T.C.

SELÇUK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

ESNEK AKIŞ TİPİ VE ÇOK İŞLEMCİLİ ESNEK AKIŞ TİPİ ÇİZELGELEME PROBLEMLERİNİN PARALEL DOYUMSUZ ALGORİTMA İLE ÇÖZÜMÜ

ENDÜSTRİ MÜHENDİSLİĞİ ANA BİLİM DALI

YÜKSEK LİSANS TEZİ

Reşide Elif ÖZTÜRK 038230002001

T.C.

SELÇUK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

ESNEK AKIŞ TİPİ VE ÇOK İŞLEMCİLİ ESNEK AKIŞ TİPİ ÇİZELGELEME PROBLEMLERİNİN PARALEL DOYUMSUZ ALGORİTMA İLE ÇÖZÜMÜ

Reşide Elif ÖZTÜRK 038230002001

YÜKSEK LİSANS TEZİ

ENDÜSTRİ MÜHENDİSLİĞİ ANA BİLİM DALI

Bu Tez 28/03/2007 Tarihinde Aşağıdaki Jüri Tarafından Oybirliği / Oy Çokluğuyla Kabul Edilmiştir.

Yrd. Doç. Dr. Orhan ENGİN	Prof. Dr. Ahmet PEKER	Yrd. Doç. Dr. M. Emin Baysal
(Danışman)	(Üye)	(Üye)

ÖZET

Yüksek Lisans Tezi

ESNEK AKIŞ TİPİ VE ÇOK İŞLEMCİLİ ESNEK AKIŞ TİPİ ÇİZELGELEME PROBLEMLERİNİN PARALEL DOYUMSUZ ALGORİTMA İLE ÇÖZÜMÜ

Reşide Elif ÖZTÜRK

Selçuk Üniversitesi Fen Bilimleri Enstitüsü

Endüstri Mühendisliği Ana Bilim Dalı

Danışman: Yrd. Doç.Dr. Orhan ENGİN

2007, 83 Sayfa

Jüri: Prof.Dr. Ahmet PEKER

Yrd. Doç.Dr. Orhan ENGİN

Yrd. Doç.Dr. M. Emin Baysal

Esnek akış tipi sistemi; hem akış tipi hem de paralel makine sistemlerinin bazı ögelerinin birleşiminden oluşan özel bir yapıya sahiptir. Esnek Akış Tipi Çizelgeleme (EATÇ) ve Çok İşlemcili Esnek Akış Tipi Çizelgeleme (ÇİEATÇ) Polinomiyel olmayan (NP) – Zor problemler olarak bilinir. Bu çalışmada, EATÇ ve ÇİEATÇ problemlerini çözmek için etkin bir Paralel Doyumsuz Algoritma (PDA) önerildi. PDA, sırasıyla yıkım ve inşa olarak iki aşamada uygulanır. Çalışmada, kontrol parametreleri olarak başlangıç popülasyonu, alt grup sayısı, iterasyon sayısı, doyum oranı ve inşa yöntemleri kullanılmıştır. Literatürdeki Carlier ve Neron'un (2000) EATÇ Problemleri ile Oğuz ve Ercan'ın (2005) ÇİEATÇ problemleri, PDA ile elde edilen değerler, literatürde çözümlenmiş Döyen (2004)'in önermiş olduğu Yapay Bağışıklık Sistemi (YBS), Neron (2001)'un Dal Sınır Algoritması (DSA), Oğuz (2005)'un Genetik Algoritma (GA) ve Ceran (2006)'ın GA metodları ile elde edilen sonuçlar ile kıyaslanmıştır. Bu kıyaslamalar, önerilen PDA yaklaşımının performansının YBS ve Ceran (2006)'ın GA yaklaşımlarının performansları ile aynı olduğunu; DSA ve Oğuz (2005)'un GA yaklaşımlarının performanslarından daha iyi olduğunu göstermektedir.

Anahtar Kelimeler: Esnek Akış Tipi Çizelgeleme Problemleri, Esnek Akış Tipi Çok İşlemli Çizelgeleme Problemleri, Paralel Doyumsuz Algoritmalar

ABSTRACT

Master Thesis

SOLVING THE HYBRID FLOW SHOP AND HYBRID FLOW SHOP WITH MULTIPROCESSOR TASK SCHEDULING PROBLEMS WITH PARALLEL GREEDY ALGORITHM

Reşide Elif ÖZTÜRK

Selçuk University

Graduate School of Natural and Applied Sciences

Department of Industrial Engineering

Supervisor: Assist. Prof. Dr. Orhan ENGİN

2007, 83 Page

Jury: Prof. Dr. Ahmet PEKER

Assist. Prof. Dr. Orhan ENGİN

Assist. Prof. Dr. M. Emin Baysal

Hybrid flow shop system has a special structure combining some elements of both the flow shop and the parallel machine systems. Hybrid Flow Shop Scheduling (HFS) and Hybrid Flow Shop Scheduling with Multiprocessor Task (HFSMT) are known to be non-polinomiyel (NP)-hard problems. In this study it was suggested that an effective Parallel Greedy Algorithm (PGA) to solve HFS and HFSMT problems. PGA is applied two phases iteratively, named destruction and construction. In study, number of started population, number of sub group, number of iteration, greedy ratio and construction methods had been used as control parameters. The Carlier and Neron (2000) HFS and Oğuz and Ercan (2005) HFSMT problems from literature had been solved with PGA. These solutions had been compared with Döyen (2004)'s Artificial Immune Systems (AIS), Neron (2001)'s Branch and Bound (B&B), Oğuz (2005)'s Genetic Algorithm (GA) and Ceran (2006)'s GA methods that had been analysed in the literature. This comparisons have indicated that the proposed PGA approach's performance is the same of the performances of AIS and Ceran (2006)'s GA approaches; is better than the performances of B&B and Oğuz (2005)'s GA approaches.

Key words: Hybrid Flow Shop Problems, Hybrid Flow Shop With Multiprocessor Tasks Scheduling Problems, Parallel Greedy Algorithms

ÖNSÖZ

Son yıllarda, çizelgeleme problemlerinde, optimum çözüme ulaşabilmek için bir çok meta sezgisel yöntemler kullanılmaktadır. Bu yöntemlerden biri olan Doyumsuz Algoritmalarla ilgili literatürde çok az çalışma yapılmıştır. Yapılan araştırmalarda genel olarak algoritmanın alt sınır değere yaklaştığı görülmüştür. Çalışmada; Doyumsuz Algoritma tekniği, akış tipi paralel makine problemlerine uygulanmıştır ve Paralel Doyumsuz Algoritma tekniği oluşturulmuştur.

Çalışmanın ortaya çıkması sürecinde yardım ve desteğini hiçbir zaman esirgemeyen, danışman hocam sayın Yrd.Doç.Dr. Orhan ENGİN'e, bilgisayar programı yazılımı sırasında yardımlarını esirgemeyen sayın End. Müh. M.Kerim YILMAZ'a ve hiçbir zaman desteğini esirgemeyen aileme, candan sevdiğim dostlarıma ve sevgili eşim Celil ÖZTÜRK'e sonsuz teşekkürlerimi sunarım.

Reşide Elif ÖZTÜRK

Mart - 2007

İÇİNDEKİLER

Sayfa No

ÖZET i
ABSTRACT ii
ÖNSÖZ iii
İÇİNDEKİLER iv
ŞEKİL LİSTESİ v
TABLO LİSTESİ vi
EKLER LİSTESİ vii
KISALTMALAR viii
1. GİRİŞ 1
2. KAYNAK ARAŞTIRMASI 3
2.1. Esnek Akış Tipi Çizelgeleme Problemleri İle İlgili Kaynak Araştırması 3
2.2. Doyumsuz Algoritma İle İlgili Kaynak Araştırması 5
3. MATERYAL VE METOT 12
3.1. Materyal 12
3.1.1. Esnek Akış Tipi Çizelgeleme (EATÇ) 12
3.1.2. Çok İşlemcili Esnek Akış Tipi Çizelgeleme (ÇİEATÇ) 13
3.2. Metod 16
3.2.1. Paralel Doyumsuz Algoritmalar (PDA) 16
3.2.1.1. Doyumsuz Rassal Uyarlamalı Arama İşlemi (DRUAİ) 19
3.2.1.2. PDA'nın uygulandığı örnekler 20
3.2.2. PDA İşlem Adımları 25
3.2.3. PDA Programı İçin Kullanılan Parametreler 27
4. ARAŞTIRMA SONUÇLARI 31
4.1. Veri Toplama 31
4.2. EATÇ Problemlerinin Çözümlerinin Analizi 33
4.3. ÇİEATÇ Problemlerinin Çözümlerinin Analizi 38
5. SONUÇ ve ÖNERİLER 52
6. KAYNAKLAR 54
EKLER 59

ŞEKİL LİSTESİ

Sayfa No

Şekil 3.1 Esnek Akış Tipi Sistem Modeli 13
Şekil 3.2 Çok İşlemcili Esnek Akış Tipi Sistem Modeli 15
Şekil 3.3 PDA'da Kullanılan İnşa Yöntemleri 18
Şekil 3.4 EKKAM'ın Kruskal's Algoritması ile Oluşturulması 21
Şekil 3.5 EKKAM'ın Prim's Algoritması ile Oluşturulması 22
Şekil 3.6 PDA İşlem Adımları 25
Şekil 4.1 PDA Programına ait ekran görüntüsü 31

TABLO LİSTESİ

Sayfa No

Tablo 2.1 Esnek Akış Tipi Çizelgeleme İle İlgili Yapılan Çalışmalar 4
Tablo 2.2 Doyumsuz Algoritma İle İlgili Yapılan Çalışmalar 5
Tablo 3.1 Gezgin Satıcı Problemi İle İlgili Bir Örnek 23
Tablo 3.2 PDA Programında Kullanılan Parametreler 28
Tablo 3.3 EATÇ Problemleri İçin Kullanılan Parametre Değerleri 29
Tablo 3.4 ÇİEATÇ Problemleri İçin Kullanılan Parametre Değerleri 30
Tablo 4.1 EATÇ 10x5 Tipi Problemler İçin Kıyaslama 33
Tablo 4.2 EATÇ 10x10 ; 15x5 ; 15x10 Tipi Problemler İçin Kıyaslama 34
Tablo 4.3 PDA, YBS ve DSA Sezgisel Metodların Performansları 37
Tablo 4.4 ÇİEATÇ *P* tipi 2 aşamalı Problemler İçin Kıyaslama 39
Tablo 4.5 ÇİEATÇ *P* tipi 5 aşamalı Problemler İçin Kıyaslama 41
Tablo 4.6 ÇİEATÇ *P* tipi 8 aşamalı Problemler İçin Kıyaslama 43
Tablo 4.7 ÇİEATÇ *Q* tipi 2 aşamalı Problemler İçin Kıyaslama 45
Tablo 4.8 ÇİEATÇ *Q* tipi 5 aşamalı Problemler İçin Kıyaslama 47
Tablo 4.9 ÇİEATÇ *Q* tipi 8 aşamalı Problemler İçin Kıyaslama 49
Tablo 4.10 PDA ve GA Sezgisel Metodlarının Performansları 51

EKLER LİSTESİ

Sayfa No

Ek 1- 1 10x5 tipi problemler için optimizasyon sonuçları.……60
Ek 1- 2 10x10 tipi problemler için optimizasyon sonuçları...……61
Ek 1- 3 15x5 tipi problemler için optimizasyon sonuçları……..…….....62
Ek 1- 4 15x10 tipi problemler için optimizasyon sonuçları…….....63
Ek 2- 1 *P* tipi 2. aşama problemleri için optimizasyon sonuçları….....64
Ek 2- 2 *P* tipi 5. aşama problemleri için optimizasyon sonuçları.....66
Ek 2- 3 *P* tipi 8. aşama problemleri için optimizasyon sonuçları.....68
Ek 3- 1 *Q* tipi 2. aşama problemleri için optimizasyon sonuçları.....70
Ek 3- 2 *Q* tipi 5. aşama problemleri için optimizasyon sonuçları.....72
Ek 3- 3 *Q* tipi 8. aşama problemleri için optimizasyon sonuçları.....74

KISALTMALAR

NP	Polinomiyel Olmayan
DA	Doyumsuz Algoritma
PDA	Paralel Doyumsuz Algoritma
DRUAİ	Doyumsuz Rassal Uyarlamalı Arama İşlem
EATÇ	Esnek Akış Tipi Çizelgeleme
ÇİEATÇ	Çok İşlemcili Esnek Akış Tipi Çizelgeleme
GA	Genetik Algoritma
YBS	Yapay Bağışıklık Sistemi
DSA	Dal Sınır Algoritması
ASD	Alt Sınır Değeri
$\mathbf{C_{max}}$	En Geç Tamamlanma Zamanı
İS	İşlem Süresi
BAİY	Birer Atlayarak İşleri Yerleştirme
İAİY	İkişer Atlayarak İşleri Yerleştirme
ÖY	Öne Yerleştirme
SY	Sona Yerleştirme
EKKAM	En Küçük Karar Ağacı Modeli

1. GİRİŞ

Esnek akış tipi çizelgeleme problemlerinin çözülmesi çok fazla çaba ve zaman gerektirdiğinden, son yıllarda bu problemler için optimal çözüm veren sezgisel yöntemler, çözüm sürecinde etkin olarak kullanılmaktadır. Son yıllarda çizelgeleme problemlerinde kullanılan meta sezgisel yöntemlerden biri de Doyumsuz Algoritma (DA)'dır.

DA, mevcut bilgiyle, bu bilginin gelecekte ne gibi etki doğuracağını belirlemeden karar alan bir algoritmadır. Algoritmayı uygularken şu an için mantıklı gözüken bir karar, belki de gelecek durumlar için daha kötü bir sonuç olabilecektir. DA, bunu düşünmeği için, çoğu zaman tasarlanması, yazılması ve anlaşılması daha kolay, çalışma sırasında da etkinliği daha fazla olabilmektedir. Fakat bu algoritmaların her zaman için tam optimum sonucu vermeleri beklenmemelidir (Anonim (1) 2004).

DA, erken karar alır. Her adım uygulandığı zaman sonraki adımda verilen karar önceki kararın yerini alır ve önceki eski kararları yeniden düşünmez. Bazı problemler için alınan kararlar kesin, tam ve doğru olmayabilir (Anonim (5) 2006).

DA, optimum çözümü ayrıntılı bir şekilde bulmaz. Çünkü her veri üzerinde ayrıntılı çalışmaz. En iyi tüm çözümlerin sonradan bulunmasını önlemek için çok erken ve çabuk bir şekilde kesin çözümler bulur. Çabuk çözüm bulmak istemesinden dolayı her zaman optimum çözüme ulaşamaz.

DA, problemin en iyi çözümünü her zaman veremese de genellikle gerçek çözümlerin tahminini iyi yapar. DA'nın temel prensibi, algoritmanın üzerinde çalışacağı elemanları bir kritere göre sıralamak ve sıra ile deneyerek en sonunda en uygun çözümü elde etmesidir.

Çalışmada, Esnek Akış Tipi Çizelgeleme (EATÇ) ve Çok İşlemcili Esnek Akış Tipi Çizelgeleme (ÇİEATÇ) problemlerini çözmek için etkili bir Paralel Doyumsuz Algoritma (PDA) geliştirilmiştir. PDA, yıkım ve inşa olarak iki aşamada uygulanır. Çalışmada; başlangıç popülasyonu, alt grup sayısı, iterasyon sayısı,

doyum oranı ve inşa yöntemleri olarak adlandırılan kontrol parametreleri tanımlanmıştır. Literatürdeki Carlier ve Neron'un (2000) EATÇ Problemleri için çalışmada önerilen PDA metodu ile elde edilen sonuçlar, Döyen (2004)'in Yapay Bağışıklık Sistemi (YBS) ve Neron (2001)'un Dal Sınır Algoritması (DSA) yöntemleri ile elde edilmiş sonuçlar ile karşılaştırılmıştır. Ayrıca Literatürdeki Oğuz ve Ercan'ın (2005) ÇİEATÇ problemleri için çalışmada önerilen PDA metodu ile elde edilen sonuçlar, Oğuz (2005)'un Genetik Algoritma (GA) ve Ceran (2006)'ın GA metodu ile elde edilmiş sonuçlar ile karşılaştırılmıştır.

Çalışmanın ikinci bölümünde esnek akış tipi çizelgeleme problemleri ve doyumsuz algoritmalar üzerine literatürde yapılmış çalışmalar özetlenmiştir. Üçüncü bölümde, tez kapsamında ele alınan materyaller olan EATÇ, ÇİEATÇ ve bu materyaller ile ilgili problemleri çözmek için kullanılan metod olan PDA yöntemi ve bu yöntemde kullanılan parametreler ve teknikler incelenmiştir. Dördüncü bölümde EATÇ ve ÇİEATÇ problemlerinin, PDA yöntemiyle elde edilen sonuçlar ve bu sonuçların literatürde önceden yapılmış çalışmalarla kıyaslanması yer almaktadır. Sonuç bölümünde ise; elde edilen bulgular tartışılmıştır.

2. KAYNAK ARAŞTIRMASI

Kaynak araştırması; Esnek Akış Tipi Çizelgeleme Problemleri ve Doyumsuz Algoritmalar olarak iki ana konu üzerinde yapılmıştır.

2.1. Esnek Akış Tipi Çizelgeleme Problemleri ile İlgili Kaynak Araştırması

Esnek akış tipi problemler üzerine geniş bir literatür bulunmasına rağmen, araştırmaların endüstriyel uygulamaları çok azdır. Gerçek çizelgeleme problemlerini çözmek için genellikle öncelik kurallarına dayalı sezgiseller kullanılmıştır. Kullanılan sezgisellerde, işler sadece ilk kademede sıralanır ve aynı sıra tüm kademeler boyunca devam eder. Fakat optimum çözüme ulaşmak için her kademede işler yeniden çizelgelenmelidir. NP (Polinomiyel olmayan) – Zor olarak bilinen esnek akış tipi çizelgeleme problemlerinin PDA metodu ile elde edilen çözüm kalitesi, PDA'da kullanılan parametrelere bağlıdır. PDA'da optimum veya optimuma yakın çözüm veren parametreler, problemlerin yapısına göre değişmektedir.

Tablo 2.1'de esnek akış tipi çizelgeleme ile ilgili son yıllarda yapılan çalışmalar, sunulmuştur.

Tablo 2.1 Esnek Akış Tipi Çizelgeleme İle İlgili Yapılan Çalışmalar

Yıl	Yazar	**EATÇ ve ÇİEATÇ Problemleri İle İlgili Yapılan Çalışmalar**
2005	Oğuz C. , Ercan F.	ÇİEATÇ problemlerinin çözümü için bir GA modeli önermişlerdir.
2004	Engin O, Döyen A.	EATÇ problemlerinin çözümü için etkili bir yapay bağışıklık sistem modeli önermişlerdir.
2004	Oğuz C., Zinder Y., Do V.H., Janiak A., Linchtenstein M.	ÇİEATÇ problemlerinin çözümü için tabu arama modeli önermişlerdir.
2004	Sivrikaya Şerifoğlu S.F. , Ulusoy G.	ÇİEATÇ problemlerinin çözümü için bir GA modeli önermişlerdir.
2002	Su L.H.	İki aşamalı (1. aşamada çoklu işlemli ve 2. aşamada tek işlemli) sınırlı bekleme zamanlı EATÇ problemini sezgisel bir algoritma ile çözmüştür.
2001	Neron E., Baptiste P. , Gupta J.N.D.	EATÇ problemleri için yeterlilik testleri ve zaman-sınırı ayarlamaları kullanımının DSA'nın verimini artırdığını göstermişlerdir.
2000	Moursli O., Pochet Y.A.	EATÇ problemlerinin çözümü için etkili bir DSA modeli önermişlerdir.
1999	Brah S.A., Loo L.L.	Beş farklı sezgisel metodun performansını farklı yapıdaki problemler üzerinde denemişlerdir.
1998	Portmann M.C., Vignier A., Dardilhac D., Dezalay D.	EATÇ problemlerinin çözümü için etkili bir DSA modeli önermişlerdir.
1998	Raine F., Artiba A., Elmaghraby S.E.	Günlük hayattan bir problemi incelemişler ve iyi sonuçlar veren birçok sezgisel metodlar önermişlerdir.
1997	Gupta J.N.D., Hariri A.M.A, Potts C.N.	Arama ağacının sınırlarını daraltmak için bir baskınlık kuralı oluşturmuşlardır. Bir dallandırma kuralı ilave etmişler ve başlangıç üst sınırını oluşturmak için 13 farklı sezgisel kullanmışlardır.
1994	Chung Y.L. , Vairaktarakis G.L.	İki aşamalı EATÇ problemi üzerinde çalışmışlardır.
1994	Edwin S.H.H., Ansari N., Ren H.	Determistik tabanlı bir ÇİEATÇ problemini, GA yöntemi ile çözmüşlerdir.
1992a	Rajendran C., Chaudhuri D.	Toplam tamamlanma zamanını minimize etmek için, DSA'yı geliştirmişlerdir.
1992b	Rajendran C., Chaudhuri D.	Toplam iş akışı zamanını minimize etmek için, DSA'yı geliştirmişlerdir.
1991	Gupta J.N.D., Tunç E.A.	Tamamlanma zamanının minimizasyonuna yönelik olurlu bir çözüm bulmak için 2 tane polinomial sınırlı sezgisel algoritma önermişlerdir.
1991	Brah S.A. , Hunsucker J.L.	En düşük alt sınır arama tekniğini, tamamlanma zamanının minimizasyonu için kullanmışlardır.
1989	Sriskandarajah C., Sethi S.P.	Tamamlanma zamanının minimizasyonu için çeşitli algoritmaların performanslarını karşılaştırmışlardır.
1988	Gupta J.N.D.	İki aşamalı EATÇ problemlerini güçlü bir sezgiselle NP (Polinomiyel olmayan) – Zor olduğunu kanıtlamıştır.
1987	Kochar S., Morris R.J.T.	EATÇ problemlerinin çözümü için sezgisel yöntemler önermişlerdir.
1971	Arthanari T.S. , Ramaurthy K.G.	EATÇ problemi üzerine ilk çalışmayı yapmış olup tamamlanma zamanının minimizasyonu için etkili bir DSA metodunu önermişlerdir.

2.2. Doyumsuz Algoritma İle İlgili Kaynak Araştırması

Doyumsuz algoritmalar, işlevlerini hızlı, kolay ve etkin yerine getirebilmekte ve sıklıkla en iyi sonucu verebilmektedirler. Bu yüzden son yıllarda bu konu üzerinde çalışmalar yoğunlaşmıştır. Günümüze kadar, doyumsuz algoritma ile ilgili yapılmış olan bazı çalışmalar Tablo 2.2'de belirtilmiştir.

Tablo 2.2 Doyumsuz Algoritma İle İlgili Yapılan Çalışmalar

Yıl	Yazar	Doyumsuz Algoritma İle İlgili Yapılan Çalışmalar
2006	Abdekhodaee A.H., Wirth A., Gan H.S.	Doyumsuz algoritmayı tek servisli iki paralel makinenin çizelgelemesinde kullanmışlardır.
2006	Ruiz R. ve Stützle T.	Permütasyon akış tipi çizelgeleme problemleri için tekrarlamalı doyumsuz algoritmasının verimini incelemişlerdir.
2005	Gupta S.R., Smith J.S.	Tek makine çizelgeleme problemleri için hazırlık zamanına bağlı sıralamada toplam gecikme zamanını en aza indirmek için Doyumsuz Rassal Uyarlamalı Arama İşlem (DRUAİ) metodu ile bir uygulama çalışması yapmışlardır.
2005	Papakonstantinou P.A.	İş çizelgeleme için öncelik algoritmalarının sınıflandırılması konusunda bir çalışma yapmışlardır.
2004	Suriyaarachchi R.H., Wirth A.	Yaygın teslim zamanları için toplam en erken ve en geç teslim zaman maliyetlerinin minimize edilmesi için tek işlemci üzerinde iş çizelgeleme probleminin çözümünü doyumsuz algoritma kullanarak bulmuşlardır.
2004	Jensen J.B., Gutin G., Yeo A.	Doyumsuz algoritmaların başarısız olmaları halinde ne gibi durumlarla karşılaşıldığını incelemişlerdir.
2004	Bertel S., Billaut J.C.	Sanayide karşılaşılan bazı çizelgeleme problemleri için etkili bir sezgisel algoritma seçimi yapmaya çalışmışlardır.
2003	Kurtz M.E., Askin R.G.	Esnek akış tipi çizelgeleme kurallarının karşılaştırılması konusunda çalışma yapmışlardır.
2003	Kang J., Park S.	Doyumsuz algoritmayı, değişken ölçülerdeki kutuların ambalajlanması problemlerinin çözümünde kullanmışlardır.
2003	Aiex R.M., Binato S. , Resende M.G.C.	İş çizelgeleme problemlerinin çözümü için döngüsel rotalı paralel DRUAİ metodunu kullanmışlardır.

Tablo 2.2 Doyumsuz Algoritma İle İlgili Yapılan Çalışmalar (Devamı)

Yıl	Yazar	Doyumsuz Algoritma İle İlgili Yapılan Çalışmalar
2001	Binato S., Hery W., Loewenstern D., Resende M.	Paralel doyumsuz algoritmalar ile ilgili DRUAİ yaklaşımını incelemişlerdir.
2001	Alidae B., Kochenberger G.A., Amini M.M.	Doyumsuz algoritmaları, seçme ve sıralama problemlerinin çözümünde kullanmışlardır.
2001	Yao M.J.	Periyodik üretimde azami yükün minimizasyonu problemlerinde doyumsuz algoritmaların uygulamasını yapmıştır.
2000	Lagodimos A.G., Leopoulos V.	Doyumsuz algoritmayı, yiyecek üreten bir fabrikada iş gücü vardiya planlanması yapmak için kullanmışlardır.
2000	Tang L., Liu J., Rong A., Yang Z.	Shangai Baoshan Demir Çelik Tesisinde demir çelik üretiminin çizelgelemesi için çok çeşitli gezgin satıcı modelini önermişler ve uygulamışlardır.
1999	Lui K.S., Zaks S.	Eş zamanlı ağların çizelgelemesinde doyumsuz algoritmaların uygulamasını yapmışlardır.
1999	Faigle U., Kern W., Nawijn W.M.	'k' aralıklı iş problemleri için doyumsuz sürekli (on-line) algoritmasının dar sınırlar içinde bir uygulamasını yapmışlardır.
1998	Anily S., Glass C.A., Hassin R.	Makinelerin bakımlarının çizelgelemesinde doyumsuz algoritma ile ilgili çalışma yapmışlar ve en uygun çözümü hesaplamışlardır.
1989	Anonim (12) - 2005	Foe ve Resende, Polinomiyel Olmayan (NP) - zor problemlerin çözümünde DRUAİ metodunu geliştirmişlerdir

Abdekhodaee A.H., Wirth A. ve Gan H.S. (2006), doyumsuz algoritmayı tek servisli iki paralel makinenin çizelgelemesinde kullanmışlardır. Çalışmaya göre tek servis, ilk işlemi veya hazırlık zamanını uygulayabilir olmalıdır. İkinci işlem hiçbir servise gerek kalmadan yürütülebilir olmalıdır. Uygulanan doyumsuz algoritmanın amacı makinelerdeki boş zamanları ve işlerin bekleme zamanlarını önlemektir. Doyumsuz algoritmanın uygulamasında ileriye ve geriye doğru yaklaşım kriterleri belirlenmiş ve bu kriterler adımlar halinde listelenmiştir. İleriye doğru listelemenin amacı, makinelerdeki boş zamanları en aza indirmektir. Bu listelemede işler, hazırlık zamanları en az olandan en çok olana doğru sıralanır. Geriye doğru listelemenin amacı, işlerin bekleme zamanını en aza indirmek veya sıfırlamaktır. Bu listelemede ise işler, işlem sürelerinin en az olanından en çok olanına doğru sıralanır. Yapılan çalışmada doyumsuz algoritmanın bu versiyonları kullanılarak makinelerdeki atıl zamanları ve işlerin bekleme süreleri en aza indirilmiştir.

Ruiz R. ve Stützle T. (2006), permütasyon akış tipi çizelgeleme problemleri için tekrarlı doyumsuz algoritmanın verimini incelemişlerdir. Tekrarlı doyumsuz algoritma iki evrede uygulanmıştır. Bunlar yıkma ve inşa etmedir. Yıkma evresi, bazı işlerin zorunlu olarak elimine edilmesinden dolayı oluşur. İnşa evresi ise elenmiş işlerin tekrardan sıralamaya konulup sezgisel bir inşa metodunun kullanılması ile oluşur. İsteğe bağlı olarak bölgesel araştırma inşa evresinden sonra yapılabilir. Yapılan çalışmada tekrarlı doyumsuz algoritmaların uygulaması çok kolay ve deneysel olarak ispat edilmiştir. Diğer metotlarla karşılaştırıldığında çok etkili bir algoritma olduğu gözlemlenmiştir.

Gupta S.R. ve Smith J.S. (2005), tek makine çizelgeleme problemleri için hazırlık zamanına bağlı dizide toplam gecikme zamanını en aza indirmek için Doyumsuz Rastgele Uyarlamalı Arama İşlem (DRUAİ) metodu ile bir uygulama çalışması yapmışlardır. Bu metoda göre uygulamaya asıl katkılar; yapısal aşamada, yeni bir maliyet fonksiyonunu, gelişme aşamasında değişken komşu araştırmasının değişkenlik miktarını belirlemektir. Çözümler bulunurken DRUAİ kullanılırsa sürekli olarak kesin en uygun çözümü bulunur. Çalışmada karınca kolonileri gibi birkaç sezgisel metodla DRUAİ metodu karşılaştırılmış ve bazen daha iyi bazen ise daha kötü yani gecikme zamanında uzamalar olduğu görülmüştür. Ama sonuçlar DRUAİ'ın istikrarlı ve rekabetçi bir tutumunun olduğunu göstermiştir.

Papakonstantinou P.A. (2005), iş çizelgeleme için öncelik algoritmalarının sınıflandırılması konusunda bir çalışma yapmışlardır. Çalışmaya göre, öncelik algoritması, doyumsuz algoritma kavramının özelliğini kaybetmeden korunarak hesaplama yapılan bir modeldir. Bu çalışmada öncelik algoritması problemleri için doyumsuz öncelik algoritmasının farklı güçte olup olmadığı belirlenmek istenmiştir. Sonuçta kesin bir öncelikli algoritma, doyumsuz öncelikli algoritma tarafından her girdi için taklit edilemediği görülmüştür.

Suriyaarachchi R.H. ve Wirth A. (2004), yaygın teslim zamanları için toplam en erken ve en geç teslim zaman maliyetlerinin minimize edilmesi için tek işlemci üzerinde iş çizelgeleme probleminin doyumsuz algoritma ve genetik algoritma ile yapılan çözümleri karşılaştırma yaparak incelemişlerdir. Çalışmada, daha yüksek değerli sırada olan ve daha az işlem süreli işler belirlenerek doyumsuz davranış

durumunda toplam en erken ve en geç teslim zamanının maliyetini minimize etmek için doyumsuz algoritma önerilmiştir. Bu algoritma bu problem için aşağıdaki gibi çalışmaktadır.

a) Verilen değerler hesaplanır. Teslim zamanında tamamlanan bir işin olduğu çizelgedeki tüm konumlar birleştirilir.

b) Bir işi belirlemek için en yüksek değerli olan ve önceden seçilmemiş konum seçilir. Konumların birleşmesi yüzünden konumun bağlı olduğu ve önceden seçilmemiş daha fazla kısım seçilir.

c) Sistemin çalışır durumda kalması için iş eklenmezse, işlem sürelerine hazırlık zamanları eklenir ve her sınıftan önceden belirlenmemiş en kısa iş göz önüne alınır. Minimum hesaplanmış iş seçilir ve seçilmiş konuma o iş ayrılır. Konumlardaki bağ yüzünden bir sonraki daha kısa hazırlık zamanlı işten ve ek bir hazırlıktan kaçınmak için bir iş seçilir. İsteğe bağlı olarak herhangi başka bağlar kırılabilir. Sistemdeki tüm işler belirlenene kadar "b" ve "c" adımları yapılmaya devam edilir.

Suriyaarachchi R.H. ve Wirth A. (2004), çalışmalarında kullandıkları bir programlama dili ile doyumsuz algoritma ve genetik algoritma yöntemleri kullanılmış ve aynı problemler çözülmüştür. Sonuçlar arasında karşılaştırma yapılmıştır. Doyumsuz algoritma, genetik algoritmaya göre daha etkin elde edilmiştir.

Jensen J.B., Gutin G. ve Yeo A. (2004), doyumsuz algoritmaların başarısız olmaları halinde ne gibi durumlarla karşılaşıldığını incelemişlerdir. Yapılan çalışmada sonlu bir sıradan değerler alındığı zaman bağımsız sistemde değerlerin en alt tabanının bulunduğu problemler için doyumsuz algoritmaların en kötü tek olası çözümü üretmesi sonucu olayların tanımlanması sağlanmıştır. Bu teoremin uygulaması gezgin satıcı ve en az iki eşit parçaya bölme problemleri için yapılmıştır.

Bertel S. ve Billaut J.C. (2004), sanayide karşılaşılan bazı çizelgeleme problemleri için etkili bir sezgisel algoritma seçimi yapmaya çalışmışlardır. İncelenen yöntemler dinamik programlama, doyumsuz algoritma ve genetik algoritmalardır. Çalışmada doyumsuz algoritma adımları çalıştırılmış ve genetik

algoritma yaklaşımı da uygulanmıştır. İki yaklaşım sonucunda en iyi çözümü veren Yavaş Oran adında yeni bir kural elde edilmiştir.

Kurtz M.E. ve Askin R.G. (2003), esnek akış çizelgeleme kurallarının karşılaştırılması konusunda çalışma yapmışlardır. Bu çalışmada, bağımsız ardışık hazırlık zamanlı esnek akış çizelgelemesini doyumsuz algoritma yaklaşımı ile incelenmiş ve sonuç olarak gereken şartları en iyi şekilde sağladığı görülmüştür.

Kang J. ve Park S. (2003), doyumsuz algoritmayı, değişken ölçülerdeki kutuların ambalajlanması problemlerinin çözümünde kullanmışlardır. Kutuların, bölünebilen ve bölünemeyen büyüklüklerinin analizi yapılmış ve kullanılan kutuların toplam maliyetini en aza indirecek uygun büyüklüklerin bulunması amaçlanmıştır. Doyumsuz algoritma bu problem için en uygun çözümü vermiştir.

Aiex R.M., Binato S. ve Resende M.G.C. (2003), iş çizelgeleme problemlerinin çözümü için döngüsel rotalı paralel DRUAİ metodunu kullanmışlardır. Sonuç olarak iş çizelgeleme problemlerinin çözümünde DRUAİ metodu kaliteli ve iyi bir yaklaşım göstererek işlerin makinelerdeki tamamlanma zamanları an aza indirilmiştir.

Binato S., Hery W., Loewenstern D. ve Resende M. (2001), paralel doyumsuz algoritmalar ile ilgili DRUAİ yaklaşımını incelemişlerdir.

Alidae B., Kochenberger G.A. ve Amini M.M. (2001), doyumsuz algoritmaları, seçme ve sıralama problemlerinin çözümünde kullanmışlardır. Yapılan çalışmada en iyi doyumsuz algoritma bulunup, bu bulunan en iyi algoritmanın uygunluğu için en gerekli ve verimli şartları araştırmışlardır. Gerçek hayatta birçok sıralama ve seçme probleminde uygulanabildiği ve uygun çözümler verdiğini görmüşlerdir.

Yao M.J. (2001), periyodik üretimde azami yükün minimizasyonu problemlerinde doyumsuz algoritmaların uygulamasını yapmıştır. Bu problem için ilk çizelgesini elde eden doyumsuz algoritmanın verimini incelemiş ve sonuç olarak en uygun üretim çizelgesini elde etmek için azami yükün bölgesel olarak azaltılmasını sağlamıştır. Doyumsuz algoritmada yerel araştırma, ilk üretim çizelgesi ile başlamış ve azami yükün daha fazla olmaması için yük düşük bir seviyede tutulmuştur.

Lagodimos A.G. ve Leopoulos V. (2000), doyumsuz algoritmayı, yiyecek üreten bir fabrikada iş gücü vardiya planlanması yapmak için kullanmışlardır. Çalışmadaki amaç, önceden tanımlanmış üretimler için uygun vardiyalarda çalışmak üzere ihtiyaç olan iş gücünü en aza indirmektir. Uygulamada doyumsuz algoritma, tek vardiyalı ve çok vardiyalı sistemler için incelenmiştir. Doyumsuz algoritma ile elde edilen sonuçlar vardiya zamanı ve iş kalitesi açısından oldukça memnun edicidir.

Tang L., Liu J., Rong A. ve Yang Z. (2000), Shangai Baoshan Demir Çelik Tesisinde demir çelik üretiminin çizelgelemesi için çok çeşitli gezgin satıcı modeli geliştirmiş ve uygulamışlardır. Bu çalışma, Çin'deki demir- çelik fabrikalarının üretim ve yönetim sistemlerini yükseltmek için büyük ölçüde harcanan çabanın bir bölümünü oluşturmaktadır. Demir çelik üretimi hazırlık maliyetlerine bağlı bir sırada yürütülmektedir. Yani hazırlık maliyeti çok yüksektir. Çalışmadaki amaç bu hazırlık maliyetlerini en aza indirmektir. Demir çelik üretimindeki çizelgeleme problemlerini, diğer sezgisel algoritmalarda, çizelgelenmemiş işlerin sırasının seçimini yaparak her çevrimde yeni bir çözüm bularak en uygun çözüme gidilebildiği fakat hazırlık maliyetlerinin arttığı gözlenir. Bu da bize doyumsuz algoritmanın tam zamanında sadece tek bir çevrimde yerel optimum çözümü bulduğunu gösterir. Daha sonraki çevrimler için yüksek hazırlık maliyetleri gerektirdiği için böyle bir çözüm normaldir.

Lui K.S. ve Zaks S. (1999), eş zamanlı ağların çizelgelemesinde doyumsuz algoritmaları uygulamışlardır. Çalışmaya göre işlemci, mesajların gitmesi gereken yerlere ulaşmasını ve ulaşırken son teslim zamanına rast gelecek en uygun çizelgelemeyi tanımlamalıdır. Söz konusu işlemci her adımda ağ bağlantısının kapasitesini aşacak kadar çok mesaj, o ağdan ayrılan herhangi bir ağ bağlantısının üzerinden gönderilemeyebilir. Bu problemin çözümü için bağımsız-dar kapasiteli ağlarda makul bir çözüm için uygun bir çizelgeleme, doyumsuz algoritmalar tarafından belirlenmiştir.

Faigle U., Kern W. ve Nawijn W.M. (1999), k aralıklı iş problemleri için doyumsuz sürekli algoritmasının bir uygulamasını yapmışlardır. k parçalı iş problemleri aslında ara çizelgeleme problemleri olarak da bilinir. Yapılan

uygulamada "*n*" adet iş için "*i*" aralıkları belirlenmiştir. "*k*" adet makinenin aralıklarının herhangi bir makinenin aralığını aşmayacak şekilde en uygun iş, makinede işlenir. İşlem görecek iş, her makineye verilen talimatlara bağlı olarak değişkenlik gösterebilir. Yapılan hesaplamalar sonucunda doyumsuz algoritma yönteminin en uygun çözümü bulduğu görülmüştür.

Anily S., Glass C.A. ve Hassin R. (1998), makinelerin bakımlarının çizelgelemesinde doyumsuz algoritma ile ilgili çalışma yapmışlar ve en uygun çözüm hesaplanmıştır.

Foe ve Resende (1989), zor bilişimsel problemlerin çözümünde DRUAİ metodunu geliştirmişlerdir. DRUAİ, yapısal adım ile gelişme adımını birleştiren eklemeli bir metottur. Yapısal adımda uygulanabilir çözüm, tekrarlı olarak kurulur. Geliştirilen metotta, öncelikle doyumsuz fonksiyon ile ilgili aday liste içerisine tüm aday elemanların listesinin eklenmesine karar verilir. Doyumsuz fonksiyon ile seçilmiş her elemanın yararı ölçülür. Her eleman ile ilişkili yararlar, yapısal adımın her iterasyonunda yenilenir. Listedeki en iyi adaylardan biri rasgele seçilir. Elemanların seçim tekniği her DRUAİ iterasyonunda elde edilmiş olan farklı çözümleri hesaba katar. Çoğu deterministlik metotlarda olduğu gibi, DRUAİ ile üretilen çözümler de garantili çözüm değildir. Ancak komşuluk tanımları ile ilgili yerel optimumlar olabilir. Bu nedenle kurulan her çözümü geliştirme girişiminde bulunmak için yerel araştımaya başvurmak hemen hemen her zaman yararlıdır. Her yapısal adımdan sonra ki gelişme adımında genellikle basit bir yerel arama işlemi yapılmaktadır. Rasgele sıralamadan oluşan başlangıç çözümünün vekil elemanları üzerinde denemeler yapılır. Yerel arama algoritması, yinelemeli bir şekilde geçerli çözümü, komşuluğundaki iyi bir çözüm ile arka arkaya değiştirmeye çalışır. Komşuluk içerisinde daha iyi bir çözüm kalmayana kadar algoritma işler (Anonim (12) 2005).

3. MATERYAL VE METOD

3.1.Materyal

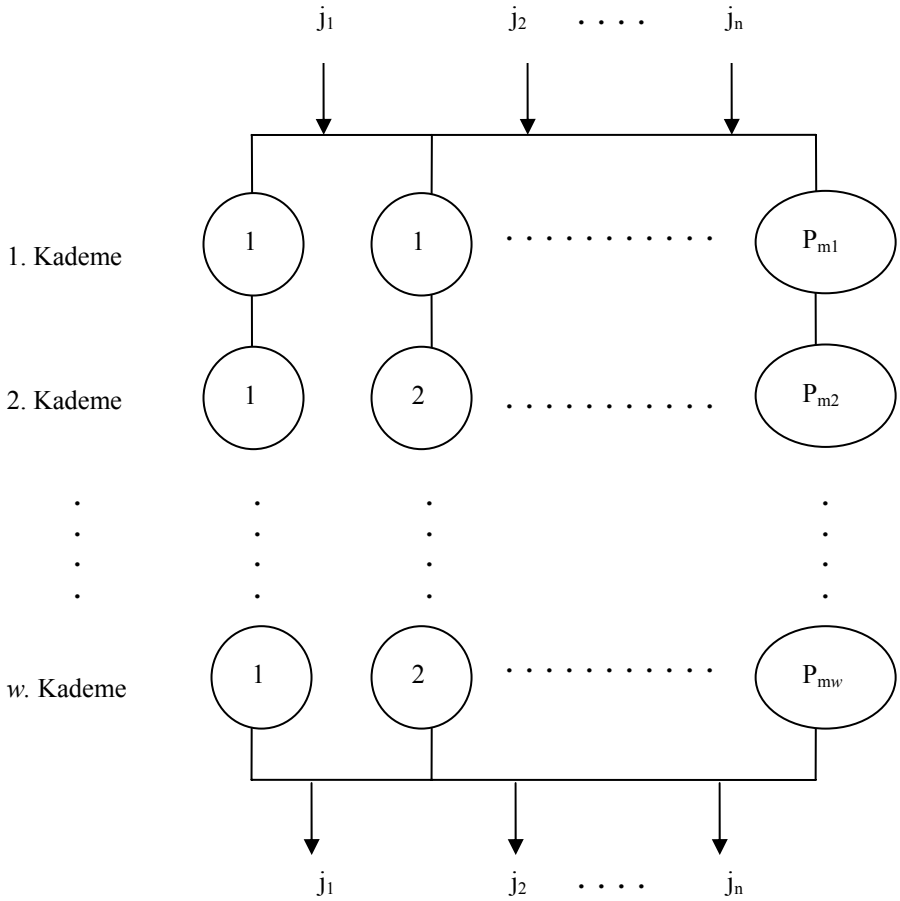
Yapılan tez çalışmasında akış tipi ve paralel makine problemlerinin birleşimi olan esnek akış tipi çizelgeleme materyali ele alınmıştır. Esnek akış tipi çizelgeleme; EATÇ ve ÇİEATÇ olmak üzere iki bölümde incelenmiştir.

3.1.1. Esnek Akış Tipi Çizelgeleme (EATÇ)

Akış tipi çizelgeleme problemlerinde her aşama tek makineden oluşmakta ve işler atölyedeki aşamalarda makineleri aynı sırada ziyaret etmektedir.

Esnek Akış Tipi Çizelgeleme (EATÇ) problemi, klasik akış tipi ve paralel makine problemlerinin bir genelleştirmesi şeklindedir.

Esnek akış tipi sistemde, makineler w tane seri kademeye yerleştirilmişlerdir. $W=(1, 2... w)$ olmak üzere bir w kademesinde, bir veya daha fazla eş makine bulunmaktadır. $j=1,2,.....,n$ olmak üzere bir j işi, önce 1. kademede sonra 2. kademede ve son olarak w. kademede işlem görür. Her j işi aynı zamanda sadece bir makinede ve her bir kademedeki makinelerden herhangi birinde işlem görür. İşlem gören her j işi, $O=O_{1J},...., O_{mJ}$ olmak üzere bir operasyon zinciri oluşturur. Farklı kademelerde j işinin m_w makine için bir O_{mj} operasyonunda $p_{1j}, p_{2j}, ..., p_{wj}$ olmak üzere p_{wj} işlem süreleri vardır. Her makine aynı zamanda en fazla bir işi işleyebilir. Bir operasyona sadece ondan önceki operasyon tamamlandıktan sonra başlanabilir. Tüm işlerin ve tüm makinelerin çizelgeleme süresince her zaman hazır olduğu varsayılmaktadır. Amaç, işlerin tamamlanma zamanını minimize edecek çizelgeyi bulmaktır. Şekil 3.1'de bir esnek akış tipi sistemin yapısı verilmiştir.



Şekil 3.1 Esnek Akış Tipi Sistem Modeli

3.1.2. Çok İşlemcili Esnek Akış Tipi Çizelgeleme (ÇİEATÇ)

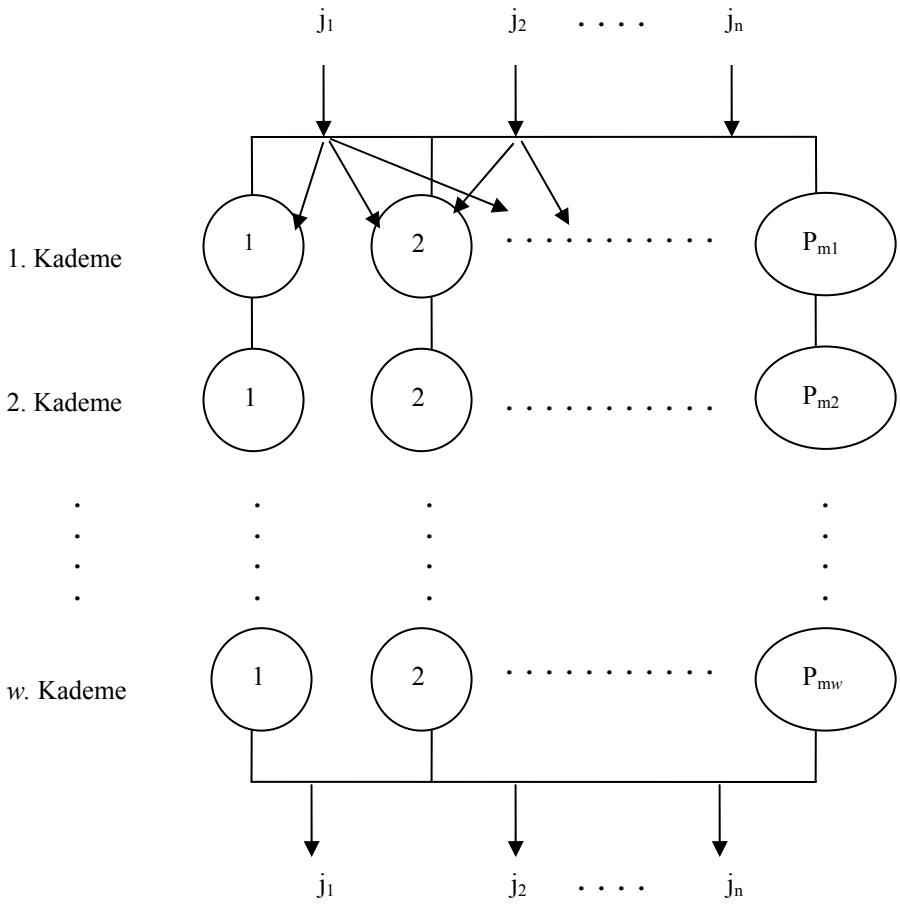
Çok İşlemcili Esnek Akış Tipi Çizelgeleme (ÇİEATÇ); esnek akış tipi çizelgeleme sistemlerinin özelliklerini taşır. Fakat ÇİEATÇ sistemlerinde, bir aşamada herhangi bir iş, birden çok makinede işlenebilmektedir. ÇİEATÇ problemi; klasik akış tipi ve paralel makine problemlerinin bir genelleştirmesi şeklindedir. ÇİEATÇ problemi, çizelgeleme uzunluğunu veya tamamlanma zamanını minimize

etmek amacıyla çok işlemcili sistem üzerinde genel bir iş çizelgesi uygulamak için bir programın oluşumu olarak açıklanabilir.

Son yıllarda ÇİEATÇ problemlerine teorik ve pratik açıdan gereken önem verilmeye başlanmıştır. Bu konu ile ilgili yapılmış olan çalışmalar 2. bölümdeki kaynak araştırması kısmında belirtilmiştir.

ÇİEATÇ sisteminde, $J = \{1, 2 \ldots n\}$ olmak üzere n tane iş, $i = \{1, 2, \ldots\ldots, k\}$ olmak üzere k tane aşamada sıralı bir şekilde işlem görmelidir. Her i aşamasında m_i tane eş paralel makine vardır. Her iş önce 1. aşamada, sonra 2. aşamada ve son olarak k. aşamada işlem görür. Her aşamada sadece bir tip görev yapılmaktadır. Her görev sadece kendinden önceki görev tamamlandıktan sonra işlemine başlayabilir. i. aşamada bulunan j. işin işlenmesi için gereken işlem hacmi $size_{ij}$, i. aşamada bulunan j. işin işlem süresi P_{ij} olsun. Diğer bir deyişle j. işin i. görevi veya aşaması, m_i. makinenin $size_{ij}$ işlem hacmi kadar yapılmalıdır (Oğuz C. ve Ercan F. 2005).

Tüm işlerin ve tüm makinelerin çizelgeleme süresince her zaman hazır olduğu varsayılmaktadır. ÇİEATÇ'nin amacı, işlerin tamamlanma zamanını minimize edecek çizelgeyi bulmaktır. Şekil 3.2'de bir çok işlemcili esnek akış tipi sistemin yapısı verilmiştir.



Şekil 3.2 Çok İşlemcili Esnek Akış Tipi Sistem Modeli

3.2. Metod

Çalışmada kullanılan metod olan Doyumsuz Algoritma (DA) genellikle akış tipi problemleri çözmek için kullanılır. Çalışmada, EATÇ ve ÇİEATÇ problemlerini çözmek için Paralel Doyumsuz Algoritma (PDA) metodu kullanılmıştır. Ele alınan materyal, esnek akış tipi çizelgeleme olduğu için DA, paralel hale gelmiş ve böylece PDA metodu oluşmuştur.

3.2.1. Paralel Doyumsuz Algoritmalar (PDA)

DA, adından da anlaşılacağı gibi, sadece şu an için elinde olan bilgiyle, bu bilginin gelecekte ne gibi etki doğuracağını tam düşünmeden karar alan algoritmalardır. Algoritmayı yazarken şu an için mantıklı gözüken bir karar, belki de gelecek durumlar için daha kötü bir karar olabilecektir. DA bunu düşünmeği için, çoğu zaman tasarlanması, yazılması ve anlaşılması daha kolay, çalışma sırasında da etkinliği daha fazla olabilmektedir. Fakat DA'nın her zaman için tam optimum sonucu vermeleri beklenmemelidir (Anonim (1) 2004).

DA, çalışırken çözüme küçük adımlarla yaklaşır ve her adım sonunda birden çok seçenek çıkabilir. Algoritma, her adımda bu seçeneklerden en iyi olanını seçer. Her zaman en iyiyi seçerek ilerlemek, sonuçta optimum çözümün elde edilmesini sağlayabilir. Ama bu her zaman için geçerli olmayabilir.

DA, kombinasyonel optimizasyon problemleri için diğer sezgisel yaklaşımlardan daha hızlı bir yaklaşım metodudur. DA metodu genelde akış tipi problemleri çözmek için kullanılır. Akış tipi ile paralel makine problemlerinin bir araya gelmesinden oluşan esnek akış tipi problemleri çözmek için kullanılan DA paralel olmakta ve PDA ortaya çıkmaktadır.

PDA genellikle iki aşamada uygulanır. Bu aşamalar; yıkım ve inşa aşamalarıdır.

Yıkım aşaması süresince bazı işler bulunan çözümlerden çıkartılır. İnşa aşamasında ise; önceden çıkartılmış işler Şekil 3.3.'de belirtilen sezgisel inşa yöntemlerinde kullanılmak üzere iş sırasına tekrar alınır.

Yıkım aşaması, n tane işin π iş sırasında uygulanır ve algoritma, n tane işten rastgele ve tekrarlamasız bir şekilde d tane iş seçer. Bu d tane iş π iş sırasından sırayla çıkartılır (Ruiz ve Stützle 2006). Bu sürecin sonucunda iki alt küme elde edilir. İlk alt küme, n-d iş sayılı π_D iş sırasından oluşan sıradır. İkinci alt küme ise d iş sayılı π_R iş sırasından oluşan sıradır. Bu π_R iş sırası, π_D iş sırasının içine tekrar ve sırayla yerleştirilecek, minimizasyonu sağlayıp tamamlanan aday çözümleri sağlayan işleri içermektedir (Ruiz ve Stützle 2006).

İnşa aşaması π_R alt kümesi ile başlar ve π_D iş sırasının içine tekrar geçen işlerin inşası d adımda gerçekleşir.

Bu çalışmada 4 sezgisel inşa yöntemi uygulanmıştır.

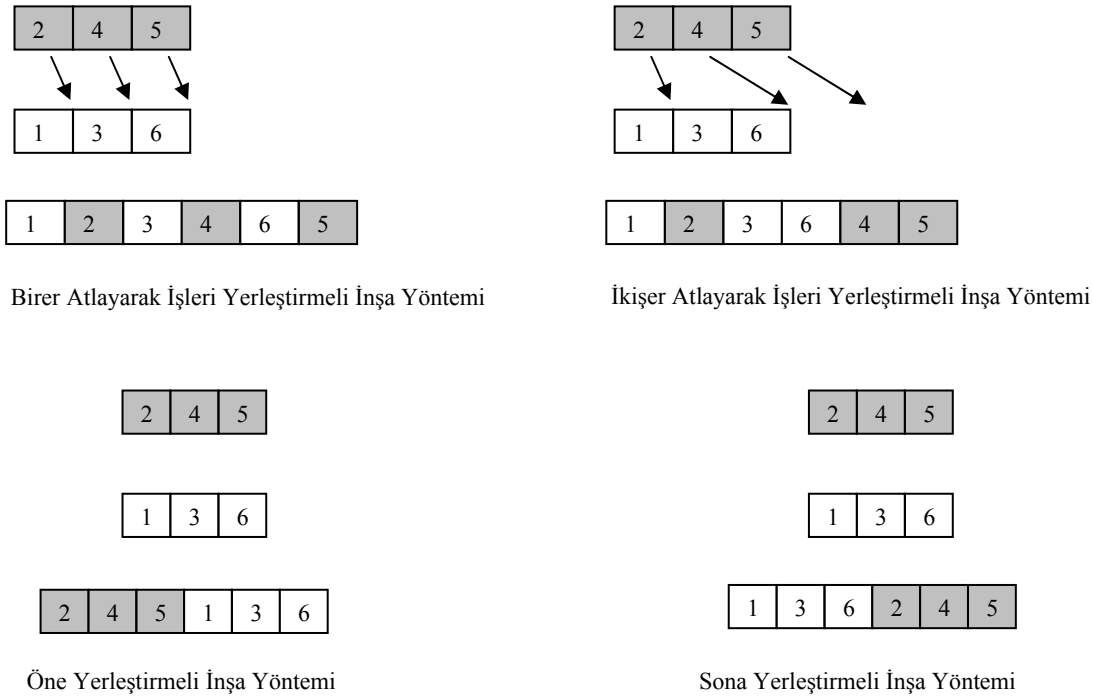
1. Birer atlayarak işleri yerleştirmeli inşa yöntemi : Bu yöntemde, π_R iş sırasının ilk işini π_D iş sırasının ilk başına yerleştirilir. Sonra π_R iş sırasının ikinci işi π_D iş sırasının üçüncü sırasına yerleştirilir. Bu işlem π_R iş sırası boşalana kadar devam eder.

2. İkişer atlayarak işleri yerleştirmeli inşa yöntemi : Bu yöntemde, π_R iş sırasının ilk işini π_D iş sırasının ilk başına yerleştirilir. Sonra π_R iş sırasının ikinci işi π_D iş sırasının dördüncü sırasına yerleştirilir. Bu işlem π_R iş sırası boşalana kadar devam eder.

3. Öne yerleştirmeli inşa yöntemi : Bu yöntemde, π_R iş sırasının tamamı aynı sıra ile, π_D iş sırasının önüne yerleştirilir.

4. Sona yerleştirmeli inşa yöntemi : Bu yöntemde, π_R iş sırasının tamamı aynı sıra ile, π_D iş sırasının sonuna yerleştirilir.

Söz konusu inşa yöntemleri bir örnek üzerinde açıklanmıştır. Bir EATÇ probleminin 6 işten oluştuğu varsayılmıştır. İş sırasının rastsal olarak [1, 2, 3, 4, 5, 6] çıkartıldığı kabul edilmiştir. Yıkım aşamasında rastgele biçimde ayrılan işler [2, 4, 5] ve [1, 3, 6] olsun. Bu işlerin inşa edilmesi için kullanılan dört yöntem Şekil 3.3'de ayrı ayrı gösterilmiştir.



Şekil 3.3 PDA'da Kullanılan İnşa Yöntemleri

PDA'nın bazı temel özellikleri aşağıda belirtilmiştir.

- Her adımda hiç endişelenmeksizin en iyi seçimi yapar. Seçimi sürekli iyileştirir.
- İşlevlerini hızlı ve kolay yerine getirir ve sıklıkla en iyi sonucu verir (Anonim (3) 2005). Fakat çabuk çözüm bulmak istemesinden dolayı her zaman optimum çözüme ulaşamaz.
- Erken karar alır. Her adım uygulandığı zaman sonraki adımda verilen karar önceki kararın yerini alır ve önceki eski kararları yeniden düşünmez. Bazı problemler için alınan kararlar kesin, tam ve doğru olmayabilir (Anonim (5) 2006).
- Çözüm küçük adımlardan oluşur.
- En uygun çözümü ayrıntılı bir şekilde bulmaz. Çünkü her veri üzerinde ayrıntılı çalışmazlar.
- Her algoritma, içeriği ve kalitesi bozulmadan bir doyumsuz algoritmaya dönüşebilir (Anonim (4) 2005).

3.2.1.1. Doyumsuz Rassal Uyarlamalı Arama İşlemi (DRUAİ)

Doyumsuz Rassal Uyarlamalı Arama İşlemi (DRUAİ) metodu Foe ve Resende tarafından, NP (Polinomiyel Olmayan) – Zor bilişimsel problemlerin çözümü için 1989 yılında geliştirilmiştir. DRUAİ algoritması yapısal ve gelişme adımları olarak iki ayrı adımdan meydan gelmektedir (Anonim (13) 2005).

Yapısal adımda uygulanabilir çözüm tekrarlı olarak kurulur. Her yapısal iterasyonda diğer elemanın seçimi yapılır ve parça seçimi konusunda rasgele sıralama işlemi ortaya konulur. Doyumsuz fonksiyon ile ilgili aday liste c içerisine tüm aday elemanların listesinin eklenmesine karar verilir. Doyumsuz fonksiyon (q : C → R) ile seçilmiş her elemanın yararı ölçülür. Nitelendirilen fonksiyon doyumsuz algoritmaya yol gösterir. Çözüm için eleman seçimine dayanan bu fonksiyon algoritmayı çalıştıran programcı tarafından oluşturulur. Her eleman ile ilişkili yararlar önceki elemanın seçiminin sebep olduğu değişimleri yansıtmak için yapısal adımın her iterasyonunda yenilenir. DRUAİ'ın olasılıksal bileşeni listedeki en iyi adaylardan birinin rasgele seçimini tanımlamaktadır. Ancak bu seçilen adayın en iyi aday olması gerekmemektedir. En iyi aday listesine sınırlanmış aday listesi denir. Elemanların seçim tekniği her DRUAİ iterasyonunda elde edilmiş olan farklı çözümleri hesaba katar. Çoğu deterministik metotlarda olduğu gibi, DRUAİ yorumları ile üretilen çözümler de garanti değildir. Basit komşuluk tanımları ile ilgili yerel optimumlar olabilir. Bu nedenle kurulan her çözümü geliştirme girişiminde bulunmak için yerel aramaya başvurmak hemen hemen her zaman yararlıdır (Anonim (13) 2005).

Her yapısal adımdan sonra ki gelişme adımında genellikle basit bir yerel arama işlemi yapılmaktadır. Rasgele sıralamadan oluşan başlangıç çözümünün vekil elemanları (asıl çözümün yerini tutan elemanlar) üzerinde denemeler yapılır. Yerel arama algoritması, yinelemeli bir şekilde geçerli çözümü komşuluğundaki iyi bir çözüm ile arka arkaya değiştirmeye çalışır. Komşuluk içerisinde daha iyi bir çözüm kalmayana kadar algoritma işler (Anonim (13) 2005).

3.2.1.2. PDA'nın uygulandığı örnekler

PDA, bir probleme optimum çözümü sağlayabilen bir yöntemdir. Bu algoritma sezgiseldir ve aşama aşama çalışır. PDA, genelde aşağıdaki problemleri çözmek için kullanılır.

- ✓ Bozuk para problemi
- ✓ En Küçük Karar Ağacı Modeli (EKKAM)
- ✓ Bilgisayar ağları
- ✓ Gezgin satıcı problemi
- ✓ Sırt çantası problemi
- ✓ Akış tipi çizelgeleme problemleri
- ✓ EATÇ problemleri
- ✓ ÇİEATÇ problemleri

PDA'nın uygulandığı örneklerden bazıları aşağıda incelenmiştir.

a) Bozuk para problemi: Bozuk para problemi, girilen para değerini elimizdeki bozuk para kümesindeki paraları kullanarak en uygun şekilde bozmak veya ödenen ücretin para üstünü en optimum şekilde nasıl verilebileceğinin belirlenmesidir. Bozuk para problemi için kullanılacak doyumsuz algoritmanın adımları aşağıdadır (Anonim (2) 2004).

Adım - 1: Seçilebilecek en yüksek değerdeki para belirlenir.
Adım - 2: Seçilen paraların istenen toplam para değerine ulaşıp ulaşmadığına bakılır.
Adım - 3: Eğer ulaştıysa işlem tamamlanmıştır. Eğer ulaşmadıysa 1. adıma dönülür.

Bu algoritmanın doyumsuz olarak tanımlanmasının sebebi, her adımdan sonra en büyük bozuk parayı seçmesidir. Çünkü en büyük bozuk parayı seçmek, optimum çözüm için ilk bakışta mantıklı görünür ve çoğu zaman da bu yaklaşım en optimum çözümü verir. Fakat bazı durumlarda bu mümkün olmamaktadır (Anonim (2) 2004).

DA'nın bozuk para problemi çözümlemesi, bir örnek üzerinde incelemiştir. Bozuk para olarak {50, 25, 10, 1} değerlerinde paralar kullanldığı varsayılmıştır. 82 değerinde bir para üstü verilmek istendiğinde, doyumsuz algoritmaya göre elde edilen bozuk para değerleri '50→25→1→1→1→1→1→1→1'dir. Bu çözümün

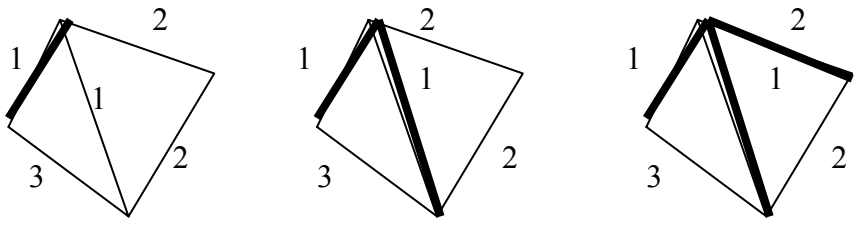
optimum çözüm olmadığı açıktır. Eğer algoritma, para üstünü; '50→10→10→10→1→1' şeklinde vermiş olsaydı, daha uygun bir çözüme ulaşılmış olurdu. DA'nın bu çözüme ulaşamamış olmasının sebebi, 25 birimlik parayı verdikten sonra, gerisini düşünmemesidir (Anonim (2) 2004).

b) En Küçük Karar Ağacı Modeli (EKKAM): Bu modelin girdisi; değer grafiği, modelin çıktısı da değerlerin toplamının başka karar ağaçlarının değerlerinin toplamından büyük olmayan karar ağacıdır. Karar ağacı, n düğümden, n-1 kenardan oluşur (Anonim (7) 2005).

Bu modele DA ilk olarak Kruskal's Algoritması ile uygulanacaktır. Bu algoritma aşağıdaki gibi çalışır (Anonim (6) 2005.

- ✓ Kenarlar en kısadan en uzuna (en küçükten en büyüğe) göre sınıflandırılır. Bu sınıflandırılan kenarlar, en kısadan başlanarak karar ağacına eklenir. Eğer bir kenar bir devir oluşturuyorsa, o köşe ağaçtan çıkartılır. Eğer bir kenar bir devir oluşturmamışsa, devir oluşturana kadar ağaçta kalır. Tüm köşeler bu şekilde eklenip sırasıyla ıskartaya alındıktan sonra karar ağacının zirvesini oluşturacak en kısa kenar bağlanır (Anonim (6) 2005).

Algoritmanın işleyişi Şekil 3.4'de sunulmuştur.

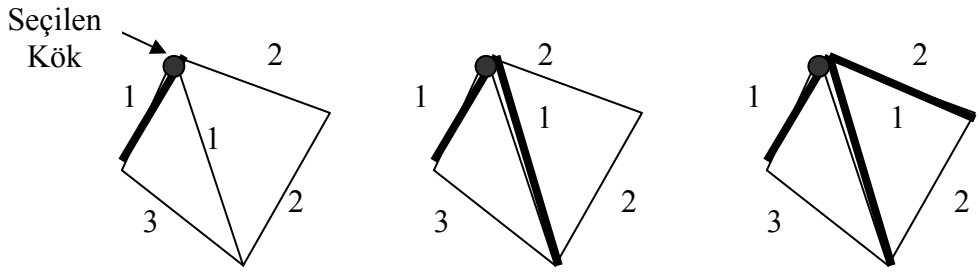


Şekil 3.4 EKKAM'ın Kruskal's Algoritması ile Oluşturulması

EKKAM'ın DA ile çözümü, 2. olarak Prim's Algoritması ile yapılacaktır (Anonim (8) 2005). Prim's Algoritması aşağıdaki gibi çalışır (Anonim (8) 2005).

- ✓ Ağaçta rastgele bir şekilde kök olarak bir düğüm seçilir. Daha sonra, bu düğümün bağlanmış olduğu en az değerdeki kenar seçilip, bu işlem aynı köke bağlı kenarların değerlerinin, azdan çoğa doğru sırayla seçilmesi ile devam eder. Bu işlem aynı köke bağlı kenar kalmayınca sona erer.

Algoritmanın işleyişi Şekil 3.5'de sunulmuştur.



Şekil 3.5 EKKAM'ın Prim's Algoritması ile Oluşturulması

Her 2 yöntem ile ilgili olarak aşağıdaki noktalar dikkati çekmektedir.

- Kruskal's Algoritması sonucunda her zaman bağlanmış kısmi bir ağaç topluluğu olur.
- Prim's Algoritması sonucunda her zaman bağlanmış olan kısmi bir ağaç olur.

c) Bilgisayar ağları: Bir bilgisayar ağı, ağ içindeki düğümler arasında dosya, mesaj aktarımı süreçlerinde doyumsuz algoritmayı kullanabilir (Anonim (6) 2005). Bilgisayar ağlarındaki doyumsuz algoritma uygulaması bir örnekle açıklanabilir. Bir bilgisayar ağındaki varsayılan iki adet düğüm ele alınsın. Bir düğümden diğerine mesaj aktarımı için geçen süreyi göstermek için her iki düğüme de değer verilebilir. Bu verilen değer; transfer mesafesini, fiberoptik kablolar ve bakır tel gibi transfer materyallerini ağdaki bilgisayarların işlemcilerini, hızını, günün hangi zamanında olduğunu ve ağın hızına etki eden diğer tüm etmenleri dikkate alır.

d) Gezgin satıcı problemi: Gezgin satıcı probleminin birçok farklı ve önemli uygulamaları vardır. Ayrıca çok zor ve verimi az olan bir çözüm yöntemidir. Buradaki amaç; problemin optimum çözümüne ulaşmak için yani minimum maliyetle veya minimum mesafede yol alınması için basit doyumsuz algoritmayı geliştirmektir (Anonim (3) 2005). Gezgin satıcı problemi, satıcının farklı sayıda ve farklı şehirleri ziyaret ettiğini kabul eder. Satıcı, bir şehri sadece bir kez ve tam olarak ziyaret etmelidir. Tüm şehirleri ziyaret ettikten sonra ziyaretine başladığı şehre geri döner. Bu problem, Tablo 3.1'de gösterilmiş olan örnek bir problem üzerinde incelenmiştir.

Tablo 3.1 Gezgin Satıcı Problemi İle İlgili Bir Örnek

Şehirler	**A**	**B**	**C**	**D**
B	10	--	--	--
C	9	11	--	--
D	15	13	11	--
E	12	11	10	12

Tablo 3.1'de belirtilen A, B, C, D, E harfleri satıcının gideceği şehirleri, rakamlar ise iki şehir arasındaki seyahat maliyetini göstermektedir.

Gezgin satıcı problemi için oluşturulan basit DA adımları aşağıda sunulmuştur (Anonim(11) 2005).

Aşama 1– İlk ziyaret edilecek şehir satıcı tarafından rastgele seçilir.

Aşama 2– Rastgele seçilen ilk şehir ziyaret edilecek şehir listesinden çıkartılır.

Aşama 3– Daha sonra, bundan önce ziyaret edilmiş şehir arasındaki seyahat maliyeti en az olan şehir seçilir.

Aşama 4– Seçilen şehir ziyaret edilecek şehir listesinden çıkartılır.

Aşama 5– Eğer ziyaret edilecek şehir listesinde şehir kalmışsa Aşama–3'e geri dönüp aynı işlemler uygulanır.

Aşama 6– Eğer ziyaret edilecek şehir listesinde şehir kalmamışsa ilk ziyaret edilen şehre geri dönülür ve böylece tur tamamlanmış olur (Anonim (11) 2005).

Bu aşamalar Tablo 3.1'deki örnek için uygulandıktan sonra aşağıdaki sıra ortaya çıkmaktadır.

A → C → E → B → D → A

9 10 11 13 15

Toplam Maliyet : 9 + 10 + 11 +13 + 15 = 58 parabirimidir (Anonim (3) 2005).

Bu problem başka bir yöntemle çözüldüğü zaman maliyeti daha az olan bir sıralama bulunur.

A → B → C → D → E → A

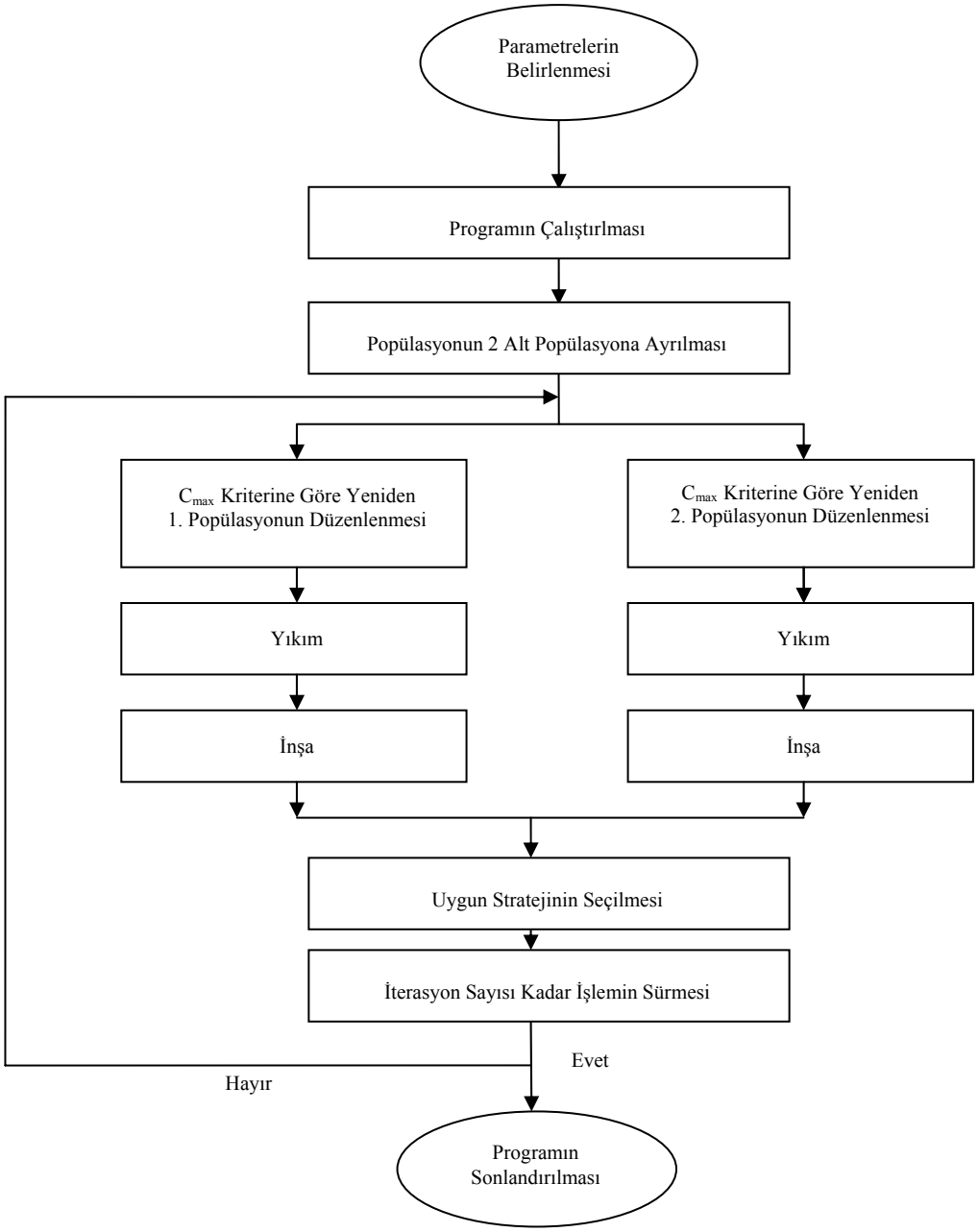
10 11 11 12 12

Toplam Maliyet : 10 + 11 + 11 + 12 + 12 = 56 parabirimidir (Anonim (3) 2005).

Görüldüğü gibi doyumsuz algoritma, her zaman optimum çözümü verememektedir.

3.2.2. PDA İşlem Adımları

Tamamlanma zamanı kriterine bağlı EATÇ ve ÇİEATÇ problemlerinin çözümü için geliştirilen PDA işlem adımları Şekil 3.6'da gösterilmiştir.



Şekil 3.6 PDA İşlem Adımları

PDA işlem adımları, aşağıda açıklanmıştır.

Adım 0: Parametrelerin belirlenmesi;

Başlangıç popülasyonu, alt grup sayısı (π), iterasyon sayısı, doyum oranı, inşa yöntemleri gibi parametre değerleri programa girilir.

Adım 1: Programın çalıştırılması;

Her iş sırası için tamamlanma zamanını bulacak şekilde başlangıç popülasyonu rassal bir şekilde oluşturulur. Program, başlangıç popülasyonun sayısı kadar iş sırası oluşturur.

Adım 2: Popülasyonun iki alt popülasyona ayrılması;

Popülasyon rassal olarak iki alt popülasyona ayrılır.

Adım 3: Yıkım;

πiş sıralarından C_{max}'ı en küçük olan iş sırası seçilir. Seçilen iş sırasından alt küme sayısı kadar rassal işler seçilip πiş sırasında o işler çıkartılır.

Adım 4: İnşa;

En iyi inşa yöntemi belirlenerek çıkartılan işler sırayla tekrar πiş sırasına yerleştirilir. Daha sonra yerel arama yöntemi ile işler, bir sonraki işin yerine atlatılarak tamamlanma zamanını minimize edecek sıra bulunmaya çalışılır.

Adım 5: Uygun Stratejinin Seçilmesi;

Her alt popülasyon ayrı ayrı gelişirken en iyi tamamlanma zamanını bulan uygun bir strateji seçilir.

Adım 6: İterasyon Sayısı Kadar İşlemin Sürmesi;

Girilen iterasyon sayısı kadar program çalıştırılmaktadır.

Adım 7: Programın Sonlandırılması;

Girilen İterasyon sayısına ulaşıldığı takdirde program sonlanır.

3.2.3. PDA Programı İçin Kullanılan Parametreler

Meta sezgisel bir yöntemin kontrol parametrelerinin belirlenmesi oldukça zor bir iştir. PDA'nın performansı, seçilen kontrol parametrelerine bağlıdır. PDA sezgisel yöntemi, birbirine etkisi olan kontrol parametreleri tarafından düzenlenmektedir.

Çalışmada; Carlier ve Neron'un (2000) Esnek Akış Tipi Çizelgeleme (EATÇ) ile Oğuz ve Ercan'ın (2005) Çok İşlemcili Esnek Akış Tipi Çizelgeleme (ÇİEATÇ) problemleri kullanılmıştır. Çalışmada, bu 2 tip problem çözümleri literatür çalışmalarında elde edilen değerler ile karşılaştırılmıştır.

Carlier ve Neron'un (2000) problemleri; *n(iş) x s(aşama)* tipi olmak üzere problemlerin boyutları: 10x5, 10x10, 15x5, 15x10 'dur. İşlem süreleri [3,20] aralığında uniform dağılmaktadır. Toplam 77 adet problem vardır. Problemler; iş sayısı ve kademe sayıları özelliklerine göre gruplandırılabilir. Bir problemin yapısını bu iki özellik belirlemektedir. Örnek bir problem notasyonu: "*j10c10a1*" şeklindedir. Burada; *j10*, 10 iş bulunduğunu; *c10*, 10 kademe bulunduğunu; *a*, kademelerdeki makine yerleşimi yapısını; en sondaki 1 ise örnek indisini göstermektedir. Kademelerdeki makine yerleşimleri aşağıdaki şekildedir:

- Orta kademede 1 makine (darboğaz), diğer kademelerde 3 makine vardır.
- İlk kademede 1 makine (darboğaz), diğerlerinde 3 makine vardır.
- Orta kademede 2 makine(darboğaz), diğer kademelerde 3 makine vardır.
- Her kademede 3 makine var (darboğaz olan kademe yoktur).

Oğuz ve Ercan'ın (2005) problemleri, *n(iş) x s(aşama)* tipi problemlerdir. Bu problemler, *P* ve *Q* olmak üzere 2 tipe ayrılmaktadır. Her 2 tipte de ayrı ayrı toplam 120 adet problem vardır. Örnek problem notasyonu *"P20S8T01"* şeklindedir. Burada *P*, problem zorluk derecesini göstermekte olup *P* ve *Q* değerlerini almaktadır. *Q* tipi problemler daha zordur. *20* sayısı, iş sayısını göstermektedir. Burada 100 iş *H1* ile temsil edilmektedir. *S8*, aşama sayısını; *T01*, ise problem indisini göstermektedir. *" n = 10, 20, 50, 100 "* değerleri, iş sayılarını, *" m = 2, 5, 8 "* değerleri ise aşama sayılarını göstermektedir. Çözümü gerçekleştirilecek olan problemlerin boyutları: *10x2, 20x2, 50x2, 100x2, 10x5, 20x5, 50x5, 100x5, 10x8, 20x8, 50x8* ve *100x8* ' dir.

Her işlemci herhangi bir anda sadece bir işlemi yürütebilmektedir. İşlemciler arıza yapmazlar. Bütün işler çizelgeleme başında hazırdırlar. İşler kesintisiz işlenir. Amaç, En Geç Tamamlanma Zamanı (C_{max})'nın en küçüklenmesidir.

Carlier ve Neron'un (2000) EATÇ Problemleri ve Oğuz ve Ercan'ın (2005) ÇİEATÇ problemlerini çözmek için geliştirilen PDA'da 5 kontrol parametresi kullanılmaktadır. Bu parametreler, Tablo 3.5'de sunulmuştur.

Tablo 3.2 PDA Programında Kullanılan Parametreler

Başlangıç Popülasyonu	
İterasyon Sayısı	
Alt Grup Sayısı (min. 2, maks. n-1)	Otomatik Değişen Alt Grup Sayısı
	Kırılma Yöntemiyle Değişen Alt Grup Sayısı
	Logaritmik Değişen Alt Grup Sayısı
İnşa Yöntemi	Birer Atlayarak İşleri Yerleştir
	İkişer Atlayarak İşleri Yerleştir
	Öne Yerleştir
	Sona Yerleştir
Doyum Oranı	0,1' den 0,9' a kadar 0,1'er artış

Alt grup sayısı; en az *2*, en fazla *n-1* değeri olarak tanımlanmış ve popülasyonun daha iyi olması yönünde değişiklik göstermektedir. Alt grup sayısı; oluşturulan programda üç alternatif yolla değişmektedir. Bu yollardan otomatik değişen alt grup sayısı yolu seçildiği takdirde, değişim basamağı değeri girilir. Programın çalışması esnasında değişim basamağı kadar iterasyon ilerledikten sonra alt grup sayısı, girilen adım değeri kadar otomatik bir şekilde değişir. 2. yol olan kırılma yöntemi seçildiği takdirde kırılma basamağı değeri girilir ve programın çalışması esnasında kırılma basamağı kadar iterasyon ilerledikçe alt grup sayısı, girilen adım değeri kadar otomatik bir şekilde artmaktadır. 3. yol olan logaritmik değişim yöntemi seçildiğinde ise alt grup sayısı logaritmik bir şekilde değişim gösterir.

İterasyon sayısı; kadar program alt popülasyonları çalıştırır.

Doyum oranı; 0,1 ile 0,9 arasında bir değerdir. Bu oran popülasyonun daha iyi olması yönünde 0,1'er değer artarak değişiklik göstermektedir.

İnşa işlemi; 4 alternatif yol ile gerçekleştirilmektedir.

- *Birer atlayarak işleri yerleştirme (BAİY) inşa yöntemi:* Bu yöntemin işleyişi, yıkım aşamasında çıkartılan işlerin, yıkımdan sonra geriye kalan iş sırasına birer atlayarak yerleşmesidir.
- *İkişer atlayarak işleri yerleştirme (İAİY) inşa yöntemi:* Bu yöntemin işleyişi, yıkım aşamasında çıkartılan işlerin, yıkımdan sonra geriye kalan iş sırasına ikişer atlayarak yerleşmesidir.
- *Öne yerleştirme (ÖY) inşa yöntemi:* Bu yöntemin işleyişi, yıkım aşamasında çıkartılan işlerin, yıkımdan sonra geriye kalan iş sırasının önüne eklenerek yerleşmesidir.
- *Sona yerleştirme (SY) inşa yöntemi:* Bu yöntemin işleyişi, yıkım aşamasında çıkartılan işlerin, yıkımdan sonra geriye kalan iş sırasının sonuna eklenerek yerleşmesidir.

Çalışmada kullanılan parametreler, bütün seviyeler için test edilmiş olup en iyi parametre seti belirlenmiştir.

Carlier ve Neron'un (2000) EATÇ Problemleri için belirlenen kontrol parametre seti değerleri Tablo 3.3'de sunulmuştur.

Tablo 3.3 EATÇ Problemleri İçin Kullanılan Parametre Değerleri

Kontrol Parametreleri	**Kontrol Parametre Değerleri**
Başlangıç Popülasyonları	15 ; 30
Alt Grup Sayıları	2 ; 3 ; 4 ; 5 ; 6 ; 7 ; 8 ; 9
İterasyon Sayıları	50 ; 100 ; 150 ; 200 ; 250 ; 500 ; 1000 ; 2500
Doyum Oranları	0,1 ; 0,2 ; 0,3 ; 0,4 ; 0,5 ; 0,6 ; 0,7 ; 0,8 ; 0,9
İnşa Yöntemleri	BAİY ; İAİY ; ÖY ; SY

Oğuz ve Ercan'ın (2005) ÇİEATÇ Problemleri için denenen belirlenen kontrol parametre seti değerleri Tablo 3.4'de sunulmuştur.

Tablo 3.4 ÇİEATÇ Problemleri İçin Kullanılan Parametre Değerleri

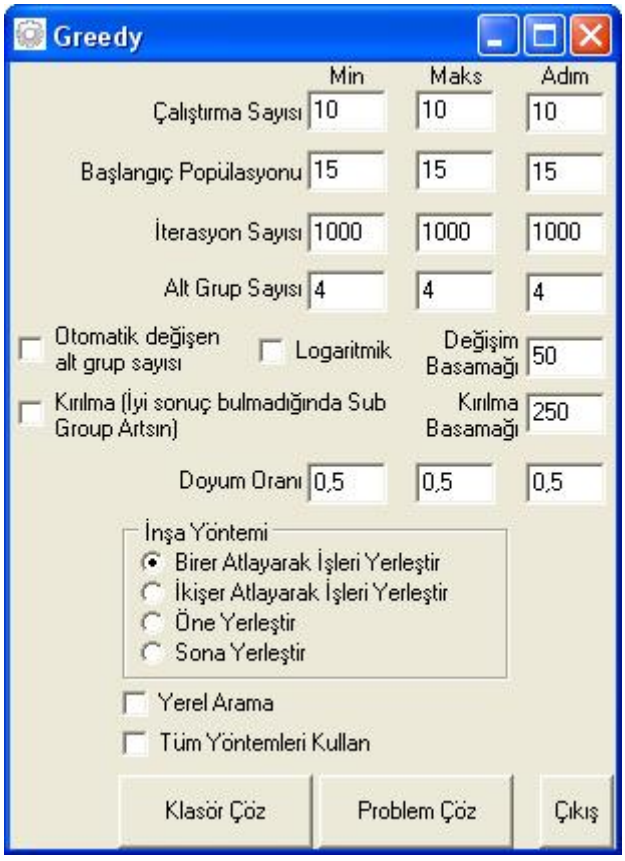
Kontrol Parametreleri	**Kontrol Parametre Değerleri**
Başlangıç Popülasyonları	15 ; 30
Alt Grup Sayıları	2 ; 3 ; 4 ; 5 ; 6 ; 7 ; 8 ; 9
İterasyon Sayıları	50 ; 100 ; 150 ; 200 ; 250 ; 500 ; 1000
Doyum Oranları	0,1; 0,2 ; 0,3 ; 0,4 ; 0,5 ; 0,6 ; 0,7 ; 0,8 ; 0,9
İnşa Yöntemleri	BAİY ; İAİY ; ÖY ; SY

Çözüm sonuçları için en önemli faktör C_{max} değeridir. C_{max} değerinden sonra dikkate alınan değer, İşlem Süresi (İS)'dir. İşlem süreleri, eşit ise iterasyon sayıları dikkate alınmaktadır. Carlier ve Neron'un (2000) tüm EATÇ problemleri için PDA metodu ile elde edilen en iyi çözümler ve parametre değerleri EK–1 'de, Oğuz ve Ercan'ın (2005) tüm ÇİEATÇ Problemleri için PDA metodu ile elde edilen en iyi çözümlerin işlem süreleri ve parametre değerleri EK–2 ve EK–3 'de sunulmuştur.

4. ARAŞTIRMA SONUÇLARI

4.1. Veri Toplama

EATÇ ve ÇİEATÇ problemlerinin, PDA ile çözümü için Borland Delphi 7.0 dilinde program hazırlandı. Programın ekran görüntüsü Şekil 4.1' de görülmektedir.



Şekil 4.1 PDA Programına Ait Ekran Görüntüsü

Test edilen esnek akış tipi problemler; Carlier ve Neron (2000) ile Oğuz ve Ercan (2005)'ın esnek akış tipi çizelgeleme problemleri olup Intel Pentium 4.3 GHz işlemcili, 512 Mb ram ve Microsoft Windows XP with SP1 işletim sistemi bulunan bir bilgisayarda hazırlanan program yardımıyla çözülmüştür.

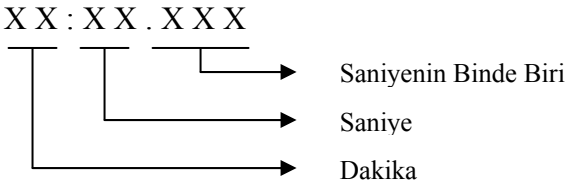
Yapılan çalışmada, Carlier ve Neron (2000) tarafından önerilen kıyaslama problemleri için PDA metodu ile elde edilen çözüm değerleri; literatürdeki, Döyen (2004)'in Yapay Bağışıklık Sistemleri (YBS) ve Neron (2001)'un Dal Sınır Algoritması (DSA) metodları ile elde edilen çözüm değerleri ile karşılaştırılmıştır. Oğuz ve Ercan (2005) tarafından önerilen kıyaslama problemleri (2004–2005) için PDA ile elde edilen sonuçlar, Oğuz (2005) 'un GA sonuçları ve Ceran (2006)'ın GA sonuçları ile karşılaştırılmıştır.

Yapılan analizlerde, PDA'nın çözüm kalitesi ve çözüme ulaşmada harcadığı süre ölçülmüştür. Çözüm kalitesi, PDA'nın elde ettiği en iyi çözüm (C_{max}) ile problemin alt sınır değeri arasındaki yüzde sapma miktarı ile ölçülmektedir. Yüzde Sapma Miktarı (1)'deki ifadeye göre hesaplanır.

$$\%\ Sapma\ Miktarı = \frac{PDA\ Çözümü\ (C_{max}) - Alt\ Sınır\ Değeri\ (ASD)}{Alt\ Sınır\ Değeri\ (ASD)}\ x\ 100 \quad (1)$$

4.2. EATÇ Problemlerinin Çözümlerinin Analizi

Carlier ve Neron'un (2000) tüm EATÇ Problemleri için geliştirilen PDA programı, en fazla 1600 sn. çalışmıştır. Elde edilen çözümlerin C_{max} değerleri ve işlem süreleri karşılaştırmalı olarak Tablo 4.1, 4.2'de sunulmuştur. Tablolarda sunulan PDA' ya ait işlem sürelerinin açıklaması (2)'de verilmiştir. YBS ve DSA 'ya ait olan işlem süreleri ise "saniye" cinsindendir. Tablolarda yer alan PDA, YBS ve DSA sonuçlarının ASD'den daha kötü olduğu sonuçlar koyu ile işaretlenmiştir.



(2)

Tablo 4.1 EATÇ 10x5 Tipi Problemler İçin Kıyaslama

Problem	PDA C_{max}	PDA İşlem Süreleri	YBS C_{max}	YBS İşlem Süreleri	DSA C_{max}	DSA İşlem Süreleri	ASD C_{max}	PDA % Sapma	YBS % Sapma	DSA % Sapma
j10c5a2	88	00:00.000	88	1	88	13	88	0	0	0
j10c5a3	117	00:00.000	117	1	117	7	117	0	0	0
j10c5a4	121	00:00.015	121	1	121	6	121	0	0	0
j10c5a5	122	00:00.000	122	1	122	11	122	0	0	0
j10c5a6	110	00:00.063	110	4	110	6	110	0	0	0
j10c5b1	130	00:00.000	130	1	130	13	130	0	0	0
j10c5b2	107	00:00.000	107	1	107	6	107	0	0	0
j10c5b3	109	00:00.000	109	1	109	9	109	0	0	0
j10c5b4	122	00:00.000	122	2	122	6	122	0	0	0
j10c5b5	153	00:00.000	153	1	153	6	153	0	0	0
j10c5b6	115	00:00.000	115	1	115	11	115	0	0	0
j10c5c1	68	00:00.016	68	32	68	28	68	0	0	0
j10c5c2	74	00:00.140	74	4	74	19	74	0	0	0
j10c5c3	**72**	**00:00.000**	**72**	**a**	71	240	71	1.4	1.4	0
j10c5c4	66	00:00.265	66	3	66	1017	66	0	0	0
j10c5c5	78	00:00.750	78	14	78	42	78	0	0	0
j10c5c6	69	00:00.000	69	12	69	4865 [b]	69	0	0	0
j10c5d1	66	00:02.140	66	5	66	6490 [b]	66	0	0	0
j10c5d2	73	00:00.000	73	31	73	2617 [b]	73	0	0	0
j10c5d3	64	00:00.016	64	15	64	481	64	0	0	0
j10c5d4	70	00:00.000	70	5	70	393	70	0	0	0
j10c5d5	66	00:00.000	66	1446	66	1627 [b]	66	0	0	0
j10c5d6	62	00:00.203	62	8	62	6861 [b]	62	0	0	0

a: PDA ve YBS, 1600 sn içinde ASD'ye ulaşamamıştır.
b: DSA, 1600 sn'den daha fazla sürede ASD'ye ulaşmıştır.

Tablo 4.2 EATÇ 10x10 ; 15x5 ; 15x10 Tipi Problemler İçin Kıyaslama

Problem	PDA C_{max}	PDA İşlem Süreleri	YBS C_{max}	YBS İşlem Süreleri	DSA C_{max}	DSA İşlem Süreleri	ASD C_{max}	PDA % Sapma	YBS % Sapma	DSA % Sapma
j10c10a1	139	00:00.020	139	1	139	41	139	0	0	0
j10c10a2	158	00:00.015	158	18	158	21	158	0	0	0
j10c10a3	148	00:00.109	148	1	148	58	148	0	0	0
j10c10a4	149	00:00.000	149	2	149	21	149	0	0	0
j10c10a5	148	00:00.000	148	1	148	36	148	0	0	0
j10c10a6	146	00:00.015	146	4	146	20	146	0	0	0
j10c10b1	163	00:00.000	163	1	163	36	163	0	0	0
j10c10b2	157	00:00.000	157	1	157	66	157	0	0	0
j10c10b3	169	00:00.000	169	1	169	19	169	0	0	0
j10c10b4	159	00:00.000	159	1	159	20	159	0	0	0
j10c10b5	165	00:00.000	165	1	165	33	165	0	0	0
j10c10b6	165	00:00.000	165	1	165	34	165	0	0	0
j10c10c1	**115**	**a**	**115**	**a**	**127**	**c**	113	1.8	1.8	12.4
j10c10c2	**117**	**a**	**119**	**a**	116	1100	116	0.86	2.6	0
j10c10c3	**116**	**a**	**116**	**a**	**133**	**c**	98	18.4	18.4	35.7
j10c10c4	**120**	**a**	**120**	**a**	**135**	**c**	103	16,5	16.5	31.1
j10c10c5	**126**	**a**	**126**	**a**	**145**	**c**	121	4.1	4.1	19.8
j10c10c6	**106**	**a**	**106**	**a**	**112**	**c**	97	9,3	9.3	15.5
j15c5a1	178	00:00.050	178	1	178	18	178	0	0	0
j15c5a2	165	00:00.000	165	1	165	35	165	0	0	0
j15c5a3	130	00:00.000	130	1	130	34	130	0	0	0
j15c5a4	156	00:00.000	156	2	156	21	156	0	0	0
j15c5a5	164	00:00.000	164	1	164	34	164	0	0	0
j15c5a6	178	00:00.000	178	1	178	38	178	0	0	0
j15c5b1	170	00:00.000	170	1	170	16	170	0	0	0
j15c5b2	152	00:00.000	152	1	152	25	152	0	0	0
j15c5b3	157	00:00.000	157	1	157	15	157	0	0	0
j15c5b4	147	00:00.000	147	1	147	37	147	0	0	0
j15c5b5	166	00:00.000	166	2	166	20	166	0	0	0
J15c5b6	175	00:00.000	175	1	175	23	175	0	0	0
j15c5c1	85	00:00.000	85	774	85	2131[b]	85	0	0	0
j15c5c2	**91**	**a**	**91**	**a**	90	184	90	1.1	1.1	0
j15c5c3	87	00:00.329	87	16	87	202	87	0	0	0
j15c5c4	89	00:00.000	89	317	**90**	**c**	89	0	0	1.1
j15c5c5	**75**	**a**	**74**	**a**	**84**	**c**	73	2.7	1.4	15.1
j15c5c6	91	00:00.438	91	19	91	57	91	0	0	0
j15c5d1	167	00:00.000	167	1	167	24	167	0	0	0
j15c5d2	**84**	**a**	**84**	**a**	**85**	**c**	82	2.4	2.4	3.7
j15c5d3	**83**	**a**	**83**	**a**	**96**	**c**	77	7.8	7.8	24.7
j15c5d4	**84**	**a**	**84**	**a**	**101**	**c**	61	37.7	37.7	65.6
j15c5d5	**80**	**a**	**80**	**a**	**97**	**c**	67	19.4	19.4	44.8
j15c5d6	**81**	**a**	**82**	**a**	**87**	**c**	79	2.5	3.8	10.1
j15c10a1	236	00:00.000	236	1	236	40	236	0	0	0
j15c10a2	200	00:00.828	200	30	200	154	200	0	0	0
j15c10a3	198	00:00.016	198	4	198	45	198	0	0	0
j15c10a4	225	00:00.015	225	12	225	78	225	0	0	0
j15c10a5	182	00:00.172	182	2	**183**	**c**	182	0	0	0.5
j15c10a6	200	00:00.000	200	2	200	44	200	0	0	0
j15c10b1	222	00:00.000	222	3	222	70	222	0	0	0
j15c10b2	187	00:00.062	187	1	187	80	187	0	0	0
j15c10b3	222	00:00.000	222	1	222	80	222	0	0	0
j15c10b4	221	00:00.000	221	1	221	84	221	0	0	0
j15c10b5	200	00:00.000	200	1	200	84	200	0	0	0
j15c10b6	219	00:00.000	219	1	219	67	219	0	0	0

a: PDA ve YBS, 1600 sn içinde ASD'ye ulaşamamıştır.
b: DSA, 1600 sn'den daha fazla sürede ASD'ye ulaşmıştır.
c: DSA, 1600 sn'den daha fazla sürede ASD'ye ulaşamamıştır.

Carlier ve Neron (2001); 10x5 ve 15x5 problemlerinin *c* ve *d* tiplerini (toplam 24 problem) zor problemler, 10x5, 10x10 ve 15x5 problemlerinin *a* ve *b* tiplerini ve 10x10 problemlerinin *c* tipini kolay problemler (toplam 53 problem) olarak sınıflandırmışlardır.

Zor problemler yerel optimum noktaları sayısı ve olurlu çözüm sayısı göz önüne alınırsa daha karmaşık problemlerdir. Bu nedenle herhangi bir sezgisel arama tekniğinin lokal optimuma takılıp kalma şansı daha yüksektir (Döyen 2004).

PDA ve YBS metodu ile çözümleme sonucunda kolay problemlerden 47 adedi, ASD (Alt Sınır Değeri)'ye ulaşmıştır. DSA metodu ile çözümleme sonucunda kolay problemlerin 46 adedi, ASD'ye ulaşmıştır.

PDA ile çözümleme sonucunda 15 adet, YBS ile çözümleme sonucunda 16 adet ve DSA ile çözümleme sonucunda 17 adet zor problemin çözümleri, ASD'ye ulaşmıştır.

PDA metodu, kolay problemler için YBS ve DSA yöntemlerine göre daha etkin olduğu görülmüştür. Fakat PDA metodu, zor problemler için YBS ve DSA yöntemlerine göre daha az etkin bir yöntem olduğu belirlenmiştir.

Dikkat edileceği üzere, PDA yöntemi ile çözülen *a* ve *b* tipi problemlerde ASD ile aynı çözümler elde edilirken, *c* ve *d* tipi problemlerde daha çok sapma görülmektedir. Makine yerleşimi yapısı problemin zorluğu üzerinde önemli bir etkiye sahiptir. Problemin büyüklüğü ise PDA' yı etkilememektedir.

Problemler için bulunan alt sınır değerleri, ASD'den çok daha düşük olabilirler. Bulunan bu değerler belki de ASD'dir. Bunlardan daha iyi bir çözüm yoktur.

Tablo 4.1'e göre; PDA metodu ile, 10 iş x 5 kademe boyutlu problemler için; *a, b, c, d* yapılarında toplam 23 problem çözülmüştür. Toplam 22 problemin hepsi için alt sınır değerler (optimal sonuçlar) elde edilmiştir. Sadece 1 problem (*j10c5c3*) için optimal sonuç bulunamamıştır. YBS ile de sadece aynı problem için ASD'ye ulaşılamamıştır. DSA yöntemi ile 23 problemin hepsi için de ASD'ye ulaşılmıştır.

Tablo 4.2'ye göre; PDA metodu ile, 10 iş x 10 kademe boyutlu problemler için *a, b, c* yapılarında toplam 18 problem çözülmüştür. PDA, YBS ve DSA ile

yapılan çözümlemelerde *a* ve *b* türünde 12 problem için optimal sonuçları elde edilmiştir. *c* türündeki problemler için PDA ve YBS ile bulunan sonuçlar aynıdır. Sadece bir problem için (*j10c10c2*) PDA, YBS'den daha iyi sonuç elde edilmiştir. Fakat *c* türündeki problemler için hem PDA hem de YBS ile çözümlerde ASD'ye ulaşılamamıştır. DSA ile çözümde *c* türündeki problemlerde sadece bir problem (*j10c10c2*) için ASD elde edilmiştir.

Tablo 4.2'ye göre; PDA metodu ile, 15 iş x 5 kademe boyutlu problemler için *a, b, c, d* yapılarında toplam 24 problem çözülmüştür. Hem PDA metodu hem de YBS metodu da 6 problem için ASD'ye ulaşamamıştır. DSA yönteminin bulduğu çözümler sonucunda 7 problem de ASD'ye ulaşılamamıştır. *'j15c5c5'* problemi için PDA metodu YBS metodundan daha kötü bir değer bulmuştur. Bunun yanında *'j15c5d6'* problemi için PDA metodu YBS metodundan daha iyi bir değer bulmuştur. *'j15c5c2'* problemi için PDA metodu DSA metodundan daha kötü bir değer bulmuştur. Bunun yanında *'j15c5c4, j15c5c5, j15c5d2, j15c5d3, j15c5d4, j15c5d5, j15c5d6'* problemleri için PDA metodu DSA metodundan daha iyi bir değer bulmuştur.

Tablo 4.2'ye göre; PDA metodu ile, 15 iş x 10 kademe boyutlu problemlerde, *a* ve *b* tipi (toplam 12 problem) tüm problemlerde PDA ve YBS yöntemi ile yapılan çözümlemelerde optimal sonuçlara ulaşılmıştır. DSA yöntemi ile çözümde *a* tipi problemlerden *'j15c10a5'* probleminin sonucu ASD'den daha kötü sonuç çıkmıştır.

77 problemin tamamı göz önüne alındığında, YBS metodunun ASD'den sapması % 1.657 ve DSA metodunun sapması % 3.6 iken PDA'nın, ASD'den sapması % 1.50 olduğu görülmüştür.

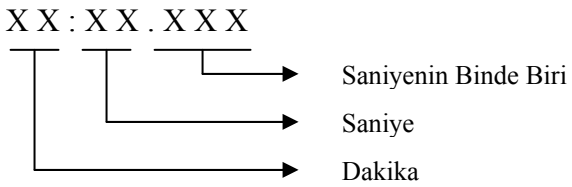
PDA'nın; kolay problemler için ASD'den ortalama yüzde sapma miktarı % 0,99 , zor problemler için ASD'den ortalama sapması % 3,10 olarak tespit edilmiştir. Tablo 4.3'de görüldüğü gibi PDA metodu, Döyen (2004)'in YBS metoduna göre çözüm yüzdesi bakımından aynı, sapma miktarı bakımından ise daha iyi bir performans gösterirken; Neron (2001)'un DSA metoduna göre çözüm yüzdesi bakımından kolay problemler için aynı, zor problemler için daha az, sapma miktarı bakımından ise her iki tip problem için de daha iyi bir performans göstermektedir.

Tablo 4.3 PDA, YBS ve DSA Sezgisel Metodların Performansları

Metod	Kolay Problemler		Zor Problemler	
	% Çözüm	% Sapma	% Çözüm	% Sapma
PDA	88.7	0.99	66.7	3,10
YBS	88.7	0.99	66.7	3,13
DSA	88.7	2.17	70.8	6,88

4.3. ÇİEATÇ Problemlerinin Çözümlerinin Analizi

Çalışmada, ÇİEATÇ problemleri olarak Oğuz ve Ercan (2005)'ın kıyaslama problemleri çözülmüştür. PDA metodu ile elde edilen çözümler ve Oğuz (2005)'un GA ve Ceran (2006)'ın GA yaklaşımları ile elde ettikleri çözümler, karşılaştırmalı olarak Tablo 4.4, 4.5, 4.6, 4.7, 4.8 ve 4.9'da sunulmuştur. Tablolarda; elde edilen çözüm değerlerinin, ASD'den sapmaları ve çalışmada PDA metodu ile elde edilen çözümlerin işlem süreleri sunulmuş olup bu işlem sürelerinin açıklaması (3)'de verilmiştir. Tablolarda sunulan PDA, Oğuz (2005)'un GA ve Ceran (2006)'ın GA metodları ile elde edilmiş çözümlerin ASD'ye ulaşamamış çözüme sahip problemlerin yüzde sapma miktarları, koyu ile işaretlenmiştir.



(3)

Tablo 4.4 ÇİEATÇ *P* Tipi 2 Aşamalı Problemler İçin Kıyaslama

Problem	OĞUZ GA	CERAN GA	PDA	
	% Sapma	% Sapma	% Sapma	İşlem Süreleri
P10S2T01	0.000	0.000	0.000	00:00.062
P10S2T02	0.000	0.000	0.000	00:00.016
P10S2T03	**5.960**	**5.960**	**5.960**	00:01.406
P10S2T04	0.000	0.000	0.000	00:00.000
P10S2T05	0.000	0.000	0.000	00:00.000
P10S2T06	0.000	0.000	0.000	00:00.000
P10S2T07	0.000	0.000	0.000	00:00.000
P10S2T08	0.000	0.000	0.000	00:00.000
P10S2T09	0.000	0.000	0.000	00:00.000
P10S2T10	0.000	0.000	0.000	00:00.000
P20S2T01	**1.997**	**2.611**	**1.690**	00:00.015
P20S2T02	**2.368**	**1.639**	**1.275**	00:02.516
P20S2T03	0.000	0.000	0.000	00:00.000
P20S2T04	0.000	0.000	0.000	00:00.000
P20S2T05	0.000	0.000	0.000	00:00.000
P20S2T06	0.000	0.000	0.000	00:00.063
P20S2T07	**0.341**	**0.341**	**0.341**	00:00.765
P20S2T08	**0.278**	**0.278**	**0.278**	00:00.968
P20S2T09	0.000	0.000	0.000	00:00.000
P20S2T10	0.000	0.000	0.000	00:00.484
P50S2T01	**1.753**	**0.605**	**1.088**	00:13.265
P50S2T02	0.000	0.000	0.000	00:00.000
P50S2T03	**0.590**	0.000	0.000	00:00.125
P50S2T04	**1.238**	**0.681**	**1.423**	01:39.297
P50S2T05	**0.353**	**0.235**	**0.588**	00:00.078
P50S2T06	0.000	0.000	0.000	00:00.016
P50S2T07	**2.716**	**0.692**	**2.130**	00:47.172
P50S2T08	0.000	0.000	0.000	00:00.015
P50S2T09	0.000	0.000	0.000	00:00.047
P50S2T10	**0.262**	**0.052**	**0.052**	00:03.891
P1HS2T01	**0.534**	**0.721**	**0.507**	02:55.437
P1HS2T02	0.000	0.000	0.000	00:00.016
P1HS2T03	**0.814**	**0.693**	**0.603**	01:08.890
P1HS2T04	0.000	0.000	0.000	00:00.172
P1HS2T05	0.000	0.000	0.000	00:00.156
P1HS2T06	0.000	0.000	0.000	00:00.031
P1HS2T07	**0.926**	**0.657**	**0.538**	02:03.328
P1HS2T08	**0.020**	0.000	0.000	00:00.172
P1HS2T09	**0.105**	**0.683**	**0.394**	00:00.188
P1HS2T10	**0.097**	**0.242**	**0.242**	01:47.734

Tablo 4.4'e göre; PDA yöntemi ile yapılan çözümleme sonucunda, *P10S2* tipi Oğuz ve Ercan (2005) 'ın kıyaslama problemlerinden 9 tanesinin çözüm değerleri ASD'ye ulaşmıştır. ASD'ye ulaşamayan *P10S2T03* probleminin çözüm değeri, Oğuz (2005)'un ve Ceran (2006)'ın elde ettikleri çözüm değeri ile aynıdır. Bu gruptaki Oğuz (2005)'un ve Ceran (2006)'ın elde ettikleri çözüm değerlerinden 9'ar tanesi de ASD'ye ulaşmıştır.

Tablo 4.4'e göre; PDA yöntemi ile yapılan çözümleme sonucunda, *P20S2* tipi Oğuz ve Ercan (2005) 'ın kıyaslama problemlerinden 6 tanesinde ASD'ye ulaşılmıştır. ASD'ye ulaşamayan *P20S2T01, P20S2T02* problemlerinin çözüm değerleri, Oğuz (2005)'un ve Ceran (2006)'ın elde ettikleri çözüm değerlerinden daha iyidir. ASD'ye ulaşamayan *P20S2T07, P20S2T08* problemlerinin çözüm değerleri, Oğuz (2005) ve Ceran (2006)'ın elde ettikleri çözüm değerleri ile aynıdır. Bu gruptaki Oğuz (2005)'un ve Ceran (2006)'ın elde ettikleri çözüm değerlerinden 6'şar tanesi ASD'ye ulaşmıştır.

Tablo 4.4'e göre; PDA yöntemi ile yapılan çözümleme sonucunda, *P50S2* tipi Oğuz ve Ercan (2005) 'ın kıyaslama problemlerinden 5 tanesinde ASD'ye ulaşılmıştır. ASD'ye ulaşamamış 5 problem çözümünden 3 tanesinin çözüm değeri, Oğuz (2005)'un çözümlerinden daha iyi; 2 tanesinin çözüm değeri ise daha kötüdür. ASD'ye ulaşamamış 5 adet problemlerden 1 tanesinin çözüm değeri Ceran (2006) 'ın elde ettiği çözüm ile aynı; 4 tanesinin çözüm değerleri, Ceran (2006)'ın elde ettiği değerlerden daha kötüdür. Bu gruptaki Oğuz (2005)'un elde ettiği çözüm değerlerinden sadece 4 tanesi ve Ceran (2006)'ın elde ettiği çözüm değerlerinden 5 tanesi ASD'ye ulaşmıştır.

Tablo 4.4'e göre; PDA yöntemi ile yapılan çözümleme sonucunda, *P1HS2* tipi Oğuz ve Ercan (2005) 'ın kıyaslama problemlerinden 5 tanesinde ASD'ye ulaşılmıştır. ASD'ye ulaşamamış 5 problem çözümünden 3 problemin çözüm değerleri, Oğuz (2005)'un elde ettiği çözüm değerlerinden daha iyi; 2 problemin çözüm değerleri daha kötü; 3 problemin çözüm değerleri, Ceran (2006) 'ın elde ettiği çözüm değerlerinden daha iyi; 2 problemin çözüm değerleri ile aynı değerdedir. Bu gruptaki Oğuz (2005)'un elde ettiği çözüm değerlerinden sadece 4 tanesi ve Ceran (2006) 'ın elde ettiği çözüm değerlerinden 5 tanesi ASD'ye ulaşmıştır.

Tablo 4.5 ÇİEATÇ *P* tipi 5 aşamalı problemler için kıyaslama

Problem	OĞUZ GA	CERAN GA	PDA	
	% Sapma	% Sapma	% Sapma	İşlem Süreleri
P10S5T01	**4.530**	**4.878**	**4.878**	00:04.844
P10S5T02	**6.022**	**6.022**	**6.022**	00:00.000
P10S5T03	**7.540**	**7.540**	**7.540**	00:01.984
P10S5T04	**8.609**	**11.755**	**11.755**	00:01.469
P10S5T05	**3.684**	**3.684**	**3.684**	00:00.360
P10S5T06	0.000	0.000	0.000	00:00.000
P10S5T07	**8.210**	**8.210**	**8.210**	00:02.063
P10S5T08	**2.742**	**2.258**	**2.258**	00:03.516
P10S5T09	**9.076**	**10.191**	**9.076**	00:00.040
P10S5T10	**10.417**	**12.660**	**12.179**	00:00.000
P20S5T01	**1.567**	**1.567**	**1.567**	00:01.610
P20S5T02	**7.165**	**7.165**	**6.858**	00:04.188
P20S5T03	**2.561**	**3.492**	**6.170**	00:00.062
P20S5T04	**2.268**	0.000	0.000	00:00.047
P20S5T05	**2.313**	**1.682**	**3.049**	00:00.031
P20S5T06	**2.650**	**1.880**	**2.479**	00:00.047
P20S5T07	0.000	0.000	0.000	00:01.891
P20S5T08	**3.519**	**3.128**	**3.519**	00:00.000
P20S5T09	0.000	0.000	0.000	00:00.000
P20S5T10	**7.165**	**6.858**	**7.165**	00:44.784
P50S5T01	**0.911**	**0.911**	**0.835**	02:05.547
P50S5T02	**4.539**	**1.099**	**0.532**	00:33.921
P50S5T03	**0.998**	**0.998**	**0.998**	00:44.484
P50S5T04	**0.618**	**2.523**	**4.325**	00:00.704
P50S5T05	**2.577**	0.000	0.000	00:04.781
P50S5T06	0.000	0.000	0.000	00:01.734
P50S5T07	**1.100**	**0.513**	**0.477**	01:47.234
P50S5T08	**2.447**	**1.099**	**0.745**	00:34.812
P50S5T09	**5.096**	0.000	0.000	00:57.703
P50S5T10	**0.271**	**0.587**	**1.264**	01:07.375
P1HS5T01	**3.493**	0.000	0.000	00:00.594
P1HS5T02	**1.543**	0.000	0.000	00:13.265
P1HS5T03	**3.819**	**0.674**	**2.272**	01:10.688
P1HS5T04	**1.425**	**1.130**	**1.548**	00:00.156
P1HS5T05	**2.347**	0.000	0.000	00:11.531
P1HS5T06	**3.591**	**2.924**	**4.488**	00:00.266
P1HS5T07	**0.300**	0.000	0.000	00:16.094
P1HS5T08	**0.673**	0.000	0.000	00:17.469
P1HS5T09	**1.379**	0.000	0.000	01:31.640
P1HS5T10	**3.248**	**2.994**	**3.389**	00:00.375

Tablo 4.5'e göre; PDA yöntemi ile yapılan çözümleme sonucunda, *P10S5* tipi Oğuz ve Ercan (2005) 'ın kıyaslama problemlerinden *P10S5T06* problem çözümünde ASD'ye ulaşılmıştır. *P10S5T08* probleminin çözümünde ASD'ye ulaşılamamasına rağmen Oğuz (2005)'un elde ettiği çözüm değerinden daha iyi bir değer elde edilmiştir. *P10S5T09, P10S5T10* problemlerinin çözümünde ASD'ye ulaşılamamasına rağmen Ceran (2006)'ın elde ettiği çözüm değerlerinden daha iyi değerler elde edilmiştir. Problemlerden 5 tanesinde ise ASD'ye ulaşılamamasına

rağmen Oğuz (2005)'un elde ettiği çözüm değerleri ile aynı değerler elde edilmiştir. Problemlerden 7 tanesinde ise ASD'ye ulaşılamamasına rağmen Ceran (2006)'ın elde ettiği çözüm değerleri ile aynı değerler elde edilmiştir. 3 problem çözümünde ASD'ye ulaşılamadığı gibi Oğuz (2005)'un elde ettiği çözüm değerlerinden daha kötü değerler elde edilmiştir. Bu gruptaki Oğuz (2005)'un ve Ceran (2006)'ın elde ettikleri çözümlerden 1'er tanesi ASD'ye ulaşmıştır.

Tablo 4.5'e göre; PDA yöntemi ile yapılan çözümleme sonucunda, *P20S5* tipi Oğuz ve Ercan (2005) 'ın kıyaslama problemlerinden 3 problem çözümünde ASD'ye ulaşılmıştır. ASD'ye ulaşamamış 7 problemden 3'ünün çözümünde Oğuz (2005)'un elde ettiği değerlerden daha iyi; 3 tanesinde aynı; 1 tanesinde (*P20S5T03*) ise daha kötü bir değer elde edilmiştir. ASD'ye ulaşamamış 7 problemden 1'inin çözümünde Ceran (2006)'ın elde ettiği değerlerden daha iyi; 1 tanesinde aynı; 5 tanesinde ise daha kötü değerler elde edilmiştir. Bu gruptaki Oğuz (2005)'un elde ettiği çözümlerden 2 tanesi ve Ceran (2006)'ın elde ettiği çözümlerden 3 tanesi ASD'ye ulaşmıştır.

Tablo 4.5'e göre; PDA yöntemi ile yapılan çözümleme sonucunda, *P50S5* tipi Oğuz ve Ercan (2005) 'ın kıyaslama problemlerinden 3 problem çözümünde ASD'ye ulaşılmıştır. ASD'ye ulaşamamış 7 problemden 4'ünün çözüm değeri, Oğuz (2005)'un elde ettiği değerlerden daha iyi; 1 tanesinde aynı; 2 tanesinde kötüdür. ASD'ye ulaşamamış 7 problemden 4'ünün çözüm değeri, Ceran (2006)'ın elde ettiği değerlerden daha iyi; 1 tanesinde aynı; 2 tanesinde daha kötüdür. Bu gruptaki Oğuz (2005)'un elde ettiği çözüm değerlerinden 1 tanesi ve Ceran (2006)'ın elde ettiği çözüm değerlerinden 3 tanesi ASD'ye ulaşmıştır.

Tablo 4.5'e göre; PDA yöntemi ile yapılan çözümleme sonucunda, *P1HS5* tipi Oğuz ve Ercan (2005) 'ın kıyaslama problemlerinden 6 problem çözümünde ASD'ye ulaşılmıştır. ASD'ye ulaşamamış 4 problemden 1'inin çözüm değeri, Oğuz (2005)'un elde etmiş olduğu değerlerden daha iyi; 3 tanesinde kötüdür. ASD'ye ulaşamamış 4 problemden 4'ünün çözüm değeri de Ceran (2006)'ın elde ettiği değerlerden daha kötüdür. Bu gruptaki Ceran (2006)'ın elde ettiği değerlerden 6 tanesi ASD'ye ulaşmıştır. Fakat Oğuz (2005)'un elde etmiş olduğu değerlerden hiçbiri ASD'ye ulaşamamıştır.

Tablo 4.6 ÇİEATÇ *P* tipi 8 aşamalı problemler için kıyaslama

Problem	OĞUZ GA	CERAN GA	PDA	
	% Sapma	% Sapma	% Sapma	İşlem Süreleri
P10S8T01	**21.268**	**23.662**	**23.662**	00:58.907
P10S8T02	**26.179**	**31.870**	**28.455**	00:00.000
P10S8T03	**20.027**	**21.786**	**21.786**	00:31.345
P10S8T04	**13.091**	**17.350**	**14.038**	00:00.000
P10S8T05	**1.902**	**4.212**	**4.212**	00:01.392
P10S8T06	**0.409**	**0.409**	**0.409**	00:00.000
P10S8T07	**24.757**	**30.097**	**26.214**	00:00.000
P10S8T08	**19.479**	**24.540**	**21.933**	00:00.000
P10S8T09	**0.830**	**2.075**	**2.075**	00:00.000
P10S8T10	**19.589**	**20.000**	**20.000**	00:28.562
P20S8T01	**8.163**	**9.599**	**9.448**	00:00.125
P20S8T02	0.000	0.000	0.000	00:10.813
P20S8T03	**2.573**	**2.744**	**4.117**	01:21.219
P20S8T04	**1.593**	**7.080**	**3.805**	00:00.188
P20S8T05	**3.152**	**2.300**	**2.641**	00:00.234
P20S8T06	**8.163**	**4.913**	**9.675**	00:00.141
P20S8T07	**23.795**	**28.821**	**27.487**	00:00.235
P20S8T08	**3.791**	**3.791**	**2.729**	00:37.093
P20S8T09	0.000	0.000	0.000	00:04.828
P20S8T10	**4.117**	**3.945**	**4.803**	00:00.031
P50S8T01	**2.709**	**5.288**	**6.578**	00:00.172
P50S8T02	**2.750**	0.000	0.000	04:03.593
P50S8T03	**0.936**	**0.787**	**1.161**	00:00.359
P50S8T04	**4.760**	**2.696**	**3.917**	00:00.171
P50S8T05	**3.639**	**0.986**	**2.691**	00:13.594
P50S8T06	**1.875**	**0.313**	**1.641**	00:50.234
P50S8T07	**5.436**	**2.544**	**3.701**	00:33.453
P50S8T08	**2.434**	**1.881**	**1.917**	00:11.828
P50S8T09	**4.760**	**3.412**	**4.465**	00:31.813
P50S8T10	**5.371**	**3.187**	**4.726**	00:16.734
P1HS8T01	**2.877**	0.000	0.000	14:55.688
P1HS8T02	**3.568**	**0.157**	**0.801**	00:31.610
P1HS8T03	**1.987**	**0.644**	**1.472**	00:00.000
P1HS8T04	**2.149**	**0.103**	**0.498**	00:05.735
P1HS8T05	**1.944**	**0.019**	**0.731**	00:50.985
P1HS8T06	**3.422**	**1.104**	**2.907**	00:47.750
P1HS8T07	**2.426**	**0.019**	**1.203**	02:18.532
P1HS8T08	**7.294**	**5.584**	**8.451**	00:00.609
P1HS8T09	**1.951**	**0.020**	**1.334**	00:57.891
P1HS8T10	**3.838**	**0.185**	**0.985**	00:36.500

Tablo 4.6'ya göre; PDA yöntemi ile yapılan çözümleme sonucunda, *P10S8* tipi Oğuz ve Ercan (2005) 'ın kıyaslama problemlerinden hiçbirinin çözümü, ASD'ye ulaşamamıştır. ASD'ye ulaşamamış 10 problemden 1'inin çözüm değeri, Oğuz (2005)'un elde ettiği değer ile aynı; 9'unun çözüm değerleri ise Oğuz (2005)'un elde ettiği değerlerden daha kötüdür. ASD'ye ulaşamamış 10 problemden 4'ünün çözüm değerleri, Ceran (2006)'ın elde etmiş olduğu değerlerden daha iyi; 6'sının çözüm değerleri Ceran (2006)'ın elde etmiş olduğu değerler ile aynıdır. Bu

gruptaki Oğuz (2005)'un ve Ceran (2006)'ın elde ettikleri çözümlerden hiçbiri ASD'ye ulaşamamıştır.

Tablo 4.6'ya göre; PDA yöntemi ile yapılan çözümleme sonucunda *P20S8* tipi Oğuz ve Ercan (2005) 'ın kıyaslama problemlerinden 2 problemin çözüm değeri ASD'ye ulaşmıştır. ASD'ye ulaşamamış 8 problemden 2'sinin çözüm değerleri, Oğuz (2005)'un elde etmiş olduğu değerlerden daha iyi; 6 tanesinde ise daha kötüdür. ASD'ye ulaşamamış 8 problemden 4'ünün çözüm değeri Ceran (2006)'ın elde ettiği değerlerden daha iyi; 4'ünün çözüm değeri daha kötüdür. Bu gruptaki Oğuz (2005)'un ve Ceran (2006) 'ın elde ettikleri çözümlerden 2'şer tanesi ASD'ye ulaşmıştır.

Tablo 4.6'ya göre; PDA yöntemi ile yapılan çözümleme sonucunda, *P50S8* tipi Oğuz ve Ercan (2005) 'ın kıyaslama problemlerinden 1 problemin çözüm değeri ASD'ye ulaşmıştır. ASD'ye ulaşamamış 9 problemden 7'sinin çözüm değerleri, Oğuz (2005)'un elde etmiş olduğu değerlerden daha iyi; 2 tanesinde ise daha kötüdür. ASD'ye ulaşamamış 9 problemden 9'unun da çözüm değeri Ceran (2006)'ın elde ettiği değerlerden daha kötüdür. Bu gruptaki Oğuz (2005)'un elde ettiği çözümlerden hiçbiri ASD'ye ulaşamamışken, Ceran (2006)'ın elde ettiği çözümlerden 1 tanesi ASD'ye ulaşmıştır.

Tablo 4.6'ya göre; PDA yöntemi ile yapılan çözümleme sonucunda, *P1HS8* tipi Oğuz ve Ercan (2005) 'ın kıyaslama problemlerinden 1 problemin çözüm değeri ASD'ye ulaşmıştır. ASD'ye ulaşamamış 9 problemden 8'inin çözüm değerleri, Oğuz (2005)'un elde etmiş olduğu değerlerden daha iyi; 1 tanesinde ise daha kötüdür. ASD'ye ulaşamamış 9 problemden 9'unun da çözüm değeri Ceran (2006)'ın elde ettiği değerlerden daha kötüdür. Bu gruptaki Oğuz (2005)'un elde ettiği çözümlerden hiçbiri ASD'yi bulamamışken, Ceran (2006)'ın elde ettiği çözümlerden 1 tanesi ASD'ye ulaşmıştır.

Tablo 4.7 ÇİEATÇ *Q* tipi 2 aşamalı problemler için kıyaslama

Problem	OĞUZ GA	CERAN GA	PDA	
	% Sapma	% Sapma	% Sapma	İşlem Süreleri
Q10S2T1	0.000	0.000	0.000	00:00.000
Q10S2T2	**5.012**	**5.012**	**5.012**	00:00.000
Q10S2T3	**9.859**	0.000	0.000	00:00.234
Q10S2T4	0.000	0.000	0.000	00:00.000
Q10S2T5	0.000	**19.209**	**19.209**	00:00.000
Q10S2T6	**5.817**	**6.094**	**5.817**	00:00.000
Q10S2T7	**0.286**	**0.286**	**0.286**	00:00.343
Q10S2T8	**12.333**	**93.333**	**93.333**	00:00.000
Q10S2T9	**4.076**	**4.076**	**4.076**	00:00.078
Q10S2T10	**7.331**	**7.331**	**7.331**	00:00.062
Q20S2T1	**7.193**	**7.334**	**7.193**	00:00.125
Q20S2T2	**0.892**	**35.414**	**35.414**	00:00.000
Q20S2T3	**4.690**	**47.280**	**46.529**	00:00.000
Q20S2T4	0.000	0.000	0.000	00:00.125
Q20S2T5	**0.844**	**1.547**	**0.703**	00:00.032
Q20S2T6	**0.426**	**1.420**	**0.426**	00:01.890
Q20S2T7	0.000	0.000	0.000	00:00.000
Q20S2T8	**0.369**	**0.369**	**0.369**	00:00.000
Q20S2T9	0.000	0.000	0.000	00:00.000
Q20S2T10	**1.826**	**2.283**	**1.674**	00:00.047
Q50S2T1	**0.161**	**0.161**	**0.161**	00:00.078
Q50S2T2	**0.329**	**0.329**	**0.329**	00:13.259
Q50S2T3	**1.651**	**4.068**	**4.068**	00:00.110
Q50S2T4	**0.623**	**0.249**	**0.187**	00:00.156
Q50S2T5	0.000	0.000	0.000	00:01.046
Q50S2T6	**2.353**	**22.215**	**21.799**	00:00.110
Q50S2T7	0.000	0.000	0.000	00:00.109
Q50S2T8	**3.292**	**7.267**	**7.081**	00:00.156
Q50S2T9	**2.747**	**2.129**	**2.816**	00:00.063
Q50S2T10	**2.567**	**10.334**	**9.949**	00:00.141
Q1HS2T1	**2.250**	**2.214**	**2.179**	00:24.422
Q1HS2T2	**1.904**	**1.360**	**1.666**	00:00.203
Q1HS2T3	**3.849**	**4.364**	**3.952**	00:00.234
Q1HS2T4	0.000	**5.171**	**5.008**	00:00.500
Q1HS2T5	**0.029**	**0.029**	**0.029**	00:23.672
Q1HS2T6	**0.911**	**0.705**	**1.058**	00:00.265
Q1HS2T7	**1.891**	**2.165**	**2.247**	00:00.438
Q1HS2T8	**0.118**	**0.118**	**0.355**	00:28.438
Q1HS2T9	**1.642**	0.000	0.000	02:58.468
Q1HS2T10	**1.885**	**1.778**	**1.885**	01:19.234

Tablo 4.7'ye göre; PDA yöntemi ile yapılan çözümleme sonucunda, *Q10S2* tipi Oğuz ve Ercan (2005)'ın kıyaslama problemlerinden 3 problemin çözümleri ASD'ye ulaşmıştır. ASD'ye ulaşamamış 7 problemden 5'inin çözüm değerleri, Oğuz (2005)'un elde etmiş olduğu değerler ile aynı; 2 tanesinde ise daha kötüdür. ASD'ye ulaşamamış 7 problemden 6'sının çözüm değeri, Ceran (2006)'ın elde ettiği değerler ile aynı; 1 tanesinin çözüm değeri daha iyidir. Bu gruptaki Oğuz (2005)'un ve Ceran (2006)'ın elde ettikleri çözümlerden 3'er tanesi ASD'ye ulaşmıştır.

Tablo 4.7'ye göre; PDA yöntemi ile yapılan çözümleme sonucunda, *Q20S2* tipi Oğuz ve Ercan (2005)'ın kıyaslama problemlerinden 3 problemin çözümleri ASD'ye ulaşmıştır. ASD'ye ulaşamamış 7 problemden 2'sinin çözüm değerleri, Oğuz (2005)'un elde etmiş olduğu değerlerden daha iyi; 3'ünün çözüm değerleri ile aynı; 2 tanesinde ise daha kötüdür. ASD'ye ulaşamamış 7 problemden 5'inin çözüm değerleri, Ceran (2006)'ın elde ettiği değerlerden daha iyi; 2'sinin çözüm değerleri ile aynıdır. Bu gruptaki Oğuz (2005)'un ve Ceran (2006)'ın elde ettikleri çözümlerden 1'er tanesi ASD'ye ulaşmıştır.

Tablo 4.7'ye göre; PDA yöntemi ile yapılan çözümleme sonucunda, *Q50S2* tipi Oğuz ve Ercan (2005)'ın kıyaslama problemlerinden 2 problemin çözümünde ASD'ye ulaşılmıştır. ASD'ye ulaşamamış 8 problemden 1'inin çözüm değeri, Oğuz (2005)'un elde ettiği değerden daha iyi; 2'sinin çözüm değerleri, Oğuz (2005)'un elde ettiği değerler ile aynı; 5'inin çözüm değerleri, Oğuz (2005)'un elde ettiği değerlerden daha kötüdür. ASD'ye ulaşamamış 8 problemden 4'ünün çözüm değerleri, Ceran (2006)'ın elde ettiği değerlerden daha iyi; 3'ünün çözüm değerleri, Ceran (2006)'ın elde ettiği değerler ile aynı; 1'inin çözüm değeri, Ceran (2006)'ın elde ettiği değerden daha kötüdür. Bu gruptaki Oğuz (2005)'un ve Ceran (2006)'ın elde ettikleri çözümlerden 2'şer tanesi ASD'ye ulaşmıştır.

Tablo 4.7'ye göre; PDA yöntemi ile yapılan çözümleme sonucunda, *Q1HS2* tipi Oğuz ve Ercan (2005) 'ın kıyaslama problemlerinden 1 problem çözümünde ASD'ye ulaşılmıştır. ASD'ye ulaşamamış 9 problemden 2'sinin çözüm değerleri, Oğuz (2005)'un elde ettiği değerlerden daha iyi; 2'sinin çözüm değerleri, Oğuz (2005)'un elde ettiği değerler ile aynı; 5'inin çözüm değerleri, Oğuz (2005)'un elde ettiği değerlerden daha kötüdür. ASD'ye ulaşamamış 9 problemden 3'ünün çözüm değerleri, Ceran (2006)'ın elde ettiği değerlerden daha iyi; 2'sinin çözüm değerleri, Ceran (2006)'ın elde ettiği değerler ile aynı; 4'ünün çözüm değerleri, Ceran (2006)'ın elde ettiği değerlerden daha kötüdür. Bu gruptaki Oğuz (2005)'un ve Ceran (2006)'ın elde ettikleri çözümlerden 1'er tanesi ASD'ye ulaşmıştır.

Tablo 4.8 ÇİEATÇ *Q* tipi 5 aşamalı problemleri için kıyaslama

Problem	OĞUZ GA	CERAN GA	PDA	
	% Sapma	% Sapma	% Sapma	İşlem Süreleri
Q10S5T1	**9.291**	**9.291**	**9.291**	00:00.563
Q10S5T2	**1.536**	**2.560**	**1.877**	00:01.140
Q10S5T3	**5.664**	**9.027**	**7.434**	00:00.000
Q10S5T4	**6.919**	**14.827**	**7.084**	00:01.503
Q10S5T5	**2.574**	**2.574**	**2.574**	00:16.125
Q10S5T6	**8.155**	**10.485**	**8.155**	00:00.000
Q10S5T7	**15.146**	**15.146**	**15.146**	00:00.500
Q10S5T8	**3.811**	0.000	0.000	00:02.734
Q10S5T9	0.000	0.000	0.000	00:00.875
Q10S5T10	**13.163**	**13.163**	**13.163**	00:00.766
Q20S5T1	**3.533**	**2.257**	**2.257**	00:58.781
Q20S5T2	**7.970**	**7.846**	**7.970**	00:00.063
Q20S5T3	**13.626**	**12.915**	**13.626**	00:00.047
Q20S5T4	**4.277**	**9.017**	**9.364**	00:00.046
Q20S5T5	**13.907**	0.000	0.000	00:39.687
Q20S5T6	**7.093**	**12.093**	**10.930**	00:00.047
Q20S5T7	**4.580**	**4.580**	**4.580**	00:09.828
Q20S5T8	**11.578**	**7.761**	**11.323**	00:00.109
Q20S5T9	**1.647**	**1.207**	**3.074**	00:00.031
Q20S5T10	**5.345**	**4.120**	**5.122**	00:00.016
Q50S5T1	**15.558**	**11.163**	**13.707**	01:30.344
Q50S5T2	**9.406**	**8.567**	**7.839**	00:00.125
Q50S5T3	**10.042**	0.000	0.000	00:56.000
Q50S5T4	**11.830**	0.000	0.000	01:15.266
Q50S5T5	**17.784**	**13.644**	**15.685**	00:26.532
Q50S5T6	**12.231**	**9.966**	**11.665**	00:00.203
Q50S5T7	**8.294**	**1.417**	**3.622**	00:42.891
Q50S5T8	**11.680**	**0.410**	**0.359**	01:02.594
Q50S5T9	**2.479**	0.000	0.000	00:41.203
Q50S5T10	**2.723**	**0.050**	**1.733**	00:46.906
Q1HS5T1	**18.948**	**7.813**	**10.089**	00:33.641
Q1HS5T2	**15.451**	0.000	0.000	00:57.797
Q1HS5T3	**16.207**	**0.227**	**2.502**	03:41.358
Q1HS5T4	**18.948**	**8.213**	**11.750**	02:01.359
Q1HS5T5	**24.866**	**10.729**	**13.658**	01:21.000
Q1HS5T6	**17.349**	**4.855**	**7.610**	00:21.656
Q1HS5T7	**24.796**	**7.974**	**9.139**	01:07.140
Q1HS5T8	**18.322**	**2.657**	**4.895**	01:20.500
Q1HS5T9	**21.104**	**0.443**	**3.276**	01:33.938
Q1HS5T10	**22.710**	**6.609**	**8.952**	01:14.921

Tablo 4.8'e göre; PDA yöntemi ile yapılan çözümleme sonucunda, *Q10S5* tipi Oğuz ve Ercan (2005) 'ın kıyaslama problemlerinden 2 problemin çözümünde ASD'ye ulaşılmıştır. ASD'ye ulaşamamış 8 problemden 5'inin çözüm değerleri, Oğuz (2005)'un elde ettiği değerler ile aynı; 3'ünün çözüm değerleri, Oğuz (2005)'un elde ettiği değerlerden daha kötüdür. ASD'ye ulaşamamış 8 problemden 4'ünün çözüm değerleri, Ceran (2006)'ın elde ettiği değerlerden daha iyi; 4'ünün çözüm değerleri, Ceran (2006)'ın elde ettiği değerler ile aynıdır. Bu gruptaki Oğuz

(2005)'un elde ettiği çözümlerinden 1 tanesi ve Ceran (2006)'ın elde ettiği çözümlerinden 2 tanesi ASD'ye ulaşmıştır.

Tablo 4.8'e göre; PDA yöntemi ile yapılan çözümleme sonucunda, *Q20S5* tipi Oğuz ve Ercan (2005) 'ın kıyaslama problemlerinden 1 problemin çözümünde ASD'ye ulaşılmıştır. ASD'ye ulaşamamış 9 problemden 3'ünün çözüm değerleri, Oğuz (2005)'un elde ettiği değerlerden daha iyi; 3'ünün çözüm değerleri, Oğuz (2005)'un elde ettiği değerler ile aynı; 3'ünün çözüm değerleri, Oğuz (2005)'un elde ettiği değerlerden daha kötüdür. ASD'ye ulaşamamış 9 problemden 1'inin çözüm değeri, Ceran (2006)'ın elde ettiği değerden daha iyi; 2'sinin çözüm değerleri, Ceran (2006)'ın elde ettiği değerler ile aynı; 6'sının çözüm değerleri, Ceran (2006)'ın elde ettiği değerlerden daha kötüdür. Bu gruptaki Oğuz (2005)'un elde ettiği çözümlerden hiçbirisi ASD'ye ulaşmazken, Ceran (2006)'ın elde ettiği çözümlerden 2 tanesi ASD'ye ulaşmıştır.

Tablo 4.8'e göre; PDA yöntemi ile yapılan çözümleme sonucunda, *Q50S5* tipi Oğuz ve Ercan (2005)'ın kıyaslama problemlerinden 3 problemin çözümü ASD'ye ulaşmıştır. ASD'ye ulaşamamış 7 problemden 7'sinin de çözüm değerleri, Oğuz (2005)'un elde ettiği değerlerden daha iyidir. ASD'ye ulaşamamış 7 problemden 2'sinin çözüm değerleri, Ceran (2006)'ın elde ettiği değerlerden daha iyi; 5'inin çözüm değerleri, Ceran (2006)'ın elde ettiği değerlerden daha kötüdür. Bu gruptaki Oğuz (2005)'un elde ettiği çözümlerden hiçbirisi ASD'ye ulaşmazken, Ceran (2006)'ın elde ettiği çözümlerden 3 tanesi ASD'ye ulaşmıştır.

Tablo 4.8'e göre; PDA yöntemi ile yapılan çözümleme sonucunda, *Q1HS5* tipi Oğuz ve Ercan (2005)'ın kıyaslama problemlerinden 1 problemin çözümünde ASD'ye ulaşılmıştır. ASD'ye ulaşamamış 9 problemden 9'unun da çözüm değerleri, Oğuz (2005)'un elde ettiği değerlerden daha iyidir. ASD'ye ulaşamamış 9 problemden 9'unun da çözüm değerleri, Ceran (2006)'ın elde ettiği değerlerden daha kötüdür. Bu gruptaki Oğuz (2005)'un elde ettiği çözümlerden hiçbirisi ASD'ye ulaşmazken, Ceran (2006)'ın elde ettiği çözümlerden 1 tanesi ASD'ye ulaşmıştır.

Tablo 4.9 ÇİEATÇ *Q* tipi 8 aşamalı problemleri için kıyaslama

Problem	OĞUZ GA	CERAN GA	PDA	
	% Sapma	% Sapma	% Sapma	İşlem Süreleri
Q10S8T1	**15.860**	**25.134**	**25.134**	00:03.781
Q10S8T2	**7.570**	**2.789**	**2.789**	00:23.875
Q10S8T3	**8.005**	**7.349**	**7.349**	00:01.125
Q10S8T4	**2.289**	**2.169**	**2.169**	00:09.110
Q10S8T5	**19.635**	**20.700**	**20.700**	00:01.031
Q10S8T6	**13.018**	**17.456**	**17.456**	00:02.250
Q10S8T7	**14.944**	**32.432**	**32.432**	00:01.718
Q10S8T8	**13.580**	**14.938**	**14.938**	00:20.209
Q10S8T9	**18.053**	**19.937**	**19.937**	00:31.725
Q10S8T10	**7.261**	**4.331**	**4.331**	00:08.562
Q20S8T1	**11.168**	**11.578**	**10.656**	00:00.157
Q20S8T2	**18.653**	**24.283**	**21.192**	00:00.110
Q20S8T3	**10.169**	**7.439**	**10.829**	00:00.203
Q20S8T4	**9.436**	**8.201**	**9.347**	00:31.625
Q20S8T5	**18.469**	**21.327**	**19.286**	00:00.203
Q20S8T6	**11.168**	**10.041**	**10.143**	00:00.156
Q20S8T7	**40.285**	**21.625**	**24.808**	00:48.328
Q20S8T8	**21.613**	**22.151**	**22.366**	00:18.977
Q20S8T9	**20.128**	**21.949**	**18.522**	00:00.063
Q20S8T10	**14.924**	**13.807**	**13.909**	00:00.047
Q50S8T1	**20.491**	**18.705**	**19.151**	00:00.140
Q50S8T2	**14.607**	**14.921**	**15.236**	00:00.609
Q50S8T3	**9.231**	**0.585**	**1.921**	01:06.516
Q50S8T4	**24.140**	**19.677**	**22.742**	00:00.000
Q50S8T5	**22.415**	**11.916**	**14.016**	00:12.875
Q50S8T6	**13.731**	**10.985**	**13.068**	00:00.313
Q50S8T7	**9.445**	**0.688**	**4.035**	00:45.797
Q50S8T8	**24.389**	**32.048**	**31.505**	00:00.390
Q50S8T9	**14.828**	**12.827**	**16.624**	00:00.218
Q50S8T10	**15.652**	**7.851**	**8.807**	01:04.078
Q1HS8T1	**10.581**	**6.784**	**9.298**	00:00.656
Q1HS8T2	**8.463**	**2.414**	**6.153**	01:32.781
Q1HS8T3	**14.856**	**9.711**	**14.691**	04:24.343
Q1HS8T4	**16.203**	**7.787**	**12.035**	00:55.281
Q1HS8T5	**15.998**	**9.870**	**15.157**	01:25.203
Q1HS8T6	**16.969**	**13.994**	**16.601**	00:01.391
Q1HS8T7	**15.692**	**8.967**	**13.286**	01:03.172
Q1HS8T8	**15.458**	**15.197**	**17.952**	02:56.815
Q1HS8T9	**11.234**	0.000	0.000	19:00.266
Q1HS8T10	**9.151**	**5.693**	**9.710**	01:33.675

Tablo 4.9'a göre; PDA yöntemi ile yapılan çözümleme sonucunda, *Q10S8* tipi Oğuz ve Ercan (2005) 'ın kıyaslama problemlerinden hiçbirisi ASD'ye ulaşamamıştır. ASD'ye ulaşamamış 10 problemden 6'sının çözüm değerleri, Oğuz (2005)'un elde ettiği değerlerden daha iyi; 4'ünün çözüm değerleri ise Oğuz (2005)'un elde ettiği değerlerden daha kötüdür. ASD'ye ulaşamamış 10 problemden 4'ünün çözüm değeri Ceran (2006)'ın elde ettiği değerden daha iyi; 6'sının çözüm değerleri, Ceran (2006)'ın elde ettiği değerlerden daha kötüdür. Bu gruptaki Oğuz

(2005)'un ve Ceran (2006)'ın elde ettikleri çözümlerden hiçbirisi ASD'ye ulaşamamıştır.

Tablo 4.9'a göre; PDA yöntemi ile yapılan çözümleme sonucunda, *Q20S8* tipi Oğuz ve Ercan (2005) 'ın kıyaslama problemlerinden hiçbirisi ASD'ye ulaşamamıştır. ASD'ye ulaşamamış 10 problemden 6'sının çözüm değerleri, Oğuz (2005)'un elde ettiği değerlerden daha iyi; 4'ünün çözüm değerleri ise Oğuz (2005)'un elde ettiği değerlerden daha kötüdür. ASD'ye ulaşamamış 10 problemden 4'ünün çözüm değeri Ceran (2006)'ın elde ettiği değerden daha iyi; 6'sının çözüm değerleri, Ceran (2006)'ın elde ettiği değerlerden daha kötüdür. Bu gruptaki Oğuz (2005)'un ve Ceran (2006)'ın elde ettikleri çözümlerden hiçbirisi ASD'ye ulaşamamıştır.

Tablo 4.9'a göre; PDA yöntemi ile yapılan çözümleme sonucunda, *Q50S8* tipi Oğuz ve Ercan (2005) 'ın kıyaslama problemlerinden hiçbirisi ASD'ye ulaşamamıştır. ASD'ye ulaşamamış 10 problemden 7'sinin çözüm değerleri, Oğuz (2005)'un elde ettiği değerlerden daha iyi; 3'ünün çözüm değerleri ise Oğuz (2005)'un elde ettiği değerlerden daha kötüdür. ASD'ye ulaşamamış 10 problemden 1'inin çözüm değeri Ceran (2006)'ın elde ettiği değerden daha iyi; 9'unun çözüm değerleri, Ceran (2006)'ın elde ettiği değerlerden daha kötüdür. Bu gruptaki Oğuz (2005)'un ve Ceran (2006)'ın elde ettikleri çözümlerden hiçbirisi ASD'ye ulaşamamıştır.

Tablo 4.9'a göre; PDA yöntemi ile yapılan çözümleme sonucunda, *Q1HS8* tipi Oğuz ve Ercan (2005)'ın kıyaslama problemlerinden sadece 1 tanesi ASD'ye ulaşmıştır. ASD'ye ulaşamamış 9 problemden 7'sinin çözüm değerleri, Oğuz (2005)'un elde ettiği değerlerden daha iyi; 2'sinin çözüm değerleri ise Oğuz (2005)'un elde ettiği değerlerden daha kötüdür. ASD'ye ulaşamamış 9 problemden 9'unun da çözüm değerleri, Ceran (2006)'ın elde ettiği değerlerden daha kötüdür. Bu gruptaki Oğuz (2005)'un elde ettiği çözümlerden hiçbirisi ASD'ye ulaşmazken, Ceran (2006)'ın elde ettiği çözümlerden 1 tanesi ASD'ye ulaşmıştır.

PDA, Oğuz (2005)'un ve Ceran (2006)'ın GA metodlarının Oğuz ve Ercan (2005)'ın *P* tipi ve *Q* tipi Problemleri üzerinde gösterdikleri performanslar, Tablo 4.10'da sunulmuştur.

Tablo 4.10 PDA ve GA Sezgisel Metodların Performansları

Metod	***P* Tipi Problemler**		***Q* Tipi Problemler**	
	% Çözüm	**% Sapma**	**% Çözüm**	**% Sapma**
PDA	35,00	3,4255	14,17	9,3489
OĞUZ GA	24,17	3,4980	8,33	9.4122
CERAN GA	35,00	3,1691	14,17	8.1201

5. SONUÇ ve ÖNERİLER

Araştırmada, EATÇ ve ÇİEATÇ problemlerinin çözümünde PDA metodunun işlerliği ve performansı incelenmiştir.

EATÇ ve ÇİEATÇ problemleri NP (Polinomiyel Olmayan) – Zor problemler olarak bilinmektedir. Bu problemleri optimal şekilde çözmek için geliştirilmiş bir yöntem bulunmamakta ve daha çok sezgisel yöntemler kullanılmaktadır. Akış tipi çizelgeleme problemlerinde her aşama tek makineden oluşmakta ve işler atölyedeki aşamalarda makineleri aynı sırada ziyaret etmektedir. Esnek akış tipi çizelgeleme problemi ise klasik akış tipi ve paralel makine problemlerinin bir birleşimi şeklindedir. Bu problemler gerçek hayatta karşılaşılabilecek çizelgeleme problemleridir. Özellikle esnek akış tipi çizelgeleme uygulamaları sıkça görülmektedir. Gerçek hayattaki esnek akış tipi sistemlerinde genellikle aynı kademedeki makineler eştir. Her kademede sadece bir tip görev yapılmaktadır. Her görev sadece kendinden önceki görev tamamlandıktan sonra işlemine başlayabilir.

Tez çalışmasında literatürde tanımlanan Oğuz ve Ercan (2005) 'ın ÇİEATÇ ile Carlier ve Neron (2000) 'un EATÇ problemlerinin, PDA yöntemi ile çözümü sonucu elde edilen değerlerin birçoğunun ASD'ye yaklaştığı görülmüştür. Çalışmada esnek akış tipi problemlerinin çözümüne başlamadan önce ASD'ye ulaştıracak kontrol parametreleri tanımlanmıştır. Tanımlanan kontrol parametre değerleri ile Oğuz ve Ercan (2005) 'ın ÇİEATÇ problemleri ve Carlier ve Neron (2000) 'un EATÇ problemleri tekrarlı bir şekilde çözülmüştür. Carlier ve Neron (2000) 'un EATÇ problemleri için PDA metodu ile elde edilen çözümler, Döyen (2004) 'in YBS ve Neron (2001)'un DSA sezgisel yaklaşımları ile elde edilen çözümler ile karşılaştırılmıştır. Oğuz ve Ercan (2005)'ın ÇİEATÇ problemleri için PDA ile elde edilen çözümler, Oğuz (2005)'un GA ve Ceran (2006)'ın GA sezgisel yaklaşımları ile elde edilen çözümler ile karşılaştırılmıştır.

Çalışmada EATÇ problemleri için yapılan karşılaştırmalar sonucunda; PDA metodu ile Carlier ve Neron (2000) 'un kolay problemlerinden % 88'inin çözümünün

ASD'ye ulaştığı belirlenmiştir. Bu oranın, YBS ve DSA metodlarında elde edilmiş olan oranlar ile aynı olduğu gözlenmiştir. ASD'den sapma yüzdesi bakımından PDA metodu ile çözümlemede % 0.99 sapma oranı elde edilmiştir. Bu oran YBS metodu ile aynı, DSA metodundan daha iyi bir orandır. Bu oranlar Carlier ve Neron (2000) 'un kolay problemleri üzerindeki uygulama bakımından PDA'nın YBS ile aynı, DSA'dan daha fazla performansa sahip olduğu belirlenmiştir. PDA metodu ile Carlier ve Neron (2000) 'un zor problemlerinden % 66,7'sinin çözümü ASD'ye ulaşmıştır. Bu çözüm oranının, YBS çözüm oranı ile aynı; DSA çözüm oranından daha iyi olduğu belirlenmiştir. % 3,10 sapma miktarı ile PDA ile elde edilen çözümlerin YBS ve DSA ile elde edilen çözümlerden daha etkin olduğu belirlenmiştir.

Çalışmada, ÇİEATÇ problemleri için yapılan karşılaştırmalar sonucunda PDA metodu ile *P* tipi problemlerin çözüm değerlerinin % 35,00'i ASD'ye ulaşmıştır. Bu oran, Ceran (2006)'ın GA yöntemi ile elde edilen çözüm oranı ile aynı; Oğuz (2005)'un GA yöntemi ile elde edilen çözüm oranından daha iyidir. % 3,4255 sapma miktarı ile PDA; Ceran (2006)'ın önerdiği GA metodundan daha az; Oğuz (2005)'un önerdiği GA metodundan daha fazla performansa sahiptir. *Q* tipi problemleri için PDA metodu ile problemlerin % 14,17'si ASD'ye ulaşmıştır. Bu oran, Ceran (2006)'ın GA yöntemi ile elde edilen çözüm oranı ile aynı; Oğuz (2005)'un GA yöntemi ile elde edilen çözüm oranından daha iyidir. % 9,3489 sapma miktarı ile PDA; Ceran (2006)'ın önerdiği GA metodundan daha az; Oğuz (2005)'un önerdiği GA metodundan daha fazla performansa sahiptir.

PDA metodu; Carlier ve Neron (2000)'un EATÇ problemleri için kullanıldığında Döyen (2004)'in YBS metoduna göre çözüm yüzdesi bakımından aynı, sapma miktarı bakımından ise daha iyi bir performans gösterirken; Neron (2001)'un DSA metoduna göre çözüm yüzdesi bakımından kolay problemler için aynı, zor problemler için daha az, sapma miktarı bakımından ise her iki tip problem için de daha iyi bir performans göstermektedir.

PDA metodu; Oğuz ve Ercan (2005)'ın ÇİEATÇ problemleri için kullanıldığında; çözüm oranı bakımından Ceran (2006)'ın önerdiği GA metodu ile aynı; Oğuz (2005)'un önerdiği GA metodundan daha fazla performansa sahiptir.

6. KAYNAKLAR

Anonim (1): Greedy Algoritmaları,
http://www.students.itu.edu.tr/~gungorse/algoritma/Greedy.php ,
Selami GÜNGÖR, İTÜ, Bilgisayar Mühendisliği Bölümü, 2004

Anonim (2): Bozuk Para Algoritması,
http://www.students.itu.edu.tr/~gungorse/algoritma/bozukPara.php,
Selami GÜNGÖR, İTÜ, Bilgisayar Mühendisliği Bölümü, 2004

Anonim (3): Greedy Algorithms,
http://sci.vu.edu.au/~amca/SCM2711/ch06_heuristics.pdf,
Alasdair McAndrew , Victoria Üniversitesi, Bilgisayar ve Matematik Bilimleri Bölümü, 2005

Anonim (4): Greedy Algorithms: Interval Scheduling,
http://www.csc.liv.ac.uk/~darek/COMP523/lecture6.pdf ,
Dariusz KOWALSKİ, Liverpool Üniversitesi Bilgisayar Mühendisliği Bölümü, 2005

Anonim (5): Greedy Algorithm,
http://en.wikipedia.org/wiki/Greedy_algorithm, 2006

Anonim (6): Greedy Algorithms,
http://www.bookrags.com/sciences/computerscience/Greedy-algorithms-wcs.html, 2005

Anonim (7): Greedy Algorithms: Minimum Spanning Tree
http://www.csc.liv.ac.uk/~darek/COMP523/lecture6.pdf,
Dariusz KOWALSKİ, Liverpool Üniversitesi Bilgisayar Mühendisliği Bölümü, 2005

Anonim (8): Introduction to Algorithms Part 2 : Greedy Algorithms,Dynamic Programming,Graph Algorithms
http://www.cs.clemson.edu/~goddard/handouts/texts/840/part2.pdf,
Wayne Goddard, Clemson Üniversitesi, Bilgisayar Mühendisliği Bölümü, 2005

Anonim (11): Search Method
http://www.cs.man.ac.uk/~jls/CS2411/Search.pdf,
Jonathan Shapiro, Manchester Üniversitesi Bilgisayar Mühendisliği Bölümü, 2005

Anonim (12): GRASP
http://dokumanlar.com/dokuman/2005_6/grasp-algoritmasi, 2005

Anonim (13): GRASP
http://www.odevsitesi.com/ , 2005

Abdekhodaee, A.H., Wirth. A., Gan H.S. 2006. Scheduling two paralel machines with a single server: the general case, Computers & Operations Research, Vol. 33. 994-1009

Aiex, R. M., Binato, S., Resende M.G.C. 2003. Paralel GRASP With Path – Relinking For Job Shop Scheduling, Parallel Computing, Vol.29. 393-430

Alidae, B., Kochenberger, G.A., Amini M.M. 2001. Greedy Solutions Of Selection and Ordering Problems, European Journal of Operational Research,Vol.134 .203-215

Anily, S., Glass, C. A. ve Hassin, R. 1998. The Scheduling of maintenance service, Discrete Applied Mathematics, Vol.82.27-42

Arthanari,T.S., Ramamurthy, K.G., 1971. An Extension of Two Machines Sequencing Problem, Opsearch, 8, 10-22.

Bertel, S., Billaut, J.C. 2004. A Genetic algorithm for an industrial multiprocessor flow shop scheduling problem with recirculation, European Journal of Operational Research,Vol.159. 651-662

Binato, S., Hery, W., Loewenstern, D., Resende, M., 2001, A GRASP for job shop scheduling in: C. Riberio, P. Hansen (Eds), Essays and Surveys on Metaheuristic, Kluwer Academic Publishers, 59-79

Brah, S.A., Hunsucker,J.L., 1991, Branch and Bound Algorithm for The Flow Shop With Multiple Processors, European Journal of Operational Research, 51, 88-99.

Brah,S.A, Loo, L.L., 1999, Heuristics For Scheduling in A Flow Shop With Multiple Processors, European Journal of Operational Research, 113, 113-122.

Carlier,J., Neron,E., 2000. An exact method for solving the multiprocessor flowshop, R.A.I.R.O-R.O.,34, 1-25.

Ceran, G., 2006, Esnek akış tipi çizelgeleme problemlerini veri madenciliği kullanarak genetik algoritmalar yardımıyla çözülmesi, Yüksek Lisans Tezi, Selçuk Üniversitesi

Chung,Y.L., Vairaktarakis, G.L., 1994, Minimizing makespan in hybrid flowshops, Operations Research Letters, Vol.16.

Döyen, A., 2004, Akış Tipi Çizelgeleme Problemlerinin Yapay Bağışıklık Sistemleri İle Çözümü ve Parametre Optimizasyonu, Yüksek Lisans Tezi, Selçuk Üniversitesi

Engin, O. ,Döyen, A., 2004, Esnek Akış Tipi Çizelgeleme Problemlerinin A New approach to solve hybrid flow shop scheduling problems by artificial immune system, Future generation computer systems, 20, 1083-1095.

Edwin, S.H.H., Ansari, N., Ren, H., 1994, A Genetic algorithm for Multiprocessor Scheduling, IEEE Transactions on Parallel and Distributed Systems, 5,2, 113-120.

Faigle, U., Kern, W., Nawijn, W. M. 1999. A Greedy On-line Algorithm For The k-Track Assignment Problem, Journal of Algorithms, Vol.31 . 196-210

Gupta, J.N.D., 1988. Two-stage hybrid flowshop scheduling problem, Operational Research Society, 39, 359-364.

Gupta,J.N.D., Tunç,E.A.,1991. Schedules for a two stage hybrid flowshop with paralel machines at the second stage, International Journal of Production Research, 29, 1489-1502.

Gupta,J.N.D., Hariri, A.M.A., Potts, C.N.,1997. Scheduling a two stage hybrid flow shop with paralel machines at the first stage. J. Ann.Oper.Res.,69, 171-191.

Gupta, S.R. , Smith, J.S., 2005. Algorithms for single machine total tardiness scheduling with sequence dependent setups, European Journal of Operational Research.

Jensen, J.B., Gutin, G., Yeo, A. 2004. When the greedy algorithm fails, Discrete Optimization, Vol 1. 121-127

Kang, J., Park, S. 2003. Algorithms for variable sized bin packing problem, European Journal of Operational Research,Vol.147.365-372

Kochar,S., Morris,R.J.T., 1987. Heuristic methods for flexible flow line scheduling, Journal of Manufacturing Systems, 6, 299-314.

Kurtz, M.E., Askin, R.G. 2003. Comparing Scheduling Rules For Flexible Flow Lines, International Journal Of Production Economics, Vol. 84. 371-388

Lagodimos, A. G. ,Leopoulos V. 2000. Greedy Heuristic Algorithms For Manpower Shift Planning, International Journal of Production Economics,Vol.68.95-106

Lui, K–S., Zaks, S. 1999. Scheduling In Synchronous Networks and The Greedy Algorithm, Theorical Computer Science, Vol.220 . 157-183

Moursli,O., Pochet,Y.A., 2000. Branch and bound algorithm for the hybrid flowshop, International journal of Production Economics, 64, 113-125.

Neron,E., Baptiste,P., Gupta, J.N.D., 2001, Solving hybrid flow shop problem using energetic reasoning and global operations, Omega, 29, 501-511.

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

Oğuz, C., Zinder, Y., Do, V.H., Janiak, A., Linchtenstein, M., 2004, Hybrid flow shop scheduling problems with multiprocessor task systems, European Journal of Operational Research, 152, 115-131.

Oğuz, C., Ercan, F., 2005, A Genetic Algorithm For Hybrid Flow shop Scheduling With multiprocessor tasks, journal of scheduling 8 323-351.

Papakonstantinou, P.A., 2005. Hierarchies for classes of priority algorithms for job scheduling, Theorical Computer Science.

Portmann, M.C. , Vignier, A., Dardilhac, D., Dezalay, D., 1998. Branch and bound crossed with GA to solve hybrid flowshops, European Journal of Operational Research, 107, 389-400.

Rajendran, C., Chaudhuri, D., 1992a. Scheduling in n-job, m-stage flowshop with paralel processors to minimize makespan. International Journal of Production Economics, 27,137-143.

Rajendran, C., Chaudhuri, D., 1992b. A multi-stage paralel processor flowshop problem with minimum flowtime. European Journal of Operational Research, 57, 111-122.

Riane,F., Artiba,A., Elmaghraby, S.E.,1998. A hybrid three-stage flowshop problem: efficient heuristics to minimize makespan, European Journal of Operational Research, 109, 321-329.

Ruiz, R., Stützle, T. 2006. A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem, European Journal of Operational Research.

Sivrikaya Şerifoğlu, S.,F., Ulusoy G., 2004, Multiprocessor task scheduling in multistage hybrid flow shops: a genetic algorithm approach, Journal of the Operational Research Society, 55, 504-512.

Sriskandarajah,C., Sethi,S.P.,1989. Scheduling algorithms for flexible flowshops: worst and average case performance, European Journal of Operational Research, 43,143-160.

Su, L.H, 2002, A hybrid two-stage flowshop with limited waiting time constraints, Computers & Industrial Engineering Vol. 44. 409-424

Suriyaarachchi, R. H., Wirth, A. 2004. Earliness/Tardiness Scheduling with a common due date and family setups, Computers & Industrial Engineering, Vol.47 . 275-288

Tang, L., Liu, J., Rong, A., Yang, Z. 2000. A multiple Travelling Salesman Problem Model For Hot Rolling Scheduling In Shanghai Baoshan Iron&Steel Complex, European Journal of Operational Research, Vol.124. 267-282

Yao, M.-J. 2001. The Peak Load Minimization Problem In Cyclic Production, Computers & Operation Research, Vol.28. 1441-1460

EKLER

Ek 1- 1 10x5 tipi problemler için optimizasyon sonuçları

No	Problem Tipi	Popülasyon	Iterasyon Sayısı	Alt Grup Sayısı	Doyum Oranı	İnşa Metodu	C_{max}	Iterasyon	İşlem Süreleri (dk:sn.sn'nin binde biri)
1	*j10c5a2*	15	100	9	0,5	BAİY	88	0	00:00.000
2	*j10c5a3*	30	1000	9	0,5	BAİY	117	0	00:00.000
3	*j10c5a4*	30	1000	9	0,5	BAİY	121	0	00:00.015
4	*j10c5a5*	30	1000	9	0,5	BAİY	122	0	00:00.000
5	*j10c5a6*	30	1000	9	0,5	BAİY	110	3	00:00.063
6	*j10c5b1*	30	1000	9	0,5	BAİY	130	0	00:00.000
7	*j10c5b2*	30	1000	9	0,5	BAİY	107	0	00:00.000
8	*j10c5b3*	30	1000	9	0,5	BAİY	109	0	00:00.000
9	*j10c5b4*	30	1000	9	0,5	BAİY	122	0	00:00.000
10	*j10c5b5*	30	1000	9	0,5	BAİY	153	0	00:00.000
11	*j10c5b6*	30	1000	9	0,5	BAİY	115	0	00:00.000
12	*j10c5c1*	15	50	5	0,4	BAİY	68	0	00:00.016
13	*j10c5c2*	30	1000	5	0,5	BAİY	74	17	00:00.140
14	*j10c5c3*	15	50	2	0,1	SY	72	0	00:00.000
15	*j10c5c4*	50	2500	2	0,3	SY	66	10	00:00.265
16	*j10c5c5*	30	1000	5	0,5	BAİY	78	13	00:00.750
17	*j10c5c6*	15	50	2	0,3	BAİY	69	0	00:00.000
18	*j10c5d1*	30	1000	9	0,5	BAİY	66	56	00:02.140
19	*j10c5d2*	15	50	8	0,6	BAİY	73	0	00:00.000
20	*j10c5d3*	15	1000	7	0,3	ÖY	64	28	00:00.016
21	*j10c5d4*	15	50	2	0,4	SY	70	0	00:00.000
22	*j10c5d5*	30	150	3	0,5	ÖY	66	0	00:00.000
23	*j10c5d6*	30	1000	9	0,5	BAİY	62	63	00:00.203

Ek 1- 2 10x10 tipi problemler için optimizasyon sonuçları

No	Problem Tipi	Popülasyon	Iterasyon Sayısı	Alt Grup Sayısı	Doyum Oranı	İnşa Metodu	C_{max}	Iterasyon	İşlem Süreleri (dk:sn.sn'nin binde biri)
1	*j10c10a1*	15	100	9	0,5	BAİY	139	0	00:00.020
2	*j10c10a2*	30	100	6	0,3	BAİY	158	0	00:00.015
3	*j10c10a3*	30	100	2	0,5	SY	148	0	00:00.109
4	*j10c10a4*	30	100	2	0,1	İAİY	149	0	00:00.000
5	*j10c10a5*	30	100	2	0,1	BAİY	148	0	00:00.000
6	*j10c10a6*	30	100	2	0,1	BAİY	146	1	00:00.015
7	*j10c10b1*	30	100	2	0,1	BAİY	163	0	00:00.000
8	*j10c10b2*	30	100	6	0,1	BAİY	157	1	00:00.000
9	*j10c10b3*	30	100	6	0,2	BAİY	169	1	00:00.000
10	*j10c10b4*	30	100	2	0,1	BAİY	159	0	00:00.000
11	*j10c10b5*	30	100	2	0,1	BAİY	165	0	00:00.000
12	*j10c10b6*	30	100	2	0,1	ÖY	165	0	00:00.000
13	*j10c10c1*	15	50	3	0,2	İAİY	115	0	00:00.000
14	*j10c10c2*	15	500	2	0,4	ÖY	117	27	00:00.094
15	*j10c10c3*	15	50	4	0,7	İAİY	116	0	00:00.000
16	*j10c10c4*	15	50	3	0,1	BAİY	120	0	00:00.000
17	*j10c10c5*	15	50	2	0,5	İAİY	126	0	00:00.000
18	*j10c10c6*	15	50	4	0,4	BAİY	106	43	00:00.125

Ek 1- 3 15x5 tipi problemler için optimizasyon sonuçları

No	Problem Tipi	Popülasyon	Iterasyon Sayısı	Alt Grup Sayısı	Doyum Oranı	İnşa Metodu	C_{max}	Iterasyon	İşlem Süreleri (dk:sn.sn'nin binde biri)
1	*j15c5a1*	15	100	9	0,5	BAİY	178	2	00:00.050
2	*j15c5a2*	30	100	2	0,7	BAİY	165	0	00:00.000
3	*j15c5a3*	30	100	2	0,2	SY	130	0	00:00.000
4	*j15c5a4*	30	100	2	0,2	BAİY	156	0	00:00.000
5	*j15c5a5*	30	100	2	0,3	İAİY	164	0	00:00.000
6	*j15c5a6*	30	100	2	0,3	SY	178	0	00:00.000
7	*j15c5b1*	30	100	2	0,1	İAİY	170	0	00:00.000
8	*j15c5b2*	30	100	2	0,1	BAİY	152	0	00:00.000
9	*j15c5b3*	30	100	2	0,1	SY	157	0	00:00.000
10	*j15c5b4*	30	100	4	0,1	ÖY	147	0	00:00.000
11	*j15c5b5*	30	100	5	0,4	SY	166	100	00:00.000
12	*j15c5b6*	30	100	2	0,1	SY	175	0	00:00.000
13	*j15c5c1*	30	150	4	0,6	İAİY	85	0	00:00.000
14	*j15c5c2*	30	100	5	0,5	BAİY	91	58	00:01.265
15	*j15c5c3*	15	250	4	0,3	İAİY	87	122	00:00.329
16	*j15c5c4*	30	150	5	0,3	İAİY	89	0	00:00.000
17	*j15c5c5*	15	100	4	0,3	ÖY	75	0	00:00.000
18	*j15c5c6*	30	1000	5	0,5	BAİY	91	85	00:00.438
19	*j15c5d1*	30	100	2	0,1	İAİY	167	0	00:00.000
20	*j15c5d2*	30	200	5	0,4	İAİY	84	0	00:00.000
21	*j15c5d3*	15	100	5	0,6	İAİY	83	0	00:00.016
22	*j15c5d4*	15	200	6	0,5	BAİY	84	0	00:00.000
23	*j15c5d5*	15	100	3	0,6	SY	80	0	00:00.000
24	*J15c5d6*	15	200	9	0,8	BAİY	81	0	00:00.000

Ek 1- 4 15x10 tipi problemler için optimizasyon sonuçları

No	Problem Tipi	Popülasyon	Iterasyon Sayısı	Alt Grup Sayısı	Doyum Oranı	İnşa Metodu	C_{max}	Iterasyon	İşlem Süreleri (dk:sn.sn'nin binde biri)
1	*j15c10a1*	30	100	2	0,1	BAİY	236	0	00:00.000
2	*j15c10a2*	30	100	2	0,1	SY	200	72	00:00.828
3	*j15c10a3*	30	100	2	0,1	BAİY	198	0	00:00.016
4	*j15c10a4*	30	100	2	0,1	İAİY	225	0	00:00.015
5	*j15c10a5*	30	100	2	0,1	SY	182	10	00:00.172
6	*j15c10a6*	30	100	2	0,2	ÖY	200	0	00:00.000
7	*j15c10b1*	30	100	2	0,1	SY	222	0	00:00.000
8	*j15c10b2*	30	100	2	0,2	BAİY	187	4	00:00.062
9	*j15c10b3*	30	100	2	0,1	BAİY	222	0	00:00.000
10	*j15c10b4*	30	100	2	0,1	BAİY	221	0	00:00.000
11	*j15c10b5*	30	100	7	0,1	BAİY	200	0	00:00.000
12	*J15c10b6*	30	100	2	0,1	ÖY	219	0	00:00.000

Ek 2- 1 *P* tipi 2. aşama problemleri için optimizasyon sonuçları

No	Problem Tipi	Popülasyon	Iterasyon Sayısı	Alt Grup Sayısı	Doyum Oranı	İnşa Metodu	Iterasyon	İşlem Süreleri (dk:sn.sn'nin binde biri)
1	*P10S2T01*	30	500	8	0,5	BAİY	0	00:00.062
2	*P10S2T02*	30	1000	5	0,5	BAİY	1	00:00.016
3	*P10S2T03*	30	1000	5	0,5	BAİY	105	00:01.406
4	*P10S2T04*	30	1000	5	0,5	BAİY	0	00:00.000
5	*P10S2T05*	30	100	2	0,1	BAİY	0	00:00.000
6	*P10S2T06*	30	100	2	0,1	BAİY	0	00:00.000
7	*P10S2T07*	30	100	2	0,1	BAİY	0	00:00.000
8	*P10S2T08*	30	100	6	0,1	SY	0	00:00.000
9	*P10S2T09*	30	100	2	0,1	BAİY	0	00:00.000
10	*P10S2T10*	30	100	2	0,1	İAİY	0	00:00.000
11	*P20S2T01*	30	500	7	0,3	İAİY	0	00:00.015
12	*P20S2T02*	30	100	4	0,6	SY	19	00:02.516
13	*P20S2T03*	30	100	2	0,1	İAİY	0	00:00.000
14	*P20S2T04*	30	100	2	0,1	SY	0	00:00.000
15	*P20S2T05*	30	100	2	0,2	BAİY	0	00:00.000
16	*P20S2T06*	30	100	2	0,4	BAİY	0	00:00.063
17	*P20S2T07*	30	100	2	0,6	ÖY	10	00:00.765
18	*P20S2T08*	30	100	4	0,4	SY	34	00:00.968
19	*P20S2T09*	30	100	2	0,1	BAİY	0	00:00.000
20	*P20S2T10*	30	100	4	0,7	SY	12	00:00.484

Ek 2- 1 *P* tipi 2. aşama problemleri için optimizasyon sonuçları (devamı)

No	Problem Tipi	Popülasyon	Iterasyon Sayısı	Alt Grup Sayısı	Doyum Oranı	İnşa Metodu	Iterasyon	İşlem Süreleri (dk:sn.sn'nin binde biri)
21	*P50S2T01*	30	100	2	0,7	İAİY	42	00:13.265
22	*P50S2T02*	30	100	2	0,1	BAİY	0	00:00.000
23	*P50S2T03*	15	50	6	0,6	İAİY	0	00:00.125
24	*P50S2T04*	15	500	4	0,7	SY	333	01:39.297
25	*P50S2T05*	30	100	2	0,4	ÖY	0	00:00.078
26	*P50S2T06*	30	100	3	0,1	BAİY	1	00:00.016
27	*P50S2T07*	30	100	5	0,7	İAİY	85	00:47.172
28	*P50S2T08*	30	100	3	0,1	BAİY	0	00:00.015
29	*P50S2T09*	30	100	5	0,1	İAİY	1	00:00.047
30	*P50S2T10*	30	100	2	0,2	ÖY	90	00:03.891
31	*P1HS2T01*	15	500	2	0,6	SY	184	02:55.437
32	*P1HS2T02*	30	100	5	0,1	İAİY	0	00:00.016
33	*P1HS2T03*	30	100	4	0,5	BAİY	83	01:08.890
34	*P1HS2T04*	30	100	5	0,1	BAİY	0	00:00.172
35	*P1HS2T05*	30	100	2	0,1	İAİY	0	00:00.156
36	*P1HS2T06*	30	100	2	0,1	SY	0	00:00.031
37	*P1HS2T07*	30	100	3	0,6	ÖY	69	02:03.328
38	*P1HS2T08*	30	100	2	0,1	SY	0	00:00.172
39	*P1HS2T09*	30	500	5	0,5	SY	0	00:00.188
40	*P1HS2T10*	30	100	3	0,6	SY	58	01:47.734

Ek 2- 2 *P* tipi 5. aşama problemleri için optimizasyon sonuçları

No	Problem Tipi	Popülasyon	Iterasyon Sayısı	Alt Grup Sayısı	Doyum Oranı	İnşa Metodu	Iterasyon	İşlem Süreleri (dk:sn.sn'nin binde biri)
1	*P10S5T01*	30	100	3	0,4	İAİY	32	00:04.844
2	*P10S5T02*	30	100	4	0,5	İAİY	100	00:00.000
3	*P10S5T03*	30	100	5	0,7	ÖY	6	00:01.984
4	*P10S5T04*	15	250	2	0,6	BAİY	35	00:01.469
5	*P10S5T05*	30	100	2	0,7	ÖY	2	00:00.360
6	*P10S5T06*	30	100	4	0,4	İAİY	100	00:00.000
7	*P10S5T07*	30	100	3	0,7	BAİY	10	00:02.063
8	*P10S5T08*	30	100	2	0,7	İAİY	20	00:03.516
9	*P10S5T09*	15	100	3	0,3	ÖY	0	00:00.040
10	*P10S5T10*	15	100	5	0,2	BAİY	0	00:00.000
11	*P20S5T01*	30	100	2	0,5	BAİY	26	00:01.610
12	*P20S5T02*	30	100	2	0,5	SY	32	00:04.188
13	*P20S5T03*	15	100	5	0,5	SY	0	00:00.062
14	*P20S5T04*	15	50	2	0,6	ÖY	0	00:00.047
15	*P20S5T05*	15	500	5	0,3	SY	0	00:00.031
16	*P20S5T06*	10	500	9	0,8	ÖY	0	00:00.047
17	*P20S5T07*	30	100	3	0,3	BAİY	14	00:01.891
18	*P20S5T08*	30	100	4	0,5	BAİY	100	00:00.000
19	*P20S5T09*	30	100	7	0,2	SY	100	00:00.000
20	*P20S5T10*	15	100	3	0,7	SY	49	00:44.784

Ek 2- 2 *P* tipi 5. aşama problemleri için optimizasyon sonuçları (devamı)

No	Problem Tipi	Popülasyon	Iterasyon Sayısı	Alt Grup Sayısı	Doyum Oranı	İnşa Metodu	Iterasyon	İşlem Süreleri (dk:sn.sn'nin binde biri)
21	*P50S5T01*	30	100	3	0,8	SY	81	02:05.547
22	*P50S5T02*	30	100	2	0,3	SY	64	00:33.921
23	*P50S5T03*	30	100	6	0,8	İAİY	79	00:44.484
24	*P50S5T04*	15	100	2	0,5	BAİY	0	00:00.704
25	*P50S5T05*	30	100	7	0,3	İAİY	19	00:04.781
26	*P50S5T06*	30	100	4	0,4	BAİY	2	00:01.734
27	*P50S5T07*	30	100	2	0,6	SY	90	01:47.234
28	*P50S5T08*	30	100	2	0,6	SY	36	00:34.812
29	*P50S5T09*	30	100	2	0,8	ÖY	81	00:57.703
30	*P50S5T10*	15	200	3	0,6	İAİY	71	01:07.375
31	*P1HS5T01*	15	50	2	0,5	SY	0	00:00.594
32	*P1HS5T02*	30	100	2	0,5	İAİY	19	00:13.265
33	*P1HS5T03*	30	100	5	0,6	SY	63	01:10.688
34	*P1HS5T04*	15	500	2	0,4	SY	0	00:00.156
35	*P1HS5T05*	30	100	4	0,3	SY	14	00:11.531
36	*P1HS5T06*	30	100	2	0,3	İAİY	0	00:00.266
37	*P1HS5T07*	30	100	3	0,2	SY	71	00:16.094
38	*P1HS5T08*	30	100	2	0,3	ÖY	12	00:17.469
39	*P1HS5T09*	30	100	5	0,6	İAİY	89	01:31.640
40	*P1HS5T10*	30	500	2	0,5	SY	0	00:00.375

Ek 2- 3 *P* tipi 8. aşama problemleri için optimizasyon sonuçları

No	Problem Tipi	Popülasyon	Iterasyon Sayısı	Alt Grup Sayısı	Doyum Oranı	İnşa Metodu	Iterasyon	İşlem Süreleri (dk:sn.sn'nin binde biri)
1	*P10S8T01*	15	200	4	0,5	İAİY	157	00:58.907
2	*P10S8T02*	15	50	2	0,1	SY	50	00:00.000
3	*P10S8T03*	15	200	3	0,4	SY	78	00:31.345
4	*P10S8T04*	15	500	7	0,2	İAİY	0	00:00.000
5	*P10S8T05*	15	200	2	0,1	İAİY	9	00:01.392
6	*P10S8T06*	30	100	5	0,7	İAİY	100	00:00.000
7	*P10S8T07*	15	50	5	0,8	SY	50	00:00.000
8	*P10S8T08*	15	50	2	0,2	SY	0	00:00.000
9	*P10S8T09*	15	50	4	0,2	SY	0	00:00.000
10	*P10S8T10*	30	100	6	0,8	BAİY	57	00:28.562
11	*P20S8T01*	15	50	2	0,8	SY	0	00:00.125
12	*P20S8T02*	30	100	6	0,3	BAİY	26	00:10.813
13	*P20S8T03*	30	100	3	0,8	ÖY	71	01:21.219
14	*P20S8T04*	30	500	6	0,7	İAİY	0	00:00.188
15	*P20S8T05*	30	100	7	0,8	BAİY	0	00:00.234
16	*P20S8T06*	15	50	2	0,8	İAİY	0	00:00.141
17	*P20S8T07*	30	1000	2	0,8	SY	0	00:00.235
18	*P20S8T08*	30	100	4	0,4	ÖY	76	00:37.093
19	*P20S8T09*	30	100	3	0,5	SY	6	00:04.828
20	*P20S8T10*	30	1000	2	0,1	SY	0	00:00.031

Ek 2- 3 *P* tipi 8. aşama problemleri için optimizasyon sonuçları (devamı)

No	Problem Tipi	Popülasyon	Iterasyon Sayısı	Alt Grup Sayısı	Doyum Oranı	İnşa Metodu	Iterasyon	İşlem Süreleri (dk:sn.sn'nin binde biri)
21	*P50S8T01*	15	100	4	0,6	SY	0	00:00.172
22	*P50S8T02*	15	100	3	0,4	İAİY	82	04:03.593
23	*P50S8T03*	30	500	3	0,5	SY	0	00:00.359
24	*P50S8T04*	15	500	3	0,6	BAİY	0	00:00.171
25	*P50S8T05*	15	100	2	0,5	BAİY	30	00:13.594
26	*P50S8T06*	15	100	3	0,4	ÖY	98	00:50.234
27	*P50S8T07*	15	100	2	0,5	SY	60	00:33.453
28	*P50S8T08*	15	100	4	0,3	İAİY	95	00:11.828
29	*P50S8T09*	15	150	2	0,7	İAİY	129	00:31.813
30	*P50S8T10*	15	100	2	0,3	ÖY	97	00:16.734
31	*P1HS8T01*	15	100	3	0,8	BAİY	62	14:55.688
32	*P1HS8T02*	15	100	4	0,5	SY	94	00:31.610
33	*P1HS8T03*	15	100	2	0,5	SY	100	00:00.000
34	*P1HS8T04*	15	100	3	0,3	SY	25	00:05.735
35	*P1HS8T05*	15	100	4	0,3	ÖY	82	00:50.985
36	*P1HS8T06*	15	100	4	0,5	BAİY	53	00:47.750
37	*P1HS8T07*	15	100	3	0,7	BAİY	99	02:18.532
38	*P1HS8T08*	30	250	8	0,4	SY	0	00:00.609
39	*P1HS8T09*	15	100	3	0,7	BAİY	85	00:57.891
40	*P1HS8T10*	15	100	5	0,8	BAİY	87	00:36.500

Ek 3 - 1 *Q* tipi 2. aşama problemleri için optimizasyon sonuçları

No	Problem Tipi	Popülasyon	Iterasyon Sayısı	Alt Grup Sayısı	Doyum Oranı	İnşa Metodu	Iterasyon	İşlem Süreleri (dk:sn.sn'nin binde biri)
1	*Q10S2T1*	15	200	3	0,7	ÖY	0	00:00.000
2	*Q10S2T2*	15	200	5	0,4	BAİY	0	00:00.000
3	*Q10S2T3*	15	200	2	0,5	BAİY	25	00:00.234
4	*Q10S2T4*	15	200	2	0,5	İAİY	0	00:00.000
5	*Q10S2T5*	15	100	2	0,2	ÖY	0	00:00.000
6	*Q10S2T6*	15	100	2	0,3	SY	0	00:00.000
7	*Q10S2T7*	15	200	3	0,4	BAİY	11	00:00.343
8	*Q10S2T8*	15	200	2	0,1	BAİY	0	00:00.000
9	*Q10S2T9*	15	200	2	0,3	BAİY	17	00:00.078
10	*Q10S2T10*	15	200	2	0,3	ÖY	17	00:00.062
11	*Q20S2T1*	30	100	8	0,4	SY	0	00:00.125
12	*Q20S2T2*	15	100	9	0,4	SY	0	00:00.000
13	*Q20S2T3*	15	200	5	0,4	BAİY	0	00:00.000
14	*Q20S2T4*	30	100	9	0,1	İAİY	3	00:00.125
15	*Q20S2T5*	15	100	4	0,8	İAİY	0	00:00.032
16	*Q20S2T6*	30	100	3	0,5	İAİY	18	00:01.890
17	*Q20S2T7*	30	100	7	0,1	ÖY	0	00:00.000
18	*Q20S2T8*	30	100	2	0,1	İAİY	0	00:00.000
19	*Q20S2T9*	30	100	2	0,1	İAİY	0	00:00.000
20	*Q20S2T10*	30	200	9	0,6	İAİY	0	00:00.047

Ek 3 - 1 *Q* tipi 2. aşama problemleri için optimizasyon sonuçları (devamı)

No	Problem Tipi	Popülasyon	Iterasyon Sayısı	Alt Grup Sayısı	Doyum Oranı	İnşa Metodu	Iterasyon	İşlem Süreleri (dk:sn.sn'nin binde biri)
21	*Q50S2T1*	15	200	2	0,1	İAİY	4	00:00.078
22	*Q50S2T2*	15	200	5	0,3	SY	148	00:13.259
23	*Q50S2T3*	30	250	7	0,5	SY	0	00:00.110
24	*Q50S2T4*	30	250	6	0,5	SY	0	00:00.156
25	*Q50S2T5*	15	200	5	0,7	İAİY	12	00:01.046
26	*Q50S2T6*	30	250	9	0,5	ÖY	0	00:00.110
27	*Q50S2T7*	15	200	4	0,2	SY	0	00:00.109
28	*Q50S2T8*	30	250	3	0,7	BAİY	0	00:00.156
29	*Q50S2T9*	30	250	2	0,3	BAİY	0	00:00.063
30	*Q50S2T10*	30	250	2	0,6	SY	0	00:00.141
31	*Q1HS2T1*	15	200	4	0,6	İAİY	154	00:24.422
32	*Q1HS2T2*	30	100	4	0,4	İAİY	0	00:00.203
33	*Q1HS2T3*	30	250	3	0,4	SY	0	00:00.234
34	*Q1HS2T4*	30	100	3	0,8	BAİY	0	00:00.500
35	*Q1HS2T5*	15	200	3	0,1	ÖY	108	00:23.672
36	*Q1HS2T6*	15	250	3	0,8	SY	0	00:00.265
37	*Q1HS2T7*	30	250	3	0,6	BAİY	0	00:00.438
38	*Q1HS2T8*	15	200	5	0,8	SY	52	00:28.438
39	*Q1HS2T9*	15	200	3	0,8	BAİY	141	02:58.468
40	*Q1HS2T10*	15	200	3	0,6	İAİY	165	01:19.234

Ek 3 - 2 *Q* tipi 5. aşama problemleri için optimizasyon sonuçları

No	Problem Tipi	Popülasyon	Iterasyon Sayısı	Alt Grup Sayısı	Doyum Oranı	İnşa Metodu	Iterasyon	İşlem Süreleri (dk:sn.sn'nin binde biri)
1	*Q10S5T1*	15	200	2	0,6	BAİY	22	00:00.563
2	*Q10S5T2*	15	200	2	0,4	SY	30	00:01.140
3	*Q10S5T3*	15	100	3	0,1	SY	0	00:00.000
4	*Q10S5T4*	15	100	5	0,4	İAİY	27	00:01.503
5	*Q10S5T5*	15	200	5	0,7	İAİY	131	00:16.125
6	*Q10S5T6*	15	50	3	0,2	ÖY	0	00:00.000
7	*Q10S5T7*	15	200	2	0,6	ÖY	17	00:00.500
8	*Q10S5T8*	15	200	4	0,3	İAİY	60	00:02.734
9	*Q10S5T9*	15	200	3	0,7	İAİY	8	00:00.875
10	*Q10S5T10*	15	200	4	0,7	ÖY	8	00:00.766
11	*Q20S5T1*	30	100	4	0,7	BAİY	93	00:58.781
12	*Q20S5T2*	15	50	4	0,8	SY	0	00:00.063
13	*Q20S5T3*	15	100	4	0,7	BAİY	0	00:00.047
14	*Q20S5T4*	30	250	9	0,3	SY	0	00:00.046
15	*Q20S5T5*	30	100	4	0,6	SY	70	00:39.687
16	*Q20S5T6*	15	500	7	0,5	SY	0	00:00.047
17	*Q20S5T7*	30	100	2	0,4	SY	83	00:09.828
18	*Q20S5T8*	30	100	3	0,7	İAİY	0	00:00.109
19	*Q20S5T9*	15	100	4	0,3	SY	0	00:00.031
20	*Q20S5T10*	15	100	6	0,1	İAİY	0	00:00.016

Ek 3- 2 *Q* tipi 5. aşama problemleri için optimizasyon sonuçları (devamı)

No	Problem Tipi	Popülasyon	Iterasyon Sayısı	Alt Grup Sayısı	Doyum Oranı	İnşa Metodu	Iterasyon	İşlem Süreleri (dk:sn.sn'nin binde biri)
21	*Q50S5T1*	15	200	4	0,8	BAİY	180	01:30.344
22	*Q50S5T2*	30	200	2	0,3	SY	0	00:00.125
23	*Q50S5T3*	15	200	5	0,8	ÖY	155	00:56.000
24	*Q50S5T4*	15	200	5	0,8	BAİY	73	01:15.266
25	*Q50S5T5*	15	150	4	0,6	BAİY	126	00:26.532
26	*Q50S5T6*	30	100	3	0,4	SY	0	00:00.203
27	*Q50S5T7*	15	200	5	0,6	SY	198	00:42.891
28	*Q50S5T8*	15	200	5	0,6	BAİY	162	01:02.594
29	*Q50S5T9*	15	200	4	0,5	BAİY	123	00:41.203
30	*Q50S5T10*	15	200	5	0,6	ÖY	172	00:46.906
31	*Q1HS5T1*	15	150	4	0,5	BAİY	133	00:33.641
32	*Q1HS5T2*	15	150	5	0,7	BAİY	108	00:57.797
33	*Q1HS5T3*	15	200	2	0,8	BAİY	166	03:41.358
34	*Q1HS5T4*	15	200	2	0,3	İAİY	193	02:01.359
35	*Q1HS5T5*	15	200	2	0,3	BAİY	140	01:21.000
36	*Q1HS5T6*	15	150	5	0,3	BAİY	132	00:21.656
37	*Q1HS5T7*	15	150	5	0,8	SY	115	01:07.140
38	*Q1HS5T8*	15	150	4	0,7	İAİY	119	01:20.500
39	*Q1HS5T9*	15	150	3	0,6	ÖY	116	01:33.938
40	*Q1HS5T10*	15	150	2	0,7	ÖY	139	01:14.921

Ek 3- 3 *Q* tipi 8. aşama problemleri için optimizasyon sonuçları

No	Problem Tipi	Popülasyon	Iterasyon Sayısı	Alt Grup Sayısı	Doyum Oranı	İnşa Metodu	Iterasyon	İşlem Süreleri (dk:sn.sn'nin binde biri)
1	*Q10S8T1*	15	200	3	0,7	BAİY	50	00:03.781
2	*Q10S8T2*	15	150	5	0,8	BAİY	109	00:23.875
3	*Q10S8T3*	15	150	2	0,3	SY	18	00:01.125
4	*Q10S8T4*	15	150	4	0,4	BAİY	78	00:09.110
5	*Q10S8T5*	15	200	3	0,1	BAİY	52	00:01.031
6	*Q10S8T6*	15	200	4	0,1	BAİY	139	00:02.250
7	*Q10S8T7*	15	200	5	0,1	İAİY	84	00:01.718
8	*Q10S8T8*	15	200	4	0,4	SY	68	00:20.209
9	*Q10S8T9*	15	200	5	0,5	ÖY	88	00:31.725
10	*Q10S8T10*	15	150	5	0,4	İAİY	137	00:08.562
11	*Q20S8T1*	30	250	6	0,6	SY	0	00:00.157
12	*Q20S8T2*	30	250	3	0,4	İAİY	0	00:00.110
13	*Q20S8T3*	30	250	5	0,7	ÖY	0	00:00.203
14	*Q20S8T4*	15	150	4	0,8	SY	102	00:31.625
15	*Q20S8T5*	30	250	7	0,6	İAİY	0	00:00.203
16	*Q20S8T6*	30	250	4	0,5	SY	0	00:00.156
17	*Q20S8T7*	15	150	4	0,5	İAİY	135	00:48.328
18	*Q20S8T8*	15	200	5	0,2	ÖY	188	00:18.977
19	*Q20S8T9*	15	500	4	0,5	İAİY	0	00:00.063
20	*Q20S8T10*	15	100	6	0,4	BAİY	0	00:00.047

Ek 3- 3 *Q* tipi 8. aşama problemleri için optimizasyon sonuçları (devamı)

No	Problem Tipi	Popülasyon	Iterasyon Sayısı	Alt Grup Sayısı	Doyum Oranı	İnşa Metodu	Iterasyon	İşlem Süreleri (dk:sn.sn'nin binde biri)
21	*Q50S8T1*	30	500	2	0,2	SY	0	00:00.140
22	*Q50S8T2*	30	50	5	0,8	BAİY	0	00:00.609
23	*Q50S8T3*	15	150	2	0,4	ÖY	108	01:06.516
24	*Q50S8T4*	15	150	4	0,7	İAİY	150	00:00.000
25	*Q50S8T5*	15	150	2	0,4	SY	68	00:12.875
26	*Q50S8T6*	15	500	9	0,8	SY	0	00:00.313
27	*Q50S8T7*	15	150	2	0,8	ÖY	122	00:45.797
28	*Q50S8T8*	30	250	3	0,5	SY	0	00:00.390
29	*Q50S8T9*	15	50	9	0,7	İAİY	0	00:00.218
30	*Q50S8T10*	15	150	3	0,8	İAİY	128	01:04.078
31	*Q1HS8T1*	15	50	4	0,8	SY	0	00:00.656
32	*Q1HS8T2*	15	100	5	0,8	ÖY	75	01:32.781
33	*Q1HS8T3*	15	200	2	0,8	İAİY	196	04:24.343
34	*Q1HS8T4*	15	100	4	0,4	İAİY	97	00:55.281
35	*Q1HS8T5*	15	100	5	0,6	SY	97	01:25.203
36	*Q1HS8T6*	30	150	2	0,8	İAİY	0	00:01.391
37	*Q1HS8T7*	15	100	3	0,7	İAİY	93	01:03.172
38	*Q1HS8T8*	15	200	3	0,7	SY	158	02:56.815
39	*Q1HS8T9*	15	100	3	0,4	İAİY	85	19:00.266
40	*Q1HS8T10*	15	200	4	0,7	İAİY	110	01:33.675