sonuçlar elde edilmiştir.

Q5S8 tipi problemlerin tümünde GA ve TB‘den daha başarılı sonuçlar bulunmuştur. Ayrıca 6 numaralı problemde YBA‘dan daha başarılı, 1 ve 4 numaralı problemlerde YBA ile aynı, kalan 7 problemde ise YBA ‘dan daha kötü sonuçlar bulunmuştur.

QH1S2 tipi problemlerden 10 numaralı problemde TB‘den daha iyi sonuç bulunmasına rağmen, kalan 9 problemin hepsinde diğer üç algoritmadan daha kötü sonuçlar elde edilmiştir.

QH1S5 tipi problemlerin tümünde GA ve TB‘den daha başarılı sonuçlar bulunmasına rağmen YBA ‘dan daha başarısız sonuçlar elde edilmiştir.

QH1S8 tipi problemlerin tümünde GA ve TB‘den daha başarılı sonuçlar bulunmuştur. Ayrıca 1 ve 10 numaralı problemlerde YBA‘dan daha başarılı sonuçlar bulunmasına rağmen, kalan 8 problemde YBA‘dan daha kötü sonuçlar elde edilmiştir.

P10, P20 ve P50 tipi problemlerde 2, 5 ve 8 aşamalı türlerin genelinde diğer algoritmalara nispeten yüksek başarıya ulaşılmıştır.

Aynı şekilde Q10, Q20 ve Q50 tipi problemlerde 2, 5 ve 8 aşamalı türlerin genelinde diğer algoritmalara nispeten yüksek başarıya ulaşılmıştır.

P5 ve Q5 tipi problemlerde GA ve TB den başarılı sonuçlara ulaşılmasına rağmen YBA'dan genel olarak başarısız sonuçlara rastlanmaktadır.

PH1 tipi problemlerde daha başarılı sonuçlar bulmamıza rağmen, QH1 tipi problemlerde YBA, GA ve TB daha başarısız sonuçlar bulmaktadır.

Ayrıca genel olarak HAA algoritması diğer algoritmalardan daha düşük CPU zamanı göstermektedir.

Geliştirilen HAA çözümü 300 problemin 142 tanesinde diğer algoritmalardan daha başarılı, 27 tanesinde eşit sonuç ve 131 tanesinde de kötü sonuç bulmuştur. Önerilen yöntem diğer çalışmalara kıyasla en fazla başarılı sonuç veren yöntem olarak Çizelge 4.2' de görülmektedir.

Çizelge 4.2. Algoritmaların başarı oranları.

Yöntem	Problem Adedi	Başarılı Sonuç	Eşit Sonuç	Kötü Sonuç	Başarı oranı %
HA	300	142	27	131	56
YBA	300	135	27	138	54
GA	300	17	-	283	5,6
TB	300	13	-	287	4,3

Çizelge 4.3. Tüm sonuçların gösterimi.

Problem	**BNS**	**HHB**	**İS**	**HHDAO**	**OUI**	**CPU**
P10S2T01	100	100	50	50	0,2393	1,2511
P10S2T02	100	100	500	80	0,0995	1,3099
P10S2T03	5000	1000	500	30	0,1797	1,3074
P10S2T04	100	10	50	30	0,0984	1,2988
P10S2T05	100	100	500	50	0,1704	1,3605
P10S2T06	100	10	500	80	0,1575	1,3089
P10S2T07	100	10	100	80	0,0902	1,4018
P10S2T08	100	10	50	50	0,0917	1,4138
P10S2T09	1000	100	100	80	0,1945	1,2551
P10S2T10	100	10	50	50	0,0997	1,3069
P10S5T01	1000	100	50	50	0,2355	1,6579
P10S5T02	100	10	100	50	0,1000	1,7308
P10S5T03	1000	10	50	80	0,2264	1,6187
P10S5T04	1000	100	50	80	0,0921	1,6786

Çizelge 4.3. Tüm sonuçların gösterimi (Devam).

Problem	BNS	HHB	İS	HHDAO	OUI	CPU
P10S5T05	10000	10	50	50	0,1584	1,6430
P10S5T06	1000	1000	500	50	0,1136	1,7377
P10S5T07	5000	1000	100	50	0,1009	1,6814
P10S5T08	1000	1000	100	30	0,1771	1,6634
P10S5T09	5000	10	500	30	0,1797	1,7536
P10S5T10	10000	100	100	80	0,1793	1,6724
P10S8T01	1000	1000	100	30	0,1900	2,0316
P10S8T02	5000	100	100	50	0,2435	2,2087
P10S8T03	5000	100	100	30	0,1819	2,0291
P10S8T04	10000	1000	100	30	0,1774	2,0221
P10S8T05	100	10	100	30	0,1512	2,1501
P10S8T06	100	10	50	80	0,1066	2,0579
P10S8T07	5000	100	50	80	0,1908	2,2111
P10S8T08	5000	100	500	80	0,1871	1,9663
P10S8T09	5000	100	100	80	0,1241	2,1297
P10S8T10	100	100	500	80	0,1646	2,0456
P20S2T01	5000	1000	50	80	0,1407	2,4275
P20S2T02	10000	100	50	80	0,1518	2,5343
P20S2T03	100	100	50	30	0,0828	2,3001
P20S2T04	10000	10	100	30	0,0914	2,5275
P20S2T05	1000	100	500	30	0,0852	2,4098
P20S2T06	100	10	100	50	0,0706	2,5205
P20S2T07	100	10	50	50	0,0956	2,4475
P20S2T08	5000	100	100	80	0,0948	2,4697
P20S2T09	1000	10	100	30	0,0670	2,5830
P20S2T10	1000	100	500	30	0,0968	2,3321
P20S5T01	100	100	500	50	0,1021	3,4380
P20S5T02	1000	100	50	30	0,0946	3,4853
P20S5T03	5000	10	100	30	0,1033	3,4993
P20S5T04	10000	1000	50	30	0,1159	3,2823
P20S5T05	5000	100	100	30	0,0916	3,5780
P20S5T06	1000	10	500	80	0,0672	3,5521
P20S5T07	5000	100	100	80	0,0909	3,3907
P20S5T08	1000	100	50	30	0,0880	3,5380
P20S5T09	5000	100	500	50	0,0964	3,5225
P20S5T10	1000	1000	100	30	0,0954	3,4891
P20S8T01	10000	100	50	80	0,0931	4,6744
P20S8T02	1000	100	500	50	0,0736	4,7746
P20S8T03	1000	100	500	30	0,0981	4,8885
P20S8T04	10000	1000	500	50	0,1005	4,8118
P20S8T05	1000	10	50	50	0,0956	4,6970
P20S8T06	5000	100	100	30	0,0905	4,6516

Çizelge 4.3. Tüm sonuçların gösterimi (Devam).

Problem	BNS	HHB	İS	HHDAO	OUI	CPU
P20S8T07	1000	10	50	30	0,1217	4,9340
P20S8T08	5000	100	100	80	0,0659	4,4559
P20S8T09	5000	100	50	30	0,0905	4,8928
P20S8T10	1000	1000	500	30	0,1048	4,8927
P50S2T01	10000	1000	500	30	0,0527	8,1115
P50S2T02	1000	10	100	50	0,0357	7,9435
P50S2T03	5000	100	50	30	0,0503	8,2148
P50S2T04	1000	10	100	50	0,0264	8,2480
P50S2T05	1000	100	100	30	0,0516	8,1083
P50S2T06	100	10	500	30	0,0345	8,3936
P50S2T07	5000	1000	500	50	0,0372	8,7204
P50S2T08	100	10	100	80	0,0392	8,0663
P50S2T09	100	100	50	80	0,0368	7,9194
P50S2T10	5000	10	100	50	0,0418	7,9957
P50S5T01	10000	10	500	50	0,0371	13,3402
P50S5T02	10000	100	50	30	0,0298	13,2018
P50S5T03	100	100	500	30	0,0416	13,0267
P50S5T04	1000	10	500	50	0,0522	13,5049
P50S5T05	5000	100	50	50	0,0380	13,3083
P50S5T06	5000	1000	500	50	0,0379	13,5104
P50S5T07	10000	1000	50	50	0,0390	12,8954
P50S5T08	5000	10	500	80	0,0330	13,1838
P50S5T09	5000	100	100	80	0,0366	12,9015
P50S5T10	100	100	50	50	0,0491	15,1987
P50S8T01	10000	10	50	30	0,0345	17,9678
P50S8T02	1000	1000	100	50	0,0413	19,2852
P50S8T03	1000	100	100	80	0,0485	18,5095
P50S8T04	5000	10	50	80	0,0340	19,4084
P50S8T05	5000	1000	500	50	0,0462	17,8579
P50S8T06	1000	1000	500	80	0,0395	18,7635
P50S8T07	1000	1000	500	50	0,0327	20,2240
P50S8T08	10000	100	50	30	0,0199	18,5968
P50S8T09	5000	1000	500	50	0,0221	18,5425
P50S8T10	1000	1000	50	50	0,0364	17,4357
P5S2T01	100	10	50	30	0,1991	0,8969
P5S2T02	100	100	100	30	0,3339	0,9475
P5S2T03	100	10	50	50	0,2000	0,9358
P5S2T04	100	10	100	80	0,3654	0,9980
P5S2T05	100	100	500	80	0,2779	0,9224
P5S2T06	100	10	50	50	0,3583	0,8979
P5S2T07	100	10	50	30	0,1662	0,9216
P5S2T08	100	10	50	30	0,1981	0,8914

Çizelge 4.3. Tüm sonuçların gösterimi (Devam).

Problem	BNS	HHB	İS	HHDAO	OUI	CPU
P5S2T09	100	10	50	50	0,2453	0,9640
P5S2T10	100	10	50	30	0,3547	0,9349
P5S5T01	100	10	50	50	0,2554	1,1332
P5S5T02	100	10	50	30	0,2314	1,0261
P5S5T03	100	10	50	50	0,2236	1,0627
P5S5T04	100	10	50	30	0,2216	1,0862
P5S5T05	100	10	100	80	0,2118	1,0965
P5S5T06	100	10	50	50	0,2677	1,1105
P5S5T07	5000	1000	50	80	0,2056	1,1090
P5S5T08	100	100	100	50	0,3352	1,0660
P5S5T09	100	10	50	50	0,1932	1,1005
P5S5T10	100	10	50	50	0,2159	1,1831
P5S8T01	100	10	50	50	0,2179	1,1501
P5S8T02	100	10	50	50	0,3737	1,3048
P5S8T03	100	10	50	50	0,2284	1,2199
P5S8T04	100	10	50	30	0,3338	1,2768
P5S8T05	100	10	50	50	0,4529	1,4249
P5S8T06	1000	100	100	80	0,3151	1,3011
P5S8T07	100	10	100	30	0,4723	1,2965
P5S8T08	100	10	50	50	0,2851	1,1645
P5S8T09	100	10	100	30	0,3214	1,2693
P5S8T10	100	100	50	50	0,3225	1,2360
PH1S2T01	10000	1000	500	80	0,0189	26,1599
PH1S2T02	5000	100	500	30	0,0179	23,5638
PH1S2T03	100	10	50	30	0,0240	24,1873
PH1S2T04	1000	1000	500	30	0,0158	23,5922
PH1S2T05	1000	100	100	50	0,0184	23,9191
PH1S2T06	1000	100	50	30	0,0152	23,9633
PH1S2T07	1000	100	500	50	0,0255	26,4242
PH1S2T08	1000	1000	500	30	0,0198	23,4390
PH1S2T09	1000	100	500	50	0,0260	23,9620
PH1S2T10	1000	10	500	80	0,0335	24,4601
PH1S5T01	100	100	100	80	0,0194	42,4223
PH1S5T02	10000	1000	500	50	0,0226	42,9017
PH1S5T03	5000	1000	50	30	0,0231	50,0158
PH1S5T04	1000	1000	100	80	0,0170	47,7975
PH1S5T05	1000	10	50	30	0,0208	42,8577
PH1S5T06	1000	10	50	30	0,0179	42,6732
PH1S5T07	10000	10	50	80	0,0153	44,1845
PH1S5T08	5000	10	100	80	0,0182	44,0444
PH1S5T09	1000	1000	100	50	0,0196	42,5337
PH1S5T10	10000	1000	100	50	0.0187	45,6064

Çizelge 4.3. Tüm sonuçların gösterimi (Devam).

Problem	BNS	HHB	İS	HHDAO	OUI	CPU
PH1S8T01	10000	100	500	80	0,0241	65,7716
PH1S8T02	1000	100	50	80	0,0170	66,7667
PH1S8T03	5000	10	100	30	0,0198	60,1391
PH1S8T04	1000	10	500	80	0,0164	60,8451
PH1S8T05	5000	1000	50	80	0,0185	62,8389
PH1S8T06	5000	10	100	50	0,0122	66,8408
PH1S8T07	10000	100	50	80	0,0206	66,0380
PH1S8T08	100	10	50	50	0,0233	69,8777
PH1S8T09	1000	1000	100	30	0,0174	68,4157
PH1S8T10	1000	1000	500	80	0,0199	61,0338
Q10S2T01	5000	1000	500	50	0,1718	1,7624
Q10S2T02	5000	100	50	80	0,1971	1,4131
Q10S2T03	10000	10	500	30	0,1363	1,4047
Q10S2T04	100	100	100	30	0,1695	1,4568
Q10S2T05	5000	10	50	80	0,1781	1,4464
Q10S2T06	1000	1000	500	80	0,2133	1,4161
Q10S2T07	100	10	500	30	0,2103	1,4468
Q10S2T08	100	10	500	30	0,1797	1,3992
Q10S2T09	5000	100	50	30	0,1729	1,4682
Q10S2T10	1000	1000	500	50	0,2642	1,5409
Q10S5T01	1000	10	500	80	0,1998	1,9696
Q10S5T02	1000	1000	100	50	0,1899	1,9576
Q10S5T03	5000	10	500	50	0,1893	2,0136
Q10S5T04	1000	10	500	30	0,1853	1,9439
Q10S5T05	5000	10	50	50	0,1914	1,9906
Q10S5T06	1000	1000	500	80	0,1803	1,9533
Q10S5T07	1000	1000	500	80	0,2054	1,9695
Q10S5T08	1000	100	100	50	0,2186	1,9647
Q10S5T09	5000	100	500	30	0,2303	1,9551
Q10S5T10	100	100	500	30	0,2718	1,9500
Q10S8T01	10000	100	500	50	0,1855	2,5018
Q10S8T02	1000	100	500	30	0,1945	2,5079
Q10S8T03	1000	10	50	80	0,1719	2,5278
Q10S8T04	10000	1000	100	30	0,1725	2,5889
Q10S8T05	5000	1000	50	30	0,1658	2,5612
Q10S8T06	100	100	50	80	0,2152	2,5068
Q10S8T07	1000	100	50	50	0,1971	2,4904
Q10S8T08	5000	1000	50	30	0,1586	2,4988
Q10S8T09	10000	100	50	80	0,1650	2,5068
Q10S8T10	5000	100	100	80	0,1917	2,4775
Q20S2T01	10000	100	100	50	0,1367	3,9712
Q20S2T02	10000	1000	100	80	0,0882	3,3716

Çizelge 4.3. Tüm sonuçların gösterimi (Devam).

Problem	BNS	HHB	İS	HHDAO	OUI	CPU
Q20S2T03	5000	1000	50	80	0,1122	2,7921
Q20S2T04	100	10	100	80	0,1248	2,7811
Q20S2T05	5000	1000	100	30	0,1303	2,8045
Q20S2T06	1000	100	500	30	0,1278	2,7970
Q20S2T07	5000	100	100	50	0,1299	2,7715
Q20S2T08	5000	1000	500	30	0,1301	2,7541
Q20S2T09	1000	10	500	30	0,1317	2,7643
Q20S2T10	5000	100	500	80	0,1160	2,7523
Q20S5T01	100	10	50	30	0,0852	4,1970
Q20S5T02	10000	10	100	30	0,1153	4,2093
Q20S5T03	1000	100	100	80	0,0740	4,1739
Q20S5T04	100	10	50	50	0,1183	4,1970
Q20S5T05	5000	100	50	80	0,0955	4,1998
Q20S5T06	1000	10	100	50	0,0885	4,1910
Q20S5T07	1000	100	50	50	0,0953	4,1925
Q20S5T08	1000	1000	50	50	0,0948	4,2150
Q20S5T09	1000	100	50	50	0,0955	5,4749
Q20S5T10	10000	100	50	80	0,1002	4,5411
Q20S8T01	100	10	50	80	0,1307	5,5818
Q20S8T02	1000	1000	50	80	0,0990	5,6095
Q20S8T03	1000	1000	100	80	0,1120	5,6151
Q20S8T04	1000	10	100	80	0,1172	5,5997
Q20S8T05	10000	1000	500	80	0,0952	5,6636
Q20S8T06	1000	1000	500	50	0,0810	7,1458
Q20S8T07	10000	100	50	80	0,1177	7,0110
Q20S8T08	10000	10	500	80	0,1089	5,8459
Q20S8T09	1000	100	50	80	0,1313	5,6300
Q20S8T10	5000	1000	500	80	0,0901	7,3037
Q50S2T01	10000	1000	500	30	0,0387	8,8454
Q50S2T02	10000	1000	500	50	0,0609	8,8540
Q50S2T03	1000	10	50	50	0,0341	10,101
Q50S2T04	10000	10	500	50	0,0531	8,8458
Q50S2T05	1000	1000	100	80	0,0434	8,8451
Q50S2T06	5000	100	50	50	0,0770	8,9034
Q50S2T07	10000	10	500	30	0,0544	12,1427
Q50S2T08	5000	10	50	50	0,0477	8,8382
Q50S2T09	100	100	500	80	0,0499	8,8637
Q50S2T10	10000	100	100	30	0,0510	8,8627
Q50S5T01	5000	10	50	30	0,0293	16,6793
Q50S5T02	100	10	50	30	0,0412	16,4591
Q50S5T03	1000	10	500	80	0,0282	18,1027
Q50S5T04	1000	1000	50	80	0,0322	17,3486

Çizelge 4.3. Tüm sonuçların gösterimi (Devam).

Problem	BNS	HHB	İS	HHDAO	OUI	CPU
Q50S5T05	1000	100	50	80	0,0371	14,8517
Q50S5T06	10000	100	100	30	0,0389	14,8786
Q50S5T07	100	100	50	80	0,0186	15,0802
Q50S5T08	100	100	100	80	0,0476	16,8495
Q50S5T09	5000	100	50	50	0,0418	18,4718
Q50S5T10	10000	10	50	50	0,0374	14,9900
Q50S8T01	5000	10	50	50	0,0444	22,6281
Q50S8T02	10000	100	100	80	0,0347	20,8624
Q50S8T03	1000	10	100	30	0,0370	20,8939
Q50S8T04	10000	1000	500	80	0,0634	22,6012
Q50S8T05	10000	1000	500	30	0,0360	22,4896
Q50S8T06	100	10	100	30	0,0497	24,0652
Q50S8T07	5000	1000	50	30	0,0383	20,9074
Q50S8T08	1000	10	50	30	0,0391	22,5404
Q50S8T09	100	10	50	50	0,0385	22,7378
Q50S8T10	1000	1000	500	50	0,0342	23,8007
Q5S2T01	100	10	50	30	0,3443	1,0121
Q5S2T02	100	10	50	50	0,3897	0,9975
Q5S2T03	100	10	50	30	0,2144	1,0409
Q5S2T04	100	10	50	50	0,4299	0,9982
Q5S2T05	100	10	50	30	0,3797	1,0060
Q5S2T06	100	10	50	50	0,4472	0,9985
Q5S2T07	1000	100	500	30	0,2875	0,9922
Q5S2T08	100	10	50	30	0,2281	0,9934
Q5S2T09	100	10	100	50	0,5945	0,9973
Q5S2T10	100	10	50	50	0,4073	1,0025
Q5S5T01	100	10	50	50	0,5174	1,2460
Q5S5T02	100	10	50	50	0,3181	1,2354
Q5S5T03	100	10	500	80	0,3672	1,2234
Q5S5T04	100	10	50	50	0,3316	1,2345
Q5S5T05	100	10	500	30	0,2964	1,2286
Q5S5T06	100	10	50	30	0,4358	1,2333
Q5S5T07	100	10	50	30	0,2703	1,2450
Q5S5T08	100	10	50	80	0,2500	1,2342
Q5S5T09	100	10	100	50	0,2576	1,2385
Q5S5T10	100	10	50	30	0,2957	1,2304
Q5S8T01	100	100	100	50	0,3079	1,4777
Q5S8T02	100	10	50	30	0,2380	1,4743
Q5S8T03	100	10	50	80	0,2476	1,4586
Q5S8T04	100	10	50	30	0,2000	1,4505
Q5S8T05	100	10	50	50	0,3321	1,4594
Q5S8T06	100	100	50	80	0,2170	1,4606

Çizelge 4.3. Tüm sonuçların gösterimi (Devam).

Problem	BNS	HHB	İS	HHDAO	OUI	CPU
Q5S8T07	100	10	50	50	0,2658	1,4730
Q5S8T08	100	10	50	50	0,3831	1,4656
Q5S8T09	100	10	50	30	0,2951	1,7591
Q5S8T10	100	10	50	50	0,3656	2,4736
QH1S2T01	100	10	500	30	0,0216	25,8810
QH1S2T02	5000	1000	100	80	0,0293	25,328
QH1S2T03	10000	1000	500	80	0,0300	25,8673
QH1S2T04	5000	100	100	80	0,0223	25,9334
QH1S2T05	10000	100	50	30	0,0362	27,5827
QH1S2T06	1000	10	50	80	0,0219	25,8033
QH1S2T07	10000	100	50	50	0,0216	29,2495
QH1S2T08	100	10	100	80	0,0151	25,8497
QH1S2T09	5000	100	50	50	0,0260	27,3392
QH1S2T10	1000	10	100	30	0,0249	25,9096
QH1S5T01	100	100	100	30	0,0207	49,3934
QH1S5T02	1000	100	500	30	0,0194	45,9774
QH1S5T03	10000	10	500	80	0,0212	51,0058
QH1S5T04	1000	100	500	30	0,0144	47,7177
QH1S5T05	10000	10	50	30	0,0199	47,7270
QH1S5T06	10000	100	100	80	0,0221	45,9603
QH1S5T07	1000	10	500	50	0,0179	47,6356
QH1S5T08	100	10	500	30	0,0190	49,3656
QH1S5T09	1000	10	100	80	0,0137	49,3103
QH1S5T10	1000	10	500	30	0,0271	45,9417
QH1S8T01	10000	100	500	30	0,0199	71,1067
QH1S8T02	5000	100	50	30	0,0258	67,8431
QH1S8T03	100	10	100	30	0,0262	69,1972
QH1S8T04	100	10	50	30	0,0188	69,2168
QH1S8T05	100	10	100	80	0,0213	71,0641
QH1S8T06	10000	100	50	30	0,0206	67,7810
QH1S8T07	100	10	100	80	0,0223	67,7848
QH1S8T08	1000	1000	100	80	0,0155	69,5026
QH1S8T09	10000	1000	50	30	0,0192	69,4265
QH1S8T10	1000	1000	100	80	0,0112	67,8579

5. SONUÇLAR VE ÖNERİLER

5.1 Sonuçlar

Paralel çok işlemcili makine çizelgeleme, NP-zor olarak bilinen problem sınıfına girmekte ve optimum çözümü çok zor bulunan veya sadece gerçek çözüme yaklaşık çözümleri olan problem türüdür. Araştırmada; harmoni arama algoritması ile bulanık paralel makine çizelgeleme problemlerinin çözümleri gerçekleştirilmiştir.

Araştırmada, bulanık paralel makine çizelgeleme problemlerinde harmoni arama algoritması ile bulunan sonuçlar literatürdeki çalışmalarla kıyaslanmıştır. Literatürde bulunan Oğuz (2005) problemlerini bulanıklaştıran ve çözümünü yapan Gözen (2007)'in ve Yılmaz (2008)'ın bulduğu sonuçlar ile kıyaslamalar yapılmıştır. Harmoni arama algoritması çözümleri için C# programlama dilinde bilgisayar yazılımı yapılmıştır.

Harmoni arama algoritması çözümleri için 1000 başlangıç nota sayısı, 100 harmoni hafızası boyutu, 100 iterasyon ve %80 HHDAO alınmıştır. Ancak bazı problemler için HHDAO'nun %30 olduğu değer ile çözüm yapılması tavsiye edilebilir.

Çalışma sonucunda geliştirilen HAA, Gözen (2007)'in GA' sı ile kıyaslandığında 283 başarılı ve 17 başarısız sonuç olmak üzere %94 başarı göstermiştir. Yine aynı şekilde Gözen (2007)'in TB 'si ile kıyaslandığında 287 başarılı ve 13 başarısız sonuç olmak üzere %95 başarı göstermiştir. Son olarak HAA, Yılmaz (2008)'ın YBA ile kıyaslandığında 142 başarılı, 27 aynı ve 131 başarısız sonuç ile %56 başarı göstermiştir.

Yapılan çalışma sonucunda Harmoni arama algoritmasının, bulanık paralel makine problemlerinin çözümünde etkili ve rekabetçi bir yöntem olduğu gösterilmiştir.

5.2 Öneriler

Bundan sonraki çalışmalarda bulanık paralel makine çizelgeleme problemlerinin diğer meta sezgisel yöntemlerle çözülerek optimum/optimuma yakın sonuçlar bulmak denenebilir. Ya da hibrid algoritmalar kullanılarak bu sonuçlar geliştirilmeye çalışılabilir.

KAYNAKLAR

Ahmad M.İ., 2004, Gömülü Sistemlerle Bulanık Mantık, Bileşim Yayınevi.

Akyol D., 2004, Application of neural networks to heuristic scheduling algorithms, *Computer&Industrial Engineering (46),* 679-696.

Akkoyunlu M. C. ve Engin O., 2011, Kesikli Harmoni Arama Algoritması ile Optimizasyon Problemlerinini Çözümü: Literatür Araştırması, *S.Ü. Müh.-Mim. Fak. Derg.*, c.26, s.4.

Alataş, B., 2010, Chaotic harmony search algorithms, *Applied Mathematics and Computation (216),* 2687–2699.

Allahverdi, A., Ng, C.T., Cheng, T.C.E., Kovalyov, M.Y., 2008, A survey of scheduling problems with setup time sor costs, *European Journal of Operation Research 187(3), 985*-1032.

Anglani, A., Grieco, A., Guerriero, E., Musmanno, R., 2005, Robust scheduling of parallel machines with sequence-dependent set-up costs, *European Journal of Operational Research (161),* 704-720.

Ayvaz, M. T., 2007, Simultaneous determination of aquifer parameters and zone structures with fuzzy c-means clustering and meta-heuristic harmony search algorithm, *Advances in Water Resources (30)*, 2326–2338

Ayvaz, M. T., 2009, Application of Harmony Search algorithm to the solution of groundwater management models, *Advances in Water Resources (32)*), 916–924.

Brar, Y., Dhillon, S., Kothari, P., 2005, Fuzzy satisfying multi-objective generation scheduling based on simplex weightage pattern search, *Electrical Power and Energy Systems, (27*), 518-527.

Ceylan, H., Haldenbilen, S., ve Başkan, Ö., 2008, Transport energy modeling with meta-heuristic harmony search algorithm,an application to Turkey, *Energy Policy (36)*, 2527– 2535.

Chanas, S., Kasperski, A., 2001, Minimizing maximum lateness in a single machine scheduling problem with fuzzy processing times and fuzzy due dates, *Engineering Applications of Artificial Intelligence (14)*, 377-386.

Choi, S.H., Wang, K., 2012, Flexible flowshop scheduling with stochastic processing times: A decomposition based approach, *Computers & Industrial Engineering 63(2),* 362-373.

Coelho, L. S., ve Bernert, D. L. A., 2009, An improved harmony search algorithm for synchronization of discrete-time chaotic systems, *Chaos, Solutions and Fractals (41)*, 2526–2532

Cura, T., 2008, Modern Sezgisel Teknikler ve Uygulamaları, *Papatya Yayınları*, İstanbul, 14-15.

Döyen, A., 2004, Akış Tipi Çizelgeleme Problemlerinin Yapay Bağışıklık Sistemleri İle Çözümü ve Parametre Optimizasyonu, Yüksek Lisans Tezi, Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Endüstri Mühendisliği Anabilim Dalı.

Elmas Ç., 2003, Bulanık Mantık Denetleyiciler, Seçkin Yayınevi.

Engin, O., Ceran, G., Yılmaz, M.K., 2011, An Efficient Genetic Algorithm for Hybrid Flowshop Scheduling with Multi-processor Task Problems, *Applied Soft Computing 11(3)*, 3056-3065.

Engin, O., Döyen, A., 2004, A New Approach To Solve Hybrid Flow Shop Scheduling Problems By Artificial Immune System, *Future Generation Computer Systems, (20),* 1083-1095.

Etaner, Ş., 1993, Bulanık Kontrol, *Otomasyon Dergisi*, Şubat, 107-112.

Fesanghary, M., Mahdavi, M., Jolandan, M. M., ve Alizadeh, Y., 2008, Hybridizing harmony search algorithm with sequential quadratic programming for engineering optimization problems, *Computer Methods Appl. Mech. Engrg. (197)*, 3080–3091

Gao, K. Z., Pan, Q. K. ve Li, J. Q., 2011, Discrete harmony search algorithm for the no-wait flow shop scheduling problem with total flow time criterion, *The International Journal of Advanced Manufacturing Technology*, DOI 10.1007/s00170-011-3197-6.

Geem, Z.W., 2012, Effects of initial memory and identical harmony in global optimization using harmony search algorithm, *Applied Mathematics and Computation 218(22)*, 11337-11343.

Geem, Z. W., Kim, J. H., ve Loganathan, G. V., 2001, A new heuristic optimization algorithm: harmony search, *Simulations (76)*, 60-68.

Geem, Z. W., 2008, Novel derivative of harmony search algorithm for discrete design variables, *Applied Mathematics and Computation (199)*, 223–230.

Geem, Z. W., ve Sim, K. B., 2010, Parameter-setting-free harmony search algorithm, *Applied Mathematics and Computation (217)*, 3881–3889.

Gonzales, I., Vela, C., Puente, J., 2005, An Evolutinary Approach to Designing and Solving Fuzzy Job Shop Problems, *Artificial Intelligence and Knowledge Engineering Applications*, 3562, 74-83.

Gözen, Ş., 2007. Bulanık Esnek Akış Tipi Çok Prosesli Çizelgeleme Problemlerinin Genetik Algoritma Ve Tavlama Benzetimi İle Çözümü, Yüksek Lisans Tezi, Fen Bilimleri Enstitüsü, Selçuk Üniversitesi.

Gupta, J.N.D., 1988, Two-Stage Hybrid Flowshop Scheduling Problem, *Operational Research Society, (39),* 359-364.

Han S., Ishii H., Fuji S., 1994, One machine scheduling problem with fuzzy duedates, European Journal of Operational Research *(79)*, 1–12.

Han, Y.Y., Pan, Q.K., Liang, J.J. ve Li, J., 2010, A Hybrid Discrete Harmony Search Algorithm for blocking flow shop Scheduling, *Bio-Inspired Computing: Theories and Applications (BIC-TA), 2010 IEEE Fifth International Conference,* Changsha, 435-438.

Hapke, M., Slowinski, R., 2000, Fuzzy set approach to multiobjective and multi-mode project scheduling under uncertainty In: Slowinski, R., Hapke, M. (Eds.), Scheduling Under Fuzziness. Physica-Verlag, Heidelberg, pp. 197–221.

Herroelen, W., Leus, R., 2005, Project scheduling under uncertainty: Survey and research potentials, *European Journal of Operational Research (165),* 289-306.

Ishibuchi H., Yamamoto N., Murata T., Tanaka H., 1994, Genetic algorithms and neighborhood search algorithms for fuzzy flow shop scheduling problems, Fuzzy Sets and Systems (67), 81–100.

Jaberipour, M., ve Khorram, E., 2010, Two improved harmony search algorithms for solving engineering optimization problems, *Communication in Nonlinear Science and Numerical Simulation (15),* 3316–3331.

Jungwattanakit, J., Reodecha, M., Chaovalitwongse, P., Werner, F., 2009, A comparison of scheduling algorithms for flexible flowshop problems with unrelated paralel machines, setup times, and dual criteria, *Computers & Operation Research 36(2*), 358-378.

Kasperski, A., 2005, A possibilistic approach to sequencing problems with fuzzy Parameters, *Fuzzy Sets and Systems*, 77-86.

Kaynak, O., ve Armağan G., 1992, Bulanık Mantık, Otomasyon Dergisi, 74-82.

Khademi, H.Z., Fakhrzad, M.B., 2011, Solving flexible flowshop problem with a hybrid genetic algorithm and data mining: A fuzzy approach, *Expert Systems with Applications 38(6),* 7609-7615.

Kianfar, K., Ghomi, S.M.T.F., Jadid, A.O., 2012, Study of stochastic sequence dependent flexible flowshop via developing a dispatching rule and a hybrid GA, *Engineering Applications of Artificial Intelligence 25(3),* 494-506.

Kim, K.,Yun, Y., Yoon, J., 2005, Hybrid genetic algorithm with adaptive abilities for resource-constrained multiple project scheduling, *Computers in Industry (56),* 143-160.

Klir, G., Clair, U., Yuan, B., 1997, Fuzzy Set Theory Foundations and Applications, Prentice Hall.

Kosko, B., 1993, Fuzzy Thinking: The New Science of Fuzzy Logic, Library of Congress Catalogging-in-Publication Data, 58-62.

Kuroda, M., Wang, Z., 1996, Fuzzy job shop scheduling, *International Journal of Production Economics*, 45–51.

Lee, K. S., ve Geem, Z. W., 2004, A new structural optimization method based on the harmony search algorithm, *Computers and Structures (82)*, 781–798

Lee, K. S., ve Geem, Z. W., 2005, A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice, *Comput. Methods Appl. Mech. Engrg.(194)*, 3902–3933.

Li, D., Wu, C., Tsai, T., Cang F.M., 2004, Using mega-fuzzification and data trend estimation in small data set learning for early FMS scheduling knowledge, Computers& Operations Research.

Litoiu, M., Tadei, R., 2001, Fuzzy scheduling with application to real-time systems, *Fuzzy Sets and Systems (121)*, 523-535.

Logendran, R., DeSzoeke, P., Barnard, F., 2006, Sequence dependent group problems in flexible flowshops, *International Journal of Production Economics 102(1*), 66-86.

Mahdavi, M., Fesanghary, M., ve Damangir, E., 2007, An improved harmony search algorithm for solving optimization problems, *Applied Mathematics and Computation (188)*, 1567–1579.

Majozi , T., Zhu (Frank), X.X., 2005, A combined fuzzy set theory and MILP approach in integration of planning and scheduling of batch plants—Personel evaluation and allocation, *Computers and Chemical Engineering (29)*, 2029-2047.

Nawaz, M., Enscore, E. E. J.,ve Ham, I., 1983, A heuristic algorithm for them-machine, n-job flow shop sequencing problem, *International Journal of Management Science (11)*, 91 -95.

Oğuz, C., 2005, Kıyaslama Problemleri, http://home.ku.edu.tr/~coguz/Research/ dataset2.zip

Oğuz, C., Ercan, F., 2005, A Genetic Algorithm for Hybrid Flow Shop Scheduling with Multiprocessor Tasks, *Journal of Scheduling,(8)*, 323-351.

Omran, M. G. H., ve Mahdavi, M., 2008, Global-best harmony search, *Applied Mathematics and Computation (198)*, 643–656.

Özkan, O., 1997, Sıcaklı ve Nemin Bulanık Mantık Yöntemiyle Kontrolü, Yüksek Lisans Tezi, Fen Bilimleri Enstitüsü, Selçuk Üniversitesi.

Öztürk, R.E., 2007, Esnek Akış Tipi Ve Çok İşlemcili Esnek Akış Tipi Çizelgeleme Problemlerinin Paralel Doyumsuz Algoritma İle Çözümü, Yüksek Lisans Tezi, Fen Bilimleri Enstitüsü, Selçuk Üniversitesi.

Özyön, S., Yaşar, C., ve Temurtaş, H., 2011, Harmoni Arama Algoritmasının Çevresel Ekonomik Güç Dağıtım Problemlerine Uygulanması, *Çukurova Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi 26(2),* 65-76.

Pan, Q. K., Suganthan, P. N., Taşgetiren, M. F., ve Liang, J.J., (2010a), A self-adaptive global best harmony search algorithm for continuous optimization problems, *Applied Mathematics and Computation (216)*, 830–848

Pan, Q. K., Duan, J. H., Liang, J. J., Gao, K. Ve Li J., (2010b), A Novel Discrete Harmony Search Algorithm for Scheduling Lot-streaming Flow Shops, *Control and Decision Conference (CCDC)*, 2010 Chinese, Xuzhou, 1531-1536

Pan, Q. K., Suganthan, P. N., Liang, J.J., ve Taşgetiren, M. F., 2011, A local-best harmony search algorithm with dynamic sub-harmony memories for lot-streaming flow shop scheduling problem, *Expert Systems with Applications (38)*, 3252–3259.

Peng, J., Liu, B., 2004, Parallel machine scheduling models with fuzzy processing times, *Information Sciences (166),* 49-66.

Rao, S. S., 1996, "Engineering Optimization: Theory and Practice", Third Edition, *John Wiley & Son*, New York.

Ruiz, R., Rodriguez, J.A.V., 2010, The hybrid flowshop scheduling problem, *European Journal of Operation Research 205(1),* 1-18.

Sağıroğlu, Ş., Beşdok, E., Erler, M., 2003, Mühendislikte Yapay Zeka Uygulamaları-I: Yapay Sinir Ağları", Ufuk Kitap Kırtasiye-Yayıncılık Tic. Ltd. Şti., ISBN: 975-95948-5-4, Kayseri.

Saka, M., P., 2009, Optimum design of steel sway frames to BS5950 using harmony search algorithm, *Journal of Constructional Steel Research (65)*, 36–43.

Shiau, D.F., Cheng, S.C., Huang, Y.M., 2008, Proportionate flexible flowshop scheduling via a hybrid constructive genetic algorithm, *Expert Systems with Applications 34(2),* 1133-1143.

Temiz, İ., Erol, S., 2004, " n /m/ F /Q_1 C_{maks} +Q_2 T_{maks} Problemi İçin Bulanık Genetik Algoritma, Yöneylem Araştırması/ Endüstri Mühendisliği-XXIV Ulusal Kongresi, 15-18 Haziran Gaziantep-Adana.

Wang, J.R., 1999. A fuzzy set approach to activity scheduling for product development. *Journal of the Operational Research Society (50)*, 1217–1228.

Wang, J., 2004. A fuzzy robust scheduling approach for product development projects, *European Journal of Operational Research (152)*, 180–194.

Wang, L., Pan, Q. K., ve Taşgetiren M. F., 2010, Minimizing the total flow time in a flow shop with blocking by using hybrid harmony search algorithms, *Expert Systems with Applications (37)*, 7929–7936

Wang, L., Pan, Q. K., ve Taşgetiren., M. F., 2011, A hybrid harmony search algorithm for the blocking permutation flow shop scheduling problem, *Computers & Industrial Engineering (61)*, 76–83.

Yao, J., Lin, F.T., 2002, Constructing a fuzzy flow-shop sequencing model based on statistical data, *International Journal of Approximate Reasoning (29)*, 215-234.

Yılmaz, M.K., 2008. Bulanık Çok İşlemcili Esnek Akış TipimÇizelgeleme Problemlerinin Çözümünde Yeni bir Yapay Bağışıklık Algoritması Yaklaşımı, Yüksek Lisans Tezi, Fen Bilimleri Enstitüsü, Selçuk Üniversitesi.

Yuen, D.W., Wang, C., Wang, D., 2002, The single machine ready time scheduling problem with fuzzy processing times, *Fuzzy Sets and Systems (127)*, 117-129.

Zadeh, L.A.,1984, Making computers think like people, IEEE. Spectrum (59), 26.

Zou, D., Gao, L., Wu, J., Li, S., ve Li Y., (2010a), A novel global harmony search algorithm for reliability problems, *Computers & Industrial Engineering (58)*, 307–316

Zou, D., Gao, L., Wu, J., Li, S., Wu, J., ve Wang, X., (2010b), A novel global harmony search algorithm for task assignment problem, *The Journal of Systems and Software (83)*, 1678–1688

Zou, D., Gao, l., Li, S., ve Wu, J., (2011a), Solving 0–1 knapsack problem by a novel global harmony search algorithm, *Applied Soft Computing (11)*, 1556–1564

Zou, D., Gao, L., Li, S., ve Wu, J., (2011b), An effective global harmony search algorithm for reliability problems, *Expert Systems with Applications (38)*, 4642–4648.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı	**:**	Mehmet Cabir Akkoyunlu
Uyruğu	**:**	Türkiye Cumhuriyeti
Doğum Yeri ve Tarihi	**:**	Ermenek 07.01.1988
Telefon	**:**	0 505 844 58 51
Faks	**:**	-
e-mail	**:**	mehmetcabira@hotmail.com

EĞİTİM

Derece		**Adı, İlçe, İl**	**Bitirme Yılı**
Lise	:	Bayburt Anadolu Lisesi, Merkez, Bayburt	2005
Üniversite	:	Selçuk Üniversitesi, Selçuklu, Konya	2009
Yüksek Lisans	:	Selçuk Üniversitesi, Selçuklu, Konya	2013
Doktora	:		

İŞ DENEYİMLERİ

Yıl	**Kurum**	**Görevi**
2010-	Bartın Üniversitesi	Araştırma Görevlisi
2011-2013	Selçuk Üniversitesi	Araştırma Görevlisi

UZMANLIK ALANI

İş çizelgeleme, Kalite kontrol, Optimizasyon.

YABANCI DİLLER

İngilizce

YAYINLAR

- Ahmet SARUCAN, Mehmet Cabir AKKOYUNLU, Aydoğan BAŞ, 2010"Analitik hiyerarşi proses yöntemi ile rüzgar türbin seçimi", S.Ü. Müh.-Mim. Fak. Derg., 25.

- Mehmet Cabir AKKOYUNLU, Orhan ENGİN, 2011, Kesikli Harmoni Arama Algoritması ile Optimizasyon Problemlerinin Çözümü: Literatür Araştırması, S.Ü. Müh.-Mim. Fak. Derg., 26(4), 140-148. **(Yüksek Lisans Tezinden Yapılmıştır)**