T.C.
ERCİYES ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
İŞLETME ANABİLİM DALI
ÜRETİM YÖNETİMİ VE PAZARLAMA BİLİM DALI

ESNEK AKIŞ TİPİ ÇİZELGELEME PROBLEMİNE SEZGİSEL BİR YAKLAŞIM VE BİR UYGULAMA

Hazırlayan
Emre ASLAN

Danışman
Prof. Dr. Osman UNUTULMAZ

Doktora Tezi

Haziran 2014
KAYSERİ

T.C.
ERCİYES ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
İŞLETME ANABİLİM DALI
ÜRETİM YÖNETİMİ VE PAZARLAMA BİLİM DALI

ESNEK AKIŞ TİPİ ÇİZELGELEME PROBLEMİNE SEZGİSEL BİR YAKLAŞIM VE BİR UYGULAMA

(Doktora Tezi)

Hazırlayan
Emre ASLAN

Danışman
Prof. Dr. Osman UNUTULMAZ

Haziran 2014
KAYSERİ

BİLİMSEL ETİĞE UYGUNLUK

Bu çalışmadaki tüm bilgilerin, akademik ve etik kurallara uygun bir şekilde elde edildiğini beyan ederim. Aynı zamanda bu kural ve davranışların gerektirdiği gibi, bu çalışmanın özünde olmayan tüm materyal ve sonuçları tam olarak aktardığımı ve referans gösterdiğimi belirtirim.

Emre ASLAN

"Esnek Akış Tipi Çizelgeleme Problemine Sezgisel Bir Yaklaşım ve Bir Uygulama" adlı Doktora tezi, Erciyes Üniversitesi Lisansüstü Tez Önerisi ve Tez Yazma Yönergesi'ne uygun olarak hazırlanmıştır.

Tezi Hazırlayan

Emre ASLAN

Danışman

Prof. Dr. Osman UNUTULMAZ

İşletme ABD Başkanı

Prof. Dr. Osman UNUTULMAZ

Prof. Dr. Osman UNUTULMAZ danışmanlığında Emre ASLAN tarafından hazırlanan "Esnek Akış Tipi Çizelgeleme Problemine Sezgisel Bir Yaklaşım ve Bir Uygulama" adlı bu çalışma jürimiz tarafından Erciyes Üniversitesi Sosyal Bilimler Enstitüsü Üretim Yönetimi ve Pazarlama Anabilim Dalında **Doktora** tezi olarak kabul edilmiştir.

23/06/2014

JÜRİ:

Danışman	: Prof. Dr. Osman UNUTULMAZ	.....
Üye	: Prof. Dr. Şevki ÖZGENER	.....
Üye	: Doç. Dr. Lale ÖZBAKIR	.....
Üye	: Doç. Dr. Mehmet Sıtkı İLKAY	.....
Üye	: Yrd. Doç. Dr. Yasemin YAVUZ	.....

ONAY:
Bu tezin kabulü Enstitü Yönetim Kurulunun 14/07/2014 tarih ve ..16..... sayılı kararı ile onaylanmıştır.

14/07/2014

Prof. Dr. Lütfullah CEBECİ

Enstitü Müdürü

TEŞEKKÜR

Doktora sürecinin başından itibaren tatlı sert uyarılarıyla birlikte desteğini esirgemeyen danışmanım Prof. Dr. Osman Unutulmaz'a,

komite toplantılarında ve onun haricinde sorularımı yanıtlayarak yol gösteren Doç. Dr. Mehmet S. İlkay ve Doç. Dr. Lale Özbakır'a,

matematiksel modelin çözülmesi ve GAMS konusundaki yardımlarından dolayı Yrd. Doç. Dr. Yasemin Yavuz'a,

uygulama problemindeki işletme ile bağlantı kurulması ve verilerin elde edilmesi konusundaki yardımlarından dolayı Yrd. Doç. Dr. Alperen M. Yiğit'e,

tezimi gözden geçirdiği için Arş. Gör. Türker Şimşek'e

ve varlıklarından güç aldığım ailem ve eşime,

teşekkür ederim.

ESNEK AKIŞ TİPİ ÇİZELGELEME PROBLEMİNE
SEZGİSEL BİR YAKLAŞIM
VE BİR UYGULAMA

Emre ASLAN
Erciyes Üniversitesi, Sosyal Bilimler Enstitüsü
Doktora Tezi, Haziran 2014
Danışman: Prof. Dr. Osman UNUTULMAZ

KISA ÖZET

Esnek akış tipi çizelgeleme problemi birbirine seri aşamaları olan, bu aşamalardan en az birinde paralel makineler bulunan ve işlerin aynı rotayı takip ettiği çizelgeleme problemi tipidir. Akış tipi çizelgeleme probleminin paralel makineli hali veya paralel makine çizelgeleme probleminin çok aşamalı hali olarak da düşünülebilir.

Bu tezde, esnek akış tipi çizelgeleme probleminde aşama atlama, partilere ayırma, sıra bağımlı hazırlık zamanları, makine uygunluk ve iş öncelik kısıtları birlikte ele alınmıştır. Ofis mobilyaları üreten bir tesiste uygulaması olan bu çalışmada 6 aşama ve 66 iş vardır.

Problem için öncelikle karma tamsayılı programlama modeli geliştirilmiş, model GAMS dilinde yazılmış ve CPLEX çözücü ile çözüm aranmıştır. 2 saat, 1 gün ve 3 gün boyunca maksimum tamamlanma zamanı ve toplam gecikmeyi minimize etmeye çalışan iki ayrı amaç fonksiyonu için çalıştırılmış fakat optimum çözüm bulunamamıştır. Bu nedenle söz konusu problem için çok daha kısa sürelerde kabul edilebilir sonuçlar vermesi beklenen sıralama kurallarına dayalı bazı sezgisel modeller geliştirilerek kullanılmıştır. Uyarlanmış Johnson Kuralı (UJK), En Az Boşluk Kuralı (EABK) ve bu kurallarla karşılaştırma yapılabilmesi için En Kısa İşlem Zamanı (EKİZ) ve En Uzun İşlem Zamanı (EUİZ) kuralları tezin uygulama problemine ve oluşturulan farklı büyüklükteki test problemlerine uygulanmış ve gerekli karşılaştırmalar yapılmıştır.

UJK sezgiselinin C_{max} performans kriterinde en iyi sonuçları verdiği tespit edilmiştir. UJK sezgiseli uygulama probleminde matematiksel modelin 3 gün çalışarak ulaştığı C_{max} değerinden %34 daha düşük bir değeri 1 saniyeden daha kısa sürede elde edebilmektedir. Bununla beraber EABK sezgiselinin büyük boyutlu problemlerde gecikme ile ilgili kriterler dikkate alındığında öne çıktığı görülmüştür. Problemlerde ilk aşama sonrası partilere ayırma alternatifi denenmiş ve bu duruma özel Partilere Ayrılmış En Az Boşluk Kuralı (PEABK) geliştirilmiştir. PEABK gecikme ile ilgili kriterlerde hem küçük hem de büyük boyutlu problemlerde EABK sezgiselinden daha iyi sonuçlar vermiştir.

Anahtar Kelimeler: Esnek akış tipi çizelgeleme, öncelik kısıtları, makine uygunluk kısıtları, karma tamsayılı programlama, sıralama kuralları

A HEURISTIC APPROACH AND AN APPLICATION TO FLEXIBLE FLOWSHOP SCHEDULING PROBLEM

Emre ASLAN
Erciyes University, Institute for Social Sciences
PhD Thesis, June 2014
Supervisor: Prof. Dr. Osman UNUTULMAZ

ABSTRACT

The flexible flow shop scheduling problem is such a flow shop scheduling problem that has serial stages, at least one of the serial stages has parallel machines and the jobs follow the same path. The flexible flow shop scheduling problem may be considered as a flow shop scheduling problem with parallel machines or a multi-stage parallel machine scheduling problem.

In this thesis; stage skipping, lot splitting, sequence-dependent setup times, machine eligibility and the precedence constraints are taken into account all together at the same time while developing the methods for solving the flexible flow shop scheduling problem. The identified model is being employed to solve the existing problem of an office furniture producing facility that is composed of 6 stages and 66 jobs.

At the beginning a mixed integer programing model is developed. This model is coded in GAMS programming language and the solution is searched employing CPLEX solver. It is run for 2 hours, 1 day and 3 days for two different objective functions minimizing makespan and total tardiness. However, the optimum solution is not reached. Then with the expectation of reaching reasonable solutions within a short time period heuristic methods based on priority rules are developed and employed. Modified Johnson Rule, Minimum Slack Rule and to be able to compare the results Shortest Processing Time and Longest Processing Time rules are also employed while solving the office furniture producing facility problem and the similar problems of different size. Finally such obtained results are compared.

It is observed that the Modified Johnson Heuristic gives the best result on the C_{max} performance criteria. In fact, the Modified Johnson heuristic finds a C_{max} value, within a second, which is 34% smaller than the C_{max} value obtained by the mathematical model in 3 days. However, under the tardiness criterion, Minimum Slack Rule heuristic is prominent for large size problems. An alternative of lot splitting after the first step was experimented and Lot-Splitted Minimum Slack Rule was developed special to this situation. Lot-Splitted Minimum Slack Rule obatined better values than Minimum Slack Rule according to tardiness-related criteria on both small and large size problems.

Keywords: Flexible flowshop scheduling, precedence constraints, machine eligibility constraints, mixed integer programming, priority rules.

İÇİNDEKİLER

BİLİMSEL ETİĞE UYGUNLUK ..... **i**
YÖNERGEYE UYGUNLUK ..... **ii**
KABUL VE ONAY ..... **iii**
TEŞEKKÜR… ..... **iv**
KISA ÖZET… ..... **v**
ABSTRACT… ..... **iii**
İÇİNDEKİLER ..... **v**
TABLOLAR LİSTESİ ..... **viii**
ŞEKİLLER LİSTESİ ..... **xi**

GİRİŞ 1

1. BÖLÜM

1 ÇİZELGELEME **4**
1.1 Üretimde Çizelgeleme 5
1.2 Çizelgelerin Grafiksel Gösterimi 7
1.2.1 Ayrık Graf 8
1.2.2 Gannt Şeması 9
1.3 Çizelge Sınıfları 10
1.4 Çizelgelerde Performans Kriterleri 13
1.5 Çizelgeleme Problemlerinin Sınıflandırılması 14
1.5.1 Notasyon 14
1.5.2 Sınıflandırma 17
1.6 Çözüm Yaklaşımları ve Karmaşıklık 22
1.6.1 Optimizasyon 22
1.6.1.1 Etkin Yöntemler 22
1.6.1.2 Sayım Yöntemleri 23
1.6.2 Karmaşıklık 24
1.6.3 Yaklaşık Yöntemler 29
1.6.3.1 Sezgisel Yöntemler 29
1.6.3.2 Metasezgisel Yöntemler 31

2. BÖLÜM

2 ESNEK AKIŞ TİPİ ÇİZELGELEME PROBLEMİ **34**
2.1 Klasik Akış Tipi Çizelgeleme Problemi 34

2.1.1 Klasik Akış Tipi Çizelgeleme Problemi için Matematiksel Model 36
2.2 Esnek Akış Tipi Çizelgeleme Problemi 38
2.2.1 Esnek Akış Tipi Çizelgeleme Problemi için Matematiksel Model 40
2.2.2 Esnek Akış Tipi Çizelgeleme Problemlerinde Karmaşıklık 42
2.2.3 Literatür 43
2.2.3.1 2 Aşamalı Esnek Akış Tipi Çizelgeleme Çalışmaları 56
2.2.3.1.1 Tek Aşamasında Özdeş Paralel Makineler Olan Çalışmalar 56
2.2.3.1.2 Her İki Aşamasında Özdeş Paralel Makineler Olan Çalışmalar 59
2.2.3.2 3 Aşamalı Esnek Akış Tipi Çizelgeleme Çalışmaları 62
2.2.3.3 Çok Aşamalı Esnek Akış Tipi Çizelgeleme Çalışmaları 64
2.2.3.3.1 Özdeş Paralel Makineli Çalışmalar 64
2.2.3.3.2 Farklı Hızda Paralel Makineli Çalışmalar 71
2.2.3.3.3 İlişkisiz Paralel Makineli Çalışmalar 71

3. BÖLÜM

3 UYGULAMA 74
3.1 İşletmenin ve Üretim Sürecinin Tanıtımı 74
3.2 Çizelgelenecek İşler 78
3.2.1 İşlerin İşlem Sürelerinin Hesaplanması 84
3.2.1.1 Kesim Süreleri 84
3.2.1.2 Bantlama Süreleri 86
3.2.1.3 CNC Süreleri 87
3.2.1.4 Delik İşlemi Süreleri 88
3.2.1.5 Montaj Süreleri 89
3.2.1.6 Paketleme Süreleri 91
3.3 Uygulama Problemi için Varsayımlar 94
3.4 Uygulama Problemi için Matematiksel Model 95
3.4.1 Matematiksel Modelin Çözümü 100
3.5 Uygulama Problemi İçin Sezgisel Yöntemler 100
3.5.1 Sıralama Kuralları 101
3.5.1.1 En Kısa İşlem Zamanı 101
3.5.1.2 En Uzun İşlem Zamanı 103
3.5.1.3 Uyarlanmış Johnson Kuralı 104
3.5.1.4 En Az Boşluk Kuralı 106

3.5.2 Test Problemleri 107
3.6 Bulgular 108
3.6.1 Yöntemlerin Sonuçlarının Karşılaştırılması 108
3.6.2 Yöntemlerin Sapma Yüzdelerinin Karşılaştırılması 113
3.6.3 Kesim İşlemi Sonrası Partilere Ayırma 116
SONUÇ……..... 121
EKLER……..... 125
EK-1: GAMS DİLİNDE MATEMATİKSEL MODELİN YAZILIŞI 125
EK-2: SEZGİSEL SIRALAMA KURALLARI ALGORİTMALARI 128
EK-3: SEZGİSEL SIRALAMA KURALLARININ C# KODLARI 137
EK-4: TEST PROBLEMLERİ 155
EK-5: TEST PROBLEMLERİNİN MATEMATİKSEL VE SEZGİSEL …….. MODELLERLE ÇÖZÜM SONUÇLARI 170
EK-6: FFS6J66 PROBLEMİNİN MATEMATİKSEL (C_{max}), UJK, EABK ve PEABK YÖNTEMLERİNE GÖRE ÇÖZÜMLERİNİN GANNT. ŞEMALARI 192
KAYNAKLAR 201
ÖZ GEÇMİŞ……..... 217

TABLOLAR LİSTESİ

Tablo 1: Polinom Zamanlı Çözülebilen Problemler 26
Tablo 2: Sıradan Anlamda *NP*-zor Problemler 27
Tablo 3: Güçlü Anlamda *NP*-zor Problemler 27
Tablo 4: İncelenen Çalışmaların Aşama ve Paralel Makine Durumuna Göre Yüzdesi 46
Tablo 5: Esnek Akış Tipi Çizelgeleme İle İlgili Literatür Sınıflandırması 49
Tablo 6: Çizelgeleme İçin Seçilen Ürünler, Türleri ve Adetleri 79
Tablo 7: Ürünler, Ürün Bileşenleri ve Özellikleri 79
Tablo 8: Ürün Bileşenlerinin Rotaları 81
Tablo 9: Kesme İşlemi İçin İşlerin İşlem Süreleri 84
Tablo 10: Bantlama İşlemi İçin İşlerin İşlem Süreleri 86
Tablo 11: CNC Makinesinde İşlerin İşlem Süreleri 88
Tablo 12: Delik İşlemi İçin İşlerin İşlem Süreleri 88
Tablo 13: Montaj İçin Uygun Parti Büyüklükleri 90
Tablo 14: Montaj İçin İşlerin İşlem Süreleri 90
Tablo 15: Paketleme İçin İşlerin İşlem Süreleri 91
Tablo 16: Belirlenen Parti Büyüklükleri İçin İşlem Süreleri (dk.) 93
Tablo 17: Matematiksel Çözüm Sonuçları 100
Tablo 18: FFS3J5 problemi işlem süreleri ve teslim tarihleri (dk.) 101
Tablo 19: FFS3J5 problemi 1. aşamadaki sıra bağımlı hazırlık süreleri (dk.) 102
Tablo 20: FFS3J5 problemi aşamalardaki makine sayıları 102
Tablo 21: FFS3J5 problemi 2. aşamadaki makine uygunluk kısıtları 102
Tablo 22: Aşama ve İş Sayılarına Göre Test Problemlerinin İsimleri 108
Tablo 23: Matematiksel ve sezgizel yöntem sonuçlarının karşılaştırılması 109
Tablo 24: Problemlere göre optimum ya da bilinen en iyi değerden sapma yüzdeleri 113
Tablo 25: Performans kriterlerine göre optimum ya da bilinen en iyi değerden sapma yüzdeleri 115
Tablo 26: Partilere Ayrılmış FFS3J5 problemi işlem süreleri ve teslim tarihleri (dk.) 117
Tablo 27: Partilere Ayrılmış FFS3J5 problemi 2. aşamadaki makine uygunluk kısıtları 117
Tablo 28: PEABK sezgiselinin diğer yöntemlerle sonuçlarının karşılaştırılması 119
Tablo 29: FFS3J5 problemi işlem süreleri ve teslim tarihleri (dk.) 155

Tablo 30: FFS3J5 problemi 1. aşamadaki sıra bağımlı hazırlık süreleri (dk.) 155
Tablo 31: FFS3J5 problemi aşamalardaki makine sayıları 155
Tablo 32: FFS3J5 problemi 2. aşamadaki makine uygunluk kısıtları 155
Tablo 33: FFS3J15 problemi işlem süreleri ve teslim tarihleri (dk.) 156
Tablo 34: FFS3J15 problemi 1. aşamadaki sıra bağımlı hazırlık süreleri (dk.) 156
Tablo 35: FFS3J15 problemi aşamalardaki makine sayıları 156
Tablo 36: FFS3J15 problemi 2. aşamadaki makine uygunluk kısıtları 156
Tablo 37: FFS3J30 problemi işlem süreleri ve teslim tarihleri (dk.) 157
Tablo 38: FFS3J30 problemi aşamalardaki makine sayıları 157
Tablo 39: FFS3J30 problemi 2. aşamadaki makine uygunluk kısıtları 157
Tablo 40: FFS3J30 problemi 1. aşamadaki sıra bağımlı hazırlık süreleri (dk.) 158
Tablo 41: FFS6J5 problemi işlem süreleri ve teslim tarihleri (dk.) 159
Tablo 42: FFS6J5 problemi 1. aşamadaki sıra bağımlı hazırlık süreleri (dk.) 159
Tablo 43: FFS6J5 problemi aşamalardaki makine sayıları 159
Tablo 44: FFS6J5 problemi 2. aşamadaki makine uygunluk kısıtları 159
Tablo 45: FFS6J15 problemi işlem süreleri ve teslim tarihleri (dk.) 160
Tablo 46: FFS6J15 problemi 1. aşamadaki sıra bağımlı hazırlık süreleri 160
Tablo 47: FFS6J15 problemi aşamalardaki makine sayıları 160
Tablo 48: FFS6J15 problemi 2. aşamadaki makine uygunluk kısıtları 160
Tablo 49: FFS6J30 problemi işlem süreleri ve teslim tarihleri (dk.) 161
Tablo 50: FFS6J30 problemi aşamalardaki makine sayıları 161
Tablo 51: FFS6J30 problemi 2. aşamadaki makine uygunluk kısıtları 161
Tablo 52: FFS6J30 problemi 1. aşamadaki sıra bağımlı hazırlık süreleri (dk.) 162
Tablo 53: FFS6J66 problemi işlem süreleri ve teslim tarihleri 163
Tablo 54: FFS6J66 problemi aşamalardaki makine sayıları 164
Tablo 55: FFS6J66 problemi 2. aşamada makine uygunluk kriterleri 164
Tablo 56: FFS6J66 problemi 2. aşamada 1. makinede hazırlık süreleri 164
Tablo 57: Partilere Ayrılmış FFS3J5 problemi işlem süreleri ve teslim tarihleri (dk.) 165
Tablo 58: Partilere Ayrılmış FFS3J5 problemi 2. aşamadaki makine uygunluk kısıtları 165
Tablo 59: Partilere Ayrılmış FFS3J15 problemi işlem süreleri ve teslim tarihleri (dk.) 165

Tablo 60: Partilere Ayrılmış FFS3J15 problemi 2. aşamadaki makine uygunluk kısıtları 165
Tablo 61: Partilere Ayrılmış FFS3J30 problemi işlem süreleri ve teslim tarihleri (dk.) 166
Tablo 62: Partilere Ayrılmış FFS3J30 problemi 2. aşamadaki makine uygunluk kısıtları 166
Tablo 63: Partilere Ayrılmış FFS6J5 problemi işlem süreleri ve teslim tarihleri (dk.) 167
Tablo 64: Partilere Ayrılmış FFS6J5 problemi 2. aşamadaki makine uygunluk kısıtları 167
Tablo 65: Partilere Ayrılmış FFS6J15 problemi işlem süreleri ve teslim tarihleri (dk.) 167
Tablo 66: Partilere Ayrılmış FFS6J15 problemi 2. aşamadaki makine uygunluk kısıtları 167
Tablo 67: Partilere Ayrılmış FFS6J30 problemi işlem süreleri ve teslim tarihleri (dk.) 168
Tablo 68: Partilere Ayrılmış FFS6J30 problemi 2. aşamadaki makine uygunluk kısıtları 168
Tablo 69: Partilere Ayrılmış FFS6J66 problemi işlem süreleri ve teslim tarihleri (dk.) 169

ŞEKİLLER LİSTESİ

Şekil 1: Bir İmalat Sisteminde Bilgi Akış Şeması 7
Şekil 2: Ayrık Graf 8
Şekil 3: Uygun Bir Çizelge 9
Şekil 4: Makine Odaklı Gannt Şeması 9
Şekil 5: İş Odaklı Gannt Şeması 10
Şekil 6 : İşlem Bölmesiz Çizelgelerin Sınıflandırılması 12
Şekil 7 : Ertelemesiz Olmayan Aktif Bir Çizelge 12
Şekil 8 : Aktif Olmayan Yarı Aktif Bir Çizelge 13
Şekil 9: Çizelgeleme Problemleri için Karmaşıklık Hiyerarşisi (a) Makine ortamları .....
(b) İşlem kısıtlamaları (c) Amaç fonksiyonları 28
Şekil 10: Akış Tipinde Bir İşin Öncelik Yapısı 34
Şekil 11: Klasik Akış Tipinde İş Akışı 35
Şekil 12: Genel Akış Tipi Bir Atölyede İş Akışı 35
Şekil 13: Esnek Akış Tipinde İş Akışı 39
Şekil 14: Yıllara Göre Esnek Akış Tipi Çizelgeleme Çalışma Sayıları 46
Şekil 15: Amaç Fonksiyonlarının Dağılımı 47
Şekil 16: Kullanılan Tekniklerin Dağılımı 48
Şekil 17: Literatür Sınıflandırması 55
Şekil 18: Fabrika Yerleşimi 75
Şekil 19: 60 Adet A ürünü için Akış Şeması 83
Şekil 20: EKİZ Kuralına Göre FFS3J5 Probleminin Gannt Şeması 103
Şekil 21: EUİZ Kuralına Göre FFS3J5 Probleminin Gannt Şeması 104
Şekil 22: UJK Sezgiseline Göre FFS3J5 Probleminin Gannt Şeması 105
Şekil 23: EABK Sezgiseline Göre FFS3J5 Probleminin Gannt Şeması 107
Şekil 24: $C_{\max}$ Kriterine Göre Yöntemlerin Sonuçlarının Karşılaştırması 110
Şekil 25: $\sum T_j$ Kriterine göre Yöntemlerin Sonuçlarının Karşılaştırması 110
Şekil 26: FFS6J66 Probleminin Matematiksel Model ($C_{\max}$), UJK ve EABK .....
yöntemlerine göre çözümlerinin Gannt Şemaları 112
Şekil 27: PEABK Sezgiseline Göre FFS3J5 Probleminin Gannt Şeması 118
Şekil 28: FFS6J66 probleminin EABK ve PEABK yöntemlerine göre çözümlerinin .....
Gannt Şemaları 120

Şekil 29: FFS6J66 Probleminin Matematiksel Model (C_{max}) ve UJK yöntemine .....
göre çözümlerinin Gannt Şemaları (0-300 dk.) 192
Şekil 30: FFS6J66 Probleminin Matematiksel Model (C_{max}) ve UJK yöntemine .....
göre çözümlerinin Gannt Şemaları (300-600 dk.) 193
Şekil 31: FFS6J66 Probleminin Matematiksel Model (C_{max}) ve UJK yöntemine .....
göre çözümlerinin Gannt Şemaları (600-900 dk.) 194
Şekil 32: FFS6J66 Probleminin Matematiksel Model (C_{max}) ve UJK yöntemine .....
göre çözümlerinin Gannt Şemaları (900-1200 dk.) 195
Şekil 33: FFS6J66 Probleminin Matematiksel Model (C_{max}) ve UJK yöntemine .....
göre çözümlerinin Gannt Şemaları (1200-1278 dk.) 196
Şekil 34: FFS6J66 Probleminin EABK ve PEABK yöntemlerine göre çözümlerinin .....
Gannt Şemaları (0-300 dk.) 197
Şekil 35: FFS6J66 Probleminin EABK ve PEABK yöntemlerine göre çözümlerinin .....
Gannt Şemaları (300-600 dk.) 198
Şekil 36: FFS6J66 Probleminin EABK ve PEABK yöntemlerine göre çözümlerinin .....
Gannt Şemaları (600-900 dk.) 199
Şekil 37: FFS6J66 Probleminin EABK ve PEABK yöntemlerine göre çözümlerinin .....
Gannt Şemaları (900-1027 dk.) 200

GİRİŞ

Günümüzde müşterilerin zevk ve beğenilerinin sürekli değişmesi ve taleplerinin çeşitlilik göstermesi, işletmelerin rekabet edebilmek için bu talepleri karşılayacak ürün çeşitliliğine sahip olmasını ve taleplere hızlı bir şekilde cevap vermesini gerektirmektedir. Müşteri taleplerinin günümüzdeki kadar çeşitli olmadığı dönemlerde işletmeler yeterli mamul stoğu bulundurarak bu taleplere hızlı cevap verebilmekteydi. Fakat günümüzdeki talep yelpazesine bağlı ürün çeşitliliği, ürün yaşam döngülerinin kısalığı, ürünlerin kısa sürede demode olup yeni modellerin piyasaya sürülmesi gerekliliği gibi nedenler stok tutmayı işletmeler için çok maliyetli hale getirmektedir. Müşterilerin sürekli değişen taleplerine üretim sisteminin hızlı cevap vermesi günümüz şartlarında rekabet için bir gerekliliktir.

Belirli bir zaman periyodunda belirli sipariş ve işleri üretmesi gereken bir işletmenin bunları işleme sırası; siparişlerin zamanında teslimi, toplam üretim zamanı, işlerin ve makinelerin bekleme süreleri, süreç içi envanter gibi pek çok kriteri etkilemektedir. Bu bağlamda bir işletmenin üreteceği işleri belli kriterleri dikkate alarak sıralayacak, operasyonların hangi makinede ne zaman işlenmeye başlanıp ne zaman biteceğini belirleyecek, dahası alternatif sıralamalar oluşturup bunları performans kriterlerine göre karşılaştıracak çizelgeleme araçları kullanması işletme açısından çok faydalı olacaktır.

Çizelgeleme genel olarak kaynakların görevlere tahsisi, bir üretim işletmesi özelinde ise makinelerin işlere tahsisi olarak kabaca tarif edilebilir. Operasyonların hangi makinelerde hangi sırayla işleneceği, işlenmeye ne zaman başlanacağı gibi sorulara cevap arar. İşlerin sistemde izlediği rotaya ve makine ortamına (makine sayısı ve dizilimine) göre çizelgeleme problemleri belli sınıflara ayrılmaktadır. Bir üretim sisteminde ürün çeşidi ve sürecin karmaşıklığı arttıkça buna bağlı olarak çizelgelenecek işlerin ve bu işlerin işlem göreceği makine veya aşamaların sayısı da artmakta, çizelgeleme probleminin boyutu büyümektedir. Çizelgeleme probleminin boyutu

büyüdükçe zorluk derecesi de üstel olarak artmaktadır. Bu tarz problemlerde problem boyutu büyüdükçe matematiksel modeller optimum çözümü bulmakta yetersiz kalmaktadır. Bu durumda optimum çözüm garanti etmeyen fakat kısa süre içinde kabul edilebilir sonuçlar veren sezgisel veya metasezgisel yöntemler öne çıkmaktadır.

Bu tezde, birbirine seri aşamaları olan, bu aşamalardan en az birinde paralel makineler bulunan ve işlerin aynı rotayı takip ettiği *esnek akış tipi çizelgeleme problemi*ne kapsamında aşama atlama, partilere ayırma, sıra bağımlı hazırlık zamanları, makine uygunluk ve iş öncelik kısıtları ele alınmıştır. Tezin uygulaması ofis mobilyaları üreten bir işletmede yapılmıştır. Uygulama probleminde 6 aşama ve 66 iş vardır.

Çizelgeleme ile ilgili literatürde, uygulama problemleri ile ilgilenen çalışmalardan ziyade belli problem tiplerindeki test problemleri üzerinde daha hızlı veya daha iyi sonuçlar verecek yeni yöntemler geliştiren çalışmalar vardır. Literatürde sıkça rastlanan çalışmaların aksine, bu tezin amacı bir işletmedeki uygulama problemini tanımlayıp, bu probleme çözüm getirecek yöntem geliştirmektir. Tezin uygulama problemi aşama atlama, partilere ayırma, sıra bağımlı hazırlık zamanları, makine uygunluk ve iş öncelik kısıtlarını birlikte ele alan, aşamalarında özdeş paralel makineler bulunan bir esnek akış tipi çizelgeleme problemidir.

Tezin 2. bölümünde çizelgeleme ile ilgili genel bilgiler verilmiş ve tanımlamalar yapılmıştır. Çizelgeleme problemlerinin sınıflandırılması ve gösterimine değinilmiş ve çözüm yaklaşımlarından bahsedilmiştir. İki temel çözüm yaklaşımı, optimizasyon ve yaklaşık yöntemlerdir. Optimizasyon başlığı altında etkin yöntemler ve sayım yöntemlerinden, yaklaşık yöntemler başlığı altında ise sezgisel ve metasezgisel yöntemlerden bahsedilmiştir.

3. bölümde esnek akış tipi çizelgeleme daha geniş olarak tanımlanmış, esnek akış tipi çizelgeleme problemlerinin karmaşıklığına değinilmiş ve bu problem ile ilgili literatür incelemesi yapılmıştır. Literatürdeki çalışmalar hakkında genel istatistiklere değinilip tezin esnek akış tipi çizelgeleme problemindeki ek kısıtlarından en az birisini içeren çalışmalardan bahsedilmiştir.

4. bölümde uygulama problemine konu olan işletme tanıtılmış, çizelgeleme problemi için seçilen ürünler, bu ürünlerin alt parçaları olan işler ve bu işlerin işlem sürelerinin

nasıl hesaplandığı açıklanmıştır. İşler, işlem süreleri, işlerin rotaları, teslim tarihleri gibi bilgiler verildikten sonra, tanımlanan problem için kurulan matematiksel model verilmiştir. Matematiksel model; maksimum tamamlanma zamanı ve toplam gecikmeyi minimize etmeyi amaçlayan iki farklı amaç fonksiyonu için GAMS dilinde yazılıp CPLEX ile çözülmüştür. Matematikel modeller ile uygulama probleminin optimum çözümü bulunamadığı için sezgisel yöntemler geliştirilmiş ve uygulama probleminin özelliklerini yansıtan, farklı iş ve aşama sayıları olan test problemleri oluşturulmuştur. İki aşamalı akış tipi çizelgeleme problemlerinde optimum çözüm veren Johnson Kuralı modifiye edilip bu problem tipine uyarlanmış (Uyarlanmış Johnson Kuralı), bunun dışında gecikme ile ilgili performans kriterleri için En Az Boşluk Kuralı probleme uyarlanarak kullanılmıştır. Bu yöntemlerle karşılaştırma yapılabilmesi için En Kısa İşlem Zamanı ve En Uzun İşlem Zamanı kurallarına da yer verilmiştir. En Az Boşluk Kuralı küçük boyutlu problemlerde iyi performans gösteremediği için problemin yapısında bir iyileştirilmeye gidilmiş ve Partilere Ayrılmış En Az Boşluk Kuralı geliştirilerek kullanılmıştır. Son olarak hem tez problemi hem de oluşturulan 3 ve 6 aşamalı, 5, 15 ve 30 işi olan test problemleri üzerinde matematiksel modellerin ve sezgisel yöntemlerin verdiği sonuçlar karşılaştırılmıştır.

Sonuç bölümünde bulgular çerçevesinde matematiksel ve sezgisel yöntemlere dair yorumlarla beraber uygulamanın yapıldığı işletmeye ve gelecek çalışmalara yönelik önerilere yer verilmiştir.

1. BÖLÜM

1 ÇİZELGELEME

Bir *çizelge,* genellikle olayların ne zaman olmasının beklendiğini anlatır, belli aktivitelerin zamanlaması için bir plan ortaya çıkarır ve "Eğer her şey yolunda giderse, belli bir olay ne zaman olacak?" sorusunun cevabını verir. "Ne zaman" sorusuna cevaplar genellikle *zamanlama* hakkında bir bilgiyle gelir. Fakat, zamanlama yerine *sıralama* da kullanılabilir (Baker and Trietsch, 2009, p.1). *Sıra*, işlerin makinelerdeki işlem görme düzenidir. Bu nedenle bir işlem sırası çeşitli operasyonların başlama ve bitiş zamanları hakkında açık bir bilgi vermez. Fakat *çizelge* sıralama bilgisinin yanında zaman programını da içerir (French, 1982, p.26).

Çizelgeleme, pek çok üretim ve hizmet endüstrisinde düzenli olarak kullanılan bir karar verme sürecidir. Belirli bir zaman periyodunda, kaynakların görevlere tahsis edilmesi ile ilgilenir ve bir veya daha fazla amacı optimize etmekle uğraşır. Bir organizasyonda kaynaklar ve görevler farklı şekillerde olabilir. Kaynaklar; bir atölyede makineler, havaalanında pistler, inşaat alanında işçiler, bilgisayarda işlemci olabilir. Görevler ise üretim sürecindeki operasyonlar, havaalanında kalkışlar ve inişler, inşaat projesinde aşamalar ve bilgisayar programında yapılan işlemler olabilir. Her görevin belli bir öncelik seviyesi, olası en erken başlama zamanı ve bir teslim tarihi vardır. Amaçlar da farklı şekillerde olabilir. Bir amaç son görevin tamamlanma zamanının minimizasyonu olabilirken, bir başkası teslim tarihinden sonra tamamlanan görevlerin sayısının minimizasyonu olabilir (Pinedo, 2008, p.1).

Uygulamada, çizelge seçiminin sistemin performansı üzerinde önemli etkileri mevcuttur. Rastgele bir çizelge seçmek yerine uygun bir çizelge aramak için bir miktar zaman ve çaba harcamanın faydası vardır. Fakat çizelgeleme teknik ve uygulama

açılarından zor olabilmektedir. Teknik olarak karşılaşılan zorluklar kombinatoryal optimizasyon[1] ve modelleme ile ilgili karşılaşılan zorluklardır. Uygulama kısmındaki zorluklar ise tamamen farklıdır. Bunlar mevcut çizelgeleme probleminin analizinde kullanılan modelin doğruluğu ve gereken girdi verilerinin güvenilirliğine bağlıdır (Pinedo, 2008, p.4).

1.1 Üretimde Çizelgeleme

Çizelgeleme, bir ürünü üretmek veya hizmeti sağlamak için çalışan, ekipman ve tesislere ne zaman ihtiyaç duyulacağını belirler. Planlamanın, üretim başlamadan önceki son aşamasıdır. Operasyonların tipine göre çizelgeleme fonksiyonu önemli ölçüde değişir. Örneğin ilaç veya kimyasal madde üretimi gibi *süreç endüstrileri*nde çizelgeleme, kazana girecek içeriklerin karışımını veya sistemin ne zaman bir karışımın üretimini durdurup, kazanı temizleyip, yeni bir karışımın üretimine başlayacağını belirler. *Kitle üretimi*nde, üretimin çizelgesi hattın planı ortaya konduğunda az çok belirlenmiştir. Ürünler hat boyunca bir istasyondan diğerine her seferinde aynı belirlenmiş sıra ile akış gösterirler. Günlük çizelgeleme kararları, parçaların hattı ne kadar hızlı besleyeceği ve hattın günlük ne kadar çalışacağından oluşur. Buna ek olarak karışık ürünlü bir montaj hattında ürünlerin sırası da belirlenmelidir. *Proje*ler için çizelgeleme kararları çok sayıda ve birbirleriyle ilişkilidir. Bunlar için PERT ve CPM gibi özelleşmiş proje çizelgeleme teknikleri vardır. *Parti* veya *atölye tipi* üretimde, çizelgeleme kararları genellikle daha karmaşıktır. Çizelgeleme, bir parçanın hangi makinede işlenmek için rotalanacağını ve parçaların işlenme sırasını belirler. Atölyede çizelgelemeyi zorlaştıran ana unsur her biri farklı rota ve işlem gereksinimlerine sahip olabilen işlerin çeşitliliğidir (Russell and Taylor, 2003, p.583).

Çizelgeleme alanındaki pek çok gelişme imalat alanındaki problemlere yönelik olarak ortaya çıkmıştır. Bu nedenle çizelgeleme probleminden bahsedilirken genellikle imalat ile ilgili sözcükler kullanılmaktadır. Kaynaklara genellikle *makineler*, görevlere de *işler* denmektedir. Bazen işler *operasyonlar* denen pek çok basit görevden oluşuyor olabilir.

---

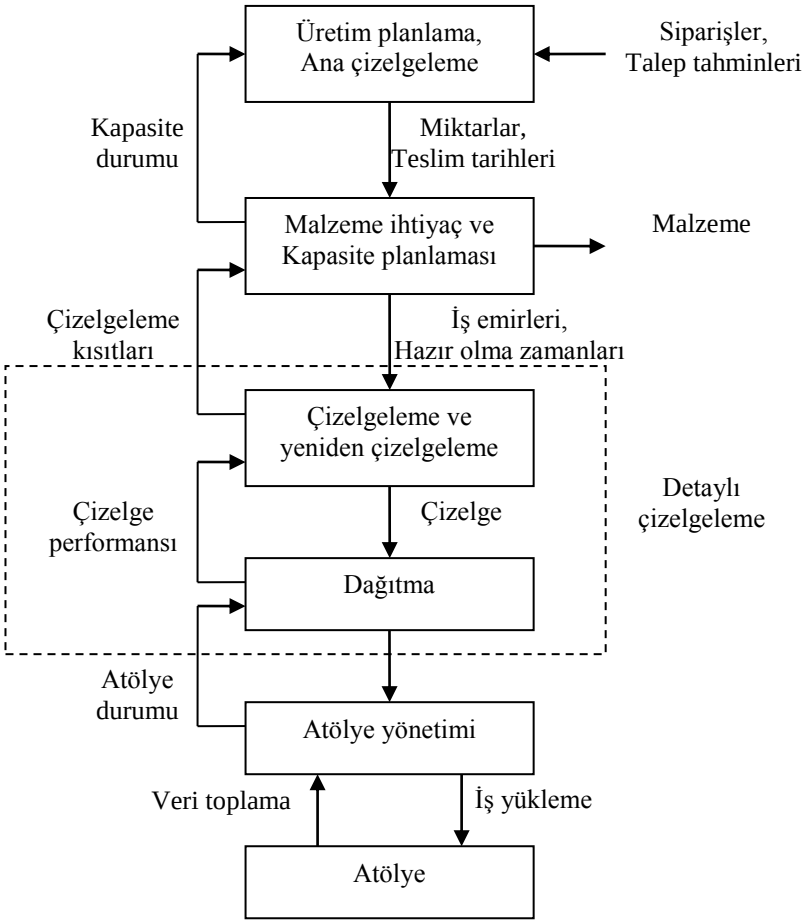
[1] Kombinatoryal optimizasyon problemleri, uygun çözümler kümesinin kesikli bir kümeye indirgendiği veya indirgenebileceği, amacın mümkün olan en iyi çözümü bulmak olduğu optimizasyon problemleridir (Zobolas; Tarantilis; Ioannou, 2008, p.10).

Çizelgeleme problemine konu olan ortam *atölye* olarak adlandırılır (Baker and Trietsch, 2009, p.3).

Çizelgeleme kararları toplam tesis ve ekipman kaynaklarını içeren kapasite planlamasıyla başlar. Kapasite planları çoğunlukla yıllık veya dönemliktir. Yeni ekipman ve tesisler alındıkça veya mevcutlardan bazıları boşa çıkarıldıkça kapasite planı da değişir. Uzun dönemli planlama kararları tesislerin kullanımına, stoklara, insanlara ve dış yüklenicilere göre verilir. Uzun dönemli planlar genellikle aylıktır ve kaynaklar toplam birim, ton veya atölye saat gibi genel bir ölçüye göre tahsis edilir. Fakat ana çizelge uzun vadeli planı parçalara böler ve spesifik bir ürün veya ürün hattı için haftalık bir çizelge geliştirir. Daha sonra kısa dönemli çizelgeler kapasite kararları, uzun vadeli planlama ve ana çizelgeleri; iş sıralamalarına ve çalışan, malzeme ve makinelerin spesifik olarak atanmış haline dönüştürür (Heizer and Render, 2006, p.590-591).

İşletmeye gelen siparişler, teslim tarihleri belirlenmiş işlere dönüştürülmelidir. Bu işler çoğunlukla bir iş merkezindeki makinelerde belirli bir sıra ile işlem görürler. Belli makineler yoğun olduğunda işlerin işlem görmesi ertelenebilir veya yoğun makinelere yüksek öncelikli işler geldiğinde işlem bölme meydana gelebilir. Çizelgeler üzerinde önemli etkilere sahip olduğundan makine arızaları veya beklenenden uzun işlem süreleri gibi atölyedeki öngörülemeyen olaylar da dikkate alınmalıdır. Böyle bir ortamda detaylı bir görev çizelgesi geliştirilmesi, operasyonların kontrolü ve etkinliğin korunmasına yardımcı olur (Pinedo, 2008, p.5).

Organizasyonun çizelgeleme sürecini etkileyen tek parçası atölye değildir. Çizelgeleme, tüm organizasyon için orta ve uzun vadeli planlamayı yürüten üretim planlama sürecinden de etkilenir. Bu süreç, stok seviyeleri, talep tahminleri ve kaynak ihtiyaçlarına dayanan firmanın toplam ürün karmasını ve uzun vadeli kaynak tahsisini optimize etmeye çalışır. Bu yüksek planlama seviyesindeki kararlar çizelgeleme sürecini doğrudan etkileyebilir. Şekil 1'de bir imalat sisteminde bilgi akışını gösteren bir şema görülmektedir (Pinedo, 2008, p.5).



Şekil 1: Bir İmalat Sisteminde Bilgi Akış Şeması
Kaynak: Pinedo, 2008, p.5

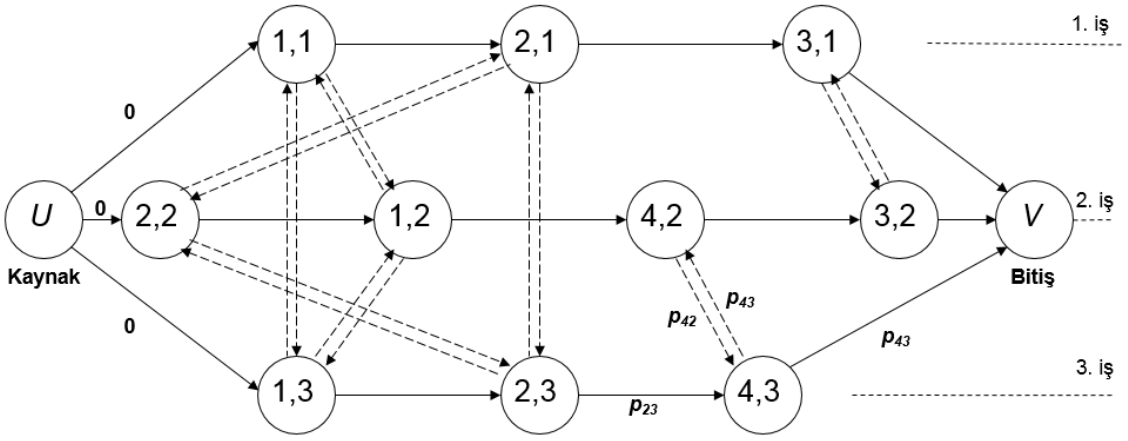
Bir imalat ortamında, çizelgeleme fonksiyonu başka karar verici fonksiyonlarla etkileşimde olmalıdır. Yaygın olarak kullanılan popüler bir sistem Malzeme İhtiyaç Planlaması (MİP) sistemidir. Bir çizelge ortaya çıkarıldıktan sonra tüm hammaddelerin ve kaynakların belirlenen zamanlarda hazır olmaları gereklidir. Tüm işlerin hazır olma zamanları üretim planlama/çizelgeleme sistemiyle birlikte MİP sistemi tarafından belirlenir (Pinedo, 2008, p.6).

1.2 Çizelgelerin Grafiksel Gösterimi

Çizelgeler genellikle Ayrık Graf ve Gannt Şeması olarak adlandırılan iki farklı şekilde grafiksel olarak temsil edilirler.

1.2.1 Ayrık Graf

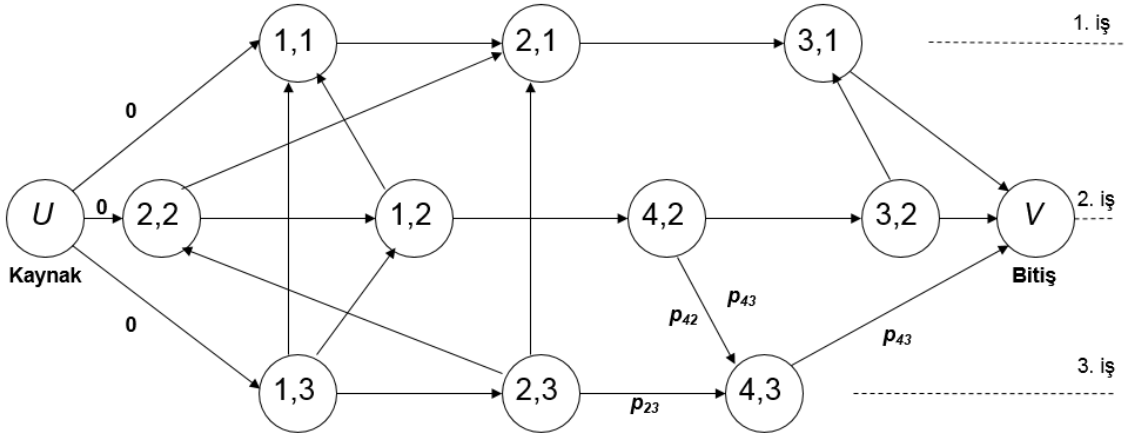
Ayrık graf, $G = (N, A, B)$ şeklinde gösterilir. G yönlü grafik, N düğüm kümesi, A ve B ise yaylar kümesini ifade eder. N düğümleri n iş üzerinde gerçekleştirilmesi gereken tüm (i, j) operasyonlarına karşılık gelmektedir. A birleşik yayları işlerin rotalarını temsil etmektedir. Eğer $(i, j) \rightarrow (k, j)$ yayı A'nın bir parçasıysa, j işi k makinesinden önce i makinesinde işlenmelidir, yani (i, j) operasyonu (k, j) operasyonundan önce gelmektedir. Aynı makinede işlem görmesi gereken iki ayrı işe ait iki operasyon birbirlerine farklı yönlere giden ayrık yay ile bağlıdır. B ayırıcı yayları m adet çift yay kliği oluşturur, her *klik* bir makineye denk gelir. (Klik graf teorisinde herhangi iki düğümün birbirine bağlı olduğu duruma karşılık gelen bir terimdir.) Aynı klikteki tüm operasyonlar (düğümler) aynı makinede işlem görmelidir. Bir düğümden çıkan birleşik veya ayrık tüm yaylar, düğüm tarafından ifade edilen operasyonun işlem süresi kadar (p_{ij}) ağırlığa sahiptir. Ek olarak kukla düğümler olan bir U (kaynak) ve V (bitiş) düğümü vardır. Kaynak düğümü olan U, n işin ilk operasyonlarına giden n adet birleşik yaya sahiptir, bitiş düğümü olan V ise son operasyonlardan gelen n adet birleşik yaya sahiptir. Kaynaktan çıkan yayların ağırlığı 0 (sıfır)dır. Şekil 2'de üç iş ve dört makineden oluşan atölye tipi çizelgeleme problemi için bir ayrık graf görülmektedir (Pinedo, 2008, p.180-181).



Şekil 2: Ayrık Graf

Uygun bir çizelge, çevrimsiz (acyclic) bir yönlü graf oluşturacak şekilde her ayrık yay çiftinden birinin seçimine karşılık gelir. Bir klikten ayrık yayların seçimi çevrimsiz olmalıdır. Böyle bir seçim, operasyonların bir makinedeki işlem görme sırasını da

belirler (Pinedo, 2008, p.181). Şekil 3, seçim yapılmış uygun bir çizelgeyi göstermektedir.



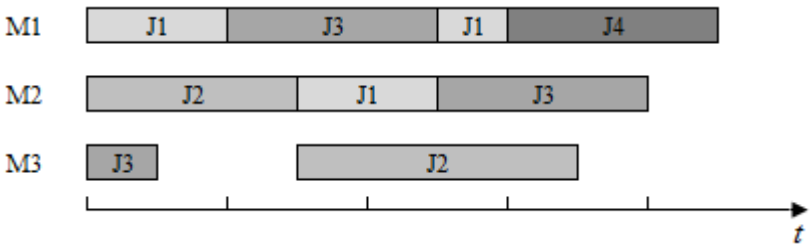
Şekil 3: Uygun Bir Çizelge

Belirli bir çizelge için tüm işlerin maksimum tamamlanma zamanı (C_{max}), kaynaktan bitişe en uzun ağırlıklı yolun uzunluğuna eşittir. Bu yola kritik yol denir.

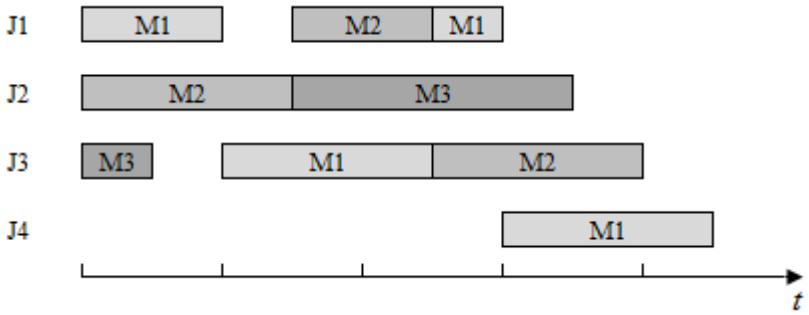
1.2.2 Gannt Şeması

Gannt şeması, çizelgelerin gösterimi için basit ve yaygın olarak kullanılan bir modeldir. Temel haliyle kaynakların zamana yayılmasını gösterir. Kaynaklar dikey eksende, zaman ölçeği de yatay eksende yer alır. Temel Gannt şeması işlem sürelerinin kesin olarak bilindiğini varsayar (Baker and Trietsch, 2009, p.2-3).

Çizelgeler Gannt şeması yardımıyla gösterilebilir. Gannt şemaları genellikle kullanıldığı haliyle Şekil 4'teki gibi makine odaklı olabileceği gibi, Şekil 5'teki gibi iş odaklı da olabilir (Brucker, 2007, p.1-2).



Şekil 4: Makine Odaklı Gannt Şeması



Şekil 5: İş Odaklı Gannt Şeması

Bir Gannt şeması, problemi görselleştirmesinin yanında çizelgeleri karşılaştırma ve performanslarını ölçme amacına da hizmet eder (Baker and Trietsch, 2009, p.3).

1.3 Çizelge Sınıfları

Aynı makinede iki ayrı işe ait zaman aralıkları çakışmıyorsa, aynı işe ait farklı makinelerdeki iki zaman aralığı çakışmıyorsa ve probleme özel karakteristikler (öncelik kısıtları vs.) sağlanıyorsa bir çizelge *uygun* olarak adlandırılır. Uygun olmakla beraber belirli bir kriteri minimize ediyorsa da *optimal*dir (Brucker, 2007, p.3).

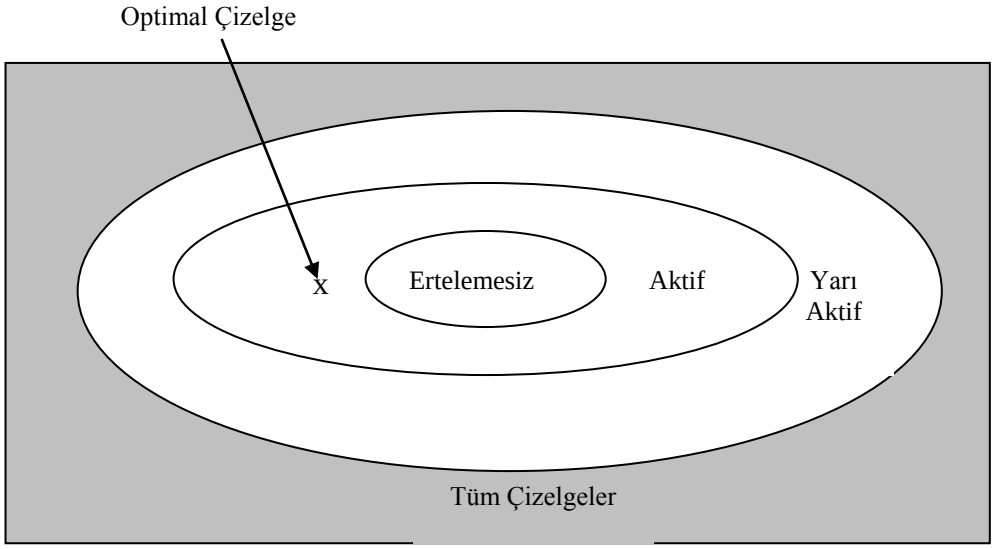
Prensipte, herhangi bir problem için uygun çizelgelerin sayısı sonsuzdur. Çünkü birbiri ardına gelen operasyonlar arasına isteğe bağlı olarak atıl zaman konulabilir. Bu şekilde bir atıl zaman herhangi bir düzenli performans kriterine bir fayda sağlamasa da her makine için operasyon sıralaması belirlenebilir. Düzenli bir performans kriteri, işlerin tamamlanma zamanlarındaki artışa bağlı olarak azalmayan bir fonksiyondur. Gerçekte operasyonların mümkün olduğu kadar derli toplu çizelgelenmesi tercih edilir. Herhangi bir makinede sıralama ile oynamadan bazı operasyonlar daha erkene alınabiliyorsa gereksiz atıl zaman vardır. Bu şekilde bazı operasyonların başlama zamanlarını düzenlemek, çizelgenin geri kalanını aynen muhafaza ederek bir operasyonu ifade eden kutuyu Gannt diyagramında sola hareket ettirmekle eşdeğerdir. Bu tip bir düzenleme *yerel sola kayma* olarak adlandırılır. Her makine için operasyon sıralamasının belli olduğu, yerel sola kaymanın mümkün olmadığı sadece bir çizelge vardır (Baker and Trietsch, 2009, p.328). Çizelgede işlem sıralamasını değiştirmeden veya teknolojik kısıtları ve hazır olma zamanlarını ihlal etmeden bir operasyonun daha önce başlamasını sağlamak mümkün değilse bu çizelgeye *yarı-aktif* çizelge denir. Başka bir deyişle, yarı-aktif bir çizelgede her operasyon işlem görmesi mümkün olan en erken zamanda başlar (French, 1982, p.27). Yarı-aktif çizelgeler kümesi tüm çizelgeler kümesine baskındır ve

herhangi bir düzenli performans kriterini optimize etmek için sadece yarı-aktif çizelgeleri dikkate almak yeterlidir (Baker and Trietsch, 2009, p.328).

Yarı-aktif çizelgelerin sayısı sonlu ise de bu sayı biraz büyüktür. Bununla birlikte yarı-aktif çizelgeler arasında baskın bir alt küme bulmak mümkündür. Sıralamayı değiştirerek herhangi başka bir operasyonu ertelemeden bazı operasyonları erkene alma durumuna *global sola kayma* denir. Global sola kaymanın mümkün olmadığı tüm çizelgelerin kümesi *aktif* çizelgeler kümesi olarak adlandırılır (Baker and Trietsch, 2009, p.331). Aktif bir çizelgede işlem sırası öyledir ki, diğer operasyonları ertelemeden veya teknolojik kısıtları ihlal etmeden herhangi bir operasyon daha erken başlayamaz (French, 1982, p.157). Aktif çizelgelerin, yarı-aktif çizelgelerin alt kümesi olduğu açıktır. Yarı-aktif çizelgeler kümesinin tüm çizelgeler kümesine baskın olduğu gibi, aktif çizelgeler kümesi de yarı-aktif çizelgeler kümesine baskındır. Başka bir deyişle, herhangi bir düzenli performans kriteri optimize edileceğinde sadece aktif çizelgeleri dikkate almak yeterlidir. Aktif çizelgelerin sayısı rotaların ve işlem sürelerinin bir fonksiyonudur, fakat yarı-aktif çizelgelerin sayısı sadece rotaların bir fonksiyonudur. Genellikle belli bir yarı-aktif çizelge bir dizi global sola kayma ile pek çok farklı aktif çizelgeye dönüştürülebilir (Baker and Trietsch, 2009, p.331).

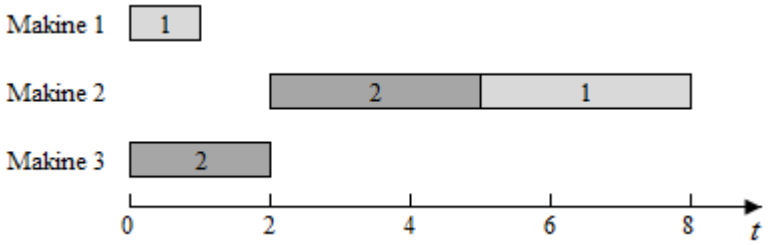
Aktif çizelgelerin sayısı fazla olabilir, bazen aktif çizelgelerin bir alt kümesi olan *ertelemesiz* çizelgelere odaklanmak uygundur. Ertelemesiz bir çizelgede bir operasyonun işlemine başlayabilecek durumda olan hiçbir makine boş bırakılmaz. Tüm ertelemesiz çizelgeler sola kaymaya izin vermediklerinden aktif çizelgelerdir. Diğer yandan, pek çok aktif çizelge ertelemesiz çizelge değildir. Bu yüzden ertelemesiz çizelgelerin sayısı aktif çizelgelerden önemli ölçüde az olabilir. Fakat ertelemesiz alt kümenin optimum sonucu içereceğinin garantisi yoktur (Baker and Trietsch, 2009, p.332-333). Bir çizelgenin ertelemesiz olması mecburi olmayan atıl sürenin (unforced idleness) önlenmesine eşdeğerdir. İşlem bölmeye de müsaade eden ve düzenli bir amaç fonksiyonu olanlar dahil olmak üzere pek çok model için ertelemesiz olan optimal çizelgeler vardır. Fakat mecburi olmayan atıl sürelerin avantajlı olduğu modeller de vardır (Pinedo, 2008, p.22).

Şekil 6'da üç sınıfa ayrılan işlem bölmesiz çizelgeler görülmektedir: Ertelemesiz çizelgeler, aktif çizelgeler ve yarı aktif çizelgeler (Pinedo, 2008, p.25).



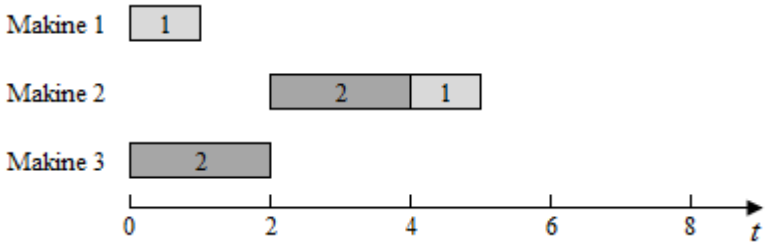
Şekil 6 : İşlem Bölmesiz Çizelgelerin Sınıflandırılması
Kaynak: Pinedo, 2008, p.25.

İki iş ve üç makinenin olduğu bir örnekte, 1. iş 1. makinede bir birim zamana, 2. makinede üç birim zamana ihtiyaç duymaktadır. 2. iş 3. makinede iki birim zamana, 2. makinede 3 birim zamana ihtiyaç duymaktadır. Her iki iş de en son 2. makinede işlem görmelidir. Şekil 7'deki gibi 2. makinede 2. işin 1. işten önce işlendiği çizelge ele alınırsa, bu çizelgenin aktif olduğu ortadadır. 2. makinedeki işlerin sıralarını tersine çevirmek 2. işin işlemini geciktirir. Fakat çizelge ertelemesiz değildir. 2. makine, t = 1'de uygun bir iş olmasına rağmen t = 2'ye kadar boş bekler (Pinedo, 2008, p.24).



Şekil 7 : Ertelemesiz Olmayan Aktif Bir Çizelge

Üç makine ve iki işin olduğu bir atölyede iki işin rotası önceki örnekteki gibidir. 1. işin işlem süreleri 1. ve 2. makinelerin her ikisinde bire eşittir. 2. işin işlem süreleri 2. ve 3. makinelerin her ikisinde ikiye eşittir. Şekil 8'deki gibi 2. işin 2. makinede 1. işten önce işlendiği çizelge ele alınsın. Çizelgeye göre 2. işin 2. makinedeki operasyonu t = 2'de başlamakta ve 1. işin 2. makinedeki operasyonu t = 4'te başlamaktadır. Bu çizelge yarı aktiftir. Fakat aktif değildir, çünkü 1. iş 2. makinede 2. işin operasyonunu ertelemeden işlem görebilir (Pinedo, 2008, p.25).



Şekil 8 : Aktif Olmayan Yarı Aktif Bir Çizelge

1.4 Çizelgelerde Performans Kriterleri

Çizelgelemedeki amaç, işlerin makinelerdeki işlemlerini bazı performans kriterlerini optimize edecek şekilde bir zaman programına oturtmaktır. Çizelgelerdeki performans kriterleri ile ilgili cevaplanması gereken iki soru vardır (French, 1982, p.25):

- Belli bir performans kriteri açısından optimal bir çizelge mevcut mudur?
- Belli bir kritere göre optimal bir çizelge bulunduğunda, başka bir kriter açısından durum ne olacaktır?

İlk soru açısından düzenli performans kriterlerinin olduğu durumlarda optimal çizelgeler mevcuttur. Sonlu sayıda operasyon sıralaması vardır ve bir operasyon sırası, işlerin her makinedeki işlem görme düzenini verir. Yani m makinenin her birinde $\{J_1, J_2, \ldots J_n\}$ işlerinin bir permutasyonunu belirtir. $\{J_1, J_2, \ldots J_n\}$ işlerinin de $n!$ tane permutasyonu olduğu için $(n!)^m$ tane olası operasyon sıralaması vardır. Gerçekte teknolojik kısıtlardan dolayı bu sıralamalardan pek çoğu uygun çözüm olmayabilir. Böylece $(n!)^m$ taneden daha az operasyon sıralaması olmaktadır. Bu sıralamalardan en az bir tanesi de performans kriteri açısından optimum çözümü vermelidir (French, 1982, p.27).

İkinci soru için; performans kriteri maliyetin sadece bir bileşenini oluşturur, uygulamada problemler için performans kriterini toplam maliyetlerin bir temsilcisi olarak kabul etmemek gerekir; maliyetin bir bileşeni minimize edilirken başka bir bileşeninin de minimize edilip edilmediğini bilmek yararlı olur. Pek çok bileşeni eşzamanlı olarak minimize eden çizelgeler, genel olarak sadece tek bir bileşeni minimize eden çizelgelerden daha tatmin edicidir. Bir performans kriterine göre optimal

olan bir çizelge, başka bir kriterine göre de optimal ise, bu iki performans ölçüsü *eşdeğer*dir (equivalent) (French, 1982, p.26-28).

Uygulamada çizelgeleme problemlerinin çoğu çok amaçlıdır. Araştırma camiasında çizelgeleme probleminin çözümü için uzun yıllardır yapılan pek çok çalışmada çoğunlukla benimsenen yöntem, birden çok amacı karar mekanizması tarafından belirlenen ağırlıklarla tek bir değerde toplayıp, daha sonra bu ağırlıklara uygun çözümün bulunmasıdır. Ancak gerçek hayatta çok amaçlı çizelgeleme problemlerinde karar mekanizmasına birden fazla uygun çözümü sunup, bunlar arasından seçim yapabilme kabiliyetine sahip olmak büyük önem taşımaktadır. Bu yöntemin de iki farklı yolu mevcuttur. Bunlardan birincisi, farklı amaçlarla birden çok arama yapılarak, her biri için farklı çözüm türetmektir. İkincisi ise çok amaçlı birden fazla çözümü bir aramada elde edebilmektir. Bu ikinci yöntem, Pareto optimizasyon tekniklerinin, çok amaçlı çizelgeleme problemlerinde uygulanması üzerine odaklanılmasını sağlamıştır (Özbakır, 2004, p.12).

1.5 Çizelgeleme Problemlerinin Sınıflandırılması

1.5.1 Notasyon

Önceden bilinen veriler çizelgeleme sürecinde *girdi* olarak kullanılırlar ve bu tip verileri belirtmek için küçük harfler kullanılır. Çizelgeleme kararları sonucunda ortaya çıkan bilgiler çizelgeleme fonksiyonunun *çıktı*larıdır ve bu tip verileri belirtmek için büyük harfler kullanılır (Baker and Trietsch, 2009, p.11-12).

Tüm çizelgeleme problemlerinde işlerin ve makinelerin sayısının sonlu olduğu kabul edilir. İşlerin sayısı *n* ile makinelerin sayısı da *m* ile gösterilir. Genellikle, *j* indisi bir işe, *i* ise bir makineye tekabül eder. Eğer bir iş belli sayıda işlem basamağı veya operasyon gerektiriyorsa, (*i, j*) çifti *j* işinin *i* makinesindeki operasyonunu belirtir. Aşağıdaki bilgiler *j* işi ile ilgilidir (Pinedo, 2008, 13; French, 1982, p.10-13).

p_{ij} *işlem süresi* (processing time) *j* işinin *i* makinesindeki işlem süresini belirtir. Eğer *j* işinin işlem süresi makineye bağlı değilse veya *j* işi tek bir makinede işlem görüyorsa *i* indisi gösterilmeyebilir.

r_j *hazır olma zamanı* (ready time, release date) İşin sisteme ulaştığı zamandır, *j* işinin işlemine başlanabilecek en erken zamandır.

d_j *teslim tarihi* (due date) *j* işinin daha önce belirlenmiş sevkiyat veya tamamlanma zamanıdır (müşteriye söz verilen tarih). Bir işin teslim tarihinden sonra tamamlanmasına izin verilir, fakat bir ceza uygulanır. Eğer bir teslim tarihinin kesinlikle karşılanması gerekiyorsa buna mecburi teslim tarihi (deadline) denir ve $\overline{d}_j$ ile gösterilir.

w_j *ağırlık* (weight) *j* işinin sistemdeki diğer işlere oranla önemini gösteren bir öncelik faktörüdür.

a_j *müsait zaman* (allowance) *j* işinin hazır olma zamanı ve teslim tarihi arasındaki işlem görme için müsait olan zamandır. $a_j = d_j - r_j$

W_{ij} *bekleme süresi* (waiting time) *j* işinin *i* makinesindeki operasyonundan önce bekleme süresidir.

W_j *toplam bekleme zamanı* (total waiting time) *j* işinin bütün makinelerdeki toplam bekleme zamanını ifade etmektedir. $W_j = \sum_{i=1}^{m} W_{ij}$

C_j *tamamlanma zamanı* (completion time) *j* işinin işlemlerinin bittiği zamanı ifade eder.

$$C_j = r_j + \sum_{i=1}^{m} (W_{ij} + p_{ij})$$

F_j *akış zamanı* (flow time) *j* işinin atölyede geçirdiği süredir. $F_j = C_j - r_j$

L_j *geç kalma* (lateness) Bir işin tamamlanma zamanı ve teslim zamanı arasındaki farktır. $L_j = C_j - d_j$. Bir iş erken ise, yani teslim tarihinden önce tamamlandıysa L_j negatiftir. Geç kalmanın aksine, teslim tarihinden sonra tamamlanması durumunda gecikme gibi negatif değerler almayan değişkenlerin kullanımına daha sık rastlanır.

T_j *gecikme* (tardiness) $T_j = \max\{L_j, 0\}$

E_j *erken tamamlanma* (earliness) $E_j = \max\{-L_j, 0\}$

$C_{\max}$ *maksimum tamamlanma zamanı* (maximum completion time, total production time, makespan) $C_{\max} = \max\{C_j\}$

$F_{\max}$ *maksimum akış zamanı* (maximum flow time) $F_{\max} = \max\{F_j\}$

Tüm işlerin $t = 0$ anında hazır olduğu durumlarda $C_{\max}$ ve $F_{\max}$ aynıdır.

$\overline{C}$ *ortalama tamamlanma zamanı* (mean completion time) $\bar{C} = \frac{\sum C_j}{n}$

$\overline{F}$ *ortalama akış zamanı* (mean flow time) $\bar{F} = \frac{\sum F_j}{n}$

$L_{\max}$ *maksimum geç kalma* (maximum lateness) $L_{\max} = \max\{L_j\}$

$T_{\max}$ *maksimum gecikme* (maximum tardiness) $T_{\max} = \max\{T_j\}$

U_j *geciken iş sayısı* (number of tardy jobs) Teslim zamanından sonra tamamlanan işlerin sayısı

I_i *atıl zaman* (idle time) i makinesinde boş geçen zaman $I_i = C_{\max} - \sum_{j=1}^{n} p_{ij}$

$\sum_{j=1}^{n} p_{ij}$, i makinesindeki toplam işlem süresidir.

$N_w(t)$ t zamanında makineler arasında bekleyen veya işlem için hazır olmayan işlerin sayısı

$N_p(t)$ t zamanında işlem gören işlerin sayısı

$N_c(t)$ t zamanına kadar tamamlanmış işlerin sayısı

$N_u(t)$ t zamanına kadar henüz bitirilmemiş işlerin sayısı

$N_w(t) + N_p(t) + N_c(t) = n$ $\qquad$ $N_u(0) = n$

$N_w(t) + N_p(t) = N_u(t)$ $\qquad$ $N_u(C_{\max}) = 0$

1.5.2 Sınıflandırma

Çizelgeleme problemlerinin sınıflandırılmasında kullanılan dörtlü ve üçlü olarak adlandırılabilecek iki farklı gösterim biçimi vardır.

Dörtlü gösterimde çizelgeleme problemleri *n*/*m*/*A*/*B* tarzında dört parametre için dört alan ile ifade edilir (French, 1982, p.14):

n iş sayısı

m makine sayısı

A atölye ortamındaki akış düzeni veya disiplinini ifade eder. *m* = 1 olduğunda *A* boş bırakılır.

- *F* tüm işler için makine sıralaması aynı olan akış tipindeki durumlar için kullanılır.
- *P* tüm işler için makine sıralaması aynı olmakla beraber, her makine için iş sırasının da aynı olduğu permutasyon akış tipi durumlar için kullanılır. Bu durumda çizelge tamamen, her bir makinede işlem görecek işlerin sırasını veren 1'den *n*'e kadar sayıların permutasyonu olur.
- *G* genel atölye tipi durumlar için teknolojik kısıtlar şeklinde sınırlamaların olmadığı durumlarda kullanılır.

B çizelgenin değerlendirileceği performans ölçüsü

Örnek olarak $n/2/F/C_{max}$, *n* iş, 2 makine ve amacın maksimum tamamlanma zamanını minimize etmek olduğu akış tipi bir çizelgeleme problemini ifade etmektedir.

Dörtlü gösterim temel problemler için uygun olsa da, daha karmaşık (işlem bölme, bağımlı işler vb.) problemlere geçildiğinde üçlü gösterim daha uygun olmaktadır (Jain and Meeran ,1999, p.396).

Bir çizelgeleme problemi üçlü $\alpha \mid \beta \mid \gamma$ gösterimi ile de sınıflandırılabilir. α alanı makine ortamını gösterir ve tek bir veri barındırır. β alanı işlem karakteristikleri ve kısıtlara ilişkin detaylar barındırır. Bu alanda hiçbir veri olmayabileceği gibi, tek veya çoklu veriler olabilir. γ alanı minimize edilecek amacı gösterir ve çoğunlukla tek veri içerir (Pinedo, 2008, p.14).

Makine ortamı ile ilgili tanımlar (Pinedo, 2008, p.14-15):

Tek makine (1) makine ortamları içinde en basitidir ve tüm diğer karmaşık makine ortamlarının özel bir durumudur.

Özdeş paralel makineler (*Pm*) Birbirine paralel *m* adet özdeş makine vardır. *j* işi tek bir operasyon gerektirir ve *m* adet makinenin herhangi birinde işlenebilir.

Farklı hızdaki paralel makineler (*Qm*) Birbirine paralel *m* adet farklı hıza sahip makine vardır. *i* makinesinin hızı v_i ile gösterilir.

İlişkisiz paralel makineler (*Rm*) Birbirine paralel *m* adet makine vardır. *Qm*'den farkı makine hızının işe göre değişkenlik göstermesidir. *j* işinin *i* makinesindeki hızı v_{ij} ile gösterilir.

Akış tipi (*Fm*) Birbirine seri *m* adet makine vardır. Her iş *m* adet makinenin her birinde işlem görmelidir. Her iş aynı rotayı takip etmelidir. Eğer ilk giren ilk çıkar kuralı geçerli ise permutasyon akış tipine dönüşür ve bu gösterimin β kısmında belirtilir.

Esnek akış tipi (*FFc*) Esnek akış tipi, akış tipi ve paralel makine ortamının genelleştirilmiş halidir. Birbirine seri *m* adet makine yerine seri halde *c* tane aşamanın olduğu ve her aşamada belli sayıda paralel makinenin olduğu durumdur. Esnek akış tipi literatürde melez akış tipi ve çok-işlemcili akış tipi olarak da geçmektedir.

Atölye tipi (*Jm*) *m* makineli bir atölyede her işin önceden belirlenmiş bir rotası vardır. Her işin her makineyi en çok bir defa veya birden fazla ziyaret etmesine göre bir ayrım yapılabilir.

Esnek atölye tipi (*FJc*) Esnek atölye tipi, atölye tipi ve paralel makine ortamının genelleştirilmiş halidir. Seri *m* adet makine yerine, her birinde belli sayıda paralel makinenin olduğu *c* adet iş merkezi vardır. Her işin atölye boyunca takip edeceği kendi rotası vardır.

Serbest dolaşımlı atölye tipi (Open shop) (*Om*) *m* adet makine vardır. Her iş *m* makinenin her birinde işlem görmelidir. Fakat bazı işlem süreleri sıfır olabilir. Makinelerde işlerin rotalarıyla ilgili bir kısıtlama yoktur. Çizelge hazırlanırken her iş için bir rota belirlenir.

İş karakteristikleri ile ilgili tanımlar (Pinedo, 2008, p.15-16):

Hazır olma zamanı (r_j) Bu sembol β alanında olduğunda, j işi işlemine hazır olma zamanından (r_j) önce başlayamaz. Eğer β alanında r_j yoksa j işi işlemine herhangi bir zamanda başlayabilir. Hazır olma zamanının aksine, teslim tarihine bu alanda yer verilmez, amaç fonksiyonunun tipi, teslim tarihi olup olmadığı hakkında yeterince bilgi verir.

İşlem bölme (*prmp*) İşlem bölme, bir işi bir makinede işlem görmeye başladıktan sonra bitmesine kadar beklemenin gerekli olmadığını belirtir. Çizelgeyi düzenleyen kişi bir işin işlemini herhangi bir zamanda yarıda kesip, makineye farklı bir iş yükleyebilir.

Öncelik kısıtlamaları (*prec*) Öncelik kısıtlamaları, tek makine veya paralel makine ortamında ortaya çıkabilir ve başka bir iş işlem görmeye başlamadan önce bir veya daha fazla işin bitirilmesi gerekliliğini ifade eder. Öncelik kısıtlamalarının pek çok özel şekli vardır (Bkz: Brucker, 2007, p.3-4). Eğer her işin en çok bir önceli ve en çok bir ardılı varsa bu kısıtlamalara *zincir* (chain) denir. Eğer her işin en çok bir ardılı varsa *içağaç* (intree), en çok bir öncülü varsa *dışağaç* (outtree) denir.

Sıra bağımlı hazırlık zamanları (s_{jk}) s_{jk}, j ile k işlerinin işlemleri arasındaki hazırlık zamanını ifade eder. s_{0k}, k işi sıralamada ilk sırada ise k işi için hazırlık zamanını gösterir. s_{j0}, j işi sıralamada son sırada ise j işi sonrasındaki hazırlık zamanını ifade eder. s_{0k} ve s_{j0} sıfır olabilir. Eğer j ve k işleri arasındaki hazırlık zamanı makineye bağlıysa i indisi de kullanılarak s_{ijk} halini alır. Eğer β alanında s_{jk} yoksa tüm hazırlık zamanları sıfır veya sıralamadan bağımsız kabul edilir. Hazırlık zamanları sıralamadan bağımsız olduğunda işlem sürelerine dahil edilirler.

İş aileleri (*fmls*) n iş bu durumda F farklı iş ailesine dahildir. Aynı aileden işlerin farklı işlem süreleri olabilir, aynı makinede birbiri ardına hazırlık zamanı gerektirmeden işlem görebilirler. Fakat makine bir aileden diğerine geçtiğinde hazırlık gerektirir.

Yığın işleme (*batch*(*b*)) Bir makine belli sayıda (b) işi eşzamanlı olarak işliyor olabilir. Yığın içindeki işlerin işlem süreleri aynı olmayabilir, bu durumda tüm yığının tamamlanması yığın içindeki en uzun işlem süresine sahip işin bitimiyle olur.

Arızalar (*brkdwn*) Makine arızaları, bir makinenin sürekli olarak müsait olmayabileceğini belirtir.

Makine uygunluk kısıtlamaları (M_j) Birbirine paralel *m* makine olma durumunda (*Pm*) M_j ifadesi, m makineden M_j kümesine ait olanların *j* işini yapabileceğini belirtir.

Permutasyon (*prmu*) Akış tipinde ortaya çıkabilecek bir kısıt makinelerin önündeki kuyrukların ilk giren ilk çıkar disiplinine göre işlenebileceğidir. Bu, işlerin ilk makineye girdiği sıranın sistem boyunca korunacağını ifade eder.

Bloke etme (*block*) Akış tipinde ortaya çıkabilecek bir olgudur. Eğer iki makine arasında sınırlı bir ara stok alanı varsa, stok alanı tamamen dolduğunda önceki makinenin işlemini bitirdiği işi serbest bırakmasına izin verilmez. Burada tamamlanmış işin önceki makinede kalması ile bu makinenin sıradaki işe başlaması engellenmiş (bloke edilmiş) olmaktadır.

Beklemesiz (*nwt*) Bu gereksinim de akış tipinde ortaya çıkan bir olgudur. İşlerin ardışık iki makine arasında beklemesine izin verilmez. İlk makinedeki işin başlama zamanı, işin herhangi bir makineyi beklemeden tesiste gezebilmesini sağlayacak şekilde ertelenmelidir.

Yeniden dolaşma (*rcrc*) Bir iş, bir makine veya iş merkezini birden çok kez ziyaret ettiğinde atölye veya esnek atölye tipi problemlerde ortaya çıkar.

Optimalite kriteri ile ilgili tanımlar (Pinedo, 2008, p.18-19):

Minimize edilmek istenen amaç her zaman işlerin tamamlanma zamanının bir fonksiyonudur, bu da tabi ki çizelgeye bağlıdır. *j* işinin *i* makinesindeki tamamlanma zamanı C_{ij} ile gösterilir. *j* işinin sistemden çıkışı (işlem görmesi gereken son makinedeki tamamlanma zamanı) C_j ile gösterilir. Amaç teslim tarihinin bir fonksiyonu olabilir. *j* işinin geç kalması;

$$L_j = C_j - d_j$$

şeklinde ifade edilir. Geç kalma *j* işi geç tamamlandığında pozitif, erken tamamlandığında negatif çıkar. *j* işinin gecikmesi ise;

$$T_j = \max(C_j - d_j, 0) = \max(L_j, 0)$$

şeklinde gösterilir. Gecikme ve geç kalma arasındaki fark, gecikmenin asla negatif olmamasıdır. *j* işinin birim cezası;

$$U_j = \begin{cases} 1 \text{ eğer } C_j > d_j \\ 0 \text{ diğer} \end{cases}$$

şeklinde gösterilir.

Maksimum tamamlanma zamanı ($C_{\max}$) Maksimum tamamlanma zamanı $\max(C_1,\ldots,C_n)$, sistemi terk eden son işin tamamlanma zamanına eşittir. Minimum bir maksimum tamamlanma zamanı genellikle iyi bir makine kullanımı anlamına gelir.

Maksimum geç kalma ($L_{\max}$) Maksimum geç kalma, $\max(L_1,\ldots,L_n)$ şeklinde tanımlanır. Teslim tarihini en çok ihlal eden iş ile ilgilenir.

Toplam ağırlıklı tamamlanma zamanı ($\sum w_j C_j$) *n* adet işin ağırlıklı tamamlanma zamanlarının toplamı, çizelge tarafından yüklenilen toplam elde bulundurma veya stok maliyetleri hakkında bilgi verir.

Toplam ağırlıklı gecikme ($\sum w_j T_j$) Her işin önemine göre gecikmelerinin toplamıdır.

Geciken işlerin ağırlıklı sayısı ($\sum w_j U_j$) Geciken işlerin önemlerinin yani ağırlıklarının toplamıdır.

Yukarıdaki amaç fonksiyonları *düzenli* performans kriterleridir. Düzenli bir performans kriteri, işlerin tamamlanma zamanlarındaki artışa bağlı olarak azalmayan bir fonksiyondur. Araştırmacılar düzenli olmayan amaç fonksiyonlarını da çalışmaya başlamışlardır. Örneğin teslim tarihi d_j olan *j* işi, erken tamamlanma cezasının konusu olabilir. *j* işinin erken tamamlanması;

$$E_j = \max(d_j - C_j, 0)$$

olarak ifade edilir. Erken tamamlanma cezası C_j ile artmamaktadır. Toplam erken tamamlanma ve toplam gecikmeyi içeren bir amaç düzenli değildir.

$$\sum_{j=1}^{n} E_j + \sum_{j=1}^{n} T_j \ ,$$

Düzenli olmayan daha genel bir amaç, toplam ağırlıklı erken tamamlanma ile toplam ağırlıklı gecikmenin toplamıdır.

$$\sum_{j=1}^{n} w_j^{'} E_j + \sum_{j=1}^{n} w_j^{''} T_j \ .$$

Erken tamamlanma ve gecikmenin ağırlıkları birbirinden farklı olabilir.

1.6 Çözüm Yaklaşımları ve Karmaşıklık

Çizelgeleme problemlerine getirilen çözüm yaklaşımları, optimum çözüm bulmaya çalışan ve yaklaşık çözüm bulmaya çalışan yöntemler olarak ikiye ayrılabilir. Çizelgeleme problemleri zor problemler olarak kabul edildiğinden özellikle büyük boyutlu problemlerde optimum çözümden feragat edip daha kısa sürede optimum olmayan çözümler bulan yöntemler kullanılmaktadır.

1.6.1 Optimizasyon

Optimum sonucu garantileyen yöntemlerdir, fakat çizelgeleme problemlerinde problem boyutu büyüdükçe yetersiz kalmaktadırlar.

1.6.1.1 Etkin Yöntemler

Optimum sonucu polinom zamanda[2] bulabilen çözüm adımlarıdır. İki makinelik akış tipi çizelgeleme için geliştirilmiş optimum çözüm bulan yöntem (Johnson, 1954), iki iş ve *m* makine veya iki makine ve *n* işlik bir çizelgeleme problemi için optimum çözüm bulan yöntem (Aker, 1956), iki iş ve *m* makine probleminin genişletilip üç iş ve on makineli bir problemine optimum çözümü bulan yöntem (Szwarc, 1960), iki makineli ve birim işlem zamanlı atölye tipi çizelgeleme problemi için optimum çözüm bulan etkin yöntem (Hefetz and Adiri, 1982), iki makineli düzenli bir amaç fonksiyonuna

[2] Bir problemi çözmek için gereken zaman girdinin boyutu ile üstel olarak artmıyorsa bu problemler polinom zamanlıdır. Polinom zamanlı olmayan bazı problemlerin bazı özel durumlarını polinom zamanda çözebilen algoritmalar vardır, bu algoritmalara da polinom zamanlı algoritmalar denir.

sahip çizelgeleme problemlerine polinom zamanlı çözüm bulan bir yöntem (Brucker; Kravchenko; Sotskov, 1997) bu başlık altında incelenebilir.

1.6.1.2 Sayım Yöntemleri

1960'lı yıllarda kesin çözümler bulabilmek için daha özenli ve karmaşık matematiksel yapıları kullanan tam sayım (enumeration) algoritmaları kullanılmıştır. Bu stratejilerin çok büyük teorik değerleri olsa da, bu çalışmaların çoğunun sonucu hayal kırıklığı olmuştur (Jain and Meran, 1999, p.396-397).

Dal ve Sınır Algoritması temel bir sayım stratejisidir. Tüm uygun çizelgelerden oluşan çözüm uzayında dinamik olarak yapılanan bir ağaç araştırılır. Bu teknik ağacın büyük kısımlarının araştırmadan çıkarılması için prosedürler ve kurallar formüle eder (Jain and Meeran, 1999, p.398). Dal, geniş çaplı bir problemi iki veya daha fazla alt probleme bölme işlemi için, sınır ise bir alt problemin optimal sonucu için alt sınır hesaplanması işlemi için kullanılır (Baker and Trietsch, 2009, p.47). Bu yöntemle de ancak küçük boyutlu problemler makul bir zaman içinde çözülebilmektedir. Bunun temel sebebi de erken çözüm aşamalarında problemin büyük kısmını çözüm dışında bırakacak şekilde güçlü eleme kuralları ortaya konamamasıdır (Lageweg; Lenstra; Rinnooy Kan, 1977, p.448). Bu yöntemin kullanıldığı bir çalışmada genel *n* iş *m* makine çizelgeleme problemi için karma tamsayılı bir formülasyon geliştirilmiş ve iki makine üç iş olan bir problemin optimal çözümü bulunmuştur (Greenberg, 1968). Carlier and Pinson (1989) ünlü bir 10x10 atölye çizelgeleme problemini ilk defa dal ve sınır yöntemiyle çözmüşlerdir. Brucker; Jurisch; Sievers (1994), hızlı bir dal sınır algoritması geliştirilmiş ve 20 yıldır optimum şekilde çözülmemiş olan 10x10 boyutlarındaki bir karşılaştırma problemine çözüm bulmuşlardır.

Matematiksel yöntemler arasında karma tamsayılı doğrusal programlama ve lagrange gevşetmesi yöntemleri vardır. Bowman (1959) üç iş dört makineli bir çizelgeleme problemi için deneysel bir tamsayılı doğrusal programlama modeli geliştirmiştir. 0-1 tamsayılı doğrusal programlama modeli tek makine çizelgelemede (Dyer and Wolsey, 1990) ve yeniden işlemeli atölye tipi çizelgeleme (Pan and Chen, 2005) problemlerinde kullanıldığı gibi akış tipi çizelgeleme problemlerinde de kullanılmıştır. Lagrange yöntemi de atölye tipi çizelgeleme problemleri için kullanılmıştır. Fisher (1973)

Lagrange çarpanlarını kullanarak atölye tipi çizelgeleme problemi için bir yöntem geliştirmiştir.

Sayım yöntemlerinin kısıtlamalarından dolayı, yaklaşık yöntemler uygun bir alternatif haline gelmiştir. Bu yöntemler optimal çözüm garanti etmezken hızlı çözüm üretebildiklerinden dolayı geniş çaplı problemler için kullanılabilir (Jain and Meeran, 1999, p.398).

Bu noktada, yaklaşık yöntemler başlığı altındaki sezgisel yöntemler üzerinde durmadan önce optimal çözüm sağlayan yöntemlerin neden yetersiz kaldıkları ile ilgili olarak karmaşıklık konusu üzerinde durmakta fayda vardır.

1.6.2 Karmaşıklık

Bilgisayar biliminin *karmaşıklık teorisi* olarak bilinen bir dalındaki gelişmeler, çizelgeleme problemleri ve çözüm teknikleri arasındaki ilişki üzerine faydalı bir bakış açısı sağlamaktadır. Karmaşıklık kavramı, çözüm algoritmasının gerektirdiği hesaplama çabasına tekabül etmektedir. Hesaplama karmaşıklığı büyüklük derecesi gösterimi notasyonu ile tanımlanır. *n* büyüklüğünde bir problem için (*n,* problemi nitelemek için gerekli olan bilgi miktarını belirtmektedir) belli bir algoritmanın kullanıldığı varsayılırsa, algoritmanın gerektirdiği hesaplama sayısı *n*'in bir fonksiyonu ile üstten sınırlandırılır (Baker and Trietsch, 2009, p.6).

Polinom zaman karmaşıklığına sahip olanlar (*P*) ve olmayanlar (*NP*) şeklinde iki tip algoritmadan bahsedilebilir. *P* sınıfı polinom zamanlı algoritmaların, *NP* (non-deterministic polynomial time) sınıfı ise üstel zamanlı algoritmaların bulunduğu problemlerden oluşur (French, 1982, p.145). Eğer *x* girdilerin boyutu ve *y* de bir sabit ise, $O(x^y)$ zaman gereksinimli bir problemin polinom karmaşıklığa sahip olduğu söylenir ve *P* ile gösterilir. Eğer problemin karmaşıklığı $O(y^x)$ ise problem *NP*-zor olarak düşünülür. Bu durumda optimal bir algoritma, girdiye bağlı, üstel olarak artan hesaplama adımları gerektirir (Jain and Meeran, 1999, p.391).

NP-complete olarak adlandırılan bir problem sınıfı, çok sayıda iyi bilinen ve zor kombinatoryal problemleri içerir. Bu problemler, eğer içlerinden birisi polinom bir algoritmayla çözülebilirse, diğerlerinin de çözülebileceği şekilde denktirler. Fakat

matematikçiler ve bilgisayar bilimcileri tarafından yıllardır yapılan çalışmalar bu sınıfta herhangi bir problem için polinom zamanlı bir algoritma ortaya çıkaramamıştır ve böyle bir algoritmanın olmadığı tahmin edilmektedir. En az bunlar kadar, hatta daha da zor olan optimizasyon problemleri *NP*-zor problemler olarak adlandırılır (Baker and Trietsch, 2009, p.6).

n x m boyutunda bir atölye çizelgeleme probleminin üst sınırı $(n!)^m$ şeklinde bulunur, dolayısıyla 20 x 10 boyutunda bir problemin 7,2651 x 10^{183} olası çözümü vardır. Tüm bu ihtimallerin içinden uygun çizelgeleri ve optimal olanı bulmak için sayım yöntemi uygulanabilir değildir (Jain and Meeran, 1999, p.391).

Ignall ve Schrage (1965, p.409), maksimum tamamlanma zamanını minimize etme amaçlı 3 makineli akış tipi bir çizelgeleme problemi için geliştirdikleri algoritmanın n+1 iş için n iş için gereken zamanın iki katını gerektirdiğini bildirmişlerdir. n işin olduğu bir problemi çözmek 1 saniye sürüyorsa n+r işli bir problemi çözmek 2^r saniye sürecektir. r=20 için 2^r saniye 12 güne tekabül etmektedir (French, 1982, p.137). Saniyede 1.000.000 altküme oluşturan ve değerlendirebilen dinamik programlama algoritması için bir bilgisayar uygulaması düşünülürse, 25 işlik bir problemin çözümü yaklaşık yarım dakikalık bir bilgisayar zamanı tutar. Fakat 35 işlik bir problem 9 saat, 45 işlik bir problemin çözümü ise bir yıldan fazla bir zaman sürer (Baker and Trietsch, 2009, p.57).

Problemleri *P* ve *NP* olarak ayırmanın faydası, büyük çapta bir *NP*-zor problemi çözme ihtiyacı ile karşılaşıldığında, eldeki teknikler ile optimal çözümlerin bulunamayabileceğinin bilinmesidir. Bu durumda optimal çözüm garanti etmeyen fakat daha az hesaplama gereksinimi duyan sezgisel çözüm prosedürü kullanmak daha iyi olacaktır. *NP*-zor durumlar, herhangi mantıklı bir sıralamayı kullanarak atölyede işi yapmanın, problemi mümkün en hızlı bilgisayarda optimal olarak çözmekten daha kısa sürdüğünde ortaya çıkar. Bu nedenle, sezgisellere başvurmak uygulamada bir istisna yerine kural gibidir (Baker and Trietsch, 2009, p.6).

Çizelgelemede sıkça karşılaşılan *NP*-zor problemler için mevcut kolay çözüm yoktur. Dahası matematiksel görüşe göre de, asla kolay bir çözüm olmayacaktır. Öncelikle karşılaşılan problemin *NP*-zor olup olmadığı bilinmelidir, bunun için literatür

araştırılarak daha önce *NP*-zor olduğunun gösterilip gösterilmediğine bakılabilir. Eğer büyük bir *NP*-zor problem olduğu belirlenirse, makul bir zamanda optimal çözümünün bulunmasının imkansız olduğu kabul edilmelidir. Fakat pratikte problem çözümsüz de bırakılmamalıdır. Çizelgeleme problemleri matematikçilerin kendi zevkleri için icat ettikleri problemler değildir. Temellerini gerçek hayattan alırlar, işlerin operasyonlarının sıralanması gerektiği için ortaya çıkmışlardır. Makul bir zamanda en iyi çözümü bulmak mümkün değilse, bilgi ve tecrübe kullanılarak, optimal değilse bile ortalamadan daha iyi performans göstermesi beklenen bir çözüm bulunabilir. İşte bunu yapacak algoritmalara *sezgisel algoritmalar* denir (French, 1982, p.155).

Pinedo (2008, p.604-605) çizelgeleme problemlerini polinom zamanda çözülenler ve çözülemeyenler olarak listelemiştir. Tablo 1'de polinom zamanda çözülebilen problemler gösterilmektedir.

Tablo 1: Polinom Zamanlı Çözülebilen Problemler

Tek Makine	**Paralel Makineler**	**Atölyeler**
$1 \mid r_j, p_j=1, prec \mid \sum C_j$	$P2 \mid p_j=1, prec \mid L_{max}$	$O2 \mid\mid C_{max}$
$1 \mid r_j, prmp \mid \sum C_j$	$P2 \mid p_j=1, prec \mid \sum C_j$	
$1 \mid tree \mid \sum w_j C_j$		$Om \mid r_j, prmp \mid L_{max}$
	$Pm \mid p_j=1, tree \mid C_{max}$	
$1 \mid prec \mid L_{max}$	$Pm \mid prmp, tree \mid C_{max}$	$F2 \mid block \mid C_{max}$
$1 \mid r_j, prmp, prec \mid L_{max}$	$Pm \mid p_j=1, outtree \mid \sum C_j$	$F2 \mid nwt \mid C_{max}$
	$Pm \mid p_j=1, intree \mid L_{max}$	
$1 \mid\mid \sum U_j$	$Pm \mid prmp, intree \mid L_{max}$	$Fm \mid p_{ij}=p_j \mid \sum C_j$
$1 \mid r_j, prmp \mid \sum U_j$		$Fm \mid p_{ij}=p_j \mid L_{max}$
$1 \mid r_j, p_j=1 \mid \sum w_j U_j$	$Q2 \mid prmp, prec \mid C_{max}$	$Fm \mid p_{ij}=p_j \mid \sum U_j$
	$Q2 \mid r_j, prmp, prec \mid L_{max}$	
$1 \mid r_j, p_j=1 \mid \sum w_j T_j$		$J2 \mid\mid C_{max}$
	$Qm \mid r_j, p_j=1 \mid C_{max}$	
	$Qm \mid p_j=1, M_j \mid C_{max}$	
	$Qm \mid r_j, p_j=1 \mid \sum C_j$	
	$Qm \mid prmp \mid \sum C_j$	
	$Qm \mid p_j=1 \mid \sum w_j C_j$	
	$Qm \mid p_j=1 \mid L_{max}$	
	$Qm \mid prmp \mid \sum U_j$	
	$Qm \mid p_j=1 \mid \sum w_j U_j$	
	$Qm \mid p_j=1 \mid \sum w_j T_j$	
	$Rm \mid\mid \sum C_j$	
	$Rm \mid r_j, prmp \mid L_{max}$	

Kaynak: Pinedo, 2008, p.604.

Teorik olarak öyle olsa da, pratik bir bakış açısından, tüm *NP*-zor problemler eşit zorlukta değildir. Bazı *NP*-zor problemler (sıradan anlamda *NP*-zor) dinamik

programlama kullanılarak sahte polinom zamanda çözülebilmektedir. Eğer çözülecek problemin boyutları çok büyük değilse bu yaklaşım tatminkar sonuçlar verebilir (Brucker, 2007, p.51). Tablo 2'de sıradan anlamda *NP*-zor problemler görülmektedir. (*) ile gösterilen problemler için sahte polinom zamanlı algoritmalar vardır.

Tablo 2: Sıradan Anlamda *NP*-zor Problemler

Tek Makine	Paralel Makineler	Atölyeler
$1 \mid\mid \sum w_j U_j$ (*)	$P2 \mid\mid C_{\max}$ (*)	$O2 \mid prmp \mid \sum C_j$
$1 \mid r_j, prmp \mid \sum w_j U_j$ (*)	$P2 \mid r_j, prmp \mid \sum C_j$	
	$P2 \mid\mid \sum w_j C_j$ (*)	$O3 \mid\mid C_{\max}$
$1 \mid\mid \sum T_j$ (*)	$P2 \mid r_j, prmp \mid \sum U_j$	$O3 \mid prmp \mid \sum w_j U_j$
	$Pm \mid prmp \mid \sum w_j C_j$	
	$Qm \mid\mid \sum w_j C_j$ (*)	
	$Rm \mid r_j \mid C_{\max}$ (*)	
	$Rm \mid\mid \sum w_j U_j$ (*)	
	$Rm \mid prmp \mid \sum w_j U_j$	

Kaynak: Pinedo, 2008, p.605.

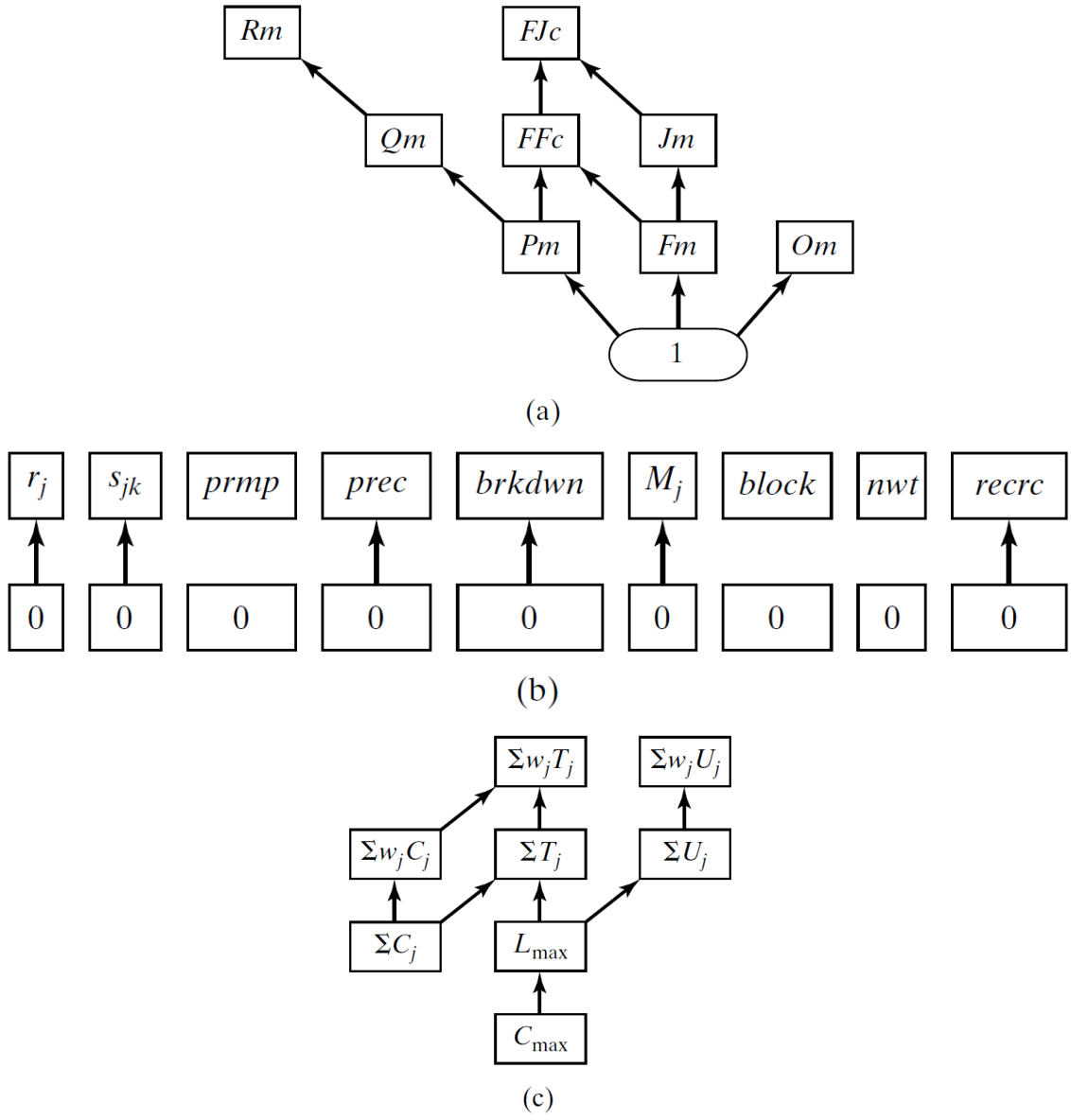
Tablo 3'de güçlü anlamda *NP*-zor problemler görülmektedir. Buradaki problemler güçlü anlamda *NP*-zor olan en basit problemlerdir.

Tablo 3: Güçlü Anlamda *NP*-zor Problemler

Tek Makine	Paralel Makineler	Atölyeler
$1 \mid s_{jk} \mid C_{\max}$	$P2 \mid chains \mid C_{\max}$	$F2 \mid r_j \mid C_{\max}$
	$P2 \mid chains \mid \sum C_j$	$F2 \mid r_j, prmp \mid C_{\max}$
$1 \mid r_j \mid \sum C_j$	$P2 \mid prmp, chains \mid \sum C_j$	$F2 \mid\mid \sum C_j$
$1 \mid prec \mid \sum C_j$	$P2 \mid p_j=1, tree \mid \sum w_j C_j$	$F2 \mid prmp \mid \sum C_j$
$1 \mid r_j, prmp, tree \mid \sum C_j$		$F2 \mid\mid L_{\max}$
$1 \mid r_j, prmp \mid \sum w_j C_j$	$R2 \mid prmp, chains \mid C_{\max}$	$F2 \mid prmp \mid L_{\max}$
$1 \mid r_j, p_j=1, tree \mid \sum w_j C_j$		
$1 \mid p_j=1, prec \mid \sum w_j C_j$		$F3 \mid\mid C_{\max}$
		$F3 \mid prmp \mid C_{\max}$
$1 \mid r_j \mid L_{\max}$		$F3 \mid nwt \mid C_{\max}$
$1 \mid r_j \mid \sum U_j$		$O2 \mid r_j \mid C_{\max}$
$1 \mid p_j=1, chains \mid \sum U_j$		$O2 \mid\mid \sum C_j$
		$O2 \mid prmp \mid \sum w_j C_j$
$1 \mid r_j \mid \sum T_j$		$O2 \mid\mid L_{\max}$
$1 \mid p_j=1, chains \mid \sum T_j$		
$1 \mid\mid \sum w_j T_j$		$O3 \mid prmp \mid \sum C_j$
		$J2 \mid rcrc \mid C_{\max}$
		$J2 \mid p_j=1, rcrc \mid C_{\max}$

Kaynak: Pinedo, 2008, p.605.

Bu tabloları Şekil 9 ile birlikte ele almak gerekir. Bu şekillerde makine ortamı, işlem kısıtlamaları ve amaç fonksiyonlarına göre karmaşıklık hiyerarşisi verilmektedir (Pinedo, 2008, p.27-28). Şekle göre aşağıdan yukarı doğru gidildikçe problemin karmaşıklığı artmaktadır. Çizelgeleme problemlerinin karmaşıklığı ile ilgili ek bilgi için bkz: Lawler; Lenstra; Rinnooy Kan v.d. (1993).



Şekil 9: Çizelgeleme Problemleri için Karmaşıklık Hiyerarşisi
(a) Makine ortamları (b) İşlem kısıtlamaları (c) Amaç fonksiyonları
Kaynak: Pinedo, 2008, p.27.

1.6.3 Yaklaşık Yöntemler

Optimal çözüm prosedürleri yerine sezgisel çözüm prosedürlerinin kullanılmasının pek çok nedeni vardır. İlk olarak model genellikle gerçek hayattaki problemin basit bir temsilcisidir, dolayısıyla modelin optimal sonucu gerçek problem için en iyi sonuç olmayabilir. İkinci olarak, gerçek hayattaki neredeyse tüm çizelgeleme problemleri NP-zor sınıfındadır, bir bilgisayarda hatta bilgisayarlardan oluşan bir iş istasyonunda optimal bir çözüm bulmak çok uzun zaman alabilir. Üçüncü olarak, uygulamada çizelgeleme ortamında önemli miktarda tesadüfilik vardır; bu yüzden birkaç saat içinde makine ortamında veya iş kümelerinde değişiklikler gibi tesadüfi olayların olabileceği bir sistemde, optimal bir çözüm için çok büyük hesaplama zamanları harcamak makul değildir (Pinedo, 2008, p.450).

Genel olarak çizelgeleme problemleri, kombinatoryal prosedürler, simulasyon teknikleri ve sezgisel çözüm yaklaşımlarının geliştirilmesi, uygulanması ve değerlendirilmesi için bir odak noktası haline gelmiştir (Baker and Trietsch, 2009, p.5).

Yaklaşık algoritmalar kendi içinde kabaca sezgisel ve metasezgisel yöntemler olarak ikiye ayrılabilir. Aralarındaki temel fark, sezgisel yöntemlerin genellikle daha basit, tek çözüm üreten kurallara sahip olması, metasezgisel yöntemlerin ise bir başlangıç çözümünden hareket ederek, karmaşık ve zeki yapısıyla her adımda yeni çözümler üreterek başlangıç çözümünü iyileştirmeye çalışmasıdır. Metasezgisel algoritmaların zeki bir yapısı olduğundan yapay zeka arama teknikleri olarak da adlandırılmaktadır.

1.6.3.1 Sezgisel Yöntemler

İlk ve en basit sezgisel yöntemler *öncelik dağıtma kuralları*dır. Bu yapısal teknikler sıralama için uygun tüm operasyonlara bir öncelik verir, sonra da en yüksek önceliğe sahip operasyonu seçer. Uygulanmaları çok kolaydır ve düşük hesaplama yüküne sahiplerdir. Ancak öncelik dağıtma kurallarının çokça probleme bağımlı olmalarından dolayı herhangi bir kural öne çıkamamaktadır (Jain and Meeran, 1999, p.398). Bu kurallar kendi içinde *statik* ve *dinamik* olarak ikiye ayrılabilir. Statik kurallar zamana bağlı değildir, sadece iş ve/veya makine verisinin bir fonksiyonudur. Dinamik kurallar zamana bağlıdır. Bir başka sınıflandırma kuralların *yerel* veya *küresel* olmasıdır. Yerel kurallar işin beklediği sıra veya işin sırada olduğu makineye ait bilgiyi kullanır. Küresel

bir kural, işin rotası üzerindeki bir sonraki makinedeki işlem süresi gibi diğer makineleri de dikkate alan bilgiler kullanır (Pinedo, 2008, p.372). Panwalkar ve Iskander (1977) kendilerinden önceki 20 yılda yapılan çalışmalardaki 100'den fazla öncelik kuralını sıralamışlar ve sınıflandırmışlardır. Bu kurallardan bazılarına aşağıda kısaca değinilmiştir (Özbakır, 2004, p.91; Pinedo, 2005, p.414-415; T'kindt and Billaut, 2006, p.21-22):

SPT (Shortest Processing Time First): En kısa işlem süresine sahip operasyon seçilir.

LPT (Longet Processing Time First): En uzun işlem süresine sahip operasyon seçilir.

WSPT (Weighted Shortest Processing Time First): Ağırlıklandırılmış en kısa işlem süresine sahip operasyon seçilir.

EDD (Earliest Due Date): En erken teslim tarihine sahip işe ait operasyon seçilir.

MS (Minimum Slack First): Teslim tarihine en az süre kalan işin operasyonu seçilir.

SRPT (Shortest Remaining Processing Time First): Geriye kalan işlem süresi en kısa işe ait operasyon seçilir.

SST (Shortest Setup Time First): Bir makine boşaldığında en kısa hazırlık zamanı gerektiren iş seçilir.

MRWT (Most Remaining Work Time): Kendisi ve kendisinden sonraki işlem süreleri toplamı en fazla olan operasyon seçilir.

LRWT (Least Remaining Work Time): Kendisi ve kendisinden sonraki işlem süreleri toplamı en az olan operasyon seçilir.

FCFS (First Come First Served): Makineye ilk gelen iş seçilir.

Bu basit sıralama kurallarından bir kaçını birlikte kullanan *bileşik sıralama kuralları* da vardır. Basit kurallar iş ve/veya makine özelliklerinin bir fonksiyonudur. Bir *özellik* (attribute) bir iş veya bir makine ile ilgili sabit veya zamana bağlı olabilen bir niteliktir. İş özellikleri; ağırlık, işlem süresi ve teslim tarihi, makine özellikleri, işlem için bekleyen iş sayısı gibi niteliklerdir. Her basit kuralın toplam sıralamaya katkı oranını belirleyen bileşik kural içinde bir ölçekleme parametresi (scaling parameter) vardır.

Ölçekleme parametreleri kuralı tasarlayan tarafından sabit yapılabilir veya zamanın ya da çizelgelenecek belli bir iş kümesinin bir fonksiyonu olarak değişken olabilir. Örneğin ATC (Apparent Tardiness Cost) WSPT ve MS kurallarını birleştiren bir bileşik sıralama kuralıdır (Pinedo, 2008, p.373-374).

Çizelgeleme problemine iyi bir çözüm arayışında kullanılabilecek bir yol da herhangi bir uygun çözüm ile başlayıp, üzerinde düzenleme yapıp, düzenlemenin herhangi bir iyileşme sağlayıp sağlamadığını kontrol edip, performans ölçüsünde herhangi bir gelişme sağlanamayana kadar düzenleme ve iyileşmeyi kontrol etme döngüsünü devam ettirmektir. Her aşamada sadece bir düzenleme yapmak yerine, farklı düzenlemelerden bir çeşit ele alıp içlerinden en umut vaat edeni seçmek *komşuluk arama tekniği* olarak geçmektedir (French, 1982, p.168).

Kayan darboğaz prosedürü (shifting bottleneck) bir seferde bir makineyi çizelgeler. Böylece, prosedürün herhangi bir aşamasında çizelgelenmiş makineler kümesi olan X, ve bunun tamamlayıcısı X' vardır. X' kümesinden bir makine seçilip tüm operasyonları çizelgelenerek X kümesine geçmesi sağlanır. Bu adım X' kümesindeki diğer makinelere ait bilgileri yeniden düzenlemeye imkan verir, bu bilgilere dayanarak, çizelgelenecek yeni makine seçilir. Sonrasında tüm makineler çizelgelenene kadar bu süreç adımları tekrarlanır. Bir sonraki çizelgelenecek makinenin seçimi, prosedürün püf noktasıdır (Baker and Trietsch, 2009, p.337).

1.6.3.2 Metasezgisel Yöntemler

Yapay zeka arama teknikleri optimizasyon problemlerini çözmede göreceli olarak yeni yaklaşımlardır. Yapay zeka arama teknikleri optimal çözüm garanti etmemelerine rağmen diğer sezgisellerden farklıdır. Genellikle probleme özgü genel bir bilgiyi kullanarak veya öğrenerek iyi çözüm arayan sezgiseller yapay zeka arama teknikleri kategorisine aittir denilebilir. Optimizasyon problemlerinin çözümünde genel olarak uygulanan çeşitli yapay zeka arama teknikleri vardır: yapay sinir ağları, genetik algoritmalar, tabu arama ve benzetimli tavlama bunlardan bazılarıdır (Wang, 2005, p.82).

Burada bu yöntemlerin detayına girmeden, sadece en çok bilinen ve kullanılan üç tanesi hakkında kısaca bilgi verilecektir.

Komşuluk arama yöntemi her adımda sadece iyileşme sağlayan (amaç minimizasyon ise daha küçük değere sahip bir amaç fonksiyonu) çözümleri kabul eder. İlk aşamalarda amaç fonksiyonunda hızlı bir düşüş olur; yani çözümdeki iyileşme hızlıdır, fakat arama sürecinin sonuna doğru çözümdeki iyileşmeler çok daha yavaşlar. Bu yöntemin eksik yanı yerel optimuma yakalanma eğilimidir. İyi çözümler veren bir yola girmek güzeldir fakat bu yol küresel optimuma çıkmayabilir. Böyle durumlarda eski çözümden daha kötü bir çözümü kabul edip, yerel optimum tuzağından kurtulup, optimum sonuca giden yeni bir yol bulmak istenir. *Tabu arama*nın kötü çözümleri de kabul etme esnekliği bunu sağlar. Daha kötü bir çözüme gittikten sonra tekrar daha iyi olan eski çözümlere dönmenin önüne geçmek için geçmişteki istenen sayıda çözüm tabu listesinde tutulur (Baker and Trietsch, 2009, p.70-71). Tabu aramanın yerel optimali aşmak amacıyla kullandığı temel prensip, değerlendirme fonksiyonu denilen bir fonksiyon tarafından her iterasyonda yüksek uygunluk değerine sahip hareketin bir sonraki çözümü oluşturmak amacıyla seçilmesine dayanmaktadır. Tabu aramanın altında yatan temel fikir, arama işlemini kontrol etmek amacıyla esnek bir hafıza yapısının kullanılmasıdır (Karaboğa, 2004, p.50). Temel yaklaşım, son çözüme götüren adımın dairesel hareketler yaratmasını engellemek için bir sonraki döngüde tekrarının yasaklanması veya cezalandırılmasıdır. Tabu arama daha önce incelenmiş olan bir yol olmadığı sürece her çözümü araştırabilen bir süreçtir. Böylece yeni bir çözüm uzayının incelenmesi suretiyle yerel minimumdan kaçınılarak istenilen çözüme ulaşılabilmektedir (Cura, 2008, p.65).

Tabu arama da komşuluk arama gibi ilk aşamalarda çözümde hızlı bir iyileşme sağlar. Fakat *benzetimli tavlama* prosedürü erken aşamalardan itibaren daha yavaş bir iyileşme, daha yavaş bir düşüş benimser (Baker and Trietsch, 2009, p.72). Bu algoritma metallerin ısıl işlemi ile bir optimizasyon problemine çözüm araştırma olayları arasındaki benzerlikte ilham alınarak ortaya konulmuştur. Tavlama bir katının sıcaklığının belirli bir maksimum dereceye kadar arttırılarak tekrar azaltılması işlemini tanımlar. Optimizasyon problemi ile tavlama arasında kurulan benzerlikte, katının durumları optimizasyon probleminin muhtemel çözümlerine, durumların enerjileri de çözümlere ait amaç fonksiyonu değerlerine, minimum enerji yani durağan durum (donma) da nihai çözüme karşılık gelmektedir (Karaboğa, 2004, p.27-28). Benzetimli tavlamada her aşama bir öncekinden daha düşük sıcaklıkta gerçekleşecektir. Erken arama aşamalarında (yüksek sıcaklıklarda) daha kötü bir çözüme gitme esnekliği vardır,

fakat ilerleyen arama aşamalarında (düşük sıcaklıklarda) bu esneklik iyice düşer. Bu sayede aramanın başlangıcında amaç fonksiyon değeri dalgalanma eğilimindedir, fakat arama sürecinin sonuna doğru bu pek görülmez (Baker and Trietsch, 2009, p.72). Benzetimli tavlama prosedürü bir dizi iterasyondan oluşur. Başlangıç çözümünden itibaren her tekrarlamada komşu çözümlere hareketle yeni bir çözüm bulunur. Yeni bulunan çözüm hem bir önceki çözümle hem de o zamana kadar bulunan en iyi çözümle karşılaştırılır. Eğer yeni çözüm bir önceki çözümden iyiyse kabul edilir, o zamana kadar bulunan en iyi çözümden iyiyse, yeni en iyi çözüm olur. Fakat yeni çözüm bir önceki çözümden daha iyi değilse kabul edilip edilmemesi bir olasılığa bağlanır. Bu olasılık da prosedürün başlangıç aşamalarında yüksektir, fakat her tekrarlamada bu olasılık belirlenen bir kademede düşer (Pinedo, 2008, p.380-381). Benzetimli tavlamanın diğer yöntemlerle karşılaştırıldığında en önemli avantajı yerel minimumdan kurtulabilme yeteneğidir (Cura, 2008, p.43).

Genetik algoritma bir evrimsel algoritma türüdür. Evrimsel bir algoritma doğal evrimin benzetilmesidir. Doğada evrim en uygun olanların hayatta kalmasıyla gerçekleşmektedir. Zayıf bireyler yeni bireyleri ya çok az üretebilecek kadar yaşarlar ya da hiç yeni birey oluşturamadan ölürler. Ancak daha güçlü olanlar daha uzun hayatta kalacaklarından çok daha fazla yeni birey üretecek ve kendine benzer özelliklere sahip evlatları olacaktır (Cura, 2008, p.87-88). Genetik algoritma arama uzayında bulunan çözümlerin bazılarının oluşturduğu bir başlangıç popülasyonunu kullanmaktadır. Başlangıç popülasyonu her jenerasyonda doğal seçilim ve tekrar üreme işlemleri vasıtasıyla art arda gerçekleştirilir. En son jenerasyonun en kaliteli bireyi, problem için nihai çözüm olmaktadır. Bu çözüm her zaman optimum çözüm olmayabilir ama optimuma yakın bir çözümdür (Karaboğa, 2004, p.78).

Bu yöntemlerin birkaçının birlikte kullanılması yani melezlenmesi de mümkündür. Genellikle zaman ve çözüm kalitesi açısından en iyi sonuçlar, çeşitli yöntemleri birleştirmelerinden dolayı, melez algoritmalar tarafından verilmektedir (Jain and Meeran, 1999, p.423).

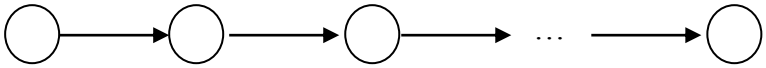
2. BÖLÜM

2 ESNEK AKIŞ TİPİ ÇİZELGELEME PROBLEMİ

2.1 Klasik Akış Tipi Çizelgeleme Problemi

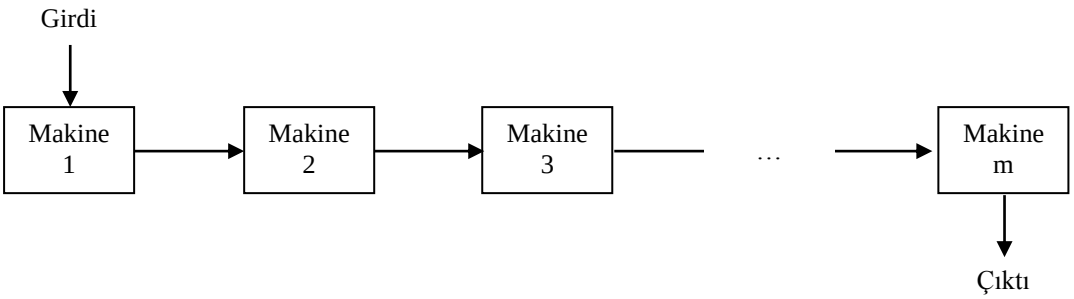
Pek çok imalat ve montaj tesisinde her bir iş bir dizi operasyondan oluşur. Çoğunlukla, bu operasyonlar her işin aynı rotayı izleyeceği şekilde aynı sırayla yapılır. Makinelerin seri şekilde düzenlendiği kabul edilir ve bu imalat ortamına akış tipi atölye (flow shop) denir. Ardışık makineler arasında yarı mamul stoklama kapasitesinin sınırsız olduğu kabul edilir. Bu genellikle işlem gören ürünlerin fiziksel olarak küçük olduğu, büyük miktarlarda ürünlerin makineler arasında stoklanmasının kolay olduğu durumlarda geçerlidir. Ürünler fiziksel olarak büyük olduğunda iki ardışık makine arasındaki stoklama kapasitesi sınırlı olabilir, bu da bloke etmeye neden olur. Bloke etme, ara stok kapasitesi dolu olduğunda ve bir önceki makinenin işlemini bitirdiğinde ara stoğa yeni bir iş bırakmasına izin verilmediğinde ortaya çıkar. Bu durumda sıradaki işin o makinede işleme başlamasını engelleyecek şekilde o iş bir önceki makinede beklemelidir (Pinedo, 2008, p.151).

Akış tipinde bir iş *operasyon* denilen ayrı görevlere bölünmüştür ve her bir operasyon farklı bir makinede işlem görür. Bu bağlamda bir *iş*, özel bir öncelik yapısına sahip operasyonlar topluluğudur. Şekil 10'daki gibi ilkinden sonraki her operasyonun tek öncülü vardır ve sondan önceki her operasyonun tek ardılı vardır (Baker and Trietsch, 2009, p.225).



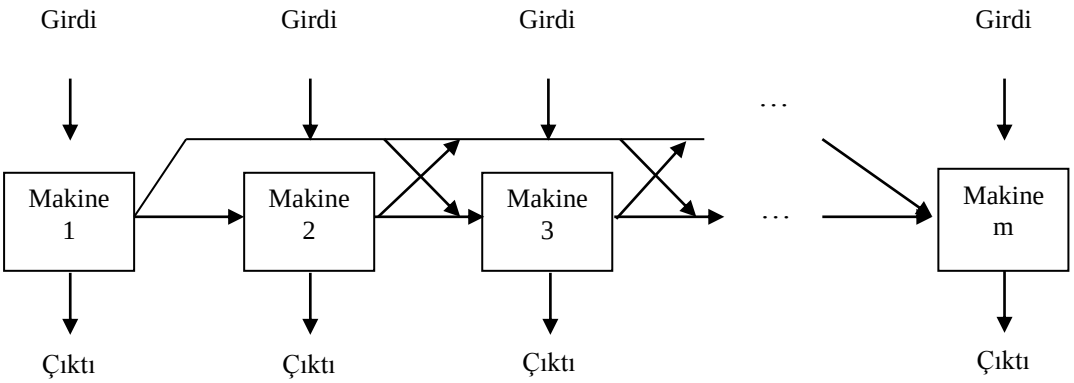
Şekil 10: Akış Tipinde Bir İşin Öncelik Yapısı

Akış tipindeki atölyede *m* makine vardır ve klasik akış tipi modelinde her işin her biri farklı makine gerektiren *m* operasyonu vardır. Böylece akış tipi atölyedeki makineler 1, 2, …, *m* şeklinde; *j* işinin operasyonları ise gereken makineye karşılık gelecek şekilde (1, *j*), (2, *j*), …, (*m*, *j*) şeklinde numaralandırılabilir. Şekil 11 bütün işlerin her makinede bir işlem gerektirdiği klasik akış tipi atölyeyi temsil etmektedir (Baker and Trietsch, 2009, p.225).



Şekil 11: Klasik Akış Tipinde İş Akışı

Daha genel bir durumda işlerin operasyon sayısı *m*'den daha az olabilir (Şekil 12). İşlerin operasyonları her zaman ardışık makinelerde olmayabilir veya ilk ve son operasyonlar her zaman 1 ve *m* makinelerinde olmayabilir. Bununla beraber işin akışı halen tek yönlüdür ve genel akış tipi bazı operasyonların zamanı sıfır olacak şekilde klasik akış tipi problem şeklinde ifade edilebilir (Baker and Trietsch, 2009, p.226).



Şekil 12: Genel Akış Tipi Bir Atölyede İş Akışı

Akış tipi çizelgeleme probleminde (Brucker, 2007, p.174);

- Her *j* işi işlem zamanı p_{ij} (i=1,…,m) olan *m* adet O_{ij} operasyonundan oluşur, O_{ij}, M_i makinesinde işlem görmelidir.

- Her j=1,…,n için $O_{ij} \rightarrow O_{i+1,j}$ (j=1,…,m-1) şeklinde öncelik kısıtlamaları vardır. Örneğin her iş önce makine 1'de, sonra makine 2'de, sonra makine 3'te ve en son olarak makine m'de işlem görecektir.

Problemin amacı her i makinesi için bir π_i iş sırası bulmaktır.

Maksimum tamamlanma zamanını minimize etme amaçlı $F2 \parallel C_{max}$ problemi polinom zamanlı çözülebilen tek akış tipi problemdir. Johnson tarafından bu duruma ilişkin geliştirilmiş bir algoritma vardır. Bu probleme optimal bir çözüm bulmak için her iki makinede de iş sıralarının aynı olduğu çizelgelere odaklanılır. İş sıralamalarının her makinede aynı olacak şekilde sınırlandığı akış tipi çizelgelere *permutasyon akış tipi çizelge* denir. 2 veya 3 makine durumunda akış tipi çizelgeleme probleminin optimal çözümünün permutasyon akış tipi çizelgeleme ile bulunan optimal çözümden daha iyi olmadığı ispatlanmıştır. Fakat makine sayısının 3'ten fazla olduğu problemlerde bu durum ortadan kalkar (Brucker, 2007, p.174-175).

$Fm \parallel C_{max}$ problemi için optimal bir çizelge arayışında işlerin tüm sistemi aynı sırayla geçeceği bir permutasyonun yeterli olup olmayacağı sorusu ortaya çıkar. Fiziksel olarak bir işin meşgul bir makinede sıra bekleyen diğer bir işin önüne geçmesi mümkündür. Makineler *ilk giren ilk çıkar* prensibine göre işlem yapmıyor olabilir ve işlerin makinelerden geçiş sıraları bir makineden diğerine değişebilir. İki makine arasında sırada bekleyen işlerin sıralarını değiştirmenin daha küçük tamamlanma zamanı sağlayacağı durumlar olabilir. Fakat, ilk iki veya son iki makine arasında iş sırasının değişmediği optimal çizelgelerin her zaman olduğu gösterilmiştir. Bu da $F2 \parallel C_{max}$ ve $F3 \parallel C_{max}$ problemleri için makineler arası işlerde sıra değişimi gerektirmeyen optimal çizelgelerin olduğunu göstermektedir. Sıra değişimlerine izin verildiğinde optimal çizelge bulmak, sıra değişimlerine izin verilmediğinde optimal çizelge bulmaktan çok daha zordur (Pinedo, 2008, p.152).

2.1.1 Klasik Akış Tipi Çizelgeleme Problemi için Matematiksel Model

Maksimum tamamlanma zamanını minimize etmeyi amaçlayan klasik akış tipi çizelgeleme problemi için matematiksel model aşağıdaki gibidir.

$$Min\ C_{max} \quad (1)$$

Sınırlar:

$$\sum_{j=1}^{J} Q_{ijm} \leq 1 \quad ; i \neq j, \forall\, i, m \quad (2)$$

$$\sum_{i=1}^{J} Q_{ijm} \leq 1 \quad ; i \neq j, \forall\, j, m \quad (3)$$

$$\sum_{i=1}^{J}\sum_{j=1}^{J} Q_{ijm} = J - 1 \quad ; i \neq j, \forall\, m \quad (4)$$

$$C_{jm} \geq P_{jm} \ ; \ \forall\, j, m = 1 \quad (5)$$

$$C_{jm} - C_{im} + L\left(1 - Q_{ijm}\right) \geq P_{jm} \ ; \ \forall\, j, i \neq j, m \quad (6)$$

$$C_{jm} - C_{j(m-1)} \geq P_{jm} \ ; \ \forall\, j, m = 2 \ldots M \quad (7)$$

$$C_j = C_{jm^*} \ ; \ \forall\, j \quad (8)$$

$$C_j \leq C_{max} \ ; \ \forall\, j \quad (9)$$

İndisler ve Parametreler:

i,j	İş indisleri
J	Toplam iş sayısı
m	Makine indisi
*m**	Sonuncu makine
M	Toplam makine sayısı
P_{jm}	*j* işinin *m* makinesindeki işlem süresi
L	Büyük bir sayı

Karar Değişkenleri:

C_{max}	Maksimum tamamlanma zamanı
C_j	*j* işinin tamamlanma zamanı
C_{jm}	*j* işinin *m* makinesinden ayrılma zamanı
Q_{ijm}	= 1 eğer *j* işi *m* makinesinde *i* işinden sonra geliyorsa
	= 0 diğer

2 no.lu sınır her *i* işini en fazla bir *j* işinin takip etmesini garantiler. 3 no.lu sınır ise her *j* işinin en fazla bir *i* işini takip etmesini garantilemektedir.

4 no.lu sınır, her makine için birbirini takip eden iş sayısının toplam iş sayısından bir eksik olmasını sağlamaktadır. Örneğin 5 iş varsa bunların en fazla 4'ü birbirini takip edebilir.

5 no.lu sınır *j* işinin ilk makineden ayrılma zamanının o işin o makinedeki işlem süresinden daha büyük olmasını garantilemektedir (Özellikle işin ilk makinedeki işlemi için gerekli bir sınır).

6 no.lu sınır *j* işi o makinede *i* işinden sonra işlem görecekse (Q_{ijm} = 1 ise), *j* işinin makineden ayrılma zamanı ile *i* işinin makineden ayrılma zamanları arasındaki farkın en az *j* işinin *m* makinesindeki işlem süresi kadar olmasını sağlamaktadır. Eğer *j* işi *i* işinden sonra işlem görmeyecekse (Q_{ijm} = 0 ise), *L* (büyük bir sayı) iki işin o makineden ayrılma süreleri arasındaki farka eklenerek her durumda işlem süresinden büyük olmasını sağlar.

7 no.lu sınır, *j* işinin *m* makinesindeki operasyonu ile bir önceki operasyonu arasında geçen zamanın en az *j* işinin *m* makinesindeki işlem süresi kadar olmasını garantilemektedir.

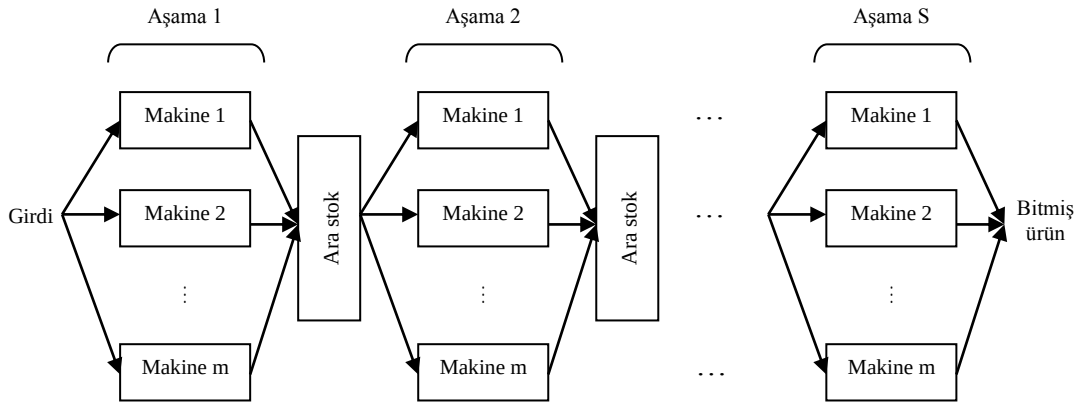
8 no.lu sınır her işin tamamlanma zamanının o işin işlem göreceği son makineden ayrılma zamanına eşit olmasını sağlamaktadır.

9 no.lu sınır tamamlanma zamanının işler içinde tamamlanma zamanı en büyük olan işe eşit olmasını sağlamaktadır.

2.2 Esnek Akış Tipi Çizelgeleme Problemi

Akış tipine göre daha genel bir makine ortamı, birbirine seri belli sayıda aşamadan ve her aşamada birbirine paralel belli sayıda makineden oluşur (Pinedo, 2008, p.151). Esnek akış tipi ortam aynı zamanda melez akış tipi (hybrid flow shop), paralel makineli akış tipi (flow shop with parallel machines), çok işlemcili akış tipi (multiprocessor flow shop) veya esnek akış hattı (flexible flow line) olarak da adlandırılmaktadır (Quadt and Kuhn, 2007b, p.686; Wang, 2005, p.78).

Esnek akış tipi çizelgeleme, her aşamada tek makine bulunan klasik akış tipi modelin ve tek aşamalı paralel makine modelinin genelleştirilmiş halidir (Cheng; Karuno; Kise, 2001, p.140). Paralel makine çizelgeleme problemi işlerin makinelere tahsisi ile ilgilenirken, akış tipi çizelgeleme problemi aşamalarda işlerin sıralanması ile ilgilenir. Esnek akış tipi çizelgelemede de esas karar verilecek şey, bir veya daha fazla kritere göre işlerin her aşamada farklı makinelerde işlem göreceği sırayı belirlemek üzere her aşamada işleri makinelere atamak ve çizelgelemektir (Ribas; Leisten; Framinan, 2010, p.1439). Akış tipi çizelgeleme probleminde bir dizi farklı makine her aşamada bir makine olacak şekilde çok sayıda aşamaya yerleştirilmiştir. Paralel makine çizelgeleme probleminde, bir dizi özdeş makine vardır ve hepsi aynı aşamadadır. Esnek akış tipi çizelgeleme probleminde birbirine seri *s* aşamaya yerleştirilen bir grup makine vardır. *l* aşamasında l=1,…,s, birbirine paralel m_l makine vardır. *j* işi, j=1,…,n, her aşamada makinelerin herhangi birinde işlem görmelidir. *j* işinin değişik aşamalardaki işlem süresi paralel makinelerin özdeş olması durumunda p_{1j}, p_{2j}, …, p_{sj} şeklinde gösterilir. Her makine belli bir anda sadece bir işi işleyebilir (Wang, 2005, 78). Bu durum Şekil 13'de gösterilmiştir.



Şekil 13: Esnek Akış Tipinde İş Akışı
Kaynak: Quadt and Kuhn (2007b, p.687).

Esnek akış tipi makine ortamı pek çok endüstride yaygındır. Eğer bir aşamadaki işlem süreleri diğer aşamalardan önemli ölçüde yüksekse, o aşamaya yeni bir makine eklemek yaygındır. Talepte değişmeler olduğunda zaman içinde yeni makineler satın alınır ve bunların genellikle hızları farklı olur. Makine veya kaynaklar maliyetli olduğunda, eski versiyonları hala ekonomik değere sahip olduklarından, yeni alınan makine kadar hızlı olmasalar da kullanılmaya devam ederler (Crowder, 2006, p.5).

Esnek akış tipi ortam, her birinde birbirine paralel birkaç makinenin çalıştığı bir dizi üretim aşamasından oluşur. Bazı aşamalarda tek makine olabilir, fakat en az bir aşamada birden çok makine olması gerekir. İşlerin atölye içindeki akışı tek yönlüdür. Her iş her aşamada bir makinede işlem görür ve bir veya daha fazla aşamadan geçer. Her aşamadaki paralel makineler özdeş, farklı hızda veya ilişkisiz olabilir (Linn and Zhang, 1999, p.57).

2.2.1 Esnek Akış Tipi Çizelgeleme Problemi için Matematiksel Model

Klasik akış tipi çizelgeleme problemi için olan matematiksel modele her aşamada paralel makine olma durumu eklenerek elde edilen, maksimum tamamlanma zamanını minimize etmeyi amaçlayan esnek akış tipi çizelgeleme problemi için matematiksel model aşağıda verilmiştir.

$$Min\ C_{max} \tag{1}$$

Sınırlar:

$$\sum_{m=1}^{Ms} X_{sjm} = 1 \quad ;\ \forall\ s,j \tag{2}$$

$$X_{sjm} - \sum_{i<>j}^{J} Q_{sijm} \geq 0 \quad ;\ \forall\ s,j,m \tag{3}$$

$$X_{sim} - \sum_{j<>i}^{J} Q_{sijm} \geq 0 \quad ;\ \forall\ s,i,m \tag{4}$$

$$\sum_{j=1}^{J} X_{sjm} - \sum_{i<>j}^{J} \sum_{j<>i}^{J} Q_{sijm} = 1 \quad ;\ \forall\ s,m \tag{5}$$

$$Q_{sijm} + Q_{sjim} \leq 1 \quad ;\ \forall\ s,j,i \neq j,m \tag{6}$$

$$C_{sjm} \geq P_{sjm} * X_{sjm} \quad ;\quad s = 1, \forall\ j,m \tag{7}$$

$$C_{sjm} - C_{sim} + L\left(1 - Q_{sijm}\right) \geq P_{sjm} \ ;\ \forall\ s,j,i \neq j,m \tag{8}$$

$$C_{sjm} - K_{(s-1)j} \geq P_{sjm} - L\left(1 - X_{sjm}\right) \ ;\ \forall s \neq 1,j,i \neq j,m \tag{9}$$

$$C_{sjm} \leq K_{sj} \ ;\ \forall\ s,j,m \tag{10}$$

$$K_{s*j} \leq C_{max} \ ;\ \forall\ j \tag{11}$$

İndisler ve Parametreler:

i,j İş indisleri
J Toplam iş sayısı
s Aşama indisi
s^* Son aşama
m Makine indisi
Ms s aşamasındaki paralel makine sayısı
P_{sjm} j işinin s aşamasında m makinesindeki işlem süresi
L Büyük bir sayı

Karar Değişkenleri:

C_{max} Tamamlanma zamanı
C_{sjm} j işinin s aşamasında m makinesinden ayrılma zamanı
K_{sj} j işinin s aşamasından ayrılma zamanı
Q_{sijm} = 1 eğer s aşamasında m makinesinde j işi i işinden sonra geliyorsa
= 0 diğer
X_{sjm} = 1 eğer j işi s aşamasında m makinesinde işlem görecekse
= 0 diğer

2 no.lu sınır j işinin her aşamada sadece bir makinede işlem görmesini garantilemektedir.

3 no.lu sınır s aşamasındaki m makinesinde her j işinin en çok bir i işini takip edeceğini ifade etmektedir. 4 no.lu sınır s aşamasındaki m makinesinde her i işinin en çok bir j işi tarafından takip edileceğini ifade etmektedir.

5 no.lu sınır her aşamadaki her makine için birbirini takip eden iş sayısının o makinede işlem görecek iş sayısından bir eksik olmasını sağlamaktadır.

6 no.lu sınıra göre her aşamada ve makinede i ve j işleri birbirini takip edebilir veya etmeyebilir. Etmesi durumunda ya i işi j işini takip edebilir ya da j işi i işini takip edebilir, aynı anda bunlardan en fazla biri gerçekleşebilir.

7 no.lu sınır ilk aşama için *j* işinin *m* makinesinden ayrılma zamanının o işin o makinedeki işlem süresine eşit veya ondan daha büyük olmasını garantiler.

8 no.lu sınır *s* aşamasında *j* işi *m* makinesinde *i* işinin peşinde işlem görecekse *j* işinin makineden ayrılma zamanı ile *i* işinin makineden ayrılma zamanları arasındaki farkın en az *j* işinin *m* makinesindeki işlem süresi kadar olmasını sağlamaktadır.

9 no.lu sınır *j* işinin *s* aşamasında *m* makinesinden ayrılma zamanı ile *s*-1 aşamasından ayrılma zamanı arasındaki sürenin en az *j* işinin *s* aşamasında *m* makinesindeki işlem süresi kadar olmasını sağlamaktadır.

10 no.lu sınır *j* işinin *s* aşamasından ayrılma zamanının *j* işinin *s* aşamasında işlem göreceği *m* makinesinden ayrılma zamanına eşit olmasını sağlamaktadır. *j* işinin işlem görmeyeceği makineden ayrılma zamanı 9 no.lu sınırdan dolayı daha küçük bir değer olacağı için *j* işinin *s* aşamasından ayrılma zamanının *j* işinin işlem göreceği makineden ayrılma zamanına eşit olması sağlanmaktadır.

11 no.lu sınır tamamlanma zamanının işler içinde son aşamadan ayrılma zamanı en büyük olan işe eşit olmasını sağlamaktadır.

2.2.2 Esnek Akış Tipi Çizelgeleme Problemlerinde Karmaşıklık

Esnek akış tipi çizelgeleme probleminin karmaşıklığından dolayı farklı çözüm teknikleri geliştirilmiştir. Genellikle, yavaş olmasına rağmen, optimum yaklaşımlar, optimum çözümler üretmektedir. Büyük çaplı esnek akış tipi problemlere uygulandığında optimum yaklaşımın sonuç vermesi saatlerce veya günlerce sürebilmektedir. Diğer yandan, sezgisel yaklaşımlar çok daha hızlıdır; fakat optimum bir çözüm garanti etmez. Gerçekte kombinatoryal bir problem çözüleceğinde, sezgisel bir yaklaşım tercih edilir. Son yıllarda kullanılmaya başlanan yapay zeka arama teknikleri de gitgide popüler olmaktadır (Wang, 2005, p.79).

İki aşamalı ve aşamalarının en az birinde iki paralel özdeş makine bulunan işlem bölmeli (preemptive) akış tipi çizelgeleme probleminin güçlü anlamda NP-zor (NP-hard in the strong sense) bir problem olduğu gösterilmiştir (Hoogeveen; Lenstra; Veltman, 1996). Garey; Johnson; Sethi (1976) *m* makinelik bir akış tipi çizelgeleme probleminde maksimum tamamlanma zamanı amacı için $m \geq 3$ olduğunda, ortalama akış zamanı

amacı için $m \geq 2$ olduğunda; atölye tipi çizelgeleme probleminde de tamamlanma zamanı amacı için $m \geq 2$ olduğunda problemin NP-complete olduğunu göstermişlerdir. Gupta (1988) iki aşamalı özdeş paralel makineli akış tipi çizelgeleme probleminin NP-complete olduğunu göstermiştir. Hoogeven; Lenstra; Veltman (1996) iki aşamalı, aşamalarının birinde en az iki özdeş paralel makine bulunan ve maksimum tamamlanma zamanını minimize etmeyi amaçlayan esnek akış tipi çizelgeleme probleminin güçlü anlamda NP-zor olduğunu göstermiştir.

Aşağıda belirtilen akış tipi problemlerin güçlü anlamda NP-zor problemler olduğu gösterilmiştir (Pinedo, 2008, p.605):

$F2 \mid r_j \mid C_{\max}$
$F2 \mid r_j, prmp \mid C_{\max}$
$F2 \mid \mid \sum C_j$
$F2 \mid prmp \mid \sum C_j$
$F2 \mid \mid L_{\max}$
$F2 \mid prmp \mid L_{\max}$
$F3 \mid \mid C_{\max}$
$F3 \mid prmp \mid C_{\max}$
$F3 \mid nwt \mid C_{\max}$

Esnek akış tipi çizelgeleme problemi, klasik akış tipi çizelgeleme probleminden daha karmaşık olduğundan dolayı 2 ve daha üstü aşamalı esnek akış tipi çizelgeleme problemlerinin tamamının güçlü anlamda *NP*-zor olduğu kabul edilebilir.

2.2.3 Literatür

Akış tipi çizelgeleme ile ilgili olarak literatürde gösterilen ilk çalışma Johnson'un çalışmasıdır (1954). Johnson, 2 aşamalı akış tipi problemler için optimum çözüm veren basit bir kural geliştirmiştir. Bu algoritma 3 aşamalı problemlerin özel bir durumu için de çalışmaktadır fakat 4 ve daha fazla aşamalar için bu yöntemle optimum çözüm bulunamamaktadır.

Esnek akış tipi çizelgeleme problemi ile ilgili literatürde pek çok çalışma olmasına rağmen bunların büyük çoğunluğu teorik çalışmalardır (Lin and Liao, 2003, p.133).

Literatürde ilgili konuda pek çok çalışma olmasına rağmen gerçek hayattaki üretim sistemleri daha karmaşıktır ve teori ile mevcut yöntemlerin uygulanması arasında ciddi bir boşluk vardır (Ruiz and Maroto, 2006, p.781).

Sayıca az da olsa literatürde farklı endüstrilerde uygulaması olan esnek akış tipi çizelgeleme çalışmaları bulunmaktadır. Kablo üretiminde (Narasimhan ve Panwalkar, 1984), baskı devre kartı (PCB) üretiminde (Wittrock, 1985; Wittrock, 1988; Jin; Ohno; Sawik et al. 2002; Alisantoso; Khoo; Jiang, 2003), paketleme tesisinde (Adler; Fraiman; Kobacker et al., 1993), fotoğraf filmi üretiminde (Tsubone; Ohba; Takamuki et al., 1993), traktör imalatında (Ding and Kittichartphayak, 1994), bir gıda işleme tesisinde (Randhawa; Juwono; Burhanuddin, 1994), otomobil montajında (Agnetis; Pacifici; Rossi et al., 1997), uçak motorları üreten bir işletmede (Li, 1997; Huang and Li, 1998), etiket üretiminde (Lin and Liao, 2003), seramik üretiminde (Ruiz and Maroto, 2006), çelik üretiminde (Tseng; Liao; Liao, 2008) ve biyosüreç (antibiyotik, antikor enzim üretimi) endüstrisinde (Gicquel; Hege; Minoux et al., 2012) esnek akış tipi çizelgeleme uygulamalarına rastlamak mümkündür.

Arthanary ve Ramaswamy (1971)[3] ve Salvador (1973)[4] bazı kaynaklarda (Linn and Zhang, 1999, p.57; Cheng; Karuno; Kise, 2001, p.140; Wang, 2005, p.78) esnek akış tipi çizelgeleme problemini ilk ortaya atan çalışmalar olarak ifade edilmektedir.

Linn ve Zhang (1999) esnek akış tipi çizelgeleme çalışmalarını üç ana başlıkta değerlendirmişlerdir: süreç karmaşıklığı, modelleme kriteri ve çözüm yöntemleri. Çalışmaları süreç karmaşıklığına göre kabaca üç gruba ayırmışlardır: iki aşamalı, üç aşamalı ve k aşamalı (k>3) esnek akış tipi çizelgeleme problemleri. Bu başlıklar altında literatürdeki çalışmaları incelemişlerdir. Çalışmaların daha çok iki aşamalı problemler üzerine yoğunlaştığını, bunların da çoğunun aşamalardaki özdeş paralel makineler olan çalışmalar olduğunu söylemektedirler. Maksimum tamamlanma zamanı minimizasyonunun en yaygın kullanılan çizelgeleme kriteri olduğunu ve en yaygın kullanılan çözüm yaklaşımının dal-sınır ve sezgisellerin olduğunu söylemektedirler.

---

[3] Arthanary, T.S. and K.G. Ramaswamy (1971), An extension of two machines sequencing problem, *Opsearch*, 8, 10-22.

[4] Salvador, M.S. (1973), A solution to a special case of flow shop scheduling problems, in *Symposium of Theory of Scheduling and Applications*, S.E. Elmaghraby (ed.), New York: Springer.

Wang (2005), literatürdeki esnek akış tipi çizelgeleme problemi ile ilgili çalışmaları optimum, sezgisel ve yapay zeka yöntemleri ile getirdikleri çözümlere göre incelemiştir.

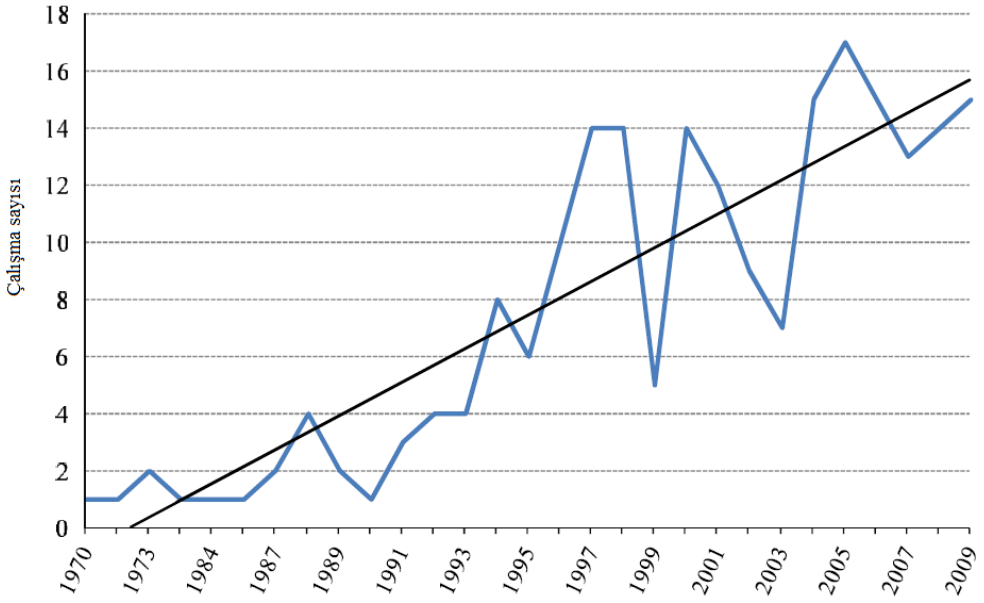
Kis ve Pesch (2005), tamamlanma zamanını ve ortalama akış zamanını minimize etmeyi amaçlayan esnek akış tipi çizelgeleme problemlerine kesin çözüm getiren çalışmaları incelemişlerdir. İki-aşamalı problemleri dahil etmeyip, ikiden çok aşamalı problemlere bakmışlardır.

Quadt ve Kuhn (2007b) esnek akış tipi çizelgeleme ile ilgili bir sınıflandırma çalışması yapmışlardır. Aşamaları sınırlandırılmış çalışmalar (2-aşamalı, 3 aşamalı gibi) ve işlem bölme içeren çalışmalar dahil edilmemiştir. İnceledikleri çalışmaları çözüm yöntemlerine optimal prosedürler ve sezgiseller olarak iki sınıfta incelemişlerdir. Sezgiseller; ayrışma yaklaşımları (decomposition approach) ve bütünsel (holistic) yaklaşımlar olarak ikiye ayrılmıştır. Ayrışma yaklaşımları iş odaklı, aşama odaklı ve problem odaklı olarak üçe ayrılmıştır.

Ribas; Leisten; Framinan (2010), esnek akış tipi çizelgeleme ile ilgili çalışmaları üretim bakış açısıyla dikkate almışlardır. Çalışmaların uygulaması olup olmamasından bağımsız olarak daha gerçekçi, üretim şartlarını dikkate alan çalışmalara bakmışlardır. Çalışmaları öncelikle problem karakteristiklerine, sonra da çözüm yaklaşımlarına göre sınıflandırmışlardır. Problem karakteristiklerinde makine karakteristikleri (özdeş, farklı hızda ve ilişkisiz paralel makineler) ve iş kısıtlarını (sıra bağımlı ve bağımsız olarak hazırlık zamanları, makine boşaltma ve taşıma süreleri, iş öncelikleri, parti büyüklüğü belirleme [lot sizing], parti bölme [lot splitting], ve yeniden işlemeli [re-entrant / recirculation]) dikkate alarak bir sınıflandırma yapmışlardır. Çözüm yaklaşımlarını ise kesin, sezgisel, melez ve simülasyon/karar destek sistemi şeklinde sınıflandırmışlardır. Sezgiselleri kendi içinde yapısal (constructive) ve iyileştirici (improvement) olarak ikiye ayırmışlar, metasezgiselleri, gelişimci sınıfı altında incelemişlerdir.

Ruiz ve Vazquez-Rodriguez (2010) esnek akış tipi çizelgeleme problemi için en kapsamlı ve güncel literatür taramasını yapmışlardır. 200'den fazla çalışmayı problem tipi (aşama sayısı), paralel makinelerin durumu, amaç fonksiyonu ve kullanılan yöntemlere göre incelemişlerdir. Kullanılan yöntemleri kesin, sezgisel ve metasezgisel algoritmalar başlıkları altında sınıflandırmışlardır. En kapsamlı ve güncel literatür taraması olmasından dolayı, bu çalışmadan bazı istatistiki grafikler burada da

verilecektir. Şekil 14'te esnek akış tipi çizelgeleme üzerine yapılan çalışma sayısının geçmiş on yıllarda arttığı görülmektedir (Ruiz and Vazquez-Rodriguez, 2010, p.9).



Şekil 14: Yıllara Göre Esnek Akış Tipi Çizelgeleme Çalışma Sayıları
Kaynak: Ruiz and Vazquez-Rodriguez. (2010, p.9).

Tablo 4'te incelenen çalışmaların aşama sayısına ve paralel makinelerin durumuna (özdeş, farklı hızda veya ilişkisiz) dağılımları verilmiştir. Çalışmaların dörtte birini özdeş paralel makineli 2 aşamalı problemler oluşturmaktadır. Üçte birini ise 2 aşamalı problemler oluşturmaktadır. Çalışmaların büyük çoğunluğunun (%83,72) aşamalarda özdeş makineler olan çalışmalar olduğu görülmektedir. İlişkisiz paralel makineli, *m*-aşamalı problemler en genel durumdur ve uygulamada karşılaşılması en muhtemel olandır. Dolayısıyla gerçek problemlerle ilgilenen çalışmaların çoğu *m*–aşamalı ve ilişkisiz paralel makinelidir (Ruiz and Vazquez-Rodriguez, 2010, p.9).

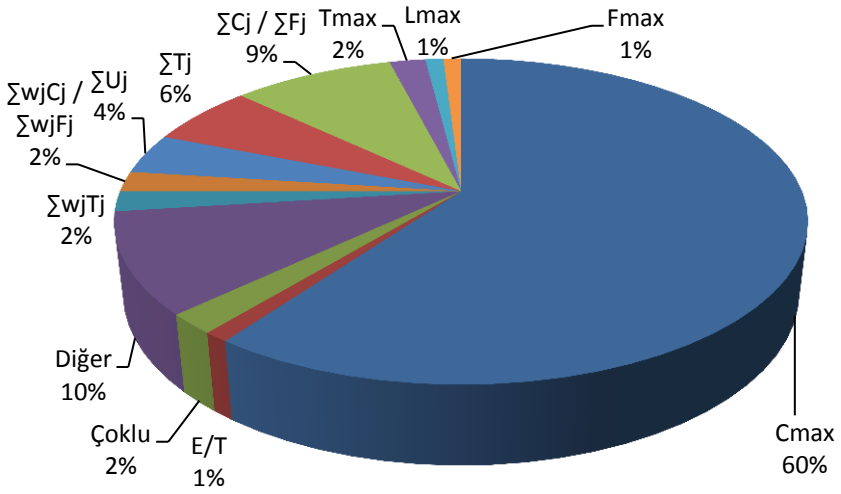
Tablo 4: İncelenen Çalışmaların Aşama ve Paralel Makine Durumuna Göre Yüzdesi

Aşama Sayısı	Paralel Makine Durumu			Toplam
	Özdeş	Farklı hızda	İlişkisiz	
2	25,12	1,86	4,65	31,63
3	4,19	1,4	0	5,59
m	54,41	1,4	6,97	62,78
Toplam	83,72	4,66	11,62	100

Kaynak: Ruiz and Vazquez-Rodriguez. (2010, p.9).

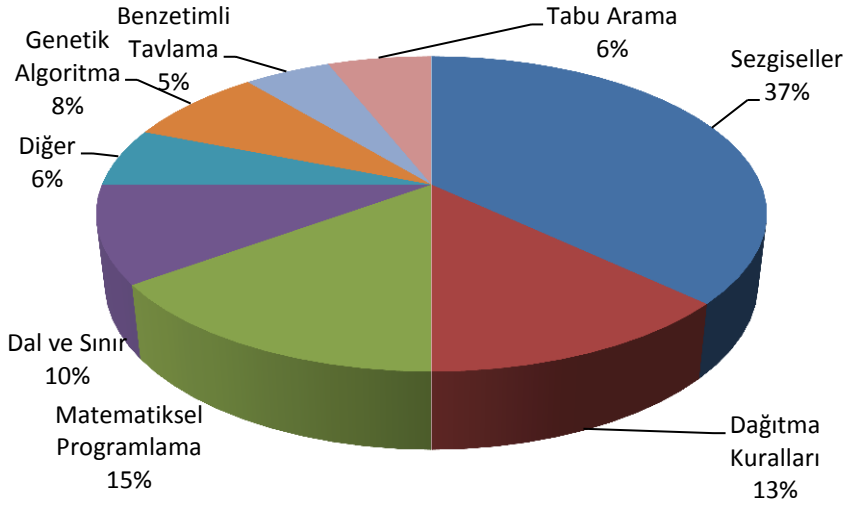
Şekil 15'de incelenen esnek akış tipi çizelgeleme çalışmalarının amaç fonksiyonlarına göre dağılımları verilmiştir. Maksimum tamamlanma zamanı (C_{max}) literatürde en çok

kullanılan kriterdir (%60). Toplam / ortalama tamamlanma veya akış zamanı ve bunların ağırlıklandırılmış halleri ($\sum C_j$, $\sum F_j$, $\sum w_j C_j$, $\sum w_j F_j$) tüm çalışmaların %11'ini oluşturmaktadır. İncelenen çalışmaların sadece %1'i erken tamamlanma / geç bitirme kriteri üzerinedir. Çok amaçlı problemlerle ilgilenen çalışma sayısı da düşüktür (%2). Diğer kriteri maliyet fonksiyonu ve probleme veya duruma özgü amaç fonksiyonlarını içermektedir (Ruiz and Vazquez-Rodriguez, 2010, p.9).



Şekil 15: Amaç Fonksiyonlarının Dağılımı
Kaynak: Ruiz and Vazquez-Rodriguez. (2010, p.9).

Şekil 16'da incelenen problemlerin çözümünde kullanılan tekniklerin dağılımı verilmiştir. Matematiksel programlama ve dal ve sınır yöntemi toplandığında incelenen çalışmaların dörtte birine tekabül etmektedir. Sezgiseller kategorisi altında bilinen bir metasezgisel şablonu içermeyen ve spesifik olan pek çok algoritma ve yöntem sınıflandırılmıştır. Çoğu zaman bu yöntemler büyük problemleri sınırlı bir performansla çözebilen tek geçişli sezgisellerdir. Dağıtma kurallarıyla birlikte sezgiseller incelenen çalışmaların %50'sini oluşturmaktadır. Geriye kalan %25'te benzetimli tavlama, tabu arama, genetik algoritma ve karınca kolonisi, parçacık sürü optimizasyonu gibi diğer metasezgisel yöntemlerdir (Ruiz and Vazquez-Rodriguez, 2010, p.9-13).



Şekil 16: Kullanılan Tekniklerin Dağılımı
Kaynak: Ruiz and Vazquez-Rodriguez. (2010, p.9).

Tablo 5'te bu tez çalışması için yukarıda bahsi geçen esnek akış tipi çizelgeleme problemi ile ilgili literatür çalışmalarından ve veritabanlarından aramalar sonucu ulaşılabilen 100'ün üzerinde çalışma süreç karmaşıklığına, çizelgeleme kriterine ve çözüm yöntemlerine göre verilmiştir. Uygulama problemi olan çalışmalar * ile gösterilmiştir.

Bu çalışmalar genel olarak aşama sayısını sınırlandırılan ve sınırlandırmayan çalışmalar olarak ikiye ayrılabilir. Aşama sayısı sınırlandırılmış çalışmalar, belli bir şekilde tanımlanmış esnek akış tipi çizelgeleme problemine genel bir çözüm getirmek yerine, 2 veya 3 aşamalı problemlere özgü çözüm getiren çalışmalardır. Bunlar içinde de genellikle 2 aşamalı problemler ağırlıktadır.

Tablo 5: Esnek Akış Tipi Çizelgeleme İle İlgili Literatür Sınıflandırması

Yazar(lar)	Süreç Karmaşıklığı	Çizelgeleme Kriteri(Tek/Çok)	Çözüm Yöntemi
* Narasimhan and Panwalkar, 1984	*FF*2(*Rm*$_2$)	İkinci aşamada makine atıl bekleme ve işlerin bekleme süreleri	Sezgisel
* Wittrock, 1985	*FF*3(*Pm*) aşama atlama	$C_{\max}$ ve ara stoklar	Sezgisel
Kochhar and Morris, 1987	*FFc*(*Pm*) |*block* , *brkdwn*|	Ortalama akış süresi	Sezgisel
Narasimhan and Mangiameli, 1987	*FF*2(*Pm*)	Siparişlerin bekleme süresi, makine atıl bekleme süresi, $C_{\max}$ ve makine tamamlanma zamanları ortalaması	Sezgisel
Gupta, 1988	*FF*2(*Pm*)	$C_{\max}$	Sezgisel
* Wittrock, 1988	*FF*3(*Pm*)	$C_{\max}$ ve ara stoklar	Sezgisel
Sriskandarajah and Sethi, 1989	*FF*2(*Pm*)	$C_{\max}$	Sezgisel
Brah and Hunsucker, 1991	*FFc*(*Pm*)	$C_{\max}$	Dal ve sınır
Gupta and Tunc, 1991	*FF*2(*Pm*$_2$)	$C_{\max}$	Sezgisel
Rajendran and Chaudhuri, 1992a	*FFc*(*Pm*)	Toplam akış zamanı	Dal ve sınır ile bir sezgisel
Rajendran and Chaudhuri, 1992b	*FFc*(*Pm*)	$C_{\max}$	Dal ve sınır
* Adler; Fraiman; Kobacker et al. 1993	*FF*3(*Rm*) | s_{jk} | aşama atlama	$\sum w_j T_j$, toplam hazırlık zamanı, ara stoklar	Sezgisel
Luss and Rosenwein, 1993	*FFc*(*Pm*)	$\sum w_j T_j$	Sezgisel
* Tsubone; Ohba; Takamuki et al. 1993	*FF*2	Hazırlık zamanı oranı, ara stoklar ve üretim oranı	Sezgisel
Chang and Liao, 1994	*FFc*(*Pm*)	Gecikme/erken tamamlanma, ara stok	Sezgisel
* Ding and Kittichartphayak, 1994	*FFc*(*Pm*)	$C_{\max}$	Sezgisel

Gupta and Tunc, 1994	$FF2(Pm_2)$	C_{max}	Sezgisel
Hunsucker and Shah, 1994	$FFc(Pm)$	C_{max}, ortalama ve maksimum akış zamanı	Sezgisel
Lee and Vairaktarakis, 1994	$FF2(Pm)$	C_{max}	Sezgisel
* Randhawa; Juwono; Burhanuddin, 1994	FFc	C_{max}, makine kullanım oranı	Sezgisel
Chen, 1995	$FF2(Pm_1)$	C_{max}	Sezgisel
Santos; Hunsucker; Deal, 1995	$FFc(Pm)$	C_{max}	Sezgisel
Sawik, 1995	$FF3(Pm)$ $\|block\|$	C_{max}	Sezgisel
Uetake; Tsubone; Ohba, 1995	$FF2(Rm_2)$	C_{max} ve ara stok	Sezgisel
Guinet and Solomon, 1996	$FFc(Pm)$	$\max(w_jT_j)$ ve C_{max}	Sezgisel
Guinet; Solomon; Kedia et al. 1996	$FF2(Pm)$	C_{max}	Karma tamsayılı programlama ve sezgisel
Hoogeveen; Lenstra; Veltman, 1996	$FF2(Pm_1)$ $\|prmp\|$	C_{max}	Güçlü anlamda NP-zor olduğunu göstermişlerdir
Santos; Hunsucker; Deal, 1996	$FFc(Pm)$	C_{max}	Sezgisel
Vignier; Billaut; Proust et al. 1996	$FF2(Pm)$	$\max(T_j)$	Sezgisel
Vignier; Dardilhac; Dezalay et al. 1996	$FFc(Pm)$	$\sum C_j$	Dal ve sınır
* Agnetis; Pacifici; Rossi et al. 1997	$FFc(Pm)$	$\sum U_j$ ve ara stok	Sezgisel
Gupta; Hariri; Potts, 1997	$FF2(Pm_1)$	C_{max}	Dal ve sınır, sezgisel
* Li, 1997	$FF2(Pm_2)$ $\|fmls, batch\|$	C_{max}	Sezgisel
Oğuz and Ercan, 1997	$FF2(Pm)$	C_{max}	Sezgisel
Vignier; Commandeur; Proust, 1997	$FFc(Pm)$	C_{max}	Dal ve sınır
* Huang and Li, 1998	$FF2(Qm_2)$ $\|fmls\|$	C_{max}	Sezgisel
Lee and Vairaktarakis,	$FF2(Pm)$	C_{max}	Sezgisel

1998			
Portmann; Vignier; Dardilhac et al. 1998	*FFc*(*Pm*)	$C_{\max}$	Dal ve sınır ile metasezgisel (genetik algoritma)
Yang, 1998	*FFc*(*Pm*) | r_i |	$\sum w_i T_i$	Sezgisel ile dal ve sınır
Brah and Loo, 1999	*FFc*(*Pm*)	$C_{\max}$ ve ortalama akış zamanı	Sezgisel
Verma and Dessouky, 1999	*FFc*(*Qm*) özdeş işler	$C_{\max}$	Sezgisel
Yu; Ohsato; Kawakami et al. 1999	*FFc*	$C_{\max}$ ve ortalama akış zamanı	Sezgisel
Botta-Genoulaz, 2000	*FFc*(*Pm*) |*prec*|	max($w_j T_j$)	Sezgisel
Carlier and Neron, 2000	*FFc*(*Pm*)	$C_{\max}$	Dal ve sınır
Jayamohan and Rajendran, 2000	*FF*3	Akış zamanı ve gecikme	Sezgisel
Koulamas and Kyparisis, 2000	*FF*2 *FF*3(*Pm*)	$C_{\max}$	Sezgisel
Liu and Chang, 2000	*FFc*(*Pm*) | s_{jk} |	Teslim tarihini karşılama, ara stoklar, hazırlık maliyetleri	Sezgisel
Moursli and Pochet, 2000	*FFc*(*Pm*)	$C_{\max}$	Dal ve sınır, alt ve üst sınırlar için sezgisel
Sawik, 2000	*FF*3(*Pm*) |*block*|	$C_{\max}$	Karma tamsayılı programlama
Yang; Kreipl; Pinedo, 2000	*FFc*(*Pm*) |r_j|	$\sum w_j T_j$	Sezgisel
Azizoğlu; Çakmak; Kondakci, 2001	*FFc*(*Pm*)	Toplam akış zamanı	Dal ve sınır
Cheng; Karuno; Kise, 2001	*FFc*(*Pm*)	max($w_j T_j$)	Sezgisel
Kyparisis and Koulamas, 2001	*FFc*(*Pm*)	$\sum w_j C_j$	Sezgisel
Negenman, 2001	*FFc*(*Pm*)	$C_{\max}$	Metasezgisel (çeşitli yerel arama algoritmaları)
Neron; Baptiste; Gupta, 2001	*FFc*(*Pm*)	$C_{\max}$	Dal ve sınır ile ilgili düzenlemeler
Soewandi and Elmaghraby, 2001	*FF*3(*Pm*)	$C_{\max}$	Sezgisel
Gupta; Krüger; Lauff	*FFc*(*Pm*) |*prmu*|	Erken tamamlanma	Sezgisel

et al. 2002	değişken işlem süreleri	ve gecikme cezaları, ağırlıklı tamamlanma zamanları ve teslim tarihi atama maliyetlerinin toplamı	
* Jin; Ohno; Sawik et al. 2002	*FF*3(*Pm*)	$C_{\max}$	Metasezgisel (genetik algoritma)
Sawik, 2002	*FF*3(*Pm*) |*block*|	$C_{\max}$	Karma tamsayılı programlama
Wang and Li, 2002	*FFc*(*Pm*)	$C_{\max}$	Metasezgisel (genetik algoritma)
* Alisantoso; Khoo; Jiang, 2003	*FFc*(*Pm*) aşama atlama	*Ortalama gecikme*	Metasezgisel (Bağışıklık algoritması)
Kurz and Askin, 2003	*FFc*(*Pm*) | s_{jk} | aşama atlama	$C_{\max}$	Sezgisel
Soewandi and Elmaghraby, 2003	*FF*2(Q_m)	$C_{\max}$	Sezgisel
* Lin and Liao, 2003	*FF*2(Pm_2) | s_{jk}, M_j |	max(w_jT_j)	Sezgisel
Engin and Döyen, 2004	*FFc*(*Pm*)	$C_{\max}$	Metasezgisel (yapay bağışıklık sistemi)
Kurz and Askin, 2004	*FFc*(*Pm*) | s_{jk} | aşama atlama	$C_{\max}$	Sezgisel
Logendran; Carson; Hanson, 2005	*FFc*(*Pm*)	$C_{\max}$	Sezgisel
Oğuz and Ercan, 2005	*FFc*(*Pm*)	$C_{\max}$	Metasezgisel (genetik algoritma)
Wang; Xing; Bai, 2005	*FF2*(*Pm*) |*nwt*| atıl zaman yok	$C_{\max}$	Sezgisel
Xie and Wang, 2005	*FF*2(*Pm*) |*brkdwn*|	$C_{\max}$	Sezgisel
Akrami; Karimi; Hosseini, 2006	*FFc*(*Pm*) |*block*|	Hazırlık maliyetleri, ara stok ve bitmiş ürün stok maliyetleri	Karma tamsayılı lineer olmayan programlama, metasezgisel (genetik algoritma ve tabu arama)
Allaoui and Artiba, 2006	*FF*2(Pm_2) |*brkdwn*|	$C_{\max}$	Dal ve sınır ile sezgisel
Crowder, 2006	*FFc*(*Qm*) |s_{jk}, *block* |	$C_{\max}$	Metasezgisel (başlangıç çözümü için sezgisel, sonrasında benzetimli tavlama)
Haouri; Hidri; Gharbi, 2006	*FF*2(*Pm*)	$C_{\max}$	Dal ve sınır
Kyparisis and	*FFc* (*Qm*)	$C_{\max}$	Sezgisel

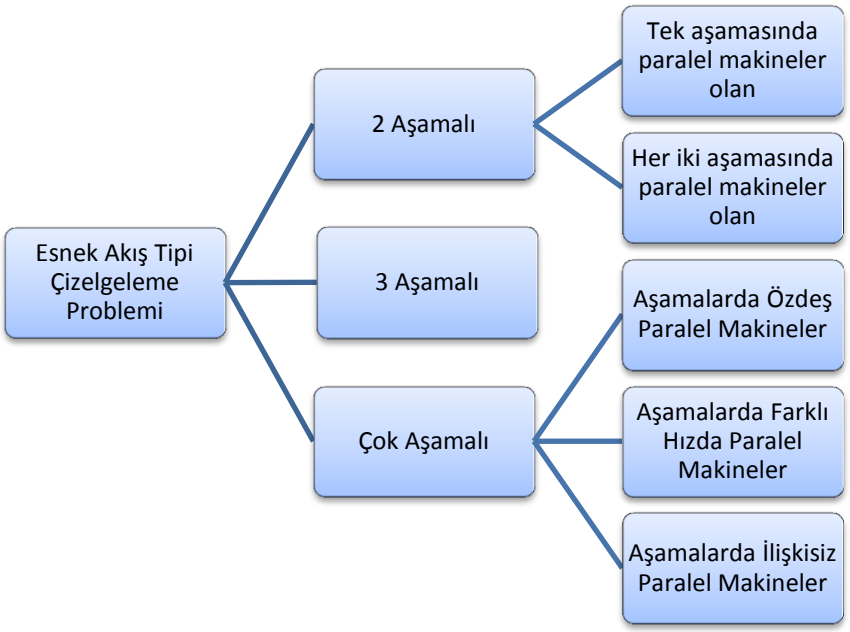
Koulamas, 2006a			
Kyparisis and Koulamas, 2006b	*FF*2 (*Qm*)	C_{max}	Sezgisel
Logendran; deSzoeke; Barnard, 2006	*FFc*(*Pm*) | s_{jk} |	C_{max}	Metasezgisel (tabu arama)
* Ruiz and Maroto, 2006	*FFc*(*Rm*) | s_{jk} |	C_{max}	Metasezgisel (genetik algotirma)
Zandieh; Fatemi Ghomi; Moattar Hosseini, 2006	*FFc*(*Pm*) | s_{jk} |	C_{max}	Metasezgisel (bağışıklık algoritması)
Caricato; Grieco; Serino, 2007	*FF*2(Pm_1)	C_{max}	Sezgisel (gezgin satıcı problemine dayalı) ve karma tamsayılı programlamaya dayalı
Janiak; Kozan; Lichtenstein et al. 2007	*FFc*(*Pm*) | r_j |	$\sum w_jE_j$, $\sum w_jT_j$ ve toplam ağırlıklı bekleme süresi	Sezgisel ve metasezgisel (tabu arama ve benzetimli tavlama)
Jenabi; Fatemi Ghomi; Torabi et al. 2007	*FFc*(*Rm*)	Toplam hazırlık ve stok maliyetleri	Karma 0-1 lineer olmayan programlama ve metasezgisel (genetik algoritma melezi ve benzetimli tavlama)
Quadt and Kuhn, 2007a	*FFc*	Hazırlık maliyetleri ve ortalama akış zamanı	Metasezgisel (genetik algoritma)
Paternina-Arboleda; Montoya-Torres; Acero Dominguez et al. 2008	*FFc* (*Pm*)	C_{max}	Sezgisel
Ruiz; Şerifoğlu; Urlings, 2008	*FFc*(*Rm*) |r_j, s_{jk}, M_j, *prec*| aşama atlama	C_{max}	Karma tamsayılı programlama ve sezgisel
Shiau; Cheng; Huang, 2008	*FFc*(*Pm*) Oransal	$\sum w_jC_j$	Metasezgisel (genetik algoritma ve tabu arama melezi)
* Tseng; Liao; Liao, 2008	*FF*2(Pm_2) aşama atlama	C_{max}	Sezgisel
Amin-Naseri and Beheshti-Nia, 2009	*FFc*(*Qm*) |*batch*|	C_{max}	Karma tamsayılı programlama, sezgisel ve metasezgisel (üç boyutlu genetik algoritma)
Behnamian; Fatemi Ghomi; Zandieh, 2009	*FFc* | s_{jk} |	Çok amaçlı C_{max} ve $\sum(E_j+T_j)$	Metasezgisel (benzetimli tavlama ve değişken komşuluk arama melezi)
Choi and Lee, 2009	*FF*2(*Pm*)	U_j	Dal ve sınır, sezgisel
Figielska, 2009	*FF*2(Rm_1) |*prmp*|	C_{max}	Metasezgisel (genetik algoritma ve benzetimli tavlama)

Jouglet; Oğuz; Sevaux, 2009	$FFc(Pm)$	$C_{\max}$	Metasezgisel (memetik algoritma)
Jungwattanakit; Reodecha; Chaovalitwongse et al. 2009	$FFc(Rm) \mid s_{jk} \mid$	$C_{\max}$ ve U_j	Karma tamsayılı programlama, sezgisel ve metasezgisel (benzetimli tavlama, tabu arama ve genetik algoritma)
Karimi; Zandieh; Kar-Amooz, 2009	$FFc(Pm) \mid s_{jk} \mid$	Çok amaçlı $C_{\max}$ ve $\sum w_j T_j$	Metasezgisel (genetik algoritma)
Kim; Joo; Shin, 2009	$FF2(Pm_1) \mid r_j \mid$	$C_{\max}$	Sezgisel
Naderi; Zandieh; Shirazi, 2009	$FFc(Pm) \mid s_{jk} \mid$ taşıma zamanı	$\sum w_j T_j$	Karma tamsayılı lineer programlama ve metasezgisel (elektromanyetizma algoritması)
Tavakkoli-Moghaddam; Safaesi; Sassani, 2009	$FFc(Pm) \mid block \mid$	$C_{\max}$	Metasezgisel (memetik algoritma – genetik algoritma ve değişken komşuluk arama melezi)
Zandieh and Gholami, 2009	$FFc(Pm) \mid brkdwn, s_{jk} \mid$	$C_{\max}$	Metasezgisel (bağışıklık algoritması)
Khalouli; Ghedjati; Hamzaoui, 2010	$FFc(Pm)$	Erken tamamlanma geç bitirme cezaları toplamı	Metasezgisel (karınca kolonisi optimizasyonu)
Urling; Ruiz; Stützle, 2010	$FFc(Rm) \mid r_j, prec, s_{jk}, M_j \mid$	$C_{\max}$	Metasezgisel
Behnamian and Fatemi Ghomi, 2011	$FFc(Rm) \mid s_{jk} \mid$	$C_{\max}$ ve toplam kaynak tahsis maliyeti (çok amaçlı)	Metasezgisel (genetil algoritma ve değişken komşuluk arama melezi)
Behnamian and Zandieh, 2011	$FFc(Pm) \mid s_{jk} \mid$ Sınırlı bekleme süresi	Erken tamamlanma ve karesel geç bitirme cezaları	Karma tamsayılı doğrusal programlama ve metasezgisel (kesikli koloni rekabetçi algoritma)
Choong; Phon-Amnuaisuk; Alias, 2011	$FFc(Pm)$	$C_{\max}$	Metasezgisel (parçacık sürü optimizasyonu, benzetimli tavlama ve tabu arama melezi)
Hidri and Haouari, 2011	$FFc(Pm)$	$C_{\max}$	Sezgisel
Hmida; Haouari; Huguet et al. 2011	$FF2(Pm)$	$C_{\max}$	Sezgisel
Yang, 2011	$FF2 \mid M_j \mid$	$C_{\max}$	Sezgisel
Carpov; Carlier; Nace et al. 2012	$FF2(Pm_2) \mid nwt \mid$	$C_{\max}$	Sezgisel

* Gicquel; Hege; Minoux et al. 2012	*FF4*(*Pm*) |*block*, *nwt*|	$\sum w_j T_j$	Karma tamsayılı doğrusal programlama
Liao; Tjandradjaja; Chung, 2012	*FFc*(*Pm*)	C_{max}	Metasezgisel (parçacık sürü optimizasyonu ile darboğaz sezgiseli melezi, yerel optimumdan kaçınmak için benzetimli tavlama)
Niu; Zhou; Fei et al. 2012	*FFc*(*Pm*)	Ortalama akış zamanı	Metasezgisel (bağışıklık ve quantum algoritmaları melezi)
Wang and Liu, 2013a	$FF2(Pm_2) \mid M_j \mid$	C_{max}	Sezgisel (dal ve sınıra dayalı)
Wang and Liu, 2013b	$FF2(Pm_2) \mid nwt \mid$	C_{max}	Metasezgisel (genetik algoritma)

* uygulamalı çalışmalar

Tablo 5'te verilmiş olan literatürdeki çalışmalardan Şekil 17'deki aşama sayıları ve paralel makine durumuna göre yapılmış sınıflandırma esas alınarak kısaca bahsedilecektir.



Şekil 17: Literatür Sınıflandırması

2.2.3.1 2 Aşamalı Esnek Akış Tipi Çizelgeleme Çalışmaları

2.2.3.1.1 Tek Aşamasında Özdeş Paralel Makineler Olan Çalışmalar

İlk olarak iki aşamalı, yalnız bir aşamasında özdeş paralel makineler bulunan, maksimum tamamlanma zamanını minimize etmeyi amaçlayan problemler ele alınacaktır.

Gupta ve Tunc (1991), iki aşamalı, ikinci aşamada özdeş paralel makineler bulunan akış tipi çizelgeleme problemine sezgisel yöntemlerle çözüm aramışlardır. İkinci aşamadaki makinelerin sayısı toplam iş sayısına eşit veya ondan daha fazla olduğu durumlarda EUİZ kuralının optimum sonuç verdiği ifade edilmiştir. Makine sayısının iş sayısından az olduğu durumlar için iki sezgisel algoritma önerilmiş ve maksimum tamamlanma zamanını minimize etme performansına göre değerlendirilmiştir. İş sayısındaki artışa bağlı olarak algoritmaların etkinliği de artmaktadır. Sekiz veya daha az iş olan problemlerde sezgisel yöntemlerin sonuçlarının alt sınırdan sapmasına çare olarak sezgiseli düzenleyecek bir dal ve sınır algoritması eklenmiştir. Aynı yazarlar başka bir çalışmalarında (1994), iki aşamalı, ayrı hazırlık zamanları olan esnek akış tipi çizelgeleme problemine sezgisel yöntemlerle çözüm getirmişlerdir. Problemin ilk aşamada tek makine, ikinci aşamada ise iş sayısından daha fazla paralel özdeş makine olması durumunda optimal çözüm veren bir algoritma geliştirmişlerdir. İkinci aşamadaki paralel makine sayısının iş sayısından daha az olduğu durumlar için sezgiselleri kullanmışlardır.

Chen (1995), iki aşamalı ve ilk aşamasında özdeş paralel makineler olan, ikinci aşamasında tek makine bulunan esnek akış tipi çizelgeleme problemi için bazı sezgisel yöntemleri karşılaştıran bir çalışma yapmıştır.

Hoogeveen; Lenstra; Veltman (1996), iki aşamalı ve aşamalarının en az birinde iki özdeş paralel makine olan işlem bölmeli esnek akış tipi çizelgeleme probleminin güçlü anlamda NP-zor olduğunu göstermişlerdir.

Gupta; Hariri; Potts (1997), ilk aşamasında çok sayıda özdeş paralel makine olan, ikinci aşamasında tek makine olan, iki aşamalı esnek akış tipi çizelgeleme problemine maksimum tamamlanma zamanını minimize etmeye çalışan dal ve sınır algoritması, bu

algoritmanın yetmediği yerler için de iki sezgisel yöntem geliştirmişlerdir. Yöntemleri 50 işten 250 işe, ilk aşamada 2'den 10 paralel makineye kadar kombinasyonlu test problemleriyle karşılaştırmışlardır.

Li (1997), uçak motorları üreten bir işletmedeki gerçek bir uygulama problemden yola çıkarak, iki aşamalı, ilk aşamasında tek makine, ikinci aşamasında özdeş paralel makineler bulunan, ürün aileleri, hazırlık zamanları, yığın işleme ve parilere ayırma bulunan esnek akış tipi çizelgeleme problemine sezgisel yöntemlerle çözüm getirmiştir.

Allaoui ve Artiba (2006), ilk aşamasında tek makine, ikinci aşamasında özdeş paralel makineler olan iki aşamalı, arızalar ve bakımlardan dolayı makinelerin müsait olmama durumlarının olduğu esnek akış tipi çizelgeleme problemi için maksimum tamamlanma zamanını minimize etmeyi amaçlayan küçük çaplı problemler için dal ve sınır algoritması geliştirmişler; büyük problemler için ise üç sezgisel yöntemin en kötü durum performanslarını[5] (worst case performance bound) incelemişlerdir.

Caricato; Grieco; Serino (2007), ilk aşamasında paralel özdeş iki makine, ikinci aşamasında tek makine olan iki aşamalı esnek akış tipi çizelgeleme problemi için maksimum tamamlanma zamanını minimize etmeyi amaçlayan bazı sezgisel yöntemlerin performansını karma tamsayılı programlama ile elde ettikleri optimal çözümlerle karşılaştırmışlardır.

Tseng; Liao; Liao (2008) iki aşamalı, ilk aşamasında tek makine, ikinci aşamasında özdeş paralel makineler olan, bazı işlerin ilk aşamayı atladığı, bir çelik fabrikasındaki esnek akış tipi çizelgeleme problemine belli bir permutasyon çizelgeyi permutasyon olmayan çizelgeye çeviren ve maksimum tamamlanma zamanını minimize etmeyi amaçlayan bir sezgisel yöntem geliştirmişlerdir.

---

[5] Bir algoritmanın bulduğu optimum çözüm $C_{max} \leq pC^*_{max}$ ise o algoritmanın en kötü durum performans sınırı p olduğu söylenir. Bu tanımdan her zaman $p \geq 1$ olacağı görülebilir. Eğer tüm $\varepsilon > 0$ için $C_{max} > (p - \varepsilon)C^*_{max}$'ı sağlayan I durumu (instance) varsa sınıra *sıkı* (tight) denir (Xie and Wang, 2005, p.1631).
S_H bir H sezgiselinin ürettiği çizelge, S^* ise optimum çizelge olmak üzere herhangi bir problem için $C_{max}(S_H)/C_{max}(S^*) \leq p$ durumu için H sezgiselinin en kötü durum performansı p'dir. Herhangi bir problem için $C_{max}(S_H) - C_{max}(S^*) \leq \alpha$ durumu için H sezgiselinin mutlak performans garantisi α'dır (Kyparisis and Koulamas, 2006a, p.986),

Kim; Joo ve Shin (2009), iki aşamalı, ilk aşamasında paralel özdeş makineler, ikinci aşamasında tek makine olan ve hazır olma zamanı ile ürün karması oranı sınırları bulunan bir akış tipi çizelgeleme problemi için maksimum tamamlanma zamanını minimize etmeyi amaçlayan sezgisel yöntemler geliştirmişlerdir.

Yang (2011), iki aşamalı, ilk aşamasında tek makine, ikinci aşamasında tahsis edilmiş iki makine bulunan esnek akış tipi çizelgeleme problemi için tamamlanma zamanını minimize etmeyi amaçlayan iki sezgisel yöntem geliştirmiş ve en kötü durum sınırları bulmuştur.

Carpov; Carlier; Nace vd. (2012), iki aşamalı, ilk aşamasında tek, ikinci aşamasında özdeş paralel makineler bulunan, beklemeli ve beklemesiz iki esnek akış tipi çizelgeleme problemi için maksimum tamamlanma zamanını minimize etmeyi amaçlayan bir sezgisel yöntem geliştirmişlerdir.

Wang ve Liu (2013a), iki aşamalı, ilk aşamasında tek, ikinci aşamasında ise işlere tahsis edilmiş paralel makineler bulunan esnek akış tipi çizelgeleme problemi için maksimum tamamlanma zamanını minimize etmeyi amaçlayan dal ve sınır temelli bir sezgisel yöntem geliştirmişlerdir. Sezgiselin 60 saniye içinde optimale %2 yaklaşan çözümler bulabildiği ifade edilmektedir. Aynı problemin ilk aşamasında tek makine ikinci aşamasında özdeş paralel makineler olan beklemesiz bir türü için (Wang and Liu, 2013b) maksimum tamamlanma zamanını minimize etmeyi amaçlayan bir genetik algoritma geliştirmişlerdir.

Lin ve Liao (2003), etiket üreten bir tesisinin iki aşamalı, ilk aşamada sıra bağımlı hazırlık zamanı olan, ikinci aşamada tahsis edilmiş makineleri olan esnek akış tipi çizelgeleme problemine sezgisel bir yöntemle çözüm getirmişlerdir. Amaç ağırlıklı maksimum gecikmeyi minimize edecek şekilde günlük ürün karmasını çizelgelemektir. Sezgisel yöntem ilk aşamada toplam hazırlık zamanı, ikinci aşamada ilk giren ilk çıkar kuralının birleşiminden oluşmaktadır. Sezgisel yöntemin performansı küçük çaplı problemlerin optimal çözümleriyle ve işletmenin mevcut çizelgeleme yönteminin sonucuyla karşılaştırılarak değerlendirilmiştir.

Huang ve Li (1998), Li (1997) çalışmasındaki probleme çok benzer iki aşamalı, ilk aşamasında tek, ikinci aşamasında farklı hızlı paralel makineler bulunan, ürün aileleri

olan ve bunlar arasında hazırlık süreleri olan maksimum tamamlanma zamanını minimize etmeyi amaçlayan esnek akış tipi çizelgeleme problemi için iki sezgisel, sekiz tane de sıralama kuralını karşılaştırmışlardır.

Narasimhan ve Panwalkar (1984), iki aşamalı, ilk aşamasında tek makine, ikinci aşamasında ilişkisiz paralel makineler olan (kablo üretim tesisinde uygulaması olan), ikinci aşamada makinelerin ve işlerin bekleme sürelerini minimize etmeyi amaçlayan esnek akış tipi çizelgeleme problemi için farklı sıralama kurallarını karşılaştırmışlardır.

Uetake; Tsubone; Ohba (1995), iki aşamalı, ilk aşamasında tek makine, ikinci aşamasında ise ilişkisiz paralel makineler bulunan, maksimum tamamlanma zamanı ve ara stok miktarının minimize edilmesinin amaçlandığı esnek akış tipi çizelgeleme problemi için sıralama kurallarının etkisini karşılaştırmışlardır.

Figielska (2009), kaynak kısıtlı, iki aşamalı, ilk aşamasında ilişkisiz paralel makineler bulunan, ikinci aşamasında tek makine bulunan, işlem bölmenin olduğu, esnek akış tipi çizelgeleme problemi için tamamlanma zamanını minimize etmeye çalışan bir genetik algoritma ve benzetimli tavlama algoritması geliştirmiştir.

2.2.3.1.2 Her İki Aşamasında Özdeş Paralel Makineler Olan Çalışmalar

İki aşamalı, her iki aşamasında da özdeş paralel makineler olan, maksimum tamamlanma zamanını minimize etmeyi amaçlayan esnek akış tipi çizelgeleme problemini ele alan çalışmalara aşağıda değinilmiştir.

Gupta (1988), iki aşamalı, her aşamada çoklu makine olan esnek akış tipi çizelgeleme problemini ele almış ve bu problemin NP-complete olduğunu ifade etmiştir. Problemin ikinci aşamasında tek makine olan özel bir durum için yaklaşık sonuç bulan etkin bir sezgisel yöntem geliştirmiştir.

Sriskandarajah ve Sethi (1989), iki aşamalı maksimum tamamlanma zamanını minimize etme amaçlı esnek akış tipi çizelgeleme problemi için sezgisel yöntemler geliştirmişlerdir.

Lee ve Vairaktarakis (1994), iki aşamalı, aşamalarında paralel özdeş makineler olan iki aşamalı esnek akış tipi çizelgeleme problemi için maksimum tamamlanma zamanını

minimize etmeyi amaçlayan bir sezgisel yöntem geliştirmişlerdir. Bu sezgisel yöntem üzerinden ikiden daha fazla aşamalı problemler için başka bir sezgisel yöntem daha geliştirmişlerdir. Lee ve Vairaktarakis (1998), her iki aşamasında da özdeş paralel makineler bulunan, sadece ilk aşamasında özdeş paralel makineler bulunan ve sadece ikinci aşamasında özdeş paralel makineler bulunan iki aşamalı esnek akış tipi çizelgeleme problemleri için sezgisel yöntemler geliştirmişlerdir.

Guinet; Solomon; Kedia et al. (1996), maksimum tamamlanma zamanını minimize etmeyi amaçlayan iki aşamalı, aşamalarında özdeş paralel makineler olan esnek akış tipi çizelgeleme problemi için karma tamsayılı programlama modeli kurmuşlar, fakat problem güçlü anlamda NP-zor olduğu için ilk aşamada sıralama, ikinci aşamada ise makinelere atama yapan bir sezgisel yöntem geliştirmişlerdir.

Oğuz ve Ercan (1997), iki aşamalı, aşamalarında özdeş paralel makineler olan ve birim işlem süresine sahip esnek akış tipi çizelgeleme problemi için maksimum tamamlanma zamanını minimize etmeyi amaçlayan sezgisel bir yöntem geliştirmişlerdir.

Koulamas ve Kyparisis (2000), iki ve üç aşamalı, aşamalarında özdeş paralel makineler bulunan ve maksimum tamamlanma zamanını minimize etmeyi amaçlayan esnek akış tipi çizelgeleme problemleri için sezgisel yöntemler geliştirmişler ve bu yöntemlerin asimptotik optimal sonuçlar verdiğini yani iş sayısı arttıkça sonuçların optimale yaklaştığını ifade etmektedirler.

Wang; Xing; Bai (2005), iki aşamalı, ilk aşamasında m, ikinci aşamasında m+1 özdeş paralel makine olan, beklemesiz ve ikinci aşamada makinelerin atıl beklemesinin olmadığı esnek akış tipi çizelgeleme problemi için maksimum tamamlanma zamanını minimize etmeyi amaçlayan sezgisel bir algoritma geliştirmişlerdir.

Xie ve Wang (2005), iki aşamalı, aşamalarında özdeş paralel makineler olan, makinelerin bakım arıza gibi nedenlerden her zaman müsait olmadığı esnek akış tipi çizelgeleme probleminin karmaşıklığını ele alıp, sezgisel algoritmaların en kötü durum sınırlarını bulmuşlardır. İki aşamalı, aşamalarının herhangi birinde özdeş paralel makineler bulunan esnek akış tipi çizelgeleme problemi NP-zor olduğu için, bu probleme bir de makinelerin müsait olma durumu eklendiğinde güçlü anlamda NP-zor hale gelmektedir.

Haouri; Hidri; Gharbi (2006), iki aşamalı, aşamalarında özdeş paralel makineler olan esnek akış tipi çizelgeleme problemi için maksimum tamamlanma zamanını minimize etmeyi amaçlayan bir dal ve sınır algoritması geliştirmişlerdir.

Hmida; Haouari; Huguet vd. (2011), iki aşamalı, aşamalarında özdeş paralel makineler olan esnek akış tipi çizelgeleme problemi için maksimum tamamlanma zamanını minimize etmeyi amaçlayan, Johnson Kuralı'na dayanan bir sezgisel algoritma geliştirmişlerdir.

İki aşamalı, aşamalarında özdeş paralel makineler olan maksimum tamamlanma zamanı dışında bir kriteri minimize etmeyi amaçlayan esnek akış tipi çizelgeleme problemlerini ele alan çalışmalar aşağıda özetlenmiştir.

Narasimhan ve Mangiameli (1987), iki aşamalı esnek akış tipi çizelgeleme problemi için beş farklı kriteri (siparişlerin bekleme zamanı ve makine atıl zamanları toplamı, siparişlerin bekleme zamanı, makine atıl zamanı, tamamlanma zamanı ve ortalama makine tamamlanma zamanı) minimize etmeyi amaçlayan bir sezgisel yöntem geliştirmişlerdir. Bu yöntem daha önce Narasimhan ve Panwalkar'ın (1984) kullandığı sezgiselin geliştirilmiş halidir ve diğer sezgisellere göre tüm kriterler açısından daha iyi sonuçlar vermektedir.

Vignier; Billaut; Proust vd. (1996), iki aşamalı, ilk aşamasında paralel özdeş makineler olan, ikinci aşamasında tek makine olan, maksimum gecikmeyi minimize etmeyi amaçlayan esnek akış tipi çizelgeleme problemi için iki sezgisel yöntem geliştirmişlerdir. İlk sezgiselde tüm makineler müsait olduğu durum, ikincisinde ise bakım politikaları dikkate alınarak paralel makinelerden birinin zaman zaman müsait olmadığı durum ele alınmıştır.

Choi ve Lee (2009), iki aşamalı, aşamalarında özdeş paralel makineler olan esnek akış tipi çizelgeleme problemi için geciken iş sayısını minimize etmeyi amaçlayan küçük çaplı problemler için dal ve sınır algoritması, büyük çaplı problemler için ise sezgisel model geliştirmişlerdir.

Soewandi ve Elmaghraby (2003), *n* iş, iki aşamalı, her aşamasında farklı hızda paralel makineler bulunan esnek akış tipi çizelgeleme problemini için maksimum tamamlanma

zamanını minimize etmeyi amaçlayan sezgisel yöntemler geliştirmişlerdir. Bunlardan birisi en kötü durum performans sınırı oluşturmaktadır. Kyparisis ve Koulamas (2006b), iki aşamalı, aşamalarında farklı hızlı paralel makineler bulunan esnek akış tipi çizelgeleme problemi için tamamlanma zamanını minimize etmeyi amaçlayan Soewandi ve Elmaghraby'nin (2003) sezgiselini temel alan makine hızına bağımlı ve makine hızından bağımsız sezgisel yöntemler geliştirmişlerdir.

Tsubone; Ohba; Takamuki et al. (1993), fotoğraf filmi üretiminde uygulaması olan iki aşamalı esnek akış tipi çizelgeleme problemi için hazırlık zamanının işlem zamanına oranı, ulaşılabilen üretimin planlanan üretime oranı ve ara stokları azaltmayı amaçlayan bir sezgisel yöntem geliştirmişlerdir. Sonuçta her üç kriter açısından da iyileşme elde etmişlerdir. Bunun yanında eskiden 2 gün olan çizelge oluşturmak için harcanan zaman yarım güne düşürülmüştür.

2.2.3.2 3 Aşamalı Esnek Akış Tipi Çizelgeleme Çalışmaları

Wittrock (1985), PCB üreten bir tesiste üç aşamalı, aşamalarında paralel özdeş makineler bulunan, aşama atlamalı, maksimum tamamlanma zamanını ve ara stok miktarını minimize etmeyi amaçlayan esnek akış tipi çizelgeleme problemi için üç aşamalı bir sezgisel yöntem geliştirmişlerdir. İlk aşama makine tahsisi, ikinci aşama sıralama, üçüncü aşama ise zamanlamadır. Periyodik çizelgeleme yapılmıştır, işlerin küçük bir kümesi bir defa çizelgelenip, bu çizelge defalarca tekrarlanmıştır. Yine benzer problem bu sefer periyodik değil normal bir çizelgeleme ile benzer üç aşamalı yöntemle çözülmüştür (Wittrock, 1988).

Sawik (1995), üç aşamalı ve aşamalarında özdeş paralel makineler olan, ara stok olmayan, makine bloke etmeli, maksimum tamamlanma zamanını minimize etmeyi amaçlayan esnek akış tipi çizelgeleme problemi için sezgisel bir yöntem geliştirmiştir. Aynı yazar buna benzer başka bir çalışmasında (2000), sınırlı ara stoklu ve ara stoksuz, makine bloke etmeli esnek akış tipi çizelgeleme problemi için karma tamsayılı programlama modeli geliştirmiştir. Ara stoksuz model için 3 aşamalı, sınırlı ara stoklu model için 5 aşamalı bir probleme çözüm getirmiştir. Diğer bir çalışmada ise (Sawik, 2002), sınırlı ara stoklu, yığın çizelgeleme yaptığı tamamlanma zamanını minimize etmeyi amaçlayan esnek akış tipi çizelgeleme problemine karma tamsayılı programlama ile çözüm getirmiştir.

Soewandi ve Elmaghraby (2001), *n* iş, üç aşamalı ve her aşamada paralel özdeş makineler olan esnek akış tipi çizelgeleme problemi için maksimum tamamlanma zamanını minimize etmeyi amaçlayan sezgisel yöntemler geliştirip, bunları karşılaştıran bir çalışma yapmışlardır. Üç aşamalı problemi, iki aşamalı esnek akış tipi ve son aşamayı da paralel makinelerden oluşan tek aşama şeklinde iki parçaya ayıran atölye ayırma (shop partitioning) olarak adlandırdıkları sezgisel bir yöntem kullanmışlardır. Bunun haricinde önce problemi üç aşamalı bir akış tipi problem olarak ele alıp işleri sıralayan, sonra bu işleri her aşamada makinelere dağıtan bir sezgisel yöntem kullanmışlardır.

Jin; Ohno; Sawik vd. (2002), ilk aşamasında 6, ikinci aşamasında 10, üçüncü aşamasında 13 özdeş paralel makine bulunan 3 aşamalı baskı devre üreten bir tesisteki esnek akış tipi çizelgeleme problemine genetik algoritma ile çözüm getirmişlerdir.

Gicquel; Hege; Minoux vd. (2012), biyosüreç endüstrisinde (antibiyotik, antikor, enzim üretimi) dört aşamalı, aşamalarında özdeş paralel makineler olan, bazı aşamalar arasında beklemesiz, bir aşamada ise sınırlı bekleme süresi olan bir esnek akış tipi çizelgeleme problemi için toplam ağırlıklı gecikmeyi minimize etmeyi amaçlayan karma tamsayılı doğrusal programlama modeli geliştirmişlerdir.

Adler; Fraiman; Kobacker et al. (1993), esnek akış tipi ortama uyan bir paketleme tesisi için çizelgeleme destek sistemi geliştirmişlerdir. Öncelikli amaç toplam gecikmeyi minimize etmek, ikincil amaç toplam hazırlık zamanlarını minimize etmek, üçüncül amaç ise ara stok miktarını minimize etmektir. Üç aşamalı ve aşamalarında ilişkisiz ve farklı sayıda paralel makineler bulunan problemde her iş her aşamada işlem görmemektedir (aşama atlama) ve sıra bağımlı hazırlık zamanları vardır.

Jayamohan ve Rajendran (2000) akış zamanı ve gecikme kriterlerinin minimize edilmesinin amaçlandığı üç aşamalı esnek akış tipi çizelgeleme problemi için sıralama kurallarına bağlı iki yaklaşım geliştirmişlerdir. Birinde her aşamada farklı bir sıralama kuralı kullanılırken, diğerinde her aşama için aynı sıralama kuralı kullanılmıştır.

2.2.3.3 Çok Aşamalı Esnek Akış Tipi Çizelgeleme Çalışmaları

2.2.3.3.1 Özdeş Paralel Makineli Çalışmalar

Aşamalarında özdeş paralel makineler bulunan ve maksimum tamamlanma zamanını minimize etmeyi amaçlayan çok aşamalı esnek akış tipi çizelgeleme çalışmaları aşağıda verilmiştir.

Barh ve Hunsucker (1991), Rajendran ve Chaudhuri (1992b), Vignier; Commandeur; Proust (1997), Carlier ve Neron (2000) ve Moursli ve Pochet (2000), aşamalarında özdeş paralel makineler olan esnek akış tipi çizelgeleme problemleri için tamamlanma zamanını minimize etmeyi amaçlayan dal ve sınır algoritmasına dayalı çözümler geliştirmişlerdir. Vignier; Commandeur; Proust'un (1997) geliştirdiği yöntem aynı zamanda alt sınırların bulunmasında da kullanılmıştır. Neron; Baptiste; Gupta (2001), aynı tip problemde dal ve sınır algoritması için iyileştirmeler geliştirmişlerdir.

Ding ve Kittichartphayak (1994), aşamalarında özdeş paralel makineler olan esnek akış tipi çizelgeleme problemleri için maksimum tamamlanma zamanını minimize etmeyi amaçlayan üç ayrı sezgisel yöntemi karşılaştırmışlardır. Bu üç sezgiselin kombinasyonuyla oluşturulan sezgisel yöntem optimale yakın sonuçlar vermektedir. Bu yöntemle traktör imalatı ile ilgili bir uygulama problemini de çözmüşlerdir.

Santos; Hunsucker; Deal (1995), maksimum tamamlanma zamanını minimize etmeyi amaçlayan ve optimal sonucu bilinmeyen esnek akış tipi çizelgeleme problemleri için sezgisel yöntemlerin etkinliğinin belirlenmesinde kullanılması için küresel alt sınırlar bulmuşlardır. Başka bir çalışmada (Santos; Hunsucker; Deal, 1996), maksimum tamamlanma zamanını minimize etmeye çalışan esnek akış tipi çizelgeleme problemi için farklı sezgiselleri karşılaştırmışlardır.

Portmann; Vignier; Dardilhac vd. (1998), esnek akış tipi çizelgeleme problemleri için dal ve sınır algoritması ile genetik algoritmayı kullanarak alt sınırlarda iyileştirme sağlamışlardır. Pek çok durumda genetik algoritma optimal sonucun bulunmasına yardımcı olmuştur.

Negenman (2001), aşamalarında özdeş paralel makineler olan esnek akış tipi çizelgeleme problemi için maksimum tamamlanma zamanını minimize etmeyi

amaçlayan yerel arama prosedürlerini (benzetimli tavlama, tabu arama, değişken-derin arama ve literatürde kullanılmış bazı algoritmalar) karşılaştırmıştır.

Wang ve Li (2002), maksimum tamamlanma zamanını minimize etmeyi amaçlayan esnek akış tipi çizelgeleme problemine genetik algoritma ile bir çözüm getirmişlerdir.

Kurz ve Askin 2003 ve 2004 tarihli iki ayrı çalışmalarında, sıra bağımlı hazırlık zamanlı, aşama atlamalı esnek akış tipi çizelgeleme problemi için maksimum tamamlanma zamanını minimize etmeyi amaçlayan sezgisel yöntemleri karşılaştırmışlardır.

Engin and Döyen (2004), aşamalarında özdeş paralel makineler bulunan esnek akış tipi çizelgeleme problemi için maksimum tamamlanma zamanını minimize etmeyi amaçlayan bir yapay bağışıklık sistemi algoritması geliştirmişlerdir.

Logendran; Carson; Hanson (2005), esnek atölye tipi ortamda grup çizelgeleme ile ilgili bir çalışma yapmışlardır. İki aşamanın ilkinde aynı gruptaki işler arasında bir sıralama, sonra gruplar arasında bir sıralama yapmış, iki farklı senaryo denemişlerdir. Bir gruba ait işin bir aşamada belli bir makinede işlem görmesi durumunu *tekli hazırlık*, o aşamadaki diğer makinelerde de işlem görebilmesini *çoklu hazırlık* olarak tanımlamışlardır. Sonuçta iş sayısı arttıkça (problem çapı büyüdükçe) grup içindeki sıralama, gruplar arası sıralamadan daha önemli hale gelmektedir. Bunun yanında çoklu hazırlık durumu tekli hazırlıktan daha kötü değil ve hatta bazı durumlarda daha iyi bulunmuştur. Logendran; deSzoeke; Barnard (2006), sıra bağımlı hazırlık zamanlı esnek atölye tipi ortamda grup çizelgeleme için maksimum tamamlanma zamanını minimize etmeyi amaçlayan bir tabu arama algoritması geliştirmişlerdir. Küçük, orta ve büyük çaplı problemler için farklı algoritmalar, başlangıç çözümü için farklı mekanizmalar denemişlerdir. Maksimum tamamlanma zamanı ve hesaplama zamanı kriterlerine göre karşılaştırma yapmışlardır.

Oğuz ve Ercan (2005), aşamalarında özdeş paralel makineler bulunan esnek akış tipi çizelgeleme problemi için maksimum tamamlanma zamanını minimize etmeyi amaçlayan bir genetik algoritma geliştirmişlerdir. Jouglet; Oğuz; Sevaux (2009), aynı problem için genetik algoritma, genetik algoritmanın yerel aramasında kullanılan kısıt tabanlı bir dal ve sınır algoritması ve bir memetik algoritmayı karşılaştırmışlardır.

Zandieh; Fatemi Ghomi; Moattar Hosseini (2006), sıra bağımlı hazırlık zamanlı, aşamalarında özdeş paralel makineler olan esnek akış tipi çizelgeleme problemi için maksimum tamamlanma zamanını minimize etmeyi amaçlayan bir bağışıklık algoritması geliştirmişlerdir.

Paternina-Arboleda; Montoya-Torres; Acero Dominguez vd. (2008), k aşamalı, her aşamada belli sayıda özdeş paralel makine olan esnek akış tipi çizelgeleme problemine darboğaz temelli sezgisel bir yöntemle çözüm getirmişlerdir.

Tavakkoli-Moghaddam; Safaesi; Sassani (2009), ara stok olmayan, makine bloke etmeli esnek akış tipi çizelgeleme problemi için maksimum tamamlanma zamanını minimize etmeyi amaçlayan genetik algoritma ve değişken komşuluk aramanın birleştirildiği melez bir memetik algoritma geliştirmişlerdir.

Zandieh ve Gholami (2009), sıra bağımlı hazırlık zamanlı, makine arızaları ve bakımlardan dolayı stokastik ve deterministik şekilde makinelerin müsait olmama durumunun olduğu, aşamalarında özdeş paralel makineler olan esnek akış tipi çizelgeleme problemi için maksimum tamamlanma zamanını minimize etmeye çalışan bir bağışıklık algoritması geliştirmişlerdir.

Choong; Phon-Amnuaisuk; Alias (2011), aşamalarında özdeş paralel makineler bulunan esnek akış tipi çizelgeleme problemi için maksimum tamamlanma zamanını minimize etmeyi amaçlayan parçacık sürü optimizasyonu, benzetimli tavlama ve tabu arama melezi bir metasezgisel algoritma geliştirmişlerdir.

Hidri ve Haouari (2011), aşamalarında paralel makineler bulunan esnek akış tipi çizelgeleme probleminin maksimum tamamlanma zamanı için alt sınırlar belirlemişler ve iki aşamalı bir sezgisel yöntem geliştirmişlerdir.

Liao; Tjandradjaja; Chung (2012), aşamalarında özdeş paralel makineler olan esnek akış tipi çizelgeleme problemi için maksimum tamamlanma zamanını minimize etmeyi amaçlayan bir parçacık sürü optimizasyonu ve darboğaz sezgiseli melezi bir algoritma geliştirmişlerdir. Yerel optimumlardan kaçınmak için de benzetimli tavlamaya dayalı bir mekanizma kullanmışlardır.

Aşamalarında özdeş paralel makineler olan, fakat maksimum tamamlanma zamanı dışında başka bir amacı (da) minimize etmeye çalışan çalışmalar aşağıda verilmiştir.

Kochhar ve Morris (1987), ortalama akış süresini (işlerin sistem içinde geçirdiği sürenin ortalaması) minimize etmeyi amaçladıkları, sınırlı ara stok alanından dolayı makine bloke etmenin, makine arızalarının ve hazırlık sürelerinin olduğu bir esnek akış tipi çizelgeleme problemi için bir sezgisel yöntem geliştirmişlerdir.

Rajendran ve Chaudhuri (1992a), aşamalarında özdeş makineler olan esnek akış tipi çizelgeleme problemi için toplam akış zamanını minimize etmeye çalışan dal ve sınır algoritması ile optimale yakın sonuç veren bir sezgisel yöntem geliştirmişlerdir.

Luss ve Rosenwein (1993), esnek akış tipi bir ortam için teslim tarihi belirleme algoritması geliştirmişlerdir. Buna göre oluşturulan çizelge ilk geleni ilk atama yapan sezgisele göre toplam ağırlıklı gecikme kriterinde önemli bir gelişme sağlamıştır.

Chang ve Liao (1994), erken tamamlanma ve gecikmeden kaynaklanan cezaları ve ara stokları minimize etmeyi amaçlayan esnek akış tipi çizelgeleme problemi için optimale yakın sonuçlar veren Lagrange gevşetmesi temelli bir sezgisel yöntem geliştirmişlerdir.

Hunsucker ve Shah (1994), aşamalarında özdeş paralel makineler olan, sistem içindeki iş sayısının belli bir sayıyla sınırlı olduğu ve tamamlanma zamanı, ortalama ve maksimum akış zamanını minimize etmeyi amaçlayan esnek akış tipi çizelgeleme problemi için farklı öncelik/sıralama kurallarını simulasyonla karşılaştırmışlardır.

Guinet ve Solomon (1996), maksimum gecikmeyi ve maksimum tamamlanma zamanını minimize etme amaçlı esnek akış tipi çizelgeleme problemi için çeşitli sezgisel yöntemler geliştirmişlerdir. Bu sezgisel modelleri ürettikleri 10, 20, 30 iş, 3 ve 5 aşama ve aşamalarda 2 ve 5 makine olan test problemlerinde karşılaştırmışlardır.

Vignier; Dardilhac; Dezalay vd. (1996), toplam tamamlanma zamanını minimize etme amaçlı esnek akış tipi çizelgeleme problemi için dal ve sınır algoritması temelli bir çözüm geliştirmişlerdir.

Agnetis; Pacifici; Rossi et al. (1997), bir otomobil montaj fabrikasında uygulaması olan, geciken iş sayısı ve ara stoğu minimize etmeyi amaçlayan esnek akış tipi çizelgeleme problemine sezgisel bir yöntemle çözüm getirmişlerdir.

Yang (1998), esnek akış tipi çizelgeleme problemleri için toplam ağırlıklı gecikmeyi minimize etmeye çalışan üç ayrı sezgiseli karşılaştırmıştır. Bunların yanında küçük çaplı problemler için bir de dal ve sınır algoritması geliştirmiştir, bu yöntem 5 aşamalı, 10 işli problemleri makul sürelerde çözebilmektedir. Bunların haricinde modele makine uygunluk kısıtları eklenmiştir.

Brah ve Loo (1999) iş sayısı, aşama sayısı, aşamalardaki makine sayısı gibi problem karakteristiklerinin esnek akış tipi çizelgeleme problemlerinin çözümünde kullanılan beş farklı sezgiselin performansına etkilerini regresyonla incelemişlerdir. Problem karakteristikleri, performanstaki değişimin çoğunu açıklasa da sezgisel yöntemin de etkisi olduğu ifade edilmiştir. NEH sezgiselinin (Nawaz; Enscore; Ham, 1983) daha tutarlı sonuçlar verdiğini ifade etmektedirler.

Botta-Genoulaz (2000) öncelik kısıtları ve sıralamadan bağımsız hazırlık zamanları olan, maksimum gecikmeyi minimize etmeyi amaçlayan esnek akış tipi çizelgeleme problemi için farklı sezgisel yöntemleri karşılaştırmıştır.

Liu ve Chang (2000), sıra bağımlı hazırlık zamanlı esnek akış tipi çizelgeleme problemine Lagrange gevşetmesi temelli bir sezgiselle çözüm getirmişlerdir. Amaç erken tamamlanma ve gecikmeye ceza vererek müşterilere teslim tarihini karşılamak, ara stokları ve makine hazırlık maliyetlerini azaltmaktır.

Yang; Kreipl; Pinedo (2000), aşamalarında özdeş paralel makineler bulunan esnek akış tipi çizelgeleme problemi için toplam ağırlıklı gecikmeyi minimize etmeyi amaçlayan üç sezgisel yöntem geliştirmiş ve bunları karşılaştırmışlardır. Birisi problemin aşamalarını ayrı ayrı tek aşamalı alt problemlere ayıran ayrışma algoritması (decomposition), diğeri yerel aramaya dayalı bir algoritma, üçüncüsü ise bu ikisinin melezidir.

Azizoğlu; Çakmak; Kondakci (2001) aşamalarında özdeş paralel makineler bulunan esnek akış tipi çizelgeleme problemi için toplam akış zamanını minimize etmeyi

amaçlayan sezgisel yöntemlerde başlangıç çözümü oluşturan bir dal ve sınır algoritması geliştirmişlerdir. Yöntem orta büyüklükte problemler için makul zamanlarda optimal çözüm elde edebilmektedir. Optimal çözümü bulması kısa sürmekte fakat zamanın çoğunu çözümün optimal olduğunu göstermek için harcamaktadır.

Cheng; Karuno; Kise (2001) maksimum gecikmeyi minimize etmeyi amaçlayan *m* aşamalı özdeş makineli esnek akış tipi çizelgeleme problemi için kayan darboğaz prosedürüne dayalı bir sezgisel yöntem geliştirmişlerdir. Yöntem, problemi *m* adet paralel makine problemine ayırıp her aşamayı ayrı ayrı çözmektedir. Test problemleri üzerinde diğer bazı sezgisel yöntemlerle yaptıkları karşılaştırmada geliştirdikleri sezgiselin kısa sürede optimale yakın sonuçlar verdiğini ifade etmektedirler.

Kyparisis ve Koulamas (2001), aşamalarında özdeş paralel makineler olan esnek akış tipi çizelgeleme problemi için toplam ağırlıklı tamamlanma zamanını minimize etmeyi amaçlayan bir sezgisel yöntem geliştirmişler ve en kötü durum performans sınırları belirlemişlerdir.

Gupta; Krüger; Lauff et al. (2002), aşamalarında paralel özdeş makineler olan, atanabilir teslim tarihleri, alt ve üst değerler arasında değişken işlem süreleri olan permutasyon esnek akış tipi çizelgeleme problemi için erken tamamlanma ve gecikme cezaları, ağırlıklı tamamlanma zamanları ve teslim tarihi atama maliyetlerinin toplamını minimize etmeyi amaçlayan sezgisel yöntemler geliştirmişlerdir.

Alisantoso; Khoo; Jiang (2003), baskı devre üreten bir tesiste uygulaması olan aşamalarında özdeş paralel makineler bulunan 4 aşamalı esnek akış tipi çizelgeleme problemi için ortalama gecikmeyi minimize etmeyi amaçlayan bir bağışıklık algoritması geliştirmişlerdir.

Akrami; Karimi; Hosseini (2006), sınırlı ara stoklu ve makine bloke etmeli esnek akış tipi çizelgeleme problemi için hazırlık, ara stok ve bitmiş ürün stok maliyetlerini minimize etmeyi amaçlayan karma tamsayılı doğrusal olmayan programlama modeli geliştirmişlerdir. Fakat problemin karmaşıklığından dolayı genetik algoritma ve tabu arama yöntemlerine dayalı çözümler de geliştirmişlerdir.

Janiak; Kozan; Lichtenstein et al. (2007), aşamalarında özdeş paralel makineler olan, işlerin hazır olma zamanları ve teslim tarihleri belli olan bir esnek akış tipi çizelgeleme problemi için toplam ağırlıklı erken tamamlanma, toplam ağırlıklı gecikme ve toplam ağırlıklı bekleme sürelerini minimize etmeyi amaçlayan sezgisel ve metasezgisel (tabu arama ve benzetimli tavlama) yöntemler geliştirmişlerdir.

Shiau; Cheng; Huang (2008), oransal esnek akış tipi çizelgeleme (proportionate flexible flowshop) olarak adlandırdıkları (bir işin her aşamadaki işlem süresinin aynı olması durumu) problem için toplam ağırlıklı tamamlanma zamanını minimize etmeyi amaçlayan genetik algoritma ve tabu arama melezi bir yöntem geliştirmişlerdir.

Karimi; Zandieh; Kar-Amooz (2009), sıra bağımlı hazırlık zamanlı esnek akış tipi çizelgeleme problemi için maksimum tamamlanma zamanı ve toplam ağırlıklı gecikmeyi eşzamanlı olarak minimize etmeyi amaçlayan bir genetik algoritma geliştirmişlerdir.

Naderi; Zandieh; Shirazi (2009), sıra bağımlı hazırlık zamanları ve taşıma zamanları olan esnek akış tipi çizelgeleme problemi için toplam ağırlıklı gecikmeyi minimize etmeyi amaçlayan karma tamsayılı doğrusal programlama (küçük çaplı problemler için) ve bir metasezgisel algoritma (elektromanyetizma algoritması) geliştirmişlerdir.

Khalouli; Ghedjati; Hamzaoui (2010), aşamalarında özdeş paralel makineler olan esnek akış tipi çizelgeleme problemi için erken tamamlanma ve geç bitirme cezalarını minimize etmeyi amaçlayan bir karınca kolonisi optimizasyonu algoritması geliştirmişlerdir.

Behnamian ve Zandieh (2011), aşamalarında özdeş paralel makineler olan sıra bağımlı hazırlık zamanlı ve sınırlı bekleme süreli esnek akış tipi çizelgeleme problemi için erken tamamlanma ve karesel geç bitirme cezaları toplamını minimize etmeyi amaçlayan bir karma tamsayılı doğrusal programlama modeli ve kesikli koloni rekabetçi algoritma (discrete colonial competitive algorithm) geliştirmişlerdir.

Niu; Zhou; Fei vd. (2012) aşamalarında özdeş paralel makineler bulunan esnek akış tipi çizelgeleme problemi için ortalama akış zamanını minimize etmeyi amaçlayan

bağışıklık algoritması ve kuantum algoritmasının geliştirilmiş şekilde melezlenmesiyle elde edilen bir metasezgisel algoritma geliştirmişlerdir.

2.2.3.3.2 Farklı Hızda Paralel Makineli Çalışmalar

Aşamalarında farklı hızda paralel makineler olan esnek akış tipi çizelgeleme problemi çalışmaları aşağıda verilmiştir.

Verma ve Dessouky (1999), diğer problemlerden farklı olarak işlerin özdeş olduğu, aşamalarında farklı hızda paralel makinelerin bulunduğu maksimum tamamlanma zamanını minimize etmeye çalışan esnek akış tipi çizelgeleme problemi için sıralama kuralları ile sezgisel yöntem geliştirmişlerdir. Üç farklı sezgisel, farklı problem koşullarında öne çıkmaktadır, bu yüzden üçünün karışımı bir sezgisel geliştirmişlerdir.

Crowder (2006), aşamalarında farklı hızlarda paralel makineler bulunan ve aşamalar arasında sınırlı ara stok olan makine bloke etmeli ve sıra bağımlı hazırlık zamanlı maksimum tamamlanma zamanını minimize etmeyi amaçlayan esnek akış tipi çizelgeleme problemi için başlangıç çözümü oluşturan iki sezgisel geliştirmiş, devamında başlangıç çözümünü geliştirmek için benzetimli tavlama yöntemini kullanmıştır.

Kyparisis ve Koulamas (2006a), aşamalarında farklı hızda paralel makineler bulunan esnek akış tipi çizelgeleme problemi için maksimum tamamlanma zamanını minimize etmeyi amaçlayan sezgisel yöntemler geliştirmişlerdir.

Amin-Naseri ve Beheshti-Nia (2009), aşamalarında farklı hızda paralel makinelerin olduğu ve yığın işleme yapabilen (bir makinede eşzamanlı olarak birden fazla işin işlenebilmesi) bir esnek akış tipi çizelgeleme problemi için maksimum tamamlanma zamanını minimize etmeyi amaçlayan karma tamsayılı programlama, sezgisel ve metasezgisel yöntemler geliştirmişlerdir.

2.2.3.3.3 İlişkisiz Paralel Makineli Çalışmalar

Aşamalarında ilişkisiz paralel makineler olan esnek akış tipi çizelgeleme problemi çalışmaları aşağıda verilmiştir.

Ruiz ve Maroto (2006), seramik üretiminde uygulaması olan maksimum tamamlanma zamanını minimize etmeyi amaçlayan, aşamalarında ilişkisiz paralel makineler bulunan, sıra bağımlı hazırlık zamanlı esnek akış tipi çizelgeleme problemine genetik algoritma ile çözüm getirmişlerdir.

Jenabi; Fatemi Ghomi; Torabi et al. (2007), aşamalarında ilişkisiz paralel makineler olan esnek akış tipi çizelgeleme problemi için toplam hazırlık ve stoksuz kalmayacak şekilde stok maliyetini minimize edecek bir karma 0-1 lineer olmayan matematiksel model, büyük çaplı problemler için de melez bir genetik algoritma ve benzetimli tavlama algoritması geliştirmişlerdir. Çözüm kalitesi açısından melez genetik algoritma daha iyi iken hesaplama zamanı açısından benzetimli tavlama daha iyi olarak bulunmuştur.

Ruiz; Şerifoğlu; Urlings (2008) gerçek hayat problemine uygun olacak şekilde esnek akış tipi problemine sıra bağımlı hazırlık zamanları, aşama atlama, makine uygunluk ve öncelik kısıtları, aşamalarda ilişkisiz paralel makineler özelliklerini eklemişlerdir. Bu problem için maksimum tamamlanma zamanını minimize etmeyi amaçlayan bir karma tamsayılı programlama modeli geliştirmiş ve sezgisel yöntemleri karşılaştırmışlardır.

Jungwattanakit; Reodecha; Chaovalitwongse et al. (2009), aşamalarında ilişkisiz paralel makineler olan sıra bağımlı hazırlık zamanlı esnek akış tipi çizelgeleme problemi için maksimum tamamlanma zamanı ve geciken iş sayısını minimize etmeyi amaçlayan karma tamsayılı programlama, sezgisel ve metasezgisel yöntemler (benzetimli tavlama, tabu arama ve genetik algoritma) geliştirmişlerdir.

Urling; Ruiz; Stützle (2010), aşamalarında ilişkisiz paralel makineler bulunan, öncelik kısıtlamaları, sıra bağımlı hazırlık zamanları ve makine uygunluk kısıtları olan esnek akış tipi çizelgeleme problemi için tamamlanma zamanını minimize etmeyi amaçlayan bir metasezgisel algoritma geliştirmişlerdir.

Behnamian ve Fatemi Ghomi (2011), aşamalarında ilişkisiz paralel makineler bulunan sıra bağımlı hazırlık zamanlı esnek akış tipi çizelgeleme problemi için tamamlanma zamanı ve toplam kaynak tahsis maliyetini minimize etmeyi amaçlayan (çok amaçlı) genetik algoritma ve değişken komşuluk arama melezi bir metasezgisel algoritma geliştirmişlerdir.

Yukarıda bahsedilen çalışmalar dışında aşamalarındaki paralel makinelerin özdeş, farklı hızda veya ilişkisiz olma durumu net olarak belirtilmemiş çalışmalar aşağıda verilmiştir.

Randhawa; Juwono; Burhanuddin (1994), bir gıda işletmesinde esnek akış tipi çizelgeleme problemi için çeşitli sezgisel yöntemleri maksimum tamamlanma zamanı, makine kullanım oranı ve gecikme gibi belli performans kriterlerine göre karşılaştırmışlardır. Aşamalarındaki paralel makinelerin özdeş olmadığı belirtilmiş fakat farklı hızda mı yoksa ilişkisiz mi olduğu net olarak belirtilmemiştir.

Yu; Ohsato; Kawakami vd. (1999), esnek akış tipi çizelgeleme problemine CORBA (Common Object Request Broker Architecture) olarak adlandırdıkları, iletişim ve çizelgeleme ajanları bulunan bir yöntemle çözüm getirmişlerdir.

Quadt ve Kuhn (2007a), esnek akış tipi çizelgeleme problemi için hazırlık maliyetlerini ve ortalama akış zamanını minimize etmeyi amaçlayan bir metasezgisel (genetik algoritma) geliştirmişlerdir. Tüm aşamalarda tüm işler aynı işlem süresine sahiptir. 100 işten oluşan bir problemin bir dakika gibi kısa bir sürede çizelgelenebildiğinden gerçek hayat uygulamalarında kullanılabileceği ifade edilmiştir.

Behnamian; Fatemi Ghomi; Zandieh (2009), sıra bağımlı hazırlık zamanlı esnek akış tipi çizelgeleme problemi için maksimum tamamlanma zamanı ve erken tamamlanma ve gecikmelerin toplamını minimize etmeyi amaçlayan benzetimli tavlama ve değişken komşuluk arama melezi bir metasezgisel algoritma geliştirmişlerdir.

3. BÖLÜM

3 UYGULAMA

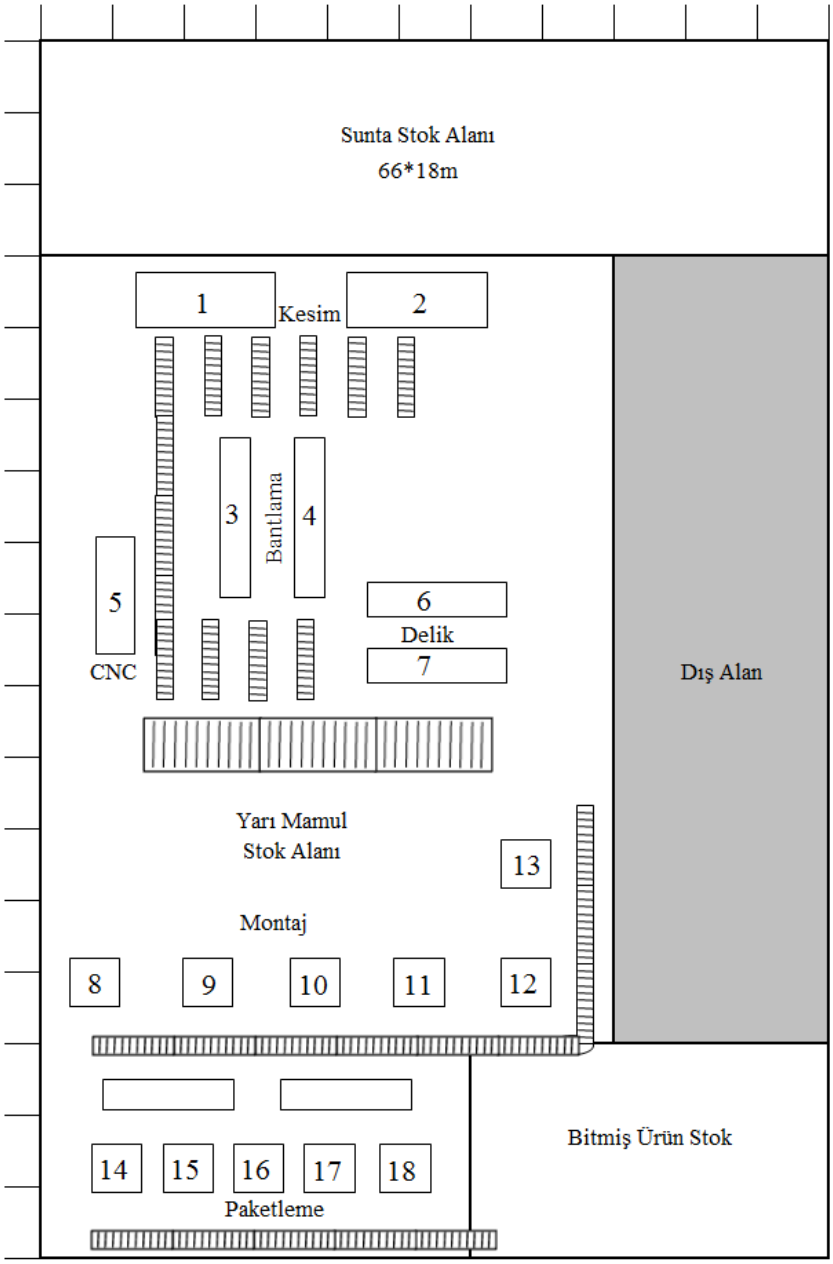
3.1 İşletmenin ve Üretim Sürecinin Tanıtımı

Tezin uygulaması, ofis mobilyaları üreten bir imalat işletmesinde yapılmıştır. İşletme Amasya'nın Merzifon ilçesinde Organize Sanayi Bölgesinde faaliyet göstermektedir. 10.000 m^2'si kapalı olmak üzere toplam 20.000 m^2'lik alana kurulu tesis, 1985 yılından beri faaliyettedir. Toplam 120 personeli bulunan işletme, yurt içine ve dışına mobilya satışı gerçekleştirmektedir. Yurt içinde kendi mağazaları ve bayileri aracılığı ile müşterilerine ulaşmaktadır. Devlet Malzeme Ofisi de yurt içindeki en önemli müşterilerindendir. Yurt dışında ise Birleşik Arap Emirlikleri, Libya, Hindistan ve Ürdün gibi ülkelere ürünlerini ihraç etmektedir.

İşletmenin masa, dolap, sehpa, komodin, etejer gibi belli ana ürün grupları altında çok çeşitli ürünleri bulunmaktadır. Her ürün grubunun farklı modelleri ve bu modellerin ebat ve renk kombinasyonları dikkate alındığında, standart dışı özel siparişle üretilen ürünlerle birlikte 1000'den fazla ürün çeşidi vardır.

İşletmede ürün çeşidi fazla olduğundan, belli bir anda sistem içinde çok çeşitli ürünün üretimi devam etmekte, makinelerin önünde farklı işlere ait parçalar sıra bekleyebilmektedir. Bunun yanında çoğunlukla siparişe yönelik çalışan bir işletme olduğundan işlerin teslim tarihleri vardır ve zamanında yetişmesi gerekmektedir. Mevcut durumda işlerin hangi sıra ile işleneceğine ustabaşı deneyimlerine dayanarak karar vermektedir. İşletmede çizelgeleme uygulaması ile hazırlık sürelerinde azalma, işlerin makine önünde bekleme sürelerinde azalma, tamamlanma zamanında düşme ve teslimat tarihlerinde iyileşme olacağı öngörülmektedir.

Fabrika'nın yerleşimi Şekil 18'de görüldüğü gibidir.



Şekil 18: Fabrika Yerleşimi

Ürünlerin temel hammaddesi suntadır. Suntalar 3660 mm. x 1830 mm. standart ebatlardaki plakalar halindedir. İşletme ürünlerinde 30 mm., 18 mm. ve 8 mm. olmak üzere üç ayrı kalınlıkta sunta kullanmaktadır. İstisnaları olmakla birlikte 30 mm.lik suntalar çoğunlukla masa, toplantı masası, banko gibi ürünlerde, 18 mm.lik suntalar sehpa, dolap, komodin gibi daha küçük ürünlerde, 8 mm.lik suntalar ise ürünlerin altlık, arkalık gibi görünmeyen kısımlarında kullanılmaktadır.

İşletme temel hammaddesi olan suntalarını iki ayrı tedarikçiden sağlamaktadır. Bunlardan birisi Kastamonu'da, diğeri ise Isparta'dadır. Suntalar bir aracı dolduracak

miktarda istenmektedir. Tedarikçide eksiksiz bulunması halinde, istenilen renk ve kalınlık kombinasyonundaki suntaların tedarik süresi 1 gündür, fakat ellerinde bulunmuyorsa ya aracı dolduracak şekilde başka renk ve kalınlık kombinasyonlarından göndermekte ya da o renklerden üretene kadar tedariği geciktirmektedirler. İşletme, suntaların tedariği ile ilgili bu sıkıntılardan dolayı farklı kalınlık ve renk kombinasyonlarından ortalama 3000 plakalık sunta stoğu bulundurmaktadır. Suntaların sunta stok alanından kesim makinelerine taşınmasında ve makineler arasında taşınmasında forklift veya özel tasarlanmış taşıma araçları kullanılmaktadır.

Genel olarak ürünler sunta stok alanından bitmiş ürün stok alanına doğru, doğrusal bir akış izlemektedirler. Ürünlerin hepsi kesim, montaj ve paketleme işlemlerini görmektedir. Kesim ve montaj arasındaki işlemlerde ise (bantlama, CNC ve delik) yine doğrusal bir akış görülmekte, fakat ürünlerden bazıları, bazı işlemleri atlamaktadır.

Kesme işlemi 1 ve 2 numaralı otomatik yüklemeli büyük makinelerde yapılmaktadır. Aynı zamanda ebatlama makineleri olarak da adlandırılan bu makineler özdeş makinelerdir ve bu aşamada makineler arasında bir paralellik söz konusudur. Yani kesme işlemi görecek bir ürün 1 veya 2 numaraları makinelerden herhangi birini seçebilir. Bu makinelerde 3'er kişi çalışmaktadır. Genel olarak bu çalışanlardan birisi (operatör) bilgisayara ölçüleri ve adetleri girmekte, diğer iki kişi ise yükleme ve boşaltma işlemlerini gerçekleştirmektedir. Üretilecek ürünün adedine göre ve makinenin yükleme ve kesme kapasitesine göre suntalar üst üste koyulup kestirilebilmektedir. Makinenin kesme kalınlık kapasitesi 80 mm.dir. Bu kapasiteye göre; hepsi aynı ebatlarda kesilmek şartıyla; 30 mm.lik sunta plakalarından 2 tane, 18 mm.lik sunta plakalarından 4 tane, 8 mm.lik sunta plakalarından ise 10 tane üst üste konulup kesilebilmektedir. Makinelere bilgisayar vasıtasıyla üretilecek parçaların ebatları ve adetleri girilmekte, buna göre de bilgisayar fireyi minimize edecek şekilde bir kesim şablonu belirlemektedir. Teorik olarak bir veya birden fazla sunta plakasından farklı ebatlarda parçalar kestirmek mümkündür. Bilgisayar buna göre bir kesim şablonu çıkarabilmektedir; fakat uygulamada bu ciddi zaman kayıplarına neden olduğundan tercih edilmemekte, bir sunta plakasından sadece aynı ebatlarda parçaların kesiminin yapılması tercih edilmektedir. Bu durum her üründen belli bir minimum miktarda üretilmesini zorunlu hale getirse de, fabrikada parti üretimi mantığı ile hareket edildiğinden, sipariş fazlası üretim bitmiş ürün stoğuna kaldırılmaktadır. Özel veya çok

düşük miktarlı bir sipariş gelmesi durumunda bu ürün için kesme makineleri meşgul edilmemekte; otomatik olmayan yatar daire kesme makineleri kullanılmaktadır.

Bantlama işlemi 3 ve 4 numaralı makinelerde yapılmaktadır. Bu makineler aslen özdeş olsalar da farklı kalınlıkta bant gerektiren ürünlerin bantlanmasında kullanılmaktadır. Bantlama makinesinde bant değişimleri hazırlık süresine neden olmaktadır. Bandın eninde ve kalınlığında bir değişiklik olmadan sadece rengi değişecekse hazırlık süresi çok kısadır. Eğer bandın eninin değişmesi gerekiyorsa, yani 30 mm.lik bir suntadan sonra 18 mm.lik bir sunta (ya da tam tersi) bantlama işlemi görmesi gerekiyorsa daha uzun bir hazırlık süresi (~30 sn.) gerekmektedir. Bu sürede hem bant değişmekte, hem de makinede gerekli ayar yapılmaktadır. Bant kalınlığı değişimi, yani 2 mm. kalınlığında bant kullanan bir suntadan sonra 0,4 mm. kalınlığında bant kullanan bir suntanın işlem görmesinin gerektiği durumda çok daha fazla bir hazırlık süresine (~7,5 dk.) ihtiyaç olduğundan ve makinedeki bant kalınlığı ile ilgili ayar değişikliği bantlama mekanizmalarının ömrünü azalttığından, bu değişikliğe gidilmek istenmemektedir. Bu yüzden bantlama makinelerinden birisi sürekli 0,4 mm kalınlığında bantlama yapmaya, diğeri ise sürekli 2 mm. kalınlığında bantlama yapmaya ayarlıdır. 30 mm. kalınlığındaki suntaların hepsi 2 mm. kalınlığında bant gerektirmektedir. 18 mm. kalınlığındaki suntaların büyük çoğunluğu 0,4 mm. kalınlığında bant gerektirmekte, fakat ürünün görünen kısımlarında kullanılan parçalarda 2 mm. bant kullanılmaktadır. 8 mm. kalınlığındaki suntalar ise bantlama işlemi görmemektedir. Bu durumda 0,4 mm. kalınlığında bantlama yapmaya ayarlı makinede sadece 18 mm. kalınlığındaki suntalar işlem görmekte ve bu makinede bant eni ile ilgili herhangi bir hazırlığa ihtiyaç duyulmamaktadır. 2 mm. kalınlığında bantlama yapmaya ayarlı makinede ise hem 30 mm. hem de 18 mm. kalınlığındaki suntalar işlem görebilmekte ve bu iki sunta arasında geçiş olduğu durumlarda kısa hazırlık süresi gerekmektedir.

5 numaralı CNC makinesinde, eğrisel işlem görmesi, kanal açılması ve standart dışı delik açılması gereken ürünlerin parçaları işlem görmektedir. CNC makinesi aynı zamanda bantlama da yapabildiği için burada işlem gören parçalar kenar bandı gerektirse dahi ayrıca bantlama makinesine girmelerine gerek kalmamaktadır.

Delik işlemi 6 ve 7 numaralı çoklu delik makinelerinde yapılmaktadır. Bu makineler özdeştir, bu aşamada paralellik söz konusudur. Yani delik işlemi görecek bir iş 6 veya 7

numaralı delik makinelerinden herhangi birini seçebilir. Aynı parça üzerinde çok sayıda delik açma veya yüksek miktarlı partilerde birbiri ardına aynı delme işleminin gerektiği durumlarda bu makineler kullanılmaktadır. Parça değiştiğinde delik açılması gereken yerler de farklı olacağından bir hazırlık süresi gerekmektedir. Ufak delik matkaplarının konumu iki yönde ayarlanabilmekte, bu sayede yeni parçanın deliklerinin konumuna göre matkapların yeri ayarlanabilmektedir.

8'den 13'e kadar numaralarla gösterilen montaj tezgahlarında ürünlerin parçaları bir araya getirilip paketlemeye sevk edilmektedir. Komodinler hariç diğer bütün ürünler monte edilmeden (demonte) paketlenmektedir. Dolayısıyla montaj tezgahlarında gerçekte montaj yapılmamakta, sadece ürünlerin parçaları bir araya getirilip, minifiks, kulp, menteşe gibi malzemeler eklenmektedir. Bir araya getirilen parçalar üst üste konmakta ve raylı bir sistem ile paketleme bölümüne sevk edilmektedir. Bütün montaj tezgahları birbirine paralel durumdadır, yani her tezgahta her ürünün montajı yapılabilmektedir. Dolayısıyla montaj için bekleyen bir parti, tezgahlardan herhangi birisini tercih edebilir.

Paketleme bölümünde ürünün kalite kontrolü ve temizliği yapılmakta (yapıştırıcı ve başka lekeler tiner ile silinmekte), hırdavat malzemeleri eklenmekte ve ürünler tartılmaktadır. Sonrasında paketin içinde oynamaması için şrink (shrink) adı verilen film ile sarılıp, komodinler hariç demonte şekilde paketlenmektedir.

3.2 Çizelgelenecek İşler

Fabrikada ürün çeşidi çok fazla olduğundan ve olağan bir çizelgeleme döneminde bu ürün çeşitlerinin tamamı üretilmediğinden çizelgeleme problemi için kullanılacak ürün sayısı sınırlandırılmıştır. Bunun için de en çok talep gören ve sık üretilen 15 ürün çizelgeleme problemi için kullanılmıştır. Bu ürünlerin en çok satılan, dolayısıyla üretilen ebatları seçilmiştir. Renk farklılıkları dikkate alınmamış; tek renk olduğu varsayılmıştır. Bu ürünler Tablo 6'da görülmektedir.

Her ürün belli sayıda bileşenden oluşmaktadır. Her bileşen ayrı bir sunta plakasından kesilerek elde edilmektedir. Her bileşen için bir sunta plakasından kaç adet çıkacağı bellidir. Ürünler için minimum üretim adetleri de, sunta plakalarından çıkan bileşenlerin

etkin kullanılacağı şekilde zaman içinde işletmede oluşturulmuş ve Tablo 6'da belirtilen adetlerdir.

Tablo 6: Çizelgeleme İçin Seçilen Ürünler, Türleri ve Adetleri

Sıra	Ürün Adı	Türü	Adet
1	A	Komodin	60
2	B	Sehpa	40
3	C	Yan Kol	24
4	D	Komodin	60
5	E	Masa	40
6	F	Etejer	40
7	G	Komodin	60
8	H	Sehpa	40
9	I	PC Masası	60
10	J	Masa	24
11	K	Masa	30
12	L	Sehpa	30
13	M	Etejer	30
14	N	Masa	30
15	O	Dolap	20

Her ürünün bileşeni partiler halinde, fakat birbirinden bağımsız şekilde üretim süreci içinde farklı rotalarda hareket edebilmektedir. Dolayısıyla, üretim süreci içinde montaj aşamasına kadar ürünler yerine ürünlerin bileşenleri izlenmeli ve çizelgelenmelidir. Bu bileşenler aynı ürüne ait olmalarına rağmen, gerektiğinde eşzamanlı olarak farklı makinelerde işlem görebilmektedir. Fakat bir ürünün montajına başlanabilmesi için o ürünün bileşenlerinin montaja kadar olan bütün işlemlerinin bitmiş olması gerekmektedir.

Ürün bileşenleri ve özellikleri Tablo 7'de verilmiştir. Örneğin A1, A model komodinin yan parçasıdır, her bir A ürünü için 2 adet A1 bileşeninden gerekmektedir. A1 bileşeni 18 mm. kalınlığında suntadan kesilmektedir ve ebatları 64 cm. $*$ 48 cm.dir.

Tablo 7: Ürünler, Ürün Bileşenleri ve Özellikleri

Ürün		Parçalar		Adet	Sunta Kalınlığı	Ölçü (cm.)
A	1	A1	Yan	2	18mm	64*48
	2	A2	Üst	1	18mm	48*37
	3	A3	Alt	1	18mm	48*40
	4	A4	Arkalık	1	8mm	62*39
	5	A5	Alınlık	3	18mm	21*40
	6	A6	Çek.Altlık	3	8mm	40*34
B	7	B1	Üst	1	30mm	50*50
	8	B2	Ayak	2	30mm	35*40
C	9	C1	Tabla	1	30mm	80*50
D	10	D1	Üst	1	30mm	41*45
	11	D2	Yan	2	18mm	48*43
	12	D3	Arkalık	1	8mm	48*39
	13	D4	Alınlık	3	18mm	16*40

	14	D5	Çek.Altlık	3	8mm	40*34
	15	D6	Alt	1	18mm	43*40
Ürün		Parçalar		Adet	Sunta Kalınlığı	Ölçü (cm.)
E	16	E1	Üst	1	30mm	180*90
	17	E2	Ayak	2	30mm	60*80
	18	E3	Perde	1	18mm	164*25
	19	E4	Dikme ve çıta	3	18mm	164*12
F	20	F1	Üst	1	30mm	90*55
	21	F2	Ayak	2	30mm	60*45
	22	F3	Perde	1	18mm	75*30
	23	F4	Klavye	1	18mm	73*40
	24	F5	Dikme	2	30mm	12*45
G	25	G1	Üst	1	30mm	41*45
	26	G2	Yan	2	18mm	48*43
	27	G3	Arkalık	1	8mm	48*39
	28	G4	Alınlık	3	18mm	16*40
	29	G5	Çek.Altlık	3	8mm	40*34
	30	G6	Alt	1	18mm	43*40
H	31	H1	Üst	1	30mm	60*50
	32	H2	Ayak	2	30mm	30*40
I	33	I1	Üst	1	18mm	90*60
	34	I2	Ayak	2	18mm	71*50
	35	I3	Orta Tabla	1	18mm	71*50
	36	I4	Alt Tabla	1	18mm	27*50
	37	I5	Klavye	1	18mm	54*35
	38	I6	Perde-1	2	18mm	57*35
J	39	J1	Üst	1	30mm	160*60
	40	J2	Fon	1	30mm	161*25
	41	J3	Ayak	2	30mm	70*60
	42	J4	Perde	1	18mm	145*40
K	43	K1	Üst	1	18mm	201*91
	44	K2	Üst Altı	1	18mm	201*91
	45	K3	Ayak	2	18mm	70*60
	46	K4	Perde	1	18mm	153*45
	47	K5	Çıta	4	18mm	70*15
L	48	L1	Üst	1	18mm	61*51
	49	L2	Üst Altı	1	18mm	61*51
	50	L3	Ayak	2	18mm	40*35
	51	L4	Orta Tabla	1	18mm	47*35
M	52	M1	Üst	1	18mm	100*60
	53	M2	Üst Altı	1	18mm	100*60
	54	M3	Ayak	2	18mm	70*45
	55	M4	Perde	1	18mm	73*45
	56	M5	Çıta	4	18mm	70*15
	57	M6	Klavye	1	18mm	71*35
N	58	N1	Üst	1	30mm	160*80
	59	N2	Ayak	2	30mm	70*60
	60	N3	Perde	1	18mm	145*40
O	61	O1	Üst	1	30mm	80*38
	62	O2	Alt Tabla	1	18mm	80*35
	63	O3	Raf	4	18mm	77*33
	64	O4	Kapak	2	18mm	72*40
	65	O5	Yan	2	18mm	175*35
	66	O6	Arkalık	1	8mm	177*79

Ürün bileşenlerinin rotaları Tablo 8'de verilmiştir. Bir ürünü oluşturmak için gereken bileşenlerin tamamı Kesim, Montaj ve Paket işlemlerini görmektedir. Bantlama, CNC

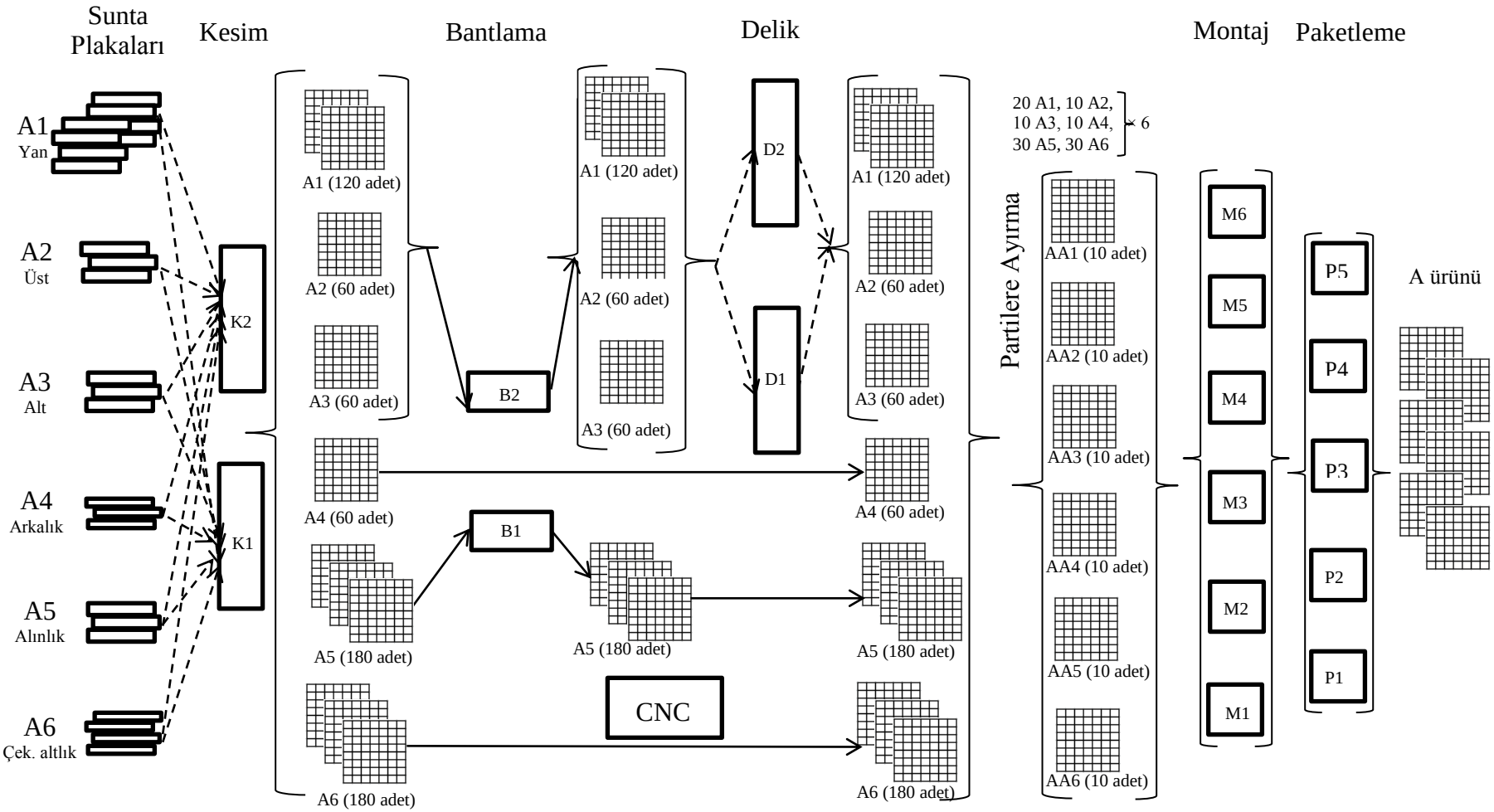
ve Delik işlemlerini ise bazı ürünler atlamaktadır. Fakat montajın gerçekleşebilmesi için montaj aşaması öncesinde ürünün bütün bileşenlerinin montaj öncesindeki işlemlerinin bitmiş olması gerekmektedir. Bütün işlemleri bitmiş bir ürüne ait bileşenler montaja gitmeden önce belli büyüklükte partiler haline getirilip montaj ve paketlemeye partiler halinde gitmektedirler.

Tablo 8: Ürün Bileşenlerinin Rotaları

	Parçalar	Rota	Ürün	Rota
1	A1	Kesim-Bantlama-Delik	AA	Montaj-Paket
2	A2	Kesim-Bantlama-Delik		
3	A3	Kesim-Bantlama-Delik		
4	A4	Kesim		
5	A5	Kesim-Bantlama		
6	A6	Kesim		
7	B1	Kesim-Bantlama-Delik	BB	Montaj-Paket
8	B2	Kesim-Bantlama-Delik		
9	C1	Kesim-Bantlama	CC	Montaj-Paket
10	D1	Kesim-Bantlama-Delik	DD	Montaj-Paket
11	D2	Kesim-Bantlama-Delik		
12	D3	Kesim		
13	D4	Kesim-Bantlama		
14	D5	Kesim		
15	D6	Kesim-Bantlama-Delik		
16	E1	Kesim-Bantlama-Delik	EE	Montaj-Paket
17	E2	Kesim-Bantlama-Delik		
18	E3	Kesim-Bantlama-Delik		
19	E4	Kesim-Bantlama-Delik		
20	F1	Kesim-Bantlama-Delik	FF	Montaj-Paket
21	F2	Kesim-Bantlama-Delik		
22	F3	Kesim-Bantlama-Delik		
23	F4	Kesim-Bantlama		
24	F5	Kesim-Bantlama-Delik		
25	G1	Kesim-Bantlama-Delik	GG	Montaj-Paket
26	G2	Kesim-Bantlama-Delik		
27	G3	Kesim		
28	G4	Kesim-Bantlama		
29	G5	Kesim		
30	G6	Kesim-Bantlama-Delik		
31	H1	Kesim-Bantlama-Delik	HH	Montaj-Paket
32	H2	Kesim-Bantlama-Delik		
33	I1	Kesim-Bantlama-Delik	II	Montaj-Paket
34	I2	Kesim-Bantlama-Delik		
35	I3	Kesim-Bantlama-Delik		
36	I4	Kesim-Bantlama-Delik		
37	I5	Kesim-Bantlama		
38	I6	Kesim-Bantlama-Delik		
39	J1	Kesim-Bantlama-Delik	JJ	Montaj-Paket
40	J2	Kesim-CNC-Delik		
41	J3	Kesim-Bantlama-Delik		
42	J4	Kesim-Bantlama-Delik		
43	K1	Kesim-CNC	KK	Montaj-Paket
44	K2	Kesim-CNC		
45	K3	Kesim-Bantlama-Delik		
46	K4	Kesim-Bantlama-Delik		
47	K5	Kesim-Bantlama		

	Parçalar	Rota	Ürün	Rota
48	L1	Kesim-CNC	LL	Montaj-Paket
49	L2	Kesim-CNC		
50	L3	Kesim-Bantlama-Delik		
51	L4	Kesim-Bantlama-Delik		
52	M1	Kesim-Bantlama	MM	Montaj-Paket
53	M2	Kesim-Bantlama-Delik		
54	M3	Kesim-Bantlama-Delik		
55	M4	Kesim-Bantlama-Delik		
56	M5	Kesim-Bantlama		
57	M6	Kesim-Bantlama		
58	N1	Kesim-Bantlama-Delik	NN	Montaj-Paket
59	N2	Kesim-Bantlama-Delik		
60	N3	Kesim-Bantlama-Delik		
61	O1	Kesim-Bantlama-Delik	OO	Montaj-Paket
62	O2	Kesim-Bantlama-Delik		
63	O3	Kesim-Bantlama		
64	O4	Kesim-Bantlama-Delik		
65	O5	Kesim-Bantlama-Delik		
66	O6	Kesim		

Örnek olarak 60 adet üretilecek A ürünü için akış şeması Şekil 19'da verilmiştir. A ürünü 6 bileşenden oluşmaktadır. Her bileşen farklı sunta plakalarından kesilmektedir. Bir A ürününde hangi bileşenden kaç adet kullanıldığı bellidir. Buna göre 60 adet A ürünü için gereken bileşen miktarları hesaplanmış, bir sunta plakasından bileşenin ölçülerine göre kaç bileşen çıkacağı belirlenmiş ve her bir bileşen için kaç tane sunta plakası kullanılacağı tespit edilmiştir. Bu sunta plakaları öncelikle herhangi bir kesim makinesinde kesilerek bileşenler oluşturulur. Kesim sonrasında A1, A2 ve A3 bileşenleri 0,4 mm kalınlığında bantlayan bantlama makinesine, A5 bileşeni ise 2 mm kalınlığında bantlayan bantlama makinesine gitmektedir. A4 ve A6 bileşenleri bantlama işlemini görmemektedir. Sonrasında A1, A2 ve A3 bileşenleri herhangi bir delik makinesinde delme işlemi görmektedirler. A4, A5 ve A6 bileşenleri delme işlemini görmemektedir. A ürününe ait bileşenlerden hiç birisi CNC makinesinde işlem görmemektedir. Montaj aşamasında bütün bileşenler tek bir montaj tezgahına yığılmamakta, bekleyen bileşenler belli parti büyüklüklerine ayrılıp, bu partiler boş olan montaj tezgahlarında montaj işlemini görmektedirler. A ürünü için parti büyüklüğü 10 olarak belirlenmiştir. Yani montaj aşaması öncesinde 20 adet A1, 10 adet A2, 10 adet A3, 10 adet A4, 30 adet A5 ve 30 adet A6 bir parti oluşturmakta ve bu şekilde bir montaj tezgahına gitmektedir. Montaj için bileşenler bir araya getirilmekte ve yine aynı partiler halinde paketlemeye sevk edilmektedir. Bütün partiler paketlenince A ürünü bitmiş sayılmaktadır.



Şekil 19: 60 Adet A ürünü için Akış Şeması

3.2.1 İşlerin İşlem Sürelerinin Hesaplanması

3.2.1.1 Kesim Süreleri

Kesim sürelerinin hesaplanması Tablo 9'da verilmiştir. Her bir ürün bileşeni için ayrı bir sunta plakası kullanılmakta, her sunta plakasından da o ebatlarına göre belli sayıda bileşen çıkmaktadır. Bir plakadan çıkan bileşen sayısına ve o bileşenden üretilmesi gereken miktara göre, kaç tane sunta plakası kullanılması gerektiği hesaplanabilmektedir. Kesim makinelerinin bir yüklemede kesebileceği sunta kapasitesi 80 mm.dir. Buna göre kaç plaka kesilmesi gerekiyorsa 80 mm.yi aşmayacak şekilde yüklenebilir. 80 mm.nin aşıldığı durumlarda kalan plakaları yeniden yüklemek gerekecektir. Buna göre 30 mm. kalınlığındaki suntalardan üst üste 2 tane, 18 mm. kalınlığındaki suntalardan üst üste 4 tane, 8 mm. kalınlığındaki suntalardan üst üste 10 tane tek yüklemede kesilebilmektedir. Standart süreler, dakika cinsinden bir yüklemede (bir ya da birden fazla plaka olabilir) sunta plakasının yüklenmesi ve makinenin boşaltılması dahil olmak üzere elde edilebilecek sayıda bileşenin kesilmesi için gereken süredir. Eğer istenen miktarda bileşen elde etmek için birden fazla yükleme yapmak gerekiyorsa standart süreyi yükleme sayısı ile çarpmak gerekir. Bunun dışında her parti değiştiğinde ilgili kesim programının ayarlanması için standart hazırlık süresi olan 0,58 dk. ve 20 plakada bir testere değişimi için gereken süre olan 8,37 dk. hazırlık süreleri vardır. Standart hazırlık süresini doğrudan işlem süresine eklemek, testere değişimi süresini ise o partideki kesilecek plaka sayısına oranlayarak işlem süresine eklemek gerekir. Kesim süreleri aşağıdaki formül ile hesaplanmıştır:

(Standart Süre × Yükleme Sayısı) + Standart Hazırlık Süresi + (Testere Değiştirme Süresi × Plaka Sayısı / Testere Değiştirmek için Gereken Plaka Sayısı)

Tablo 9: Kesme İşlemi İçin İşlerin İşlem Süreleri

	Parçalar	Standart süre	Adet	Bir Plakadan Çıkan Adet	Plaka Sayısı	Sunta kalınlığı	Yükleme Sayısı	İşlem Süresi (dk.)
1	A1	5,63	120	15	8	18mm	2	15,19
2	A2	7,82	60	28	3	18mm	1	9,66
3	A3	7,98	60	28	3	18mm	1	9,82
4	A4	3,92	60	20	3	8mm	1	5,76
5	A5	13,22	180	68	3	18mm	1	15,06
6	A6	5,00	180	45	4	8mm	1	7,25
7	B1	10,61	40	21	2	30mm	1	12,03
8	B2	9,03	80	40	2	30mm	1	10,45
9	C1	6,28	24	12	2	30mm	1	7,70
10	D1	7,61	60	32	2	30mm	1	9,03

	Parçalar	Standart süre	Adet	Bir Plakadan Çıkan Adet	Plaka Sayısı	Sunta kalınlığı	Yükleme Sayısı	İşlem Süresi (dk.)
11	D2	8,93	120	28	5	18mm	1	11,60
12	D3	6,83	60	28	3	8mm	1	8,67
13	D4	7,38	180	88	3	18mm	1	9,22
14	D5	5,31	180	45	4	8mm	1	7,56
15	D6	8,12	60	32	2	18mm	1	9,54
16	E1	3,19	40	4	10	30mm	5	20,70
17	E2	2,92	80	12	7	30mm	4	15,20
18	E3	7,68	40	14	3	18mm	1	9,52
19	E4	7,33	120	28	5	18mm	2	17,33
20	F1	4,8	40	12	4	30mm	2	11,85
21	F2	5,15	80	24	4	30mm	2	12,54
22	F3	6,58	40	24	2	18mm	1	8,00
23	F4	6,7	40	20	2	18mm	1	8,12
24	F5	7,82	80	80	1	30mm	1	8,82
25	G1	7,58	60	32	2	30mm	1	9,00
26	G2	9,26	120	28	5	18mm	1	11,93
27	G3	6,61	60	28	3	8mm	1	8,45
28	G4	7,03	180	88	3	18mm	1	8,87
29	G5	5,3	180	45	4	8mm	1	7,55
30	G6	7,67	60	32	2	18mm	1	9,09
31	H1	3,22	40	18	3	30mm	2	8,27
32	H2	10,33	80	48	2	30mm	1	11,75
33	I1	11,83	60	12	5	18mm	1	14,50
34	I2	10,31	120	15	8	18mm	2	24,55
35	I3	10,66	60	15	4	18mm	1	12,91
36	I4	8,87	60	36	2	18mm	1	10,29
37	I5	9,4	60	30	2	18mm	1	10,82
38	I6	8,91	120	30	4	18mm	1	11,16
39	J1	6,43	24	6	4	30mm	2	15,11
40	J2	8,37	24	14	2	30mm	1	9,79
41	J3	4,88	48	15	4	30mm	2	12,01
42	J4	7,62	24	8	3	18mm	1	9,46
43	K1	6,26	30	2	15	18mm	4	31,89
44	K2	6,34	30	2	15	18mm	4	32,21
45	K3	9,23	60	15	4	18mm	1	11,48
46	K4	7,95	30	8	4	18mm	1	10,20
47	K5	16,07	120	60	2	18mm	1	17,49
48	L1	5,47	30	18	2	18mm	1	6,89
49	L2	5,71	30	18	2	18mm	1	7,13
50	L3	8,99	60	45	2	18mm	1	10,41
51	L4	6,55	30	35	1	18mm	1	7,55
52	M1	6,62	30	9	4	18mm	1	8,87
53	M2	6,88	30	9	4	18mm	1	9,13
54	M3	7,67	60	20	3	18mm	1	9,51
55	M4	6,86	30	20	2	18mm	1	8,28
56	M5	12,12	120	60	2	18mm	1	13,54
57	M6	6,73	30	25	2	18mm	1	8,15
58	N1	4,56	30	4	8	30mm	4	22,15
59	N2	4,76	60	15	4	30mm	2	11,76
60	N3	7,43	30	8	4	18mm	1	9,68
61	O1	6,93	20	16	2	30mm	1	8,35
62	O2	6,5	20	20	1	18mm	1	7,50
63	O3	8,9	80	20	4	18mm	1	11,15
64	O4	7,16	40	20	2	18mm	1	8,58
65	O5	6,09	40	10	4	18mm	1	8,34
66	O6	5,7	20	4	5	8mm	1	8,37

3.2.1.2 Bantlama Süreleri

Bantlama sürelerinin hesaplanması Tablo 10'da verilmiştir. 8 mm.lik suntalar bantlama işlemi görmemektedir. Bunun dışında CNC makinesinde işlem gören parçaların gerekiyorsa kenar bantları da CNC makinesinde yapıldığı için ayrıca bantlama işlemi görmemektedir. Bantlama için bir standart süre verilmemiştir. Bantlama makinesinin hızı arttırılabilmesine rağmen standart olarak 15 mt. / dk. hızda sabit tutulmaktadır. Buna göre;

[Bantlanacak Kenar Sayısı × (Bantlanacak Kenar Uzunluğu + Bırakılan Boşluk) × Adet] / Bantlama Makinesi Hızı

formülü ile bantlama işlem süreleri hesaplanabilmektedir. Bırakılan boşluk küçük parçalar için 30 cm., 1 mt. uzunluğu aşan parçalar için 70 cm. olarak belirtilmiştir.

Tablo 10: Bantlama İşlemi İçin İşlerin İşlem Süreleri

	Parçalar	Adet	Ölçü	Bantlanacak Kenar Sayısı	Bant Kalınlığı	İşlem Süresi (dk.)
1	A1	120	64*48	2	0,4 mm	15,04
2	A2	60	48*37	2	0,4 mm	6,24
3	A3	60	48*40	4	0,4 mm	11,84
4	A4	60	62*39	-	-	-
5	A5	180	21*40	4	2 mm	28,8
6	A6	180	40*34	-	-	-
7	B1	40	50*50	2	2 mm	4,27
8	B2	80	35*40	4	2 mm	14,51
9	C1	24	80*50	4	2 mm	6,08
10	D1	60	41*45	4	2 mm	11,68
11	D2	120	48*43	2	0,4 mm	12,48
12	D3	60	48*39	-	-	-
13	D4	180	16*40	4	2 mm	27,84
14	D5	180	40*34	-	-	-
15	D6	60	43*40	4	0,4 mm	11,68
16	E1	40	180*90	4	2 mm	21,87
17	E2	80	60*80	2	2 mm	10,67
18	E3	40	164*25	1	0,4 mm	3,01
19	E4	120	164*12	1	0,4 mm	9,04
20	F1	40	90*55	4	2 mm	10,99
21	F2	80	60*45	2	2 mm	9,60
22	F3	40	75*30	1	0,4 mm	2,67
23	F4	40	73*40	4	0,4 mm	9,28
24	F5	80	12*45	2	2 mm	8,00
25	G1	60	41*45	4	2 mm	11,68
26	G2	120	48*43	2	0,4 mm	12,48
27	G3	60	48*39	-	-	-
28	G4	180	16*40	4	2 mm	27,84
29	G5	180	40*34	-	-	-
30	G6	60	43*40	4	0,4 mm	11,52
31	H1	40	60*50	4	2 mm	9,07
32	H2	80	30*40	2	2 mm	7,47
33	I1	60	90*60	4	2 mm	16,80

	Parçalar	Adet	Ölçü	Bantlanacak Kenar Sayısı	Bant Kalınlığı	İşlem Süresi (dk.)
34	I2	120	71*50	2	0,4 mm	16,16
35	I3	60	71*50	2	0,4 mm	8,08
36	I4	60	27*50	2	0,4 mm	6,40
37	I5	60	54*35	4	0,4 mm	12,00
38	I6	120	57*35	2	0,4 mm	13,92
39	J1	24	160*60	4	2 mm	11,52
40	J2	24	161*25	-	-	-
41	J3	48	70*60	2	2 mm	6,40
42	J4	24	145*40	2	0,4 mm	6,88
43	K1	30	201*91	-	-	-
44	K2	30	201*91	-	-	-
45	K3	60	70*60	2	0,4 mm	8,00
46	K4	30	153*45	2	0,4 mm	8,92
47	K5	120	70*15	2	0,4 mm	16,00
48	L1	30	61*51	-	-	-
49	L2	30	61*51	-	-	-
50	L3	60	40*35	2	0,4 mm	5,60
51	L4	30	47*35	2	0,4 mm	3,08
52	M1	30	100*60	4	2 mm	12,00
53	M2	30	100*60	4	2 mm	12,00
54	M3	60	70*45	2	0,4 mm	8,00
55	M4	30	73*45	2	0,4 mm	4,12
56	M5	120	70*15	2	2 mm	16,00
57	M6	30	71*35	4	0,4 mm	6,64
58	N1	30	160*80	4	2 mm	15,20
59	N2	60	70*60	2	2 mm	8,00
60	N3	30	145*40	2	0,4 mm	8,60
61	O1	20	80*38	4	2 mm	4,75
62	O2	20	80*35	4	0,4 mm	4,69
63	O3	80	77*33	2	0,4 mm	11,41
64	O4	40	72*40	4	2 mm	9,17
65	O5	40	175*35	2	0,4 mm	13,07
66	O6	20	177*79	-	-	-

3.2.1.3 CNC Süreleri

CNC makinesindeki işlem süreleri Tablo 11'de verilmiştir. CNC'ye giren parça sayısı fazla değildir. Genelde eğrisel işlem görmesi gereken parçalar CNC'ye girmektedir. CNC standart süresine parçanın yerleştirilmesi, çıkarılması gibi süreler dahildir. Makinede işlem gören ilk parça da dahil olmak üzere partiler arası geçişte makineden uygun programın seçilmesi, artıkların temizlenmesi vb. işler için standart hazırlık süresi vardır (2,67 dk.). Bu süre standart olduğu için işlem süresine eklenebilir. CNC makinesinde işlem sürelerinin hesaplanmasında kullanılan formül:

(Adet × Standart Süre) + Standart Hazırlık Süresi

Tablo 11: CNC Makinesinde İşlerin İşlem Süreleri

	Parçalar	Adet	Standart Süre	İşlem Süresi (dk.)
39	J1	24	-	-
40	J2	24	2,02	51,15
41	J3	48	-	-
42	J4	24	-	-
43	K1	30	4	122,67
44	K2	30	4,94	150,87
45	K3	60	-	-
46	K4	30	-	-
47	K5	120	-	-
48	L1	30	2,5	77,67
49	L2	30	3,03	93,57
50	L3	60	-	-
51	L4	30	-	-

3.2.1.4 Delik İşlemi Süreleri

Delik makinesindeki işlem sürelerinin hesaplanması Tablo 12'de verilmiştir. Partiler arası geçişte matkapların sökülmesi, yeni partiye göre ayarlanması için standart bir hazırlık süresi vardır (3,27 dk.). Delik işlemi için işlerin işlem sürelerinin hesaplanmasında kullanılan formül:

(Adet × Standart Süre) + Standart Hazırlık Süresi

Tablo 12: Delik İşlemi İçin İşlerin İşlem Süreleri

	Parçalar	Adet	Standart Süre	İşlem Süresi (dk.)
1	A1	120	0,13	18,87
2	A2	60	0,14	11,67
3	A3	60	0,12	10,47
4	A4	60	-	-
5	A5	180	-	-
6	A6	180	-	-
7	B1	40	0,11	7,67
8	B2	80	0,13	13,67
9	C1	24	-	-
10	D1	60	0,12	10,47
11	D2	120	0,15	21,27
12	D3	60	-	-
13	D4	180	-	-
14	D5	180	-	-
15	D6	60	0,1	9,27
16	E1	40	0,12	8,07
17	E2	80	0,2	19,27
18	E3	40	0,12	8,07
19	E4	120	0,24	32,07
20	F1	40	0,12	8,07
21	F2	80	0,25	23,27
22	F3	40	0,15	9,27
23	F4	40	-	-
24	F5	80	0,3	27,27

	Parçalar	Adet	Standart Süre	İşlem Süresi (dk.)
25	G1	60	0,12	10,47
26	G2	120	0,15	21,27
27	G3	60	-	-
28	G4	180	-	-
29	G5	180	-	-
30	G6	60	0,1	9,27
31	H1	40	0,1	7,27
32	H2	80	0,12	12,87
33	I1	60	0,15	12,27
34	I2	120	0,14	20,07
35	I3	60	0,18	14,07
36	I4	60	0,14	11,67
37	I5	60	-	-
38	I6	120	0,14	20,07
39	J1	24	0,95	26,07
40	J2	24	0,35	11,67
41	J3	48	0,22	13,83
42	J4	24	0,24	9,03
43	K1	30	-	-
44	K2	30	-	-
45	K3	60	0,14	11,67
46	K4	30	0,24	10,47
47	K5	120	-	-
48	L1	30	-	-
49	L2	30	-	-
50	L3	60	0,15	12,27
51	L4	30	0,16	8,07
52	M1	30	-	-
53	M2	30	0,14	7,47
54	M3	60	0,14	11,67
55	M4	30	0,16	8,07
56	M5	120	-	-
57	M6	30	-	-
58	N1	30	0,2	9,27
59	N2	60	0,16	12,87
60	N3	30	0,18	8,67
61	O1	20	0,16	6,47
62	O2	20	0,18	6,87
63	O3	80	-	-
64	O4	40	0,1	7,27
65	O5	40	0,25	13,27
66	O6	20	-	-

3.2.1.5 Montaj Süreleri

Montaj süreleri nispeten uzun olduğu için bir ürüne ait bileşenlerin tamamını bir montaj masasına yönlendirmektense bileşenleri belli büyüklükte partilere bölüp birden fazla masaya partiler halinde dağıtma yolu tercih edilmektedir. Tablo 13'de her bir iş için uygun parti büyüklükleri ve adetleri verilmiştir.

Tablo 13: Montaj İçin Uygun Parti Büyüklükleri

Sıra	Ürün Adı	Parti Büyüklüğü	Parti Adedi
1	A	10	6
2	B	20	2
3	C	24	1
4	D	10	6
5	E	10	4
6	F	10	4
7	G	10	6
8	H	20	2
9	I	20	3
10	J	12	2
11	K	10	3
12	L	10	3
13	M	10	3
14	N	10	3
15	O	10	2

Montaj ile ilgili süreler Tablo 14'de verilmiştir. Standart süreler her bir ürünün parçalarının bir araya getirilip, gerekli malzemelerin eklenip, belli bir sırada dizilip, paketlemeye hazır hale getirilene kadar geçen süredir. Ürün bileşenleri montaja partiler halinde geldiğinden her bir partinin işlem süresi standart süre ile parti büyüklüğünün çarpılmasıyla elde edilmiştir. İş sayısı ise parti adedi kadar artmıştır. Her bir parti birbirinden bağımsız hareket edeceğinden her birini ayrı bir iş olarak ele almak gerekir.

Tablo 14: Montaj İçin İşlerin İşlem Süreleri

Sıra	Ürün Adı	Standart Süre	Parti Büyüklüğü	İşlem Süresi (dk.)
1	AA1	10,36	10	103,6
2	AA2	10,36	10	103,6
3	AA3	10,36	10	103,6
4	AA4	10,36	10	103,6
5	AA5	10,36	10	103,6
6	AA6	10,36	10	103,6
7	BB1	2,55	20	51
8	BB2	2,55	20	51
9	CC1	1,55	24	37,2
10	DD1	10,06	10	100,6
11	DD2	10,06	10	100,6
12	DD3	10,06	10	100,6
13	DD4	10,06	10	100,6
14	DD5	10,06	10	100,6
15	DD6	10,06	10	100,6
16	EE1	7,9	10	79
17	EE2	7,9	10	79
18	EE3	7,9	10	79
19	EE4	7,9	10	79
20	FF1	6,86	10	68,6
21	FF2	6,86	10	68,6
22	FF3	6,86	10	68,6
23	FF4	6,86	10	68,6
24	GG1	11,09	10	110,9

Sıra	Ürün Adı	Standart Süre	Parti Büyüklüğü	İşlem Süresi (dk.)
25	GG2	11,09	10	110,9
26	GG3	11,09	10	110,9
27	GG4	11,09	10	110,9
28	GG5	11,09	10	110,9
29	GG6	11,09	10	110,9
30	HH1	1,96	20	39,2
31	HH2	1,96	20	39,2
32	II1	3,02	20	60,4
33	II2	3,02	20	60,4
34	II3	3,02	20	60,4
35	JJ1	3,79	12	45,48
36	JJ2	3,79	12	45,48
37	KK1	9,89	10	98,9
38	KK2	9,89	10	98,9
39	KK3	9,89	10	98,9
40	LL1	2,5	10	25
41	LL2	2,5	10	25
42	LL3	2,5	10	25
43	MM1	7,12	10	71,2
44	MM2	7,12	10	71,2
45	MM3	7,12	10	71,2
46	NN1	2,96	10	29,6
47	NN2	2,96	10	29,6
48	NN3	2,96	10	29,6
49	OO1	6,96	10	69,6
50	OO2	6,96	10	69,6

3.2.1.6 Paketleme Süreleri

Paketleme ile ilgili süreler Tablo 15'de verilmiştir. Paketleme masalarına işler montajdaki gibi partiler halinde gelmektedir. Paketleme ile ilgili standart süreler, her bir ürünün temizliği ve demonte şekilde paketlenmesi için geçen sürelerdir. Dolayısıyla işlem süresini hesaplamak için doğrudan standart süre ile parti büyüklüğü çarpılmıştır.

Tablo 15: Paketleme İçin İşlerin İşlem Süreleri

Sıra	Ürün Adı	Standart Süre	Parti Büyüklüğü	İşlem Süresi (dk.)
1	AA1	4,77	10	47,7
2	AA2	4,77	10	47,7
3	AA3	4,77	10	47,7
4	AA4	4,77	10	47,7
5	AA5	4,77	10	47,7
6	AA6	4,77	10	47,7
7	BB1	2,92	20	58,4
8	BB2	2,92	20	58,4
9	CC1	2,44	24	58,56
10	DD1	5,12	10	51,2
11	DD2	5,12	10	51,2
12	DD3	5,12	10	51,2
13	DD4	5,12	10	51,2
14	DD5	5,12	10	51,2
15	DD6	5,12	10	51,2

Sıra	Ürün Adı	Standart Süre	Parti Büyüklüğü	İşlem Süresi (dk.)
16	EE1	7,16	10	71,6
17	EE2	7,16	10	71,6
18	EE3	7,16	10	71,6
19	EE4	7,16	10	71,6
20	FF1	4,41	10	44,1
21	FF2	4,41	10	44,1
22	FF3	4,41	10	44,1
23	FF4	4,41	10	44,1
24	GG1	5	10	50
25	GG2	5	10	50
26	GG3	5	10	50
27	GG4	5	10	50
28	GG5	5	10	50
29	GG6	5	10	50
30	HH1	3,14	20	62,8
31	HH2	3,14	20	62,8
32	II1	4,91	20	98,2
33	II2	4,91	20	98,2
34	II3	4,91	20	98,2
35	JJ1	5,49	12	65,88
36	JJ2	5,49	12	65,88
37	KK1	6,1	10	61
38	KK2	6,1	10	61
39	KK3	6,1	10	61
40	LL1	3,02	10	30,2
41	LL2	3,02	10	30,2
42	LL3	3,02	10	30,2
43	MM1	3,91	10	39,1
44	MM2	3,91	10	39,1
45	MM3	3,91	10	39,1
46	NN1	4,81	10	48,1
47	NN2	4,81	10	48,1
48	NN3	4,81	10	48,1
49	OO1	7,45	10	74,5
50	OO2	7,45	10	74,5

Ürün bileşenlerinin tamamı için tüm işlem süreleri Tablo 16'da bir arada görülmektedir.

İşler için teslim tarihleri olarak gün sonları belirlenmiştir. Gün ortasına yetişmesi gereken iş yoktur. AA1, CC1 ve II1 işlerinin teslim tarihi ilk günün sonu (480. dk.), AA2, JJ1, JJ2, NN1 ve OO1 işlerinin teslim tarihi ikinci günün sonu (960. dk.), geriye kalan tüm işlerin teslim tarihi ise üçüncü günün sonu (1440. dk.) olarak belirlenmiştir.

Tablo 16: Belirlenen Parti Büyüklükleri İçin İşlem Süreleri (dk.)

	Parçalar	Kesim Süresi	Bantlama Süresi	CNC Süresi	Delik Süresi		Ürünler (partilere ayrılmış)	Montaj Süresi	Paket Süresi
1	A1	15,2	15,0		18,9	1	AA1	103,6	47,7
2	A2	9,7	6,2		11,7	2	AA2	103,6	47,7
3	A3	9,8	11,8		10,5	3	AA3	103,6	47,7
4	A4	5,8	-		-	4	AA4	103,6	47,7
5	A5	15,1	28,8		-	5	AA5	103,6	47,7
6	A6	7,3	-		-	6	AA6	103,6	47,7
7	B1	12,0	4,3		7,7	7	BB1	51	58,4
8	B2	10,5	14,5		13,7	8	BB2	51	58,4
9	C1	7,7	6,1		-	9	CC1	37,2	58,6
10	D1	9,0	11,7		10,5	10	DD1	100,6	51,2
11	D2	11,6	12,5		21,3	11	DD2	100,6	51,2
12	D3	8,7	-		-	12	DD3	100,6	51,2
13	D4	9,2	27,8		-	13	DD4	100,6	51,2
14	D5	7,6	-		-	14	DD5	100,6	51,2
15	D6	9,5	11,7		9,3	15	DD6	100,6	51,2
16	E1	20,7	21,9		8,1	16	EE1	79	71,6
17	E2	15,2	10,7		19,3	17	EE2	79	71,6
18	E3	9,5	3,0		8,1	18	EE3	79	71,6
19	E4	17,3	9,0		32,1	19	EE4	79	71,6
20	F1	11,9	11,0		8,1	20	FF1	68,6	44,1
21	F2	12,5	9,6		23,3	21	FF2	68,6	44,1
22	F3	8,0	2,7		9,3	22	FF3	68,6	44,1
23	F4	8,1	9,3		-	23	FF4	68,6	44,1
24	F5	8,8	8,0		27,3				
25	G1	9,0	11,7		10,5	24	GG1	110,9	50
26	G2	11,9	12,5		21,3	25	GG2	110,9	50
27	G3	8,5	-		-	26	GG3	110,9	50
28	G4	8,9	27,8		-	27	GG4	110,9	50
29	G5	7,6	-		-	28	GG5	110,9	50
30	G6	9,1	11,5		9,3	29	GG6	110,9	50
31	H1	8,3	9,1		7,3	30	HH1	39,2	62,8
32	H2	11,8	7,5		12,9	31	HH2	39,2	62,8
33	I1	14,5	16,8		12,3				
34	I2	24,6	16,2		20,1	32	II1	60,4	98,2
35	I3	12,9	8,1		14,1	33	II2	60,4	98,2
36	I4	10,3	6,4		11,7	34	II3	60,4	98,2
37	I5	10,8	12,0		-				
38	I6	11,2	13,9		20,1				
39	J1	15,1	11,5	-	26,1				
40	J2	9,8	-	51,2	11,7	35	JJ1	45,5	65,9
41	J3	12,0	6,4	-	13,9	36	JJ2	45,5	65,9
42	J4	9,5	6,9	-	9,0				
43	K1	31,9	-	122,7	-				
44	K2	32,2	-	150,9	-	37	KK1	98,9	61
45	K3	11,5	8,0	-	11,7	38	KK2	98,9	61
46	K4	10,2	8,9	-	10,5	39	KK3	98,9	61
47	K5	17,5	16,0	-	-				
48	L1	6,9	-	77,7	-	40	LL1	25	30,2
49	L2	7,1	-	93,6	-	41	LL2	25	30,2
50	L3	10,4	5,6	-	12,3	42	LL3	25	30,2
51	L4	7,6	3,1	-	8,1				
52	M1	8,9	12,0		-				
53	M2	9,1	12,0		7,5	43	MM1	71,2	39,1
54	M3	9,5	8,0		11,7	44	MM2	71,2	39,1
55	M4	8,3	4,1		8,1	45	MM3	71,2	39,1
56	M5	13,5	16,0		-				
57	M6	8,2	6,6		-				
58	N1	22,2	15,2		9,3	46	NN1	29,6	48,1
59	N2	11,8	8,0		12,9	47	NN2	29,6	48,1
60	N3	9,7	8,6		8,7	48	NN3	29,6	48,1
61	O1	8,4	4,8		6,5				
62	O2	7,5	4,7		6,9				
63	O3	11,2	11,4		-	49	OO1	69,6	74,5
64	O4	8,6	9,2		7,3	50	OO2	69,6	74,5
65	O5	8,3	13,1		13,3				
66	O6	8,4	-		-				

3.3 Uygulama Problemi için Varsayımlar

Genel olarak çizelgeleme problemlerine ilişkin varsayımlar (French, 1982, 8-9) uygulama problemi için de geçerlidir:

- Her iş bir varlıktır (entity): Her iş farklı operasyonlardan oluşmasına rağmen aynı işin iki operasyonu eş zamanlı olarak yapılamaz.

- İşlem bölme yoktur: Bir makinede başka bir operasyonun başlayabilmesi için önce başlamış bir operasyonun bitmesi gerekir. Makinede başlanmış bir iş yarım bırakılıp başka bir işe devam edilmez.

- Bir iş her aşamada bir tane olmak üzere, *s* farklı operasyona sahip olmak zorunda değildir: Klasik akış tipi problemde bir işin aynı makinede ikinci kez işlem görmesine izin verilmemekte, aynı şekilde her iş her makinede işlem görmektedir. İşin bir veya daha fazla makineyi atlaması mümkün değildir. Fakat ele alınan uygulama probleminde aşama atlama mümkündür, bazı işler bazı aşamaları atlamakta, o aşamalarda işlem görmemektedir.

- İş iptali söz konusu değildir: Her iş bitene kadar işlem görmelidir. Çizelgeleme süreci sonunda işlerin bütün işlemleri tamamlanmış olmalıdır.

- İşlem süreleri çizelgeden (sıralamadan) bağımsızdır: İşlem süreleri işlerin sıralamasına bağlı değildir. Fakat bazı işlemler için hazırlık süreleri sıralamaya bağımlıdır. Sıralamaya bağlı olmayan hazırlık süreleri işlem sürelerine eklenmiştir; bunlar dışında sıra bağımlı hazırlık süreleri vardır. Ayrıca işlerin makineler arasındaki taşınma zamanları ihmal edilmiştir.

- Ara stoğa izin verilmektedir: İşler bir sonraki işlem görecekleri makinelerin boşalmasını (serbest kalmasını) bekleyebilirler. Bazı imalatlar doğası gereği beklemesiz yapılmalıdır; bir operasyonu biten iş beklemeden diğer operasyona başlamalıdır. Bazı tesislerde de makineler arasında sınırlı stok alanından dolayı sınırlı bir ara stoğa izin verilmektedir. Fakat bu problem için ara stok ve ara stoğun bekleme süresi sınırlı değildir.

- Her bir makine tipinden birden fazla olabilir: Klasik akış tipi problemde işin işlemleri sırasında makine seçimi durumuna izin verilmemiştir. Fakat esnek akış tipi problemde en az bir aşamada paralel makineler vardır ve paralel makine durumu olan aşamalarda işlerin makine seçmesi mümkündür.

- Makineler boş bekleyebilir.

- Hiçbir makine, aynı anda birden fazla operasyonu işleyemez.

- Makineler asla bozulmaz ve çizelgeleme periyodu boyunca kullanıma hazırdır.

- Teknolojik kısıtlar önceden bilinir ve sabittir: Bir işin operasyon sırası sabittir ve bilinmektedir.

- Rassallık söz konusu değildir: işlerin sayısı, makinelerin sayısı, işlem süreleri, hazır olma zamanları ve problemi tanımlamak için gereken tüm diğer değerler sabittir ve bilinmektedir. Belirsizlik yoktur. Bu problemlere *statik* problemler denmektedir. İşlem zamanları ve tüm diğer parametreler sabit olduğu ve bilindiği için problem *deterministik*tir. İşlerin belirli bir zaman diliminde rastlantısal olarak geldiği problemler *dinamik*tir. İşlem zamanları vs. parametrelerin belirsiz olduğu problemlere de *stokastik* denmektedir (French, 1982, p.16).

3.4 Uygulama Problemi için Matematiksel Model

Uygulama problemi için matematiksel model, 2.2.1 no.lu başlıkta esnek akış tipi çizelgeleme problemi için verilmiş olan matematiksel modele sıra bağımlı hazırlık zamanları, aşama atlama, öncelik kısıtları ile ilgili sınırlar eklenerek elde edilmiştir. Uygulama problemi maksimum tamamlanma zamanı amaç fonksiyonu için $FF6\ (Pm)\ |\ s_{jk},\ M_j,\ prec\ |\ C_{\max}$ şeklinde, toplam gecikme amaç fonksiyonu için ise $FF6\ (Pm)\ |\ s_{jk},\ M_j,\ prec\ |\ \sum T_j$ şeklinde ifade edilebilir. Modeldeki sınırlar makinelere atama, işlerin sıralanması ve zamanlama alt başlıklarında toplanabilir. Ele alınan uygulama problemi için maksimum tamamlanma zamanını minimize etmeyi amaçlayan 0-1 karma tamsayılı doğrusal programlama modeli aşağıdaki gibidir:

$$Min\ C_{max} \tag{1}$$

Sınırlar:

$$\sum_{m \in M_s} X_{sjm} = 1 \tag{2}$$
$$(\forall\ s, j\ \in\ J_s)$$

$$X_{sjm} - \sum_{i \in J_{sm}} Q_{sijm} \geq 0 \tag{3}$$
$$(\forall\ s, j, m\ \in\ J_{sm}\ ; i \neq j)$$

$$X_{sim} - \sum_{j \in J_{sm}} Q_{sijm} \geq 0 \tag{4}$$
$$(\forall\ s, i, m\ \in\ J_{sm}\ ; i \neq j)$$

$$\sum_{j \in J_{sm}} X_{jsm} - \sum_{i \in J_{sm}} \sum_{j \in J_{sm}} Q_{sijm} = 1 \tag{5}$$
$$(\forall\ s, m\ \in\ M_s\ ; i \neq j)$$

$$C_{sjm} \geq P_{sjm} \times X_{sjm} \tag{6}$$
$$(\forall\ j, m\ \in\ J_{sm}\ ; s = 1)$$

$$AS_i \geq K_{sj} \tag{7}$$
$$(\forall\ i, j\ \in\ G_{ij}\ ; s = s' - 1)$$

$$C_{sjm} \geq A_{sj} + P_{sjm} \times X_{sjm} \tag{8}$$
$$(\forall\ j, m\ \in\ J_{sm}\ ; s = s')$$

$$C_{sjm} - C_{sim} + L\left(1 - Q_{sijm}\right) \geq P_{sjm} + Q_{sijm} \times SU_{ijs} \tag{9}$$
$$(\forall\ s, i, j, m\ \in\ J_{sm}\ ; i \neq j)$$

$$C_{sjm} - K_{(s-1)j} \geq P_{sjm} - L\left(1 - X_{sjm}\right) \tag{10}$$
$$(\forall\ s, j, m\ \in\ J_{sm}\ ; s > 1)$$

$$K_{sj} \geq K_{(s-1)j} \quad (11)$$
$$(\forall\, s,j \in JB_s\,; s > 1)$$

$$C_{sjm} \leq K_{sj} \quad (12)$$
$$(\forall\, s,j,m \in J_{sm})$$

$$K_{sj} \leq C_{max} \quad (13)$$
$$(\forall\, j \in JA_s\,; s = s*)$$

C_{sjm}, K_{sj}, AS_j, $C_{\max} \geq 0$; X_{sjm}, $Q_{sijm} = \{0,1\}$

İndisler, Kümeler ve Parametreler:

i, j İş indisleri

s Aşama indisi

s' Montaj aşaması

$s*$ Son aşama

m Makine indisi

M_s Aşamalardaki makineler kümesi

J_s Aşamalardaki işler kümesi

J_{sm} Aşamalardaki işler ve makineler kümesi

JB_s Montaj öncesindeki aşamalarda işler kümesi

JA_s Montaj ve sonrasındaki aşamalarda işler kümesi

G_{ij} Montaj ve sonrasındaki işler ile montaj öncesindeki işler arasında ilişkiler kümesi

P_{sjm} j işinin s aşamasında m makinesindeki işlem süresi

SU_{ijs} s aşamasında i işinden j işine hazırlık zamanı

L Büyük bir sayı

Karar Değişkenleri:

C_{max} Maksimum tamamlanma zamanı

C_{sjm} j işinin s aşamasında m makinesinden ayrılma zamanı

K_{sj} j işinin s aşamasından ayrılma zamanı

AS_j j işinin montaja başlayabileceği zaman

Q_{sijm} = 1 eğer s aşamasındaki m makinesinde j işi i işinden sonra geliyorsa
= 0 diğer

X_{sjm} = 1 eğer *j* işi *s* aşamasındaki *m* makinesinde işlem görecekse
= 0 diğer

2 no.lu sınır *j* işinin her aşamada sadece bir makinede işlem görmesini garantiliyor.

3 no.lu sınır *r* aşamasındaki *m* makinesinde her *j* işinin en çok bir *i* işini takip edeceğini ifade etmektedir. 4 no.lu sınır *r* aşamasındaki *m* makinesinde her *i* işinin en çok bir *j* işi tarafından takip edileceğini ifade etmektedir.

5 no.lu sınır her aşamadaki her makinede takip sayısının (Q_{sijm}) o makinede işlem görecek iş sayısından bir eksik olmasını sağlamaktadır. Örneğin bir aşamadaki makineye 5 iş atanmışsa o makinede 4 iş birbirini takip etmektedir.

6 no.lu sınır ilk aşama için *j* işinin *m* makinesinden ayrılma zamanının en az o işin o makinedeki işlem süresi kadar olmasını garantilemektedir. İlk aşamada hazırlık süreleri işlem sürelerine eklendiğinden hazırlık sürelerini denkleme katmaya gerek yoktur.

7 no.lu sınır bir işin montajına başlanabilecek zamanı belirlemektedir. O işin montajına başlanabilmesi için o ürüne ait bütün bileşenlerin yani montaj öncesi işlerin bitmiş olmasını garantilemektedir. Montaj öncesindeki aşamalardaki işlerin montaj aşamasında dönüşeceği işler bir kümede takip edilmektedir. Buna göre bir işin montajına başlanabilmesi için o işe bağlı bütün montaj öncesi işlerin bitmiş olması gerekmekte, son biten işin tamamlanma zamanı ise montaja başlama için alt sınırı oluşturmaktadır.

8 no.lu sınır 6 no.lu sınırın montaj aşamasındaki durumu olmakla birlikte, bu aşamada j işinin montaja başlayabileceği zaman (AS_j) üzerine *j* işinin işlem süresi eklenmektedir.

9 no.lu sınır *s* aşamasında *j* işi *m* makinesinde *i* işinden sonra işlem görecekse *j* işinin makineden ayrılma zamanı ile *i* işinin makineden ayrılma zamanları arasındaki farkın en az *j* işinin *m* makinesindeki işlem süresi ile *i* ve *j* işleri arasındaki hazırlık süresi toplamı kadar olmasını sağlamaktadır.

10 no.lu sınır *j* işinin *s* aşamasında *m* makinesinden ayrılma zamanı ile s-1 aşamasından ayrılma zamanı arasındaki sürenin en az *j* işinin *s* aşamasında *m* makinesindeki işlem süresi kadar olmasını sağlamaktadır.

11 no.lu sınır problemin montaj öncesinde aşama atlama olmasından dolayı konulan bir sınırdır. j işinin s-1 aşamasını atlıyor olması durumunda $K_{(s-1)j}$ değişkeninin 0 değerini almaması için bir işin bir aşamada işlem görmese bile o aşamadan ayrılma zamanının en az bir önceki aşamadan ayrılma zamanı kadar olmasını sağlamaktadır.

12 no.lu sınır bir işin bir aşamadan ayrılma zamanının o işin o aşamadaki makinelerden ayrılma zamanlarından büyük olanı kadar olmasını sağlamaktadır.

13 no.lu sınır maksimum tamamlanma zamanının işler içinde son aşamadan ayrılma zamanı en büyük olan işe eşit olmasını sağlamaktadır.

Klasik akış tipi çizelgeleme problemleri literatürüne ilişkin bir çalışmada (Dudek; Panwalkar; Smith, 1992) maksimum tamamlanma zamanı minimizasyonunun akademik çalışmalarda yoğun olarak kullanılmasının aksine, pratikte önemli olmadığı ifade edilmektedir. Tamamlanma zamanının önemli olmamasının nedenlerini literatürde problemlerin tanımlandığı haliyle aslında pratikte çok nadir var olması ya da hiç olmaması, olsa bile çok kriterli olması, varsayımların fazla kısıtlayıcı olması, algoritmaların varsayılan durumlar dışındaki durumları ele almada esnek ve etkili olmaması ve gerçek akış tipi ortamların genellikle statik değil dinamik olması şeklinde açıklamaktadırlar. Bu nedenle tezin uygulama problemi için de firma açısından maksimum tamamlanma zamanından daha önemli bir kriter olan gecikmelerin toplamının minimize edilmesi daha önem arz etmektedir. Buna ilişkin model için mevcut modele sadece gecikmelerle ilgili sınır eklenir.

$$K_{sj} - T_j \leq D_j \;\; ; \forall\, j \in JA_s \,; s = s* \qquad (14)$$

Amaç fonksiyonu da toplam gecikmeyi minimize edecek şekilde değiştirilir.

$$Min \sum_{j \in JA_s} T_j \qquad (15)$$

D_j j işinin teslim tarihi

T_j j işinin gecikme süresi

3.4.1 Matematiksel Modelin Çözümü

Model, matematiksel modelleme dili GAMS'da yazılmış (GAMS'da yazılmış hali için bkz: EK-1) ve çözücü olarak CPLEX 11.2.1 kullanılmıştır. Problemin çözümünde 2,40 GHz Core İ5 işlemcili, 6 GB RAM'li bir bilgisayar kullanılmıştır.

Problem maksimum tamamlanma zamanı ve toplam gecikmeyi ayrı amaç fonksiyonlarında minimize edecek şekilde çözülmüştür. Ancan iki amaç fonksiyonu için de optimum çözüme ulaşılamamıştır. Maksimum tamamlanma zamanı kriterini minimize etme amacı için 1 hafta (604.800 saniye) boyunca çalıştırılmasına rağmen optimum çözüm bulunamadığı gibi, 3 gün çalışma sonucu elde edilen çözümden daha iyi bir çözüme ulaşılamamıştır. 3 gün, 1 gün ve 2 saat süre çalıştırılma ile elde edilen sonuçlar Tablo 17'de verilmiştir (Matematiksel modellerin çözüm sonuçları için bkz. EK-5).

En iyi C_{max} (maksimum tamamlanma zamanı) değeri (1278), maksimum tamamlanma zamanını minimize etmeyi amaçlayan modelin 3 gün süreyle çalıştırılması sonucu elde edilmiştir. $\sum T_j$ amaç fonksiyonlu matematiksel modelin 3 gün süreyle çalışma sonucu bulabildiği en iyi $\sum T_j$ değeri 2297,4'tür.

Tablo 17: Matematiksel Çözüm Sonuçları

Amaç Fonksiyonu	Çalıştırma Süresi	C_{max}	$\sum T_j$
C_{max}	2 saat	2381,1	14617,7
	1 gün	1576	4014
	3 gün	1278	1468,8
$\sum T_j$	2 saat	1813,2	3650,2
	1 gün	1685,0	2798,3
	3 gün	1585,9	2297,4

3.5 Uygulama Problemi İçin Sezgisel Yöntemler

Matematiksel model ile bu büyüklükte bir problemde kısa sürede optimum çözüme ulaşılamadığı için temel bazı sıralama kurallarına dayalı iki sezgisel yöntem probleme uyarlanmış (Uyarlanmış Johnson Kuralı ve En Az Boşluk Kuralı), karşılaştırma için de

iki basit sıralama kuralı (En Kısa İşlem Zamanı ve En Uzun İşlem Zamanı) kullanılmıştır. Sezgisel modeller C# dilinde kodlanmıştır (Kodlamalar için bkz: EK-3).

3.5.1 Sıralama Kuralları

3.5.1.1 En Kısa İşlem Zamanı

SPT (Shortest Processing Time) olarak da adlandırılan En Kısa İşlem Zamanı (EKİZ) kuralı probleme göre uyarlanmıştır. Her bir işin toplam işlem süresini hesaplayıp, bu değerlere göre işleri artan şekilde sıralayan bir yöntemdir. Uygulama probleminde montaj öncesi işler montaj aşamasında farklı işlere dönüştüğü için toplam işlem süreleri ürün bazında alınmıştır. Her bir ürünün tüm bileşenlerinin tüm aşamalardaki toplam işlem sürelerinin artan şekilde sıralaması esas sıralamayı oluşturur. Ürünün montaj öncesi bileşenleri için montaj öncesi aşamalardaki toplam işlem süresine göre artan bir sıralama yapılır. Uygulama probleminde montaj ve sonrasındaki aynı ürüne ait işlerin işlem süreleri eşittir fakat eşit olmayan durumlar için de artan şekilde sıralama yapılabilir. Bu yöntem aşağıdaki gibi işlemektedir (Daha detaylı hali için bkz: EK-2).

Her bir ürünün tüm aşamalardaki işlerinin işlem sürelerini topla.
Bu sürelere göre ürünleri **artan** şekilde sırala.
- Sıradaki ilk ürünü seç.
- Seçilen ürünün montaj öncesindeki işlerinin işlem süreleri toplamına göre **artan** şekilde sırala.
  - Sıradaki ilk işi seç.
  - Seçilen işi işlem göreceği aşamalarda işin önceki aşamalardaki bitiş süresini ve makinelerin müsait olma durumlarını dikkate alarak uygun makineye ata.
  - Ürünün montaj öncesindeki işleri bitene kadar devam et.
- Seçilen ürünün montaj ve sonrasındaki işleri (işlem süreleri eşit olduğu için) sıradan seç.
  - Seçilen işi işlem göreceği aşamalarda montaj öncesi mevcut ürüne ait en geç biten işten erken başlamamak koşuluyla, işin önceki aşamalardaki bitiş süresini ve makinelerin müsait olma durumlarını da dikkate alarak uygun makineye ata.
  - Ürünün montaj ve sonrasındaki işleri bitene kadar devam et.
- Ürünler bitene kadar devam et.

Her kuralın nasıl çalıştığı 3 aşama 5 işlik bir örnek problemde gösterilecektir. FFS3J5 olarak adlandırılan bu probleme ait işlem süreleri ve teslim tarihleri Tablo 18'de, sıra bağımlı hazırlık zamanları Tablo 19'da, aşamalardaki makine sayıları Tablo 20'de ve makine uygunluk kısıtları Tablo 21'de görülmektedir.

Tablo 18: FFS3J5 problemi işlem süreleri ve teslim tarihleri (dk.)

		Montaj öncesi aşamalar		Montaj	Teslim Tarihleri	
Ürün	İş	1. aşama	2. aşama	3. aşama	İş	Tarih(dk.)

A	1	13	-	1	42	1	56
	2	8	16	2	42	2	-
	3	20	13				
B	4	-	5	3	28	3	31
	5	12	6	4	28	4	52

Tablo 19: FFS3J5 problemi 1. aşamadaki sıra bağımlı hazırlık süreleri (dk.)

İşe / İşten	1	2	3	4	5
1		2	2		8
2	2		2		8
3	2	2			8
4					
5	8	8	8		

Tablo 20: FFS3J5 problemi aşamalardaki makine sayıları

Aşama	1	2	3
Makine Sayısı	3	4	4

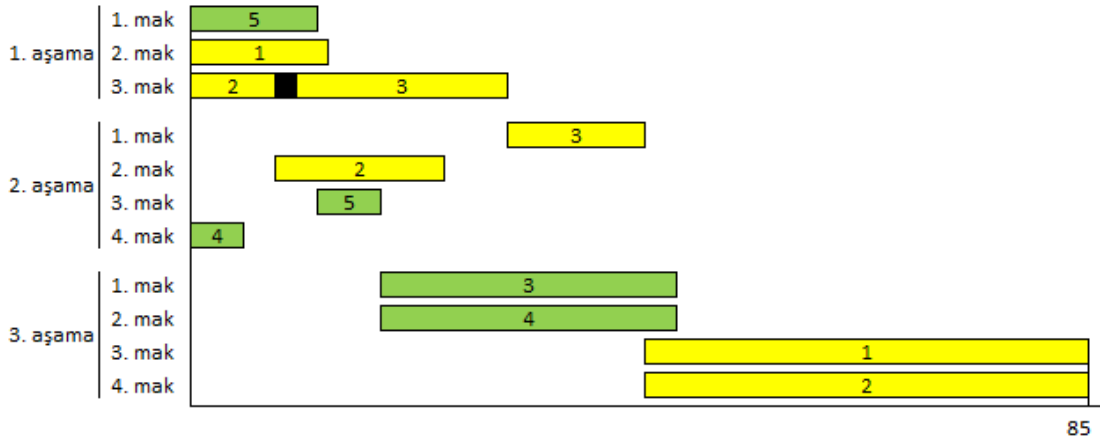
Tablo 21: FFS3J5 problemi 2. aşamadaki makine uygunluk kısıtları

İş	1	2	3	4	5
İşlem göreceği makine	-	2	1	4	3

Örnek problem üzerinde EKİZ kuralı aşağıdaki gibi çalışmaktadır:

A ürünü toplam işlem süresi : (13+8+16+20+13)+(42+42) = 154
B ürünü toplam işlem süresi : (5+12+6)+(28+28) = 79
Toplam işlem sürelerine göre ürünler B-A şeklinde sıralanacak.
B ürüne ait montaj öncesi işler:
4. işin işlem süreleri toplamı : 5
5. işin işlem süreleri toplamı : 12+6 = 18
İşler 4-5 şeklinde sıralanacak.
Montaj ve sonrasında işlem süreleri eşit olduğu için işler 3-4 şeklide sıralanacak.
A ürününe ait montaj öncesi işler:
1. işin işlem süreleri toplamı : 13
2. işin işlem süreleri toplamı : 8+16 = 24
3. işin işlem süreleri toplamı : 20+13 = 33
İşler 1-2-3 şeklinde sıralanacak.
Montaj ve sonrasında işlem süreleri eşit olduğu için işler 1-2 şeklinde sıralanacak.
Montaj öncesi aşamalarda sıralama : 4-5-1-2-3
Montaj ve sonrası aşamalarda sıralama : 3-4-1-2

Bu sıralamalara ve problemin diğer bilgilerine göre (makine uygunluk kısıtları, sıra bağımlı hazırlık zamanları ve öncelik kısıtları) işler Gannt şemasında Şekil 20'deki gibi sıralanmaktadır.



Şekil 20: EKİZ Kuralına Göre FFS3J5 Probleminin Gannt Şeması

3.5.1.2 En Uzun İşlem Zamanı

LPT (Longest Processing Time) olarak da adlandırılan En Uzun İşlem Zamanı (EUİZ) kuralı, sıralamada EKİZ kuralının tam tersidir. Ürünlerin ve işlerin sıralaması artan şekilde değil azalan şekilde yapılmaktadır. Bu yöntem aşağıdaki gibi işlemektedir (Daha detaylı hali için bkz: EK-2).

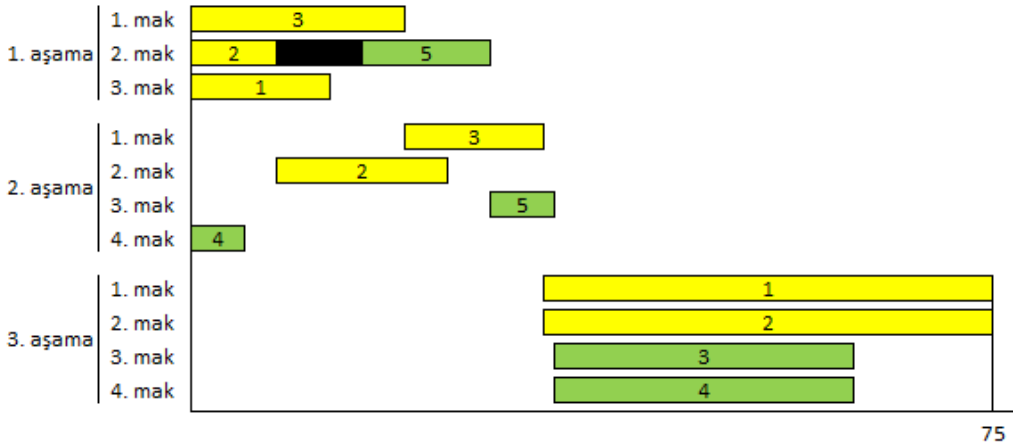
Her bir ürünün tüm aşamalardaki işlerinin işlem sürelerini topla.
Bu sürelere göre ürünleri **azalan** şekilde sırala.
Sıradaki ilk ürünü seç.
Seçilen ürünün montaj öncesindeki işlerinin işlem süreleri toplamına göre **azalan** şekilde sırala.
Sıradaki ilk işi seç.
Seçilen işi işlem göreceği aşamalarda işin önceki aşamalardaki bitiş süresini ve makinelerin müsait olma durumlarını dikkate alarak uygun makineye ata.
Ürünün montaj öncesindeki işleri bitene kadar devam et.
Seçilen ürünün montaj ve sonrasındaki işleri (işlem süreleri eşit olduğu için) sıradan seç.
Seçilen işi işlem göreceği aşamalarda montaj öncesi mevcut ürüne ait en geç biten işten erken başlamamak koşuluyla, işin önceki aşamalardaki bitiş süresini ve makinelerin müsait olma durumlarını da dikkate alarak uygun makineye ata.
Ürünün montaj ve sonrasındaki işleri bitene kadar devam et.
Ürünler bitene kadar devam et.

FFS3J5 örnek problemi üzerinde bu kural aşağıdaki gibi çalışmaktadır.

A ürünü toplam işlem süresi : (13+8+16+20+13)+(42+42) = 154
B ürünü toplam işlem süresi : (5+12+6)+(28+28) = 79
Toplam işlem sürelerine göre ürünler A-B şeklinde sıralanacak.
A ürüne ait montaj öncesi işler:
1. işin işlem süreleri toplamı : 13
2. işin işlem süreleri toplamı : 8+16 = 24
3. işin işlem süreleri toplamı : 20+13 = 33
İşler 3-2-1 şeklinde sıralanacak.

Montaj ve sonrasında işlem süreleri eşit olduğu için işler 1-2 şeklinde sıralanacak.
B ürününe ait montaj öncesi işler:
4. işin işlem süreleri toplamı : 5
5. işin işlem süreleri toplamı : 12+6 = 18
İşler 5-4 şeklinde sıralanacak.
Montaj ve sonrasında işlem süreleri eşit olduğu için işler 3-4 şeklide sıralanacak.
Montaj öncesi aşamalarda sıralama : 3-2-1-5-4
Montaj ve sonrası aşamalarda sıralama : 1-2-3-4

Bu sıralamalara ve problemin diğer bilgilerine göre (makine uygunluk kısıtları, sıra bağımlı hazırlık zamanları ve öncelik kısıtları) işler Gannt şemasında Şekil 21'deki gibi sıralanmaktadır.



Şekil 21: EUİZ Kuralına Göre FFS3J5 Probleminin Gannt Şeması

3.5.1.3 Uyarlanmış Johnson Kuralı

Johnson kuralı 2 aşamalı akış tipi çizelgeleme problemleri için optimum çözüm veren bir algoritmadır. İşlerin iki aşamadaki işlem süreleri karşılaştırılır ve en kısa işlem süresinden başlanarak işler sıralanır. En kısa işlem süresi eğer ilk aşamada ise bu iş sıralamada öne, ikinci aşamada ise sona yerleştirilerek ana sıralama elde edilir. Bu tezin uygulama problemi de montaj öncesi ile montaj ve sonrası şeklinde 2 aşamalı olarak ele alınıp Johnson kuralı uyarlanarak uygulanabilir. Daha etkin bir sıralama için sıralamada işlem süreleri değil, bir ürüne ait montaj öncesi toplam işlem sürelerinin, o ürünün montaj öncesi ile montaj ve sonrasındaki toplam işlem süresine oranı kullanılmıştır. Uyarlanmış Johnson Kuralı (UJK) aşağıdaki gibi işlemektedir (Daha detaylı hali için bkz: EK-2).

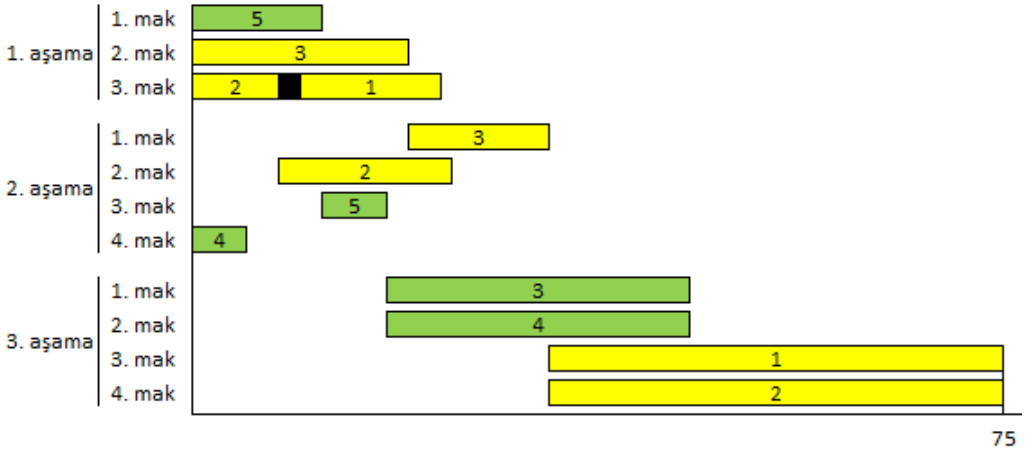
Her bir ürünün tüm aşamalardaki işlerinin işlem sürelerini topla. Montaj öncesindeki işlerinin toplam süresini montaj öncesi ve sonrası işlerin toplam süresine oranla.
Ürünleri bu oranlara göre **artan** şekilde sırala.

Sıradaki ilk ürünü seç.
Seçilen ürünün montaj öncesindeki işleri işlem süreleri toplamına göre **azalan** şekilde sırala.
Sıradaki ilk işi seç.
Seçilen işi işlem göreceği aşamalarda işin önceki aşamalardaki bitiş süresini ve makinelerin müsait olma durumlarını dikkate alarak uygun makineye ata.
Ürünün montaj öncesindeki işleri bitene kadar devam et.
Seçilen ürünün montaj ve sonrasındaki işleri (işlem süreleri eşit olduğu için) sıradan seç.
Seçilen işi işlem göreceği aşamalarda montaj öncesi mevcut ürüne ait en geç biten işten erken başlamamak koşuluyla, işin önceki aşamalardaki bitiş süresini ve makinelerin müsait olma durumlarını da dikkate alarak uygun makineye ata.
Ürünün montaj ve sonrasındaki işleri bitene kadar devam et.
Ürünler bitene kadar devam et.

FFS3J5 örnek problemi üzerinde bu kural aşağıdaki gibi çalışmaktadır.

A ürünü toplam işlem süresi : (13+8+16+20+13)+(42+42) = 154
B ürünü toplam işlem süresi : (5+12+6)+(28+28) = 79
Toplam işlem sürelerine göre ürünler B-A şeklinde sıralanacak.
B ürününe ait montaj öncesi işler:
4. işin işlem süreleri toplamı : 5
5. işin işlem süreleri toplamı : 12+6 = 18
İşler 5-4 şeklinde sıralanacak.
Montaj ve sonrasında işlem süreleri eşit olduğu için işler 3-4 şeklide sıralanacak.
A ürününe ait montaj öncesi işler:
1. işin işlem süreleri toplamı : 13
2. işin işlem süreleri toplamı : 8+16 = 24
3. işin işlem süreleri toplamı : 20+13 = 33
İşler 3-2-1 şeklinde sıralanacak.
Montaj ve sonrasında işlem süreleri eşit olduğu için işler 1-2 şeklinde sıralanacak.
Montaj öncesi aşamalarda sıralama : 5-4-3-2-1
Montaj ve sonrası aşamalarda sıralama : 3-4-1-2

Bu sıralamalara ve problemin diğer bilgilerine göre (makine uygunluk kısıtları, sıra bağımlı hazırlık zamanları ve öncelik kısıtları) işler Gannt şemasında Şekil 22'deki gibi sıralanmaktadır.



Şekil 22: UJK Sezgiselineе Göre FFS3J5 Probleminin Gannt Şeması

3.5.1.4 En Az Boşluk Kuralı

MS (Minimum Slack) olarak da adlandırılan En Az Boşluk Kuralı (EABK) gecikmeyi azaltmayı amaçlar. Teslim tarihi ile işin toplam süresi arasındaki fark işin boşluğudur, bu kurala göre de boşluğu en az olan işten başlayarak sıralama yapılır. Bu yöntem aşağıdaki gibi işlemektedir (Daha detaylı hali için bkz: EK-2).

> Montaj aşamasına giren her bir işin teslim tarihi ile o işin montaj ve sonrasındaki toplam işlem süresi ve o işin ait olduğu ürünün montaj öncesi işlerinin toplam işlem süresi toplamı arasındaki farkı (boşluk) hesapla.
> İşleri boşluklara göre **artan** şekilde sırala.
>> Sıradaki ilk işi seç.
>> Seçilen işin ait olduğu ürünün montaj öncesi işleri atanmadıysa montaj öncesindeki işleri işlem süreleri toplamına göre **azalan** şekilde sırala.
>>> Sıradaki ilk işi seç.
>>> Seçilen işi işlem göreceği aşamalarda işin önceki aşamalardaki bitiş süresini ve makinelerin müsait olma durumlarını dikkate alarak uygun makineye ata.
>>> Ürünün montaj öncesindeki işleri bitene kadar devam et.
>> Seçilen montaj aşamasına giren işi işlem göreceği aşamalarda montaj öncesi mevcut ürüne ait en geç biten işten erken başlamamak koşuluyla, işin önceki aşamalardaki bitiş süresini ve makinelerin müsait olma durumlarını da dikkate alarak uygun makineye ata.
>> İşler bitene kadar devam et.

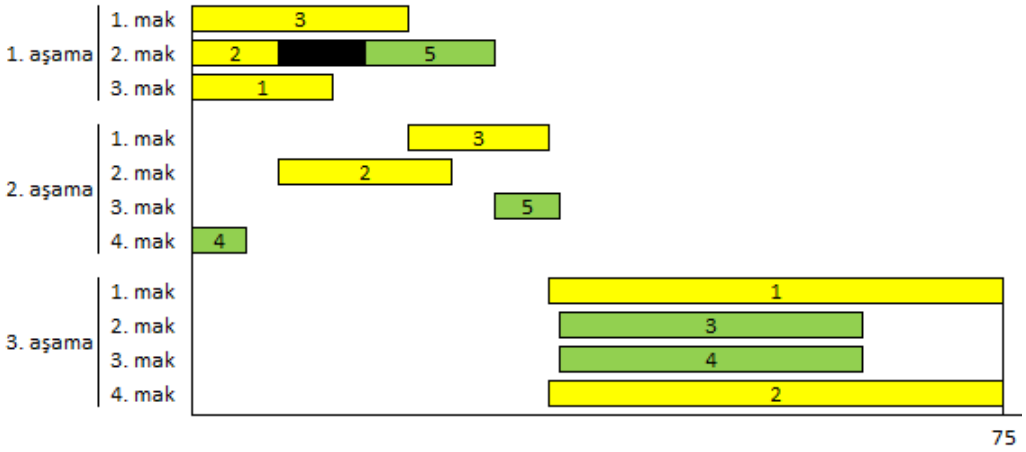
FFS3J5 örnek problemi üzerinde bu kural aşağıdaki gibi çalışmaktadır.

Montaj ve sonrasında 1. işin boşluğu = 1. işin teslim tarihi – (1. işin montaj ve sonrasındaki toplam işlem süresi + A ürününe ait işlerin montaj öncesi toplam işlem süreleri) : 56 – [42+(13+8+16+20+13)] = -56
Montaj ve sonrasında 2. işin boşluğu (Bu işin teslim tarihi olmadığı için yüksek bir değer verilmiştir = 1000 – [42+(13+8+16+20+13)] = 888
Montaj ve sonrasında 3. işin boşluğu = 31 – [28+(5+12+6)] = -20
Montaj ve sonrasında 4. İşin boşluğu = 52 – [28+(5+12+6)] = 1
Boşluklarına göre işler 1-3-4-2 şeklinde sıralanacak.

1. işin ait olduğu a ürününe ait montaj öncesi işler:
    1. işin işlem süreleri toplamı : 13
    2. işin işlem süreleri toplamı : 8+16 = 24
    3. işin işlem süreleri toplamı : 20+13 = 33
    İşler 3-2-1 şeklinde sıralanacak.
3. işi ait olduğu a ürününe ait montaj öncesi işler:
    4. işin işlem süreleri toplamı : 5
    5. işin işlem süreleri toplamı : 12+6 = 18
    İşler 5-4 şeklinde sıralanacak.
4. işin ait olduğu a ürününe ait montaj öncesi işler daha önce atanmış.
2. işin ait olduğu a ürününe ait montaj öncesi işler daha önce atanmış.

Montaj öncesi aşamalarda sıralama : 3-2-1-5-4
Montaj ve sonrası aşamalarda sıralama : 1-3-4-2

Bu sıralamalara ve problemin diğer bilgilerine göre (makine uygunluk kısıtları, sıra bağımlı hazırlık zamanları ve öncelik kısıtları) işler Gannt şemasında Şekil 23'deki gibi sıralanmaktadır.



Şekil 23: EABK Sezgiseline Göre FFS3J5 Probleminin Gannt Şeması

3.5.2 Test Problemleri

Çizelgeleme probleminin çözümüne ilişkin farklı tekniklerin performanslarının karşılaştırılabilmesi için aynı problemler üzerinde test edilmeleri gerekir. Bu nedenle "karşılaştırma problemleri" (benchmark problems) çizelgeleme problemi için geliştirilen algoritmaların test edilip karşılaştırılabilmesi için ortak bir standart sunar. Karşılaştırma problemleri farklı boyutlarda ve zorluklarda oldukları için bir yöntemin yeteneğini ve sınırlarını belirlemek oldukça kolaydır (Jain and Meeran 1999, 402).

Sezgiseller birbirleriyle pek çok kritere göre karşılaştırılabilirler. Uygulamada, üç kriter önemlidir. İlki, elde edilen çözümün kalitesi; ikincisi, iyi bir çözüm üretmek için gereken bilgisayar zamanı miktarı; üçüncüsü, sezgiseli geliştirmek ve uygulamak için gereken zaman. Akademik çevrelerde sezgisellerin karşılaştırıldığı çalışmalar genellikle ilk iki kriteri kullanırlar. Fakat, endüstri ortamında üçüncü kriter, kritik bir öneme sahiptir. Endüstride bir sezgiseli geliştirme süresinin kısa olması önemlidir. Bu kriter, uygulamada kayan darboğaz sezgiseli gibi karmaşık ayrışma teknikleri yerine yerel arama sezgisellerinin genellikle daha popüler olmalarının da bir nedenidir (Pinedo, 2008, 550-551).

Özellikle montaj aşamasında işlerin farklılaşmasından ve problemin kendine has özelliklerinin (aşama atlama, partilere ayırma, sıra bağımlı hazırlık zamanları, makine uygunluk ve öncelik kısıtları) fazla olmasından dolayı bu tezin problemine benzer bir probleme literatürde rastlanmamış olmasından dolayı tezin kendi test problemleri oluşturulmuştur. 3 aşamalı (2 montaj öncesi, 1 montaj) ve 6 aşamalı (4 montaj öncesi, 2

montaj ve sonrası) iki ayrı makine ortamında 5, 15 ve 30 iş olan (montaj sonrasında 4, 10 ve 20 işe dönüşen) 6 problem oluşturulmuştur. Aşamalardaki makine sayısı 2-6 arasında rassal olarak belirlenmiştir. Montaj öncesi aşamalarda bir işin aşama atlayıp atlamayacağı, atlayacaksa hangi aşamayı atlayacağı yine rassal olarak belirlenmiştir. Standart olarak ilk aşamada sıra bağımlı hazırlık zamanları, ikinci aşamada ise makine uygunluk kısıtları olduğu varsayılmış, hazırlık zamanları aynı ürüne ait işler arasında 2 dk., farklı ürünlere ait işler arasında 8 dk. olarak belirlenmiştir. 2. aşamadaki makine uygunluk kısıtları, işlerin paralel makinelerden sadece birinde işlem göreceği şekilde tasarlanmış ve bu makine rassal olarak belirlenmiştir. İşlem süreleri montaj öncesinde 5-20 dk., montaj ve sonrasında ise 20-50 dk. arasında rassal olarak belirlenmiştir. Montaj ve sonrası her bir işin teslim tarihi olup olmayacağı rassal olarak belirlenmiştir. Teslim tarihi olan işlerin teslim tarihleri her bir işin montaj öncesi işlerinin toplam sürelerinin ortalaması ile montaj ve sonrası işlem süreleri toplamının rassal olarak belirlenen 0,5 ile 1,5 arasındaki katsayı ile çarpılmasıyla elde edilmiştir. Test problemleri iş ve aşama sayısına göre Tablo 22'deki gibi isimlendirilmiştir.

Tablo 22: Aşama ve İş Sayılarına Göre Test Problemlerinin İsimleri

	5 iş	15 iş	30 iş
3 aşama	FFS3J5	FFS3J15	FFS3J30
6 aşama	FFS6J5	FFS6J15	FFS6J30

Test problemlerinin tüm bilgileri (işlem süreleri, hazırlık süreleri, aşamalardaki makine sayıları vs.) EK-4'te yer almaktadır. 6 aşama 66 işlik (montaj sonrası 50 işe dönüşen) tezin uygulama problemi de FFS6J66 olarak adlandırılmış ve test problemleri formatında EK-4'te yer verilmiştir.

3.6 Bulgular

3.6.1 Yöntemlerin Sonuçlarının Karşılaştırılması

Test problemleri ile birlikte tezin uygulama problemi matematiksel modellerle ve sezgisel yöntemlerle çözülmüş ve performans kriterlerine göre sonuçları Tablo 23'te verilmiştir (Matematiksel ve sezgisel yöntemlerin her bir problemin çözümleri için bkz. EK-5).

Test problemleri için matematiksel modellere 1 günlük (86400 s.) süre sınırı konulmuştur. Küçük çaplı problemler (3 ve 6 aşamalı 5 işli problemler FFS3J5 ve FFS6J5) için matematiksel modeller kısa süre içinde optimum çözümü bulmuşsa da iş ve aşama sayısı arttıkça optimumu bulma süresi uzamış, daha da arttıkça 1 günlük süre optimum çözüm için yeterli gelmemiştir. Optimumun bulunamadığı durumlarda matematiksel modelin 1 günlük süre içinde ulaştığı en iyi sonuç verilmiştir.

Tablo 23: Matematiksel ve sezgizel yöntem sonuçlarının karşılaştırılması

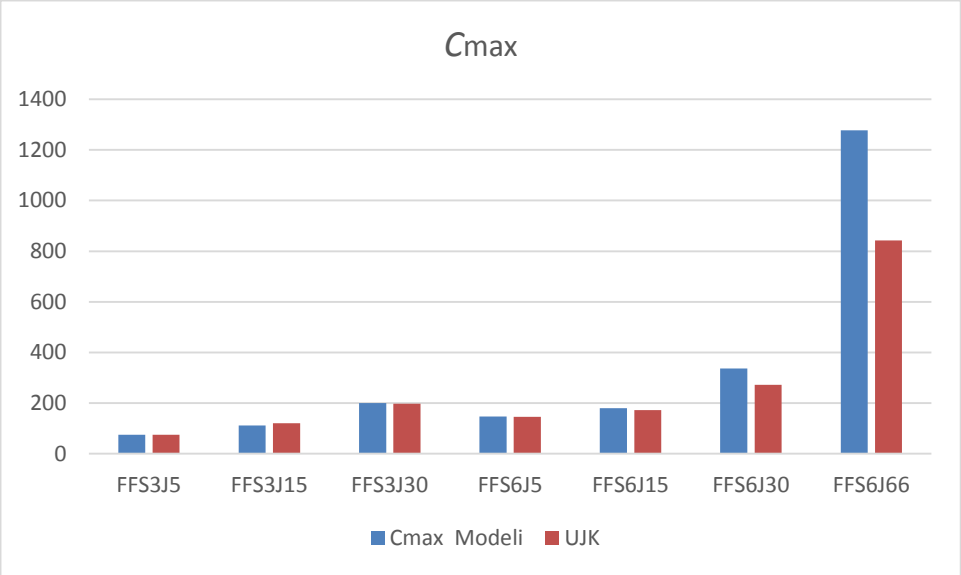
↓Problemler	Yöntemler→	Mat. C_{max}	Mat. $\sum T_j$	UJK	EABK	EKİZ	EUİZ
FFS3J5	Süre (sn.)	<1*	<1*	<1	<1	<1	<1
	C_{max}	75	75	75	75	85	75
	$\sum T_j$	70	34	34	60	44	60
FFS3J15	Süre (sn.)	288*	86400	<1	<1	<1	<1
	C_{max}	111	167	120	126	136	131
	$\sum T_j$	199	112	271	143	188	317
FFS3J30	Süre (sn.)	86400	86400	<1	<1	<1	<1
	C_{max}	200	244	197	252	219	214
	$\sum T_j$	1405	968	1015	1261	1117	1139
FFS6J5	Süre (sn.)	<1*	5*	<1	<1	<1	<1
	C_{max}	147	145	145	145	160	145
	$\sum T_j$	13	3	10	10	9	10
FFS6J15	Süre (sn.)	86400	75378*	<1	<1	<1	<1
	C_{max}	180	229	172	184	183	184
	$\sum T_j$	374	148	235	185	232	283
FFS6J30	Süre (sn.)	86400	86400	<1	<1	<1	<1
	C_{max}	337	410	272	312	287	278
	$\sum T_j$	1597	866	787	813	720	1088
FFS6J66	Süre (sn.)	259200	259200	<1	<1	<1	<1
	C_{max}	1278	1585,9	842,8	998,9	970,8	870,9
	$\sum T_j$	1468,8	2297,4	277,4	0	350,6	560,3

* Optimum çözüm

Sezgisel yöntemler problem büyüklüğünden bağımsız olarak 1 saniyenin altında sonuç vermişlerdir. Matematiksel olarak optimum çözümün bulunabildiği problemlerin (Bu sonuçlar Tablo 23'te * ile belirtilmiştir.) bazılarında sezgisel yöntemlerin de optimum çözüme ulaştığı görülmektedir. Ama sezgisel yöntemlerin asıl gücü, problem boyutu büyüdükçe ortaya çıkmaktadır. Problem boyutu büyüdükçe matematiksel modeller optimum çözümü makul sürelerde bulamamaktadır; sezgisel yöntemler ise matematiksel modellerin 1 günlük çalışmayla bulduğu optimum olmayan sonuçlardan çok daha iyi sonuçlar vermiştir.

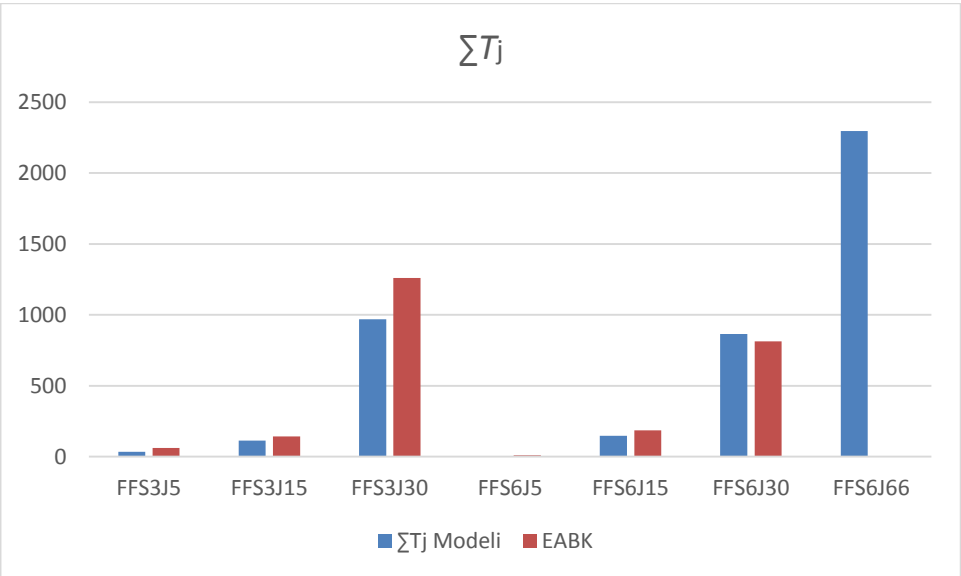
C_{max} performans kriterine göre C_{max} amaç fonksiyonlu matematiksel model ile UJK sezgiselinin problemlerdeki sonuçlarının karşılaştırması Şekil 24'te verilmiştir. UJK

sezgiseli bir problem haricinde (FFS3J15) C_{max} performans kriteri açısından tüm problemlerde en iyi değere ulaşmıştır.



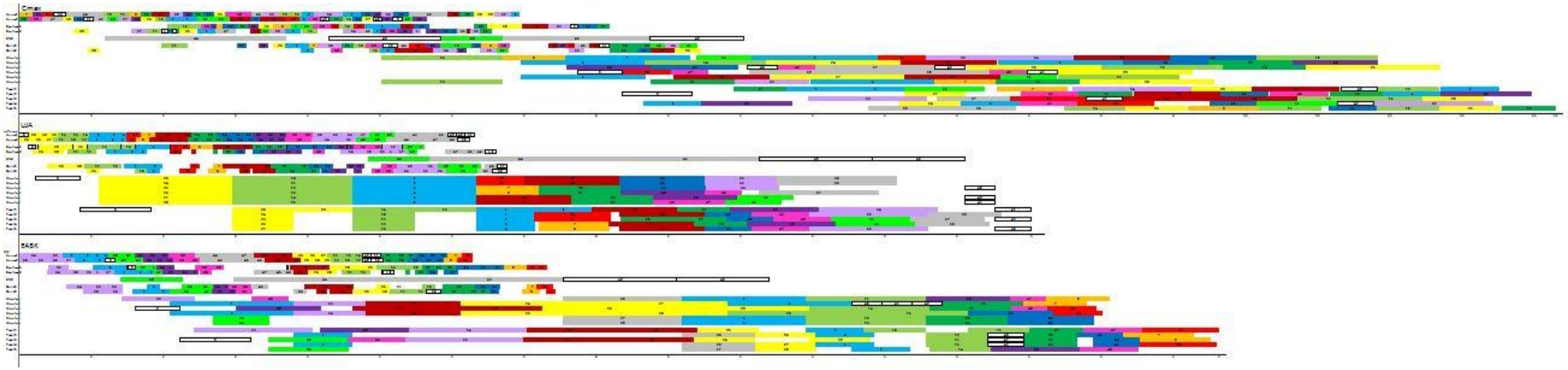
Şekil 24: C_{max} Kriterine Göre Yöntemlerin Sonuçlarının Karşılaştırması

$\sum T_j$ performans kriterine göre $\sum T_j$ amaç fonksiyonlu matematiksel model ile EABK sezgiselinin problemlerdeki sonuçlarının karşılaştırması Şekil 25'de verilmiştir. Küçük boyutlu problemlerde $\sum T_j$ amaç fonksiyonlu matematiksel model öne çıkmaktadır. Problem boyutu büyüdüğünde ise EABK sezgiseli $\sum T_j$ amaç fonksiyonlu matematiksel modelden daha iyi sonuçlar vermektedir. Özellikle FFS6J66 probleminde EABK sezgiseli toplam gecikmeyi sıfırlamayı başarmıştır.



Şekil 25: $\sum T_j$ Kriterine göre Yöntemlerin Sonuçlarının Karşılaştırması

Ele alınan uygulama problemi olan FFS6J66 probleminin C_{max} amaç fonksiyonlu matematiksel model, UJK ve EABK sezigisellerine göre çözümleri Şekil 26'da Gannt şeması olarak verilmiştir (Gannt şemalarının daha detaylı hali için bkz. EK-6). Her iki sezgiselin de işlerin maksimum tamamlanma zamanını ciddi oranda azalttığı şekilden görülmektedir. Bunun haricinde matematiksel model ile elde edilen çizelgede işler oldukça dağınıkken, sezgisellerle elde edilen çizelgelerde işler daha derli topludur; yani aynı ürüne ait işler birlikte çizelgelenmiştir, aralarına başka işler girmemiştir. UJK sezgiseli permutasyon tipinde çizelgeleme yaptığı için ilk aşamadaki sıralama tüm aşamalarda korunmaktadır. EABK sezgiselinde ise öncelik en az boşluğu olan işi atamak olduğundan montaj öncesi aşamalarda permutasyon tipi olsa da, montaj ve sonrasında bu kural bozulmaktadır.



Şekil 26: FFS6J66 Probleminin Matematiksel Model (C_{max}), UJK ve EABK yöntemlerine göre çözümlerinin Gannt Şemaları

Not: Gannt şemalarının daha detaylı hali için bkz. EK-6

3.6.2 Yöntemlerin Sapma Yüzdelerinin Karşılaştırılması

Yöntemlerin ne kadar iyi sonuca ulaşabildiğini daha net görebilmek için her problemde her bir performans kriterinde ulaşılan en iyi değeri 0'a eşitleyip, diğerlerinin o değerden sapma yüzdesini gösteren bir tablo oluşturulmuştur (Tablo 24). Her performans kriteri için sapma yüzdeleri aşağıdaki formülle hesaplanmıştır.

$$Sapma\ Yüzdesi = \frac{Yöntemin\ elde\ ettiği\ değer - Bilinen\ en\ iyi\ değer}{Bilinen\ en\ iyi\ değer} \times 100$$

Bu tablo sayesinde hangi problemlerde hangi yöntemlerin hangi performans kriterleri açısından en iyi değerlere ulaştığı, diğer yöntemlerin en iyi değerden ne oranda saptığı daha net görülmektedir.

Tablo 24: Problemlere göre optimum ya da bilinen en iyi değerden sapma yüzdeleri

Yöntemler→ ↓Problemler		Mat. C_{max}	Mat. $\sum T_j$	UJK	EABK	EKİZ	EUİZ
FFS3J5	C_{max}	0,0	0,0	0,0	0,0	13,3	0,0
	$\sum T_j$	105,9	0,0	0,0	76,5	29,4	76,5
FFS3J15	C_{max}	0,0	50,5	8,1	13,5	22,5	18,0
	$\sum T_j$	77,7	0,0	142,0	27,7	67,9	183,0
FFS3J30	C_{max}	1,5	23,9	0,0	27,9	11,2	8,6
	$\sum T_j$	45,1	0,0	4,9	30,3	15,4	17,7
FFS6J5	C_{max}	1,4	0,0	0,0	0,0	10,3	0,0
	$\sum T_j$	333,3	0,0	233,3	233,3	200,0	233,3
FFS6J15	C_{max}	4,7	33,1	0,0	7,0	6,4	7,0
	$\sum T_j$	152,7	0,0	58,8	25,0	56,8	91,2
FFS6J30	C_{max}	23,9	50,7	0,0	14,7	5,5	2,2
	$\sum T_j$	121,8	20,3	9,3	12,9	0,0	51,1
FFS6J66	C_{max}	51,6	88,2	0,0	18,5	15,2	3,3
	$\sum T_j$	146780,0	229640,0	27640,0	0,0	34960,0	55930,0

FFS6J66 probleminde EABK sezgiseli $\sum T_j$ performans kriterlerini sıfırladığı için, 0 değerinden sapma yüzdeleri de hesaplanamayacağı için, sapma yüzdelerinin hesaplanmasında bu değer 1 olarak alınmıştır.

FFS3J5 probleminde EKİZ haricinde tüm yöntemler en iyi C_{max} değerine ulaşmıştır. $\sum T_j$ performans kriterinde ise $\sum T_j$ amaç fonksiyonlu matematiksel model en iyi değere ulaşmıştır.

FFS3J15 probleminde C_{max} amaç fonksiyonlu matematiksel model C_{max} kriteri açısından, $\sum T_j$ amaç fonksiyonlu matematiksel model de $\sum T_j$ performans kriteri açısından ise en iyi değere ulaşmıştır.

FFS3J30 probleminde C_{max} performans kriterleri açısından UJK, $\sum T_j$ performans kriterleri açısından ise $\sum T_j$ amaç fonksiyonlu matematiksel model en iyi değere ulaşmıştır.

FFS6J5 probleminde C_{max} performans kriterleri açısından UJK, $\sum T_j$ performans kriterleri açısından ise $\sum T_j$ amaç fonksiyonlu matematiksel model en iyi değere ulaşmıştır.

FFS6J15 probleminde C_{max} performans kriterleri açısından UJK, $\sum T_j$ performans kriterleri açısından ise $\sum T_j$ amaç fonksiyonlu matematiksel model en iyi değere ulaşmıştır.

FFS6J30 probleminde C_{max} performans kriterleri açısından UJK, $\sum T_j$ performans kriterleri açısından ise EKİZ kuralı iyi değere ulaşmıştır.

FFS6J66 probleminde EABK sezgiseli $\sum T_j$ performans kriterinde olabilecek en iyi değere ulaştığı için bu kriterler açısından diğer yöntemlerin sapma yüzde değerleri anlamsız hale gelmiştir. C_{max} performans kriteri açısından UJK sezgiseli en iyi değere ulaşmıştır.

Yöntemleri birbirleriyle daha iyi karşılaştırabilmek için aynı yöntemlere göre ve performans kriterlerine göre Tablo 24'deki değerleri düzenleyip yeni bir tablo oluşturmak gerekir. Bir performans kriteri açısından en iyi sonuç veren yöntemi tespit edebilmek için performans kriterlerine göre de bir tablo oluşturmak gerekmektedir.

Tablo 25 performans kriterlerine göre optimum ya da bilinen en iyi değerlerden sapma yüzdelerini göstermektedir. Bu tabloya göre bir performans kriterinde en iyi sonucu veren yöntem tespit edilebilir. Dolayısıyla amaç belli bir performans kriterini minimize etmek olduğunda o kriterde en iyi sonucu veren yöntem kullanılabilir. FFS6J66 probleminde toplam gecikme kriterinde sapma yüzde değerleri çok yüksek olduğun için bu problem hariç ve dahil olmak üzere iki ayrı sapma ortalaması hesaplanmıştır.

C_{max} performans kriterinde en iyi sonucu veren yöntem UJK sezgiselidir. Bir problem hariç tüm problemlerde en iyi C_{max} değerini bulabilmiştir.

Tablo 25: Performans kriterlerine göre optimum ya da bilinen en iyi değerden sapma yüzdeleri

↓Kriterler	Yöntemler→ ↓Problemler	Mat. C_{max}	Mat. $\sum T_j$	UJK	EABK	EKİZ	EUİZ
C_{max}	FFS3J5	0,0	0,0	0,0	0,0	13,3	0,0
	FFS3J15	0,0	50,5	8,1	13,5	22,5	18,0
	FFS3J30	1,5	23,9	0,0	27,9	11,2	8,6
	FFS6J5	1,4	0,0	0,0	0,0	10,3	0,0
	FFS6J15	4,7	33,1	0,0	7,0	6,4	7,0
	FFS6J30	23,9	50,7	0,0	14,7	5,5	2,2
	FFS6J66	51,6	88,2	0,0	18,5	15,2	3,3
	Ort. Sapma*	5,2	26,4	1,4	10,5	11,5	6,0
	Ort. Sapma**	11,9	35,2	1,2	11,7	12,1	5,6
$\sum T_j$	FFS3J5	105,9	0,0	0,0	76,5	29,4	76,5
	FFS3J15	77,7	0,0	142,0	27,7	67,9	183,0
	FFS3J30	45,1	0,0	4,9	30,3	15,4	17,7
	FFS6J5	333,3	0,0	233,3	233,3	200,0	233,3
	FFS6J15	152,7	0,0	58,8	25,0	56,8	91,2
	FFS6J30	121,8	20,3	9,3	12,9	0,0	51,1
	FFS6J66	146780,0	229640,0	27640,0	0,0	34960,0	55930,0
	Ort. Sapma*	139,4	3,4	74,7	67,6	61,6	108,8
	Ort. Sapma**	21088,1	32808,6	4012,6	58,0	5047,1	8083,3

* İlk 6 problemin ortalaması ** 7 problemin ortalaması

$\sum T_j$ performans kriterinde en iyi sonucu veren yöntem FFS6J66 problemi ortalamaya dahil edilmediğinde $\sum T_j$ amaç fonksiyonlu matematiksel modeldir. Fakat FFS6J66 problemi ortalamaya dahil edildiğinde EABK kuralı yine açık bir farkla öne çıkmaktadır. Amaç toplam gecikmeyi minimize etmek olduğunda küçük boyutlu problemler için $\sum T_j$ amaç fonksiyonlu matematiksel model, büyük boyutlu problemler için EABK sezgiseli kullanılabilir.

UJK sezgiseli C_{max} performans kriterinde en iyi sonucu veren yöntemdir. Bir problem hariç, problemlerin tamamında en iyi C_{max} değerine ulaşmış, en iyi değere ulaşamadığı problemde de C_{max} amaç fonksiyonlu matematiksel modelin ulaştığı optimum sonuca en çok yaklaşan yöntem olmuştur. Hem küçük boyutlu, hem büyük boyutlu problemlerde C_{max} performans kriterinde kullanılabilir. FFS6J66 probleminde matematiksel modelin üç gün çalışarak elde ettiği sonuçtan %34 daha iyi bir C_{max} değeri elde etmiştir. En iyi sezgiselden ise %3,2 daha iyi sonuç elde etmiştir.

EABK sezgiseli toplam gecikme kriterini azaltması beklentisiyle kullanılmış, fakat küçük boyutlu problemlerde beklentiyi karşılayamamıştır. FFS6J66 probleminde ise diğer tüm yöntemlerden çok daha iyi sonuç vermiştir. Küçük boyutlu problemlerde iyi

sonuçlar elde edememesinin nedeni sezgiselin ve problemin yapısından kaynaklanmaktadır. Sezgisel, problemdeki montaj ve sonrasındaki işlerin teslim tarihleri ve işlem süresi arasındaki boşluğa göre işleri seçmektedir. Montaj ve sonrasında bir ürüne ait birden fazla iş olmakta ve bu işlerin teslim tarihleri birbirinden farklı olabilmektedir. Montaj ve sonrasındaki işlerin teslim tarihlerine göre ayrı zamanlarda sıralanması gerekirken bu işlerin ilkinin montaj işlemine başlayabilmesi için önce bu işin ait olduğu ürünün montaj öncesi bütün işlerinin bitmiş olması gerekmektedir. Bu durum küçük çaplı problemlerde yeterince zaman genişliği olmadığından EABK sezgiselinin iyi sonuçlar verememesine neden olmaktadır. Büyük problemlerde yeterince zaman genişliği olduğu için EABK'nın sonuçları iyileşmektedir. Ayrıca küçük çaplı problemlerde matematiksel modeller optimumu bulmakta veya yaklaşmaktayken, büyük çaplı problemlerde optimuma yaklaşamamaları EABK'nın gecikmeyle ilgili sonuçlarının göreceli olarak çok daha iyi olmasını sağlamaktadır.

3.6.3 Kesim İşlemi Sonrası Partilere Ayırma

Problemin yapısından kaynaklı En Az Boşluk Kuralının küçük boyutlu problemlerde beklenilen performansı verememesinden dolayı, problemin yapısında bir değişiklik yapılarak hem süreçte hem de gecikmelerde bir iyileşme sağlanabilir. Bir ürüne ait montaj ve sonrasında partilere ayrılmış farklı teslim tarihli birden fazla sayıda iş olması durumunda bu işlerin sıralaması farklı olabilmekte, fakat problemin yapısından dolayı bu işlerden herhangi birinin başlayabilmesi için o ürüne ait montaj öncesi işlerin tamamının bitmiş olması gerekmektedir. Dolayısıyle EABK'ndaki boşluğu az olan bir iş kendi ürününe ait boşluğu fazla olan bir işin de montaj öncesi işlerinin bitmesini beklemek zorunda kalmaktadır. Bunun önüne geçmek için kesim işlemi sonrası o ürüne ait farklı teslim tarihli işlere göre partilere ayırma işlemi yapılmıştır. Kesim sonrası partilere ayrılan işlerde, teslim tarihi erken olan işe ait partilerin işlem görmesi, kalan partilerin ise zamanı geldiğinde işlem görmesi küçük boyutlu problemler için de gecikmeleri azaltacaktır. Kesim işlemi sonrası partilere ayırmanın süreç içindeki yarı mamul miktarını arttırmayacağı öngörülmektedir; normalde montaj öncesinde bekleyen yarı mamuller, bu durumda kesim işlemi sonrasında bekleme yapacaktır. Kesim işlemi sonrası partilere ayırma durumunda kesimden sonra iş sayısı artmaktadır. Test problemleri ve tezin uygulama probleminin partilere ayrılmış haldeki bilgileri (iş sayısı, işlem süreleri, teslim tarihleri vs.) EK-4'te verilmiştir.

Problemin yapısında ve işlerin akışında bu değişiklik için geliştirilen Partilere Ayrılmış En Az Boşluk Kuralı (PEABK) aşağıdaki gibi işlemektedir (Daha detaylı hali için bkz: EK-2, C# dilinde kodlanmış hali için bkz: EK-3).

Montaj ve sonrasındaki her bir işin teslim tarihi ile o işin ait olduğu ürünün montaj öncesi ilk aşamadaki işleri ile montaj öncesi ilk aşama sonrası partilere ayrılmış işlerinin işlem süreleri toplamı arasındaki farkı (boşluk) hesapla.
Montaj ve sonrası işleri boşluklara göre **artan** şekilde sırala
Sıradaki ilk işi seç
Seçilen işin ait olduğu ürünün montaj öncesi ilk aşamadaki işleri atanmadıysa bu işleri işlem sürelerine göre azalan şekilde sırala
Sıradaki ilk işi seç
Seçilen işi makinelerin müsait olma durumuna göre ata
İşler bitene kadar devam et
Partilere ayrılmış işleri işlem sürelerine göre azalan şekilde sırala
Sıradaki ilk işi seç
Seçilen işi işlem göreceği aşamalarda işin önceki aşamalardaki bitiş süresini ve makinelerin müsait olma durumunu dikkate alarak uygun makineye ata.
Ürünün montaj öncesindeki işleri bitene kadar devam et
Seçilen montaj ve sonrası işi işlem göreceği aşamalarda montaj öncesi mevcut ürüne ait en geç biten işten erken başlamamak koşuluyla, işin önceki aşamalardaki bitiş süresini ve makinelerin müsait olma durumlarını da dikkate alarak uygun makineye ata.
İşler bitene kadar devam et.

FFS3J5 örnek probleminin ilk aşama sonrası partilere ayrılmış halinde işlem süreleri ve teslim tarihleri Tablo 26'da, makine uygunluk kısıtları ise Tablo 27'de görülmektedir.

Tablo 26: Partilere Ayrılmış FFS3J5 problemi işlem süreleri ve teslim tarihleri (dk.)

		Montaj öncesi aşamalar			Montaj		Teslim Tarihleri	
Ürün	İş	1. aşama		2. aşama		3. aşama	İş	Tarih(dk.)
A	1	13	1	-	1	42	1	56
	2	8	2	8				
	3	20	3	6,5				
			4	-	2	42	2	-
			5	8				
			6	6,5				
B	4	-	7	2,5	3	28	3	31
			8	3				
	5	12	9	2,5	4	28	4	52
			10	3				

Tablo 27: Partilere Ayrılmış FFS3J5 problemi 2. aşamadaki makine uygunluk kısıtları

İş	1	2	3	4	5	6	7	8	9	10
İşlem göreceği makine	-	2	1	-	2	1	4	3	4	3

Örnek problem üzerinde PEABK kuralı aşağıdaki gibi çalışmaktadır.

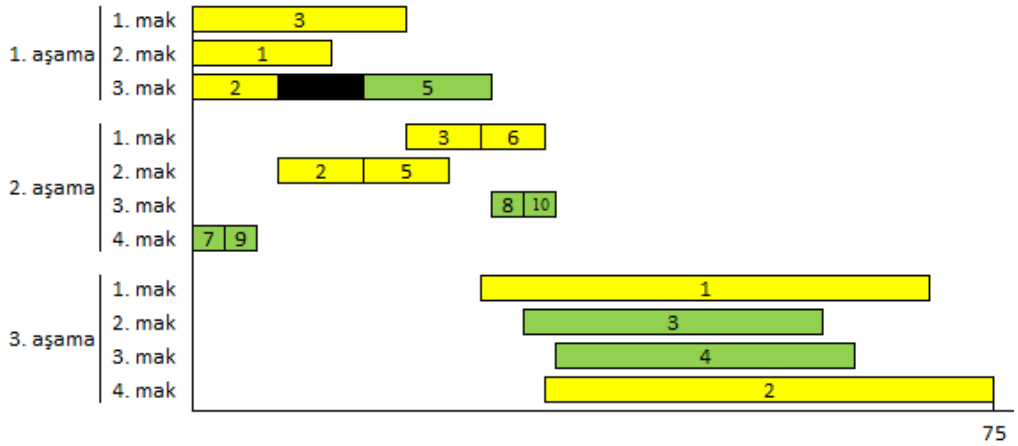
Montaj ve sonrasında 1. işin boşluğu = 1. işin teslim tarihi – (1. işin montaj ve sonrasındaki toplam işlem süresi + A ürününe ait işlerin ilk aşamadaki işlem süresi + ilk aşama sonrası montaj ve sonrasında 1. işe ait partideki toplam işlem süresi) : 56 – [42+(13+8+20)+(8+6,5)] = -41,5

Montaj ve sonrasında 2. işin boşluğu (Bu işin teslim tarihi olmadığı için yüksek bir değer verilmiştir = 1000 – [42+(13+8+20)+(8+6,5)] = 902,5
Montaj ve sonrasında 3. işin boşluğu = 31 – [28+(12)+(2,5+3)] = -14,5
Montaj ve sonrasında 4. İşin boşluğu = 52 – [28+(12)+(2,5+3)] = 6,5
Boşluklarına göre işler 1-3-4-2 şeklinde sıralanacak.

1. işin ait olduğu a ürününe ait montaj öncesi ilk aşamadaki işler:
    1. işin işlem süresi : 13
    2. işin işlem süresi : 8
    3. işin işlem süresi : 20
    İşler 3-1-2 şeklinde sıralanacak.
1. işe ait montaj öncesi ilk aşama sonrası partideki işler:
    2. işin işlem süresi : 8
    3. işin işlem süresi : 6,5
    İşler 2-3 şeklinde sıralanacak.
3. işe ait olduğu a ürününe ait montaj öncesi ilk aşamadaki işler:
    5. işin işlem süresi : 12
    İşler 5 şeklinde sıralanacak.
3. işe ait montaj öncesi ilk aşama sonrası partideki işler:
    7. işin işlem süresi : 2,5
    8. işin işlem süresi : 3
    İşler 8-7 şeklinde sıralanacak.
4. işin ait olduğu a ürününe ait montaj öncesi ilk aşamadaki işler daha önce atanmış.
4. işe ait montaj öncesi ilk aşama sonrası partideki işler:
    9. işin işlem süresi : 2,5
    10. işin işlem süresi : 3
    İşler 10-9 şeklinde sıralanacak.
2. işin ait olduğu a ürününe ait montaj öncesi işler daha önce atanmış.
2. işe ait montaj öncesi ilk aşama sonrası partideki işler:
    5. işin işlem süresi : 8
    6. işin işlem süresi : 6,5
    İşler 5-6 şeklinde sıralanacak.

Montaj öncesi ilk aşamada sıralama : 3-1-2-5
Montaj öncesi ilk aşama sonrası sıralama : 2-3-8-7-10-9-5-6
Montaj ve sonrası aşamalarda sıralama : 1-3-4-2

Bu sıralamalara ve problemin diğer bilgilerine göre (makine uygunluk kısıtları, sıra bağımlı hazırlık zamanları ve öncelik kısıtları) işler Gannt şemasında Şekil 27'deki gibi sıralanmaktadır.



Şekil 27: PEABK Sezgiselline Göre FFS3J5 Probleminin Gannt Şeması

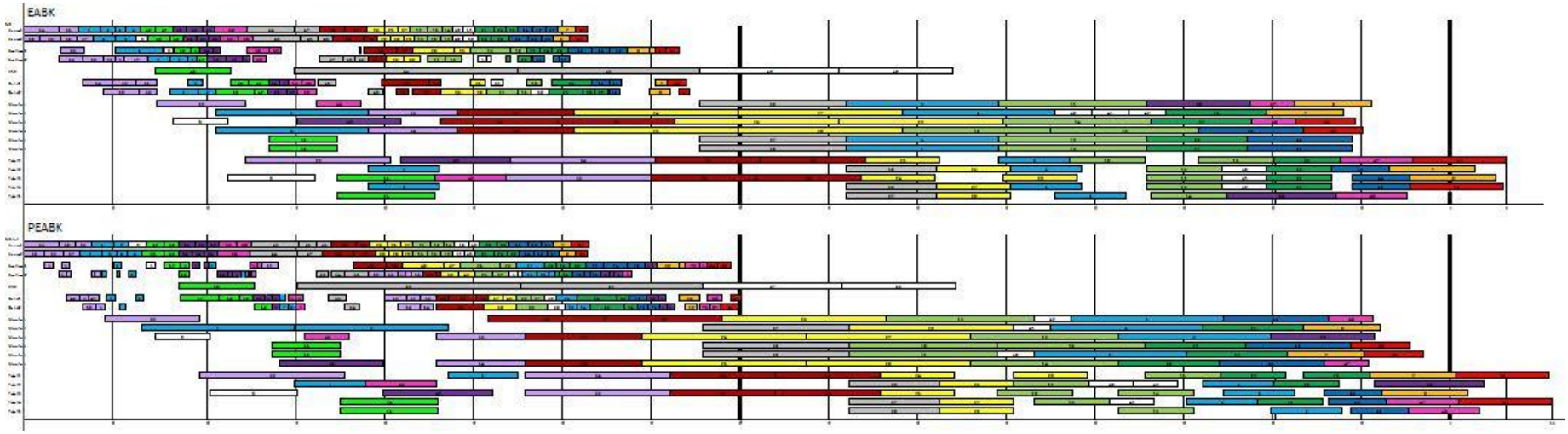
Test problemleri ve tezin uygulama probleminin ilk aşama sonrası partilere ayrılmış hallerinin PEABK sezgiseli ile çözüm sonuçları EK-5'd3 verilmiştir. PEABK sezgiselinin diğer yöntemlerle karşılaştırılması Tablo 28'de verilmiştir.

PEABK sezgizelinin EABK'dan her problemde gecikme ile ilgili kriterlerde daha iyi sonuçlar verdiği görülmektedir. PEABK en küçük boyutlu FFS3J5 problemi haricindeki problemlerin neredeyse tamamında $T_{\max}$ ve $\sum T_j$ kriterleri açısından en iyi değerlere ulaşmıştır. FFS6J66 probleminde (tezin uygulama problemi) EABK ile benzer şekilde gecikme ile ilgili tüm kriterleri sıfırlamayı başarmıştır. Bu haliyle PEABK küçük boyutlu problemlerde gecikmeleri azaltmak için iyi bir yöntemdir, problemin yapısındaki bu değişiklik ise tezin uygulama problemine konu olan işletme için bir süreç iyileştirmesi ve gecikmeleri azaltıcı bir alternatif olarak düşünülebilir.

Tablo 28: PEABK sezgiselinin diğer yöntemlerle sonuçlarının karşılaştırılması

↓Problemler	Yöntemler→	Mat. $C_{\max}$	Mat. $\sum T_j$	UJK	EABK	PEABK	EKİZ	EUİZ
FFS3J5	Süre (sn.)	<1*	<1*	<1	<1	<1	<1	<1
	$C_{\max}$	75	75	75	75	75	85	75
	$\sum T_j$	70	34	34	60	50,5	44	60
FFS3J15	Süre (sn.)	288*	86400	<1	<1	<1	<1	<1
	$C_{\max}$	111	167	120	126	120	136	131
	$\sum T_j$	199	112	271	143	104,5	188	317
FFS3J30	Süre (sn.)	86400	86400	<1	<1	<1	<1	<1
	$C_{\max}$	200	244	197	252	244	219	214
	$\sum T_j$	1405	968	1015	1261	1190,5	1117	1139
FFS6J5	Süre (sn.)	<1*	5*	<1	<1	<1	<1	<1
	$C_{\max}$	147	145	145	145	165	160	145
	$\sum T_j$	13	3	10	10	0	9	10
FFS6J15	Süre (sn.)	86400	75378*	<1	<1	<1	<1	<1
	$C_{\max}$	180	229	172	184	176,5	183	184
	$\sum T_j$	374	148	235	185	101,5	232	283
FFS6J30	Süre (sn.)	86400	86400	<1	<1	<1	<1	<1
	$C_{\max}$	337	410	272	312	324	287	278
	$\sum T_j$	1597	866	787	813	487	720	1088
FFS6J66	Süre (sn.)	259200	259200	<1	<1	<1	<1	<1
	$C_{\max}$	1278	1585,9	842,8	998,9	1027,4	970,8	870,9
	$\sum T_j$	1468,8	2297,4	277,4	0	0	350,6	560,3

FFS6J66 probleminin EABK ve PEABK sezgisellerine göre çözümleri Şekil 28'de Gannt şeması olarak verilmiştir (Gannt şemalarının daha detaylı hali için bkz. EK-6).



Şekil 28: FFS6J66 probleminin EABK ve PEABK yöntemlerine göre çözümlerinin Gannt Şemaları

Not: Gannt şemalarının daha detaylı hali için bkz. EK-6

SONUÇ

Bu tezde aşama atlama, partilere ayırma, sıra bağımlı hazırlık zamanları, makine uygunluk ve iş öncelik kısıtları bulunan bir esnek akış tipi çizelgeleme problemine çözüm aranmıştır. Ofis mobilyaları üreten bir işletmedeki uygulama probleminde 6 aşama ve 66 iş vardır.

Problem için öncelikle 0-1 karma tamsayılı programlama modeli geliştirilmiş, model GAMS dilinde yazılmış ve CPLEX çözücü ile çözüm aranmıştır. 2 saat, 1 gün ve 3 gün boyunca maksimum tamamlanma zamanı ve toplam gecikmeyi minimize etmeye çalışan iki ayrı amaç fonksiyonu için çalıştırılmış fakat optimum çözüm bulunamamıştır. Problem makine ve iş sayısının az olduğu durumlarda bile *NP*-zor bir problemdir. Bu nedenle optimum çözüm garanti etmeyen fakat kısa sürede kabul edilebilir sonuçlar veren sezgisel yöntemlere başvurulmuştur.

İki aşamalı akış tipi çizelgeleme problemlerinde optimum sonuç veren Johnson Kuralı modifiye edilerek ve tezin uygulama problemine uyarlanarak (UJK) kullanılmıştır. İşlerin teslim tarihleri ile toplam işlem süresi arasındaki boşluklara göre işleri sıralayarak gecikmeleri azaltmaya çalışan En Az Boşluk Kuralı (EABK) da uyarlanarak kullanılmıştır. Bu iki kural ile karşılaştırma yapabilmek için En Kısa İşlem Zamanı (EKİZ) ve En Uzun İşlem Zamanı (EUİZ) kuralları da tezin problemine uyarlanmıştır.

UJK sezgiseli C_{max} performans kriteri açısından hem matematiksel hem de diğer sezgisel yöntemlerden daha iyi sonuçlar elde etmiştir. UJK sezgiseli tezin uygulama problemi açısından C_{max} amaç fonksiyonlu matematiksel modelin 3 gün çalışarak elde ettiği değerden (1278) %34 daha düşük bir değeri (842,8) 1 saniyeden daha kısa bir sürede elde etmektedir.

EABK sezgiseli, küçük boyutlu problemlerde iyi değerlere ulaşamamış olmasına rağmen, problem boyutu büyüdükçe toplam gecikme kriterinde daha iyi sonuçlar vermektedir. Tezin uygulama probleminde ise toplam gecikmeyi sıfırlayarak diğer hiçbir yöntemin elde edemediği bir sonuca ulaşmıştır. EABK'nın küçük boyutlu problemlerde iyi sonuçlar verememesinden dolayı, problemde ilk aşama sonrası partilere ayırma uygulanmış ve buna uygun Partilere Ayrılmış En Az Boşluk Kuralı (PEABK) geliştirilmiştir. PEABK, küçük boyutlu problemlerde de, büyük boyutlu problemlerde de gecikmeyle ilgili kriterlerde daha iyi sonuçlara ulaşmıştır.

Çizelgeleme ile ilgili literatürdeki çalışmalar, gerçek hayat problemlerine çözüm aramaktan çok belli problem tiplerine daha iyi veya daha hızlı çözüm getirecek yöntemler geliştirmek üzerine yoğunlaşmıştır. Bu tezde ele alınan uygulama problemi; aşama atlama, partilere ayırma, sıra bağımlı hazırlık zamanları, makine uygunluk ve iş öncelik kısıtları olan bir esnek akış tipi çizelgeleme problemidir. Tez probleminin bir gerçek hayat problemi olması ve bu problemi çok kısa sürede çözerek çizelgeleme ile ilgili performans kriterleri açısından tatminkar sonuçlar ortaya çıkaran sezgisel yöntemlerin probleme uyarlanıp kullanılmış olması tezin literatüre ve uygulamaya katkılarıdır.

Tezin uygulamasının yapıldığı işletmeye ilişkin de bazı öneriler yerine getirildiği taktirde çizelgeleme performans kriterleri açısından daha iyi sonuçlar almak mümkün olabilir. Uygulama probleminde 2. aşama olan bantlama aşamasında makineler 0,4 mm. ve 2 mm. kalınlığında bantlama yapmak üzere ayrılmıştır. Makinelerde daha uzun bir hazırlık süresine neden olsa da bant kalınlığı değişimine izin verecek şekilde her iki makinede de hem 0,4 mm. hem de 2 mm. kalınlığında bant kullanılması durumu makinelerdeki iş yükünün dengelenmesine yardımcı olacaktır. Bu durum probleme yansıtılıp bir fayda sağlayıp sağlamayacağı incelenebilir.

İşletmede mevcut durumda kesme işleminde bir ürüne ait bileşenlerden her biri ayrı sunta plakalarından kesilerek oluşturulmaktadır. Farklı bileşenlerin ebatları da farklı olduğundan her plakadan farklı sayıda bileşen elde edilmektedir. İşletmede bir ürüne ait parti büyüklükleri de bu plakalardan çıkan bileşen sayılarının mümkün olduğunca ortak katı olacak şekilde zaman içinde belirlenmiş miktarlardır. Bu şekilde parti büyüklüğü belirlemek yerine, ürüne ait aynı kalınlıkta suntadan üretilen farklı bileşenlerin aynı

plakadan kesilmesi ile daha düşük parti büyüklüğü elde edilebilir. Bunun tercih edilmemesinin nedeni, işletmede parti üretimi mantığının olması ve farklı ölçülerde bileşenlerin tek plakadan kesim işleminin zahmetli olmasıdır. İşletme bu zahmetten kaçınmak isterken sipariş adedinden fazla üretip gereksiz stok yüküne katlanmak zorunda kalmaktadır. Süreç içinde stoğa üretilen ürün miktarından dolayı belki yetişmesi gereken diğer işlerini ertelemek zorunda kalmaktadır.

Mevcut sezgisel yöntemlerde bir ürüne ait işlerin sıralamasında EKİZ yöntemi haricinde toplam işlem süresi en uzun işi ilk atama yoluna gidilmektedir. Bazı işler bazı aşamaları atladığından işlerin sıralamasını belirlerken işlem süresi ile birlikte işin işlem göreceği aşama sayısı da dikkate alınabilir. Bunun dışında yöntemlerin mevcut halinde bir iş bir aşamada makineler arasında seçim yapacakken en erken müsait olan makineyi tercih etmektedir. Bu da makine eğer çok erken müsait oluyorsa aradaki zamanda makinenin boş kalmasına neden olmaktadır. En erken müsait olan makine yerine iş, başlayabileceği zamandan daha erken müsait olan makineler içinde en geç müsait olan makineye atanırsa, erken müsait olan makinenin daha az boş kalmasına imkan sağlanabilir. Yine yöntemlerin mevcut halinde makineye yeni atanacak iş, makineye son atanan işten sonra atanmaktadır. Fakat atanacak işin kısıtlarını sağlamak koşuluyla makineye önceki atanmış işler arasındaki boşluklara atama yapılabilirse performans kriterlerini daha da iyileştirmek mümkün olabilir.

Çizelgeleme problemlerinin çözümü için kullanılan matematiksel, sezgisel ve metasezgisel yöntemlerin birbirlerine göre avantajlı ve dezavantajlı yönleri vardır: Matematiksel modeller bir problemi tanımlamak, problemin çerçevesini çizmek için kullanılmaktadır. Bunun yanında küçük boyutlu problemlerde optimum sonucu kısa sürede bulması matematiksel modelleri kullanışlı kılmaktadır. Problem boyutu büyüdükçe de gerekirse uzun süreler çalıştırılarak problemin optimum çözümünü bulmak, geliştirilen sezgisellerin en iyi değere ne kadar yaklaşabildiklerini tespit edebilmek için faydalıdır. Bununla birlikte büyük boyutlu problemlerde uzun sürelerde dahi optimum çözüme ulaşamaması matematiksel modellerin dezavantajıdır. Ayrıca optimum çözüm kimi zaman uygulanabilir olmayabilir; bir performans kriterini minimize etmek uğruna ürünler çok dağınık şekilde çizelgelenmiş olabilir; bu da optimum olsa dahi bir çizelgenin uygulanabilirliğini zorlaştıran bir unsurdur.

Sezgisel modeller problem boyutundan bağımsız olarak çok kısa sürede sonuç vermektedirler. Mevcut bir kural probleme uyarlandığında veya probleme özgü yeni bir kural geliştirildiğinde daha uygulanabilir çizelgeler ortaya çıkarma imkanı vardır. Ayrıca bir sezgisel yöntemin geliştirilmesi ve kodlanması (metasezgisel yöntemlere göre) daha kolaydır. Sezgisel yöntemlerin optimum sonucu garanti etmemesi, problem boyutu büyüdükçe optimum sonuçtan uzaklaşabilmesi gibi dezavantajları da vardır.

Metasezgisel yöntemler de problem boyutundan bağımsız olarak kısa sürede sonuç vermektedir. Genel olarak zeki bir şekilde arama yaptıklarından sezgisel yöntemlerden daha iyi sonuç verme potansiyelleri vardır. Sezgisel yöntemler gibi optimum sonucu garanti edememesi, sezgisel yöntemlerden daha karmaşık olduklarından onlar kadar kısa sürede sonuç vermemesi ve yöntemi geliştirmenin ve kodlamanın daha karmaşık olması da dezavantajlarıdır.

Yöntemlerin bu avantaj ve dezavantajları göz önünde bulundurulduğunda, gerçek hayatta ürün çeşidi ve çizelgelenecek iş sayısı fazla olan bir işletmede matematiksel modellerin kullanılabilirliği zayıftır. Özellikle her an yeni bir siparişin gelme olasılığı bulunan dinamik bir ortamda, her an yeniden çizelgeleme ihtiyacı olabilir. Böyle bir durumda da çok kısa sürede sonuç verecek yöntemlerin kullanılması gerekmektedir. Bu şartlar altında da özellikle geliştirme ve kodlama kolaylığının yanında çözüm üretme hızından dolayı sezgisel yöntemler öne çıkmaktadır. İleride metasezgisel yöntemlerin de bu probleme uygulanıp, sezgisel yöntemlerle karşılaştırılması faydalı olacaktır.

EKLER

EK-1: GAMS DİLİNDE MATEMATİKSEL MODELİN YAZILIŞI

```
  SETS
        S               tüm aşamalar                            / 1*6   /
        J               tüm işler                               / 1*116  /
        M               makineler                               / 1*6   /
        FRS(S)          montaj öncesi ilk aşama                 / 1     /
        LST(S)          montaj öncesi son aşama                 / 4     /
        FIRST(S)        montaj aşaması                          / 5     /
        LAST(S)         montaj sonrası son aşama                / 6     /


        rota1(S,J)       hangi s aşamalarında hangi j işleri var
/
1.1     ,       1.2     ,       1.3     …
…
2.1     ,       2.2     ,       2.3     …
…
…
6.67*116
/
        morota1(S,J)       montaj öncesi hangi s aşamalarında hangi j işleri var
/
1.1     ,       1.2     ,       1.3     …
…
2.1     ,       2.2     ,       2.3     …
…
4.61    ,       4.62    ,                       4.64    ,       4.65
/
        msrota1(S,J)       montaj ve sonrası hangi s aşamalarında hangi j işleri var
/
5.67*116

6.67*116
/
        rota2(S,M)      hangi s aşamalarında hangi m makineleri var
/
1.1     ,       1.2
2.1     ,       2.2
3.1
4.1     ,       4.2
5.1*6
6.1*5
/
        rota(S,J,M)     s aşamasındaki j işi m makinesinde - işlerin rotaları
/
1.1.1   ,       1.1.2   ,       1.2.1   ,       1.2.2   …
…
1.65.1  ,       1.65.2  ,       1.66.1  ,       1.66.2
…
…
6.67*116.1*5
/
        morota(S,J,M)     montaj öncesi s aşamasındaki j işi m makinesinde - işlerin
rotaları
/
```

```
1.1.1    ,       1.1.2   ,       1.2.1   ,       1.2.2   ,       …
…
1.65.1  ,       1.65.2   ,       1.66.1   ,       1.66.2
…
…
4.64.1  ,       4.64.2  ,       4.65.1  ,       4.65.2
/
         msrota(S,J,M)     s aşamasındaki j işi m makinesinde - işlerin rotaları
/
5.67*116.1*6
6.67*116.1*5
/
         mont(J,J)     (montaj aşamasında d indisi ile gösterilen işin montaj öncesinde
bileşenleri hangi c indisleriydi)
/
67.1    ,     67.2     ,     67.3     ,     67.4     ,     67.5     ,      67.6
68.1    ,     68.2     ,     68.3     ,     68.4     ,     68.5     ,      68.6
…
…
116.61  ,     116.62   ,     116.63   ,     116.64   ,     116.65   ,     116.66
/

         duedate(J)
/
67*116
/
;

alias (J, I);

  parameters
         P(S,J,M)  işlem süreleri
/
1.1.1=15.2  ,    1.1.2=15.2  ,   1.2.1=9.7   ,   1.2.2=9.7   ,   1.3.1=9.8   ,
…
…
6.115*116.1*5=74.5
/
         SU(I,J,S)       montaj öncesi aşamalar için hazırlık süreleri
/
7.5.2=0.5   ,    7.13.2=0.5  ,   7.28.2=0.5  ,   7.33.2=0.5  ,   7.52.2=0.5  ,
7.53.2=0.5  ,   7.56.2=0.5  ,   7.64.2=0.5
…
…
61.5.2=0.5  ,    61.13.2=0.5  ,  61.28.2=0.5  ,  61.33.2=0.5  ,  61.52.2=0.5  ,
61.53.2=0.5  ,  61.56.2=0.5  ,  61.64.2=0.5
/
         DUE(J)            d işinin teslim zamanı
/
67=480   ,       68=960  ,       69=1440 ,       70=1440 ,       71=1440
…
…
113=1440,       114=1440,       115=960 ,       116=1440
/
;

SCALAR LARGE       çok büyük bir sayı    /200000/ ;
VARIABLES
         TOTTARD         toplam gecikme
         T(J)            h işinin gecikmesi
         CMAX            tamamlanma zamanı
         C(S,J,M)        j işinin s aşamasında m makinesinden ayrılma zamanı
         K(S,J)          j işinin s aşamasından ayrılma zamanı
         AS(J)           f işinin en erken montaja başlayabileceği zaman
         X(S,J,M)        j işi s aşamasında m makinesinde işlem görecekse 1 değilse 0
         Q(S,I,J,M)      m makinesinde j işi i işini takip ediyorsa 1 değilse 0
BINARY VARIABLES X, Q ;
POSITIVE VARIABLES T, TOTTARD, TOTCOMP, C, K, AS ;
EQUATIONS
         ISMAKINE(S,J)           12. bir iş bir aşamada sadece bir makinede işlem
görebilir
         TAKIP1(S,J,M)           14. bir işi en fazla bir iş takip edebilir
         TAKIP2(S,I,M)           16. bir iş en fazla bir işi takip edebilir
         TAKIP3(S,M)             18. bir aşamada makinedeki iş sayısı o makinede
sıralama sayısından bir fazladır
```

```
         ILKISLEM(S,J,M)         7. ilk makineden ayrılma zamanı en az o makinedeki
işlem süresi kadar olmalı
         MONTAJ(I,S,J)           22. montaj aşamasından ayrılma zamanı en az o işin
bileşenlerinin montaj öncesi son aşamadan ayrılma zamanlarından büyük olmalı
         ILKISLEMA(S,J,M)        .
         ISTAKIP(S,I,J,M)        8. bir makinede birbirini takip eden işler arasındaki
sürenin en az ilk işin işlem süresi kadar olması
         ISASAMA(S,J,M)          10. bir işin işlem göreceği iki aşama arasında geçen
süre en az o işin ilk aşamadaki işlem süresi kadar olmalı
         ONCEKIASAMA(S,J)        . aşama atlama durumunda kr-1j lerin doğru
hesaplanabilmesi için
         ISBITIS(S,J,M)          3. her işin her aşamadaki tamamlanma zamanı en az o
işin o aşamada işlem göreceği makineden ayrılma zamanı kadardır
         SONIS(S,J)              2. tamamlanma zamanını son biten işe eşitle
         ISGECIKME(S,J)          h işinin gecikmesi
         TOPGECIKME              toplam gecikme denklemi     ;

ISMAKINE(S,J)$(rota1(S,J)) ..
SUM(M, X(S,J,M)$(rota(S,J,M))) =E= 1 ;

TAKIP1(S,J,M)$(rota(S,J,M)) ..
X(S,J,M) - SUM(I, Q(S,I,J,M)$(ord(J)<>ord(I) and rota(S,I,M))) =G= 0 ;

TAKIP2(S,I,M)$(rota(S,I,M)) ..
X(S,I,M) - SUM(J, Q(S,I,J,M)$(ord(J)<>ord(I) and rota(S,J,M))) =G= 0  ;

TAKIP3(S,M)$(rota2(S,M)) ..
SUM(J, X(S,J,M)$(rota(S,J,M))) - SUM(I, SUM(J, Q(S,I,J,M)$(rota(S,J,M) and rota(S,I,M)
and ord(J)<>ord(I)))) =E= 1  ;

ILKISLEM(S,J,M)$(morota(S,J,M) and FRS(S)) ..
C(S,J,M) =G= P(S,J,M) * X(S,J,M) ;

MONTAJ(I,S,J)$(LST(S) and mont(I,J)) ..
AS(I) =G= K(S,J) ;

ILKISLEMA(S,J,M)$(rota2(S,M) and FIRST(S)) ..
C(S,J,M) =G= AS(J) + P(S,J,M) * X(S,J,M) ;

ISTAKIP(S,I,J,M)$(rota(S,J,M) and rota(S,I,M) and ord(J)<>ord(I)) ..
C(S,J,M) - C(S,I,M) + LARGE * (1 - Q(S,I,J,M)) =G= P(S,J,M) + Q(S,I,J,M) * SU(I,J,S) ;

ISASAMA(S,J,M)$(rota(S,J,M) and ord(S)>1) ..
C(S,J,M) - K(S-1,J) =G= P(S,J,M) - LARGE * (1 - X(S,J,M)) ;

ONCEKIASAMA(S,J)$(morota1(S,J) and ord(S)>1) ..
K(S,J) =G= K(S-1,J) ;

ISBITIS(S,J,M)$(rota(S,J,M)) ..
C(S,J,M) =L= K(S,J)  ;

SONIS(S,J)$(LAST(S) and msrota1(S,J)) ..
K(S,J) =L= CMAX  ;

ISGECIKME(S,J)$(LAST(S) and msrota1(S,J)) ..
K(S,J) - T(J) =L= DUE(J) ;

TOPGECIKME ..
TOTTARD =E= SUM(J, T(J)$(duedate(J))) ;

MODEL ffspwsupasmblycmax3 /ALL/ ;

Option MIP=CPLEX;

ffspwsupasmblycmax3.iterlim = 100000000;
ffspwsupasmblycmax3.reslim = 3600;
ffspwsupasmblycmax3.nodlim = 100000000;

Option limrow=3;

SOLVE ffspwsupasmblycmax3 USING MIP MINIMIZING CMAX ;

display C.l, X.l, Q.l ;
```

EK-2: SEZGİSEL SIRALAMA KURALLARI ALGORİTMALARI

İndisler, Kümeler ve Parametreler

a : ürün indisi
b : BAR sıralaması indisi
BAR_a : a ürününün montaj öncesi işlem süreleri toplamının montaj sonrası işlem süreleri toplamına oranı
d : PT_{aj} sıralaması indisi
DD_h : h işinin teslim tarihi
h : montaj ve sonrası için iş indisi
I_h : Montaj ve sonrasındaki h işine ait olan i işleri kümesi
i : ilk aşama sonrası partilere ayırma durumunda ilk aşama sonrası montaj öncesi iş indisi
H_a : a ürününe ait işler (partiler) (montaj sonrası)
j : Montaj öncesi için iş indisi
J_a : a ürününe ait işler (bileşenler) (montaj öncesi)
PA_{sh} : s aşamasında h işinin işlem süresi (aşamalarda makineler özdeş olduğu için tüm makinelerde aynı)
PL_{ri} : r aşamasında i işinin işlem süresi (aşamalarda makineler özdeş olduğu için tüm makinelerde aynı)
P_{rj} : r aşamasında j işinin işlem süresi (aşamalarda makineler özdeş olduğu için tüm makinelerde aynı)
PT_a : a ürününün toplam işlem süresi
PTA_a : a ürününün montaj ve sonrası aşamalardaki tüm partilerinin toplam işlem süresi
PT_{aj} : montaj öncesinde bir bileşenin tüm operasyonlarının işlem süreleri toplamı
$PTBA_a$: a ürününün tüm bileşenlerinin montaj öncesi aşamalardaki tüm operasyonlarının toplam işlem süresi
r : Montaj öncesi için aşama indisi
s : Montaj ve sonrası için aşama indisi
SL_h : h işinin boşluğu
$SU_{rMJrm}j$: r aşamasında m makinesindeki son işten (MJ_{rm}) yeni işe (j) geçerken hazırlık süresi

Değişkenler

CL_{rim} : r aşamasında i işinin m makinesinden ayrılma zamanı (Montaj öncesi ilk aşama sonrası işler için)
C_{rjm} : r aşamasında j işinin m makinesinden ayrılma zamanı (Montaj öncesi işler için)
F_{shm} : s aşamasında h işinin m makinesinden ayrılma zamanı (Montaj ve sonrası işler için)
MAA_{sm} : s aşamasındaki m makinesinin uygun olma zamanı (başta 0) (machine available)
MA_{rm} : r aşamasındaki m makinesinin uygun olma zamanı (başta 0) (machine available)
MJ_{rm} : r aşamasındaki m makinesindeki son iş (başta 0)
$PSTL_{rim}$: r aşamasında i işinin m makinesinde olası başlama zamanı
PST_{rjm} : r aşamasında j işinin m makinesinde olası başlama zamanı
STA_{shm} : s aşamasındaki h işinin m makinesinde operasyona başlama zamanı
STL_{rim} : r aşamasındaki i işinin m makinesinde operasyona başlama zamanı
ST_{rjm} : r aşamasındaki j işinin m makinesinde operasyona başlama zamanı
$TSTA_h$: h işinin geçici başlayabilme zamanı
TST_j : j işinin geçici başlayabilme zamanı
$TSTL_i$: i işinin geçici başlayabilme zamanı

En Kısa İşlem Zamanı Kuralı

Her bir ürün için montaj öncesi işler ile montaj ve sonrası işlerin işlem sürelerini toplayarak ürünün toplam işlem süresini hesapla.

$$PTBA_a = \sum_{j=1}^{J} \sum_{r=1}^{R} P_{rj} \quad ; \quad j \in J_a, \qquad \forall\, a$$

$$PTA_a = \sum_{h=1}^{H} \sum_{s=1}^{S} PA_{sh} \quad ; \quad h \in H_a, \qquad \forall\, a$$

$PT_a = PTBA_a + PTA_a \quad ; \quad \forall\, a$

Ürünleri PT_a değerlerine göre artan şekilde sırala.

Bu sıralamada ilk ürünü seç.

Seçilen ürünün montaj öncesi işlerinin toplam işlem sürelerini hesapla.

$$PT_{aj} = \sum_{r=1}^{R} P_{rj} \quad ; \quad \forall\, j$$

İşleri PT_{aj} değerlerine göre artan şekilde sırala.

Sıralamadaki ilk işi seç.

Seçilen işin atanacağı ilk aşamadan başla.

Makinenin müsait olma zamanına makinedeki son iş ile mevcut iş arasındaki hazırlık süresini de ekleyerek makinenin işe başlayabileceği zamanı belirle.

$PST_{rjm} = MA_{rm} + SU_{rMJ_{rm}j} \; ; \; \forall \; m$

PST_{rjm}'ler içinde en küçük olanı bulup j şinin r aşamasındaki operasyonunu m makinesine ata.

$PST_{rjm} >= TST_j$ ise

$ST_{rjm} = PST_{rjm}$

değilse

$ST_{rjm} = TST_j$

$C_{rjm} = ST_{rjm} + P_{rj}$
$TST_j = C_{rjm}$
$MA_{rm} = TST_j$
$MJ_{rm} = j$

İşin diğer aşamalardaki diğer operasyonları için tekrarla.

PT_{aj} sıralamasındaki diğer işler için tekrarla.

Seçilmiş ürünün montaj ve sonrası aşamalarını sıradan atamak için a ürününe ait TST_j ler içinde en yüksek olanı seç.

$TSTA_h = Max\left(TST_{J_a}\right) \quad ; \quad \forall\, H_a$

İlk işi seç (işlem süreleri birbirine eşit, eşit olmayan durumlar için işlem süresi uzun olan da seçilebilir)

Seçilen işin ilk aşamasından başla.

Min MAA_{sm}'yi bul.

$MAA_{sm} >= TSTA_h$ ise

$STA_{shm} = MAA_{sm}$

değilse

$STA_{shm} = TSTA_h$

$F_{shm} = STA_{shm} + PA_{sh}$
$TSTA_h = F_{shm}$
$MAA_{sm} = TSTA_h$

İşin diğer aşamalardaki diğer operasyonları için tekrarla.

Sıralamadaki diğer ürünler için tekrarla.

F_{shm} 'lerden en büyüğünü bul, bu $C_{\max}$ yani en son biten işin bitiş zamanıdır.
Tüm gecikmelerin toplamını hesapla; $\sum T_j$

En Uzun İşlem Zamanı Kuralı

Her bir ürün için montaj öncesi işler ile montaj ve sonrası işlerin işlem sürelerini toplayarak ürünün toplam işlem süresini hesapla.

$$PTBA_a = \sum_{j=1}^{J} \sum_{r=1}^{R} P_{rj} \quad ; \quad j \in J_a, \qquad \forall\, a$$

$$PTA_a = \sum_{h=1}^{H} \sum_{s=1}^{S} PA_{sh} \quad ; \quad h \in H_a, \qquad \forall\, a$$

$PT_a = PTBA_a + PTA_a \quad ; \quad \forall\, a$

Ürünleri PT_a değerlerine göre azalan şekilde sırala.

Bu sıralamada ilk ürünü seç.

Seçilen ürünün montaj öncesi işlerinin toplam işlem sürelerini hesapla.

$$PT_{aj} = \sum_{r=1}^{R} P_{rj} \quad ; \quad \forall\, j$$

İşleri PT_{aj} değerlerine göre azalan şekilde sırala.

Sıralamadaki ilk işi seç.

Seçilen işin atanacağı ilk aşamadan başla.

Makinenin müsait olma zamanına makinedeki son iş ile mevcut iş arasındaki hazırlık süresini de ekleyerek makinenin işe başlayabileceği zamanı belirle.

$PST_{rjm} = MA_{rm} + SU_{rMJ_{rm}j} \; ; \; \forall\; m$

PST_{rjm}'ler içinde en küçük olanı bulup j şinin r aşamasındaki operasyonunu m makinesine ata.

$PST_{rjm} >= TST_j$ ise

$ST_{rjm} = PST_{rjm}$

değilse

$ST_{rjm} = TST_j$

$C_{rjm} = ST_{rjm} + P_{rj}$
$TST_j = C_{rjm}$
$MA_{rm} = TST_j$
$MJ_{rm} = j$

İşin diğer aşamalardaki diğer operasyonları için tekrarla.

PT_{aj} sıralamasındaki diğer işler için tekrarla.

Seçilmiş ürünün montaj ve sonrası aşamalarını sıradan atamak için a ürününe ait TST_j ler içinde en yüksek olanı seç.

$$TSTA_h = Max(TST_{J_a}) \quad ; \quad \forall H_a$$

İlk işi seç (işlem süreleri birbirine eşit, eşit olmayan durumlar için uzun işlem süreli olan da seçilebilir)

Seçilen işin ilk aşamasından başla.

Min MAA_{sm}'yi bul.

$MAA_{sm} >= TSTA_h$ ise

$STA_{shm} = MAA_{sm}$

değilse

$STA_{shm} = TSTA_h$

$F_{shm} = STA_{shm} + PA_{sh}$
$TSTA_h = F_{shm}$
$MAA_{sm} = TSTA_h$

İşin diğer aşamalardaki diğer operasyonları için tekrarla.

Sıralamadaki diğer ürünler için tekrarla.

F_{shm} 'lerden en büyüğünü bul, bu $C_{\max}$ yani en son biten işin bitiş zamanıdır.
Tüm gecikmelerin toplamını hesapla; $\sum T_j$

Uyarlanmış Johnson Kuralı

Her bir ürün için montaj öncesi işleri ile montaj ve sonrası işlerin işlem sürelerini toplayarak ürünün toplam işlem süresini hesapla.

$$PTBA_a = \sum_{j=1}^{J} \sum_{r=1}^{R} P_{rj} \quad ; \quad j \in J_a, \qquad \forall a$$

$$PTA_a = \sum_{h=1}^{H} \sum_{s=1}^{S} PA_{sh} \quad ; \quad h \in H_a, \qquad \forall a$$

$PT_a = PTBA_a + PTA_a \quad ; \quad \forall a$

Her bir ürün için montaj öncesi işlem süresini toplam işlem süresine oranlayarak sıralama oranını bul.

$BAR_a = PTBA_a / PT_a \quad ; \quad \forall\, a$ (Before assembly ratio)

Ürünleri bu orana göre artan şekilde sırala.

Bu sıralamada ilk ürünü seç.

Seçilen ürünün montaj öncesi işlerinin toplam işlem sürelerini hesapla.

$$PT_{aj} = \sum_{r=1}^{R} P_{rj} \quad ; \quad \forall\, j$$

İşleri PT_{aj} değerlerine göre azalan şekilde sırala.

Sıralamadaki ilk işi seç.

Seçilen işin atanacağı ilk aşamadan başla.

Makinenin müsait olma zamanına makinedeki son iş ile mevcut iş arasındaki hazırlık süresini de ekleyerek makinenin işe başlayabileceği zamanı belirle.

$PST_{rjm} = MA_{rm} + SU_{rMJ_{rm}j} \; ; \; \forall\; m$

PST_{rjm}'ler içinde en küçük olanı bulup j şinin r aşamasındaki operasyonunu m makinesine ata.

$PST_{rjm} >= TST_j$ ise

$ST_{rjm} = PST_{rjm}$

değilse

$ST_{rjm} = TST_j$

$C_{rjm} = ST_{rjm} + P_{rj}$
$TST_j = C_{rjm}$
$MA_{rm} = TST_j$
$MJ_{rm} = j$

İşin diğer aşamalardaki diğer operasyonları için

PT_{aj} sıralamasındaki diğer işler için tekrarla

Seçilmiş ürünün montaj ve sonrası aşamalarını sıradan atamak için a ürününe ait TST_j ler içinde en yüksek olanı seç.

$TSTA_h = Max\left(TST_{J_a}\right) \quad ; \quad \forall\, H_a$

İlk işi seç (işlem süreleri birbirine eşit, eşit olmayan durumlar için uzun işlem süreli olan da seçilebilir)

Seçilen işin ilk aşamasından başla.

Min MAA_{sm}'yi bul.

$MAA_{sm} >= TSTA_h$ ise

$STA_{shm} = MAA_{sm}$

değilse

$STA_{shm} = TSTA_h$

$F_{shm} = STA_{shm} + PA_{sh}$
$TSTA_h = F_{shm}$
$MAA_{sm} = TSTA_h$

İşin diğer aşamalardaki diğer operasyonları için tekrarla.

Sıralamadaki diğer ürünler için tekrarla.

F_{shm} 'lerden en büyüğünü bul, bu $C_{\max}$ yani en son biten işin bitiş zamanıdır.
Tüm gecikmelerin toplamını hesapla; $\sum T_j$

En Az Boşluk Kuralı

Montaj ve sonrasındaki her bir işin boşluğunu hesaplamak için her işin teslim tarihinden o işin montaj ve sonrasındaki aşamalardaki işlem süresi ile bağlı olduğu ürünün montaj öncesi toplam işlem süreleri toplamını çıkartmak gerekir.

$$SL_h = DD_h - (PTBA_a + \sum_{s=1}^{S} P_{sh}) \quad ; \quad \forall\, h$$

$$PTB_a = \sum_{j=1}^{J} \sum_{r=1}^{R} P_{rj} \quad ; \quad j \in J_a, \quad \forall\, a$$

İşleri SL_h değerine göre artan şekilde sırala.

Bu sıralamada ilk işi seç.

Seçilen işin bağlı olduğu ürünün montaj öncesi işleri atanmadıysa montaj öncesi işleri toplam işlem sürelerini hesapla.

$$PT_{aj} = \sum_{r=1}^{R} P_{rj} \quad ; \quad \forall\, j$$

İşleri PT_{aj} değerlerine göre azalan şekilde sırala.

Sıralamadaki ilk işi seç.

Seçilen işin atanacağı ilk aşamadan başla.

Makinenin müsait olma zamanına makinedeki son iş ile mevcut iş arasındaki hazırlık süresini de ekleyerek makinenin işe başlayabileceği zamanı belirle.

$$PST_{rjm} = MA_{rm} + SU_{rMJ_{rm}j} \;;\; \forall\; m$$

PST_{rjm}'ler içinde en küçük olanı bulup j şinin r aşamasındaki operasyonunu m makinesine ata.

$PST_{rjm} >= TST_j$ ise

$ST_{rjm} = PST_{rjm}$

değilse

$ST_{rjm} = TST_j$

$C_{rjm} = ST_{rjm} + P_{rj}$
$TST_j = C_{rjm}$
$MA_{rm} = TST_j$
$MJ_{rm} = j$

İşin diğer aşamalardaki diğer operasyonları için tekrarla.

PT_{aj} sıralamasındaki diğer işler için tekrarla.

Seçilmiş ürünün montaj ve sonrası aşamalarını sıradan atamak için a ürününe ait TST_j ler içinde en yüksek olanı seç.

$TSTA_h = Max(TST_{J_a})$; $\forall H_a$

Seçilen işin ilk aşamasından başla.

Min MAA_{sm}'yi bul.

$MAA_{sm} >= TSTA_h$ ise

$STA_{shm} = MAA_{sm}$

değilse

$STA_{shm} = TSTA_h$

$F_{shm} = STA_{shm} + PA_{sh}$
$TSTA_h = F_{shm}$
$MAA_{sm} = TSTA_h$

İşin diğer aşamalardaki diğer operasyonları için tekrarla.

Diğer montaj sonrası işler için tekrarla.

F_{shm} 'lerden en büyüğünü bul, bu $C_{\max}$ yani en son biten işin bitiş zamanıdır.
Tüm gecikmelerin toplamını hesapla; $\sum T_j$

Partilere Ayrılmış En Az Boşluk Kuralı

Montaj ve sonrası aşamalarda her bir işin boşluğunu hesaplamak için her işin teslim tarihinden o işin montaj ve sonrası aşamalardaki işlem süresi ile bağlı olduğu ürünün montaj öncesi ilk aşamadaki işleri ve montaj öncesi partilere ayrılmış işlerden ilgili işin işlem süreleri toplamını çıkartmak gerekir.

$$SL_h = DD_h - \left(\left(\sum_{j=1}^{J} P_{1j}\right) + \left(\sum_{i=1}^{I}\sum_{r=2}^{R} PL_{ri}\right) + \left(\sum_{s=1}^{S} P_{sh}\right)\right) \quad ; \; j \in J_a, i \in I_h, \quad \forall h$$

İşleri SL_h değerine göre artan şekilde sırala.

Bu sıralamada ilk işi seç.

Seçilen işin bağlı olduğu ürünün montaj öncesi ilk aşamadaki işleri atanmadıysa bu işleri işlem sürelerine göre azalan şekilde sırala

Sıralamadaki ilk işi seç.

Makinenin müsait olma zamanına makinedeki son iş ile mevcut iş arasındaki hazırlık süresini de ekleyerek makinenin işe başlayabileceği zamanı belirle.

$$PST_{rjm} = MA_{rm} + SU_{rMJ_{rm}j} \;\; ; \;\; \forall \; m$$

PST_{rjm}'ler içinde en küçük olanı bulup j şinin r aşamasındaki operasyonunu m makinesine ata.

$PST_{rjm} >= TST_j$ ise

$$ST_{rjm} = PST_{rjm}$$

değilse

$$ST_{rjm} = TST_j$$

$C_{rjm} = ST_{rjm} + P_{rj}$
$TST_j = C_{rjm}$
$MA_{rm} = TST_j$
$MJ_{rm} = j$

İşin diğer aşamalardaki diğer operasyonları için tekrarla.

Montaj öncesi ilk aşamadan sonra partilere ayrılmış işlerden montaj ve sonrası boşluğa göre seçilmiş işe bağlı olanları seç, bu işlerin işlem sürelerini topla ve bu süreye göre azalan şekilde bu işleri sırala

$$\sum_{r=2}^{R} PL_{ri} \quad ; \quad \forall \, i \in I_h$$

Partilere ayrılmış her bir işin işe başlayabilme zamanı ilk aşamada bağlı olduğu işin bitiş zamanıdır.

$$TSTL_i = TST_j \quad ; \; j \in J_a \, , \qquad \forall \, i \in I_h$$

Sıralamadaki ilk işi seç

Atanacağı aşamadaki makinenin müsait olma durumunu ve işin önceki aşamadan çıkış zamanını dikkate alarak ve makinedeki son iş ile mevcut iş arasındaki hazırlık süresini de ekleyerek makinenin işe başlayabileceği zamanı belirle.

$$PSTL_{rim} = MA_{rm} + SU_{rMJ_{rm}i} \;\; ; \;\; \forall \, m$$

$PSTL_{rim}$'ler içinde en küçük olanı bulup i işinin r aşamasındaki operasyonunu m makinesine ata

$PSTL_{rim} >= TSTL_i$ ise

$$STL_{rim} = PSTL_{rim}$$

değilse

$$STL_{rim} = TSTL_i$$

$CL_{rim} = STL_{rim} + PL_{ri}$
$TSTL_i = CL_{rim}$
$MA_{rm} = TSTL_i$
$MJ_{rm} = i$

Sıralamadaki diğer işler için tekrarla.

Seçilmiş ürünün montaj ve sonrası aşamalarını sıradan atamak için *a* ürününe ait $TSTL_i$ ler içinde en yüksek olanı seç.

$$TSTA_h = Max\left(TSTL_{i_a}\right) \;\; ; \;\; \forall H_a$$

Seçilen işin ilk aşamasından başla.

Min MAA_{sm}'yi bul.

$MAA_{sm} >= TSTA_h$ ise

$STA_{shm} = MAA_{sm}$

değilse

$STA_{shm} = TSTA_h$

$F_{shm} = STA_{shm} + PA_{sh}$
$TSTA_h = F_{shm}$
$MAA_{sm} = TSTA_h$

İşin diğer aşamalardaki diğer operasyonları için tekrarla.

Diğer montaj sonrası işler için tekrarla.

F_{shm} 'lerden en büyüğünü bul, bu $C_{\max}$ yani en son biten işin bitiş zamanıdır.
Tüm gecikmelerin toplamını hesapla; $\sum T_j$

EK-3: SEZGİSEL SIRALAMA KURALLARININ C# KODLARI

En Kısa İşlem Zamanı Kuralı C# Kodları

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.IO;
using System.Text;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int r, s, m, j, a, J = ., h, H = ., R = ., M = ., S = ., A = .;
            float[,] P ={
                {.,.},
                {.,.}
                      }; // montaj öncesi işlem süreleri r,j
            float[,] PA ={
                           {.,.,.}
                       }; // montaj sonrası işlem süreleri s,h
            float[, ,] SU ={
                             {{.,.},{.,.},{.,.}},
                                       {{.,.},{.,.},{.,.}},
                                       {{.,.},{.,.},{.,.}}
                          }; // hazırlık zamanları r,i,j
            int[] MS = { .,. }; // montaj öncesi aşamalardaki makine sayıları r
            int[] MSA = { . }; // montaj sonrası aşamalardaki makine sayısı s
            int[] MEC = { .,.,. }; // machine eligibility constraints j
            int[] MECEXT = { .,. }; // montaj öncesi hangi aşamalarda MEC olduğunu söylüyor, 0 yok 1 var
            float[] DUE = { .,.,. }; // teslim tarihi h
            float[] T = new float[H]; // her işin gecikmesi
            int[] STJ = { .,. }; // a ürünü hangi j işinden başlıyor a
            int[] FNJ = { .,. }; // a ürünü hangi j işinde bitiyor a
            int[] STJA = { .,.}; // a ürünü montaj sonrasında hangi h işinden başlıyor
            int[] FNJA = { .,. }; // a ürünü montaj sonrasında hangi h işinde bitiyor
            float[] TST = new float[J]; // temp.start time, geçiici başlama zamanı
            float[] TSTA = new float[H]; // montaj sonrası geçici başlama zamanı
            float[, ,] PST = new float[R, J, M]; // probable start time, olası başlama zamanı
            float[, ,] ST = new float[R, J, M]; // r aşamasında j işinin m makinesinde başlama zamanı
            float[, ,] STA = new float[S, H, M]; // s aşamasında h işinin m makinesinde başlama zamanı
            float[, ,] C = new float[R, J, M]; // r aşamasında j işinin m makinesinde tamamlanma zamanı
            float[, ,] F = new float[S, H, M]; // s aşamasında h işinin m makinesinde tamamlanma zamanı
            float[] K = new float[H]; // her işin bitiş zamanı
            float[,] MA = new float[R, M]; // r aşamasında m makinesinin müsait olma zamanı
            float[,] MAA = new float[S, M]; // s aşamasında m makinesinin müsait olma zamanı
            int[,] MJ = new int[R, M]; // r aşamasında m makinesindeki son iş
            int[] B = { .,.,. }; // Ana ürün sıralaması için değişken
            int[] D = { .,.,. }; // ürün içindeki işlerin sıralaması için değişken
            float[] PTBA = new float[A]; // a ürününün tüm bileşenlerinin montaj öncesi toplam işlem süreleri
toplamı
            float[] PTA = new float[A]; // a ürününün montaj sonrasındaki toplam işlem süresi
            float[] PT = new float[A]; // a ürününün toplam işlem süresi
            float[] PTAJ = new float[J]; // j işinin montaj öncesi tüm işlem sürelerinin toplamı
            float[] BAR = new float[A]; // a ürününün sıralama oranı
            StreamWriter dosya = new StreamWriter("d:\\ffs3j5SPT.txt");

            for (r = 0; r <= R - 1; r++) // ilk etapta montaj öncesi tüm aşamalarda tüm makinelerde olmayan
bir iş (J+1) olarak tanımlanıyor. Bu olmayan işten olan başka herhabgi bir işe geçişte hazırlık süresi 0
            {
                for (m = 0; m <= MS[r] - 1; m++)
                {
                    MJ[r, m] = J;
                }
            }
            // Sıralama Kuralı
            for (a = 0; a <= A - 1; a++)
            {
                for (j = STJ[a]; j <= FNJ[a]; j++)
                {
                    for (r = 0; r <= R - 1; r++)
                        PTBA[a] = PTBA[a] + P[r, j];
                }
            }
            for (a = 0; a <= A - 1; a++)
            {
                for (h = STJA[a]; h <= FNJA[a]; h++)
```

```
                {
                    for (s = 0; s <= S - 1; s++)
                        PTA[a] = PTA[a] + PA[s, h];
                }
            }
            for (a = 0; a <= A - 1; a++)
            {
                PT[a] = PTBA[a] + PTA[a];
            }
            for (int y = 0; y < PT.Length - 1; y++) //BAR oranını küçükten büyüğe sıralama
            {
                for (int z = 1; z < PT.Length - y; z++)
                {
                    if (PT[z] < PT[z - 1])
                    {
                        float gecici1 = PT[z - 1];
                        int gecici2 = B[z - 1];
                        PT[z - 1] = PT[z];
                        B[z - 1] = B[z];
                        PT[z] = gecici1;
                        B[z] = gecici2;
                    }
                }
            }

            // makinelere atama
            for (a = 0; a <= A - 1; a++)
            {
                for (j = 0; j <= J - 1; j++) // j iş sıralaması her üründe 0'dan başlamayıp devam ettiğinden
PTAJ sıralamasında sıkıntı çıkarıyor, o yüzden başta hepsini 0'ladık
                    PTAJ[j] = 1000;
                int sıra = B[a]; // sıra değişkenine BAR'ı küçükten büyüğü sıralarken aynı şekilde sıralanmış
B[] dizisindeki ilk ürünü atıyoruz.
                for (j = STJ[sıra - 1]; j <= FNJ[sıra - 1]; j++)
                    PTAJ[j] = 0;
                for (j = STJ[sıra - 1]; j <= FNJ[sıra - 1]; j++) // bir işin aşamalardaki toplam işlem süresi
PTaj döngüsü
                {
                    for (r = 0; r <= R - 1; r++)
                    {
                        PTAJ[j] = PTAJ[j] + P[r, j];
                    }
                }
                for (int y = 0; y < D.Length - 1; y++) // PTAJ'leri sıraya dizmek için kullanılacak D
dizisinin her yeni üründe bştan sıralamak için
                {
                    for (int z = 1; z < D.Length - y; z++)
                    {
                        if (D[z] < D[z - 1])
                        {
                            int gecici = D[z - 1];
                            D[z - 1] = D[z];
                            D[z] = gecici;
                        }
                    }
                }
                for (int y = 0; y < PTAJ.Length - 1; y++) // PTaj'leri küçükten büyüğe sıralama döngüsü
                {
                    for (int z = 1; z < PTAJ.Length - y; z++)
                    {
                        if (PTAJ[z] < PTAJ[z - 1])
                        {
                            float gecici1 = PTAJ[z - 1];
                            int gecici2 = D[z - 1];
                            PTAJ[z - 1] = PTAJ[z];
                            D[z - 1] = D[z];
                            PTAJ[z] = gecici1;
                            D[z] = gecici2;
                        }
                    }
                }
                float maximum = 0;
                for (int k = 0; k <= FNJ[sıra - 1] - STJ[sıra - 1]; k++)
                {
                    int sıra2 = D[k];
                    for (r = 0; r <= R - 1; r++) //işin her aşamalarına atama yapacak döngü
                    {
                        if (P[r, sıra2 - 1] > 0) // eğer aşamadaki işlem süresi 0 ise aşamayı atla
                        {
                            float minimum = 10000;
                            int makinetercih = 0;
                            if (MECEXT[r] == 1)
                            {
```

```
                                makinetercih = MEC[sıra2 - 1] - 1;
                                int makinedekii$1 = MJ[r, makinetercih];
                                PST[r, sıra2 - 1, makinetercih] = MA[r, makinetercih] + SU[r, makinedekii$1,
sıra2 - 1];
                            }
                            else
                            {
                                for (m = 0; m <= MS[r] - 1; m++)
                                {
                                    int makinedekii$ = MJ[r, m];
                                    PST[r, sıra2 - 1, m] = MA[r, m] + SU[r, makinedekii$, sıra2 - 1];
                                    if (PST[r, sıra2 - 1, m] < minimum)
                                    {
                                        minimum = PST[r, sıra2 - 1, m];
                                        makinetercih = m;
                                    }
                                }
                            }
                            if (PST[r, sıra2 - 1, makinetercih] >= TST[sıra2 - 1])
                                ST[r, sıra2 - 1, makinetercih] = PST[r, sıra2 - 1, makinetercih];
                            else
                                ST[r, sıra2 - 1, makinetercih] = TST[sıra2 - 1];
                            C[r, sıra2 - 1, makinetercih] = ST[r, sıra2 - 1, makinetercih] + P[r, sıra2 - 1];
                            TST[sıra2 - 1] = C[r, sıra2 - 1, makinetercih];
                            MA[r, makinetercih] = TST[sıra2 - 1];
                            MJ[r, makinetercih] = (sıra2 - 1);
                            dosya.WriteLine("ST" + (r + 1) + "." + sıra2 + "." + (makinetercih + 1) + " : " +
ST[r, sıra2 - 1, makinetercih] + "  C" + (r + 1) + "." + sıra2 + "." + (makinetercih + 1) + " : " + C[r, sıra2
- 1, makinetercih]);
                            if (TST[sıra2 - 1] > maximum)
                                maximum = TST[sıra2 - 1];
                        }
                    }
                }
                for (int k = STJA[sıra - 1]; k <= FNJA[sıra - 1]; k++)
                {
                    TSTA[k] = maximum;
                }
                int makinetercih2 = 0;
                for (h = STJA[sıra - 1]; h <= FNJA[sıra - 1]; h++)
                {
                    for (s = 0; s <= S - 1; s++)
                    {
                        float minima = 10000;
                        for (m = 0; m <= MSA[s] - 1; m++)
                        {
                            if (MAA[s, m] < minima)
                            {
                                minima = MAA[s, m];
                                makinetercih2 = m;
                            }
                        }
                        if (MAA[s, makinetercih2] >= TSTA[h])
                        {
                            STA[s, h, makinetercih2] = MAA[s, makinetercih2];
                        }
                        else
                        {
                            STA[s, h, makinetercih2] = TSTA[h];
                        }
                        F[s, h, makinetercih2] = STA[s, h, makinetercih2] + PA[s, h];
                        TSTA[h] = F[s, h, makinetercih2];
                        MAA[s, makinetercih2] = TSTA[h];
                        dosya.WriteLine("STA" + (s + 1) + "." + (h + 1) + "." + (makinetercih2 + 1) + " : " +
STA[s, h, makinetercih2] + "  F" + (s + 1) + "." + (h + 1) + "." + (makinetercih2 + 1) + " : " + F[s, h,
makinetercih2]);
                    }
                }
            }
            float Totcomp = 0;
            for (h = 0; h <= H - 1; h++)
            {
                float maximum = 0;
                for (m = 0; m <= MSA[S - 1] - 1; m++)
                {
                    if (F[S - 1, h, m] > maximum)
                    {
                        maximum = F[S - 1, h, m];
                    }
                }
                K[h] = maximum;
                Totcomp = Totcomp + K[h];
            }
```

```
            float Cmax = 0;
            for (h = 0; h <= H - 1; h++)
            {
                for (m = 0; m <= MSA[S - 1] - 1; m++)  // m.leri saydırmaya hiç gerek olmadan h döngüsü ile
K[h] lar içinden maximum Cmax olur.
                {
                    if (F[S - 1, h, m] > Cmax) // S'in döngüye girmesi sıkıntı olursa s=S s<=S'e kadar
döngükurulabilir.
                        Cmax = F[S - 1, h, m];
                }
            }
            int Uj = 0;
            for (h = 0; h <= H - 1; h++)
            {
                if (K[h] > DUE[h])
                {
                    T[h] = K[h] - DUE[h];
                    Uj = Uj + 1;
                }
            }
            float Tmax = 0;
            float Tottard = 0;
            for (h = 0; h <= H - 1; h++)
            {
                if (T[h] > Tmax)
                {
                    Tmax = T[h];
                }
                Tottard = Tottard + T[h];
            }
            dosya.WriteLine("ffs" + (S + R) + "j" + J + " Johnson Sonuçlar:");
            dosya.WriteLine("Cmax : " + Cmax);
            dosya.WriteLine("Tmax : " + Tmax);
            dosya.WriteLine("Toplam Uj : " + Uj);
            dosya.WriteLine("Totcomp : " + Totcomp);
            dosya.WriteLine("Tottard : " + Tottard);
            dosya.Close();
            Console.WriteLine("Dosya yazımı başarı ile tamamlandı..");
            Console.ReadLine();
        }
    }
}
```

En Uzun İşlem Zamanı Kuralı C# Kodları

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.IO;
using System.Text;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int r, s, m, j, a, J = ., h, H = ., R = ., M = ., S = ., A = .;
            float[,] P ={
                {.,.},
                {.,.}
                       }; // montaj öncesi işlem süreleri r,j
            float[,] PA ={
                         {.,.,.}
                      }; // montaj sonrası işlem süreleri s,h
            float[, ,] SU ={
                            {{.,.},{.,.},{.,.}},
                                     {{.,.},{.,.},{.,.}},
                                     {{.,.},{.,.},{.,.}}
                         }; // hazırlık zamanları r,i,j
            int[] MS = { .,. }; // montaj öncesi aşamalardaki makine sayıları r
            int[] MSA = { . }; // montaj sonrası aşamalardaki makine sayısı s
            int[] MEC = { .,.,. }; // machine eligibility constraints j
            int[] MECEXT = { .,. }; // montaj öncesi hangi aşamalarda MEC olduğunu söylüyor, 0 yok 1 var
            float[] DUE = { .,.,. }; // teslim tarihi h
            float[] T = new float[H]; // her işin gecikmesi
            int[] STJ = { .,. }; // a ürünü hangi j işinden başlıyor a
            int[] FNJ = { .,. }; // a ürünü hangi j işinde bitiyor a
            int[] STJA = { .,.}; // a ürünü montaj sonrasında hangi h işinden başlıyor
            int[] FNJA = { .,. }; // a ürünü montaj sonrasında hangi h işinde bitiyor
            float[] TST = new float[J]; // temp.start time, geçiici başlama zamanı
            float[] TSTA = new float[H]; // montaj sonrası geçici başlama zamanı
```

```
            float[, ,] PST = new float[R, J, M]; // probable start time, olası başlama zamanı
            float[, ,] ST = new float[R, J, M]; // r aşamasında j işinin m makinesinde başlama zamanı
            float[, ,] STA = new float[S, H, M]; // s aşamasında h işinin m makinesinde başlama zamanı
            float[, ,] C = new float[R, J, M]; // r aşamasında j işinin m makinesinde tamamlanma zamanı
            float[, ,] F = new float[S, H, M]; // s aşamasında h işinin m makinesinde tamamlanma zamanı
            float[] K = new float[H]; // her işin bitiş zamanı
            float[,] MA = new float[R, M]; // r aşamasında m makinesinin müsait olma zamanı
            float[,] MAA = new float[S, M]; // s aşamasında m makinesinin müsait olma zamanı
            int[,] MJ = new int[R, M]; // r aşamasında m makinesindeki son iş
            int[] B = { .,.,. }; // Ana ürün sıralaması için değişken
            int[] D = { .,.,. }; // ürün içindeki işlerin sıralaması için değişken
            float[] PTBA = new float[A]; // a ürününün tüm bileşenlerinin montaj öncesi toplam işlem süreleri
toplamı
            float[] PTA = new float[A]; // a ürününün montaj sonrasındaki toplam işlem süresi
            float[] PT = new float[A]; // a ürününün toplam işlem süresi
            float[] PTAJ = new float[J]; // j işinin montaj öncesi tüm işlem sürelerinin toplamı
            float[] BAR = new float[A]; // a ürününün sıralama oranı
            StreamWriter dosya = new StreamWriter("d:\\ffs3j5LPT.txt");

            for (r = 0; r <= R - 1; r++) // ilk etapta montaj öncesi tüm aşamalarda tüm makinelerde olmayan
bir iş (J+1) olarak tanımlanıyor. Bu olmayan işten olan başka herhabgi bir işe geçişte hazırlık süresi 0
            {
                for (m = 0; m <= MS[r] - 1; m++)
                {
                    MJ[r, m] = J;
                }
            }
            // Sıralama Kuralı
            for (a = 0; a <= A - 1; a++)
            {
                for (j = STJ[a]; j <= FNJ[a]; j++)
                {
                    for (r = 0; r <= R - 1; r++)
                        PTBA[a] = PTBA[a] + P[r, j];
                }
            }
            for (a = 0; a <= A - 1; a++)
            {
                for (h = STJA[a]; h <= FNJA[a]; h++)
                {
                    for (s = 0; s <= S - 1; s++)
                        PTA[a] = PTA[a] + PA[s, h];
                }
            }
            for (a = 0; a <= A - 1; a++)
            {
                PT[a] = PTBA[a] + PTA[a];
            }
            for (int y = 0; y < PT.Length - 1; y++) //BAR oranını küçükten büyüğe sıralama
            {
                for (int z = 1; z < PT.Length - y; z++)
                {
                    if (PT[z] > PT[z - 1])
                    {
                        float gecici1 = PT[z - 1];
                        int gecici2 = B[z - 1];
                        PT[z - 1] = PT[z];
                        B[z - 1] = B[z];
                        PT[z] = gecici1;
                        B[z] = gecici2;
                    }
                }
            }

            // makinelere atama
            for (a = 0; a <= A - 1; a++)
            {
                for (j = 0; j <= J - 1; j++) // j iş sıralaması her üründe 0'dan başlamayıp devam ettiğinden
PTAJ sıralamasında sıkıntı çıkarıyor, o yüzden başta hepsini 0'ladık
                    PTAJ[j] = 0;
                int sıra = B[a]; // sıra değişkenine BAR'ı küçükten büyüğü sıralarken aynı şekilde sıralanmış
B[] dizisindeki ilk ürünü atıyoruz.
                for (j = STJ[sıra - 1]; j <= FNJ[sıra - 1]; j++) // bir işin aşamalardaki toplam işlem süresi
PTaj döngüsü
                {
                    for (r = 0; r <= R - 1; r++)
                    {
                        PTAJ[j] = PTAJ[j] + P[r, j];
                    }
                }
                for (int y = 0; y < D.Length - 1; y++) // PTAJ'leri sıraya dizmek için kullanılacak D
dizisinin her yeni üründe bştan sıralamak için
                {
```

```
                    for (int z = 1; z < D.Length - y; z++)
                    {
                        if (D[z] < D[z - 1])
                        {
                            int gecici = D[z - 1];
                            D[z - 1] = D[z];
                            D[z] = gecici;
                        }
                    }
                }
                for (int y = 0; y < PTAJ.Length - 1; y++) // PTaj'leri büyükten küçüğe sıralama döngüsü
                {
                    for (int z = 1; z < PTAJ.Length - y; z++)
                    {
                        if (PTAJ[z] > PTAJ[z - 1])
                        {
                            float gecici1 = PTAJ[z - 1];
                            int gecici2 = D[z - 1];
                            PTAJ[z - 1] = PTAJ[z];
                            D[z - 1] = D[z];
                            PTAJ[z] = gecici1;
                            D[z] = gecici2;
                        }
                    }
                }
                float maximum = 0;
                for (int k = 0; k <= FNJ[sıra - 1] - STJ[sıra - 1]; k++)
                {
                    int sıra2 = D[k];
                    for (r = 0; r <= R - 1; r++) //işin her aşamalarına atama yapacak döngü
                    {
                        if (P[r, sıra2 - 1] > 0) // eğer aşamadaki işlem süresi 0 ise aşamayı atla
                        {
                            float minimum = 10000;
                            int makinetercih = 0;
                            if (MECEXT[r] == 1)
                            {
                                makinetercih = MEC[sıra2 - 1] - 1;
                                int makinedekiiş1 = MJ[r, makinetercih];
                                PST[r, sıra2 - 1, makinetercih] = MA[r, makinetercih] + SU[r, makinedekiiş1,
sıra2 - 1];
                            }
                            else
                            {
                                for (m = 0; m <= MS[r] - 1; m++)
                                {
                                    int makinedekiiş = MJ[r, m];
                                    PST[r, sıra2 - 1, m] = MA[r, m] + SU[r, makinedekiiş, sıra2 - 1];
                                    if (PST[r, sıra2 - 1, m] < minimum)
                                    {
                                        minimum = PST[r, sıra2 - 1, m];
                                        makinetercih = m;
                                    }
                                }
                            }
                            if (PST[r, sıra2 - 1, makinetercih] >= TST[sıra2 - 1])
                                ST[r, sıra2 - 1, makinetercih] = PST[r, sıra2 - 1, makinetercih];
                            else
                                ST[r, sıra2 - 1, makinetercih] = TST[sıra2 - 1];
                            C[r, sıra2 - 1, makinetercih] = ST[r, sıra2 - 1, makinetercih] + P[r, sıra2 - 1];
                            TST[sıra2 - 1] = C[r, sıra2 - 1, makinetercih];
                            MA[r, makinetercih] = TST[sıra2 - 1];
                            MJ[r, makinetercih] = (sıra2 - 1);
                            dosya.WriteLine("ST" + (r + 1) + "." + sıra2 + "." + (makinetercih + 1) + " : " +
ST[r, sıra2 - 1, makinetercih] + "  C" + (r + 1) + "." + sıra2 + "." + (makinetercih + 1) + " : " + C[r, sıra2
- 1, makinetercih]);
                            if (TST[sıra2 - 1] > maximum)
                                maximum = TST[sıra2 - 1];
                        }
                    }
                }
                for (int k = STJA[sıra - 1]; k <= FNJA[sıra - 1]; k++)
                {
                    TSTA[k] = maximum;
                }
                int makinetercih2 = 0;
                for (h = STJA[sıra - 1]; h <= FNJA[sıra - 1]; h++)
                {
                    for (s = 0; s <= S - 1; s++)
                    {
                        float minima = 10000;
                        for (m = 0; m <= MSA[s] - 1; m++)
                        {
```

```
                            if (MAA[s, m] < minima)
                            {
                                minima = MAA[s, m];
                                makinetercih2 = m;
                            }
                        }
                        if (MAA[s, makinetercih2] >= TSTA[h])
                        {
                            STA[s, h, makinetercih2] = MAA[s, makinetercih2];
                        }
                        else
                        {
                            STA[s, h, makinetercih2] = TSTA[h];
                        }
                        F[s, h, makinetercih2] = STA[s, h, makinetercih2] + PA[s, h];
                        TSTA[h] = F[s, h, makinetercih2];
                        MAA[s, makinetercih2] = TSTA[h];
                        dosya.WriteLine("STA" + (s + 1) + "." + (h + 1) + "." + (makinetercih2 + 1) + " : " +
STA[s, h, makinetercih2] + "  F" + (s + 1) + "." + (h + 1) + "." + (makinetercih2 + 1) + " : " + F[s, h,
makinetercih2]);
                    }
                }
            }
            float Totcomp = 0;
            for (h = 0; h <= H - 1; h++)
            {
                float maximum = 0;
                for (m = 0; m <= MSA[S - 1] - 1; m++)
                {
                    if (F[S - 1, h, m] > maximum)
                    {
                        maximum = F[S - 1, h, m];
                    }
                }
                K[h] = maximum;
                Totcomp = Totcomp + K[h];
            }
            float Cmax = 0;
            for (h = 0; h <= H - 1; h++)
            {
                for (m = 0; m <= MSA[S - 1] - 1; m++)  // m.leri saydırmaya hiç gerek olmadan h döngüsü ile
K[h] lar içinden maximum Cmax olur.
                {
                    if (F[S - 1, h, m] > Cmax) // S'in döngüye girmesi sıkıntı olursa s=S s<=S'e kadar
döngükurulabilir.
                        Cmax = F[S - 1, h, m];
                }
            }
            int Uj = 0;
            for (h = 0; h <= H - 1; h++)
            {
                if (K[h] > DUE[h])
                {
                    T[h] = K[h] - DUE[h];
                    Uj = Uj + 1;
                }
            }
            float Tmax = 0;
            float Tottard = 0;
            for (h = 0; h <= H - 1; h++)
            {
                if (T[h] > Tmax)
                {
                    Tmax = T[h];
                }
                Tottard = Tottard + T[h];
            }
            dosya.WriteLine("ffs" + (S + R) + "j" + J + " Johnson Sonuçlar:");
            dosya.WriteLine("Cmax : " + Cmax);
            dosya.WriteLine("Tmax : " + Tmax);
            dosya.WriteLine("Toplam Uj : " + Uj);
            dosya.WriteLine("Totcomp : " + Totcomp);
            dosya.WriteLine("Tottard : " + Tottard);
            dosya.Close();
            Console.WriteLine("Dosya yazımı başarı ile tamamlandı..");
            Console.ReadLine();
        }
    }
}
```

Uyarlanmış Johnson Kuralı C# Kodları

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.IO;
using System.Text;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int r, s, m, j, a, J = ., h, H = ., R = ., M = ., S = ., A = .;
            float[,] P ={
                {.,.},
                {.,.}
                      }; // montaj öncesi işlem süreleri r,j
            float[,] PA ={
                          {.,.,.}
                       }; // montaj sonrası işlem süreleri s,h
            float[, ,] SU ={
                             {{.,.},{.,.},{.,.}},
                                      {{.,.},{.,.},{.,.}},
                                      {{.,.},{.,.},{.,.}}
                         }; // hazırlık zamanları r,i,j
            int[] MS = { .,. }; // montaj öncesi aşamalardaki makine sayıları r
            int[] MSA = { . }; // montaj sonrası aşamalardaki makine sayısı s
            int[] MEC = { .,.,. }; // machine eligibility constraints j
            int[] MECEXT = { .,. }; // montaj öncesi hangi aşamalarda MEC olduğunu söylüyor, 0 yok 1 var
            float[] DUE = { .,.,. }; // teslim tarihi h
            float[] T = new float[H]; // her işin gecikmesi
            int[] STJ = { .,. }; // a ürünü hangi j işinden başlıyor a
            int[] FNJ = { .,. }; // a ürünü hangi j işinde bitiyor a
            int[] STJA = { .,.}; // a ürünü montaj sonrasında hangi h işinden başlıyor
            int[] FNJA = { .,. }; // a ürünü montaj sonrasında hangi h işinde bitiyor
            float[] TST = new float[J]; // temp.start time, geçiici başlama zamanı
            float[] TSTA = new float[H]; // montaj sonrası geçici başlama zamanı
            float[, ,] PST = new float[R, J, M]; // probable start time, olası başlama zamanı
            float[, ,] ST = new float[R, J, M]; // r aşamasında j işinin m makinesinde başlama zamanı
            float[, ,] STA = new float[S, H, M]; // s aşamasında h işinin m makinesinde başlama zamanı
            float[, ,] C = new float[R, J, M]; // r aşamasında j işinin m makinesinde tamamlanma zamanı
            float[, ,] F = new float[S, H, M]; // s aşamasında h işinin m makinesinde tamamlanma zamanı
            float[] K = new float[H]; // her işin bitiş zamanı
            float[,] MA = new float[R, M]; // r aşamasında m makinesinin müsait olma zamanı
            float[,] MAA = new float[S, M]; // s aşamasında m makinesinin müsait olma zamanı
            int[,] MJ = new int[R, M]; // r aşamasında m makinesindeki son iş
            int[] B = { .,.,. }; // Ana ürün sıralaması için değişken
            int[] D = { .,.,. }; // ürün içindeki işlerin sıralaması için değişken
            float[] PTBA = new float[A]; // a ürününün tüm bileşenlerinin montaj öncesi toplam işlem süreleri
toplamı
            float[] PTA = new float[A]; // a ürününün montaj sonrasındaki toplam işlem süresi
            float[] PT = new float[A]; // a ürününün toplam işlem süresi
            float[] PTAJ = new float[J]; // j işinin montaj öncesi tüm işlem sürelerinin toplamı
            float[] BAR = new float[A]; // a ürününün sıralama oranı
            StreamWriter dosya = new StreamWriter("d:\\ffs3j5johnson.txt");

            for (r = 0; r <= R - 1; r++) // ilk etapta montaj öncesi tüm aşamalarda tüm makinelerde olmayan
bir iş (J+1) olarak tanımlanıyor. Bu olmayan işten olan başka herhabgi bir işe geçişte hazırlık süresi 0
            {
                for (m = 0; m <= MS[r] - 1; m++)
                {
                    MJ[r, m] = J;
                }
            }
            // Sıralama Kuralı
            for (a = 0; a <= A - 1; a++)
            {
                for (j = STJ[a]; j <= FNJ[a]; j++)
                {
                    for (r = 0; r <= R - 1; r++)
                        PTBA[a] = PTBA[a] + P[r, j];
                }
            }
            for (a = 0; a <= A - 1; a++)
            {
                for (h = STJA[a]; h <= FNJA[a]; h++)
                {
                    for (s = 0; s <= S - 1; s++)
                        PTA[a] = PTA[a] + PA[s, h];
                }
```

```
            }
            for (a = 0; a <= A - 1; a++)
            {
                PT[a] = PTBA[a] + PTA[a];
            }
            for (a = 0; a <= A - 1; a++)
            {
                BAR[a] = PTBA[a] / PT[a];
            }
            for (int y = 0; y < BAR.Length - 1; y++) //BAR oranını küçükten büyüğe sıralama
            {
                for (int z = 1; z < BAR.Length - y; z++)
                {
                    if (BAR[z] < BAR[z - 1])
                    {
                        float gecici1 = BAR[z - 1];
                        int gecici2 = B[z - 1];
                        BAR[z - 1] = BAR[z];
                        B[z - 1] = B[z];
                        BAR[z] = gecici1;
                        B[z] = gecici2;
                    }
                }
            }

            // makinelere atama
            for (a = 0; a <= A - 1; a++)
            {
                for (j = 0; j <= J - 1; j++) // j iş sıralaması her üründe 0'dan başlamayıp devam ettiğinden
PTAJ sıralamasında sıkıntı çıkarıyor, o yüzden başta hepsini 0'ladık
                    PTAJ[j] = 0;
                int sıra = B[a]; // sıra değişkenine BAR'ı küçükten büyüğü sıralarken aynı şekilde sıralanmış
B[] dizisindeki ilk ürünü atıyoruz.
                for (j = STJ[sıra - 1]; j <= FNJ[sıra - 1]; j++) // bir işin aşamalardaki toplam işlem süresi
PTaj döngüsü
                {
                    for (r = 0; r <= R - 1; r++)
                    {
                        PTAJ[j] = PTAJ[j] + P[r, j];
                    }
                }
                for (int y = 0; y < D.Length - 1; y++) // PTAJ'leri sıraya dizmek için kullanılacak D
dizisinin her yeni üründe bştan sıralamak için
                {
                    for (int z = 1; z < D.Length - y; z++)
                    {
                        if (D[z] < D[z - 1])
                        {
                            int gecici = D[z - 1];
                            D[z - 1] = D[z];
                            D[z] = gecici;
                        }
                    }
                }
                for (int y = 0; y < PTAJ.Length - 1; y++) // PTaj'leri büyükten küçüğe sıralama döngüsü
                {
                    for (int z = 1; z < PTAJ.Length - y; z++)
                    {
                        if (PTAJ[z] > PTAJ[z - 1])
                        {
                            float gecici1 = PTAJ[z - 1];
                            int gecici2 = D[z - 1];
                            PTAJ[z - 1] = PTAJ[z];
                            D[z - 1] = D[z];
                            PTAJ[z] = gecici1;
                            D[z] = gecici2;
                        }
                    }
                }
                float maximum = 0;
                for (int k = 0; k <= FNJ[sıra - 1] - STJ[sıra - 1]; k++)
                {
                    int sıra2 = D[k];
                    for (r = 0; r <= R - 1; r++) //işin her aşamalarına atama yapacak döngü
                    {
                        if (P[r, sıra2 - 1] > 0) // eğer aşamadaki işlem süresi 0 ise aşamayı atla
                        {
                            float minimum = 10000;
                            int makinetercih = 0;
                            if (MECEXT[r] == 1)
                            {
                                makinetercih = MEC[sıra2 - 1]-1;
                                int makinedekiiş1 = MJ[r, makinetercih];
```

```
                                PST[r, sıra2 - 1, makinetercih] = MA[r, makinetercih] + SU[r, makinedekiiş1,
sıra2 - 1];
                            }
                            else
                            {
                                for (m = 0; m <= MS[r] - 1; m++)
                                {
                                    int makinedekiiş = MJ[r, m];
                                    PST[r, sıra2 - 1, m] = MA[r, m] + SU[r, makinedekiiş, sıra2 - 1];
                                    if (PST[r, sıra2 - 1, m] < minimum)
                                    {
                                        minimum = PST[r, sıra2 - 1, m];
                                        makinetercih = m;
                                    }
                                }
                            }
                            if (PST[r, sıra2 - 1, makinetercih] >= TST[sıra2 - 1])
                                ST[r, sıra2 - 1, makinetercih] = PST[r, sıra2 - 1, makinetercih];
                            else
                                ST[r, sıra2 - 1, makinetercih] = TST[sıra2 - 1];
                            C[r, sıra2 - 1, makinetercih] = ST[r, sıra2 - 1, makinetercih] + P[r, sıra2 - 1];
                            TST[sıra2 - 1] = C[r, sıra2 - 1, makinetercih];
                            MA[r, makinetercih] = TST[sıra2 - 1];
                            MJ[r, makinetercih] = (sıra2 - 1);
                            dosya.WriteLine("ST" + (r + 1) + "." + sıra2 + "." + (makinetercih + 1) + " : " +
ST[r, sıra2 - 1, makinetercih] + "  C" + (r + 1) + "." + sıra2 + "." + (makinetercih + 1) + " : " + C[r, sıra2
- 1, makinetercih]);
                            if (TST[sıra2 - 1] > maximum)
                                maximum = TST[sıra2 - 1];
                        }
                    }
                }
                for (int k = STJA[sıra - 1]; k <= FNJA[sıra - 1]; k++)
                {
                    TSTA[k] = maximum;
                }
                int makinetercih2 = 0;
                for (h = STJA[sıra - 1]; h <= FNJA[sıra - 1]; h++)
                {
                    for (s = 0; s <= S - 1; s++)
                    {
                        float minima = 10000;
                        for (m = 0; m <= MSA[s]-1; m++)
                        {
                            if (MAA[s, m] < minima)
                            {
                                minima = MAA[s, m];
                                makinetercih2 = m;
                            }
                        }
                        if (MAA[s, makinetercih2] >= TSTA[h])
                        {
                            STA[s, h, makinetercih2] = MAA[s, makinetercih2];
                        }
                        else
                        {
                            STA[s, h, makinetercih2] = TSTA[h];
                        }
                        F[s, h, makinetercih2] = STA[s, h, makinetercih2] + PA[s, h];
                        TSTA[h] = F[s, h, makinetercih2];
                        MAA[s, makinetercih2] = TSTA[h];
                        dosya.WriteLine("STA" + (s + 1) + "." + (h + 1) + "." + (makinetercih2 + 1) + " : " +
STA[s, h, makinetercih2] + "  F" + (s + 1) + "." + (h + 1) + "." + (makinetercih2 + 1) + " : " + F[s, h,
makinetercih2]);
                    }
                }
            }
            float Totcomp = 0;
            for (h = 0; h <= H - 1; h++)
            {
                float maximum = 0;
                for (m = 0; m <= MSA[S - 1] - 1; m++)
                {
                    if (F[S - 1, h, m] > maximum)
                    {
                        maximum = F[S - 1, h, m];
                    }
                }
                K[h] = maximum;
                Totcomp = Totcomp + K[h];
            }
            float Cmax = 0;
            for (h = 0; h <= H - 1; h++)
```

```
            {
                for (m = 0; m <= MSA[S - 1] - 1; m++)  // m.leri saydırmaya hiç gerek olmadan h döngüsü ile
K[h] lar içinden maximum Cmax olur.
                {
                    if (F[S - 1, h, m] > Cmax) // S'in döngüye girmesi sıkıntı olursa s=S s<=S'e kadar
döngükurulabilir.
                        Cmax = F[S - 1, h, m];
                }
            }
            int Uj = 0;
            for (h = 0; h <= H - 1; h++)
            {
                if (K[h] > DUE[h])
                {
                    T[h] = K[h] - DUE[h];
                    Uj = Uj + 1;
                }
            }
            float Tmax = 0;
            float Tottard = 0;
            for (h = 0; h <= H - 1; h++)
            {
                if (T[h] > Tmax)
                {
                    Tmax = T[h];
                }
                Tottard = Tottard + T[h];
            }
            dosya.WriteLine("ffs" + (S + R) + "j" + J + " Johnson Sonuçlar:");
            dosya.WriteLine("Cmax : " + Cmax);
            dosya.WriteLine("Tmax : " + Tmax);
            dosya.WriteLine("Toplam Uj : " + Uj);
            dosya.WriteLine("Totcomp : " + Totcomp);
            dosya.WriteLine("Tottard : " + Tottard);
            dosya.Close();
            Console.WriteLine("Dosya yazımı başarı ile tamamlandı..");
            Console.ReadLine();
        }
    }
}
```

En Az Boşluk Kuralı C# Kodları

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.IO;
using System.Text;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int r, s, m, j, a, J = ., h, H = ., R = ., M = ., S = ., A = .;
            float[,] P ={
                {.,.},
                {.,.}
                      }; // montaj öncesi işlem süreleri r,j
            float[,] PA ={
                          {.,.,.}
                      }; // montaj sonrası işlem süreleri s,h
            float[, ,] SU ={
                            {{.,.},{.,.},{.,.}},
                                      {{.,.},{.,.},{.,.}},
                                      {{.,.},{.,.},{.,.}}
                          }; // hazırlık zamanları r,i,j
            int[] MS = { .,. }; // montaj öncesi aşamalardaki makine sayıları r
            int[] MSA = { . }; // montaj sonrası aşamalardaki makine sayısı s
            int[] MEC = { .,.,. }; // machine eligibility constraints j
            int[] MECEXT = { .,. }; // montaj öncesi hangi aşamalarda MEC olduğunu söylüyor, 0 yok 1 var
            float[] DUE = { .,.,. }; // teslim tarihi h
            float[] T = new float[H]; // her işin gecikmesi
            int[] STJ = { .,. }; // a ürünü hangi j işinden başlıyor a
            int[] FNJ = { .,. }; // a ürünü hangi j işinde bitiyor a
            int[] STJA = { .,.}; // a ürünü montaj sonrasında hangi h işinden başlıyor
            int[] FNJA = { .,. }; // a ürünü montaj sonrasında hangi h işinde bitiyor
            float[] TST = new float[J]; // temp.start time, geçiici başlama zamanı
            float[] TSTA = new float[H]; // montaj sonrası geçici başlama zamanı
            float[, ,] PST = new float[R, J, M]; // probable start time, olası başlama zamanı
```

```
            float[, ,] ST = new float[R, J, M]; // r aşamasında j işinin m makinesinde başlama zamanı
            float[, ,] STA = new float[S, H, M]; // s aşamasında h işinin m makinesinde başlama zamanı
            float[, ,] C = new float[R, J, M]; // r aşamasında j işinin m makinesinde tamamlanma zamanı
            float[, ,] F = new float[S, H, M]; // s aşamasında h işinin m makinesinde tamamlanma zamanı
            float[] K = new float[H]; // her işin bitiş zamanı
            float[,] MA = new float[R, M]; // r aşamasında m makinesinin müsait olma zamanı
            float[,] MAA = new float[S, M]; // s aşamasında m makinesinin müsait olma zamanı
            int[,] MJ = new int[R, M]; // r aşamasında m makinesindeki son iş
            int[] B = { .,.,.,. }; // Ana sıralaması için değişken
            int[] D = { .,.,.,. }; // ürün içindeki işlerin sıralaması için değişken
            int[] atama = new int[A]; // en başta hepsi 0, sonrasında atanan işler için 1
            float[] SL = new float[H]; // her bir h işinin boşluğu
            float[] PTBA = new float[A]; // a ürününün tüm bileşenlerinin montaj öncesi toplam işlem süreleri
toplamı
            float[] PTH = new float[H]; // montaj sonrasındaki her bir h ürününün işlem süresi
            int[] HAB = new int[H]; // hangi h işi hangi hangi a ürününe ait
            float[] PTAJ = new float[J]; // j işinin montaj öncesi tüm işlem sürelerinin toplamı
            float[] maximus = new float[A]; // montaj sonrası atamalarda montaj öncesi işlerin en geç bitiş
zamanı
            StreamWriter dosya = new StreamWriter("d:\\ffs3j5MS.txt");

            for (r = 0; r <= R - 1; r++) // ilk etapta montaj öncesi tüm aşamalarda tüm makinelerde olmayan
bir iş (J+1) olarak tanımlanıyor. Bu olmayan işten olan başka herhabgi bir işe geçişte hazırlık süresi 0
            {
                for (m = 0; m <= MS[r] - 1; m++)
                {
                    MJ[r, m] = J;
                }
            }
            for (a = 0; a <= A - 1; a++)
            {
                atama[a] = 0;
            }
            for (a = 0; a <= A - 1; a++)
            {
                for (h = STJA[a]; h <= FNJA[a]; h++)
                {
                    HAB[h] = a;
                }
            }

            // Sıralama Kuralı
            for (a = 0; a <= A - 1; a++)
            {
                for (j = STJ[a]; j <= FNJ[a]; j++)
                {
                    for (r = 0; r <= R - 1; r++)
                        PTBA[a] = PTBA[a] + P[r, j];
                }
            }
            for (a = 0; a <= A - 1; a++)
            {
                for (h = STJA[a]; h <= FNJA[a]; h++)
                {
                    for (s = 0; s <= S - 1; s++)
                    {
                        PTH[h] = PTH[h] + PA[s, h];
                    }
                    SL[h] = DUE[h] - (PTBA[a] + PTH[h]);
                }
            }
            for (int y = 0; y < SL.Length - 1; y++) //BAR oranını küçükten büyüğe sıralama
            {
                for (int z = 1; z < SL.Length - y; z++)
                {
                    if (SL[z] < SL[z - 1])
                    {
                        float gecici1 = SL[z - 1];
                        int gecici2 = B[z - 1];
                        SL[z - 1] = SL[z];
                        B[z - 1] = B[z];
                        SL[z] = gecici1;
                        B[z] = gecici2;
                    }
                }
            }

            // makinelere atama
            for (h = 0; h <= H - 1; h++)
            {
                for (j = 0; j <= J - 1; j++) // j iş sıralaması her üründe 0'dan başlamayıp devam ettiğinden
PTAJ sıralamasında sıkıntı çıkarıyor, o yüzden başta hepsini 0'ladık
                    PTAJ[j] = 0;
```

```
                int sıra = B[h]; // sıra değişkenine SLack'ı küçükten büyüğü sıralarken aynı şekilde
sıralanmış B[] dizisindeki ilk ürünü atıyoruz.
                int sıra3 = HAB[sıra-1]; // h işinin ait olduğu a ürününü sıra3 değişkenine atıyoruz.
                if (atama[sıra3] == 0)
                {
                    atama[sıra3] = 1;
                    for (j = STJ[sıra3]; j <= FNJ[sıra3]; j++) // bir işin aşamalardaki toplam işlem süresi
PTaj döngüsü
                    {
                        for (r = 0; r <= R - 1; r++)
                        {
                            PTAJ[j] = PTAJ[j] + P[r, j];
                        }
                    }
                    for (int y = 0; y < D.Length - 1; y++) // PTAJ'leri sıraya dizmek için kullanılacak D
dizisinin her yeni üründe bştan sıralamak için
                    {
                        for (int z = 1; z < D.Length - y; z++)
                        {
                            if (D[z] < D[z - 1])
                            {
                                int gecici = D[z - 1];
                                D[z - 1] = D[z];
                                D[z] = gecici;
                            }
                        }
                    }
                    for (int y = 0; y < PTAJ.Length - 1; y++) // PTaj'leri büyükten küçüğe sıralama döngüsü
                    {
                        for (int z = 1; z < PTAJ.Length - y; z++)
                        {
                            if (PTAJ[z] > PTAJ[z - 1])
                            {
                                float gecici1 = PTAJ[z - 1];
                                int gecici2 = D[z - 1];
                                PTAJ[z - 1] = PTAJ[z];
                                D[z - 1] = D[z];
                                PTAJ[z] = gecici1;
                                D[z] = gecici2;
                            }
                        }
                    }
                    maximus[sıra3] = 0;
                    for (int k = 0; k <= FNJ[sıra3] - STJ[sıra3]; k++)
                    {
                        int sıra2 = D[k];
                        for (r = 0; r <= R - 1; r++) //işin her aşamalarına atama yapacak döngü
                        {
                            if (P[r, sıra2 - 1] > 0) // eğer aşamadaki işlem süresi 0 ise aşamayı atla
                            {
                                float minimum = 10000;
                                int makinetercih = 0;
                                if (MECEXT[r] == 1)
                                {
                                    makinetercih = MEC[sıra2 - 1] - 1;
                                    int makinedekiiş1 = MJ[r, makinetercih];
                                    PST[r, sıra2 - 1, makinetercih] = MA[r, makinetercih] + SU[r,
makinedekiiş1, sıra2 - 1];
                                }
                                else
                                {
                                    for (m = 0; m <= MS[r] - 1; m++)
                                    {
                                        int makinedekiiş = MJ[r, m];
                                        PST[r, sıra2 - 1, m] = MA[r, m] + SU[r, makinedekiiş, sıra2 - 1];
                                        if (PST[r, sıra2 - 1, m] < minimum)
                                        {
                                            minimum = PST[r, sıra2 - 1, m];
                                            makinetercih = m;
                                        }
                                    }
                                }
                                if (PST[r, sıra2 - 1, makinetercih] >= TST[sıra2 - 1])
                                    ST[r, sıra2 - 1, makinetercih] = PST[r, sıra2 - 1, makinetercih];
                                else
                                    ST[r, sıra2 - 1, makinetercih] = TST[sıra2 - 1];
                                C[r, sıra2 - 1, makinetercih] = ST[r, sıra2 - 1, makinetercih] + P[r, sıra2 -
1];
                                TST[sıra2 - 1] = C[r, sıra2 - 1, makinetercih];
                                MA[r, makinetercih] = TST[sıra2 - 1];
                                MJ[r, makinetercih] = (sıra2 - 1);
```

```
                                dosya.WriteLine("ST" + (r + 1) + "." + sıra2 + "." + (makinetercih + 1) + " :
" + ST[r, sıra2 - 1, makinetercih] + "  C" + (r + 1) + "." + sıra2 + "." + (makinetercih + 1) + " : " + C[r,
sıra2 - 1, makinetercih]);
                                if (TST[sıra2 - 1] > maximus[sıra3])
                                    maximus[sıra3] = TST[sıra2 - 1];
                            }
                        }
                    }
                }
                for (int k = STJA[sıra3]; k <= FNJA[sıra3]; k++)
                {
                    TSTA[k] = maximus[sıra3];
                }
                int makinetercih2 = 0;
                for (s = 0; s <= S - 1; s++)
                {
                    float minima = 10000;
                    for (m = 0; m <= MSA[s] - 1; m++)
                    {
                        if (MAA[s, m] < minima)
                        {
                            minima = MAA[s, m];
                            makinetercih2 = m;
                        }
                    }
                    if (MAA[s, makinetercih2] >= TSTA[sıra-1])
                    {
                        STA[s, sıra-1, makinetercih2] = MAA[s, makinetercih2];
                    }
                    else
                    {
                        STA[s, sıra-1, makinetercih2] = TSTA[sıra-1];
                    }
                    F[s, sıra-1, makinetercih2] = STA[s, sıra-1, makinetercih2] + PA[s, sıra-1];
                    TSTA[sıra-1] = F[s, sıra-1, makinetercih2];
                    MAA[s, makinetercih2] = TSTA[sıra-1];
                    dosya.WriteLine("STA" + (s + 1) + "." + sıra + "." + (makinetercih2 + 1) + " : " + STA[s,
sıra-1, makinetercih2] + "  F" + (s + 1) + "." + sıra + "." + (makinetercih2 + 1) + " : " + F[s, sıra-1,
makinetercih2]);
                }
            }
            float Totcomp = 0;
            for (h = 0; h <= H - 1; h++)
            {
                float maximum = 0;
                for (m = 0; m <= MSA[S - 1] - 1; m++)
                {
                    if (F[S - 1, h, m] > maximum)
                    {
                        maximum = F[S - 1, h, m];
                    }
                }
                K[h] = maximum;
                Totcomp = Totcomp + K[h];
            }
            float Cmax = 0;
            for (h = 0; h <= H - 1; h++)
            {
                for (m = 0; m <= MSA[S - 1] - 1; m++)  // m.leri saydırmaya hiç gerek olmadan h döngüsü ile
K[h] lar içinden maximum Cmax olur.
                {
                    if (F[S - 1, h, m] > Cmax) // S'in döngüye girmesi sıkıntı olursa s=S s<=S'e kadar
döngükurulabilir.
                        Cmax = F[S - 1, h, m];
                }
            }
            int Uj = 0;
            for (h = 0; h <= H - 1; h++)
            {
                if (K[h] > DUE[h])
                {
                    T[h] = K[h] - DUE[h];
                    Uj = Uj + 1;
                }
            }
            float Tmax = 0;
            float Tottard = 0;
            for (h = 0; h <= H - 1; h++)
            {
                if (T[h] > Tmax)
                {
                    Tmax = T[h];
                }
```

```
                Tottard = Tottard + T[h];
            }
            dosya.WriteLine("ffs" + (S + R) + "j" + J + " MS Sonuçlar:");
            dosya.WriteLine("Cmax : " + Cmax);
            dosya.WriteLine("Tmax : " + Tmax);
            dosya.WriteLine("Toplam Uj : " + Uj);
            dosya.WriteLine("Totcomp : " + Totcomp);
            dosya.WriteLine("Tottard : " + Tottard);
            dosya.Close();
            Console.WriteLine("Dosya yazımı başarı ile tamamlandı..");
            Console.ReadLine();
        }
    }
}
```

Partilere Ayrılmış En Az Boşluk Kuralı C# Kodları

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.IO;
using System.Text;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int r, s, m, j, a, J = 93, h, H = 50, R = 4, M = 6, S = 2, A = 20;
            float[,] P ={
                        .,.

                        }; // montaj öncesi işlem süreleri r,j
            float[,] PA ={
                          .,.
                      }; // montaj sonrası işlem süreleri s,h
            float[, ,] SU = new float[R, J + 1, J + 1];
            int[] MS = { .,. }; // montaj öncesi aşamalardaki makine sayıları r
            int[] MSA = { .,. }; // montaj sonrası aşamalardaki makine sayısı s
            int[] MEC = { .,. }; // machine eligibility constraints j
            int[] MECEXT = { .,. }; // montaj öncesi hangi aşamalarda MEC olduğunu söylüyor, 0 yok 1 var
            float[] DUE = { .,. }; // teslim tarihi h
            float[] T = new float[H]; // her işin gecikmesi
            int[] STJ = { .,. }; // a ürünü hangi j işinden başlıyor a
            int[] FNJ = { .,. }; // a ürünü hangi j işinde bitiyor a
            int[] STJA = { .,. }; // a ürünü montaj sonrasında hangi h işinden başlıyor
            int[] FNJA = { .,. }; // a ürünü montaj sonrasında hangi h işinde bitiyor
            float[] TST = new float[J]; // temp.start time, geçiici başlama zamanı
            float[] TSTA = new float[H]; // montaj sonrası geçici başlama zamanı
            float[, ,] PST = new float[R, J, M]; // probable start time, olası başlama zamanı
            float[, ,] ST = new float[R, J, M]; // r aşamasında j işinin m makinesinde başlama zamanı
            float[, ,] STA = new float[S, H, M]; // s aşamasında h işinin m makinesinde başlama zamanı
            float[, ,] C = new float[R, J, M]; // r aşamasında j işinin m makinesinde tamamlanma zamanı
            float[, ,] F = new float[S, H, M]; // s aşamasında h işinin m makinesinde tamamlanma zamanı
            float[] K = new float[H]; // her işin bitiş zamanı
            float[,] MA = new float[R, M]; // r aşamasında m makinesinin müsait olma zamanı
            float[,] MAA = new float[S, M]; // s aşamasında m makinesinin müsait olma zamanı
            int[,] MJ = new int[R, M]; // r aşamasında m makinesindeki son iş
            int[] B = { .,. }; // Ana sıralaması için değişken
            int[] D = { .,. }; // ürün içindeki işlerin sıralaması için değişken
            int[] atama = new int[A]; // en başta hepsi 0, sonrasında atanan işler için 1
            float[] SL = new float[H]; // her bir h işinin boşluğu
            float[] PTBA = new float[A]; // a ürününün tüm bileşenlerinin montaj öncesi toplam işlem süreleri
toplamı
            float[] PTH = new float[H]; // montaj sonrasındaki her bir h ürününün işlem süresi
            int[] HAB = new int[H]; // hangi h işi hangi hangi a ürününe ait
            int[] SET = { .,. }; // i işlerinin setup durumlarını gösteriyor, aynı rakamlar arasında setup
yok, 0 ile diğerleri arasında yok, sadece 1 ve 2 arasında hazırlık var.
            float[] PTAJ = new float[J]; // j işinin montaj öncesi tüm işlem sürelerinin toplamı
            float[] maximus = new float[A]; // montaj sonrası atamalarda montaj öncesi işlerin en geç bitiş
zamanı
            StreamWriter dosya = new StreamWriter("d:\\ffs6j93MS.txt");

            for (r = 1; r <= R - 1; r++)
            {
                for (int i = 0; i <= J; i++)
                {
                    for (j = 0; j <= J; j++)
                    {
                        SU[r, i, j] = 0;
```

```
                    }
                }
            }
            for (int i = 0; i <= J - 1; i++)
            {
                for (j = 0; j <= J - 1; j++)
                {
                    if (SET[i] + SET[j] == 3)
                        SU[1, i, j] = 0.5f;
                }
            }
            for (r = 0; r <= R - 1; r++) // ilk etapta montaj öncesi tüm aşamalarda tüm makinelerde olmayan
bir iş (J+1) olarak tanımlanıyor. Bu olmayan işten olan başka herhabgi bir işe geçişte hazırlık süresi 0
            {
                for (m = 0; m <= MS[r] - 1; m++)
                {
                    MJ[r, m] = J;
                }
            }
            for (a = 0; a <= A - 1; a++)
            {
                atama[a] = 0;
            }
            for (a = 0; a <= A - 1; a++)
            {
                for (h = STJA[a]; h <= FNJA[a]; h++)
                {
                    HAB[h] = a;
                }
            }

            // Sıralama Kuralı
            for (a = 0; a <= A - 1; a++)
            {
                for (j = STJ[a]; j <= FNJ[a]; j++)
                {
                    for (r = 0; r <= R - 1; r++)
                        PTBA[a] = PTBA[a] + P[r, j];
                }
            }
            for (a = 0; a <= A - 1; a++)
            {
                for (h = STJA[a]; h <= FNJA[a]; h++)
                {
                    for (s = 0; s <= S - 1; s++)
                    {
                        PTH[h] = PTH[h] + PA[s, h];
                    }
                    SL[h] = DUE[h] - (PTBA[a] + PTH[h]);
                }
            }
            for (int y = 0; y < SL.Length - 1; y++) //BAR oranını küçükten büyüğe sıralama
            {
                for (int z = 1; z < SL.Length - y; z++)
                {
                    if (SL[z] < SL[z - 1])
                    {
                        float gecici1 = SL[z - 1];
                        int gecici2 = B[z - 1];
                        SL[z - 1] = SL[z];
                        B[z - 1] = B[z];
                        SL[z] = gecici1;
                        B[z] = gecici2;
                    }
                }
            }

            // makinelere atama
            for (h = 0; h <= H - 1; h++)
            {
                for (j = 0; j <= J - 1; j++) // j iş sıralaması her üründe 0'dan başlamayıp devam ettiğinden
PTAJ sıralamasında sıkıntı çıkarıyor, o yüzden başta hepsini 0'ladık
                    PTAJ[j] = 0;
                int sıra = B[h]; // sıra değişkenine SLack'ı küçükten büyüğü sıralarken aynı şekilde
sıralanmış B[] dizisindeki ilk ürünü atıyoruz.
                int sıra3 = HAB[sıra - 1]; // h işinin ait olduğu a ürününü sıra3 değişkenine atıyoruz.
                if (atama[sıra3] == 0)
                {
                    atama[sıra3] = 1;
                    for (j = STJ[sıra3]; j <= FNJ[sıra3]; j++) // bir işin aşamalardaki toplam işlem süresi
PTaj döngüsü
                    {
                        for (r = 0; r <= R - 1; r++)
```

```
                    {
                        PTAJ[j] = PTAJ[j] + P[r, j];
                    }
                }
                for (int y = 0; y < D.Length - 1; y++) // PTAJ'leri sıraya dizmek için kullanılacak D
dizisinin her yeni üründe bştan sıralamak için
                {
                    for (int z = 1; z < D.Length - y; z++)
                    {
                        if (D[z] < D[z - 1])
                        {
                            int gecici = D[z - 1];
                            D[z - 1] = D[z];
                            D[z] = gecici;
                        }
                    }
                }
                for (int y = 0; y < PTAJ.Length - 1; y++) // PTaj'leri büyükten küçüğe sıralama döngüsü
                {
                    for (int z = 1; z < PTAJ.Length - y; z++)
                    {
                        if (PTAJ[z] > PTAJ[z - 1])
                        {
                            float gecici1 = PTAJ[z - 1];
                            int gecici2 = D[z - 1];
                            PTAJ[z - 1] = PTAJ[z];
                            D[z - 1] = D[z];
                            PTAJ[z] = gecici1;
                            D[z] = gecici2;
                        }
                    }
                }
                maximus[sıra3] = 0;
                for (int k = 0; k <= FNJ[sıra3] - STJ[sıra3]; k++)
                {
                    int sıra2 = D[k];
                    for (r = 0; r <= R - 1; r++) //işin her aşamalarına atama yapacak döngü
                    {
                        if (P[r, sıra2 - 1] > 0) // eğer aşamadaki işlem süresi 0 ise aşamayı atla
                        {
                            float minimum = 10000;
                            int makinetercih = 0;
                            if (MECEXT[r] == 1)
                            {
                                makinetercih = MEC[sıra2 - 1] - 1;
                                int makinedekiiş1 = MJ[r, makinetercih];
                                PST[r, sıra2 - 1, makinetercih] = MA[r, makinetercih] + SU[r,
makinedekiiş1, sıra2 - 1];
                            }
                            else
                            {
                                for (m = 0; m <= MS[r] - 1; m++)
                                {
                                    int makinedekiiş = MJ[r, m];
                                    PST[r, sıra2 - 1, m] = MA[r, m] + SU[r, makinedekiiş, sıra2 - 1];
                                    if (PST[r, sıra2 - 1, m] < minimum)
                                    {
                                        minimum = PST[r, sıra2 - 1, m];
                                        makinetercih = m;
                                    }
                                }
                            }
                            if (PST[r, sıra2 - 1, makinetercih] >= TST[sıra2 - 1])
                                ST[r, sıra2 - 1, makinetercih] = PST[r, sıra2 - 1, makinetercih];
                            else
                                ST[r, sıra2 - 1, makinetercih] = TST[sıra2 - 1];
                            C[r, sıra2 - 1, makinetercih] = ST[r, sıra2 - 1, makinetercih] + P[r, sıra2 -
1];
                            TST[sıra2 - 1] = C[r, sıra2 - 1, makinetercih];
                            MA[r, makinetercih] = TST[sıra2 - 1];
                            MJ[r, makinetercih] = (sıra2 - 1);
                            dosya.WriteLine("ST" + (r + 1) + "." + sıra2 + "." + (makinetercih + 1) + " :
" + ST[r, sıra2 - 1, makinetercih] + "  C" + (r + 1) + "." + sıra2 + "." + (makinetercih + 1) + " : " + C[r,
sıra2 - 1, makinetercih]);
                            if (TST[sıra2 - 1] > maximus[sıra3])
                                maximus[sıra3] = TST[sıra2 - 1];
                        }
                    }
                }
            }
            for (int k = STJA[sıra3]; k <= FNJA[sıra3]; k++)
            {
                TSTA[k] = maximus[sıra3];
```

```
                }
                int makinetercih2 = 0;
                for (s = 0; s <= S - 1; s++)
                {
                    float minima = 10000;
                    for (m = 0; m <= MSA[s] - 1; m++)
                    {
                        if (MAA[s, m] < minima)
                        {   minima = MAA[s, m];
                            makinetercih2 = m; }
                    }
                    if (MAA[s, makinetercih2] >= TSTA[sıra - 1])
                    {
                        STA[s, sıra - 1, makinetercih2] = MAA[s, makinetercih2];
                    }
                    else
                    {
                        STA[s, sıra - 1, makinetercih2] = TSTA[sıra - 1];
                    }
                    F[s, sıra - 1, makinetercih2] = STA[s, sıra - 1, makinetercih2] + PA[s, sıra - 1];
                    TSTA[sıra - 1] = F[s, sıra - 1, makinetercih2];
                    MAA[s, makinetercih2] = TSTA[sıra - 1];
                    dosya.WriteLine("STA" + (s + 1) + "." + sıra + "." + (makinetercih2 + 1) + " : " + STA[s,
sıra - 1, makinetercih2] + "  F" + (s + 1) + "." + sıra + "." + (makinetercih2 + 1) + " : " + F[s, sıra - 1,
makinetercih2]);
                }
            }
            float Totcomp = 0;
            for (h = 0; h <= H - 1; h++)
            {
                float maximum = 0;
                for (m = 0; m <= MSA[S - 1] - 1; m++)
                {
                    if (F[S - 1, h, m] > maximum)
                    {
                        maximum = F[S - 1, h, m];
                    }
                }
                K[h] = maximum;
                Totcomp = Totcomp + K[h];
            }
            float Cmax = 0;
            for (h = 0; h <= H - 1; h++)
            {
                for (m = 0; m <= MSA[S - 1] - 1; m++)  // m.leri saydırmaya hiç gerek olmadan h döngüsü ile
K[h] lar içinden maximum Cmax olur.
                {
                    if (F[S - 1, h, m] > Cmax) // S'in döngüye girmesi sıkıntı olursa s=S s<=S'e kadar
döngükurulabilir.
                        Cmax = F[S - 1, h, m];
                }
            }
            int Uj = 0;
            for (h = 0; h <= H - 1; h++)
            {
                if (K[h] > DUE[h])
                {
                    T[h] = K[h] - DUE[h];
                    Uj = Uj + 1;
                }
            }
            float Tmax = 0;
            float Tottard = 0;
            for (h = 0; h <= H - 1; h++)
            {
                if (T[h] > Tmax)
                {
                    Tmax = T[h];
                }
                Tottard = Tottard + T[h];
            }
            dosya.WriteLine("ffs" + (S + R) + "j" + J + " MS Sonuçlar:");
            dosya.WriteLine("Cmax : " + Cmax);
            dosya.WriteLine("Tmax : " + Tmax);
            dosya.WriteLine("Toplam Uj : " + Uj);
            dosya.WriteLine("Totcomp : " + Totcomp);
            dosya.WriteLine("Tottard : " + Tottard);
            dosya.Close();
            Console.WriteLine("Dosya yazımı başarı ile tamamlandı..");
            Console.ReadLine();
        }
    }
}
```

EK-4: TEST PROBLEMLERİ

FFS3J5 problemi

3 aşama (2 montaj öncesi ve 1 montaj), 2 ürüne ait, montaj sonrasında 4 işe dönüşen 5 iş, 1. aşamada sıra bağımlı hazırlık zamanları, 2. aşamada makine uygunluk kısıtları.

Tablo 29: FFS3J5 problemi işlem süreleri ve teslim tarihleri (dk.)

		Montaj öncesi aşamalar		Montaj		Teslim Tarihleri	
Ürün	İş	1. aşama	2. aşama	3. aşama		İş	Tarih(dk.)
A	1	13	-	1	42	1	56
	2	8	16	2	42	2	-
	3	20	13				
B	4	-	5	3	28	3	31
	5	12	6	4	28	4	52

Tablo 30: FFS3J5 problemi 1. aşamadaki sıra bağımlı hazırlık süreleri (dk.)

İşe / İşten	1	2	3	4	5
1		2	2		8
2	2		2		8
3	2	2			8
4					
5	8	8	8		

Tablo 31: FFS3J5 problemi aşamalardaki makine sayıları

Aşama	1	2	3
Makine Sayısı	3	4	4

Tablo 32: FFS3J5 problemi 2. aşamadaki makine uygunluk kısıtları

İş	1	2	3	4	5
İşlem göreceği makine	-	2	1	4	3

FFS3J15 problemi

3 aşama (2 montaj öncesi ve 1 montaj), 5 ürüne ait, montaj sonrasında 10 işe dönüşen 15 iş, 1. aşamada sıra bağımlı hazırlık zamanları, 2. aşamada makine uygunluk kısıtları.

Tablo 33: FFS3J15 problemi işlem süreleri ve teslim tarihleri (dk.)

		Montaj Öncesi Aşamalar		Montaj		Teslim Tarihleri	
Ürün	İş	1. aşama	2. aşama		3. aşama	İş	Tarih(dk.)
A	1	11	11	1	21	1	29
	2	8	-	2	21	2	60
	3	18	17				
B	4	18	10	3	41	3	-
	5	-	12	4	41	4	-
C	6	17	7	5	38	5	29
	7	8	18	6	38	6	-
	8	10	-				
D	9	13	-	7	21	7	49
	10	14	8	8	21	8	-
	11	-	19				
E	12	-	17				
	13	11	10	9	40	9	72
	14	15	17	10	40	10	-
	15	18	-				

Tablo 34: FFS3J15 problemi 1. aşamadaki sıra bağımlı hazırlık süreleri (dk.)

İşe / İşten	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1		2	2	8		8	8	8	8	8			8	8	8
2	2		2	8		8	8	8	8	8			8	8	8
3	2	2		8		8	8	8	8	8			8	8	8
4	8	8	8			8	8	8	8	8			8	8	8
5															
6	8	8	8	8			2	2	8	8			8	8	8
7	8	8	8	8		2		2	8	8			8	8	8
8	8	8	8	8		2	2		8	8			8	8	8
9	8	8	8	8		8	8	8		2			8	8	8
10	8	8	8	8		8	8	8	2				8	8	8
11															
12															
13	8	8	8	8		8	8	8	8	8				2	2
14	8	8	8	8		8	8	8	8	8			2		2
15	8	8	8	8		8	8	8	8	8			2	2	

Tablo 35: FFS3J15 problemi aşamalardaki makine sayıları

Aşama	1	2	3
Makine Sayısı	5	4	5

Tablo 36: FFS3J15 problemi 2. aşamadaki makine uygunluk kısıtları

İş	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
İşlem göreceği makine	2	-	2	3	4	4	3	-	-	2	1	3	1	3	-

FFS3J30 problemi

3 aşama (2 montaj öncesi ve 1 montaj), 10 ürüne ait, montaj sonrasında 20 işe dönüşen 30 iş, 1. aşamada sıra bağımlı hazırlık zamanları, 2. aşamada makine uygunluk kısıtları.

Tablo 37: FFS3J30 problemi işlem süreleri ve teslim tarihleri (dk.)

		Montaj öncesi aşamalar		Montaj		Teslim Tarihleri	
Ürün	İş	1. aşama	2. aşama	3. aşama		İş	Tarih(dk.)
A	1	-	19	1	28	1	69
	2	17	8	2	28	2	-
	3	5	13				
B	4	18	17	3	35	3	78
	5	12	-	4	35	4	42
C	6	18	11	5	28	5	37
	7	-	18	6	28	6	-
	8	16	5				
D	9	14	18	7	47	7	97
	10	15	-	8	47	8	43
	11	15	18				
E	12	14	14	9	21	9	41
	13	14	7	10	21	10	-
	14	14	12				
F	15	15	20	11	21	11	46
	16	18	12	12	21	12	34
	17	10	19				
G	18	8	-	13	47	13	86
	19	9	-	14	47	14	82
	20	19	19				
H	21	-	9	15	49	15	59
	22	15	19	16	49	16	-
	23	10	-				
I	24	-	7	17	42	17	59
	25	-	6	18	42	18	-
	26	17	-				
J	27	8	5				
	28	-	6	19	35	19	55
	29	9	11	20	35	20	-
	30	-	6				

Tablo 38: FFS3J30 problemi aşamalardaki makine sayıları

Aşama	1	2	3
Makine Sayısı	3	4	6

Tablo 39: FFS3J30 problemi 2. aşamadaki makine uygunluk kısıtları

İş	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
İşlem göreceği makine	3	2	3	2	-	2	3	1	1	-	2	4	3	2	3
İş	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
İşlem göreceği makine	3	1	-	-	1	2	4	-	4	2	-	4	1	4	2

Tablo 40: FFS3J30 problemi 1. aşamadaki sıra bağımlı hazırlık süreleri (dk.)

İşe / İşten	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1																														
2			2	8	8	8		8	8	8	8	8	8	8	8	8	8	8	8	8		8	8			8	8		8	
3		2		8	8	8		8	8	8	8	8	8	8	8	8	8	8	8	8		8	8			8	8		8	
4		8	8		2	8		8	8	8	8	8	8	8	8	8	8	8	8	8		8	8			8	8		8	
5		8	8	2		8		8	8	8	8	8	8	8	8	8	8	8	8	8		8	8			8	8		8	
6		8	8	8	8			2	8	8	8	8	8	8	8	8	8	8	8	8		8	8			8	8		8	
7																														
8		8	8	8	8	2			8	8	8	8	8	8	8	8	8	8	8	8		8	8			8	8		8	
9		8	8	8	8	8		8		2	2	8	8	8	8	8	8	8	8	8		8	8			8	8		8	
10		8	8	8	8	8		8	2		2	8	8	8	8	8	8	8	8	8		8	8			8	8		8	
11		8	8	8	8	8		8	2	2		8	8	8	8	8	8	8	8	8		8	8			8	8		8	
12		8	8	8	8	8		8	8	8	8		2	2	8	8	8	8	8	8		8	8			8	8		8	
13		8	8	8	8	8		8	8	8	8	2		2	8	8	8	8	8	8		8	8			8	8		8	
14		8	8	8	8	8		8	8	8	8	2	2		8	8	8	8	8	8		8	8			8	8		8	
15		8	8	8	8	8		8	8	8	8	8	8	8		2	2	8	8	8		8	8			8	8		8	
16		8	8	8	8	8		8	8	8	8	8	8	8	2		2	8	8	8		8	8			8	8		8	
17		8	8	8	8	8		8	8	8	8	8	8	8	2	2		8	8	8		8	8			8	8		8	
18		8	8	8	8	8		8	8	8	8	8	8	8	8	8	8		2	2		8	8			8	8		8	
19		8	8	8	8	8		8	8	8	8	8	8	8	8	8	8	2		2		8	8			8	8		8	
20		8	8	8	8	8		8	8	8	8	8	8	8	8	8	8	2	2			8	8			8	8		8	
21																														
22		8	8	8	8	8		8	8	8	8	8	8	8	8	8	8	8	8	8			2			8	8		8	
23		8	8	8	8	8		8	8	8	8	8	8	8	8	8	8	8	8	8		2				8	8		8	
24																														
25																														
26		8	8	8	8	8		8	8	8	8	8	8	8	8	8	8	8	8	8		8	8				8		8	
27		8	8	8	8	8		8	8	8	8	8	8	8	8	8	8	8	8	8		8	8			8			2	
28																														
29		8	8	8	8	8		8	8	8	8	8	8	8	8	8	8	8	8	8		8	8			8	2			
30																														

FFS6J5 problemi

6 aşama (4 montaj öncesi ve 2 montaj ve sonrası), 2 ürüne ait, montaj sonrasında 4 işe dönüşen 5 iş, 1. aşamada sıra bağımlı hazırlık zamanları, 2. aşamada makine uygunluk kısıtları.

Tablo 41: FFS6J5 problemi işlem süreleri ve teslim tarihleri (dk.)

		Montaj öncesi aşamalar				Montaj ve sonrası			Teslim Tarihi	
Ürün	İş	1. aşama	2. aşama	3. aşama	4. aşama		5. aşama	6. aşama	İş	Tarih(dk.)
A	1	9	19	10	12	1	49	40	1	153
	2	11	-	15	18	2	49	40	2	-
	3	8	16	-	12					
B	4	16	20	15	12	3	44	29	3	134
	5	8	16	10	13	4	44	29	4	162

Tablo 42: FFS6J5 problemi 1. aşamadaki sıra bağımlı hazırlık süreleri (dk.)

İşe / İşten	1	2	3	4	5
1		2	2	8	8
2	2		2	8	8
3	2	2		8	8
4	8	8	8		2
5	8	8	8	2	

Tablo 43: FFS6J5 problemi aşamalardaki makine sayıları

Aşama	1	2	3	4	5	6
Makine Sayısı	3	3	2	3	4	4

Tablo 44: FFS6J5 problemi 2. aşamadaki makine uygunluk kısıtları

İş	1	2	3	4	5
İşlem göreceği makine	3	-	3	1	2

FFS6J15 problemi

6 aşama (4 montaj öncesi ve 2 montaj ve sonrası), 5 ürüne ait, montaj sonrasında 10 işe dönüşen 15 iş, 1. aşamada sıra bağımlı hazırlık zamanları, 2. aşamada makine uygunluk kısıtları.

Tablo 45: FFS6J15 problemi işlem süreleri ve teslim tarihleri (dk.)

		Montaj Öncesi Aşamalar				Montaj ve sonrası			Teslim Tarihleri	
Ürün	İş	1. aşama	2. aşama	3. aşama	4. aşama		5. aşama	6. aşama	İş	Tarih(dk.)
A	1	12	9	-	13	1	33	49	1	117
	2	11	7	-	19	2	33	49	2	87
	3	9	11	9	14					
B	4	16	19	-	16	3	50	29	3	-
	5	-	-	10	8	4	50	29	4	-
C	6	8	19	9	11	5	24	40	5	105
	7	20	9	6	12	6	24	40	6	74
	8	11	-	16	10					
D	9	18	6	18	9	7	21	34	7	101
	10	8	16	7	5	8	21	34	8	-
	11	-	16	16	20					
E	12	-	14	11	18					
	13	16	10	7	10	9	36	42	9	170
	14	13	9	17	14	10	36	42	10	-
	15	19	18	-	11					

Tablo 46: FFS6J15 problemi 1. aşamadaki sıra bağımlı hazırlık süreleri

İşe / İşten	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1		2	2	8		8	8	8	8	8			8	8	8
2	2		2	8		8	8	8	8	8			8	8	8
3	2	2		8		8	8	8	8	8			8	8	8
4	8	8	8			8	8	8	8	8			8	8	8
5															
6	8	8	8	8			2	2	8	8			8	8	8
7	8	8	8	8		2		2	8	8			8	8	8
8	8	8	8	8		2	2		8	8			8	8	8
9	8	8	8	8		8	8	8		2			8	8	8
10	8	8	8	8		8	8	8	2				8	8	8
11															
12															
13	8	8	8	8		8	8	8	8	8				2	2
14	8	8	8	8		8	8	8	8	8			2		2
15	8	8	8	8		8	8	8	8	8			2	2	

Tablo 47: FFS6J15 problemi aşamalardaki makine sayıları

Aşama	1	2	3	4	5	6
Makine Sayısı	4	6	6	3	6	6

Tablo 48: FFS6J15 problemi 2. aşamadaki makine uygunluk kısıtları

İş	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
İşlem göreceği makine	5	4	6	5	-	1	5	-	3	4	2	3	5	4	6

FFS6J30 problemi

6 aşama (4 montaj öncesi ve 2 montaj ve sonrası), 10 ürüne ait, montaj sonrasında 20 işe dönüşen 30 iş, 1. aşamada sıra bağımlı hazırlık zamanları, 2. aşamada makine uygunluk kısıtları.

Tablo 49: FFS6J30 problemi işlem süreleri ve teslim tarihleri (dk.)

	Montaj öncesi aşamalar					Montaj ve sonrası			Teslim Tarihleri	
Ürün	İş	1. aşama	2. aşama	3. aşama	4. aşama		5. aşama	6. aşama	İş	Tarih(dk.)
A	1	-	20	-	14	1	35	30	1	-
	2	9	15	18	14	2	35	30	2	-
	3	8	14	13	10					
B	4	12	7	16	5	3	22	33	3	131
	5	8	18	5	14	4	22	33	4	-
C	6	17	15	-	7	5	41	31	5	137
	7	-	9	12	6	6	41	31	6	-
	8	17	11	6	8					
D	9	19	20	-	19	7	33	38	7	103
	10	20	15	10	14	8	33	38	8	-
	11	12	19	7	20					
E	12	6	14	8	5	9	23	48	9	122
	13	9	13	-	13	10	23	48	10	-
	14	18	14	10	-					
F	15	7	16	7	11	11	49	37	11	97
	16	7	-	8	6	12	49	37	12	118
	17	8	12	12	13					
G	18	15	-	7	16	13	25	21	13	122
	19	19	18	20	12	14	25	21	14	-
	20	16	12	5	7					
H	21	-	11	10	8	15	34	48	15	-
	22	12	18	14	7	16	34	48	16	-
	23	19	10	14	13					
I	24	-	19	9	12	17	31	21	17	144
	25	-	13	18	19	18	31	21	18	-
	26	9	10	-	14					
J	27	16	9	-	19					
	28	-	5	14	7	19	30	29	19	70
	29	7	11	14	13	20	30	29	20	137
	30	-	10	6	10					

Tablo 50: FFS6J30 problemi aşamalardaki makine sayıları

Aşama	1	2	3	4	5	6
Makine Sayısı	3	3	4	4	6	6

Tablo 51: FFS6J30 problemi 2. aşamadaki makine uygunluk kısıtları

İş	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
İşlem göreceği makine	2	2	3	3	3	3	1	1	3	3	3	2	1	1	2
İş	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
İşlem göreceği makine	-	2	-	2	3	1	2	2	1	1	2	1	2	3	3

Tablo 52: FFS6J30 problemi 1. aşamadaki sıra bağımlı hazırlık süreleri (dk.)

İşe / İşten	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1																														
2			2	8	8	8		8	8	8	8	8	8	8	8	8	8	8	8	8		8	8			8	8		8	
3		2		8	8	8		8	8	8	8	8	8	8	8	8	8	8	8	8		8	8			8	8		8	
4		8	8		2	8		8	8	8	8	8	8	8	8	8	8	8	8	8		8	8			8	8		8	
5		8	8	2		8		8	8	8	8	8	8	8	8	8	8	8	8	8		8	8			8	8		8	
6		8	8	8	8			2	8	8	8	8	8	8	8	8	8	8	8	8		8	8			8	8		8	
7																														
8		8	8	8	8	2			8	8	8	8	8	8	8	8	8	8	8	8		8	8			8	8		8	
9		8	8	8	8	8		8		2	2	8	8	8	8	8	8	8	8	8		8	8			8	8		8	
10		8	8	8	8	8		8	2		2	8	8	8	8	8	8	8	8	8		8	8			8	8		8	
11		8	8	8	8	8		8	2	2		8	8	8	8	8	8	8	8	8		8	8			8	8		8	
12		8	8	8	8	8		8	8	8	8		2	2	8	8	8	8	8	8		8	8			8	8		8	
13		8	8	8	8	8		8	8	8	8	2		2	8	8	8	8	8	8		8	8			8	8		8	
14		8	8	8	8	8		8	8	8	8	2	2		8	8	8	8	8	8		8	8			8	8		8	
15		8	8	8	8	8		8	8	8	8	8	8	8		2	2	8	8	8		8	8			8	8		8	
16		8	8	8	8	8		8	8	8	8	8	8	8	2		2	8	8	8		8	8			8	8		8	
17		8	8	8	8	8		8	8	8	8	8	8	8	2	2		8	8	8		8	8			8	8		8	
18		8	8	8	8	8		8	8	8	8	8	8	8	8	8	8		2	2		8	8			8	8		8	
19		8	8	8	8	8		8	8	8	8	8	8	8	8	8	8	2		2		8	8			8	8		8	
20		8	8	8	8	8		8	8	8	8	8	8	8	8	8	8	2	2			8	8			8	8		8	
21																														
22		8	8	8	8	8		8	8	8	8	8	8	8	8	8	8	8	8	8			2			8	8		8	
23		8	8	8	8	8		8	8	8	8	8	8	8	8	8	8	8	8	8		2				8	8		8	
24																														
25																														
26		8	8	8	8	8		8	8	8	8	8	8	8	8	8	8	8	8	8		8	8				8		8	
27		8	8	8	8	8		8	8	8	8	8	8	8	8	8	8	8	8	8		8	8			8			2	
28																														
29		8	8	8	8	8		8	8	8	8	8	8	8	8	8	8	8	8	8		8	8			8	2			
30																														

FFS6J66 Tez Uygulama Problemi

Tablo 53: FFS6J66 problemi işlem süreleri ve teslim tarihleri

		Kesim Süresi	Bantlama Süresi	CNC Süresi	Delik Süresi		Montaj Süresi	Paket Süresi	İş	Teslim Tarihi
A	1	15,2	15,0		18,9	1	103,6	47,7	1	480
	2	9,7	6,2		11,7	2	103,6	47,7	2	960
	3	9,8	11,8		10,5	3	103,6	47,7	3	1440
	4	5,8	-		-	4	103,6	47,7	4	1440
	5	15,1	28,8		-	5	103,6	47,7	5	1440
	6	7,3	-		-	6	103,6	47,7	6	1440
B	7	12,0	4,3		7,7	7	51	58,4	7	1440
	8	10,5	14,5		13,7	8	51	58,4	8	1440
C	9	7,7	6,1		-	9	37,2	58,6	9	480
D	10	9,0	11,7		10,5	10	100,6	51,2	10	1440
	11	11,6	12,5		21,3	11	100,6	51,2	11	1440
	12	8,7	-		-	12	100,6	51,2	12	1440
	13	9,2	27,8		-	13	100,6	51,2	13	1440
	14	7,6	-		-	14	100,6	51,2	14	1440
	15	9,5	11,7		9,3	15	100,6	51,2	15	1440
E	16	20,7	21,9		8,1	16	79	71,6	16	1440
	17	15,2	10,7		19,3	17	79	71,6	17	1440
	18	9,5	3,0		8,1	18	79	71,6	18	1440
	19	17,3	9,0		32,1	19	79	71,6	19	1440
F	20	11,9	11,0		8,1	20	68,6	44,1	20	1440
	21	12,5	9,6		23,3	21	68,6	44,1	21	1440
	22	8,0	2,7		9,3	22	68,6	44,1	22	1440
	23	8,1	9,3		-	23	68,6	44,1	23	1440
	24	8,8	8,0		27,3					
G	25	9,0	11,7		10,5	24	110,9	50	24	1440
	26	11,9	12,5		21,3	25	110,9	50	25	1440
	27	8,5	-		-	26	110,9	50	26	1440
	28	8,9	27,8		-	27	110,9	50	27	1440
	29	7,6	-		-	28	110,9	50	28	1440
	30	9,1	11,5		9,3	29	110,9	50	29	1440
H	31	8,3	9,1		7,3	30	39,2	62,8	30	1440
	32	11,8	7,5		12,9	31	39,2	62,8	31	1440
I	33	14,5	16,8		12,3					
	34	24,6	16,2		20,1	32	60,4	98,2	32	480
	35	12,9	8,1		14,1	33	60,4	98,2	33	1440
	36	10,3	6,4		11,7	34	60,4	98,2	34	1440
	37	10,8	12,0		-					
	38	11,2	13,9		20,1					
J	39	15,1	11,5	-	26,1					
	40	9,8	-	51,2	11,7	35	45,5	65,9	35	960
	41	12,0	6,4	-	13,9	36	45,5	65,9	36	960
	42	9,5	6,9	-	9,0					
K	43	31,9	-	122,7	-					
	44	32,2	-	150,9	-	37	98,9	61	37	1440
	45	11,5	8,0	-	11,7	38	98,9	61	38	1440
	46	10,2	8,9	-	10,5	39	98,9	61	39	1440
	47	17,5	16,0	-	-					
L	48	6,9	-	77,7	-	40	25	30,2	40	1440
	49	7,1	-	93,6	-	41	25	30,2	41	1440
	50	10,4	5,6	-	12,3	42	25	30,2	42	1440
	51	7,6	3,1	-	8,1					
M	52	8,9	12,0		-					
	53	9,1	12,0		7,5	43	71,2	39,1	43	1440
	54	9,5	8,0		11,7	44	71,2	39,1	44	1440
	55	8,3	4,1		8,1	45	71,2	39,1	45	1440
	56	13,5	16,0		-					
	57	8,2	6,6		-					
N	58	22,2	15,2		9,3	46	29,6	48,1	46	960
	59	11,8	8,0		12,9	47	29,6	48,1	47	1440
	60	9,7	8,6		8,7	48	29,6	48,1	48	1440
O	61	8,4	4,8		6,5					
	62	7,5	4,7		6,9					
	63	11,2	11,4		-	49	69,6	74,5	49	960
	64	8,6	9,2		7,3	50	69,6	74,5	50	1440
	65	8,3	13,1		13,3					
	66	8,4	-		-					

Tablo 54: FFS6J66 problemi aşamalardaki makine sayıları

Aşama	1	2	3	4	5	6
Makine sayısı	2	2	1	2	6	5

Tablo 55: FFS6J66 problemi 2. aşamada makine uygunluk kriterleri

İş	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
İşlem göreceği makine	2	2	2	-	1	-	1	1	1	1	2	-	1	-	2
İş	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
İşlem göreceği makine	1	1	2	2	1	1	2	2	1	1	2	-	1	-	2
İş	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
İşlem göreceği makine	1	1	1	2	2	2	2	2	1	-	1	2	-	-	2
İş	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
İşlem göreceği makine	2	2	-	-	2	2	1	1	2	2	1	2	1	1	2
İş	61	62	63	64	65	66									
İşlem göreceği makine	1	2	2	1	2	-									

Tablo 56: FFS6J66 problemi 2. aşamada 1. makinede hazırlık süreleri

	5	7	8	9	10	13	16	17	20	21	24	25	28	31	32	33	39	41	52	53	56	58	59	61	64
5		0,5	0,5	0,5	0,5		0,5	0,5	0,5	0,5	0,5	0,5		0,5	0,5		0,5	0,5				0,5	0,5	0,5	
7	0,5					0,5							0,5			0,5			0,5	0,5	0,5				0,5
8	0,5					0,5							0,5			0,5			0,5	0,5	0,5				0,5
9	0,5					0,5							0,5			0,5			0,5	0,5	0,5				0,5
10	0,5					0,5							0,5			0,5			0,5	0,5	0,5				0,5
13		0,5	0,5	0,5	0,5		0,5	0,5	0,5	0,5	0,5	0,5		0,5	0,5		0,5	0,5				0,5	0,5	0,5	
16	0,5					0,5							0,5			0,5			0,5	0,5	0,5				0,5
17	0,5					0,5							0,5			0,5			0,5	0,5	0,5				0,5
20	0,5					0,5							0,5			0,5			0,5	0,5	0,5				0,5
21	0,5					0,5							0,5			0,5			0,5	0,5	0,5				0,5
24	0,5					0,5							0,5			0,5			0,5	0,5	0,5				0,5
25	0,5					0,5							0,5			0,5			0,5	0,5	0,5				0,5
28		0,5	0,5	0,5	0,5		0,5	0,5	0,5	0,5	0,5	0,5		0,5	0,5		0,5	0,5				0,5	0,5	0,5	
31	0,5					0,5							0,5			0,5			0,5	0,5	0,5				0,5
32	0,5					0,5							0,5			0,5			0,5	0,5	0,5				0,5
33		0,5	0,5	0,5	0,5		0,5	0,5	0,5	0,5	0,5	0,5		0,5	0,5		0,5	0,5				0,5	0,5	0,5	
39	0,5					0,5							0,5			0,5			0,5	0,5	0,5				0,5
41	0,5					0,5							0,5			0,5			0,5	0,5	0,5				0,5
52		0,5	0,5	0,5	0,5		0,5	0,5	0,5	0,5	0,5	0,5		0,5	0,5		0,5	0,5				0,5	0,5	0,5	
53		0,5	0,5	0,5	0,5		0,5	0,5	0,5	0,5	0,5	0,5		0,5	0,5		0,5	0,5				0,5	0,5	0,5	
56		0,5	0,5	0,5	0,5		0,5	0,5	0,5	0,5	0,5	0,5		0,5	0,5		0,5	0,5				0,5	0,5	0,5	
58	0,5					0,5							0,5			0,5			0,5	0,5	0,5				0,5
59	0,5					0,5							0,5			0,5			0,5	0,5	0,5				0,5
61	0,5					0,5							0,5			0,5			0,5	0,5	0,5				0,5
64		0,5	0,5	0,5	0,5		0,5	0,5	0,5	0,5	0,5	0,5		0,5	0,5		0,5	0,5				0,5	0,5	0,5	

Kesim İşlemi Sonrası Partilere Ayrılmış FFS3J5 problemi

Tablo 57: Partilere Ayrılmış FFS3J5 problemi işlem süreleri ve teslim tarihleri (dk.)

	Montaj öncesi aşamalar				Montaj		Teslim Tarihleri	
Ürün	İş	1. aşama		2. aşama		3. aşama	İş	Tarih(dk.)
A	1	13	1	-	1	42	1	56
	2	8	2	8				
	3	20	3	6,5				
			4	-	2	42	2	-
			5	8				
			6	6,5				
B	4	-	7	2,5	3	28	3	31
	5	12	8	3				
			9	2,5	4	28	4	52
			10	3				

Tablo 58: Partilere Ayrılmış FFS3J5 problemi 2. aşamadaki makine uygunluk kısıtları

İş	1	2	3	4	5	6	7	8	9	10
İşlem göreceği makine	-	2	1	-	2	1	4	3	4	3

Kesim İşlemi Sonrası Partilere Ayrılmış FFS3J15 problemi

Tablo 59: Partilere Ayrılmış FFS3J15 problemi işlem süreleri ve teslim tarihleri (dk.)

	Montaj Öncesi Aşamalar				Montaj		Teslim Tarihleri	
Ürün	İş	1. aşama		2. aşama		3. aşama	İş	Tarih(dk.)
A	1	11	1	5,5	1	21	1	29
	2	8	2	-				
	3	18	3	8,5				
			4	5,5	2	21	2	60
			5	-				
			6	8,5				
B	4	18	7	10	3	41	3	-
	5	-	8	12	4	41	4	-
C	6	17	9	3,5	5	38	5	29
	7	8	10	9				
	8	10	11	-				
			12	3,5	6	38	6	-
			13	9				
			14	-				
D	9	13	15	-	7	21	7	49
	10	14	16	4				
	11	-	17	4,5				
			18	-	8	21	8	-
			19	4				
			20	4,5				
E	12	-	21	8,5	9	40	9	72
	13	11	22	5				
	14	15	23	8,5				
	15	18	24	-				
			25	8,5	10	40	10	-
			26	5				
			27	8,5				
			28	-				

Tablo 60: Partilere Ayrılmış FFS3J15 problemi 2. aşamadaki makine uygunluk kısıtları

İş	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
İşlem göreceği makine	2	-	2	2	-	2	3	4	4	3	-	4	3	-	-
İş	16	17	18	19	20	21	22	23	24	25	26	27	28		
İşlem göreceği makine	2	1	-	2	1	3	1	3	-	3	1	3	-		

Kesim İşlemi Sonrası Partilere Ayrılmış FFS3J30 problemi

Tablo 61: Partilere Ayrılmış FFS3J30 problemi işlem süreleri ve teslim tarihleri (dk.)

	Montaj öncesi aşamalar				Montaj		Teslim Tarihleri	
Ürün	İş	1. aşama		2. aşama		3. aşama	İş	Tarih(dk.)
A	1	-	1	9,5	1	28	1	69
	2	17	2	4				
	3	5	3	6,5				
			4	9	2	28	2	-
			5	4,5				
			6	6,5				
B	4	18	7	8,5	3	35	3	78
	5	12	8	-				
			9	8,5	4	35	4	42
			10	-				
C	6	18	11	5,5	5	28	5	37
	7	-	12	9				
	8	16	13	2,5				
			14	5,5	6	28	6	-
			15	9				
			16	2,5				
D	9	14	17	9	7	47	7	97
	10	15	18	-				
	11	15	19	9				
			20	9	8	47	8	43
			21	-				
			22	9				
E	12	14	23	7	9	21	9	41
	13	14	24	3,5				
	14	14	25	6				
			26	7	10	21	10	-
			27	3,5				
			28	6				
F	15	15	29	10	11	21	11	46
	16	18	30	6				
	17	10	31	9,5				
			32	10	12	21	12	34
			33	6				
			34	9,5				
G	18	8	35	-	13	47	13	86
	19	9	36	-				
	20	19	37	9,5				
			38	-	14	47	14	82
			39	-				
			40	9,5				
H	21	-	41	4,5	15	49	15	59
	22	15	42	9,5				
	23	10	43	-				
			44	4,5	16	49	16	-
			45	9,5				
			46	-				
I	24	-	47	3,5	17	42	17	59
	25	-	48	3				
	26	17	49	-				
			50	3,5	18	42	18	-
			51	3				
			52	-				
J	27	8	53	2,5	19	35	19	55
	28	-	54	3				
	29	9	55	5,5				
	30	-	56	3				
			57	2,5	20	35	20	-
			58	3				
			59	5,5				
			60	3				

Tablo 62: Partilere Ayrılmış FFS3J30 problemi 2. aşamadaki makine uygunluk kısıtları

İş	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
İşlem göreceği makine	3	2	3	3	2	3	2	-	2	-	2	3	1	2	3
İş	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
İşlem göreceği makine	1	1	-	2	1	-	2	4	3	2	4	3	2	3	3
İş	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
İşlem göreceği makine	1	3	3	1	-	-	1	-	-	1	2	4	-	2	4
İş	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
İşlem göreceği makine	-	4	2	-	4	2	-	4	1	4	2	4	1	4	2

Kesim İşlemi Sonrası Partilere Ayrılmış FFS6J5 problemi

Tablo 63: Partilere Ayrılmış FFS6J5 problemi işlem süreleri ve teslim tarihleri (dk.)

Montaj öncesi aşamalar						Montaj ve sonrası			Teslim Tarihi	
Ürün	İş	1. aşama		2. aşama	3. aşama	4. aşama		5. aşama	6. aşama	İş	Tarih(dk.)
A	1	9	1	9,5	5	6	1	49	40	1	153
			2	-	7,5	9					
	2	11	3	8	-	6					
	3	8	4	9,5	5	6	2	49	40	2	-
			5	-	7,5	9					
			6	8	-	6					
B	4	16	7	10	7,5	6	3	44	29	3	134
			8	8	5	6,5					
	5	8	9	10	7,5	6	4	44	29	4	162
			10	8	5	6,5					

Tablo 64: Partilere Ayrılmış FFS6J5 problemi 2. aşamadaki makine uygunluk kısıtları

İş	1	2	3	4	5	6	7	8	9	10
İşlem göreceği makine	3	-	3	3	-	3	1	2	1	2

Kesim İşlemi Sonrası Partilere Ayrılmış FFS6J15 problemi

Tablo 65: Partilere Ayrılmış FFS6J15 problemi işlem süreleri ve teslim tarihleri (dk.)

Montaj Öncesi Aşamalar						Montaj ve sonrası			Teslim Tarihleri	
Ürün	İş	1. aşama	2. aşama	3. aşama	4. aşama		5. aşama	6. aşama	İş	Tarih(dk.)
A	1	12	4,5	-	6,5	1	33	49	1	117
			3,5	-	9,5					
	2	11	5,5	4,5	7					
	3	9	4,5	-	6,5	2	33	49	2	87
			3,5	-	9,5					
			5,5	4,5	7					
B	4	16	19	-	16	3	50	29	3	-
	5	-	-	10	8	4	50	29	4	-
C	6	8	9,5	4,5	5,5	5	24	40	5	105
			4,5	3	6					
	7	20	-	8	5					
	8	11	9,5	4,5	5,5	6	24	40	6	74
			4,5	3	6					
			-	8	5					
D	9	18	3	9	4,5	7	21	34	7	101
			8	3,5	2,5					
	10	8	8	8	10					
	11	-	3	9	4,5	8	21	34	8	-
			8	3,5	2,5					
			8	8	10					
E	12	-	7	5,5	9	9	36	42	9	170
			5	3,5	5					
	13	16	4,5	8,5	7					
	14	13	9	-	5,5					
	15	19	7	5,5	9	10	36	42	10	-
			5	3,5	5					
			4,5	8,5	7					
			9	-	5,5					

Tablo 66: Partilere Ayrılmış FFS6J15 problemi 2. aşamadaki makine uygunluk kısıtları

İş	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
İşlem göreceği makine	5	4	6	5	4	6	5	-	1	5	-	1	5	-	3
İş	16	17	18	19	20	21	22	23	24	25	26	27	28		
İşlem göreceği makine	4	2	3	4	2	3	5	4	6	3	5	4	6		

Kesim İşlemi Sonrası Partilere Ayrılmış FFS6J30 problemi

Tablo 67: Partilere Ayrılmış FFS6J30 problemi işlem süreleri ve teslim tarihleri (dk.)

	Montaj öncesi aşamalar						Montaj ve sonrası			Teslim Tarihleri	
Ürün	İş	1. aşama		2. aşama	3. aşama	4. aşama		5. aşama	6. aşama	İş	Tarih(dk.)
A	1	-	1	20	-	14	1	35	30	1	-
	2	9	2	15	18	14	2	35	30	2	-
	3	8	3	14	13	10					
B	4	12	4	3,5	8	2,5	3	22	33	3	131
	5	8	5	9	2,5	7					
			6	3,5	8	2,5	4	22	33	4	-
			7	9	2,5	7					
C	6	17	8	7,5	-	3,5	5	41	31	5	137
	7	-	9	4,5	6	3					
	8	17	10	5,5	3	4					
			11	7,5	-	3,5	6	41	31	6	-
			12	4,5	6	3					
			13	5,5	3	4					
D	9	19	14	10	-	9,5	7	33	38	7	103
	10	20	15	7,5	5	7					
	11	12	16	9,5	3,5	10					
			17	10	-	9,5	8	33	38	8	-
			18	7,5	5	7					
			19	9,5	3,5	10					
E	12	6	20	7	4	2,5	9	23	48	9	122
	13	9	21	6,5	-	6,5					
	14	18	22	7	5	-					
			23	7	4	2,5	10	23	48	10	-
			24	6,5	-	6,5					
			25	7	5	-					
F	15	7	26	8	3,5	5,5	11	49	37	11	97
	16	7	27	-	4	3					
	17	8	28	6	6	6,5					
			29	8	3,5	5,5	12	49	37	12	118
			30	-	4	3					
			31	6	6	6,5					
G	18	15	32	-	3,5	8	13	25	21	13	122
	19	19	33	9	10	6					
	20	16	34	6	2,5	3,5					
			35	-	3,5	8	14	25	21	14	-
			36	9	10	6					
			37	6	2,5	3,5					
H	21	-	38	11	10	8	15	34	48	15	-
	22	12	39	18	14	7	16	34	48	16	-
	23	19	40	10	14	13					
I	24	-	41	9,5	4,5	6	17	31	21	17	144
	25	-	42	6,5	9	9,5					
	26	9	43	5	-	7					
			44	9,5	4,5	6	18	31	21	18	-
			45	6,5	9	9,5					
			46	5	-	7					
J	27	16	47	4,5	-	9,5	19	30	29	19	70
	28	-	48	2,5	7	3,5					
	29	7	49	5,5	7	6,5					
	30	-	50	5	3	5					
			51	4,5	-	9,5	20	30	29	20	137
			52	2,5	7	3,5					
			53	5,5	7	6,5					
			54	5	3	5					

Tablo 68: Partilere Ayrılmış FFS6J30 problemi 2. aşamadaki makine uygunluk kısıtları

İş	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
İşlem göreceği makine	2	2	3	3	3	3	3	3	1	1	3	1	1	3	3
İş	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
İşlem göreceği makine	3	3	3	3	2	1	1	2	1	1	2	-	2	2	-
İş	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
İşlem göreceği makine	2	-	2	3	-	2	3	1	2	2	1	1	2	1	1
İş	46	47	48	49	40	51	52	53	54						
İşlem göreceği makine	2	1	2	3	3	1	2	3	3						

Kesim İşlemi Sonrası Partilere Ayrılmış FFS6J66 problemi

Tablo 69: Partilere Ayrılmış FFS6J66 problemi işlem süreleri ve teslim tarihleri (dk.)

	İş	Kesim Süresi	İş	Bantlama Süresi	CNC Süresi	Delik Süresi	İş	Montaj Süresi	Paket Süresi	İş	Teslim Tarihi
			1	2,5		5,9					
			2	1		4,7					
			3	2		4,5	1	103,6	47,7	1	480
			4	-		-					
			5	4,8		-					
			6	-		-					
	1	15,2	7	2,5		5,9					
	2	9,7	8	1		4,7					
A	3	9,8	9	2		4,5	2	103,6	47,7	2	960
	4	5,8	10	-		-					
	5	15,1	11	4,8		-					
	6	7,3	12	-		-					
			13	10		13,7					
			14	4,2		8,9	3	103,6	47,7	3	1440
			15	7,9		8,1	4	103,6	47,7	4	1440
			16	-		-	5	103,6	47,7	5	1440
			17	19,2		-	6	103,6	47,7	6	1440
			18	-		-					
B	7	12,0	19	4,3		7,7	7	51	58,4	7	1440
	8	10,5	20	14,5		13,7	8	51	58,4	8	1440
C	9	7,7	21	6,1		-	9	37,2	58,6	9	480
	10	9,0	22	11,7		10,5	10	100,6	51,2	10	1440
	11	11,6	23	12,5		21,3	11	100,6	51,2	11	1440
D	12	8,7	24	-		-	12	100,6	51,2	12	1440
	13	9,2	25	27,8		-	13	100,6	51,2	13	1440
	14	7,6	26	-		-	14	100,6	51,2	14	1440
	15	9,5	27	11,7		9,3	15	100,6	51,2	15	1440
	16	20,7	28	21,9		8,1	16	79	71,6	16	1440
E	17	15,2	29	10,7		19,3	17	79	71,6	17	1440
	18	9,5	30	3,0		8,1	18	79	71,6	18	1440
	19	17,3	31	9,0		32,1	19	79	71,6	19	1440
	20	11,9	32	11,0		8,1					
	21	12,5	33	9,6		23,3	20	68,6	44,1	20	1440
F	22	8,0	34	2,7		9,3	21	68,6	44,1	21	1440
	23	8,1	35	9,3		-	22	68,6	44,1	22	1440
	24	8,8	36	8,0		27,3	23	68,6	44,1	23	1440
	25	9,0	37	11,7		10,5	24	110,9	50	24	1440
	26	11,9	38	12,5		21,3	25	110,9	50	25	1440
G	27	8,5	39	-		-	26	110,9	50	26	1440
	28	8,9	40	27,8		-	27	110,9	50	27	1440
	29	7,6	41	-		-	28	110,9	50	28	1440
	30	9,1	42	11,5		9,3	29	110,9	50	29	1440
H	31	8,3	43	9,1		7,3	30	39,2	62,8	30	1440
	32	11,8	44	7,5		12,9	31	39,2	62,8	31	1440
			45	5,6		6,3					
			46	5,4		8,9					
			47	2,7		6,9	32	60,4	98,2	32	480
	33	14,5	48	2,1		6,1					
	34	24,6	49	4		-					
I	35	12,9	50	4,6		8,9					
	36	10,3	51	11,2		9,3					
	37	10,8	52	10,8		14,5					
	38	11,2	53	5,4		10,5	33	60,4	98,2	33	1440
			54	4,3		8,9	34	60,4	98,2	34	1440
			55	8		-					
			56	9,3		14,5					
	39	15,1	57	11,5	-	26,1					
J	40	9,8	58	-	51,2	11,7	35	45,5	65,9	35	960
	41	12,0	59	6,4	-	13,9	36	45,5	65,9	36	960
	42	9,5	60	6,9	-	9,0					
	43	31,9	61	-	122,7	-					
	44	32,2	62	-	150,9	-	37	98,9	61	37	1440
K	45	11,5	63	8,0	-	11,7	38	98,9	61	38	1440
	46	10,2	64	8,9	-	10,5	39	98,9	61	39	1440
	47	17,5	65	16,0	-	-					
	48	6,9	66	-	77,7	-					
L	49	7,1	67	-	93,6	-	40	25	30,2	40	1440
	50	10,4	68	5,6	-	12,3	41	25	30,2	41	1440
	51	7,6	69	3,1	-	8,1	42	25	30,2	42	1440
	52	8,9	70	12,0		-					
	53	9,1	71	12,0		7,5					
M	54	9,5	72	8,0		11,7	43	71,2	39,1	43	1440
	55	8,3	73	4,1		8,1	44	71,2	39,1	44	1440
	56	13,5	74	16,0		-	45	71,2	39,1	45	1440
	57	8,2	75	6,6		-					
			76	5,1		5,3					
	58	22,2	77	2,7		6,5	46	29,6	48,1	46	960
N	59	11,8	78	2,9		5,1					
	60	9,7	79	10,1		7,3					
			80	5,3		9,7	47	29,6	48,1	47	1440
			81	5,7		6,9	48	29,6	48,1	48	1440
			82	2,4		4,9					
			83	2,4		5,1					
			84	5,7		-	49	69,6	74,5	49	960
	61	8,4	85	4,6		5,3					
	62	7,5	86	6,5		8,3					
O	63	11,2	87	-		-					
	64	8,6	88	2,4		4,9					
	65	8,3	89	2,4		5,1					
	66	8,4	90	5,7		-	50	69,6	74,5	50	1440
			91	4,6		5,3					
			92	6,5		8,3					
			93	-		-					

EK-5: TEST PROBLEMLERİNİN MATEMATİKSEL VE SEZGİSEL MODELLERLE ÇÖZÜM SONUÇLARI

FFS3J5 C_{max} amaç fonksiyonu için matematiksel model sonuçları

```
---- VAR C    r aşamasında j
işinin m makinesinden ayrılma
zamanı

1.1.1 :   13.000
1.2.2 :    8.000
1.3.3 :   20.000
1.5.1 :   33.000
2.2.2 :   24.000
2.3.1 :   33.000
2.4.4 :    5.000
2.5.3 :   39.000

---- VAR CA    s aşamasında h
işinin m makinesinden ayrılma
zamanı

1.1.2 :   75.000
1.2.1 :   75.000
1.3.4 :   67.000
1.4.3 :   67.000
```

FFS3J15 C_{max} amaç fonksiyonu için matematiksel model sonuçları

```
---- VAR C    r aşamasında j
işinin m makinesinden ayrılma
zamanı

1.1 .2 :   11.000
1.2 .5 :   48.000
1.3 .2 :   63.000
1.4 .1 :   18.000
1.6 .3 :   27.000
1.7 .3 :    8.000
1.8 .5 :   32.000
1.9 .4 :   13.000
1.10.5 :   14.000
1.13.4 :   49.000
1.14.4 :   36.000
1.15.2 :   37.000
2.1 .2 :   33.000
2.3 .2 :   80.000
2.4 .3 :   45.000
2.5 .4 :   12.000
2.6 .4 :   34.000
2.7 .3 :   35.000
2.10.2 :   22.000
2.11.1 :   19.000
2.12.3 :   17.000
2.13.1 :   59.000
2.14.3 :   62.000

---- VAR CA    s aşamasında h
işinin m makinesinden ayrılma
zamanı

1.1 .5 :  107.000
1.2 .1 :  107.000
1.3 .5 :   86.000
1.4 .1 :   86.000
1.5 .3 :   73.000
1.6 .3 :  111.000
1.7 .1 :   43.000
1.8 .5 :   43.000
1.9 .2 :  102.000
1.10.4 :  102.000
```

FFS3J30 C_{max} amaç fonksiyonu için matematiksel model sonuçları 86400s.

```
---- VAR C    r aşamasında j
işinin m makinesinden ayrılma
zamanı

1.2 .1 :   17.000
1.3 .2 :    5.000
1.4 .1 :  148.000
1.5 .1 :  162.000
1.6 .1 :   84.000
1.8 .2 :  107.000
1.9 .2 :  129.000
1.10.3 :  137.000
1.11.3 :  120.000
1.12.1 :  106.000
1.13.1 :  122.000
1.14.3 :   97.000
1.15.2 :   83.000
1.16.2 :   54.000
1.17.2 :   66.000
1.18.3 :   75.000
1.19.3 :   44.000
1.20.3 :   65.000
1.22.2 :   28.000
1.23.3 :   10.000
1.26.1 :   58.000
1.27.1 :   33.000
1.29.3 :   27.000
2.1 .3 :   37.000
2.2 .2 :   25.000
2.3 .3 :   18.000
2.4 .2 :  165.000
2.6 .2 :   95.000
2.7 .3 :   84.000
2.8 .1 :  129.000
2.9 .1 :  147.000
2.11.2 :  138.000
2.12.4 :  120.000
2.13.3 :  129.000
2.14.2 :  109.000
2.15.3 :  122.000
2.16.3 :   66.000
2.17.1 :  122.000
2.20.1 :   84.000
2.21.2 :   34.000
2.22.4 :   57.000
2.24.4 :    7.000
2.25.2 :   12.000
2.27.4 :   38.000
2.28.1 :    6.000
2.29.4 :   68.000
2.30.2 :    6.000

---- VAR CA    s aşamasında h
işinin m makinesinden ayrılma
zamanı

1.1 .2 :  128.000
1.2 .5 :  187.000
1.3 .3 :  200.000
1.4 .4 :  200.000
1.5 .1 :  157.000
1.6 .5 :  159.000
1.7 .6 :  194.000
1.8 .2 :  196.000
1.9 .1 :  178.000
1.10.1 :  199.000
1.11.2 :  149.000
1.12.4 :  165.000
1.13.3 :  153.000
1.14.5 :  131.000
1.15.6 :  147.000
1.16.3 :  106.000
1.17.2 :  100.000
1.18.4 :  100.000
1.19.4 :  135.000
1.20.1 :  129.000
```

FFS6J5 C_{max} amaç fonksiyonu için matematiksel model sonuçları

```
---- VAR C    r aşamasında j
işinin m makinesinden ayrılma
zamanı

1.1.2 :    9.000
1.2.1 :   11.000
1.3.2 :   19.000
1.4.3 :   16.000
1.5.1 :   27.000
2.1.3 :   28.000
2.3.3 :   44.000
2.4.1 :   36.000
2.5.2 :   43.000
3.1.1 :   38.000
3.2.2 :   26.000
3.4.2 :   51.000
3.5.2 :   61.000
4.1.1 :   50.000
4.2.2 :   44.000
4.3.3 :   56.000
4.4.1 :   63.000
4.5.2 :   74.000

---- VAR CA    s aşamasında h
işinin m makinesinden ayrılma
zamanı

1.1.2 :  107.000
1.2.3 :  105.000
1.3.4 :  118.000
1.4.1 :  118.000
2.1.2 :  147.000
2.2.3 :  145.000
2.3.4 :  147.000
2.4.1 :  147.000
```

FFS6J15 C_{max} amaç fonksiyonu için matematiksel model sonuçları 86400s.

```
---- VAR C    r aşamasında j
işinin m makinesinden ayrılma
zamanı

1.1 .3 :   39.000
1.2 .1 :   48.000
1.3 .1 :   35.000
1.4 .4 :   16.000
1.6 .4 :   45.000
1.7 .1 :   76.000
1.8 .4 :   35.000
1.9 .1 :   18.000
1.10.3 :   55.000
1.13.2 :   31.000
1.14.2 :   13.000
1.15.3 :   19.000
2.1 .5 :   54.000
2.2 .4 :   55.000
2.3 .6 :   48.000
2.4 .5 :   35.000
2.6 .1 :   64.000
2.7 .5 :   85.000
2.9 .3 :   24.000
2.10.4 :   71.000
2.11.2 :   16.000
2.12.3 :   14.000
2.13.5 :   45.000
```

```
2.14.4 :   22.000
2.15.6 :   37.000
3.3 .4 :   57.000
3.5 .5 :   10.000
3.6 .6 :   73.000
3.7 .4 :   91.000
3.8 .2 :   94.000
3.9 .4 :   42.000
3.10.2 :   78.000
3.11.5 :   32.000
3.12.3 :   25.000
3.13.5 :   52.000
3.14.1 :   39.000
4.1 .3 :   93.000
4.2 .2 :   81.000
4.3 .2 :   95.000
4.4 .3 :   51.000
4.5 .1 :   18.000
4.6 .3 :  116.000
4.7 .3 :  105.000
4.8 .2 :  105.000
4.9 .3 :   80.000
4.10.1 :   83.000
4.11.3 :   71.000
4.12.2 :   43.000
4.13.2 :   62.000
4.14.1 :   62.000
4.15.1 :   48.000

---- VAR CA    s aşamasında h
işinin m makinesinden ayrılma
zamanı

1.1 .6 :  131.000
1.2 .4 :  131.000
1.3 .3 :  101.000
1.4 .1 :  101.000
1.5 .5 :  140.000
1.6 .1 :  140.000
1.7 .2 :  104.000
1.8 .3 :  122.000
1.9 .4 :   98.000
1.10.6 :   98.000
2.1 .1 :  180.000
2.2 .2 :  180.000
2.3 .6 :  169.000
2.4 .5 :  130.000
2.5 .5 :  180.000
2.6 .4 :  180.000
2.7 .3 :  138.000
2.8 .3 :  172.000
2.9 .6 :  140.000
2.10.4 :  140.000
```

FFS6J30 C_{max} amaç fonksiyonu için matematiksel model sonuçları 86400s.

```
---- VAR C    r aşamasında j
işinin m makinesinden ayrılma
zamanı

1.2 .3 :    9.000
1.3 .1 :  126.000
1.4 .2 :   12.000
1.5 .1 :   24.000
1.6 .3 :  165.000
1.8 .2 :  153.000
1.9 .1 :  110.000
1.10.3 :   63.000
1.11.2 :   52.000
1.12.1 :   83.000
1.13.2 :  101.000
1.14.3 :   35.000
1.15.2 :   67.000
1.16.3 :   93.000
1.17.1 :    8.000
1.18.3 :  140.000
1.19.1 :   69.000
1.20.1 :   48.000
1.22.2 :   32.000
1.23.2 :  128.000
1.26.2 :   84.000
1.27.3 :  117.000
1.29.3 :   78.000
2.1 .2 :   56.000
2.2 .2 :   24.000
2.3 .3 :  194.000
2.4 .3 :   67.000
2.5 .3 :   42.000
2.6 .3 :  180.000
2.7 .1 :   20.000
2.8 .1 :  164.000
2.9 .3 :  143.000
2.10.3 :  123.000
2.11.3 :  108.000
2.12.2 :  101.000
2.13.1 :  114.000
2.14.1 :   66.000
2.15.2 :  161.000
2.17.2 :   36.000
2.19.2 :   87.000
2.20.3 :   60.000
2.21.1 :   11.000
2.22.2 :  119.000
2.23.2 :  145.000
2.24.1 :   39.000
2.25.1 :   52.000
2.26.2 :  171.000
2.27.1 :  126.000
2.28.2 :   61.000
2.29.3 :   89.000
2.30.3 :   77.000
3.2 .2 :  110.000
3.3 .4 :  207.000
3.4 .3 :   99.000
3.5 .4 :  124.000
3.7 .2 :   32.000
3.8 .3 :  170.000
3.10.1 :  133.000
3.11.1 :  140.000
3.12.1 :  109.000
3.14.2 :  134.000
3.15.4 :  168.000
3.16.1 :  117.000
3.17.4 :  119.000
3.18.2 :  147.000
3.19.4 :  107.000
3.20.2 :   65.000
3.21.2 :  157.000
3.22.1 :  173.000
3.23.1 :  159.000
3.24.2 :   74.000
3.25.2 :   92.000
3.28.4 :   75.000
3.29.2 :  124.000
3.30.3 :   83.000
4.1 .4 :  217.000
4.2 .4 :  188.000
4.3 .3 :  217.000
4.4 .4 :  143.000
4.5 .4 :  138.000
4.6 .3 :  224.000
4.7 .4 :  149.000
4.8 .3 :  184.000
4.9 .3 :  166.000
4.10.3 :  147.000
4.11.2 :  171.000
4.12.1 :  114.000
4.13.2 :  151.000
4.15.1 :  179.000
4.16.1 :  123.000
4.17.4 :  174.000
4.18.1 :  168.000
4.19.2 :  119.000
4.20.1 :  200.000
4.21.4 :  196.000
4.22.4 :  203.000
4.23.2 :  184.000
4.24.4 :  161.000
4.25.2 :  138.000
4.26.1 :  193.000
4.27.1 :  145.000
4.28.1 :  152.000
4.29.3 :  197.000
4.30.3 :  176.000

---- VAR CA    s aşamasında h
işinin m makinesinden ayrılma
zamanı

1.1 .1 :  262.000
1.2 .6 :  259.000
1.3 .3 :  290.000
1.4 .5 :  165.000
1.5 .4 :  265.000
1.6 .1 :  303.000
1.7 .3 :  204.000
1.8 .4 :  204.000
1.9 .6 :  174.000
1.10.1 :  174.000
1.11.2 :  228.000
1.12.5 :  228.000
1.13.6 :  284.000
1.14.5 :  253.000
1.15.2 :  262.000
1.16.3 :  238.000
1.17.6 :  224.000
1.18.6 :  315.000
1.19.3 :  268.000
1.20.1 :  227.000
2.1 .2 :  327.000
2.2 .5 :  289.000
2.3 .1 :  329.000
2.4 .5 :  198.000
2.5 .1 :  296.000
2.6 .3 :  336.000
2.7 .6 :  242.000
2.8 .3 :  242.000
2.9 .5 :  246.000
2.10.4 :  222.000
2.11.2 :  265.000
2.12.3 :  279.000
2.13.3 :  305.000
2.14.6 :  332.000
2.15.5 :  337.000
2.16.6 :  311.000
2.17.6 :  263.000
2.18.4 :  336.000
2.19.2 :  297.000
2.20.4 :  256.000
```

FFS6J66 C_{max} amaç fonksiyonu için matematiksel model sonuçları 7200s.

```
---- VAR C    r aşamasında j
işinin m makinesinden ayrılma
zamanı

1.1 .2 :  249.200
1.2 .2 :  279.100
1.3 .2 :  225.100
1.4 .2 :  139.600
1.5 .2 :  176.900
1.6 .1 :   22.500
1.7 .1 :  233.300
1.8 .1 :  273.600
1.9 .2 :  302.000
1.10.1 :  191.500
1.11.1 :  365.900
1.12.2 :  257.900
1.13.1 :   94.900
1.14.1 :   30.100
1.15.1 :  375.400
1.16.1 :  254.000
1.17.1 :   15.200
1.18.2 :   91.200
1.19.1 :  343.500
1.20.2 :  215.300
1.21.1 :  213.000
1.22.1 :  317.100
1.23.1 :  153.300
1.24.2 :   71.500
1.25.1 :  200.500
1.26.1 :   85.700
1.27.1 :  309.100
1.28.1 :  136.800
1.29.2 :   62.700
1.30.1 :  326.200
1.31.2 :  294.300
1.32.2 :   18.900
1.33.2 :  133.800
1.34.1 :  119.500
1.35.2 :   55.100
1.36.2 :  119.300
1.37.1 :  354.300
1.38.1 :  459.300
1.39.1 :  448.100
1.40.2 :   28.700
1.41.2 :  314.000
1.42.2 :  109.000
1.43.1 :   73.800
1.44.1 :  433.000
1.45.2 :  269.400
1.46.2 :   81.700
1.47.1 :  300.600
1.48.2 :  286.000
1.49.2 :    7.100
1.50.1 :  163.700
1.51.1 :  171.300
1.52.2 :  234.000
1.53.1 :  263.100
1.54.1 :  283.100
1.55.2 :   99.500
1.56.2 :   42.200
1.57.2 :  193.700
1.58.2 :  161.800
1.59.1 :   41.900
1.60.2 :  203.400
1.61.1 :  145.200
1.62.1 :  382.900
```

```
1.63.1 :  182.500
1.64.2 :  185.500
1.65.1 :  221.300
1.66.1 :  127.900
2.1 .2 :  320.600
2.2 .2 :  372.900
2.3 .2 :  335.100
2.5 .1 :  205.700
2.7 .1 :  507.200
2.8 .1 :  474.100
2.9 .1 :  513.300
2.10.1 :  535.200
2.11.2 :  517.900
2.13.1 :  502.400
2.15.2 :  441.600
2.16.1 :  342.300
2.17.1 :  416.500
2.18.2 :  409.400
2.19.2 :  381.900
2.20.1 :  360.800
2.21.1 :  426.100
2.22.2 :  323.300
2.23.2 :  391.200
2.24.1 :  214.200
2.25.1 :  584.800
2.26.2 :  429.900
2.28.1 :  563.500
2.30.2 :  358.600
2.31.1 :  573.100
2.32.1 :  349.800
2.33.1 :  443.400
2.34.2 :  505.400
2.35.2 :  366.700
2.36.2 :  341.500
2.37.2 :  529.900
2.38.2 :  473.200
2.39.1 :  459.600
2.41.1 :  320.400
2.42.2 :  115.900
2.45.2 :  417.400
2.46.2 :  459.300
2.47.2 :  489.200
2.50.2 :  347.100
2.51.2 :  294.200
2.52.1 :  313.500
2.53.1 :  301.500
2.54.2 :  291.100
2.55.2 :  445.700
2.56.1 :  397.300
2.57.2 :  406.400
2.58.1 :  380.800
2.59.1 :  405.800
2.60.2 :  399.800
2.61.1 :  365.600
2.62.2 :  450.400
2.63.2 :  305.600
2.64.1 :  523.000
2.65.2 :  234.400
3.40.1 :  247.700
3.43.1 :  196.500
3.44.1 :  613.700
3.48.1 :  369.200
3.49.1 :  462.800
4.1 .2 :  641.600
4.2 .1 :  425.700
4.3 .2 :  557.400
4.7 .2 :  546.900
4.8 .2 :  607.300
4.10.1 :  754.800
4.11.2 :  539.200
4.15.1 :  552.800
4.16.2 :  615.400
4.17.1 :  656.000
4.18.1 :  744.300
4.19.1 :  414.000
4.20.1 :  433.800
4.21.2 :  593.600
4.22.1 :  509.600
4.24.1 :  736.200
4.25.1 :  697.200
4.26.2 :  662.900
4.30.1 :  467.700
4.31.1 :  675.000
4.32.1 :  446.700
4.33.1 :  521.900
4.34.2 :  689.900
4.35.1 :  543.500
4.36.1 :  667.700
4.38.1 :  614.700
4.39.1 :  493.800
4.40.1 :  686.700
4.41.1 :  636.700
4.42.2 :  368.400
4.45.1 :  458.400
4.46.1 :  594.600
4.50.2 :  359.400
4.51.2 :  302.300
4.53.1 :  529.400
4.54.1 :  708.900
4.55.1 :  622.800
4.58.1 :  575.400
4.59.2 :  570.300
4.60.1 :  584.100
4.61.1 :  500.300
4.62.2 :  669.800
4.64.2 :  622.700
4.65.1 :  566.100

---- VAR CA   s aşamasında h
işinin m makinesinden ayrılma
zamanı

1.1 .2 : 1106.900
1.2 .3 :  851.500
1.3 .1 : 1027.500
1.4 .2 : 1568.700
1.5 .2 : 1465.100
1.6 .5 : 1126.100
1.7 .1 : 1078.500
1.8 .3 :  658.300
1.9 .4 :  817.300
1.10.1 :  923.900
1.11.6 : 1001.500
1.12.2 : 1207.500
1.13.3 :  952.100
1.14.3 : 1052.700
1.15.6 :  855.400
1.16.1 : 1268.500
1.17.3 : 1248.800
1.18.1 :  823.300
1.19.6 : 1181.300
1.20.5 : 1354.100
1.21.3 : 1317.400
1.22.2 :  892.300
1.23.2 : 1276.100
1.24.2 : 1003.300
1.25.2 :  823.700
1.26.5 :  992.900
1.27.4 :  928.300
1.28.1 : 1189.500
1.29.3 : 1428.400
1.30.5 :  881.900
1.31.2 : 1677.500
1.32.2 : 1336.500
1.33.5 : 1186.500
1.34.5 :  842.700
1.35.3 : 1498.900
1.36.6 :  900.900
1.37.2 :  712.700
1.38.5 :  712.700
1.39.5 : 1285.500
1.40.2 :  487.800
1.41.2 : 1361.500
1.42.3 : 1453.400
1.43.5 : 1425.300
1.44.6 : 1072.700
1.45.4 :  780.100
1.46.5 : 1022.500
1.47.6 : 1102.300
1.48.2 :  613.700
1.49.5 :  782.300
1.50.2 : 1638.300
2.1 .3 : 1241.900
2.2 .4 : 1354.500
2.3 .5 : 1075.200
2.4 .1 : 2158.400
2.5 .1 : 1888.800
2.6 .5 : 1663.900
2.7 .2 : 1136.900
2.8 .2 :  716.700
2.9 .1 : 2217.000
2.10.4 : 1207.500
2.11.3 : 1052.700
2.12.4 : 1258.700
2.13.1 : 1553.900
2.14.3 : 1103.900
2.15.3 : 1194.200
2.16.5 : 1616.200
2.17.4 : 1497.700
2.18.1 : 1810.900
2.19.4 : 1426.100
2.20.5 : 1398.200
2.21.1 : 1361.500
2.22.3 : 1286.000
2.23.3 : 1330.100
2.24.1 : 2049.700
2.25.1 : 1502.700
2.26.2 : 1186.900
2.27.1 :  978.300
2.28.4 : 1685.000
2.29.2 : 1478.400
2.30.1 : 1616.700
2.31.1 : 1999.700
2.32.5 : 1762.100
2.33.1 : 2315.200
2.34.4 : 1595.900
2.35.1 : 2381.100
2.36.4 : 1750.900
2.37.1 : 2110.700
2.38.1 : 1452.700
2.39.5 : 1544.600
2.40.1 : 1841.100
2.41.1 : 1391.700
2.42.5 : 1483.600
2.43.4 : 1635.000
2.44.3 : 1143.000
2.45.5 : 1437.300
2.46.1 : 1936.900
2.47.4 : 1306.800
2.48.1 : 1664.800
2.49.2 : 1428.400
2.50.1 : 1739.300
```

FFS6J66 C_{max} amaç fonksiyonu için matematiksel model sonuçları 86400s.

```
---- VAR C   r aşamasında j
işinin m makinesinden ayrılma
zamanı

1.1 .2 :  222.300
1.2 .2 :  207.100
1.3 .2 :   68.300
1.4 .2 :   58.500
1.5 .2 :  246.000
1.6 .1 :  210.400
1.7 .1 :   51.500
1.8 .1 :  135.400
1.9 .2 :  110.500
1.10.1 :  304.800
1.11.1 :  284.000
1.12.1 :  243.700
1.13.1 :  260.500
1.14.1 :  440.900
1.15.2 :   41.200
1.16.1 :  156.100
1.17.1 :   31.000
1.18.2 :    9.500
1.19.1 :  351.200
1.20.2 :   80.200
1.21.1 :  185.000
1.22.1 :  172.500
1.23.1 :   77.000
1.24.2 :  177.300
1.25.1 :  203.100
1.26.1 :  272.400
1.27.1 :   39.500
1.28.1 :   15.800
1.29.2 :  253.600
1.30.1 :   68.900
1.31.2 :  128.300
1.32.2 :  197.400
1.33.2 :  268.100
1.34.1 :  235.000
1.35.2 :   93.100
1.36.1 :  473.800
1.37.1 :  362.000
1.38.1 :  316.000
1.39.1 :  463.500
1.40.2 :  138.100
1.41.2 :   21.500
1.42.2 :  120.000
1.43.1 :  108.900
1.44.1 :  404.600
1.45.2 :   52.700
1.46.2 :   31.700
1.47.1 :  433.300
1.48.1 :    6.900
1.49.1 :  124.900
1.50.1 :  372.400
1.51.1 :  251.300
1.52.1 :  117.800
1.53.1 :  194.100
1.54.1 :  333.900
1.55.2 :  185.600
1.56.2 :  281.600
1.57.2 :  146.300
1.58.2 :  168.500
1.59.1 :  295.800
1.60.2 :  102.800
1.61.1 :  324.400
1.62.1 :  448.400
1.63.1 :  415.800
1.64.2 :  230.900
1.65.1 :   59.800
1.66.1 :  164.500
2.1 .2 :  346.100
2.2 .2 :  411.900
```

```
2.3 .2 :  179.200
2.5 .1 :  329.800
2.7 .1 :  279.800
2.8 .1 :  389.300
2.9 .1 :  230.400
2.10.1 :  342.000
2.11.2 :  308.300
2.13.1 :  446.200
2.15.2 :  405.700
2.16.1 :  178.000
2.17.1 :  514.500
2.18.2 :  167.400
2.19.2 :  360.200
2.20.1 :  374.800
2.21.1 :  298.500
2.22.2 :  274.100
2.23.2 :  369.500
2.24.1 :  250.100
2.25.1 :  242.100
2.26.2 :  394.000
2.28.1 :  503.300
2.30.2 :  164.400
2.31.1 :  288.900
2.32.1 :  204.900
2.33.1 :  463.000
2.34.2 :  251.200
2.35.2 :  263.400
2.36.2 :  480.200
2.37.2 :  381.500
2.38.2 :  331.100
2.39.1 :  475.000
2.41.1 :  211.300
2.42.2 :  426.800
2.45.2 :  271.400
2.46.2 :  317.200
2.47.2 :  469.100
2.50.2 :  485.800
2.51.2 :  429.900
2.52.1 :  129.800
2.53.1 :  223.800
2.54.2 :  419.900
2.55.2 :  255.300
2.56.1 :  363.300
2.57.2 :  152.900
2.58.1 :  275.500
2.59.1 :  417.900
2.60.2 :  295.800
2.61.1 :  346.800
2.62.2 :  453.100
2.63.2 :  441.300
2.64.1 :  259.800
2.65.2 :  287.200
3.40.1 :  189.300
3.43.1 :  312.000
3.44.1 :  556.500
3.48.1 :   84.600
3.49.1 :  405.600
4.1 .2 :  498.400
4.2 .1 :  514.500
4.3 .2 :  624.500
4.7 .2 :  584.600
4.8 .2 :  553.500
4.10.1 :  699.200
4.11.2 :  605.900
4.15.1 :  596.900
4.16.2 :  614.000
4.17.1 :  533.800
4.18.1 :  675.800
4.19.1 :  428.400
4.20.1 :  631.100
4.21.1 :  667.700
4.22.1 :  437.700
4.24.1 :  580.700
4.25.1 :  491.100
4.26.2 :  472.200
4.30.1 :  362.600
4.31.1 :  553.400
4.32.1 :  688.700
4.33.1 :  546.100
4.34.1 :  480.600
4.35.1 :  460.500
4.36.1 :  502.800
4.38.2 :  530.800
4.39.1 :  623.000
4.40.2 :  565.200
4.41.1 :  376.500
4.42.2 :  539.800
4.45.1 :  283.100
4.46.1 :  396.300
4.50.2 :  510.700
4.51.2 :  438.000
4.53.1 :  298.700
4.54.2 :  576.900
4.55.1 :  291.200
4.58.1 :  385.800
4.59.2 :  450.900
4.60.1 :  446.400
4.61.1 :  353.300
4.62.1 :  587.600
4.64.2 :  479.500
4.65.1 :  644.400

---- VAR CA    s aşamasında h
işinin m makinesinden ayrılma
zamanı

1.1 .3 :  883.100
1.2 .1 :  728.100
1.3 .1 : 1114.500
1.4 .2 :  949.900
1.5 .4 :  728.100
1.6 .6 :  759.100
1.7 .3 : 1034.700
1.8 .2 : 1000.900
1.9 .3 :  267.600
1.10.1 :  828.700
1.11.4 :  939.700
1.12.6 :  968.300
1.13.2 :  846.300
1.14.3 :  983.700
1.15.2 : 1238.700
1.16.1 : 1193.500
1.17.3 : 1113.700
1.18.5 :  840.300
1.19.6 :  867.700
1.20.5 :  736.300
1.21.3 : 1182.300
1.22.2 : 1138.100
1.23.2 : 1069.500
1.24.5 : 1401.700
1.25.2 :  646.700
1.26.5 :  951.300
1.27.4 :  839.100
1.28.1 : 1010.900
1.29.3 :  779.500
1.30.5 :  990.500
1.31.2 : 1394.600
1.32.5 :  606.500
1.33.5 : 1120.500
1.34.5 :  666.900
1.35.3 :  668.500
1.36.2 : 1355.400
1.37.2 :  745.700
1.38.6 :  655.500
1.39.5 : 1290.700
1.40.2 :  535.700
1.41.5 :  761.300
1.42.3 :  535.700
1.43.5 : 1191.700
1.44.2 : 1309.900
1.45.1 :  899.900
1.46.5 :  480.500
1.47.6 :  997.900
1.48.6 :  788.700
1.49.5 : 1060.100
1.50.3 : 1251.900
2.1 .4 : 1196.700
2.2 .2 :  775.800
2.3 .5 : 1162.200
2.4 .3 : 1232.900
2.5 .5 : 1373.300
2.6 .5 : 1254.000
2.7 .2 : 1294.800
2.8 .2 : 1526.000
2.9 .3 :  326.200
2.10.4 : 1149.000
2.11.3 :  990.900
2.12.4 : 1019.500
2.13.1 :  953.700
2.14.3 : 1430.400
2.15.3 : 1520.700
2.16.5 : 1325.600
2.17.1 : 1185.300
2.18.2 : 1206.200
2.19.3 : 1185.200
2.20.5 : 1206.300
2.21.1 : 1525.900
2.22.2 : 1467.600
2.23.3 : 1113.600
2.24.4 : 1451.700
2.25.1 :  841.500
2.26.2 : 1001.300
2.27.3 : 1570.700
2.28.2 : 1576.000
2.29.4 :  829.500
2.30.2 : 1357.600
2.31.1 : 1481.800
2.32.5 : 1532.500
2.33.3 : 1331.100
2.34.1 : 1283.500
2.35.4 : 1517.600
2.36.2 : 1423.500
2.37.1 : 1344.500
2.38.1 :  902.500
2.39.5 : 1434.300
2.40.4 : 1097.800
2.41.1 :  791.500
2.42.2 : 1236.400
2.43.4 : 1235.800
2.44.3 : 1469.500
2.45.3 : 1069.500
2.46.3 : 1379.200
2.47.4 : 1067.600
2.48.4 : 1283.900
2.49.2 : 1134.600
2.50.1 : 1419.000
```

FFS6J66 C_{max} amaç fonksiyonu için matematiksel model sonuçları 259200s.

```
---- VAR C     r aşamasında j
işinin m makinesinden ayrılma
zamanı

1.1 .2 :   148.800
1.2 .1 :   245.400
1.3 .2 :   133.600
1.4 .1 :   416.000
1.5 .1 :   285.100
1.6 .2 :   241.200
1.7 .1 :    12.000
1.8 .1 :   102.400
1.9 .2 :   318.400
1.10.1 :    80.300
1.11.1 :    91.900
1.12.1 :   226.200
1.13.2 :   123.800
1.14.2 :   278.100
1.15.1 :   235.700
1.16.2 :   222.100
1.17.1 :   125.700
1.18.1 :   367.300
1.19.2 :   193.200
1.20.1 :   379.200
1.21.2 :   270.500
1.22.2 :     8.000
1.23.1 :   110.500
1.24.1 :   154.100
1.25.1 :   172.200
1.26.2 :   114.600
1.27.2 :   295.500
1.28.1 :   388.100
1.29.1 :   395.700
1.30.2 :    46.400
1.31.1 :    28.700
1.32.2 :   233.900
1.33.1 :   410.200
1.34.1 :   270.000
1.35.1 :   217.500
1.36.1 :   312.400
1.37.2 :    94.100
1.38.1 :   136.900
1.39.2 :   175.900
1.40.2 :   328.200
1.41.2 :   160.800
1.42.2 :    83.300
1.43.1 :   357.800
1.44.1 :    71.300
1.45.2 :    73.800
1.46.1 :   204.600
1.47.2 :    37.300
1.48.2 :   302.400
1.49.2 :   258.000
1.50.1 :    39.100
1.51.2 :    62.300
1.52.2 :   287.000
1.53.1 :   163.200
1.54.1 :   294.600
1.55.2 :    54.700
1.56.1 :   325.900
1.57.2 :   201.400
1.58.1 :   194.400
1.59.2 :    19.800
1.60.2 :   250.900
1.61.1 :    20.400
1.62.1 :   302.100
1.63.2 :   339.400
1.64.2 :   102.700
1.65.2 :   310.700
1.66.1 :   145.300
2.1 .2 :   163.800
2.2 .2 :   301.300
2.3 .2 :   234.800
2.5 .1 :   317.300
2.7 .1 :   169.200
2.8 .1 :   226.600
2.9 .1 :   464.800
2.10.1 :   278.900
```

```
2.11.2 :   118.600
2.13.1 :   151.600
2.15.2 :   247.400
2.16.1 :   440.900
2.17.1 :   328.500
2.18.2 :   376.400
2.19.2 :   202.200
2.20.1 :   390.200
2.21.1 :   491.400
2.22.2 :   134.100
2.23.2 :   393.800
2.24.1 :   190.200
2.25.1 :   212.100
2.26.2 :   146.600
2.28.1 :   418.500
2.30.2 :    57.900
2.31.1 :   288.000
2.32.1 :   267.200
2.33.1 :   458.200
2.34.2 :   286.200
2.35.2 :   384.500
2.36.2 :   334.900
2.37.2 :   106.100
2.38.2 :   223.000
2.39.1 :   244.500
2.41.1 :   233.000
2.42.2 :   209.100
2.45.2 :   373.400
2.46.2 :   295.100
2.47.2 :   179.800
2.50.2 :   124.200
2.51.2 :   131.400
2.52.1 :   341.000
2.53.1 :   181.700
2.54.2 :   358.800
2.55.2 :   128.300
2.56.1 :   481.300
2.57.2 :   365.400
2.58.1 :   259.700
2.59.1 :   160.100
2.60.2 :   315.400
2.61.1 :   164.900
2.62.2 :   306.800
2.63.2 :   350.800
2.64.1 :   199.900
2.65.2 :   328.500
3.40.1 :   402.800
3.43.1 :   525.500
3.44.1 :   222.200
3.48.1 :   603.200
3.49.1 :   351.600
4.1 .1 :   241.500
4.2 .2 :   313.000
4.3 .2 :   245.300
4.7 .1 :   249.200
4.8 .1 :   399.200
4.10.1 :   289.400
4.11.1 :   139.900
4.15.2 :   301.300
4.16.1 :   449.000
4.17.2 :   545.700
4.18.1 :   483.000
4.19.2 :   345.100
4.20.1 :   526.500
4.21.2 :   514.700
4.22.1 :   278.600
4.24.1 :   518.400
4.25.1 :   222.600
4.26.2 :   567.000
4.30.2 :    67.200
4.31.1 :   463.200
4.32.1 :   338.000
4.33.2 :   470.500
4.34.2 :   365.200
4.35.2 :   398.600
4.36.1 :   549.900
4.38.1 :   269.300
4.39.1 :   377.400
4.40.1 :   538.200
4.41.1 :   563.800
4.42.2 :   407.600
4.45.1 :   474.900
4.46.1 :   325.100
4.50.1 :   314.600
4.51.1 :   491.100
4.53.1 :   189.200
4.54.2 :   526.400
4.55.1 :   385.500
4.58.2 :   269.000
4.59.1 :   302.300
4.60.1 :   407.900
4.61.2 :   371.700
4.62.1 :   455.900
4.64.1 :   207.200
4.65.1 :   351.300

---- VAR CA   s aşamasındaki h
işinin m makinesinden ayrılma
zamanı

1.1 .1 :   558.500
1.2 .1 :   713.900
1.3 .5 :   519.600
1.4 .6 :   761.100
1.5 .2 :   917.800
1.6 .2 :   519.600
1.7 .6 :   813.100
1.8 .1 :   453.900
1.9 .4 :   502.000
1.10.1 :  1130.700
1.11.5 :   964.800
1.12.6 :   401.900
1.13.1 :   401.900
1.14.2 :   621.200
1.15.3 :  1001.300
1.16.1 :   956.900
1.17.2 :   813.200
1.18.5 :   624.700
1.19.5 :   816.700
1.20.3 :  1070.900
1.21.6 :   595.100
1.22.2 :  1059.600
1.23.6 :   882.700
1.24.2 :   733.200
1.25.3 :  1182.900
1.26.3 :   899.700
1.27.5 :   736.700
1.28.6 :   994.700
1.29.4 :   976.100
1.30.4 :   542.200
1.31.1 :   754.100
1.32.1 :   815.500
1.33.6 :   656.500
1.34.1 :   876.900
1.35.5 :   863.200
1.36.1 :   609.300
1.37.3 :   761.700
1.38.4 :   807.500
1.39.4 :   707.500
1.40.3 :   631.100
1.41.4 :   864.100
1.42.3 :   787.700
1.43.3 :   597.700
1.44.1 :  1029.100
1.45.2 :   990.000
1.46.3 :   661.700
1.47.4 :   572.800
1.48.4 :   838.100
1.49.2 :  1130.200
1.50.3 :   525.500
2.1 .1 :   687.900
2.2 .4 :   831.900
2.3 .1 :   736.600
2.4 .2 :  1182.400
2.5 .1 :  1230.800
2.6 .4 :   567.300
2.7 .1 :   871.500
2.8 .5 :  1031.100
2.9 .2 :   560.600
2.10.5 :  1181.900
2.11.5 :  1083.300
2.12.1 :  1182.100
2.13.3 :  1115.100
2.14.5 :   920.700
2.15.3 :  1167.300
2.16.3 :  1062.900
2.17.1 :  1098.700
2.18.3 :   990.300
2.19.2 :   999.400
2.20.2 :  1133.700
2.21.1 :   639.200
2.22.5 :  1278.000
2.23.2 :   926.800
2.24.4 :   783.200
2.25.5 :  1232.900
2.26.5 :   971.700
2.27.2 :   786.700
2.28.3 :  1218.300
2.29.1 :  1026.100
2.30.3 :   886.500
2.31.4 :   944.800
2.32.4 :  1226.400
2.33.3 :   754.700
2.34.1 :   975.100
2.35.4 :  1096.000
2.36.1 :   803.500
2.37.3 :   822.700
2.38.5 :   868.500
2.39.5 :   768.500
2.40.1 :  1129.900
2.41.3 :   917.700
2.42.4 :  1127.200
2.43.2 :  1039.500
2.44.5 :  1129.700
2.45.4 :  1029.100
2.46.2 :   881.700
2.47.4 :   881.000
2.48.2 :  1088.600
2.49.2 :  1257.900
2.50.4 :   642.800
```

FFS3J5 $\sum T_j$ amaç fonksiyonu için matematiksel model sonuçları

```
---- VAR C   r aşamasında j
işinin m makinesinden ayrılma
zamanı

1.1.1 :   33.000
1.2.2 :    8.000
1.3.3 :   20.000
1.5.1 :   12.000
2.2.2 :   24.000
2.3.1 :   33.000
2.4.4 :    5.000
2.5.3 :   18.000

---- VAR CA   s aşamasında h
işinin m makinesinden ayrılma
zamanı

1.1.1 :   75.000
1.2.4 :   75.000
1.3.2 :   46.000
1.4.3 :   46.000
```

FFS3J15 $\sum T_j$ amaç fonksiyonu için matematiksel model sonuçları 86400s.

```
---- VAR C   r aşamasında j
işinin m makinesinden ayrılma
zamanı

1.1 .1 :   11.000
1.2 .5 :    8.000
1.3 .5 :   28.000
1.4 .4 :   66.000
1.6 .4 :   17.000
1.7 .2 :    8.000
1.8 .2 :   20.000
1.9 .3 :   29.000
1.10.3 :   14.000
1.13.1 :   30.000
1.14.4 :   40.000
1.15.2 :   46.000
2.1 .2 :   22.000
2.3 .2 :   47.000
2.4 .3 :   76.000
2.5 .4 :   12.000
2.6 .4 :   24.000
2.7 .3 :   26.000
2.10.2 :   30.000
2.11.1 :   30.000
2.12.3 :   43.000
2.13.1 :   40.000
2.14.3 :   60.000

---- VAR CA   s aşamasında h
işinin m makinesinden ayrılma
zamanı

1.1 .2 :   68.000
1.2 .5 :   68.000
1.3 .3 :  117.000
1.4 .3 :  158.000
1.5 .4 :   64.000
1.6 .2 :  167.000
1.7 .3 :   51.000
1.8 .2 :  129.000
1.9 .1 :  100.000
1.10.2 :  108.000
```

FFS3J30 $\sum T_j$ amaç fonksiyonu için matematiksel model sonuçları 86400s.

```
---- VAR C    r aşamasında j
işinin m makinesinden ayrılma
zamanı

1.2 .3 :  108.000
1.3 .1 :   31.000
1.4 .1 :   18.000
1.5 .2 :   39.000
1.6 .3 :  156.000
1.8 .2 :  121.000
1.9 .3 :   66.000
1.10.3 :   83.000
1.11.1 :   81.000
1.12.1 :  119.000
1.13.3 :  130.000
1.14.1 :  103.000
1.15.2 :   62.000
1.16.1 :  145.000
1.17.2 :  139.000
1.18.1 :   47.000
1.19.1 :   58.000
1.20.3 :   44.000
1.22.2 :   85.000
1.23.2 :   97.000
1.26.3 :   17.000
1.27.2 :    8.000
1.29.2 :   19.000
2.1 .3 :   19.000
2.2 .2 :  116.000
2.3 .3 :   44.000
2.4 .2 :   38.000
2.6 .2 :  167.000
2.7 .3 :   62.000
2.8 .1 :  126.000
2.9 .1 :   84.000
2.11.2 :   99.000
2.12.4 :  133.000
2.13.3 :  137.000
2.14.2 :  128.000
2.15.3 :   82.000
2.16.3 :  157.000
2.17.1 :  158.000
2.20.1 :   63.000
2.21.2 :   21.000
2.22.4 :  104.000
2.24.4 :    7.000
2.25.2 :   12.000
2.27.4 :   13.000
2.28.1 :    6.000
2.29.4 :   30.000
2.30.2 :    6.000

---- VAR CA   s aşamasında h
işinin m makinesinden ayrılma
zamanı

1.1 .4 :  144.000
1.2 .1 :  181.000
1.3 .2 :   74.000
1.4 .1 :   74.000
1.5 .6 :  195.000
1.6 .2 :  195.000
1.7 .5 :  146.000
1.8 .3 :  146.000
1.9 .2 :  158.000
1.10.4 :  200.000
1.11.4 :  179.000
1.12.3 :  179.000
1.13.6 :  110.000
1.14.4 :  110.000
1.15.1 :  153.000
1.16.6 :  244.000
1.17.6 :   59.000
1.18.6 :  152.000
1.19.3 :   65.000
1.20.4 :  235.000
```

FFS6J5 $\sum T_j$ amaç fonksiyonu için matematiksel model sonuçları

```
---- VAR C    r aşamasında j
işinin m makinesinden ayrılma
zamanı

1.1.3 :    9.000
1.2.2 :   11.000
1.3.2 :   21.000
1.4.1 :   16.000
1.5.3 :   25.000
2.1.3 :   28.000
2.3.3 :   44.000
2.4.1 :   36.000
2.5.2 :   41.000
3.1.2 :   38.000
3.2.1 :   26.000
3.4.1 :   51.000
3.5.2 :   51.000
4.1.1 :   50.000
4.2.3 :   44.000
4.3.2 :   56.000
4.4.1 :   63.000
4.5.3 :   64.000

---- VAR CA   s aşamasında h
işinin m makinesinden ayrılma
zamanı

1.1.4 :  105.000
1.2.1 :  105.000
1.3.2 :  108.000
1.4.3 :  108.000
2.1.4 :  145.000
2.2.1 :  145.000
2.3.3 :  137.000
2.4.2 :  137.000
```

FFS6J15 $\sum T_j$ amaç fonksiyonu için matematiksel model sonuçları 75378s.

```
---- VAR C    r aşamasında j
işinin m makinesinden ayrılma
zamanı

1.1 .3 :   36.000
1.2 .3 :   11.000
1.3 .3 :   22.000
1.4 .4 :   71.000
1.6 .2 :    8.000
1.7 .4 :   20.000
1.8 .2 :   21.000
1.9 .1 :   28.000
1.10.1 :    8.000
1.13.2 :   60.000
1.14.2 :   42.000
1.15.4 :   47.000
2.1 .5 :   47.000
2.2 .4 :   18.000
2.3 .6 :   33.000
2.4 .5 :   90.000
2.6 .1 :   27.000
2.7 .5 :   29.000
2.9 .3 :   34.000
2.10.4 :   34.000
2.11.2 :   16.000
2.12.3 :   28.000
2.13.5 :   70.000
2.14.4 :   51.000
2.15.6 :   67.000
3.3 .2 :   45.000
3.5 .6 :   10.000
3.6 .2 :   36.000
3.7 .5 :   35.000
3.8 .6 :   37.000
3.9 .6 :   55.000
3.10.3 :   56.000
3.11.3 :   49.000
3.12.4 :   39.000
3.13.1 :   77.000
3.14.4 :   68.000
4.1 .1 :   60.000
4.2 .2 :   37.000
4.3 .3 :   61.000
4.4 .1 :  106.000
4.5 .3 :  100.000
4.6 .3 :   47.000
4.7 .1 :   47.000
4.8 .2 :   47.000
4.9 .1 :   69.000
4.10.3 :   66.000
4.11.2 :   69.000
4.12.2 :   92.000
4.13.1 :   87.000
4.14.3 :   92.000
4.15.3 :   78.000

---- VAR CA   s aşamasında h
işinin m makinesinden ayrılma
zamanı

1.1 .6 :   94.000
1.2 .2 :   94.000
1.3 .4 :  197.000
1.4 .3 :  156.000
1.5 .5 :   71.000
1.6 .1 :   71.000
1.7 .4 :   90.000
1.8 .4 :  111.000
1.9 .1 :  128.000
1.10.4 :  147.000
2.1 .3 :  143.000
2.2 .2 :  143.000
2.3 .4 :  226.000
2.4 .6 :  229.000
2.5 .4 :  111.000
2.6 .1 :  111.000
2.7 .6 :  124.000
2.8 .6 :  158.000
2.9 .5 :  170.000
2.10.6 :  200.000
```

FFS6J30 $\sum T_j$ amaç fonksiyonu için matematiksel model sonuçları 86400s.

```
---- VAR C    r aşamasında j
işinin m makinesinden ayrılma
zamanı

1.2 .2 :  123.000
1.3 .1 :   68.000
1.4 .1 :  156.000
1.5 .2 :  139.000
1.6 .2 :  106.000
1.8 .2 :   87.000
1.9 .1 :  136.000
1.10.2 :   20.000
1.11.3 :   42.000
1.12.3 :  117.000
1.13.3 :   59.000
1.14.3 :   79.000
1.15.2 :   35.000
1.16.3 :    7.000
1.17.1 :   35.000
1.18.1 :  109.000
1.19.1 :   19.000
1.20.1 :   92.000
1.22.3 :  137.000
1.23.2 :   62.000
1.26.1 :   52.000
1.27.3 :  103.000
1.29.3 :   22.000
2.1 .2 :  190.000
2.2 .2 :  170.000
2.3 .3 :   82.000
2.4 .3 :  163.000
2.5 .3 :  181.000
2.6 .3 :  130.000
2.7 .1 :   52.000
2.8 .1 :  123.000
2.9 .3 :  156.000
2.10.3 :   35.000
2.11.3 :   61.000
2.12.2 :  131.000
2.13.1 :   72.000
2.14.1 :  137.000
2.15.2 :   65.000
2.17.2 :   49.000
2.19.2 :   37.000
2.20.3 :  104.000
2.21.1 :   24.000
2.22.2 :  155.000
2.23.2 :   85.000
2.24.1 :   43.000
2.25.1 :   13.000
2.26.2 :   75.000
2.27.1 :  112.000
2.28.2 :  136.000
2.29.3 :  115.000
```

```
2.30.3 :   10.000
3.2 .1 :  188.000
3.3 .3 :   95.000
3.4 .2 :  179.000
3.5 .3 :  186.000
3.7 .2 :  121.000
3.8 .4 :  129.000
3.10.2 :   45.000
3.11.1 :   68.000
3.12.4 :  139.000
3.14.4 :  149.000
3.15.2 :   72.000
3.16.4 :   15.000
3.17.4 :   69.000
3.18.1 :  116.000
3.19.4 :   57.000
3.20.2 :  109.000
3.21.2 :   34.000
3.22.3 :  169.000
3.23.4 :   99.000
3.24.1 :   52.000
3.25.4 :   33.000
3.28.3 :  150.000
3.29.2 :  135.000
3.30.2 :   16.000
4.1 .1 :  204.000
4.2 .4 :  227.000
4.3 .1 :  105.000
4.4 .3 :  184.000
4.5 .4 :  200.000
4.6 .3 :  150.000
4.7 .2 :  150.000
4.8 .3 :  140.000
4.9 .3 :  175.000
4.10.1 :   66.000
4.11.4 :   88.000
4.12.2 :  144.000
4.13.3 :   95.000
4.15.1 :   83.000
4.16.4 :   21.000
4.17.3 :   82.000
4.18.3 :  132.000
4.19.2 :   69.000
4.20.2 :  116.000
4.21.2 :   77.000
4.22.2 :  176.000
4.23.4 :  213.000
4.24.3 :   64.000
4.25.1 :   52.000
4.26.2 :   91.000
4.27.2 :  135.000
4.28.4 :  157.000
4.29.2 :  163.000
4.30.2 :   26.000

---- VAR CA    s aşamasında h
işinin m makinesinden ayrılma
zamanı

1.1 .2 :  262.000
1.2 .4 :  262.000
1.3 .5 :  222.000
1.4 .4 :  318.000
1.5 .1 :  191.000
1.6 .5 :  263.000
1.7 .3 :  208.000
1.8 .3 :  266.000
1.9 .5 :  172.000
1.10.5 :  286.000
1.11.4 :  132.000
1.12.5 :  132.000
1.13.6 :  157.000
1.14.3 :  233.000
1.15.4 :  296.000
1.16.5 :  320.000
1.17.2 :  122.000
1.18.6 :  122.000
1.19.4 :  193.000
1.20.6 :  193.000
2.1 .1 :  398.000
2.2 .4 :  292.000
2.3 .6 :  255.000
2.4 .4 :  351.000
2.5 .2 :  222.000
2.6 .2 :  365.000
2.7 .5 :  246.000
2.8 .4 :  389.000
2.9 .3 :  220.000
2.10.2 :  334.000
2.11.5 :  169.000
2.12.4 :  169.000
2.13.6 :  178.000
2.14.4 :  410.000
2.15.5 :  344.000
2.16.1 :  368.000
2.17.1 :  143.000
2.18.5 :  190.000
2.19.4 :  222.000
2.20.6 :  222.000
```

FFS6J66 $\sum T_j$ amaç fonksiyonu için matematiksel model sonuçları 7200s.

```
---- VAR C     r aşamasında j
işinin m makinesinden ayrılma
zamanı

1.1 .2 :   66.300
1.2 .1 :   49.600
1.3 .1 :  281.200
1.4 .1 :  172.300
1.5 .2 :  296.300
1.6 .2 :  415.500
1.7 .2 :  168.900
1.8 .2 :  216.300
1.9 .1 :   66.800
1.10.2 :  473.000
1.11.1 :   29.700
1.12.2 :  242.500
1.13.2 :  132.600
1.14.1 :  220.300
1.15.2 :  195.800
1.16.1 :  247.900
1.17.2 :  362.300
1.18.1 :   59.100
1.19.2 :   91.200
1.20.2 :   22.300
1.21.1 :  132.900
1.21.2 :  132.900
1.22.1 :  212.700
1.23.2 :  423.600
1.24.2 :  177.700
1.25.1 :    9.000
1.26.1 :  184.200
1.27.2 :  251.000
1.28.1 :  301.900
1.29.2 :   73.900
1.30.2 :  305.400
1.31.1 :   99.700
1.32.1 :  293.000
1.33.1 :  147.400
1.34.1 :   91.400
1.35.1 :  500.200
1.36.2 :  433.900
1.37.1 :  204.700
1.38.2 :  347.100
1.39.1 :  263.000
1.40.2 :   51.100
1.41.1 :  166.500
1.42.2 :  260.500
1.43.1 :  347.300
1.44.2 :  123.400
1.45.2 :   41.300
1.46.1 :   39.900
1.47.2 :  233.800
1.48.1 :  227.200
1.49.1 :  154.500
1.50.2 :   10.400
1.51.2 :  148.500
1.52.1 :  108.600
1.53.1 :   18.100
1.54.2 :  281.200
1.55.2 :  313.700
1.56.1 :  315.400
1.57.2 :  204.000
1.58.2 :  335.900
1.59.1 :  120.400
1.60.1 :  193.900
1.61.1 :  271.400
1.62.2 :   29.800
1.63.2 :  271.700
1.64.2 :  186.300
1.65.2 :  140.900
1.66.2 :  156.900
2.1 .2 :  247.700
2.2 .2 :  447.100
2.3 .2 :  293.000
2.5 .1 :  367.200
2.7 .1 :  325.900
2.8 .1 :  252.700
2.9 .1 :  238.200
2.10.1 :  484.700
2.11.2 :  415.000
2.13.1 :  447.300
2.15.2 :  338.300
2.16.1 :  306.100
2.17.1 :  495.400
2.18.2 :  461.600
2.19.2 :  232.700
2.20.1 :  232.100
2.21.1 :  209.400
2.22.2 :  390.000
2.23.2 :  432.900
2.24.1 :  284.200
2.25.1 :  221.100
2.26.2 :  402.500
2.28.1 :  395.000
2.30.2 :  458.600
2.31.1 :  182.000
2.32.1 :  313.600
2.33.1 :  199.300
2.34.2 :  176.700
2.35.2 :  508.300
2.36.2 :  514.700
2.37.2 :  326.600
2.38.2 :  361.000
2.39.1 :  419.000
2.41.1 :  172.900
2.42.2 :  367.900
2.45.2 :  387.300
2.46.2 :  160.500
2.47.2 :  314.600
2.50.2 :  298.600
2.51.2 :  151.600
2.52.1 :  338.400
2.53.1 :  407.000
2.54.2 :  440.900
2.55.2 :  342.400
2.56.1 :  472.500
2.57.2 :  210.600
2.58.1 :  510.600
2.59.1 :  321.600
2.60.2 :  423.600
2.61.1 :  276.200
2.62.2 :  347.100
2.63.2 :  379.300
2.64.1 :  456.500
2.65.2 :  223.700
3.40.1 :  147.000
3.43.1 :  591.900
3.44.1 :  297.900
3.48.1 :  469.200
3.49.1 :  391.500
4.1 .2 :  409.100
4.2 .2 :  487.200
4.3 .1 :  486.700
4.7 .1 :  465.100
4.8 .2 :  430.900
4.10.1 :  621.400
4.11.2 :  452.200
4.15.1 :  601.600
4.16.2 :  417.200
4.17.1 :  514.700
4.18.1 :  542.600
4.19.1 :  264.800
4.20.2 :  542.800
4.21.2 :  475.500
4.22.1 :  559.400
4.24.2 :  570.100
4.25.1 :  285.800
4.26.2 :  610.900
4.30.1 :  610.900
4.31.2 :  534.700
4.32.1 :  648.400
4.33.1 :  419.000
4.34.2 :  527.400
4.35.1 :  635.500
4.36.1 :  526.400
4.38.2 :  507.300
4.39.1 :  445.100
4.40.1 :  297.500
4.41.1 :  573.300
4.42.2 :  376.900
4.45.1 :  585.000
4.46.1 :  275.300
4.50.1 :  457.400
4.51.2 :  159.700
4.53.1 :  550.100
4.54.2 :  581.800
4.55.1 :  534.500
4.58.1 :  657.700
4.59.2 :  623.800
4.60.1 :  495.400
4.61.1 :  476.200
4.62.1 :  354.000
4.64.1 :  592.300
4.65.2 :  390.200

---- VAR CA    s aşamasında h
işinin m makinesinden ayrılma
zamanı

1.1 .4 :  590.800
1.2 .2 :  694.400
1.3 .2 :  969.800
1.4 .2 :  590.800
1.5 .4 : 1004.100
1.6 .2 : 1073.400
1.7 .6 :  741.900
```

```
1.8 .5 :  516.100
1.9 .5 :  553.300
1.10.3 : 1440.800
1.11.3 : 1541.400
1.12.4 : 1104.700
1.13.2 :  866.200
1.14.1 :  973.100
1.15.1 :  822.500
1.16.3 :  621.600
1.17.4 : 1254.900
1.18.2 : 1152.400
1.19.4 :  801.300
1.20.3 :  947.500
1.21.3 : 1241.200
1.22.3 : 1172.600
1.23.5 :  785.800
1.24.3 : 1058.500
1.25.3 : 1652.400
1.26.1 :  721.900
1.27.4 : 1592.600
1.28.3 :  803.800
1.29.2 : 1362.400
1.30.5 :  687.600
1.31.4 : 1397.600
1.32.4 :  722.300
1.33.6 :  802.300
1.34.2 : 1422.800
1.35.3 :  849.300
1.36.3 : 1104.000
1.37.3 : 1340.200
1.38.2 : 1251.400
1.39.6 :  690.900
1.40.6 : 1111.300
1.41.1 :  872.500
1.42.1 :  847.500
1.43.4 : 1175.900
1.44.2 :  765.600
1.45.3 :  692.800
1.46.3 :  878.900
1.47.4 :  830.900
1.48.5 :  717.200
1.49.4 :  900.500
1.50.4 :  661.900
2.1 .1 : 1156.200
2.2 .4 :  742.100
2.3 .2 : 1333.000
2.4 .4 :  638.500
2.5 .5 : 1051.800
2.6 .5 : 1215.400
2.7 .3 : 1813.200
2.8 .1 :  574.500
2.9 .3 :  611.900
2.10.3 : 1693.800
2.11.3 : 1592.600
2.12.1 : 1559.700
2.13.2 : 1463.600
2.14.4 : 1024.300
2.15.5 : 1310.700
2.16.3 : 1111.300
2.17.5 : 1456.800
2.18.1 : 1286.600
2.19.3 : 1213.100
2.20.3 :  991.600
2.21.2 : 1285.300
2.22.5 : 1259.500
2.23.5 :  829.900
2.24.1 : 1108.500
2.25.2 : 1702.400
2.26.1 : 1336.600
2.27.3 : 1642.600
2.28.5 : 1167.700
2.29.2 : 1412.400
2.30.5 : 1004.100
2.31.1 : 1460.400
2.32.1 :  820.500
2.33.2 :  900.500
2.34.2 : 1561.800
2.35.5 : 1117.700
2.36.3 : 1279.000
2.37.3 : 1754.800
2.38.1 : 1397.600
2.39.2 :  751.900
2.40.3 : 1141.500
2.41.3 : 1440.000
2.42.3 :  877.700
2.43.1 : 1215.000
2.44.3 :  804.700
2.45.4 : 1063.400
2.46.3 : 1039.700
2.47.1 : 1508.500
2.48.4 :  973.100
2.49.2 :  975.000
2.50.5 : 1385.200
```

FFS6J66 $\sum T_j$ amaç fonksiyonu için matematiksel model sonuçları 86400s.

```
---- VAR C    r aşamasında j
işinin m makinesinden ayrılma
zamanı

1.1 .2 :    89.000
1.2 .1 :   148.100
1.3 .1 :   231.400
1.4 .1 :   254.400
1.5 .2 :    25.500
1.6 .2 :   136.400
1.7 .2 :   221.700
1.8 .2 :   283.600
1.9 .1 :   138.400
1.10.2 :   180.300
1.11.1 :   393.800
1.12.2 :   255.400
1.13.2 :   385.500
1.14.1 :   355.100
1.15.2 :   293.100
1.16.1 :   198.100
1.17.2 :   129.100
1.18.1 :   157.600
1.19.2 :   113.900
1.20.2 :    54.800
1.21.1 :    97.100
1.22.1 :   413.600
1.23.2 :   301.200
1.24.2 :   230.500
1.25.1 :   364.100
1.26.1 :   266.300
1.27.2 :   394.000
1.28.1 :    60.900
1.29.2 :    96.600
1.30.2 :    34.600
1.31.1 :   239.700
1.32.1 :    52.000
1.33.1 :   111.600
1.34.2 :   161.000
1.35.1 :   177.400
1.36.2 :   171.300
1.37.1 :   286.800
1.38.2 :   344.100
1.39.1 :   213.200
1.40.2 :   190.100
1.41.1 :   130.700
1.42.2 :   310.700
1.43.1 :    40.200
1.44.2 :   376.300
1.45.2 :    73.800
1.46.1 :    71.100
1.47.2 :   411.500
1.48.1 :   164.500
1.49.1 :   118.700
1.50.2 :    10.400
1.51.2 :   246.700
1.52.1 :   248.600
1.53.1 :   382.200
1.54.2 :   264.900
1.55.2 :    42.900
1.56.1 :    84.600
1.57.2 :   273.100
1.58.2 :   332.900
1.59.1 :   405.600
1.60.1 :   276.000
1.61.1 :   221.600
1.62.2 :    62.300
1.63.2 :   209.700
1.64.2 :   239.100
1.65.1 :     8.300
1.66.2 :   198.500
2.1 .2 :   329.400
2.2 .2 :   373.400
2.3 .2 :   277.800
2.5 .1 :   496.500
2.7 .1 :   248.100
2.8 .1 :   459.200
2.9 .1 :   267.200
2.10.1 :   192.000
2.11.2 :   447.700
2.13.1 :   444.200
2.15.2 :   341.100
2.16.1 :   289.100
2.17.1 :   299.800
2.18.2 :   305.400
2.19.2 :   314.400
2.20.1 :   386.800
2.21.1 :   331.900
2.22.2 :   416.300
2.23.2 :   359.200
2.24.1 :   394.800
2.25.1 :   375.800
2.26.2 :   428.800
2.28.1 :    88.700
2.30.2 :   302.400
2.31.1 :   403.900
2.32.1 :   312.100
2.33.1 :   215.700
2.34.2 :   266.000
2.35.2 :   499.000
2.36.2 :   435.200
2.37.2 :   484.300
2.38.2 :   395.300
2.39.1 :   508.500
2.41.1 :   198.400
2.42.2 :   402.200
2.45.2 :   381.400
2.46.2 :    80.000
2.47.2 :   472.300
2.50.2 :    16.000
2.51.2 :   249.800
2.52.1 :   260.600
2.53.1 :   416.400
2.54.2 :   367.200
2.55.2 :   345.200
2.56.1 :   348.400
2.57.2 :   490.900
2.58.1 :   364.100
2.59.1 :   467.200
2.60.2 :   456.300
2.61.1 :   304.600
2.62.2 :   349.900
2.63.2 :   413.600
2.64.1 :   321.800
2.65.2 :   290.900
3.40.1 :   293.400
3.43.1 :   162.900
3.44.1 :   527.200
3.48.1 :   242.200
3.49.1 :   620.800
4.1 .2 :   508.100
4.2 .2 :   445.600
4.3 .1 :   469.600
4.7 .1 :   419.600
4.8 .2 :   549.100
4.10.1 :   570.400
4.11.2 :   570.400
4.15.1 :   350.400
4.16.2 :   516.200
4.17.1 :   641.400
4.18.1 :   451.600
4.19.1 :   514.600
4.20.2 :   394.900
4.21.2 :   355.200
4.22.1 :   435.400
4.24.2 :   422.200
4.25.1 :   386.300
4.26.2 :   466.900
4.30.1 :   359.700
4.31.2 :   617.900
4.32.1 :   654.300
4.33.1 :   533.800
4.34.2 :   610.600
4.35.1 :   677.700
4.36.1 :   613.400
4.38.2 :   590.500
4.39.1 :   559.900
4.40.1 :   398.000
4.41.1 :   411.900
4.42.2 :   475.900
4.45.1 :   594.400
4.46.1 :   370.200
4.50.1 :   582.700
4.51.2 :   363.300
4.53.1 :   459.100
4.54.2 :   433.900
4.55.1 :   443.500
4.58.1 :   663.600
4.59.1 :   482.500
4.60.1 :   622.100
4.61.1 :   426.100
4.62.1 :   521.500
4.64.1 :   601.700
4.65.2 :   489.200

---- VAR CA    s aşamasında h
işinin m makinesinden ayrılma
zamanı

1.1 .5 :   936.300
1.2 .5 :   832.700
1.3 .2 :  1331.200
1.4 .2 :  1045.000
1.5 .4 :   841.700
1.6 .2 :  1148.600
1.7 .6 :   744.400
1.8 .5 :   600.100
1.9 .5 :   637.300
1.10.3 :  1120.500
1.11.3 :  1221.100
1.12.4 :   942.300
1.13.2 :   671.000
1.14.1 :   866.200
```

```
1.15.1 :   715.600
1.16.3 :   762.600
1.17.4 :  1191.700
1.18.2 :  1227.600
1.19.4 :  1420.900
1.20.3 :   572.600
1.21.3 :  1400.700
1.22.3 :   504.000
1.23.1 :   504.000
1.24.3 :   683.600
1.25.3 :  1332.100
1.26.1 :   615.000
1.27.4 :  1341.900
1.28.3 :   944.800
1.29.2 :   881.000
1.30.5 :   699.500
1.31.4 :  1230.900
1.32.4 :   738.100
1.33.6 :   804.800
1.34.2 :   941.400
1.35.3 :   990.300
1.36.3 :  1545.200
1.37.3 :  1499.700
1.38.2 :   770.000
1.39.6 :   693.400
1.40.6 :   829.800
1.41.1 :   765.600
1.42.1 :   740.600
1.43.4 :  1112.700
1.44.2 :   562.100
1.45.3 :   833.800
1.46.3 :  1019.900
1.47.4 :   971.900
1.48.5 :   729.100
1.49.4 :  1041.500
1.50.4 :   671.300
2.1 .1 :   984.000
2.2 .4 :   880.400
2.3 .2 :  1492.500
2.4 .4 :  1092.700
2.5 .5 :   965.500
2.6 .5 :  1311.000
2.7 .3 :  1619.100
2.8 .1 :  1261.400
2.9 .3 :   854.400
2.10.3 :  1265.100
2.11.3 :  1370.900
2.12.1 :  1203.000
2.13.2 :  1217.200
2.14.4 :   931.600
2.15.5 :   766.800
2.16.3 :  1139.600
2.17.5 :  1263.300
2.18.1 :  1333.000
2.19.3 :  1492.500
2.20.3 :  1213.900
2.21.2 :  1444.800
2.22.5 :   855.000
2.23.5 :   810.900
2.24.1 :   886.300
2.25.2 :  1382.100
2.26.1 :   936.300
2.27.3 :  1420.900
2.28.5 :  1106.200
2.29.2 :  1166.000
2.30.5 :   917.800
2.31.1 :  1443.900
2.32.1 :   836.300
2.33.2 :   903.000
2.34.2 :  1332.100
2.35.5 :  1056.200
2.36.3 :  1685.000
2.37.3 :  1560.700
2.38.1 :  1504.900
2.39.2 :   754.400
2.40.3 :   923.700
2.41.3 :   795.800
2.42.3 :  1169.800
2.43.1 :  1151.800
2.44.3 :   893.500
2.45.4 :  1018.800
2.46.3 :  1068.000
2.47.1 :  1381.100
2.48.4 :   979.700
2.49.2 :  1116.000
2.50.5 :  1180.700
```

FFS6J66 $\sum T_j$ amaç fonksiyonu için matematiksel model sonuçları 259200s.

```
---- VAR C    r aşamasında j
işinin m makinesinden ayrılma
zamanı

1.1 .1 :  249.600
1.2 .2 :  346.300
1.3 .1 :  183.600
1.4 .1 :  173.800
1.5 .2 :  255.900
1.6 .2 :   76.500
1.7 .2 :  179.900
1.8 .2 :  380.600
1.9 .1 :  199.600
1.10.2 :  100.000
1.11.1 :  149.000
1.12.2 :  355.000
1.13.1 :  168.000
1.14.1 :  295.900
1.15.1 :  346.000
1.16.2 :   69.200
1.17.1 :  120.800
1.18.1 :   70.200
1.19.1 :  216.900
1.20.2 :  136.000
1.21.1 :  308.400
1.22.1 :  354.000
1.23.2 :   48.500
1.24.1 :  336.500
1.25.1 :  269.000
1.26.1 :  365.900
1.27.1 :    8.500
1.28.2 :  389.500
1.29.2 :  115.800
1.30.1 :  317.500
1.31.1 :  191.900
1.32.2 :  294.000
1.33.2 :   91.000
1.34.1 :  105.600
1.35.1 :   60.700
1.36.2 :  274.600
1.37.1 :   81.000
1.38.2 :   40.400
1.39.2 :  370.100
1.40.1 :  158.800
1.41.2 :  314.400
1.42.2 :  189.400
1.43.2 :  167.900
1.44.1 :   47.800
1.45.2 :  214.400
1.46.1 :  327.700
1.47.1 :  234.400
1.48.1 :  127.700
1.49.1 :   15.600
1.50.1 :  260.000
1.51.2 :  282.200
1.52.2 :    8.900
1.53.2 :   29.200
1.54.2 :  232.500
1.55.2 :  124.100
1.56.2 :  202.900
1.57.2 :  108.200
1.58.2 :  336.600
1.59.1 :  288.300
1.60.1 :  137.400
1.61.2 :  264.300
1.62.1 :  276.500
1.63.2 :   20.100
1.64.2 :  223.000
1.65.2 :  240.800
1.66.2 :  302.400
2.1 .2 :  305.500
2.2 .2 :  437.400
2.3 .2 :  417.900
2.5 .1 :  284.700
2.7 .1 :  453.100
2.8 .1 :  422.600
2.9 .1 :  349.700
2.10.1 :  231.100
2.11.2 :  161.500
2.13.1 :  195.800
2.15.2 :  380.500
2.16.1 :   91.100
2.17.1 :  433.300
2.18.2 :  451.900
2.19.2 :  314.500
2.20.1 :  147.000
2.21.1 :  374.500
2.22.2 :  356.700
2.23.2 :  290.500
2.24.1 :  441.300
2.25.1 :  296.900
2.26.2 :  393.000
2.28.1 :  481.400
2.30.2 :  448.900
2.31.1 :  343.600
2.32.1 :  448.800
2.33.1 :  403.300
2.34.2 :  177.700
2.35.2 :   68.800
2.36.2 :  424.300
2.37.2 :  262.400
2.38.2 :  191.600
2.39.1 :  386.000
2.41.1 :  321.500
2.42.2 :  431.200
2.45.2 :  368.800
2.46.2 :  348.400
2.47.2 :  250.400
2.50.2 :  320.100
2.51.2 :  351.500
2.52.1 :  243.600
2.53.1 :  334.000
2.54.2 :  328.100
2.55.2 :  360.800
2.56.1 :  218.900
2.57.2 :  269.000
2.58.1 :  364.900
2.59.1 :  304.900
2.60.2 :  146.000
2.61.1 :  408.100
2.62.2 :  281.200
2.63.2 :  339.500
2.64.1 :  314.600
2.65.2 :  406.100
3.40.1 :  249.900
3.43.1 :  466.200
3.44.1 :  198.700
3.48.1 :  543.900
3.49.1 :  343.500
4.1 .1 :  434.700
4.2 .1 :  569.500
4.3 .1 :  518.200
4.7 .1 :  577.200
4.8 .1 :  603.200
4.10.2 :  317.900
4.11.2 :  182.800
4.15.1 :  389.800
4.16.1 :  475.000
4.17.2 :  641.100
4.18.2 :  554.700
4.19.2 :  586.800
4.20.1 :  155.100
4.21.2 :  610.100
4.22.1 :  527.500
4.24.2 :  546.600
4.25.2 :  307.400
4.26.1 :  557.800
4.30.1 :  458.200
4.31.1 :  442.000
4.32.1 :  496.000
4.33.2 :  415.600
4.34.2 :  361.600
4.35.1 :  169.200
4.36.1 :  507.700
4.38.2 :  466.000
4.39.2 :  519.300
4.40.2 :  621.800
4.41.2 :  493.200
4.42.1 :  536.500
4.45.2 :  435.400
4.46.2 :  445.900
4.50.1 :  589.500
4.51.2 :  423.700
4.53.2 :  341.500
4.54.1 :  339.800
4.55.1 :  483.100
4.58.1 :  380.500
4.59.1 :  317.800
4.60.1 :  466.900
4.61.1 :  414.600
4.62.1 :  448.900
4.64.2 :  368.900
4.65.2 :  479.300

---- VAR CA    s aşamasında h
işinin m makinesinden ayrılma
zamanı

1.1 .4 :  776.700
1.2 .4 :  673.100
1.3 .5 : 1320.800
1.4 .1 :  771.800
1.5 .3 :  772.400
1.6 .6 :  736.900
1.7 .5 :  988.000
1.8 .3 :  823.400
1.9 .6 :  533.700
1.10.6 : 1189.700
1.11.4 : 1207.400
1.12.2 : 1006.000
1.13.5 :  490.400
1.14.4 :  490.400
1.15.5 :  937.000
1.16.2 : 1216.600
1.17.5 : 1106.200
1.18.1 :  850.800
1.19.6 :  926.900
1.20.1 :  919.400
```

```
1.21.4 : 1106.800
1.22.6 :  995.500
1.23.6 : 1064.100
1.24.5 : 1217.200
1.25.4 : 1318.400
1.26.5 :  811.400
1.27.3 :  668.800
1.28.5 :  700.400
1.29.6 :  847.900
1.30.6 :  633.300
1.31.5 : 1027.200
1.32.6 :  594.100
1.33.1 :  568.100
1.34.2 : 1137.600
1.35.1 :  668.200
1.36.2 :  834.200
1.37.2 :  647.900
1.38.5 :  589.400
1.39.6 : 1288.700
1.40.1 :  622.700
1.41.6 : 1089.100
1.42.5 :  836.400
1.43.2 : 1077.200
1.44.2 :  719.100
1.45.2 :  905.400
1.46.1 :  597.700
1.47.6 :  496.500
1.48.1 :  496.500
1.49.2 :  548.900
1.50.2 :  788.700
2.1 .2 : 1453.300
2.2 .5 :  947.800
2.3 .3 : 1368.500
2.4 .2 : 1179.000
2.5 .1 : 1187.100
2.6 .2 :  907.200
2.7 .1 : 1046.400
2.8 .2 : 1015.600
2.9 .4 :  592.300
2.10.1 : 1240.900
2.11.3 : 1419.700
2.12.4 : 1531.800
2.13.3 :  717.500
2.14.4 : 1318.400
2.15.4 : 1419.600
2.16.2 : 1585.900
2.17.2 : 1313.400
2.18.2 : 1087.200
2.19.4 :  998.500
2.20.2 : 1357.500
2.21.3 : 1320.800
2.22.3 : 1276.700
2.23.2 : 1131.300
2.24.4 : 1267.200
2.25.4 : 1368.400
2.26.2 :  957.200
2.27.5 :  718.800
2.28.2 :  750.400
2.29.5 : 1028.000
2.30.1 : 1139.400
2.31.2 : 1241.800
2.32.3 :  815.700
2.33.3 :  666.300
2.34.3 : 1517.900
2.35.1 :  734.100
2.36.5 :  900.100
2.37.2 : 1514.300
2.38.2 :  859.500
2.39.4 : 1480.600
2.40.5 :  978.000
2.41.5 : 1119.300
2.42.1 : 1076.600
2.43.4 : 1116.300
2.44.5 :  758.200
2.45.4 : 1037.600
2.46.2 : 1405.600
2.47.2 :  798.500
2.48.4 :  640.400
2.49.1 :  937.700
2.50.1 :  863.200
```

FFS3J5 Uyarlanmış Johnson Kuralı Sonuçları

```
ST1.5.1 : 0  C1.5.1 : 12
ST2.5.3 : 12  C2.5.3 : 18
ST2.4.4 : 0  C2.4.4 : 5
STA1.3.1 : 18  F1.3.1 : 46
STA1.4.2 : 18  F1.4.2 : 46
ST1.3.2 : 0  C1.3.2 : 20
ST2.3.1 : 20  C2.3.1 : 33
ST1.2.3 : 0  C1.2.3 : 8
ST2.2.2 : 8  C2.2.2 : 24
ST1.1.3 : 10  C1.1.3 : 23
STA1.1.3 : 33  F1.1.3 : 75
STA1.2.4 : 33  F1.2.4 : 75
```

FFS3J15 Uyarlanmış Johnson Kuralı Sonuçları

```
ST1.4.1 : 0  C1.4.1 : 18
ST2.4.3 : 18  C2.4.3 : 28
ST2.5.4 : 0  C2.5.4 : 12
STA1.3.1 : 28  F1.3.1 : 69
STA1.4.2 : 28  F1.4.2 : 69
ST1.7.2 : 0  C1.7.2 : 8
ST2.7.3 : 28  C2.7.3 : 46
ST1.6.3 : 0  C1.6.3 : 17
ST2.6.4 : 17  C2.6.4 : 24
ST1.8.4 : 0  C1.8.4 : 10
STA1.5.3 : 46  F1.5.3 : 84
STA1.6.4 : 46  F1.6.4 : 84
ST1.14.5 : 0  C1.14.5 : 15
ST2.14.3 : 46  C2.14.3 : 63
ST1.13.2 : 16  C1.13.2 : 27
ST2.13.1 : 27  C2.13.1 : 37
ST1.15.5 : 17  C1.15.5 : 35
ST2.12.3 : 63  C2.12.3 : 80
STA1.9.5 : 80  F1.9.5 : 120
STA1.10.1 : 80  F1.10.1 : 120
ST1.10.4 : 18  C1.10.4 : 32
ST2.10.2 : 32  C2.10.2 : 40
ST2.11.1 : 37  C2.11.1 : 56
ST1.9.3 : 25  C1.9.3 : 38
STA1.7.2 : 69  F1.7.2 : 90
STA1.8.3 : 84  F1.8.3 : 105
ST1.3.1 : 26  C1.3.1 : 44
ST2.3.2 : 44  C2.3.2 : 61
ST1.1.2 : 35  C1.1.2 : 46
ST2.1.2 : 61  C2.1.2 : 72
ST1.2.4 : 40  C1.2.4 : 48
STA1.1.4 : 84  F1.1.4 : 105
STA1.2.2 : 90  F1.2.2 : 111
```

FFS3J30 Uyarlanmış Johnson Kuralı Sonuçları

```
ST1.26.1 : 0  C1.26.1 : 17
ST2.24.4 : 0  C2.24.4 : 7
ST2.25.2 : 0  C2.25.2 : 6
STA1.17.1 : 17  F1.17.1 : 59
STA1.18.2 : 17  F1.18.2 : 59
ST1.22.2 : 0  C1.22.2 : 15
ST2.22.4 : 15  C2.22.4 : 34
ST1.23.3 : 0  C1.23.3 : 10
ST2.21.2 : 6  C2.21.2 : 15
STA1.15.3 : 34  F1.15.3 : 83
STA1.16.4 : 34  F1.16.4 : 83
ST1.20.3 : 18  C1.20.3 : 37
ST2.20.1 : 37  C2.20.1 : 56
ST1.19.2 : 23  C1.19.2 : 32
ST1.18.1 : 25  C1.18.1 : 33
STA1.13.5 : 56  F1.13.5 : 103
STA1.14.6 : 56  F1.14.6 : 103
ST1.29.2 : 40  C1.29.2 : 49
ST2.29.4 : 49  C2.29.4 : 60
ST1.27.1 : 41  C1.27.1 : 49
ST2.27.4 : 60  C2.27.4 : 65
ST2.28.1 : 56  C2.28.1 : 62
ST2.30.2 : 15  C2.30.2 : 21
STA1.19.1 : 65  F1.19.1 : 100
STA1.20.2 : 65  F1.20.2 : 100
ST1.4.3 : 45  C1.4.3 : 63
ST2.4.2 : 63  C2.4.2 : 80
ST1.5.1 : 57  C1.5.1 : 69
STA1.3.3 : 83  F1.3.3 : 118
STA1.4.4 : 83  F1.4.4 : 118
ST1.11.2 : 57  C1.11.2 : 72
ST2.11.2 : 80  C2.11.2 : 98
ST1.9.3 : 71  C1.9.3 : 85
ST2.9.1 : 85  C2.9.1 : 103
ST1.10.2 : 74  C1.10.2 : 89
STA1.7.1 : 103  F1.7.1 : 150
STA1.8.2 : 103  F1.8.2 : 150
ST1.2.1 : 77  C1.2.1 : 94
ST2.2.2 : 98  C2.2.2 : 106
ST2.1.3 : 0  C2.1.3 : 19
ST1.3.3 : 93  C1.3.3 : 98
ST2.3.3 : 98  C2.3.3 : 111
STA1.1.5 : 111  F1.1.5 : 139
STA1.2.6 : 111  F1.2.6 : 139
ST1.6.2 : 97  C1.6.2 : 115
ST2.6.2 : 115  C2.6.2 : 126
ST1.8.1 : 102  C1.8.1 : 118
ST2.8.1 : 118  C2.8.1 : 123
ST2.7.3 : 111  C2.7.3 : 129
STA1.5.3 : 129  F1.5.3 : 157
STA1.6.4 : 129  F1.6.4 : 157
ST1.12.3 : 106  C1.12.3 : 120
ST2.12.4 : 120  C2.12.4 : 134
ST1.14.3 : 122  C1.14.3 : 136
ST2.14.2 : 136  C2.14.2 : 148
ST1.13.2 : 123  C1.13.2 : 137
ST2.13.3 : 137  C2.13.3 : 144
STA1.9.5 : 148  F1.9.5 : 169
STA1.10.6 : 148  F1.10.6 : 169
ST1.15.1 : 126  C1.15.1 : 141
ST2.15.3 : 144  C2.15.3 : 164
ST1.16.1 : 143  C1.16.1 : 161
ST2.16.3 : 164  C2.16.3 : 176
ST1.17.3 : 144  C1.17.3 : 154
ST2.17.1 : 154  C2.17.1 : 173
STA1.11.1 : 176  F1.11.1 : 197
STA1.12.2 : 176  F1.12.2 : 197
```

FFS6J5 Uyarlanmış Johnson Kuralı Sonuçları

```
ST1.1.1 : 0  C1.1.1 : 9
ST2.1.3 : 9  C2.1.3 : 28
ST3.1.1 : 28  C3.1.1 : 38
ST4.1.1 : 38  C4.1.1 : 50
ST1.2.2 : 0  C1.2.2 : 11
ST3.2.2 : 11  C3.2.2 : 26
ST4.2.2 : 26  C4.2.2 : 44
ST1.3.3 : 0  C1.3.3 : 8
ST2.3.3 : 28  C2.3.3 : 44
ST4.3.3 : 44  C4.3.3 : 56
STA1.1.1 : 56  F1.1.1 : 105
STA2.1.1 : 105  F2.1.1 : 145
STA1.2.2 : 56  F1.2.2 : 105
STA2.2.2 : 105  F2.2.2 : 145
ST1.4.3 : 8  C1.4.3 : 24
ST2.4.1 : 24  C2.4.1 : 44
ST3.4.2 : 44  C3.4.2 : 59
ST4.4.2 : 59  C4.4.2 : 71
ST1.5.1 : 17  C1.5.1 : 25
ST2.5.2 : 25  C2.5.2 : 41
ST3.5.1 : 41  C3.5.1 : 51
ST4.5.1 : 51  C4.5.1 : 64
STA1.3.3 : 71  F1.3.3 : 115
STA2.3.3 : 115  F2.3.3 : 144
STA1.4.4 : 71  F1.4.4 : 115
STA2.4.4 : 115  F2.4.4 : 144
```

FFS6J15 Uyarlanmış Johnson Kuralı Sonuçları

```
ST1.4.1 : 0  C1.4.1 : 16
ST2.4.5 : 16  C2.4.5 : 35
ST4.4.1 : 35  C4.4.1 : 51
ST3.5.1 : 0  C3.5.1 : 10
ST4.5.2 : 10  C4.5.2 : 18
STA1.3.1 : 51  F1.3.1 : 101
STA2.3.1 : 101  F2.3.1 : 130
STA1.4.2 : 51  F1.4.2 : 101
STA2.4.2 : 101  F2.4.2 : 130
ST1.3.2 : 0  C1.3.2 : 9
ST2.3.6 : 9  C2.3.6 : 20
ST3.3.2 : 20  C3.3.2 : 29
ST4.3.3 : 29  C4.3.3 : 43
ST1.2.3 : 0  C1.2.3 : 11
ST2.2.4 : 11  C2.2.4 : 18
```

```
ST4.2.2 : 18  C4.2.2 : 37
ST1.1.4 : 0  C1.1.4 : 12
ST2.1.5 : 35  C2.1.5 : 44
ST4.1.2 : 44  C4.1.2 : 57
STA1.1.3 : 57  F1.1.3 : 90
STA2.1.3 : 90  F2.1.3 : 139
STA1.2.4 : 57  F1.2.4 : 90
STA2.2.4 : 90  F2.2.4 : 139
ST1.6.2 : 17  C1.6.2 : 25
ST2.6.1 : 25  C2.6.1 : 44
ST3.6.3 : 44  C3.6.3 : 53
ST4.6.3 : 53  C4.6.3 : 64
ST1.7.3 : 19  C1.7.3 : 39
ST2.7.5 : 44  C2.7.5 : 53
ST3.7.4 : 53  C3.7.4 : 59
ST4.7.1 : 59  C4.7.1 : 71
ST1.8.4 : 20  C1.8.4 : 31
ST3.8.5 : 31  C3.8.5 : 47
ST4.8.2 : 57  C4.8.2 : 67
```

```
STA1.5.5 : 71  F1.5.5 : 95
STA2.5.5 : 95  F2.5.5 : 135
STA1.6.6 : 71  F1.6.6 : 95
STA2.6.6 : 95  F2.6.6 : 135
ST1.14.1 : 24  C1.14.1 : 37
ST2.14.4 : 37  C2.14.4 : 46
ST3.14.6 : 46  C3.14.6 : 63
ST4.14.3 : 64  C4.14.3 : 78
ST1.15.2 : 33  C1.15.2 : 52
ST2.15.6 : 52  C2.15.6 : 70
ST4.15.2 : 70  C4.15.2 : 81
ST2.12.3 : 0  C2.12.3 : 14
ST3.12.1 : 14  C3.12.1 : 25
ST4.12.1 : 71  C4.12.1 : 89
ST1.13.1 : 39  C1.13.1 : 55
ST2.13.5 : 55  C2.13.5 : 65
ST3.13.1 : 65  C3.13.1 : 72
ST4.13.3 : 78  C4.13.3 : 88
STA1.9.3 : 90  F1.9.3 : 126
```

```
STA2.9.1 : 130  F2.9.1 : 172
STA1.10.4 : 90  F1.10.4 : 126
STA2.10.2 : 130  F2.10.2 : 172
ST2.11.2 : 0  C2.11.2 : 16
ST3.11.2 : 29  C3.11.2 : 45
ST4.11.2 : 81  C4.11.2 : 101
ST1.9.4 : 39  C1.9.4 : 57
ST2.9.3 : 57  C2.9.3 : 63
ST3.9.2 : 63  C3.9.2 : 81
ST4.9.3 : 88  C4.9.3 : 97
ST1.10.3 : 47  C1.10.3 : 55
ST2.10.4 : 55  C2.10.4 : 71
ST3.10.5 : 71  C3.10.5 : 78
ST4.10.1 : 89  C4.10.1 : 94
STA1.7.5 : 101  F1.7.5 : 122
STA2.7.5 : 135  F2.7.5 : 169
STA1.8.6 : 101  F1.8.6 : 122
STA2.8.6 : 135  F2.8.6 : 169
```

FFS6J30 Uyarlanmış Johnson Kuralı Sonuçları

```
ST1.17.1 : 0  C1.17.1 : 8
ST2.17.2 : 8  C2.17.2 : 20
ST3.17.1 : 20  C3.17.1 : 32
ST4.17.1 : 32  C4.17.1 : 45
ST1.15.2 : 0  C1.15.2 : 7
ST2.15.2 : 20  C2.15.2 : 36
ST3.15.2 : 36  C3.15.2 : 43
ST4.15.2 : 43  C4.15.2 : 54
ST1.16.3 : 0  C1.16.3 : 7
ST3.16.3 : 7  C3.16.3 : 15
ST4.16.3 : 15  C4.16.3 : 21
STA1.11.1 : 54  F1.11.1 : 103
STA2.11.1 : 103  F2.11.1 : 140
STA1.12.2 : 54  F1.12.2 : 103
STA2.12.2 : 103  F2.12.2 : 140
ST1.8.2 : 15  C1.8.2 : 32
ST2.8.1 : 32  C2.8.1 : 43
ST3.8.4 : 43  C3.8.4 : 49
ST4.8.4 : 49  C4.8.4 : 57
ST1.6.3 : 15  C1.6.3 : 32
ST2.6.3 : 32  C2.6.3 : 47
ST4.6.3 : 47  C4.6.3 : 54
ST2.7.1 : 43  C2.7.1 : 52
ST3.7.3 : 52  C3.7.3 : 64
ST4.7.1 : 64  C4.7.1 : 70
STA1.5.3 : 70  F1.5.3 : 111
STA2.5.3 : 111  F2.5.3 : 142
STA1.6.4 : 70  F1.6.4 : 111
STA2.6.4 : 111  F2.6.4 : 142
ST1.5.1 : 16  C1.5.1 : 24
ST2.5.3 : 47  C2.5.3 : 65
ST3.5.1 : 65  C3.5.1 : 70
ST4.5.2 : 70  C4.5.2 : 84
ST1.4.1 : 26  C1.4.1 : 38
ST2.4.3 : 65  C2.4.3 : 72
ST3.4.2 : 72  C3.4.2 : 88
ST4.4.3 : 88  C4.4.3 : 93
STA1.3.5 : 93  F1.3.5 : 115
STA2.3.5 : 115  F2.3.5 : 148
STA1.4.6 : 93  F1.4.6 : 115
STA2.4.6 : 115  F2.4.6 : 148
ST1.14.2 : 40  C1.14.2 : 58
ST2.14.1 : 58  C2.14.1 : 72
ST3.14.4 : 72  C3.14.4 : 82
ST1.13.3 : 40  C1.13.3 : 49
ST2.13.1 : 72  C2.13.1 : 85
ST4.13.4 : 85  C4.13.4 : 98
ST1.12.1 : 46  C1.12.1 : 52
```

```
ST2.12.2 : 52  C2.12.2 : 66
ST3.12.3 : 66  C3.12.3 : 74
ST4.12.1 : 74  C4.12.1 : 79
STA1.9.1 : 103  F1.9.1 : 126
STA2.9.1 : 140  F2.9.1 : 188
STA1.10.2 : 103  F1.10.2 : 126
STA2.10.2 : 140  F2.10.2 : 188
ST1.23.3 : 57  C1.23.3 : 76
ST2.23.2 : 76  C2.23.2 : 86
ST3.23.1 : 86  C3.23.1 : 100
ST4.23.1 : 100  C4.23.1 : 113
ST1.22.1 : 60  C1.22.1 : 72
ST2.22.2 : 86  C2.22.2 : 104
ST3.22.3 : 104  C3.22.3 : 118
ST4.22.2 : 118  C4.22.2 : 125
ST2.21.1 : 85  C2.21.1 : 96
ST3.21.4 : 96  C3.21.4 : 106
ST4.21.3 : 106  C4.21.3 : 114
STA1.15.3 : 125  F1.15.3 : 159
STA2.15.3 : 159  F2.15.3 : 207
STA1.16.4 : 125  F1.16.4 : 159
STA2.16.4 : 159  F2.16.4 : 207
ST1.2.2 : 66  C1.2.2 : 75
ST2.2.2 : 104  C2.2.2 : 119
ST3.2.2 : 119  C3.2.2 : 137
ST4.2.4 : 137  C4.2.4 : 151
ST1.3.2 : 77  C1.3.2 : 85
ST2.3.3 : 85  C2.3.3 : 99
ST3.3.1 : 100  C3.3.1 : 113
ST4.3.1 : 113  C4.3.1 : 123
ST2.1.2 : 119  C2.1.2 : 139
ST4.1.3 : 139  C4.1.3 : 153
STA1.1.5 : 153  F1.1.5 : 188
STA2.1.5 : 188  F2.1.5 : 218
STA1.2.6 : 153  F1.2.6 : 188
STA2.2.6 : 188  F2.2.6 : 218
ST2.25.1 : 96  C2.25.1 : 109
ST3.25.4 : 109  C3.25.4 : 127
ST4.25.1 : 127  C4.25.1 : 146
ST2.24.1 : 109  C2.24.1 : 128
ST3.24.1 : 128  C3.24.1 : 137
ST4.24.2 : 137  C4.24.2 : 149
ST1.26.1 : 80  C1.26.1 : 89
ST2.26.2 : 139  C2.26.2 : 149
ST4.26.1 : 149  C4.26.1 : 163
STA1.17.1 : 163  F1.17.1 : 194
STA2.17.1 : 194  F2.17.1 : 215
STA1.18.2 : 163  F1.18.2 : 194
```

```
STA2.18.2 : 194  F2.18.2 : 215
ST1.29.3 : 84  C1.29.3 : 91
ST2.29.3 : 99  C2.29.3 : 110
ST3.29.3 : 118  C3.29.3 : 132
ST4.29.2 : 149  C4.29.2 : 162
ST1.27.2 : 93  C1.27.2 : 109
ST2.27.1 : 128  C2.27.1 : 137
ST4.27.4 : 151  C4.27.4 : 170
ST2.28.2 : 149  C2.28.2 : 154
ST3.28.4 : 154  C3.28.4 : 168
ST4.28.3 : 168  C4.28.3 : 175
ST2.30.3 : 110  C2.30.3 : 120
ST3.30.3 : 132  C3.30.3 : 138
ST4.30.2 : 162  C4.30.2 : 172
STA1.19.3 : 175  F1.19.3 : 205
STA2.19.3 : 207  F2.19.3 : 236
STA1.20.4 : 175  F1.20.4 : 205
STA2.20.4 : 207  F2.20.4 : 236
ST1.10.1 : 97  C1.10.1 : 117
ST2.10.3 : 120  C2.10.3 : 135
ST3.10.1 : 137  C3.10.1 : 147
ST4.10.1 : 163  C4.10.1 : 177
ST1.9.3 : 99  C1.9.3 : 118
ST2.9.3 : 135  C2.9.3 : 155
ST4.9.4 : 170  C4.9.4 : 189
ST1.11.2 : 117  C1.11.2 : 129
ST2.11.3 : 155  C2.11.3 : 174
ST3.11.2 : 174  C3.11.2 : 181
ST4.11.2 : 181  C4.11.2 : 201
STA1.7.5 : 201  F1.7.5 : 234
STA2.7.1 : 234  F2.7.1 : 272
STA1.8.6 : 201  F1.8.6 : 234
STA2.8.2 : 234  F2.8.2 : 272
ST1.19.1 : 125  C1.19.1 : 144
ST2.19.2 : 154  C2.19.2 : 172
ST3.19.3 : 172  C3.19.3 : 192
ST4.19.3 : 192  C4.19.3 : 204
ST1.20.3 : 126  C1.20.3 : 142
ST2.20.3 : 174  C2.20.3 : 186
ST3.20.1 : 186  C3.20.1 : 191
ST4.20.1 : 191  C4.20.1 : 198
ST1.18.2 : 137  C1.18.2 : 152
ST3.18.4 : 168  C3.18.4 : 175
ST4.18.4 : 189  C4.18.4 : 205
STA1.13.1 : 205  F1.13.1 : 230
STA2.13.5 : 230  F2.13.5 : 251
STA1.14.2 : 205  F1.14.2 : 230
STA2.14.6 : 230  F2.14.6 : 251
```

FFS6J66 Uyarlanmış Johnson Kuralı Sonuçları

```
ST1.9.1 : 0  C1.9.1 : 7,7
ST2.9.1 : 7,7  C2.9.1 : 13,8
STA1.9.1 : 13,8  F1.9.1 : 51
STA2.9.1 : 51  F2.9.1 : 109,6
ST1.26.2 : 0  C1.26.2 : 11,9
ST2.26.2 : 11,9  C2.26.2 : 24,4
ST4.26.1 : 24,4  C4.26.1 : 45,7
ST1.28.1 : 7,7  C1.28.1 : 16,6
ST2.28.1 : 16,6  C2.28.1 : 44,4
ST1.25.2 : 11,9  C1.25.2 : 20,9
ST2.25.1 : 44,9  C2.25.1 : 56,6
ST4.25.2 : 56,6  C4.25.2 : 67,1
ST1.30.1 : 16,6  C1.30.1 : 25,7
ST2.30.2 : 25,7  C2.30.2 : 37,2
ST4.30.1 : 45,7  C4.30.1 : 55
ST1.27.2 : 20,9  C1.27.2 : 29,4
ST1.29.1 : 25,7  C1.29.1 : 33,3
STA1.24.2 : 67,1  F1.24.2 : 178
STA2.24.2 : 178  F2.24.2 : 228
STA1.25.3 : 67,1  F1.25.3 : 178
STA2.25.3 : 178  F2.25.3 : 228
STA1.26.4 : 67,1  F1.26.4 : 178
STA2.26.4 : 178  F2.26.4 : 228
```

```
STA1.27.5 : 67,1  F1.27.5 : 178
STA2.27.5 : 178  F2.27.5 : 228
STA1.28.6 : 67,1  F1.28.6 : 178
STA2.28.1 : 178  F2.28.1 : 228
STA1.29.1 : 67,1  F1.29.1 : 178
STA2.29.1 : 228  F2.29.1 : 278
ST1.11.2 : 29,4  C1.11.2 : 41
ST2.11.2 : 41  C2.11.2 : 53,5
ST4.11.1 : 55  C4.11.1 : 76,3
ST1.13.1 : 33,3  C1.13.1 : 42,5
ST2.13.1 : 57,1  C2.13.1 : 84,8
ST1.10.2 : 41  C1.10.2 : 50
ST2.10.1 : 85,3  C2.10.1 :97
ST4.10.2 : 97  C4.10.2 : 107,6
ST1.15.1 : 42,5  C1.15.1 : 52
ST2.15.2 : 53,5  C2.15.2 : 65,2
ST4.15.1 : 76,3  C4.15.1 : 85,6
ST1.12.2 : 50  C1.12.2 : 58,7
ST1.14.1 : 52  C1.14.1 : 59,6
STA1.10.1 : 178  F1.10.1 : 278,6
STA2.10.2 :278,6 F2.10.2 : 329,8
STA1.11.2 : 178  F1.11.2 : 278,6
STA2.11.3 :278,6 F2.11.3 : 329,8
```

```
STA1.12.3 : 178  F1.12.3 : 278,6
STA2.12.4 :278,6 F2.12.4 : 329,8
STA1.13.4 : 178  F1.13.4 : 278,6
STA2.13.5 :278,6 F2.13.5 : 329,8
STA1.14.5 : 178  F1.14.5 : 278,6
STA2.14.1 :278,6 F2.14.1 : 329,8
STA1.15.6 : 178  F1.15.6 : 278,6
STA2.15.1 : 329,8  F2.15.1 : 381
ST1.1.2 : 58,7  C1.1.2 : 73,9
ST2.1.2 : 73,9  C2.1.2 : 88,9
ST4.1.1 : 88,9  C4.1.1 : 107,8
ST1.5.1 : 59,6  C1.5.1 : 74,7
ST2.5.1 : 97,6  C2.5.1 : 126,4
ST1.3.2 : 73,9  C1.3.2 : 83,7
ST2.3.2 : 88,9  C2.3.2 : 100,7
ST4.3.2 : 107,6  C4.3.2 : 118,1
ST1.2.1 : 74,7  C1.2.1 : 84,4
ST2.2.2 : 100,7  C2.2.2 : 106,9
ST4.2.1 : 107,8  C4.2.1 : 119,5
ST1.6.2 : 83,7  C1.6.2 : 91
ST1.4.1 : 84,4  C1.4.1 : 90,2
STA1.1.1 : 278,6  F1.1.1 : 382,2
STA2.1.2 : 382,2  F2.1.2 : 429,9
```

STA1.2.2 : 278,6 F1.2.2 : 382,2
STA2.2.3 : 382,2 F2.2.3 : 429,9
STA1.3.3 : 278,6 F1.3.3 : 382,2
STA2.3.4 : 382,2 F2.3.4 : 429,9
STA1.4.4 : 278,6 F1.4.4 : 382,2
STA2.4.5 : 382,2 F2.4.5 : 429,9
STA1.5.5 : 278,6 F1.5.5 : 382,2
STA2.5.1 : 382,2 F2.5.1 : 429,9
STA1.6.6 : 278,6 F1.6.6 : 382,2
STA2.6.1 : 429,9 F2.6.1 : 477,6
ST1.32.1 : 90,2 C1.32.1 : 102
ST2.32.1 :126,9 C2.32.1 : 134,4
ST4.32.2 :134,4 C4.32.2 : 147,3
ST1.31.2 : 91 C1.31.2 : 99,3
ST2.31.1 :134,4 C2.31.1 : 143,5
ST4.31.1 :143,5 C4.31.1 : 150,8
STA1.30.1 :382,2 F1.30.1 : 421,4
STA2.30.2 :429,9 F2.30.2 : 492,7
STA1.31.2 :382,2 F1.31.2 : 421,4
STA2.31.3 :429,9 F2.31.3 : 492,7
ST1.8.2 : 99,3 C1.8.2 : 109,8
ST2.8.1 : 143,5 C2.8.1 : 158
ST4.8.2 : 158 C4.8.2 : 171,7
ST1.7.1 : 102 C1.7.1 : 114
ST2.7.1 : 158 C2.7.1 : 162,3
ST4.7.1 : 162,3 C4.7.1 : 170
STA1.7.3 : 382,2 F1.7.3 : 433,2
STA2.7.4 : 433,2 F2.7.4 : 491,6
STA1.8.4 : 382,2 F1.8.4 : 433,2
STA2.8.5 : 433,2 F2.8.5 : 491,6
ST1.19.2 :109,8 C1.19.2 : 127,1
ST2.19.2 :127,1 C2.19.2 : 136,1
ST4.19.1 : 170 C4.19.1 : 202,1
ST1.16.1 : 114 C1.16.1 : 134,7
ST2.16.1 :162,3 C2.16.1 : 184,2
ST4.16.2 : 184,2 C4.16.2 : 192,3
ST1.17.2 : 127,1 C1.17.2 : 142,3
ST2.17.1 : 184,2 C2.17.1 : 194,9
ST4.17.2 : 194,9 C4.17.2 : 214,2
ST1.18.1 : 134,7 C1.18.1 : 144,2
ST2.18.2 : 144,2 C2.18.2 : 147,2
ST4.18.1 : 202,1 C4.18.1 : 210,2
STA1.16.5 :382,2 F1.16.5 : 461,2
STA2.16.1 :477,6 F2.16.1 : 549,2
STA1.17.6 :382,2 F1.17.6 : 461,2
STA2.17.4 :491,6 F2.17.4 : 563,2
STA1.18.1 :421,4 F1.18.1 : 500,4
STA2.18.5 : 500,4 F2.18.5 : 572
STA1.19.2 :421,4 F1.19.2 : 500,4
STA2.19.2 :500,4   F2.19.2 : 572
ST1.21.2 : 142,3 C1.21.2 : 154,8
ST2.21.1 : 194,9 C2.21.1 : 204,5
ST4.21.1 : 210,2 C4.21.1 : 233,5
ST1.24.1 : 144,2 C1.24.1 : 153
ST2.24.1 : 204,5 C2.24.1 : 212,5
ST4.24.2 : 214,2 C4.24.2 : 241,5
ST1.20.1 : 153 C1.20.1 : 164,9
ST2.20.1 : 212,5 C2.20.1 : 223,5
ST4.20.1 : 233,5 C4.20.1 : 241,6
ST1.22.2 : 154,8 C1.22.2 : 162,8
ST2.22.2 : 162,8 C2.22.2 : 165,5
ST4.22.2 : 241,5 C4.22.2 : 250,8
ST1.23.2 : 162,8 C1.23.2 : 170,9
ST2.23.2 : 170,9 C2.23.2 : 180,2
STA1.20.3 :433,2 F1.20.3 : 501,8
STA2.20.3 :501,8 F2.20.3 : 545,9
STA1.21.4 :433,2 F1.21.4 : 501,8
STA2.21.3 : 545,9 F2.21.3 : 590
STA1.22.5 :461,2 F1.22.5 : 529,8
STA2.22.1 :549,2 F2.22.1 : 593,3
STA1.23.6 :461,2 F1.23.6 : 529,8
STA2.23.4 :563,2 F2.23.4 : 607,3
ST1.56.1 :164,9 C1.56.1 : 178,4
ST2.56.1 : 224 C2.56.1 : 240
ST1.54.2 : 170,9 C1.54.2 : 180,4
ST2.54.2 : 180,4 C2.54.2 : 188,4
ST4.54.1 : 241,6 C4.54.1 : 253,3
ST1.53.1 : 178,4 C1.53.1 : 187,5
ST2.53.1 : 240 C2.53.1 : 252
ST4.53.2 : 252 C4.53.2 : 259,5
ST1.52.2 : 180,4 C1.52.2 : 189,3
ST2.52.1 : 252 C2.52.1 : 264
ST1.55.1 : 187,5 C1.55.1 : 195,8
ST2.55.2 : 195,8 C2.55.2 : 199,9
ST4.55.1 : 253,3 C4.55.1 : 261,4
ST1.57.2 : 189,3 C1.57.2 : 197,5
ST2.57.2 : 199,9 C2.57.2 : 206,5
STA1.43.1 :500,4 F1.43.1 : 571,6
STA2.43.2 : 572 F2.43.2 : 611,1
STA1.44.2 :500,4 F1.44.2 : 571,6
STA2.44.5 : 572 F2.44.5 : 611,1
STA1.45.3 : 501,8 F1.45.3 : 573
STA2.45.3 : 590 F2.45.3 : 629,1
ST1.65.1 : 195,8 C1.65.1 : 204,1
ST2.65.2 : 206,5 C2.65.2 : 219,6
ST4.65.2 : 259,5 C4.65.2 : 272,8
ST1.64.2 : 197,5 C1.64.2 : 206,1
ST2.64.1 : 264 C2.64.1 : 273,2
ST4.64.1 : 273,2 C4.64.1 : 280,5
ST1.63.1 : 204,1 C1.63.1 : 215,3
ST2.63.2 : 219,6 C2.63.2 : 231
ST1.61.2 : 206,1 C1.61.2 : 214,5
ST2.61.1 : 273,7 C2.61.1 : 278,5
ST4.61.2 : 278,5 C4.61.2 : 285
ST1.62.2 : 214,5 C1.62.2 : 222
ST2.62.2 : 231 C2.62.2 : 235,7
ST4.62.1 : 280,5 C4.62.1 : 287,4
ST1.66.1 : 215,3 C1.66.1 : 223,7
STA1.49.4 :501,8 F1.49.4 : 571,4
STA2.49.1 :593,3 F2.49.1 : 667,8
STA1.50.5 :529,8 F1.50.5 : 599,4
STA2.50.4 :607,3 F2.50.4 : 681,8
ST1.58.2 : 222 C1.58.2 : 244,2
ST2.58.1 : 278,5 C2.58.1 : 293,7
ST4.58.2 : 293,7 C4.58.2 : 303
ST1.59.1 : 223,7 C1.59.1 : 235,5
ST2.59.1 : 293,7 C2.59.1 : 301,7
ST4.59.1 : 301,7 C4.59.1 : 314,6
ST1.60.1 : 235,5 C1.60.1 : 245,2
ST2.60.2 : 245,2 C2.60.2 : 253,8
ST4.60.2 : 303 C4.60.2 : 311,7
STA1.46.6 :529,8 F1.46.6 : 559,4
STA2.46.2 :611,1 F2.46.2 : 659,2
STA1.47.6 :559,4 F1.47.6 :588,9
STA2.47.5 :611,1 F2.47.5 : 659,2
STA1.48.4 : 571,4 F1.48.4 : 601
STA2.48.3 :629,1 F2.48.3 : 677,2
ST1.34.2 : 244,2 C1.34.2 : 268,8
ST2.34.2 : 268,8 C2.34.2 : 285
ST4.34.2 : 311,7 C4.34.2 : 331,8
ST1.38.1 : 245,2 C1.38.1 : 256,4
ST2.38.2 : 285 C2.38.2 : 298,9
ST4.38.1 : 314,6 C4.38.1 : 334,7
ST1.33.1 : 256,4 C1.33.1 : 270,9
ST2.33.1 : 302,2 C2.33.1 : 319
ST4.33.2 : 331,8 C4.33.2 : 344,1
ST1.35.2 : 268,8 C1.35.2 : 281,7
ST2.35.2 : 298,9 C2.35.2 : 307
ST4.35.1 : 334,7 C4.35.1 : 348,8
ST1.36.1 : 270,9 C1.36.1 : 281,2
ST2.36.2 : 307 C2.36.2 : 313,4
ST4.36.2 : 344,1 C4.36.2 : 355,8
ST1.37.1 : 281,2 C1.37.1 : 292
ST2.37.2 : 313,4 C2.37.2 : 325,4
STA1.32.1 :571,6 F1.32.1 : 632
STA2.32.2 :659,2 F2.32.2 : 757,4
STA1.33.2 :571,6 F1.33.2 : 632
STA2.33.5 :659,2 F2.33.5 : 757,4
STA1.34.3 : 573 F1.34.3 : 633,4
STA2.34.1 : 667,8 F2.34.1 : 766
ST1.40.2 : 281,7 C1.40.2 : 291,5
ST3.40.1 : 291,5 C3.40.1 : 342,7
ST4.40.1 : 348,8 C4.40.1 : 360,5
ST1.39.2 : 291,5 C1.39.2 : 306,6
ST2.39.1 : 319,5 C2.39.1 : 331
ST4.39.2 : 355,8 C4.39.2 : 381,9
ST1.41.1 : 292 C1.41.1 : 304
ST2.41.1 : 331 C2.41.1 : 337,4
ST4.41.1 : 360,5 C4.41.1 : 374,4
ST1.42.1 : 304 C1.42.1 : 313,5
ST2.42.2 : 325,4 C2.42.2 : 332,3
ST4.42.1 : 374,4 C4.42.1 : 383,4
STA1.35.6 : 589 F1.35.6 : 634,5
STA2.35.3 :677,2 F2.35.3 : 743,1
STA1.36.5 :599,4 F1.36.5 : 644,9
STA2.36.4 :681,8 F2.36.4 : 747,7
ST1.44.2 :306,6 C1.44.2 : 338,8
ST3.44.1 : 342,7 C3.44.1 : 493,6
ST1.43.1 : 313,5 C1.43.1 : 345,4
ST3.43.1 : 493,6 C3.43.1 : 616,3
ST1.47.2 : 338,8 C1.47.2 : 356,3
ST2.47.2 : 356,3 C2.47.2 : 372,3
ST1.45.1 : 345,4 C1.45.1 : 356,9
ST2.45.2 : 372,3 C2.45.2 : 380,3
ST4.45.2 : 381,9 C4.45.2 : 393,6
ST1.46.2 : 356,3 C1.46.2 : 366,5
ST2.46.2 : 380,3 C2.46.2 : 389,2
ST4.46.1 : 389,2 C4.46.1 : 399,7
STA1.37.4 :616,3 F1.37.4 : 715,2
STA2.37.3 :743,1 F2.37.3 : 804,1
STA1.38.1 : 632 F1.38.1 : 730,9
STA2.38.4 :747,7 F2.38.4 : 808,7
STA1.39.2 : 632 F1.39.2 : 730,9
STA2.39.2 :757,4 F2.39.2 : 818,4
ST1.49.1 : 356,9 C1.49.1 : 364
ST3.49.1 : 616,3 C3.49.1 : 709,9
ST1.48.1 : 364 C1.48.1 : 370,9
ST3.48.1 : 709,9 C3.48.1 : 787,6
ST1.50.2 : 366,5 C1.50.2 : 376,9
ST2.50.2 : 389,2 C2.50.2 : 394,7
ST4.50.2 : 394,7 C4.50.2 : 407
ST1.51.1 : 370,9 C1.51.1 : 378,5
ST2.51.2 : 394,7 C2.51.2 : 397,8
ST4.51.1 : 399,7 C4.51.1 : 407,8
STA1.40.3 :787,6 F1.40.3 : 812,6
STA2.40.5 :812,6 F2.40.5 : 842,8
STA1.41.6 :787,6 F1.41.6 : 812,6
STA2.41.1 :812,6 F2.41.1 : 842,8
STA1.42.5 :787,6 F1.42.5 : 812,6
STA2.42.3 :812,6 F2.42.3 : 842,8

FFS3J5 En Az Boşluk Kuralı Sonuçları

ST1.3.1 : 0 C1.3.1 : 20
ST2.3.1 : 20 C2.3.1 : 33
ST1.2.2 : 0 C1.2.2 : 8
ST2.2.2 : 8 C2.2.2 : 24
ST1.1.3 : 0 C1.1.3 : 13
STA1.1.1 : 33 F1.1.1 : 75
ST1.5.2 : 16 C1.5.2 : 28
ST2.5.3 : 28 C2.5.3 : 34
ST2.4.4 : 0 C2.4.4 : 5
STA1.3.2 : 34 F1.3.2 : 62
STA1.4.3 : 34 F1.4.3 : 62
STA1.2.4 : 33 F1.2.4 : 75

FFS3J15 En Az Boşluk Kuralı Sonuçları

ST1.7.1 : 0 C1.7.1 : 8
ST2.7.3 : 8 C2.7.3 : 26
ST1.6.2 : 0 C1.6.2 : 17
ST2.6.4 : 17 C2.6.4 : 24
ST1.8.3 : 0 C1.8.3 : 10
STA1.5.1 : 26 F1.5.1 : 64
ST1.3.4 : 0 C1.3.4 : 18
ST2.3.2 : 18 C2.3.2 : 35
ST1.1.5 : 0 C1.1.5 : 11
ST2.1.2 : 35 C2.1.2 : 46
ST1.2.5 : 13 C1.2.5 : 21
STA1.1.2 : 46 F1.1.2 : 67
ST1.14.1 : 16 C1.14.1 : 31
ST2.14.3 : 31 C2.14.3 : 48
ST1.13.3 : 18 C1.13.3 : 29
ST2.13.1 : 29 C2.13.1 : 39
ST1.15.2 : 25 C1.15.2 : 43
ST2.12.3 : 48 C2.12.3 : 65
STA1.9.3 : 65 F1.9.3 : 105
STA1.2.4 : 46 F1.2.4 : 67
ST1.10.4 : 26 C1.10.4 : 40
ST2.10.2 : 46 C2.10.2 : 54
ST2.11.1 : 39 C2.11.1 : 58
ST1.9.5 : 29 C1.9.5 : 42
STA1.7.5 : 58 F1.7.5 : 79
STA1.10.1 : 65 F1.10.1 : 105
STA1.6.2 : 67 F1.6.2 : 105
ST1.4.3 : 37 C1.4.3 : 55
ST2.4.3 : 65 C2.4.3 : 75
ST2.5.4 : 24 C2.5.4 : 36
STA1.3.4 : 75 F1.3.4 : 116
STA1.4.5 : 79 F1.4.5 : 120
STA1.8.1 : 105 F1.8.1 : 126

FFS3J30 En Az Boşluk Kuralı Sonuçları

ST1.11.1 : 0 C1.11.1 : 15
ST2.11.2 : 15 C2.11.2 : 33
ST1.9.2 : 0 C1.9.2 : 14
ST2.9.1 : 14 C2.9.1 : 32
ST1.10.3 : 0 C1.10.3 : 15
STA1.8.1 : 33 F1.8.1 : 80
ST1.15.2 : 22 C1.15.2 : 37
ST2.15.3 : 37 C2.15.3 : 57
ST1.16.1 : 23 C1.16.1 : 41
ST2.16.3 : 57 C2.16.3 : 69
ST1.17.3 : 23 C1.17.3 : 33
ST2.17.1 : 33 C2.17.1 : 52

STA1.12.2 : 69 F1.12.2 : 90
STA1.11.3 : 69 F1.11.3 : 90
ST1.6.3 : 41 C1.6.3 : 59
ST2.6.2 : 59 C2.6.2 : 70
ST1.8.2 : 45 C1.8.2 : 61
ST2.8.1 : 61 C2.8.1 : 66
ST2.7.3 : 69 C2.7.3 : 87
STA1.5.4 : 87 F1.5.4 : 115
ST1.12.1 : 49 C1.12.1 : 63
ST2.12.4 : 63 C2.12.4 : 77
ST1.14.1 : 65 C1.14.1 : 79
ST2.14.2 : 79 C2.14.2 : 91
ST1.13.3 : 67 C1.13.3 : 81
ST2.13.3 : 87 C2.13.3 : 94
STA1.9.5 : 94 F1.9.5 : 115
ST1.22.2 : 69 C1.22.2 : 84
ST2.22.4 : 84 C2.22.4 : 103
ST1.23.2 : 86 C1.23.2 : 96
ST2.21.2 : 91 C2.21.2 : 100
STA1.15.6 : 103 F1.15.6 : 152
ST1.4.1 : 87 C1.4.1 : 105
ST2.4.2 : 105 C2.4.2 : 122
ST1.5.3 : 89 C1.5.3 : 101
STA1.4.1 : 122 F1.4.1 : 157
STA1.7.2 : 90 F1.7.2 : 137
ST1.29.2 : 104 C1.29.2 : 113
ST2.29.4 : 113 C2.29.4 : 124
ST1.27.3 : 109 C1.27.3 : 117
ST2.27.4 : 124 C2.27.4 : 129
ST2.28.1 : 66 C2.28.1 : 72
ST2.30.2 : 122 C2.30.2 : 128
STA1.19.3 : 129 F1.19.3 : 164
ST1.2.1 : 113 C1.2.1 : 130
ST2.2.2 : 130 C2.2.2 : 138
ST2.1.3 : 94 C2.1.3 : 113
ST1.3.2 : 121 C1.3.2 : 126
ST2.3.3 : 126 C2.3.3 : 139
STA1.1.4 : 139 F1.1.4 : 167
ST1.20.3 : 125 C1.20.3 : 144
ST2.20.1 : 144 C2.20.1 : 163
ST1.19.2 : 134 C1.19.2 : 143
ST1.18.1 : 138 C1.18.1 : 146
STA1.14.5 : 163 F1.14.5 : 210
STA1.13.2 : 163 F1.13.2 : 210
ST1.26.2 : 151 C1.26.2 : 168
ST2.24.4 : 129 C2.24.4 : 136
ST2.25.2 : 138 C2.25.2 : 144
STA1.17.6 : 168 F1.17.6 : 210
STA1.3.1 : 157 F1.3.1 : 192
STA1.16.3 : 164 F1.16.3 : 213
STA1.6.4 : 167 F1.6.4 : 195
STA1.10.1 : 192 F1.10.1 : 213
STA1.2.4 : 195 F1.2.4 : 223
STA1.20.2 : 210 F1.20.2 : 245
STA1.18.5 : 210 F1.18.5 : 252

FFS6J5 En Az Boşluk Kuralı Sonuçları

ST1.1.1 : 0 C1.1.1 : 9
ST2.1.3 : 9 C2.1.3 : 28
ST3.1.1 : 28 C3.1.1 : 38
ST4.1.1 : 38 C4.1.1 : 50
ST1.2.2 : 0 C1.2.2 : 11
ST3.2.2 : 11 C3.2.2 : 26
ST4.2.2 : 26 C4.2.2 : 44
ST1.3.3 : 0 C1.3.3 : 8
ST2.3.3 : 28 C2.3.3 : 44
ST4.3.3 : 44 C4.3.3 : 56
STA1.1.1 : 56 F1.1.1 : 105
STA2.1.1 : 105 F2.1.1 : 145
ST1.4.3 : 8 C1.4.3 : 24
ST2.4.1 : 24 C2.4.1 : 44
ST3.4.2 : 44 C3.4.2 : 59
ST4.4.2 : 59 C4.4.2 : 71
ST1.5.1 : 17 C1.5.1 : 25
ST2.5.2 : 25 C2.5.2 : 41
ST3.5.1 : 41 C3.5.1 : 51
ST4.5.1 : 51 C4.5.1 : 64
STA1.3.2 : 71 F1.3.2 : 115
STA2.3.2 : 115 F2.3.2 : 144
STA1.4.3 : 71 F1.4.3 : 115
STA2.4.3 : 115 F2.4.3 : 144
STA1.2.4 : 56 F1.2.4 : 105
STA2.2.4 : 105 F2.2.4 : 145

FFS6J15 En Az Boşluk Kuralı Sonuçları

ST1.6.1 : 0 C1.6.1 : 8
ST2.6.1 : 8 C2.6.1 : 27
ST3.6.1 : 27 C3.6.1 : 36
ST4.6.1 : 36 C4.6.1 : 47
ST1.7.2 : 0 C1.7.2 : 20
ST2.7.5 : 20 C2.7.5 : 29
ST3.7.2 : 29 C3.7.2 : 35
ST4.7.2 : 35 C4.7.2 : 47
ST1.8.3 : 0 C1.8.3 : 11
ST3.8.3 : 11 C3.8.3 : 27
ST4.8.3 : 27 C4.8.3 : 37
STA1.6.1 : 47 F1.6.1 : 71
STA2.6.1 : 71 F2.6.1 : 111
ST1.3.4 : 0 C1.3.4 : 9
ST2.3.6 : 9 C2.3.6 : 20
ST3.3.4 : 20 C3.3.4 : 29
ST4.3.3 : 37 C4.3.3 : 51
ST1.2.4 : 11 C1.2.4 : 22
ST2.2.4 : 22 C2.2.4 : 29
ST4.2.1 : 47 C4.2.1 : 66
ST1.1.1 : 16 C1.1.1 : 28
ST2.1.5 : 29 C2.1.5 : 38
ST4.1.2 : 47 C4.1.2 : 60
STA1.2.2 : 66 F1.2.2 : 99
STA2.2.2 : 99 F2.2.2 : 148
ST1.14.3 : 19 C1.14.3 : 32
ST2.14.4 : 32 C2.14.4 : 41
ST3.14.5 : 41 C3.14.5 : 58
ST4.14.3 : 58 C4.14.3 : 72
ST1.15.2 : 28 C1.15.2 : 47
ST2.15.6 : 47 C2.15.6 : 65
ST4.15.2 : 65 C4.15.2 : 76
ST2.12.3 : 0 C2.12.3 : 14
ST3.12.6 : 14 C3.12.6 : 25
ST4.12.1 : 66 C4.12.1 : 84
ST1.13.4 : 30 C1.13.4 : 46
ST2.13.5 : 46 C2.13.5 : 56
ST3.13.6 : 56 C3.13.6 : 63
ST4.13.3 : 72 C4.13.3 : 82
STA1.9.3 : 84 F1.9.3 : 120
STA2.9.3 : 120 F2.9.3 : 162
ST2.11.2 : 0 C2.11.2 : 16
ST3.11.3 : 27 C3.11.3 : 43
ST4.11.2 : 76 C4.11.2 : 96
ST1.9.1 : 36 C1.9.1 : 54
ST2.9.3 : 54 C2.9.3 : 60
ST3.9.4 : 60 C3.9.4 : 78
ST4.9.3 : 82 C4.9.3 : 91
ST1.10.3 : 40 C1.10.3 : 48
ST2.10.4 : 48 C2.10.4 : 64
ST3.10.2 : 64 C3.10.2 : 71
ST4.10.1 : 84 C4.10.1 : 89
STA1.7.4 : 96 F1.7.4 : 117
STA2.7.4 : 117 F2.7.4 : 151
STA1.5.5 : 47 F1.5.5 : 71
STA2.5.5 : 71 F2.5.5 : 111
STA1.1.6 : 66 F1.1.6 : 99
STA2.1.6 : 99 F2.1.6 : 148
STA1.10.1 : 84 F1.10.1 : 120
STA2.10.1 : 120 F2.10.1 : 162
STA1.8.5 : 96 F1.8.5 : 117
STA2.8.5 : 117 F2.8.5 : 151
ST1.4.4 : 54 C1.4.4 : 70
ST2.4.5 : 70 C2.4.5 : 89
ST4.4.1 : 89 C4.4.1 : 105
ST3.5.1 : 36 C3.5.1 : 46
ST4.5.3 : 91 C4.5.3 : 99
STA1.3.2 : 105 F1.3.2 : 155
STA2.3.2 : 155 F2.3.2 : 184
STA1.4.6 : 105 F1.4.6 : 155
STA2.4.6 : 155 F2.4.6 : 184

FFS6J30 En Az Boşluk Kuralı Sonuçları

ST1.10.1 : 0 C1.10.1 : 20
ST2.10.3 : 20 C2.10.3 : 35
ST3.10.1 : 35 C3.10.1 : 45
ST4.10.1 : 45 C4.10.1 : 59
ST1.9.2 : 0 C1.9.2 : 19
ST2.9.3 : 35 C2.9.3 : 55
ST4.9.2 : 55 C4.9.2 : 74
ST1.11.3 : 0 C1.11.3 : 12
ST2.11.3 : 55 C2.11.3 : 74
ST3.11.2 : 74 C3.11.2 : 81
ST4.11.3 : 81 C4.11.3 : 101
STA1.7.1 : 101 F1.7.1 : 134
STA2.7.1 : 134 F2.7.1 : 172
ST1.29.3 : 20 C1.29.3 : 27
ST2.29.3 : 74 C2.29.3 : 85
ST3.29.3 : 85 C3.29.3 : 99
ST4.29.4 : 99 C4.29.4 : 112
ST1.27.2 : 27 C1.27.2 : 43
ST2.27.1 : 43 C2.27.1 : 52
ST4.27.1 : 59 C4.27.1 : 78
ST2.28.2 : 0 C2.28.2 : 5
ST3.28.4 : 5 C3.28.4 : 19
ST4.28.2 : 74 C4.28.2 : 81
ST2.30.3 : 85 C2.30.3 : 95
ST3.30.4 : 95 C3.30.4 : 101
ST4.30.1 : 101 C4.30.1 : 111
STA1.19.2 : 112 F1.19.2 : 142
STA2.19.2 : 142 F2.19.2 : 171
ST1.17.1 : 28 C1.17.1 : 36
ST2.17.2 : 36 C2.17.2 : 48
ST3.17.1 : 48 C3.17.1 : 60
ST4.17.2 : 81 C4.17.2 : 94
ST1.15.3 : 35 C1.15.3 : 42
ST2.15.2 : 48 C2.15.2 : 64
ST3.15.1 : 64 C3.15.1 : 71
ST4.15.2 : 94 C4.15.2 : 105
ST1.16.1 : 38 C1.16.1 : 45
ST3.16.1 : 71 C3.16.1 : 79
ST4.16.3 : 101 C4.16.3 : 107
STA1.11.3 : 107 F1.11.3 : 156
STA2.11.3 : 156 F2.11.3 : 193
STA1.12.4 : 107 F1.12.4 : 156
STA2.12.4 : 156 F2.12.4 : 193
ST1.19.3 : 50 C1.19.3 : 69
ST2.19.2 : 69 C2.19.2 : 87
ST3.19.1 : 87 C3.19.1 : 107
ST4.19.2 : 107 C4.19.2 : 119
ST1.20.2 : 51 C1.20.2 : 67
ST2.20.3 : 95 C2.20.3 : 107
ST3.20.2 : 107 C3.20.2 : 112
ST4.20.3 : 112 C4.20.3 : 119
ST1.18.1 : 53 C1.18.1 : 68
ST3.18.3 : 99 C3.18.3 : 106
ST4.18.1 : 111 C4.18.1 : 127
STA1.13.5 : 127 F1.13.5 : 152
STA2.13.5 : 152 F2.13.5 : 173
STA1.20.6 : 112 F1.20.6 : 142
STA2.20.6 : 142 F2.20.6 : 171
ST1.14.2 : 75 C1.14.2 : 93
ST2.14.1 : 93 C2.14.1 : 107
ST3.14.4 : 107 C3.14.4 : 117
ST1.13.1 : 76 C1.13.1 : 85
ST2.13.1 : 107 C2.13.1 : 120
ST4.13.4 : 120 C4.13.4 : 133
ST1.12.3 : 77 C1.12.3 : 83
ST2.12.2 : 87 C2.12.2 : 101
ST3.12.3 : 106 C3.12.3 : 114
ST4.12.2 : 119 C4.12.2 : 124
STA1.9.1 : 134 F1.9.1 : 157
STA2.9.2 : 171 F2.9.2 : 219
ST1.8.3 : 91 C1.8.3 : 108
ST2.8.1 : 120 C2.8.1 : 131
ST3.8.1 : 131 C3.8.1 : 137
ST4.8.3 : 137 C4.8.3 : 145
ST1.6.1 : 93 C1.6.1 : 110
ST2.6.3 : 110 C2.6.3 : 125
ST4.6.2 : 125 C4.6.2 : 132
ST2.7.1 : 131 C2.7.1 : 140
ST3.7.2 : 140 C3.7.2 : 152
ST4.7.1 : 152 C4.7.1 : 158
STA1.5.2 : 158 F1.5.2 : 199
STA2.5.6 : 199 F2.5.6 : 230
ST2.25.1 : 140 C2.25.1 : 153
ST3.25.3 : 153 C3.25.3 : 171
ST4.25.2 : 171 C4.25.2 : 190
ST2.24.1 : 153 C2.24.1 : 172
ST3.24.4 : 172 C3.24.4 : 181
ST4.24.4 : 181 C4.24.4 : 193
ST1.26.2 : 101 C1.26.2 : 110
ST2.26.2 : 110 C2.26.2 : 120
ST4.26.3 : 145 C4.26.3 : 159
STA1.17.6 : 193 F1.17.6 : 224
STA2.17.1 : 224 F2.17.1 : 245
ST1.5.3 : 116 C1.5.3 : 124
ST2.5.3 : 125 C2.5.3 : 143
ST3.5.1 : 143 C3.5.1 : 148
ST4.5.1 : 158 C4.5.1 : 172
ST1.4.1 : 118 C1.4.1 : 130
ST2.4.3 : 143 C2.4.3 : 150
ST3.4.1 : 150 C3.4.1 : 166
ST4.4.3 : 166 C4.4.3 : 171
STA1.3.5 : 172 F1.3.5 : 194

STA2.3.5 : 194 F2.3.5 : 227
STA1.8.3 : 156 F1.8.3 : 189
STA2.8.3 : 193 F2.8.3 : 231
ST1.23.2 : 118 C1.23.2 : 137
ST2.23.2 : 137 C2.23.2 : 147
ST3.23.2 : 152 C3.23.2 : 166
ST4.23.3 : 171 C4.23.3 : 184
ST1.22.3 : 132 C1.22.3 : 144
ST2.22.2 : 147 C2.22.2 : 165
ST3.22.1 : 166 C3.22.1 : 180
ST4.22.1 : 180 C4.22.1 : 187
ST2.21.1 : 172 C2.21.1 : 183
ST3.21.2 : 183 C3.21.2 : 193
ST4.21.3 : 193 C4.21.3 : 201

STA1.15.4 : 201 F1.15.4 : 235
STA2.15.4 : 235 F2.15.4 : 283
STA1.16.1 : 201 F1.16.1 : 235
STA2.16.2 : 235 F2.16.2 : 283
ST1.2.1 : 138 C1.2.1 : 147
ST2.2.2 : 165 C2.2.2 : 180
ST3.2.3 : 180 C3.2.3 : 198
ST4.2.1 : 198 C4.2.1 : 212
ST1.3.2 : 145 C1.3.2 : 153
ST2.3.3 : 153 C2.3.3 : 167
ST3.3.1 : 180 C3.3.1 : 193
ST4.3.2 : 193 C4.3.2 : 203
ST2.1.2 : 180 C2.1.2 : 200
ST4.1.4 : 200 C4.1.4 : 214

STA1.1.3 : 214 F1.1.3 : 249
STA2.1.5 : 249 F2.1.5 : 279
STA1.2.5 : 214 F1.2.5 : 249
STA2.2.6 : 249 F2.2.6 : 279
STA1.14.2 : 199 F1.14.2 : 224
STA2.14.3 : 231 F2.14.3 : 252
STA1.10.2 : 224 F1.10.2 : 247
STA2.10.1 : 247 F2.10.1 : 295
STA1.6.6 : 224 F1.6.6 : 265
STA2.6.3 : 265 F2.6.3 : 296
STA1.18.1 : 235 F1.18.1 : 266
STA2.18.5 : 279 F2.18.5 : 300
STA1.4.4 : 235 F1.4.4 : 257
STA2.4.6 : 279 F2.4.6 : 312

FFS6J66 En Az Boşluk Kuralı Sonuçları

ST1.34.1 : 0 C1.34.1 : 24,6
ST2.34.2 : 24,6 C2.34.2 : 40,8
ST4.34.1 : 40,8 C4.34.1 : 60,9
ST1.38.2 : 0 C1.38.2 : 11,2
ST2.38.2 : 40,8 C2.38.2 : 54,7
ST4.38.2 : 54,7 C4.38.2 : 74,8
ST1.33.2 : 11,2 C1.33.2 : 25,7
ST2.33.1 : 25,7 C2.33.1 : 42,5
ST4.33.1 : 60,9 C4.33.1 : 73,2
ST1.35.1 : 24,6 C1.35.1 : 37,5
ST2.35.2 : 54,7 C2.35.2 : 62,8
ST4.35.1 : 73,2 C4.35.1 : 87,3
ST1.36.2 : 25,7 C1.36.2 : 36
ST2.36.2 : 62,8 C2.36.2 : 69,2
ST4.36.2 : 74,8 C4.36.2 : 86,5
ST1.37.2 : 36 C1.37.2 : 46,8
ST2.37.2 : 69,2 C2.37.2 : 81,2
STA1.32.1 : 87,3 F1.32.1 : 147,7
STA2.32.1 : 147,7 F2.32.1 : 245,9
ST1.1.1 : 37,5 C1.1.1 : 52,7
ST2.1.2 : 81,2 C2.1.2 : 96,2
ST4.1.2 : 96,2 C4.1.2 : 115,1
ST1.5.2 : 46,8 C1.5.2 : 61,9
ST2.5.1 : 61,9 C2.5.1 : 90,7
ST1.3.1 : 52,7 C1.3.1 : 62,5
ST2.3.2 : 96,2 C2.3.2 : 108
ST4.3.1 : 108 C4.3.1 : 118,5
ST1.2.2 : 61,9 C1.2.2 : 71,6
ST2.2.2 : 108 C2.2.2 : 114,2
ST4.2.2 : 115,1 C4.2.2 : 126,8
ST1.6.1 : 62,5 C1.6.1 : 69,8
ST1.4.1 : 69,8 C1.4.1 : 75,6
STA1.1.2 : 126,8 F1.1.2 : 230,4
STA2.1.2 : 230,4 F2.1.2 : 278,1
ST1.9.2 : 71,6 C1.9.2 : 79,3
ST2.9.1 : 91,2 C2.9.1 : 97,3
STA1.9.3 : 97,3 F1.9.3 : 134,5
STA2.9.3 : 134,5 F2.9.3 : 193,1
STA1.2.4 : 126,8 F1.2.4 : 230,4
STA2.2.4 : 230,4 F2.2.4 : 278,1
ST1.40.1 : 75,6 C1.40.1 : 85,4
ST3.40.1 : 85,4 C3.40.1 : 136,6
ST4.40.1 : 136,6 C4.40.1 : 148,3
ST1.39.2 : 79,3 C1.39.2 : 94,4
ST2.39.1 : 97,3 C2.39.1 : 108,8
ST4.39.2 : 126,8 C4.39.2 : 152,9
ST1.41.1 : 85,4 C1.41.1 : 97,4
ST2.41.1 : 108,8 C2.41.1 : 115,2
ST4.41.1 : 148,3 C4.41.1 : 162,2
ST1.42.2 : 94,4 C1.42.2 : 103,9
ST2.42.2 : 114,2 C2.42.2 : 121,1
ST4.42.2 : 152,9 C4.42.2 : 161,9
STA1.35.5 : 162,2 F1.35.5 : 207,7
STA2.35.5 : 207,7 F2.35.5 : 273,6
STA1.36.6 : 162,2 F1.36.6 : 207,7
STA2.36.3 : 207,7 F2.36.3 : 273,6
ST1.65.1 : 97,4 C1.65.1 : 105,7
ST2.65.2 : 121,1 C2.65.2 : 134,2
ST4.65.2 : 161,9 C4.65.2 : 175,2
ST1.64.2 : 103,9 C1.64.2 : 112,5
ST2.64.1 : 115,7 C2.64.1 : 124,9
ST4.64.1 : 162,2 C4.64.1 : 169,5
ST1.63.1 : 105,7 C1.63.1 : 116,9
ST2.63.2 : 134,2 C2.63.2 : 145,6
ST1.61.2 : 112,5 C1.61.2 : 120,9
ST2.61.1 : 125,4 C2.61.1 : 130,2
ST4.61.1 : 169,5 C4.61.1 : 176
ST1.62.1 : 116,9 C1.62.1 : 124,4
ST2.62.2 : 145,6 C2.62.2 : 150,3
ST4.62.2 : 175,2 C4.62.2 : 182,1
ST1.66.2 : 120,9 C1.66.2 : 129,3
STA1.49.3 : 182,1 F1.49.3 : 251,7
STA2.49.1 : 251,7 F2.49.1 : 326,2
ST1.58.1 : 124,4 C1.58.1 : 146,6
ST2.58.1 : 146,6 C2.58.1 : 161,8
ST4.58.1 : 176 C4.58.1 : 185,3
ST1.59.2 : 129,3 C1.59.2 : 141,1
ST2.59.1 : 161,8 C2.59.1 : 169,8
ST4.59.2 : 182,1 C4.59.2 : 195
ST1.60.2 : 141,1 C1.60.2 : 150,8
ST2.60.2 : 150,8 C2.60.2 : 159,4
ST4.60.1 : 185,3 C4.60.1 : 194
STA1.46.1 : 195 F1.46.1 : 224,6
STA2.46.3 : 273,6 F2.46.3 : 321,7
ST1.44.1 : 146,6 C1.44.1 : 178,8
ST3.44.1 : 178,8 C3.44.1 : 329,7
ST1.43.2 : 150,8 C1.43.2 : 182,7
ST3.43.1 : 329,7 C3.43.1 : 452,4

ST1.47.1 : 178,8 C1.47.1 : 196,3
ST2.47.2 : 196,3 C2.47.2 : 212,3
ST1.45.2 : 182,7 C1.45.2 : 194,2
ST2.45.2 : 212,3 C2.45.2 : 220,3
ST4.45.1 : 220,3 C4.45.1 : 232
ST1.46.2 : 194,2 C1.46.2 : 204,4
ST2.46.2 : 220,3 C2.46.2 : 229,2
ST4.46.2 : 229,2 C4.46.2 : 239,7
STA1.37.5 : 452,4 F1.37.5 : 551,3
STA2.37.5 : 551,3 F2.37.5 : 612,3
STA1.38.6 : 452,4 F1.38.6 : 551,3
STA2.38.2 : 551,3 F2.38.2 : 612,3
STA1.39.1 : 452,4 F1.39.1 : 551,3
STA2.39.4 : 551,3 F2.39.4 : 612,3
STA1.33.2 : 230,4 F1.33.2 : 290,8
STA2.33.3 : 321,7 F2.33.3 : 419,9
STA1.34.4 : 230,4 F1.34.4 : 290,8
STA2.34.1 : 326,2 F2.34.1 : 424,4
ST1.19.1 : 196,3 C1.19.1 : 213,6
ST2.19.2 : 229,2 C2.19.2 : 238,2
ST4.19.1 : 238,2 C4.19.1 : 270,3
ST1.16.2 : 204,4 C1.16.2 : 225,1
ST2.16.1 : 225,1 C2.16.1 : 247
ST4.16.2 : 247 C4.16.2 : 255,1
ST1.17.1 : 213,6 C1.17.1 : 228,8
ST2.17.1 : 247 C2.17.1 : 257,7
ST4.17.2 : 257,7 C4.17.2 : 277
ST1.18.2 : 225,1 C1.18.2 : 234,6
ST2.18.2 : 238,2 C2.18.2 : 241,2
ST4.18.1 : 270,3 C4.18.1 : 278,4
STA1.16.3 : 278,4 F1.16.3 : 357,4
STA2.16.3 : 419,9 F2.16.3 : 491,5
STA1.17.2 : 290,8 F1.17.2 : 369,8
STA2.17.1 : 424,4 F2.17.1 : 496
STA1.18.4 : 290,8 F1.18.4 : 369,8
STA2.18.3 : 491,5 F2.18.3 : 563,1
STA1.19.3 : 357,4 F1.19.3 : 436,4
STA2.19.1 : 496 F2.19.1 : 567,6
ST1.26.1 : 228,8 C1.26.1 : 240,7
ST2.26.2 : 241,2 C2.26.2 : 253,7
ST4.26.2 : 277 C4.26.2 : 298,3
ST1.28.2 : 234,6 C1.28.2 : 243,5
ST2.28.1 : 258,2 C2.28.1 : 286
ST1.25.1 : 240,7 C1.25.1 : 249,7
ST2.25.1 : 286,5 C2.25.1 : 298,2
ST4.25.1 : 298,2 C4.25.1 : 308,7
ST1.30.2 : 243,5 C1.30.2 : 252,6
ST2.30.2 : 253,7 C2.30.2 : 265,2
ST4.30.2 : 298,3 C4.30.2 : 307,6
ST1.27.1 : 249,7 C1.27.1 : 258,2
ST1.29.2 : 252,6 C1.29.2 : 260,2
STA1.24.2 : 369,8 F1.24.2 : 480,7
STA2.24.3 : 563,1 F2.24.3 : 613,1
STA1.25.4 : 369,8 F1.25.4 : 480,7
STA2.25.1 : 567,6 F2.25.1 : 617,6
STA1.26.3 : 436,4 F1.26.3 : 547,3
STA2.26.2 : 612,3 F2.26.2 : 662,3
STA1.27.2 : 480,7 F1.27.2 : 591,6
STA2.27.4 : 612,3 F2.27.4 : 662,3
STA1.28.4 : 480,7 F1.28.4 : 591,6
STA2.28.5 : 612,3 F2.28.5 : 662,3
STA1.29.3 : 547,3 F1.29.3 : 658,2
STA2.29.3 : 658,2 F2.29.3 : 708,2
STA1.3.1 : 551,3 F1.3.1 : 654,9
STA2.3.1 : 654,9 F2.3.1 : 702,6
STA1.4.5 : 551,3 F1.4.5 : 654,9
STA2.4.2 : 662,3 F2.4.2 : 710
STA1.5.6 : 551,3 F1.5.6 : 654,9
STA2.5.4 : 662,3 F2.5.4 : 710
STA1.6.2 : 591,6 F1.6.2 : 695,2
STA2.6.5 : 695,2 F2.6.5 : 742,9
ST1.11.1 : 258,2 C1.11.1 : 269,8
ST2.11.2 : 269,8 C2.11.2 : 282,3
ST4.11.2 : 307,6 C4.11.2 : 328,9
ST1.13.2 : 260,2 C1.13.2 : 269,4
ST2.13.1 : 298,7 C2.13.1 : 326,5
ST1.10.2 : 269,4 C1.10.2 : 278,4
ST2.10.1 : 327 C2.10.1 : 338,7
ST4.10.1 : 338,7 C4.10.1 : 349,2
ST1.15.1 : 269,8 C1.15.1 : 279,3
ST2.15.2 : 282,3 C2.15.2 : 294
ST4.15.2 : 328,9 C4.15.2 : 338,2
ST1.12.2 : 278,4 C1.12.2 : 287,1
ST1.14.1 : 279,3 C1.14.1 : 286,9
STA1.10.4 : 591,6 F1.10.4 : 692,2
STA2.10.1 : 702,6 F2.10.1 : 753,8
STA1.11.1 : 654,9 F1.11.1 : 755,5
STA2.11.3 : 755,5 F2.11.3 : 806,7

STA1.12.5 : 654,9 F1.12.5 : 755,5
STA2.12.2 : 755,5 F2.12.2 : 806,7
STA1.13.6 : 654,9 F1.13.6 : 755,5
STA2.13.4 : 755,5 F2.13.4 : 806,7
STA1.14.3 : 658,2 F1.14.3 : 758,8
STA2.14.5 : 758,8 F2.14.5 : 810
STA1.15.4 : 692,2 F1.15.4 : 792,8
STA2.15.1 : 792,8 F2.15.1 : 844
ST1.49.1 : 286,9 C1.49.1 : 294
ST3.49.1 : 452,4 C3.49.1 : 546
ST1.48.2 : 287,1 C1.48.2 : 294
ST3.48.1 : 546 C3.48.1 : 623,7
ST1.50.2 : 294 C1.50.2 : 304,4
ST2.50.2 : 304,4 C2.50.2 : 309,9
ST4.50.2 : 338,2 C4.50.2 : 350,5
ST1.51.1 : 294 C1.51.1 : 301,6
ST2.51.2 : 309,9 C2.51.2 : 313
ST4.51.1 : 349,2 C4.51.1 : 357,3
STA1.40.2 : 695,2 F1.40.2 : 720,2
STA2.40.2 : 806,7 F2.40.2 : 836,9
STA1.41.2 : 720,2 F1.41.2 : 745,2
STA2.41.3 : 806,7 F2.41.3 : 836,9
STA1.42.2 : 745,2 F1.42.2 : 770,2
STA2.42.4 : 806,7 F2.42.4 : 836,9
STA1.50.1 : 755,5 F1.50.1 : 825,1
STA2.50.5 : 825,1 F2.50.5 : 899,6
ST1.21.1 : 301,6 C1.21.1 : 314,1
ST2.21.1 : 338,7 C2.21.1 : 348,3
ST4.21.2 : 350,5 C4.21.2 : 373,8
ST1.24.2 : 304,4 C1.24.2 : 313,2
ST2.24.1 : 348,3 C2.24.1 : 356,3
ST4.24.1 : 357,3 C4.24.1 : 384,6
ST1.20.2 : 313,2 C1.20.2 : 325,1
ST2.20.1 : 356,3 C2.20.1 : 367,3
ST4.20.2 : 373,8 C4.20.2 : 381,9
ST1.22.1 : 314,1 C1.22.1 : 322,1
ST2.22.2 : 322,1 C2.22.2 : 324,8
ST4.22.2 : 381,9 C4.22.2 : 391,2
ST1.23.1 : 322,1 C1.23.1 : 330,2
ST2.23.2 : 330,2 C2.23.2 : 339,5
STA1.20.5 : 755,5 F1.20.5 : 824,1
STA2.20.2 : 836,9 F2.20.2 : 881
STA1.21.6 : 755,5 F1.21.6 : 824,1
STA2.21.3 : 836,9 F2.21.3 : 881
STA1.22.3 : 758,8 F1.22.3 : 827,4
STA2.22.4 : 836,9 F2.22.4 : 881
STA1.23.2 : 770,2 F1.23.2 : 838,8
STA2.23.1 : 844 F2.23.1 : 888,1
ST1.56.2 : 325,1 C1.56.2 : 338,6
ST2.56.1 : 367,8 C2.56.1 : 383,8
ST1.54.1 : 330,2 C1.54.1 : 339,7
ST2.54.2 : 339,7 C2.54.2 : 347,7
ST4.54.1 : 384,6 C4.54.1 : 396,3
ST1.53.2 : 338,6 C1.53.2 : 347,7
ST2.53.1 : 383,8 C2.53.1 : 395,8
ST4.53.2 : 395,8 C4.53.2 : 403,3
ST1.52.1 : 339,7 C1.52.1 : 348,6
ST2.52.1 : 395,8 C2.52.1 : 407,8
ST1.55.2 : 347,7 C1.55.2 : 356
ST2.55.2 : 356 C2.55.2 : 360,1
ST4.55.1 : 396,3 C4.55.1 : 404,4
ST1.57.1 : 348,6 C1.57.1 : 356,8
ST2.57.2 : 360,1 C2.57.2 : 366,7
STA1.43.4 : 792,7 F1.43.4 : 863,9
STA2.43.2 : 881 F2.43.2 : 920,1
STA1.44.5 : 824,1 F1.44.5 : 895,3
STA2.44.3 : 895,3 F2.44.3 : 934,4
STA1.45.6 : 824,1 F1.45.6 : 895,3
STA2.45.4 : 895,3 F2.45.4 : 934,4
STA1.47.1 : 825,1 F1.47.1 : 854,7
STA2.47.1 : 888,1 F2.47.1 : 936,2
STA1.48.3 : 827,4 F1.48.3 : 857
STA2.48.5 : 899,6 F2.48.5 : 947,7
ST1.8.2 : 356 C1.8.2 : 366,5
ST2.8.1 : 408,3 C2.8.1 : 422,8
ST4.8.2 : 422,8 C4.8.2 : 436,5
ST1.7.1 : 356,8 C1.7.1 : 368,8
ST2.7.1 : 422,8 C2.7.1 : 427,1
ST4.7.1 : 427,1 C4.7.1 : 434,8
STA1.7.2 : 838,8 F1.7.2 : 889,8
STA2.7.2 : 920,1 F2.7.2 : 978,5
STA1.8.1 : 854,7 F1.8.1 : 905,7
STA2.8.3 : 934,4 F2.8.3 : 992,8
ST1.32.2 : 366,5 C1.32.2 : 378,3
ST2.32.1 : 427,1 C2.32.1 : 434,6
ST4.32.1 : 434,8 C4.32.1 : 447,7
ST1.31.1 : 368,8 C1.31.1 : 377,1
ST2.31.1 : 434,6 C2.31.1 : 443,7

ST4.31.2 : 443,7 C4.31.2 : 451 STA2.30.4 : 934,4 F2.30.4 : 997,2 STA2.31.1 : 936,2 F2.31.1 : 998,9
STA1.30.3 : 857 F1.30.3 : 896,2 STA1.31.4 : 864 F1.31.4 : 903,2

FFS3J5 Partilere Ayrılmış En Az Boşluk Kuralı Sonuçları

ST1.3.1 : 0 C1.3.1 : 20 ST1.5.3 : 16 C1.5.3 : 28 STA1.4.3 : 34 F1.4.3 : 62
ST1.1.2 : 0 C1.1.2 : 13 STL2.8.3 : 28 CL2.8.3 : 31 STL2.5.2 : 16 CL2.5.2 : 24
ST1.2.3 : 0 C1.2.3 : 8 STL2.7.4 : 0 CL2.7.4 : 2,5 STL2.6.1 : 26,5 CL2.6.1 : 33
STL2.2.2 : 8 CL2.2.2 : 16 STA1.3.2 : 31 F1.3.2 : 59 STA1.2.4 : 33 F1.2.4 : 75
STL2.3.1 : 20 CL2.3.1 : 26,5 STL2.10.3 : 31 CL2.10.3 : 34
STA1.1.1 : 26,5 F1.1.1 : 68,5 STL2.9.4 : 2,5 CL2.9.4 : 5

FFS3J15 Partilere Ayrılmış En Az Boşluk Kuralı Sonuçları

ST1.6.1 : 0 C1.6.1 : 17 STL2.21.3 : 17 CL2.21.3 : 25,5 STL2.26.1 : 50,5 CL2.26.1 :
ST1.8.2 : 0 C1.8.2 : 10 STL2.23.3 : 33 CL2.23.3 : 41,5 55,5
ST1.7.3 : 0 C1.7.3 : 8 STL2.22.1 : 36 CL2.22.1 : 41 STA1.10.2 : 58,5 F1.10.2 : 98,5
STL2.10.3 : 8 CL2.10.3 : 17 STA1.9.3 : 41,5 F1.9.3 : 81,5 STL2.13.3 : 58,5 CL2.13.3 :
STL2.9.4 : 17 CL2.9.4 : 20,5 ST1.10.4 : 26 C1.10.4 : 40 67,5
STA1.5.1 : 20,5 F1.5.1 : 58,5 ST1.9.5 : 29 C1.9.5 : 42 STL2.12.4 : 20,5 CL2.12.4 : 24
ST1.3.4 : 0 C1.3.4 : 18 STL2.17.1 : 41 CL2.17.1 : 50,5 STA1.6.1 : 67,5 F1.6.1 : 105,5
ST1.1.5 : 0 C1.1.5 : 11 STL2.16.2 : 40 CL2.16.2 : 44 ST1.4.2 : 41 C1.4.2 : 59
ST1.2.5 : 13 C1.2.5 : 21 STA1.7.4 : 50,5 F1.7.4 : 71,5 STL2.8.4 : 24 CL2.8.4 : 36
STL2.3.2 : 18 CL2.3.2 : 26,5 STL2.6.2 : 44 CL2.6.2 : 52,5 STL2.7.3 : 67,5 CL2.7.3 : 77,5
STL2.1.2 : 26,5 CL2.1.2 : 32 STL2.4.2 : 52,5 CL2.4.2 : 58 STA1.3.4 : 77,5 F1.3.4 : 118,5
STA1.1.2 : 32 F1.1.2 : 53 STA1.2.5 : 58 F1.2.5 : 79 STA1.4.5 : 79 F1.4.5 : 120
ST1.15.3 : 16 C1.15.3 : 34 STL2.25.3 : 41,5 CL2.25.3 : 50 STL2.20.1 : 55,5 CL2.20.1 : 65
ST1.14.2 : 18 C1.14.2 : 33 STL2.27.3 : 50 CL2.27.3 : 58,5 STL2.19.2 : 58 CL2.19.2 : 62
ST1.13.1 : 25 C1.13.1 : 36 STA1.8.3 : 81,5 F1.8.3 : 102,5

FFS3J30 Partilere Ayrılmış En Az Boşluk Kuralı Sonuçları

ST1.10.1 : 0 C1.10.1 : 15 ST1.5.3 : 86 C1.5.3 : 98 STL2.1.3 : 82,5 CL2.1.3 : 92
ST1.11.2 : 0 C1.11.2 : 15 STL2.9.2 : 88 CL2.9.2 : 96,5 STL2.3.3 : 149 CL2.3.3 : 155,5
ST1.9.3 : 0 C1.9.3 : 14 STA1.4.6 : 96,5 F1.4.6 : 131,5 STL2.2.2 : 158 CL2.2.2 : 162
STL2.20.1 : 14 CL2.20.1 : 23 ST1.22.2 : 89 C1.22.2 : 104 STA1.1.2 : 162 F1.1.2 : 190
STL2.22.2 : 15 CL2.22.2 : 24 ST1.23.1 : 96 C1.23.1 : 106 STL2.7.2 : 162 CL2.7.2 : 170,5
STA1.8.1 : 24 F1.8.1 : 71 STL2.42.4 : 104 CL2.42.4 : 113,5 STA1.3.3 : 170,5 F1.3.3 : 205,5
ST1.16.3 : 22 C1.16.3 : 40 STL2.41.2 : 96,5 CL2.41.2 : 101 STL2.45.4 : 126,5 CL2.45.4 : 136
ST1.15.1 : 23 C1.15.1 : 38 STA1.15.1 :113,5 F1.15.1 : 162,5 STL2.44.2 : 170,5 CL2.44.2 : 175
ST1.17.2 : 23 C1.17.2 : 33 STL2.17.1 : 64,5 CL2.17.1 : 73,5 STA1.16.1 : 175 F1.16.1 : 224
STL2.32.3 : 38 CL2.32.3 : 48 STL2.19.2 : 101 CL2.19.2 : 110 STL2.26.4 : 136 CL2.26.4 : 143
STL2.34.1 : 33 CL2.34.1 : 42,5 STA1.7.2 : 110 F1.7.2 : 157 STL2.28.2 : 175 CL2.28.2 : 181
STL2.33.3 : 48 CL2.33.3 : 54 ST1.29.3 : 106 C1.29.3 : 115 STL2.27.3 : 155,5 CL2.27.3 : 159
STA1.12.2 : 54 F1.12.2 : 75 ST1.27.2 : 112 C1.27.2 : 120 STA1.10.6 : 181 F1.10.6 : 202
STL2.29.3 : 54 CL2.29.3 : 64 STL2.55.4 : 115 CL2.55.4 : 120,5 STL2.15.3 : 159 CL2.15.3 : 168
STL2.31.1 : 42,5 CL2.31.1 : 52 STL2.54.1 : 73,5 CL2.54.1 : 76,5 STL2.14.2 : 181 CL2.14.2 : 186,5
STL2.30.3 : 64 CL2.30.3 : 70 STL2.56.2 : 110 CL2.56.2 : 113 STL2.16.1 : 152 CL2.16.1 : 154,5
STA1.11.3 : 70 F1.11.3 : 91 STL2.53.4 : 120,5 CL2.53.4 : 123 STA1.6.4 : 189,5 F1.6.4 : 217,5
ST1.6.2 : 41 C1.6.2 : 59 STA1.19.3 : 123 F1.19.3 : 158 STL2.4.3 : 168 CL2.4.3 : 177,5
ST1.8.1 : 46 C1.8.1 : 62 ST1.20.1 : 114 C1.20.1 : 133 STL2.6.3 : 177,5 CL2.6.3 : 184
STL2.12.3 : 70 CL2.12.3 : 79 ST1.19.3 : 123 C1.19.3 : 132 STL2.5.2 : 186,5 CL2.5.2 : 190,5
STL2.11.2 : 59 CL2.11.2 : 64,5 ST1.18.2 : 128 C1.18.2 : 136 STA1.2.2 : 190,5 F1.2.2 : 218,5
STL2.13.1 : 62 CL2.13.1 : 64,5 STL2.40.1 : 133 CL2.40.1 : 142,5 STL2.59.4 : 143 CL2.59.4 : 148,5
STA1.5.4 : 79 F1.5.4 : 107 STA1.14.4 :142,5 F1.14.4 : 189,5 STL2.58.1 :154,5 CL2.58.1 :157,5
ST1.12.3 : 48 C1.12.3 : 62 STL2.37.1 : 142,5 CL2.37.1 : 152 STL2.60.2 :190,5 CL2.60.2 :193,5
ST1.13.3 : 64 C1.13.3 : 78 STA1.13.5 : 152 F1.13.5 : 199 STL2.57.4 : 148,5 CL2.57.4 : 151
ST1.14.2 : 67 C1.14.2 : 81 ST1.26.3 : 140 C1.26.3 : 157 STA1.20.5 : 199 F1.20.5 : 234
STL2.23.4 : 62 CL2.23.4 : 69 STL2.47.4 : 123 CL2.47.4 : 126,5 STL2.50.4 : 151 CL2.50.4 : 154,5
STL2.25.2 : 81 CL2.25.2 : 87 STL2.48.2 : 113 CL2.48.2 : 116 STL2.51.2 :193,5 CL2.51.2 :196,5
STL2.24.3 : 79 CL2.24.3 : 82,5 STA1.17.6 :131,5 F1.17.6 : 173,5 STA1.18.6 : 202 F1.18.6 : 244
STA1.9.5 : 87 F1.9.5 : 108 ST1.2.1 : 141 C1.2.1 : 158
ST1.4.1 : 70 C1.4.1 : 88 ST1.3.2 : 144 C1.3.2 : 149

FFS6J5 Partilere Ayrılmış En Az Boşluk Kuralı Sonuçları

ST1.2.1 : 0 C1.2.1 : 11 ST1.5.2 : 17 C1.5.2 : 25 STL3.10.2 : 49,5 CL3.10.2 : 54,5
ST1.1.2 : 0 C1.1.2 : 9 STL2.7.1 : 32 CL2.7.1 : 42 STL4.10.1 : 54,5 CL4.10.1 : 61
ST1.3.3 : 0 C1.3.3 : 8 STL3.7.2 : 42 CL3.7.2 : 49,5 STA1.4.3 : 65,5 F1.4.3 : 109,5
STL2.1.3 : 9 CL2.1.3 : 18,5 STL4.7.2 : 49,5 CL4.7.2 : 55,5 STA2.4.3 : 109,5 F2.4.3 : 138,5
STL3.1.1 : 18,5 CL3.1.1 : 23,5 STL2.8.2 : 25 CL2.8.2 : 33 STL2.4.3 : 26,5 CL2.4.3 : 36
STL4.1.1 : 23,5 CL4.1.1 : 29,5 STL3.8.1 : 33 CL3.8.1 : 38 STL3.4.2 : 54,5 CL3.4.2 : 59,5
STL3.2.2 : 11 CL3.2.2 : 18,5 STL4.8.1 : 38 CL4.8.1 : 44,5 STL4.4.2 : 59,5 CL4.4.2 : 65,5
STL4.2.2 : 18,5 CL4.2.2 : 27,5 STA1.3.2 : 55,5 F1.3.2 : 99,5 STL3.5.1 : 59,5 CL3.5.1 : 67
STL2.3.3 : 18,5 CL2.3.3 : 26,5 STA2.3.2 : 99,5 F2.3.2 : 128,5 STL4.5.1 : 67 CL4.5.1 : 76
STL4.3.3 : 26,5 CL4.3.3 : 32,5 STL2.9.1 : 42 CL2.9.1 : 52 STL2.6.3 : 36 CL2.6.3 : 44
STA1.1.1 : 32,5 F1.1.1 : 81,5 STL3.9.1 : 52 CL3.9.1 : 59,5 STL4.6.2 : 65,5 CL4.6.2 : 71,5
STA2.1.1 : 81,5 F2.1.1 : 121,5 STL4.9.3 : 59,5 CL4.9.3 : 65,5 STA1.2.4 : 76 F1.2.4 : 125
ST1.4.3 : 16 C1.4.3 : 32 STL2.10.2 : 33 CL2.10.2 : 41 STA2.2.4 : 125 F2.2.4 : 165

FFS6J15 Partilere Ayrılmış En Az Boşluk Kuralı Sonuçları

ST1.7.1 : 0 C1.7.1 : 20 STL4.14.3 : 19 CL4.14.3 : 24 STL4.5.1 : 28,5 CL4.5.1 : 38
ST1.8.2 : 0 C1.8.2 : 11 STA1.6.1 : 33,5 F1.6.1 : 57,5 STL2.4.5 : 24,5 CL2.4.5 : 29
ST1.6.3 : 0 C1.6.3 : 8 STA2.6.1 : 57,5 F2.6.1 : 97,5 STL4.4.2 : 33,5 CL4.4.2 : 40
STL2.12.1 : 8 CL2.12.1 : 17,5 ST1.1.4 : 0 C1.1.4 : 12 STA1.2.2 : 42 F1.2.2 : 75
STL3.12.1 : 17,5 CL3.12.1 : 22 ST1.2.4 : 14 C1.2.4 : 25 STA2.2.2 : 75 F2.2.2 : 124
STL4.12.1 : 22 CL4.12.1 : 27,5 ST1.3.3 : 16 C1.3.3 : 25 STL2.9.1 : 17,5 CL2.9.1 : 27
STL2.13.5 : 20 CL2.13.5 : 24,5 STL2.6.6 : 25 CL2.6.6 : 30,5 STL3.9.5 : 27 CL3.9.5 : 31,5
STL3.13.2 : 24,5 CL3.13.2 : 27,5 STL3.6.4 : 30,5 CL3.6.4 : 35 STL4.9.1 : 38 CL4.9.1 : 43,5
STL4.13.2 : 27,5 CL4.13.2 : 33,5 STL4.6.3 : 35 CL4.6.3 : 42 STL2.10.5 : 29 CL2.10.5 : 33,5
STL3.14.3 : 11 CL3.14.3 : 19 STL2.5.4 : 25 CL2.5.4 : 28,5 STL3.10.6 : 33,5 CL3.10.6 : 36,5

STL4.10.2 : 40 CL4.10.2 : 46
STL3.11.3 : 19 CL3.11.3 : 27
STL4.11.3 : 42 CL4.11.3 : 47
STA1.5.3 : 47 F1.5.3 : 71
STA2.5.3 : 71 F2.5.3 : 111
STL2.3.6 : 30,5 CL2.3.6 : 36
STL3.3.1 : 36 CL3.3.1 : 40,5
STL4.3.1 : 43,5 CL4.3.1 : 50,5
STL2.2.4 : 28,5 CL2.2.4 : 32
STL4.2.2 : 46 CL4.2.2 : 55,5
STL2.1.5 : 33,5 CL2.1.5 : 38
STL4.1.3 : 47 CL4.1.3 : 53,5
STA1.1.4 : 55,5 F1.1.4 : 88,5
STA2.1.4 : 88,5 F2.1.4 : 137,5
ST1.9.2 : 19 C1.9.2 : 37
ST1.10.1 : 28 C1.10.1 : 36
STL2.17.2 : 0 CL2.17.2 : 8
STL3.17.3 : 27 CL3.17.3 : 35
STL4.17.1 : 50,5 CL4.17.1 : 60,5
STL2.15.3 : 37 CL2.15.3 : 40
STL3.15.2 : 40 CL3.15.2 : 49
STL4.15.3 : 53,5 CL4.15.3 : 58
STL2.16.4 : 36 CL2.16.4 : 44
STL3.16.5 : 44 CL3.16.5 : 47,5
STL4.16.2 : 55,5 CL4.16.2 : 58
STA1.7.5 : 60,5 F1.7.5 : 81,5
STA2.7.5 : 81,5 F2.7.5 : 115,5
ST1.15.3 : 33 C1.15.3 : 52
ST1.13.4 : 33 C1.13.4 : 49
ST1.14.1 : 44 C1.14.1 : 57
STL2.21.3 : 40 CL2.21.3 : 47
STL3.21.3 : 47 CL3.21.3 : 52,5
STL4.21.2 : 58 CL4.21.2 : 67
STL2.23.4 : 57 CL2.23.4 : 61,5
STL3.23.4 : 61,5 CL3.23.4 : 70
STL4.23.3 : 70 CL4.23.3 : 77
STL2.24.6 : 52 CL2.24.6 : 61
STL4.24.1 : 61 CL4.24.1 : 66,5
STL2.22.5 : 49 CL2.22.5 : 54
STL3.22.6 : 54 CL3.22.6 : 57,5
STL4.22.1 : 66,5 CL4.22.1 : 71,5
STA1.9.6 : 77 F1.9.6 : 113
STA2.9.6 : 113 F2.9.6 : 155
STL2.25.3 : 47 CL2.25.3 : 54
STL3.25.1 : 54 CL3.25.1 : 59,5
STL4.25.2 : 67 CL4.25.2 : 76
STL2.27.4 : 61,5 CL2.27.4 : 66
STL3.27.5 : 66 CL3.27.5 : 74,5
STL4.27.1 : 74,5 CL4.27.1 : 81,5
STL2.28.6 : 61 CL2.28.6 : 70
STL4.28.2 : 76 CL4.28.2 : 81,5
STL2.26.5 : 54 CL2.26.5 : 59
STL3.26.2 : 59 CL3.26.2 : 62,5
STL4.26.3 : 77 CL4.26.3 : 82
STA1.10.1 : 82 F1.10.1 : 118
STA2.10.1 : 118 F2.10.1 : 160
ST1.4.2 : 45 C1.4.2 : 61
STL2.7.5 : 61 CL2.7.5 : 80
STL4.7.1 : 81,5 CL4.7.1 : 97,5
STL3.8.3 : 52,5 CL3.8.3 : 62,5
STL4.8.2 : 81,5 CL4.8.2 : 89,5
STA1.3.3 : 97,5 F1.3.3 : 147,5
STA2.3.3 : 147,5 F2.3.3 : 176,5
STA1.4.2 : 97,5 F1.4.2 : 147,5
STA2.4.5 : 147,5 F2.4.5 : 176,5
STL2.20.2 : 8 CL2.20.2 : 16
STL3.20.6 : 57,5 CL3.20.6 : 65,5
STL4.20.3 : 82 CL4.20.3 : 92
STL2.18.3 : 54 CL2.18.3 : 57
STL3.18.1 : 59,5 CL3.18.1 : 68,5
STL4.18.2 : 89,5 CL4.18.2 : 94
STL2.19.4 : 66 CL2.19.4 : 74
STL3.19.2 : 74 CL3.19.2 : 77,5
STL4.19.3 : 92 CL4.19.3 : 94,5
STA1.8.5 : 94,5 F1.8.5 : 115,5
STA2.8.2 : 124 F2.8.2 : 158

FFS6J30 Partilere Ayrılmış En Az Boşluk Kuralı Sonuçları

ST1.10.1 : 0 C1.10.1 : 20
ST1.9.2 : 0 C1.9.2 : 19
ST1.11.3 : 0 C1.11.3 : 12
STL2.16.3 : 12 CL2.16.3 : 21,5
STL3.16.1 : 21,5 CL3.16.1 : 25
STL4.16.1 : 25 CL4.16.1 : 35
STL2.14.3 : 21,5 CL2.14.3 : 31,5
STL4.14.2 : 31,5 CL4.14.2 : 41
STL2.15.3 : 31,5 CL2.15.3 : 39
STL3.15.2 : 39 CL3.15.2 : 44
STL4.15.3 : 44 CL4.15.3 : 51
STA1.7.1 : 51 F1.7.1 : 84
STA2.7.1 : 84 F2.7.1 : 122
ST1.27.3 : 20 C1.27.3 : 36
ST1.29.2 : 27 C1.29.2 : 34
STL2.49.3 : 39 CL2.49.3 : 44,5
STL3.49.3 : 44,5 CL3.49.3 : 51,5
STL4.49.4 : 51,5 CL4.49.4 : 58
STL2.47.1 : 36 CL2.47.1 : 40,5
STL4.47.1 : 40,5 CL4.47.1 : 50
STL2.48.2 : 0 CL2.48.2 : 2,5
STL3.48.4 : 2,5 CL3.48.4 : 9,5
STL4.48.2 : 41 CL4.48.2 : 44,5
STL2.50.3 : 44,5 CL2.50.3 : 49,5
STL3.50.4 : 49,5 CL3.50.4 : 52,5
STL4.50.2 : 52,5 CL4.50.2 : 57,5
STA1.19.2 : 58 F1.19.2 : 88
STA2.19.2 : 88 F2.19.2 : 117
ST1.17.1 : 28 C1.17.1 : 36
ST1.15.1 : 38 C1.15.1 : 45
ST1.16.2 : 42 C1.16.2 : 49
STL2.28.2 : 36 CL2.28.2 : 42
STL3.28.1 : 42 CL3.28.1 : 48
STL4.28.1 : 50 CL4.28.1 : 56,5
STL2.26.2 : 45 CL2.26.2 : 53
STL3.26.2 : 53 CL3.26.2 : 56,5
STL4.26.3 : 56,5 CL4.26.3 : 62
STL3.27.1 : 49 CL3.27.1 : 53
STL4.27.1 : 56,5 CL4.27.1 : 59,5
STA1.11.3 : 62 F1.11.3 : 111
STA2.11.3 : 111 F2.11.3 : 148
STL2.31.2 : 53 CL2.31.2 : 59
STL3.31.3 : 59 CL3.31.3 : 65
STL4.31.2 : 65 CL4.31.2 : 71,5
STL2.29.2 : 59 CL2.29.2 : 67
STL3.29.4 : 67 CL3.29.4 : 70,5
STL4.29.4 : 70,5 CL4.29.4 : 76
STL3.30.1 : 53 CL3.30.1 : 57
STL4.30.1 : 59,5 CL4.30.1 : 62,5
STA1.12.4 : 76 F1.12.4 : 125
STA2.12.4 : 125 F2.12.4 : 162
ST1.19.3 : 44 C1.19.3 : 63
ST1.20.1 : 53 C1.20.1 : 69
ST1.18.2 : 57 C1.18.2 : 72
STL2.33.2 : 67 CL2.33.2 : 76
STL3.33.2 : 76 CL3.33.2 : 86
STL4.33.3 : 86 CL4.33.3 : 92
STL2.34.3 : 69 CL2.34.3 : 75
STL3.34.1 : 75 CL3.34.1 : 77,5
STL4.34.1 : 77,5 CL4.34.1 : 81
STL3.32.3 : 72 CL3.32.3 : 75,5
STL4.32.2 : 75,5 CL4.32.2 : 83,5
STA1.13.5 : 92 F1.13.5 : 117
STA2.13.5 : 117 F2.13.5 : 138
ST1.14.3 : 71 C1.14.3 : 89
ST1.13.1 : 77 C1.13.1 : 86
ST1.12.2 : 80 C1.12.2 : 86
STL2.20.2 : 86 CL2.20.2 : 93
STL3.20.4 : 93 CL3.20.4 : 97
STL4.20.4 : 97 CL4.20.4 : 99,5
STL2.21.1 : 86 CL2.21.1 : 92,5
STL4.21.1 : 92,5 CL4.21.1 : 99
STL2.22.1 : 92,5 CL2.22.1 : 99,5
STL3.22.3 :99,5 CL3.22.3 : 104,5
STA1.9.6 : 104,5 F1.9.6 : 127,5
STA2.9.6 : 127,5 F2.9.6 : 175,5
ST1.6.1 : 94 C1.6.1 : 111
ST1.8.2 : 94 C1.8.2 : 111
STL2.9.1 : 99,5 CL2.9.1 : 104
STL3.9.1 : 104 CL3.9.1 : 110
STL4.9.2 : 110 CL4.9.2 : 113
STL2.10.1 : 111 CL2.10.1 : 116,5
STL3.10.2 :116,5 CL3.10.2 :119,5
STL4.10.3 :119,5 CL4.10.3 :123,5
STL2.8.3 : 111 CL2.8.3 : 118,5
STL4.8.1 : 118,5 CL4.8.1 : 122
STA1.5.1 : 123,5 F1.5.1 : 164,5
STA2.5.2 : 164,5 F2.5.2 : 195,5
STL2.53.3 : 118,5 CL2.53.3 : 124
STL3.53.4 : 124 CL3.53.4 : 131
STL4.53.4 : 131 CL4.53.4 : 137,5
STL2.51.1 : 116,5 CL2.51.1 : 121
STL4.51.2 : 121 CL4.51.2 : 130,5
STL2.52.2 : 93 CL2.52.2 : 95,5
STL3.52.3 :104,5 CL3.52.3 :111,5
STL4.52.1 : 122 CL4.52.1 : 125,5
STL2.54.3 : 124 CL2.54.3 : 129
STL3.54.1 : 129 CL3.54.1 : 132
STL4.54.3 : 132 CL4.54.3 : 137
STA1.20.2 :137,5 F1.20.2 : 167,5
STA2.20.1 :167,5 F2.20.1 : 196,5
ST1.4.3 : 97 C1.4.3 : 109
ST1.5.3 : 111 C1.5.3 : 119
STL2.5.3 : 129 CL2.5.3 : 138
STL3.5.3 : 138 CL3.5.3 : 140,5
STL4.5.1 : 140,5 CL4.5.1 : 147,5
STL2.4.3 : 138 CL2.4.3 : 141,5
STL3.4.2 : 141,5 CL3.4.2 : 149,5
STL4.4.2 : 149,5 CL4.4.2 : 152
STA1.3.3 : 152 F1.3.3 : 174
STA2.3.5 : 174 F2.3.5 : 207
ST1.26.1 : 119 C1.26.1 : 128
STL2.42.1 : 121 CL2.42.1 : 127,5
STL3.42.4 : 131 CL3.42.4 : 140
STL4.42.3 : 140 CL4.42.3 : 149,5
STL2.41.1 : 127,5 CL2.41.1 : 137
STL3.41.1 : 137 CL3.41.1 : 141,5
STL4.41.4 :141,5 CL4.41.4 :147,5
STL2.43.2 : 128 CL2.43.2 : 133
STL4.43.1 :147,5 CL4.43.1 :154,5
STA1.17.5 :154,5 F1.17.5 : 185,5
STA2.17.3 :185,5 F2.17.3 : 206,5
ST1.23.2 : 119 C1.23.2 : 138
ST1.22.3 : 127 C1.22.3 : 139
STL2.39.2 : 139 CL2.39.2 : 157
STL3.39.4 : 157 CL3.39.4 : 171
STL4.39.4 : 171 CL4.39.4 : 178
STL2.40.2 : 157 CL2.40.2 : 167
STL3.40.3 : 167 CL3.40.3 : 181
STL4.40.3 : 181 CL4.40.3 : 194
STL2.38.1 : 137 CL2.38.1 : 148
STL3.38.1 : 148 CL3.38.1 : 158
STL4.38.2 : 158 CL4.38.2 : 166
STA1.15.4 : 194 F1.15.4 : 228
STA2.15.4 : 228 F2.15.4 : 276
STA1.16.6 : 194 F1.16.6 : 228
STA2.16.6 : 228 F2.16.6 : 276
ST1.2.1 : 136 C1.2.1 : 145
ST1.3.2 : 146 C1.3.2 : 154
STL2.2.2 : 167 CL2.2.2 : 182
STL3.2.2 : 182 CL3.2.2 : 200
STL4.2.1 : 200 CL4.2.1 : 214
STL2.3.3 : 154 CL2.3.3 : 168
STL3.3.1 : 168 CL3.3.1 : 181
STL4.3.2 : 181 CL4.3.2 : 191
STL2.1.2 : 182 CL2.1.2 : 202
STL4.1.4 : 202 CL4.1.4 : 216
STA1.1.1 : 216 F1.1.1 : 251
STA2.1.2 : 251 F2.1.2 : 281
STA1.2.2 : 216 F1.2.2 : 251
STA2.2.1 : 251 F2.2.1 : 281
STL2.19.3 : 168 CL2.19.3 : 177,5
STL3.19.4 : 177,5 CL3.19.4 : 181
STL4.19.2 : 191 CL4.19.2 : 201
STL2.17.3 :177,5 CL2.17.3 :187,5
STL4.17.3 : 194 CL4.17.3 : 203,5
STL2.18.3 : 187,5 CL2.18.3 : 195
STL3.18.1 : 195 CL3.18.1 : 200
STL4.18.2 : 201 CL4.18.2 : 208
STA1.8.3 : 208 F1.8.3 : 241
STA2.8.3 : 241 F2.8.3 : 279
STL2.36.2 : 202 CL2.36.2 : 211
STL3.36.3 : 211 CL3.36.3 : 221
STL4.36.3 : 221 CL4.36.3 : 227
STL2.37.3 : 195 CL2.37.3 : 201
STL3.37.4 : 201 CL3.37.4 : 203,5
STL4.37.2 : 208 CL4.37.2 : 211,5
STL3.35.1 : 200 CL3.35.1 : 203,5
STL4.35.2 :211,5 CL4.35.2 :219,5
STA1.14.5 : 227 F1.14.5 : 252
STA2.14.5 : 252 F2.14.5 : 273
STL2.12.1 : 148 CL2.12.1 : 152,5
STL3.12.2 : 200 CL3.12.2 : 206
STL4.12.1 : 214 CL4.12.1 : 217
STL2.13.1 : 152,5 CL2.13.1 : 158
STL3.13.1 :203,5 CL3.13.1 :206,5
STL4.13.4 : 216 CL4.13.4 : 220
STL2.11.3 : 201 CL2.11.3 : 208,5
STL4.11.1 : 217 CL4.11.1 : 220,5
STA1.6.4 : 228 F1.6.4 : 269
STA2.6.5 : 273 F2.6.5 : 304
STL2.23.2 : 211 CL2.23.2 : 218
STL3.23.4 : 218 CL3.23.4 : 222
STL4.23.2 : 222 CL4.23.2 : 224,5
STL2.24.1 : 158 CL2.24.1 : 164,5
STL4.24.4 : 220 CL4.24.4 : 226,5
STL2.25.1 :164,5 CL2.25.1 :171,5
STL3.25.2 : 206 CL3.25.2 : 211
STA1.10.6 : 228 F1.10.6 : 251
STA2.10.4 : 276 F2.10.4 : 324
STL2.45.1 : 171,5 CL2.45.1 : 178
STL3.45.1 :206,5 CL3.45.1 :215,5
STL4.45.1 : 220,5 CL4.45.1 : 230
STL2.44.1 : 178 CL2.44.1 : 187,5
STL3.44.2 : 211 CL3.44.2 : 215,5
STL4.44.2 :224,5 CL4.44.2 :230,5
STL2.46.2 : 218 CL2.46.2 : 223
STL4.46.4 :226,5 CL4.46.4 :233,5
STA1.18.3 : 241 F1.18.3 : 272
STA2.18.6 : 276 F2.18.6 : 297
STL2.7.3 : 208,5 CL2.7.3 : 217,5
STL3.7.1 : 217,5 CL3.7.1 : 220
STL4.7.3 : 227 CL4.7.3 : 234
STL2.6.3 : 217,5 CL2.6.3 : 221
STL3.6.2 : 221 CL3.6.2 : 229
STL4.6.1 : 230 CL4.6.1 : 232,5
STA1.4.1 : 251 F1.4.1 : 273

STA2.4.3 : 279 F2.4.3 : 312

FFS6J66 Partilere Ayrılmış En Az Boşluk Kuralı Sonuçları

ST1.34.1 : 0 C1.34.1 : 24,6
ST1.33.2 : 0 C1.33.2 : 14,5
ST1.35.2 : 14,5 C1.35.2 : 27,4
ST1.38.1 : 24,6 C1.38.1 : 35,8
ST1.37.2 : 27,4 C1.37.2 : 38,2
ST1.36.1 : 35,8 C1.36.1 : 46,1
STL2.46.2 : 24,6 CL2.46.2 : 30
STL4.46.1 : 30 CL4.46.1 : 38,9
STL2.50.2 : 35,8 CL2.50.2 : 40,4
STL4.50.2 : 40,4 CL4.50.2 : 49,3
STL2.45.1 : 14,5 CL2.45.1 : 20,1
STL4.45.1 : 38,9 CL4.45.1 : 45,2
STL2.47.2 : 40,4 CL2.47.2 : 43,1
STL4.47.1 : 45,2 CL4.47.1 : 52,1
STL2.48.2 : 46,1 CL2.48.2 : 48,2
STL4.48.2 : 49,3 CL4.48.2 : 55,4
STL2.49.2 : 48,2 CL2.49.2 : 52,2
STA1.32.1 : 55,4 F1.32.1 : 115,8
STA2.32.1 : 115,8 F2.32.1 : 214
ST1.1.2 : 38,2 C1.1.2 : 53,4
ST1.5.1 : 46,1 C1.5.1 : 61,2
ST1.3.2 : 53,4 C1.3.2 : 63,2
ST1.2.1 : 61,2 C1.2.1 : 70,9
ST1.6.2 : 63,2 C1.6.2 : 70,5
ST1.4.2 : 70,5 C1.4.2 : 76,3
STL2.1.2 : 53,4 CL2.1.2 : 55,9
STL4.1.1 : 55,9 CL4.1.1 : 61,8
STL2.3.2 : 63,2 CL2.3.2 : 65,2
STL4.3.2 : 65,2 CL4.3.2 : 69,7
STL2.2.2 : 70,9 CL2.2.2 : 71,9
STL4.2.1 : 71,9 CL4.2.1 : 76,6
STL2.5.1 : 61,2 CL2.5.1 : 66
STA1.1.2 : 76,6 F1.1.2 : 180,2
STA2.1.2 : 180,2 F2.1.2 : 227,9
ST1.9.1 : 70,9 C1.9.1 : 78,6
STL2.21.1 : 78,6 CL2.21.1 : 84,7
STA1.9.3 : 84,7 F1.9.3 : 121,9
STA2.9.3 : 121,9 F2.9.3 : 180,5
ST1.39.2 : 76,3 C1.39.2 : 91,4
ST1.41.1 : 78,6 C1.41.1 : 90,6
ST1.40.1 : 90,6 C1.40.1 : 100,4
ST1.42.2 : 91,4 C1.42.2 : 100,9
STL3.58.1 :100,4 CL3.58.1 :151,6
STL4.58.2 :151,6 CL4.58.2 :163,3
STL2.57.1 :91,4 CL2.57.1 : 102,9
STL4.57.1 : 102,9 CL4.57.1 : 129
STL2.59.1 :102,9 CL2.59.1 :109,3
STL4.59.1 : 129 CL4.59.1 : 142,9
STL2.60.2 : 100,9 CL2.60.2 : 107,8
STL4.60.1 : 142,9 CL4.60.1 : 151,9
STA1.35.4 : 163,3 F1.35.4 : 208,8
STA2.35.4 : 208,8 F2.35.4 : 274,7
STA1.36.5 : 163,3 F1.36.5 : 208,8
STA2.36.5 : 208,8 F2.36.5 : 274,7
ST1.63.1 : 100,4 C1.63.1 : 111,6
ST1.64.2 : 100,9 C1.64.2 : 109,5
ST1.61.2 : 109,5 C1.61.2 : 117,9
ST1.66.1 : 111,6 C1.66.1 : 120
ST1.65.2 : 117,9 C1.65.2 : 126,2
ST1.62.1 : 120 C1.62.1 : 127,5
STL2.86.2 : 126,2 CL2.86.2 : 132,7
STL4.86.1 : 151,9 CL4.86.1 : 160,2
STL2.85.1 : 109,8 CL2.85.1 : 114,4
STL4.85.1 : 160,2 CL4.85.1 : 165,5
STL2.83.2 : 132,7 CL2.83.2 : 135,1
STL4.83.2 : 163,3 CL4.83.2 : 168,4
STL2.82.1 : 117,9 CL2.82.1 : 120,3
STL4.82.1 : 165,5 CL4.82.1 : 170,4
STL2.84.2 : 135,1 CL2.84.2 : 140,8
STA1.49.6 : 170,4 F1.49.6 : 240
STA2.49.3 : 240 F2.49.3 : 314,5
STL2.7.2 : 140,8 CL2.7.2 : 143,3
STL4.7.2 : 168,4 CL4.7.2 : 174,3
STL2.9.2 : 143,3 CL2.9.2 : 145,3
STL4.9.1 : 170,4 CL4.9.1 : 174,9
STL2.8.2 : 145,3 CL2.8.2 : 146,3
STL4.8.2 : 174,3 CL4.8.2 : 179
STL2.11.1 : 120,8 CL2.11.1 : 125,6
STA1.2.1 : 179 F1.2.1 : 282,6
STA2.2.1 : 282,6 F2.2.1 : 330,3
ST1.58.2 : 126,2 C1.58.2 : 148,4
ST1.59.1 : 127,5 C1.59.1 : 139,3
ST1.60.1 : 139,3 C1.60.1 : 149
STL2.76.1 : 148,4 CL2.76.1 : 153,5
STL4.76.1 : 174,9 CL4.76.1 : 180,2
STL2.77.1 : 153,5 CL2.77.1 : 156,2
STL4.77.2 : 179 CL4.77.2 : 185,5
STL2.78.2 : 149 CL2.78.2 : 151,9
STL4.78.1 : 180,2 CL4.78.1 : 185,3
STA1.46.3 : 185,5 F1.46.3 : 215,1
STA2.46.2 : 227,9 F2.46.2 : 276
ST1.44.2 : 148,4 C1.44.2 : 180,6
ST1.43.1 : 149 C1.43.1 : 180,9
ST1.47.2 : 180,6 C1.47.2 : 198,1
ST1.45.1 : 180,9 C1.45.1 : 192,4
ST1.46.1 : 192,4 C1.46.1 : 202,6
STL3.62.1 : 180,6 CL3.62.1 : 331,5
STL3.61.1 : 331,5 CL3.61.1 : 454,2
STL2.63.2 : 192,4 CL2.63.2 : 200,4
STL4.63.1 : 200,4 CL4.63.1 : 212,1
STL2.64.2 : 202,6 CL2.64.2 : 211,5
STL4.64.2 : 211,5 CL4.64.2 : 222
STL2.65.2 : 211,5 CL2.65.2 : 227,5
STA1.37.2 : 454,2 F1.37.2 : 553,1
STA2.37.4 : 553,1 F2.37.4 : 614,1
STA1.38.4 : 454,2 F1.38.4 : 553,1
STA2.38.5 : 553,1 F2.38.5 : 614,1
STA1.39.5 : 454,2 F1.39.5 : 553,1
STA2.39.2 : 553,1 F2.39.2 : 614,1
STL2.52.2 : 227,5 CL2.52.2 : 238,3
STL4.52.1 : 238,3 CL4.52.1 : 252,8
STL2.56.2 : 238,3 CL2.56.2 : 247,6
STL4.56.2 : 247,6 CL4.56.2 : 262,1
STL2.51.1 : 156,7 CL2.51.1 : 167,9
STL4.51.1 : 252,8 CL4.51.1 : 262,1
STL2.53.2 : 247,6 CL2.53.2 : 253
STL4.53.1 : 262,1 CL4.53.1 : 272,6
STL2.54.2 : 253 CL2.54.2 : 257,3
STL4.54.2 : 262,1 CL4.54.2 : 271
STL2.55.2 : 257,3 CL2.55.2 : 265,3
STA1.33.3 : 272,6 F1.33.3 : 333
STA2.33.3 : 333 F2.33.3 : 431,2
STA1.34.6 : 272,6 F1.34.6 : 333
STA2.34.1 : 333 F2.34.1 : 431,2
ST1.16.2 : 198,1 C1.16.2 : 218,8
ST1.19.1 : 202,6 C1.19.1 : 219,9
ST1.17.2 : 218,8 C1.17.2 : 234
ST1.18.1 : 219,9 C1.18.1 : 229,4
STL2.31.2 : 265,3 CL2.31.2 : 274,3
STL4.31.2 : 274,3 CL4.31.2 : 306,4
STL2.28.1 : 218,8 CL2.28.1 : 240,7
STL4.28.1 : 272,6 CL4.28.1 : 280,7
STL2.29.1 : 240,7 CL2.29.1 : 251,4
STL4.29.1 : 280,7 CL4.29.1 : 300
STL2.30.2 : 274,3 CL2.30.2 : 277,3
STL4.30.1 : 300 CL4.30.1 : 308,1
STA1.16.1 : 308,1 F1.16.1 : 387,1
STA2.16.1 : 431,2 F2.16.1 : 502,8
STA1.17.3 : 333 F1.17.3 : 412
STA2.17.3 : 431,2 F2.17.3 : 502,8
STA1.18.6 : 333 F1.18.6 : 412
STA2.18.1 : 502,8 F2.18.1 : 574,4
STA1.19.1 : 387,1 F1.19.1 : 466,1
STA2.19.3 : 502,8 F2.19.3 : 574,4
ST1.26.1 : 229,4 C1.26.1 : 241,3
ST1.30.2 : 234 C1.30.2 : 243,1
ST1.25.1 : 241,3 C1.25.1 : 250,3
ST1.28.2 : 243,1 C1.28.2 : 252
ST1.27.1 : 250,3 C1.27.1 : 258,8
ST1.29.2 : 252 C1.29.2 : 259,6
STL2.38.2 : 277,3 CL2.38.2 : 289,8
STL4.38.2 : 306,4 CL4.38.2 : 327,7
STL2.40.1 : 252 CL2.40.1 : 279,8
STL2.37.1 : 280,3 CL2.37.1 : 292
STL4.37.1 : 308,1 CL4.37.1 : 318,6
STL2.42.2 : 289,8 CL2.42.2 : 301,3
STL4.42.1 : 318,6 CL4.42.1 : 327,9
STA1.24.3 : 412 F1.24.3 : 522,9
STA2.24.1 : 574,4 F2.24.1 : 624,4
STA1.25.6 : 412 F1.25.6 : 522,9
STA2.25.3 : 574,4 F2.25.3 : 624,4
STA1.26.1 : 466,1 F1.26.1 : 577
STA2.26.2 : 614,1 F2.26.2 : 664,1
STA1.27.3 : 522,9 F1.27.3 : 633,8
STA2.27.4 : 633,8 F2.27.4 : 683,8
STA1.28.6 : 522,9 F1.28.6 : 633,8
STA2.28.5 : 633,8 F2.28.5 : 683,8
STA1.29.2 : 553,1 F1.29.2 : 664
STA2.29.1 : 664 F2.29.1 : 714
ST1.11.1 : 258,8 C1.11.1 : 270,4
ST1.15.2 : 259,6 C1.15.2 : 269,1
ST1.13.2 : 269,1 C1.13.2 : 278,3
ST1.10.1 : 270,4 C1.10.1 : 279,4
ST1.12.2 : 278,3 C1.12.2 : 287
ST1.14.1 : 279,4 C1.14.1 : 287
STL2.23.2 : 301,3 CL2.23.2 : 313,8
STL4.23.2 : 327,7 CL4.23.2 : 349
STL2.25.1 : 292,5 CL2.25.1 : 320,3
STL2.22.1 : 320,8 CL2.22.1 : 332,5
STL4.22.1 : 332,5 CL4.22.1 : 343
STL2.27.2 : 313,8 CL2.27.2 : 325,5
STL4.27.1 : 343 CL4.27.1 : 352,3
STA1.10.4 : 553,1 F1.10.4 : 653,7
STA2.10.3 : 653,7 F2.10.3 : 704,9
STA1.11.5 : 553,1 F1.11.5 : 653,7
STA2.11.2 : 664,1 F2.11.2 : 715,3
STA1.12.1 : 577 F1.12.1 : 677,6
STA2.12.4 : 683,8 F2.12.4 : 735
STA1.13.3 : 633,8 F1.13.3 : 734,4
STA2.13.5 : 734,4 F2.13.5 : 785,6
STA1.14.6 : 633,8 F1.14.6 : 734,4
STA2.14.3 : 734,4 F2.14.3 : 785,6
STA1.15.4 : 653,7 F1.15.4 : 754,3
STA2.15.1 : 754,3 F2.15.1 : 805,5
ST1.50.1 : 287 C1.50.1 : 297,4
ST1.51.2 : 287 C1.51.2 : 294,6
ST1.49.2 : 294,6 C1.49.2 : 301,7
ST1.48.1 : 297,4 C1.48.1 : 304,3
STL3.67.1 : 454,2 CL3.67.1 : 547,8
STL3.66.1 : 547,8 CL3.66.1 : 625,5
STL2.68.2 : 325,5 CL2.68.2 : 331
STL4.68.2 : 349 CL4.68.2 : 361,3
STL2.69.2 : 331 CL2.69.2 : 334,1
STL4.69.1 : 352,3 CL4.69.1 : 360,4
STA1.40.5 : 653,7 F1.40.5 : 678,7
STA2.40.2 : 715,3 F2.40.2 : 745,5
STA1.41.2 : 664 F1.41.2 : 689
STA2.41.4 : 735 F2.41.4 : 765,2
STA1.42.1 : 677,6 F1.42.1 : 702,6
STA2.42.2 : 745,5 F2.42.2 : 775,7
STL2.13.2 : 334,1 CL2.13.2 : 344,1
STL4.13.1 : 360,4 CL4.13.1 : 374,1
STL2.17.1 : 333 CL2.17.1 : 352,2
STL2.15.2 : 344,1 CL2.15.2 : 352
STL4.15.2 : 361,3 CL4.15.2 : 369,4
STL2.14.2 : 352 CL2.14.2 : 356,2
STL4.14.2 : 369,4 CL4.14.2 : 378,3
STA1.3.5 : 678,7 F1.3.5 : 782,3
STA2.3.4 : 782,3 F2.3.4 : 830
STA1.4.2 : 689 F1.4.2 : 792,6
STA2.4.2 : 792,6 F2.4.2 : 840,3
STA1.5.1 : 702,6 F1.5.1 : 806,2
STA2.5.3 : 806,2 F2.5.3 : 853,9
STA1.6.3 : 734,4 F1.6.3 : 837,9999
STA2.6.5 : 838 F2.6.5 : 885,7
ST1.21.2 : 301,7 C1.21.2 : 314,2
ST1.20.1 : 304,3 C1.20.1 : 316,2
ST1.24.2 : 314,2 C1.24.2 : 323
ST1.23.1 : 316,2 C1.23.1 : 324,3
ST1.22.2 : 323 C1.22.2 : 331
STL2.36.1 : 352,7 CL2.36.1 : 360,7
STL4.36.1 : 374,1 CL4.36.1 : 401,4
STL2.33.1 : 360,7 CL2.33.1 : 370,3
STL4.33.2 : 378,3 CL4.33.2 : 401,6
STL2.32.1 : 370,3 CL2.32.1 : 381,3
STL4.32.1 : 401,4 CL4.32.1 : 409,5
STL2.34.2 : 356,2 CL2.34.2 : 358,9
STL4.34.2 : 401,6 CL4.34.2 : 410,9
STL2.35.2 : 358,9 CL2.35.2 : 368,2
STA1.20.6 : 734,4 F1.20.6 : 803
STA2.20.1 : 805,5 F2.20.1 : 849,6
STA1.21.4 : 754,3 F1.21.4 : 822,9
STA2.21.4 : 830 F2.21.4 : 874,1
STA1.22.5 : 782,3 F1.22.5 : 850,9
STA2.22.2 : 850,9 F2.22.2 : 895
STA1.23.2 : 792,6 F1.23.2 : 861,2
STA2.23.1 : 861,2 F2.23.1 : 905,3
ST1.56.1 : 324,3 C1.56.1 : 337,8
ST1.54.2 : 331 C1.54.2 : 340,5
ST1.53.1 : 337,8 C1.53.1 : 346,9
ST1.52.2 : 340,5 C1.52.2 : 349,4
ST1.55.1 : 346,9 C1.55.1 : 355,2
ST1.57.2 : 349,4 C1.57.2 : 357,6
STL2.72.2 : 368,2 CL2.72.2 : 376,2
STL4.72.1 : 409,5 CL4.72.1 : 421,2
STL2.71.1 : 381,8 CL2.71.1 : 393,8
STL4.71.2 : 410,9 CL4.71.2 : 418,4
STL2.74.1 : 393,8 CL2.74.1 : 409,8
STL2.73.2 : 376,2 CL2.73.2 : 380,3
STL4.73.2 : 418,4 CL4.73.2 : 426,5
STL2.70.1 : 409,8 CL2.70.1 : 421,8
STL2.75.2 : 380,3 CL2.75.2 : 386,9
STA1.43.6 : 803 F1.43.6 : 874,2
STA2.43.3 : 874,2 F2.43.3 : 914
STA1.44.1 : 806,2 F1.44.1 : 877,4
STA2.44.4 : 877,4 F2.44.4 : 916,5
STA1.45.4 : 822,9 F1.45.4 : 894,1
STA2.45.5 : 894,1 F2.45.5 : 933,2
STL2.92.2 : 386,9 CL2.92.2 : 393,4
STL4.92.1 : 421,2 CL4.92.1 : 429,5
STL2.91.1 : 421,8 CL2.91.1 : 426,4
STL4.91.2 : 426,5 CL4.91.2 : 431,8
STL2.89.2 : 393,4 CL2.89.2 : 395,8
STL4.89.1 : 429,5 CL4.89.1 : 434,6
STL2.88.1 : 426,9 CL2.88.1 : 429,3
STL4.88.2 : 431,8 CL4.88.2 : 436,7
STL2.90.2 : 395,8 CL2.90.2 : 401,5
STA1.50.3 : 838 F1.50.3 : 907,6
STA2.50.2 : 907,6 F2.50.2 : 982,1
ST1.7.1 : 355,2 C1.7.1 : 367,2
ST1.8.2 : 357,6 C1.8.2 : 368,1
STL2.20.1 : 429,3 CL2.20.1 : 443,8
STL4.20.1 : 443,8 CL4.20.1 : 457,5
STL2.19.1 : 443,8 CL2.19.1 : 448,1
STL4.19.2 : 448,1 CL4.19.2 : 455,8
STA1.7.5 : 850,9 F1.7.5 : 901,9
STA2.7.1 : 905,3 F2.7.1 : 963,7
STA1.8.2 : 861,2 F1.8.2 : 912,2
STA2.8.3 : 913,3 F2.8.3 : 971,7
STL2.79.1 : 448,1 CL2.79.1 : 458,2
STL4.79.2 : 458,2 CL4.79.2 : 465,5
STL2.80.1 : 458,2 CL2.80.1 : 463,5
STL4.80.1 : 463,5 CL4.80.1 : 473,2
STL2.81.2 : 401,5 CL2.81.2 : 407,2
STL4.81.2 : 465,5 CL4.81.2 : 472,4
STA1.47.6 : 874,2 F1.47.6 : 903,8
STA2.47.4 : 916,5 F2.47.4 : 964,6
STA1.48.1 : 877,4 F1.48.1 : 907
STA2.48.5 : 933,2 F2.48.5 : 981,3
ST1.32.1 : 367,2 C1.32.1 : 379
ST1.31.2 : 368,1 C1.31.2 : 376,4
STL2.44.1 : 463,5 CL2.44.1 : 471
STL4.44.2 : 472,4 CL4.44.2 : 485,3
STL2.43.1 : 471 CL2.43.1 : 480,1
STL4.43.1 : 480,1 CL4.43.1 : 487,4
STA1.30.4 : 894,1 F1.30.4 : 933,3
STA2.30.1 : 963,7 F2.30.1 : 1026,5
STA1.31.5 : 901,9 F1.31.5 : 941,1
STA2.31.4 : 964,6 F2.31.4 : 1027,4

FFS3J5 En Kısa İşlem Zamanı Sonuçları

ST2.4.4 : 0 C2.4.4 : 5
ST1.5.1 : 0 C1.5.1 : 12
ST2.5.3 : 12 C2.5.3 : 18
STA1.3.1 : 18 F1.3.1 : 46
STA1.4.2 : 18 F1.4.2 : 46
ST1.1.2 : 0 C1.1.2 : 13
ST1.2.3 : 0 C1.2.3 : 8
ST2.2.2 : 8 C2.2.2 : 24
ST1.3.3 : 10 C1.3.3 : 30
ST2.3.1 : 30 C2.3.1 : 43
STA1.1.3 : 43 F1.1.3 : 85
STA1.2.4 : 43 F1.2.4 : 85

FFS3J15 En Kısa İşlem Zamanı Sonuçları

ST1.9.1 : 0 C1.9.1 : 13
ST2.11.1 : 0 C2.11.1 : 19
ST1.10.2 : 0 C1.10.2 : 14
ST2.10.2 : 14 C2.10.2 : 22
STA1.7.1 : 22 F1.7.1 : 43
STA1.8.2 : 22 F1.8.2 : 43
ST1.2.3 : 0 C1.2.3 : 8
ST1.1.4 : 0 C1.1.4 : 11
ST2.1.2 : 22 C2.1.2 : 33
ST1.3.5 : 0 C1.3.5 : 18
ST2.3.2 : 33 C2.3.2 : 50
STA1.1.3 : 50 F1.1.3 : 71
STA1.2.4 : 50 F1.2.4 : 71
ST2.5.4 : 0 C2.5.4 : 12
ST1.4.3 : 16 C1.4.3 : 34
ST2.4.3 : 34 C2.4.3 : 44
STA1.3.5 : 44 F1.3.5 : 85
STA1.4.1 : 44 F1.4.1 : 85
ST1.8.4 : 19 C1.8.4 : 29
ST1.6.1 : 21 C1.6.1 : 38
ST2.6.4 : 38 C2.6.4 : 45
ST1.7.2 : 22 C1.7.2 : 30
ST2.7.3 : 44 C2.7.3 : 62
STA1.5.2 : 62 F1.5.2 : 100
STA1.6.3 : 71 F1.6.3 : 109
ST2.12.3 : 62 C2.12.3 : 79
ST1.15.5 : 26 C1.15.5 : 44
ST1.13.4 : 37 C1.13.4 : 48
ST2.13.1 : 48 C2.13.1 : 58
ST1.14.2 : 38 C1.14.2 : 53
ST2.14.3 : 79 C2.14.3 : 96
STA1.9.4 : 96 F1.9.4 : 136
STA1.10.1 : 96 F1.10.1 : 136

FFS3J30 En Kısa İşlem Zamanı Sonuçları

ST2.25.2 : 0 C2.25.2 : 6
ST2.24.4 : 0 C2.24.4 : 7
ST1.26.1 : 0 C1.26.1 : 17
STA1.17.1 : 17 F1.17.1 : 59
STA1.18.2 : 17 F1.18.2 : 59
ST2.28.1 : 0 C2.28.1 : 6
ST2.30.2 : 6 C2.30.2 : 12
ST1.27.2 : 0 C1.27.2 : 8
ST2.27.4 : 8 C2.27.4 : 13
ST1.29.3 : 0 C1.29.3 : 9
ST2.29.4 : 13 C2.29.4 : 24
STA1.19.3 : 24 F1.19.3 : 59
STA1.20.4 : 24 F1.20.4 : 59
ST1.5.2 : 16 C1.5.2 : 28
ST1.4.3 : 17 C1.4.3 : 35
ST2.4.2 : 35 C2.4.2 : 52
STA1.3.5 : 52 F1.3.5 : 87
STA1.4.6 : 52 F1.4.6 : 87
ST1.13.1 : 25 C1.13.1 : 39
ST2.13.3 : 39 C2.13.3 : 46
ST1.14.2 : 36 C1.14.2 : 50
ST2.14.2 : 52 C2.14.2 : 64
ST1.12.1 : 41 C1.12.1 : 55
ST2.12.4 : 55 C2.12.4 : 69
STA1.9.1 : 69 F1.9.1 : 90
STA1.10.2 : 69 F1.10.2 : 90
ST1.3.3 : 43 C1.3.3 : 48
ST2.3.3 : 48 C2.3.3 : 61
ST2.1.3 : 61 C2.1.3 : 80
ST1.2.3 : 50 C1.2.3 : 67
ST2.2.2 : 67 C2.2.2 : 75
STA1.1.3 : 80 F1.1.3 : 108
STA1.2.4 : 80 F1.2.4 : 108
ST2.7.3 : 80 C2.7.3 : 98
ST1.8.2 : 58 C1.8.2 : 74
ST2.8.1 : 74 C2.8.1 : 79
ST1.6.1 : 63 C1.6.1 : 81
ST2.6.2 : 81 C2.6.2 : 92
STA1.5.5 : 98 F1.5.5 : 126
STA1.6.6 : 98 F1.6.6 : 126
ST1.17.3 : 75 C1.17.3 : 85
ST2.17.1 : 85 C2.17.1 : 104
ST1.16.2 : 82 C1.16.2 : 100
ST2.16.3 : 100 C2.16.3 : 112
ST1.15.3 : 87 C1.15.3 : 102
ST2.15.3 : 112 C2.15.3 : 132
STA1.11.1 : 132 F1.11.1 : 153
STA1.12.2 : 132 F1.12.2 : 153
ST1.18.1 : 89 C1.18.1 : 97
ST1.19.1 : 99 C1.19.1 : 108
ST1.20.2 : 108 C1.20.2 : 127
ST2.20.1 : 127 C2.20.1 : 146
STA1.13.3 : 146 F1.13.3 : 193
STA1.14.4 : 146 F1.14.4 : 193
ST2.21.2 : 92 C2.21.2 : 101
ST1.23.3 : 110 C1.23.3 : 120
ST1.22.1 : 116 C1.22.1 : 131
ST2.22.4 : 131 C2.22.4 : 150
STA1.15.5 : 150 F1.15.5 : 199
STA1.16.6 : 150 F1.16.6 : 199
ST1.10.3 : 128 C1.10.3 : 143
ST1.9.2 : 135 C1.9.2 : 149
ST2.9.1 : 149 C2.9.1 : 167
ST1.11.1 : 139 C1.11.1 : 154
ST2.11.2 : 154 C2.11.2 : 172
STA1.7.1 : 172 F1.7.1 : 219
STA1.8.2 : 172 F1.8.2 : 219

FFS6J5 En Kısa İşlem Zamanı Sonuçları

ST1.5.1 : 0 C1.5.1 : 8
ST2.5.2 : 8 C2.5.2 : 24
ST3.5.1 : 24 C3.5.1 : 34
ST4.5.1 : 34 C4.5.1 : 47
ST1.4.2 : 0 C1.4.2 : 16
ST2.4.1 : 16 C2.4.1 : 36
ST3.4.2 : 36 C3.4.2 : 51
ST4.4.2 : 51 C4.4.2 : 63
STA1.3.1 : 63 F1.3.1 : 107
STA2.3.1 : 107 F2.3.1 : 136
STA1.4.2 : 63 F1.4.2 : 107
STA2.4.2 : 107 F2.4.2 : 136
ST1.3.3 : 0 C1.3.3 : 8
ST2.3.3 : 8 C2.3.3 : 24
ST4.3.3 : 24 C4.3.3 : 36
ST1.2.3 : 10 C1.2.3 : 21
ST3.2.1 : 34 C3.2.1 : 49
ST4.2.3 : 49 C4.2.3 : 67
ST1.1.1 : 16 C1.1.1 : 25
ST2.1.3 : 25 C2.1.3 : 44
ST3.1.1 : 49 C3.1.1 : 59
ST4.1.1 : 59 C4.1.1 : 71
STA1.1.3 : 71 F1.1.3 : 120
STA2.1.3 : 120 F2.1.3 : 160
STA1.2.4 : 71 F1.2.4 : 120
STA2.2.4 : 120 F2.2.4 : 160

FFS6J15 En Kısa İşlem Zamanı Sonuçları

ST3.5.1 : 0 C3.5.1 : 10
ST4.5.1 : 10 C4.5.1 : 18
ST1.4.1 : 0 C1.4.1 : 16
ST2.4.5 : 16 C2.4.5 : 35
ST4.4.2 : 35 C4.4.2 : 51
STA1.3.1 : 51 F1.3.1 : 101
STA2.3.1 : 101 F2.3.1 : 130
STA1.4.2 : 51 F1.4.2 : 101
STA2.4.2 : 101 F2.4.2 : 130
ST1.10.2 : 0 C1.10.2 : 8
ST2.10.4 : 8 C2.10.4 : 24
ST3.10.2 : 24 C3.10.2 : 31
ST4.10.3 : 31 C4.10.3 : 36
ST1.9.3 : 0 C1.9.3 : 18
ST2.9.3 : 18 C2.9.3 : 24
ST3.9.3 : 24 C3.9.3 : 42
ST4.9.1 : 42 C4.9.1 : 51
ST2.11.2 : 0 C2.11.2 : 16
ST3.11.4 : 16 C3.11.4 : 32
ST4.11.3 : 36 C4.11.3 : 56
STA1.7.3 : 56 F1.7.3 : 77
STA2.7.3 : 77 F2.7.3 : 111
STA1.8.4 : 56 F1.8.4 : 77
STA2.8.4 : 77 F2.8.4 : 111
ST1.8.4 : 0 C1.8.4 : 11
ST3.8.5 : 11 C3.8.5 : 27
ST4.8.1 : 51 C4.8.1 : 61
ST1.6.4 : 13 C1.6.4 : 21
ST2.6.1 : 21 C2.6.1 : 40
ST3.6.6 : 40 C3.6.6 : 49
ST4.6.2 : 51 C4.6.2 : 62
ST1.7.2 : 16 C1.7.2 : 36
ST2.7.5 : 36 C2.7.5 : 45
ST3.7.1 : 45 C3.7.1 : 51
ST4.7.3 : 56 C4.7.3 : 68
STA1.5.5 : 68 F1.5.5 : 92
STA2.5.5 : 92 F2.5.5 : 132
STA1.6.6 : 68 F1.6.6 : 92
STA2.6.6 : 92 F2.6.6 : 132
ST1.1.1 : 24 C1.1.1 : 36
ST2.1.5 : 45 C2.1.5 : 54
ST4.1.1 : 61 C4.1.1 : 74
ST1.2.3 : 26 C1.2.3 : 37
ST2.2.4 : 37 C2.2.4 : 44
ST4.2.2 : 62 C4.2.2 : 81
ST1.3.4 : 29 C1.3.4 : 38
ST2.3.6 : 38 C2.3.6 : 49
ST3.3.5 : 49 C3.3.5 : 58
ST4.3.3 : 68 C4.3.3 : 82
STA1.1.3 : 82 F1.1.3 : 115
STA2.1.3 : 115 F2.1.3 : 164
STA1.2.4 : 82 F1.2.4 : 115
STA2.2.4 : 115 F2.2.4 : 164
ST2.12.3 : 24 C2.12.3 : 38
ST3.12.2 : 38 C3.12.2 : 49
ST4.12.1 : 74 C4.12.1 : 92
ST1.13.1 : 44 C1.13.1 : 60
ST2.13.5 : 60 C2.13.5 : 70
ST3.13.4 : 70 C3.13.4 : 77
ST4.13.2 : 81 C4.13.2 : 91
ST1.15.2 : 44 C1.15.2 : 63
ST2.15.6 : 63 C2.15.6 : 81
ST4.15.3 : 82 C4.15.3 : 93
ST1.14.3 : 45 C1.14.3 : 58
ST2.14.4 : 58 C2.14.4 : 67
ST3.14.3 : 67 C3.14.3 : 84
ST4.14.2 : 91 C4.14.2 : 105
STA1.9.5 : 105 F1.9.5 : 141
STA2.9.1 : 141 F2.9.1 : 183
STA1.10.6 : 105 F1.10.6 : 141
STA2.10.2 : 141 F2.10.2 : 183

FFS6J30 En Kısa İşlem Zamanı Sonuçları

ST1.4.1 : 0 C1.4.1 : 12
ST2.4.3 : 12 C2.4.3 : 19
ST3.4.1 : 19 C3.4.1 : 35
ST4.4.1 : 35 C4.4.1 : 40
ST1.5.2 : 0 C1.5.2 : 8
ST2.5.3 : 19 C2.5.3 : 37

ST3.5.2 : 37 C3.5.2 : 42
ST4.5.2 : 42 C4.5.2 : 56
STA1.3.1 : 56 F1.3.1 : 78
STA2.3.1 : 78 F2.3.1 : 111
STA1.4.2 : 56 F1.4.2 : 78
STA2.4.2 : 78 F2.4.2 : 111
ST1.26.3 : 0 C1.26.3 : 9
ST2.26.2 : 9 C2.26.2 : 19
ST4.26.3 : 19 C4.26.3 : 33
ST2.24.1 : 0 C2.24.1 : 19
ST3.24.3 : 19 C3.24.3 : 28
ST4.24.4 : 28 C4.24.4 : 40
ST2.25.1 : 19 C2.25.1 : 32
ST3.25.4 : 32 C3.25.4 : 50
ST4.25.3 : 50 C4.25.3 : 69
STA1.17.3 : 69 F1.17.3 : 100
STA2.17.3 : 100 F2.17.3 : 121
STA1.18.4 : 69 F1.18.4 : 100
STA2.18.4 : 100 F2.18.4 : 121
ST1.18.2 : 16 C1.18.2 : 31
ST3.18.3 : 31 C3.18.3 : 38
ST4.18.1 : 40 C4.18.1 : 56
ST1.20.3 : 17 C1.20.3 : 33
ST2.20.3 : 37 C2.20.3 : 49
ST3.20.1 : 49 C3.20.1 : 54
ST4.20.4 : 54 C4.20.4 : 61
ST1.19.1 : 20 C1.19.1 : 39
ST2.19.2 : 39 C2.19.2 : 57
ST3.19.3 : 57 C3.19.3 : 77
ST4.19.1 : 77 C4.19.1 : 89
STA1.13.5 : 89 F1.13.5 : 114
STA2.13.5 : 114 F2.13.5 : 135
STA1.14.6 : 89 F1.14.6 : 114
STA2.14.6 : 114 F2.14.6 : 135
ST2.7.1 : 32 C2.7.1 : 41
ST3.7.2 : 42 C3.7.2 : 54
ST4.7.2 : 56 C4.7.2 : 62
ST1.6.2 : 39 C1.6.2 : 56
ST2.6.3 : 56 C2.6.3 : 71
ST4.6.4 : 71 C4.6.4 : 78
ST1.8.3 : 41 C1.8.3 : 58
ST2.8.1 : 58 C2.8.1 : 69
ST3.8.4 : 69 C3.8.4 : 75
ST4.8.2 : 75 C4.8.2 : 83
STA1.5.1 : 83 F1.5.1 : 124
STA2.5.1 : 124 F2.5.1 : 155

STA1.6.2 : 83 F1.6.2 : 124
STA2.6.2 : 124 F2.6.2 : 155
ST1.12.1 : 47 C1.12.1 : 53
ST2.12.2 : 57 C2.12.2 : 71
ST3.12.1 : 71 C3.12.1 : 79
ST4.12.3 : 79 C4.12.3 : 84
ST1.13.1 : 55 C1.13.1 : 64
ST2.13.1 : 69 C2.13.1 : 82
ST4.13.4 : 82 C4.13.4 : 95
ST1.14.2 : 64 C1.14.2 : 82
ST2.14.1 : 82 C2.14.1 : 96
ST3.14.2 : 96 C3.14.2 : 106
STA1.9.3 : 106 F1.9.3 : 129
STA2.9.3 : 129 F2.9.3 : 177
STA1.10.4 : 106 F1.10.4 : 129
STA2.10.4 : 129 F2.10.4 : 177
ST2.28.2 : 71 C2.28.2 : 76
ST3.28.4 : 76 C3.28.4 : 90
ST4.28.2 : 90 C4.28.2 : 97
ST2.30.3 : 71 C2.30.3 : 81
ST3.30.3 : 81 C3.30.3 : 87
ST4.30.3 : 87 C4.30.3 : 97
ST1.27.3 : 66 C1.27.3 : 82
ST2.27.1 : 96 C2.27.1 : 105
ST4.27.1 : 105 C4.27.1 : 124
ST1.29.1 : 72 C1.29.1 : 79
ST2.29.3 : 81 C2.29.3 : 92
ST3.29.1 : 92 C3.29.1 : 106
ST4.29.4 : 106 C4.29.4 : 119
STA1.19.5 : 124 F1.19.5 : 154
STA2.19.5 : 154 F2.19.5 : 183
STA1.20.6 : 124 F1.20.6 : 154
STA2.20.6 : 154 F2.20.6 : 183
ST2.1.2 : 76 C2.1.2 : 96
ST4.1.2 : 97 C4.1.2 : 111
ST1.3.1 : 87 C1.3.1 : 95
ST2.3.3 : 95 C2.3.3 : 109
ST3.3.3 : 109 C3.3.3 : 122
ST4.3.3 : 122 C4.3.3 : 132
ST1.2.2 : 90 C1.2.2 : 99
ST2.2.2 : 99 C2.2.2 : 114
ST3.2.4 : 114 C3.2.4 : 132
ST4.2.2 : 132 C4.2.2 : 146
STA1.1.1 : 146 F1.1.1 : 181
STA2.1.1 : 181 F2.1.1 : 211
STA1.2.2 : 146 F1.2.2 : 181

STA2.2.2 : 181 F2.2.2 : 211
ST1.16.3 : 90 C1.16.3 : 97
ST3.16.1 : 106 C3.16.1 : 114
ST4.16.4 : 119 C4.16.4 : 125
ST1.15.3 : 99 C1.15.3 : 106
ST2.15.2 : 114 C2.15.2 : 130
ST3.15.2 : 130 C3.15.2 : 137
ST4.15.1 : 137 C4.15.1 : 148
ST1.17.1 : 103 C1.17.1 : 111
ST2.17.2 : 130 C2.17.2 : 142
ST3.17.1 : 142 C3.17.1 : 154
ST4.17.4 : 154 C4.17.4 : 167
STA1.11.3 : 167 F1.11.3 : 216
STA2.11.3 : 216 F2.11.3 : 253
STA1.12.4 : 167 F1.12.4 : 216
STA2.12.4 : 216 F2.12.4 : 253
ST2.21.1 : 105 C2.21.1 : 116
ST3.21.3 : 122 C3.21.3 : 132
ST4.21.3 : 132 C4.21.3 : 140
ST1.22.2 : 107 C1.22.2 : 119
ST2.22.2 : 142 C2.22.2 : 160
ST3.22.3 : 160 C3.22.3 : 174
ST4.22.3 : 174 C4.22.3 : 181
ST1.23.3 : 114 C1.23.3 : 133
ST2.23.2 : 160 C2.23.2 : 170
ST3.23.4 : 170 C3.23.4 : 184
ST4.23.2 : 184 C4.23.2 : 197
STA1.15.5 : 197 F1.15.5 : 231
STA2.15.5 : 231 F2.15.5 : 279
STA1.16.6 : 197 F1.16.6 : 231
STA2.16.6 : 231 F2.16.6 : 279
ST1.9.1 : 119 C1.9.1 : 138
ST2.9.3 : 138 C2.9.3 : 158
ST4.9.1 : 158 C4.9.1 : 177
ST1.11.2 : 127 C1.11.2 : 139
ST2.11.3 : 158 C2.11.3 : 177
ST3.11.2 : 177 C3.11.2 : 184
ST4.11.4 : 184 C4.11.4 : 204
ST1.10.1 : 140 C1.10.1 : 160
ST2.10.3 : 177 C2.10.3 : 192
ST3.10.1 : 192 C3.10.1 : 202
ST4.10.1 : 202 C4.10.1 : 216
STA1.7.1 : 216 F1.7.1 : 249
STA2.7.1 : 249 F2.7.1 : 287
STA1.8.2 : 216 F1.8.2 : 249
STA2.8.2 : 249 F2.8.2 : 287

FFS6J66 En Kısa İşlem Zamanı Sonuçları

ST1.9.1 : 0 C1.9.1 : 7,7
ST2.9.1 : 7,7 C2.9.1 : 13,8
STA1.9.1 : 13,8 F1.9.1 : 51
STA2.9.1 : 51 F2.9.1 : 109,6
ST1.31.2 : 0 C1.31.2 : 8,3
ST2.31.1 : 13,8 C2.31.1 : 22,9
ST4.31.1 : 22,9 C4.31.1 : 30,2
ST1.32.1 : 7,7 C1.32.1 : 19,5
ST2.32.1 : 22,9 C2.32.1 : 30,4
ST4.32.2 : 30,4 C4.32.2 : 43,3
STA1.30.2 : 43,3 F1.30.2 : 82,5
STA2.30.2 : 82,5 F2.30.2 : 145,3
STA1.31.3 : 43,3 F1.31.3 : 82,5
STA2.31.3 : 82,5 F2.31.3 : 145,3
ST1.7.2 : 8,3 C1.7.2 : 20,3
ST2.7.1 : 30,4 C2.7.1 : 34,7
ST4.7.1 : 34,7 C4.7.1 : 42,4
ST1.8.1 : 19,5 C1.8.1 : 30
ST2.8.1 : 34,7 C2.8.1 : 49,2
ST4.8.1 : 49,2 C4.8.1 : 62,9
STA1.7.4 : 62,9 F1.7.4 : 113,9
STA2.7.4 : 113,9 F2.7.4 : 172,3
STA1.8.5 : 62,9 F1.8.5 : 113,9
STA2.8.5 : 113,9 F2.8.5 : 172,3
ST1.60.2 : 20,3 C1.60.2 : 30
ST2.60.2 : 30 C2.60.2 : 38,6
ST4.60.2 : 43,3 C4.60.2 : 52
ST1.59.1 : 30 C1.59.1 : 41,8
ST2.59.1 : 49,2 C2.59.1 : 57,2
ST4.59.2 : 57,2 C4.59.2 : 70,1
ST1.58.2 : 30 C1.58.2 : 52,2
ST2.58.1 : 57,2 C2.58.1 : 72,4
ST4.58.1 : 72,4 C4.58.1 : 81,7
STA1.46.6 : 81,7 F1.46.6 : 111,3
STA2.46.1 : 111,3 F2.46.1 : 159,4
STA1.47.1 : 81,7 F1.47.1 : 111,3
STA2.47.2 : 145,3 F2.47.2 : 193,4
STA1.48.2 : 82,5 F1.48.2 : 112,1
STA2.48.3 : 145,3 F2.48.3 : 193,4
ST1.51.1 : 41,8 C1.51.1 : 49,4
ST2.51.2 : 49,4 C2.51.2 : 52,5
ST4.51.2 : 70,1 C4.51.2 : 78,2
ST1.50.1 : 49,4 C1.50.1 : 59,8
ST2.50.2 : 59,8 C2.50.2 : 65,3
ST4.50.2 : 78,2 C4.50.2 : 90,5
ST1.48.2 : 52,2 C1.48.2 : 59,1
ST3.48.1 : 59,1 C3.48.1 : 136,8
ST1.49.2 : 59,1 C1.49.2 : 66,2
ST3.49.1 : 136,8 C3.49.1 : 230,4
STA1.40.3 : 230,4 F1.40.3 : 255,4
STA2.40.1 : 255,4 F2.40.1 : 285,6
STA1.41.1 : 230,4 F1.41.1 : 255,4
STA2.41.4 : 255,4 F2.41.4 : 285,6
STA1.42.6 : 230,4 F1.42.6 : 255,4
STA2.42.5 : 255,4 F2.42.5 : 285,6

ST1.42.1 : 59,8 C1.42.1 : 69,3
ST2.42.2 : 69,3 C2.42.2 : 76,2
ST4.42.1 : 81,7 C4.42.1 : 90,7
ST1.41.2 : 66,2 C1.41.2 : 78,2
ST2.41.1 : 78,2 C2.41.1 : 84,60001
ST4.41.2 : 90,5 C4.41.2 : 104,4
ST1.39.1 : 69,3 C1.39.1 : 84,39999
ST2.39.1 : 84,6 C2.39.1 : 96,1
ST4.39.1 : 96,1 C4.39.1 : 122,2
ST1.40.2 : 78,2 C1.40.2 : 88
ST3.40.1 : 230,4 C3.40.1 : 281,6
ST4.40.2 : 281,6 C4.40.2 : 293,3
STA1.35.2 : 293,3 F1.35.2 : 338,8
STA2.35.2 : 338,8 F2.35.2 : 404,7
STA1.36.4 : 293,3 F1.36.4 : 338,8
STA2.36.3 : 338,8 F2.36.3 : 404,7
ST1.66.1 : 84,4 C1.66.1 : 92,8
ST1.62.2 : 88 C1.62.2 : 95,5
ST2.62.2 : 95,5 C2.62.2 : 100,2
ST4.62.1 : 122,2 C4.62.1 : 129,1
ST1.61.1 : 92,8 C1.61.1 : 101,2
ST2.61.1 : 101,2 C2.61.1 : 106
ST4.61.1 : 129,1 C4.61.1 : 135,6
ST1.63.2 : 95,5 C1.63.2 : 106,7
ST2.63.2 : 106,7 C2.63.2 : 118,1
ST1.64.1 : 101,2 C1.64.1 : 109,8
ST2.64.1 : 109,8 C2.64.1 : 119
ST4.64.1 : 135,6 C4.64.1 : 142,9
ST1.65.2 : 106,7 C1.65.2 : 115
ST2.65.2 : 118,1 C2.65.2 : 131,2
ST4.65.1 : 142,9 C4.65.1 : 156,2
STA1.49.5 : 156,2 F1.49.5 : 225,8
STA2.49.1 : 285,6 F2.49.1 : 360,1
STA1.50.5 : 225,8 F1.50.5 : 295,4
STA2.50.4 : 295,4 F2.50.4 : 369,9
ST1.57.1 : 109,8 C1.57.1 : 118
ST2.57.2 : 131,2 C2.57.2 : 137,8
ST1.55.2 : 115 C1.55.2 : 123,3
ST2.55.2 : 137,8 C2.55.2 : 141,9
ST4.55.1 : 156,2 C4.55.1 : 164,3
ST1.52.1 : 118 C1.52.1 : 126,9
ST2.52.1 : 126,9 C2.52.1 : 138,9
ST1.53.2 : 123,3 C1.53.2 : 132,4
ST2.53.1 : 138,9 C2.53.1 : 150,9
ST4.53.1 : 164,3 C4.53.1 : 171,8
ST1.54.1 : 126,9 C1.54.1 : 136,4
ST2.54.2 : 141,9 C2.54.2 : 149,9
ST4.54.1 : 171,8 C4.54.1 : 183,5
ST1.56.2 : 132,4 C1.56.2 : 145,9
ST2.56.1 : 150,9 C2.56.1 : 166,9
STA1.43.1 : 255,4 F1.43.1 : 326,6
STA2.43.5 : 326,6 F2.43.5 : 365,7
STA1.44.3 : 255,4 F1.44.3 : 326,6
STA2.44.1 : 360,1 F2.44.1 : 399,2
STA1.45.6 : 255,4 F1.45.6 : 326,6

STA2.45.5 : 365,7 F2.45.5 : 404,8
ST1.23.1 : 136,4 C1.23.1 : 144,5
ST2.23.2 : 149,9 C2.23.2 : 159,2
ST1.22.1 : 144,5 C1.22.1 : 152,5
ST2.22.2 : 159,2 C2.22.2 : 161,9
ST4.22.1 : 183,5 C4.22.1 : 192,8
ST1.20.2 : 145,9 C1.20.2 : 157,8
ST2.20.1 : 167,4 C2.20.1 : 178,4
ST4.20.1 : 192,8 C4.20.1 : 200,9
ST1.24.1 : 152,5 C1.24.1 : 161,3
ST2.24.1 : 178,4 C2.24.1 : 186,4
ST4.24.1 : 200,9 C4.24.1 : 228,2
ST1.21.2 : 157,8 C1.21.2 : 170,3
ST2.21.1 : 186,4 C2.21.1 : 196
ST4.21.1 : 228,2 C4.21.1 : 251,5
STA1.20.5 : 295,4 F1.20.5 : 364
STA2.20.4 : 369,9 F2.20.4 : 414
STA1.21.1 : 326,6 F1.21.1 : 395,2
STA2.21.1 : 399,2 F2.21.1 : 443,3
STA1.22.3 : 326,6 F1.22.3 : 395,2
STA2.22.2 : 404,7 F2.22.2 : 448,8
STA1.23.6 : 326,6 F1.23.6 : 395,2
STA2.23.3 : 404,7 F2.23.3 : 448,8
ST1.37.1 : 161,3 C1.37.1 : 172,1
ST2.37.2 : 172,1 C2.37.2 : 184,1
ST1.36.2 : 170,3 C1.36.2 : 180,6
ST2.36.2 : 184,1 C2.36.2 : 190,5
ST4.36.1 : 251,5 C4.36.1 : 263,2
ST1.35.1 : 172,1 C1.35.1 : 185
ST2.35.2 : 190,5 C2.35.2 : 198,6
ST4.35.1 : 263,2 C4.35.1 : 277,3
ST1.33.2 : 180,6 C1.33.2 : 195,1
ST2.33.1 : 196,5 C2.33.1 : 213,3
ST4.33.1 : 277,3 C4.33.1 : 289,6
ST1.38.1 : 185 C1.38.1 : 196,2
ST2.38.2 : 198,6 C2.38.2 : 212,5
ST4.38.1 : 289,6 C4.38.1 : 309,7
ST1.34.2 : 195,1 C1.34.2 : 219,7
ST2.34.2 : 219,7 C2.34.2 : 235,9
ST4.34.2 : 293,3 C4.34.2 : 313,4
STA1.32.2 : 338,8 F1.32.2 : 399,2
STA2.32.5 : 404,8 F2.32.5 : 503
STA1.33.4 : 338,8 F1.33.4 : 399,2
STA2.33.4 : 414 F2.33.4 : 512,2
STA1.34.5 : 364 F1.34.5 : 424,4
STA2.34.1 : 443,3 F2.34.1 : 541,5
ST1.18.1 : 196,2 C1.18.1 : 205,7
ST2.18.2 : 235,9 C2.18.2 : 238,9
ST4.18.1 : 309,7 C4.18.1 : 317,8
ST1.17.1 : 205,7 C1.17.1 : 220,9
ST2.17.1 : 220,9 C2.17.1 : 231,6
ST4.17.2 : 313,4 C4.17.2 : 332,7
ST1.16.2 : 219,7 C1.16.2 : 240,4
ST2.16.1 : 240,4 C2.16.1 : 262,3
ST4.16.1 : 317,8 C4.16.1 : 325,9

ST1.19.1 : 220,9 C1.19.1 : 238,2
ST2.19.2 : 238,9 C2.19.2 : 247,9
ST4.19.1 : 325,9 C4.19.1 : 358
STA1.16.1 : 395,2 F1.16.1 : 474,2
STA2.16.2 : 474,2 F2.16.2 : 545,8
STA1.17.3 : 395,2 F1.17.3 : 474,2
STA2.17.3 : 474,2 F2.17.3 : 545,8
STA1.18.6 : 395,2 F1.18.6 : 474,2
STA2.18.5 : 503 F2.18.5 : 574,6
STA1.19.2 : 399,2 F1.19.2 : 478,2
STA2.19.4 : 512,2 F2.19.4 : 583,8
ST1.46.1 : 238,2 C1.46.1 : 248,4
ST2.46.2 : 248,4 C2.46.2 : 257,3
ST4.46.2 : 332,7 C4.46.2 : 343,2
ST1.45.2 : 240,4 C1.45.2 : 251,9
ST2.45.2 : 257,3 C2.45.2 : 265,3
ST4.45.2 : 343,2 C4.45.2 : 354,9
ST1.47.1 : 248,4 C1.47.1 : 265,9
ST2.47.2 : 265,9 C2.47.2 : 281,9
ST1.43.2 : 251,9 C1.43.2 : 283,8
ST3.43.1 : 283,8 C3.43.1 : 406,5
ST1.44.1 : 265,9 C1.44.1 : 298,1
ST3.44.1 : 406,5 C3.44.1 : 557,4
STA1.37.4 : 557,4 F1.37.4 : 656,3
STA2.37.1 : 656,3 F2.37.1 : 717,3
STA1.38.5 : 557,4 F1.38.5 : 656,3
STA2.38.2 : 656,3 F2.38.2 : 717,3
STA1.39.1 : 557,4 F1.39.1 : 656,3
STA2.39.3 : 656,3 F2.39.3 : 717,3
ST1.14.2 : 283,8 C1.14.2 : 291,4
ST1.12.2 : 291,4 C1.12.2 : 300,1
ST1.15.1 : 298,1 C1.15.1 : 307,6
ST2.15.2 : 307,6 C2.15.2 : 319,3
ST4.15.2 : 354,9 C4.15.2 : 364,2
ST1.10.2 : 300,1 C1.10.2 : 309,1
ST2.10.1 : 309,1 C2.10.1 : 320,8
ST4.10.1 : 358 C4.10.1 : 368,5
ST1.13.1 : 307,6 C1.13.1 : 316,8
ST2.13.1 : 321,3 C2.13.1 : 349,1
ST1.11.2 : 309,1 C1.11.2 : 320,7
ST2.11.2 : 320,7 C2.11.2 : 333,2
ST4.11.2 : 364,2 C4.11.2 : 385,5
STA1.10.3 : 474,2 F1.10.3 : 574,8
STA2.10.5 : 574,8 F2.10.5 : 626
STA1.11.6 : 474,2 F1.11.6 : 574,8
STA2.11.4 : 583,8 F2.11.4 : 635
STA1.12.2 : 478,2 F1.12.2 : 578,8
STA2.12.5 : 626 F2.12.5 : 677,2
STA1.13.3 : 574,8 F1.13.3 : 675,4
STA2.13.4 : 675,4 F2.13.4 : 726,6
STA1.14.6 : 574,8 F1.14.6 : 675,4
STA2.14.5 : 677,2 F2.14.5 : 728,4
STA1.15.2 : 578,8 F1.15.2 : 679,4
STA2.15.1 : 717,3 F2.15.1 : 768,5
ST1.4.1 : 316,8 C1.4.1 : 322,6
ST1.6.2 : 320,7 C1.6.2 : 328
ST1.2.1 : 322,6 C1.2.1 : 332,3
ST2.2.2 : 333,2 C2.2.2 : 339,4
ST4.2.1 : 368,5 C4.2.1 : 380,2
ST1.3.2 : 328 C1.3.2 : 337,8
ST2.3.2 : 339,4 C2.3.2 : 351,2
ST4.3.1 : 380,2 C4.3.1 : 390,7
ST1.5.1 : 332,3 C1.5.1 : 347,4
ST2.5.1 : 349,1 C2.5.1 : 377,9
ST1.1.2 : 337,8 C1.1.2 : 353
ST2.1.2 : 353 C2.1.2 : 368
ST4.1.2 : 385,5 C4.1.2 : 404,4
STA1.1.1 : 656,3 F1.1.1 : 759,9
STA2.1.2 : 759,9 F2.1.2 : 807,6
STA1.2.4 : 656,3 F1.2.4 : 759,9
STA2.2.3 : 759,9 F2.2.3 : 807,6
STA1.3.5 : 656,3 F1.3.5 : 759,9
STA2.3.4 : 759,9 F2.3.4 : 807,6
STA1.4.3 : 675,4 F1.4.3 : 779
STA2.4.5 : 779 F2.4.5 : 826,7
STA1.5.6 : 675,4 F1.5.6 : 779
STA2.5.1 : 779 F2.5.1 : 826,7
STA1.6.2 : 679,4 F1.6.2 : 783
STA2.6.2 : 807,6 F2.6.2 : 855,3
ST1.29.1 : 347,4 C1.29.1 : 355
ST1.27.2 : 353 C1.27.2 : 361,5
ST1.30.1 : 355 C1.30.1 : 364,1
ST2.30.2 : 368 C2.30.2 : 379,5
ST4.30.1 : 390,7 C4.30.1 : 400
ST1.25.2 : 361,5 C1.25.2 : 370,5
ST2.25.1 : 378,4 C2.25.1 : 390,1
ST4.25.1 : 400 C4.25.1 : 410,5
ST1.28.1 : 364,1 C1.28.1 : 373
ST2.28.1 : 390,6 C2.28.1 : 418,4
ST1.26.2 : 370,5 C1.26.2 : 382,4
ST2.26.2 : 382,4 C2.26.2 : 394,9
ST4.26.2 : 404,4 C4.26.2 : 425,7
STA1.24.1 : 759,9 F1.24.1 : 870,8
STA2.24.3 : 870,8 F2.24.3 : 920,8
STA1.25.4 : 759,9 F1.25.4 : 870,8
STA2.25.4 : 870,8 F2.25.4 : 920,8
STA1.26.5 : 759,9 F1.26.5 : 870,8
STA2.26.1 : 870,8 F2.26.1 : 920,8
STA1.27.3 : 779 F1.27.3 : 889,9
STA2.27.5 : 889,9 F2.27.5 : 939,9
STA1.28.6 : 779 F1.28.6 : 889,9
STA2.28.2 : 889,9 F2.28.2 : 939,9
STA1.29.2 : 783 F1.29.2 : 893,9
STA2.29.1 : 920,8 F2.29.1 : 970,8

FFS3J5 En Uzun İşlem Zamanı Sonuçları

ST1.3.1 : 0 C1.3.1 : 20
ST2.3.1 : 20 C2.3.1 : 33
ST1.2.2 : 0 C1.2.2 : 8
ST2.2.2 : 8 C2.2.2 : 24
ST1.1.3 : 0 C1.1.3 : 13
STA1.1.1 : 33 F1.1.1 : 75
STA1.2.2 : 33 F1.2.2 : 75
ST1.5.2 : 16 C1.5.2 : 28
ST2.5.3 : 28 C2.5.3 : 34
ST2.4.4 : 0 C2.4.4 : 5
STA1.3.3 : 34 F1.3.3 : 62
STA1.4.4 : 34 F1.4.4 : 62

FFS3J15 En Uzun İşlem Zamanı Sonuçları

ST1.14.1 : 0 C1.14.1 : 15
ST2.14.3 : 15 C2.14.3 : 32
ST1.13.2 : 0 C1.13.2 : 11
ST2.13.1 : 11 C2.13.1 : 21
ST1.15.3 : 0 C1.15.3 : 18
ST2.12.3 : 32 C2.12.3 : 49
STA1.9.1 : 49 F1.9.1 : 89
STA1.10.2 : 49 F1.10.2 : 89
ST1.7.4 : 0 C1.7.4 : 8
ST2.7.3 : 49 C2.7.3 : 67
ST1.6.5 : 0 C1.6.5 : 17
ST2.6.4 : 17 C2.6.4 : 24
ST1.8.4 : 10 C1.8.4 : 20
STA1.5.3 : 67 F1.5.3 : 105
STA1.6.4 : 67 F1.6.4 : 105
ST1.4.2 : 19 C1.4.2 : 37
ST2.4.3 : 67 C2.4.3 : 77
ST2.5.4 : 24 C2.5.4 : 36
STA1.3.5 : 77 F1.3.5 : 118
STA1.4.1 : 89 F1.4.1 : 130
ST1.3.1 : 23 C1.3.1 : 41
ST2.3.2 : 41 C2.3.2 : 58
ST1.1.5 : 25 C1.1.5 : 36
ST2.1.2 : 58 C2.1.2 : 69
ST1.2.3 : 26 C1.2.3 : 34
STA1.1.2 : 89 F1.1.2 : 110
STA1.2.3 : 105 F1.2.3 : 126
ST1.10.4 : 28 C1.10.4 : 42
ST2.10.2 : 69 C2.10.2 : 77
ST2.11.1 : 21 C2.11.1 : 40
ST1.9.3 : 42 C1.9.3 : 55
STA1.7.4 : 105 F1.7.4 : 126
STA1.8.2 : 110 F1.8.2 : 131

FFS3J30 En Uzun İşlem Zamanı Sonuçları

ST1.11.1 : 0 C1.11.1 : 15
ST2.11.2 : 15 C2.11.2 : 33
ST1.9.2 : 0 C1.9.2 : 14
ST2.9.1 : 14 C2.9.1 : 32
ST1.10.3 : 0 C1.10.3 : 15
STA1.7.1 : 33 F1.7.1 : 80
STA1.8.2 : 33 F1.8.2 : 80
ST1.22.2 : 22 C1.22.2 : 37
ST2.22.4 : 37 C2.22.4 : 56
ST1.23.1 : 23 C1.23.1 : 33
ST2.21.2 : 33 C2.21.2 : 42
STA1.15.3 : 56 F1.15.3 : 105
STA1.16.4 : 56 F1.16.4 : 105
ST1.20.3 : 23 C1.20.3 : 42
ST2.20.1 : 42 C2.20.1 : 61
ST1.19.1 : 41 C1.19.1 : 50
ST1.18.3 : 44 C1.18.3 : 52
STA1.13.5 : 61 F1.13.5 : 108
STA1.14.6 : 61 F1.14.6 : 108
ST1.15.2 : 45 C1.15.2 : 60
ST2.15.3 : 60 C2.15.3 : 80
ST1.16.1 : 58 C1.16.1 : 76
ST2.16.3 : 80 C2.16.3 : 92
ST1.17.3 : 60 C1.17.3 : 70
ST2.17.1 : 70 C2.17.1 : 89
STA1.11.1 : 92 F1.11.1 : 113
STA1.12.2 : 92 F1.12.2 : 113
ST1.6.2 : 68 C1.6.2 : 86
ST2.6.2 : 86 C2.6.2 : 97
ST1.8.3 : 78 C1.8.3 : 94
ST2.8.1 : 94 C2.8.1 : 99
ST2.7.3 : 92 C2.7.3 : 110
STA1.5.3 : 110 F1.5.3 : 138
STA1.6.4 : 110 F1.6.4 : 138
ST1.2.1 : 84 C1.2.1 : 101
ST2.2.2 : 101 C2.2.2 : 109
ST2.1.3 : 110 C2.1.3 : 129
ST1.3.2 : 94 C1.3.2 : 99
ST2.3.3 : 129 C2.3.3 : 142
STA1.1.5 : 142 F1.1.5 : 170
STA1.2.6 : 142 F1.2.6 : 170
ST1.4.3 : 102 C1.4.3 : 120
ST2.4.2 : 120 C2.4.2 : 137
ST1.5.2 : 107 C1.5.2 : 119
STA1.3.1 : 137 F1.3.1 : 172
STA1.4.2 : 137 F1.4.2 : 172
ST1.12.1 : 109 C1.12.1 : 123
ST2.12.4 : 123 C2.12.4 : 137
ST1.14.1 : 125 C1.14.1 : 139
ST2.14.2 : 139 C2.14.2 : 151
ST1.13.2 : 127 C1.13.2 : 141
ST2.13.3 : 142 C2.13.3 : 149
STA1.9.3 : 151 F1.9.3 : 172
STA1.10.4 : 151 F1.10.4 : 172
ST1.29.3 : 128 C1.29.3 : 137
ST2.29.4 : 137 C2.29.4 : 148
ST1.27.3 : 145 C1.27.3 : 153
ST2.27.4 : 153 C2.27.4 : 158
ST2.28.1 : 99 C2.28.1 : 105
ST2.30.2 : 151 C2.30.2 : 157
STA1.19.5 : 170 F1.19.5 : 205
STA1.20.6 : 170 F1.20.6 : 205
ST1.26.1 : 147 C1.26.1 : 164
ST2.24.4 : 158 C2.24.4 : 165
ST2.25.2 : 157 C2.25.2 : 163
STA1.17.1 : 172 F1.17.1 : 214
STA1.18.2 : 172 F1.18.2 : 214

FFS6J5 En Uzun İşlem Zamanı Sonuçları

ST1.1.1 : 0 C1.1.1 : 9
ST2.1.3 : 9 C2.1.3 : 28
ST3.1.1 : 28 C3.1.1 : 38
ST4.1.1 : 38 C4.1.1 : 50
ST1.2.2 : 0 C1.2.2 : 11
ST3.2.2 : 11 C3.2.2 : 26
ST4.2.2 : 26 C4.2.2 : 44
ST1.3.3 : 0 C1.3.3 : 8
ST2.3.3 : 28 C2.3.3 : 44
ST4.3.3 : 44 C4.3.3 : 56
STA1.1.1 : 56 F1.1.1 : 105
STA2.1.1 : 105 F2.1.1 : 145
STA1.2.2 : 56 F1.2.2 : 105
STA2.2.2 : 105 F2.2.2 : 145
ST1.4.3 : 8 C1.4.3 : 24
ST2.4.1 : 24 C2.4.1 : 44
ST3.4.2 : 44 C3.4.2 : 59
ST4.4.2 : 59 C4.4.2 : 71
ST1.5.1 : 17 C1.5.1 : 25
ST2.5.2 : 25 C2.5.2 : 41
ST3.5.1 : 41 C3.5.1 : 51
ST4.5.1 : 51 C4.5.1 : 64
STA1.3.3 : 71 F1.3.3 : 115
STA2.3.3 : 115 F2.3.3 : 144
STA1.4.4 : 71 F1.4.4 : 115
STA2.4.4 : 115 F2.4.4 : 144

FFS6J15 En Uzun İşlem Zamanı Sonuçları

ST1.14.1 : 0 C1.14.1 : 13
ST2.14.4 : 13 C2.14.4 : 22
ST3.14.1 : 22 C3.14.1 : 39
ST4.14.1 : 39 C4.14.1 : 53
ST1.15.2 : 0 C1.15.2 : 19
ST2.15.6 : 19 C2.15.6 : 37
ST4.15.2 : 37 C4.15.2 : 48
ST2.12.3 : 0 C2.12.3 : 14
ST3.12.2 : 14 C3.12.2 : 25
ST4.12.3 : 25 C4.12.3 : 43
ST1.13.3 : 0 C1.13.3 : 16
ST2.13.5 : 16 C2.13.5 : 26
ST3.13.3 : 26 C3.13.3 : 33
ST4.13.3 : 43 C4.13.3 : 53
STA1.9.1 : 53 F1.9.1 : 89
STA2.9.1 : 89 F2.9.1 : 131
STA1.10.2 : 53 F1.10.2 : 89
STA2.10.2 : 89 F2.10.2 : 131
ST1.3.4 : 0 C1.3.4 : 9
ST2.3.6 : 37 C2.3.6 : 48
ST3.3.4 : 48 C3.3.4 : 57
ST4.3.2 : 57 C4.3.2 : 71
ST1.2.4 : 11 C1.2.4 : 22
ST2.2.4 : 22 C2.2.4 : 29
ST4.2.1 : 53 C4.2.1 : 72
ST1.1.1 : 21 C1.1.1 : 33
ST2.1.5 : 33 C2.1.5 : 42
ST4.1.3 : 53 C4.1.3 : 66
STA1.1.3 : 72 F1.1.3 : 105
STA2.1.3 : 105 F2.1.3 : 154
STA1.2.4 : 72 F1.2.4 : 105
STA2.2.4 : 105 F2.2.4 : 154
ST1.6.3 : 24 C1.6.3 : 32
ST2.6.1 : 32 C2.6.1 : 51
ST3.6.5 : 51 C3.6.5 : 60
ST4.6.3 : 66 C4.6.3 : 77
ST1.7.2 : 27 C1.7.2 : 47
ST2.7.5 : 47 C2.7.5 : 56
ST3.7.6 : 56 C3.7.6 : 62
ST4.7.2 : 71 C4.7.2 : 83
ST1.8.4 : 30 C1.8.4 : 41
ST3.8.2 : 41 C3.8.2 : 57
ST4.8.1 : 72 C4.8.1 : 82
STA1.5.5 : 83 F1.5.5 : 107
STA2.5.5 : 107 F2.5.5 : 147
STA1.6.6 : 83 F1.6.6 : 107
STA2.6.6 : 107 F2.6.6 : 147
ST2.11.2 : 0 C2.11.2 : 16
ST3.11.3 : 33 C3.11.3 : 49
ST4.11.3 : 77 C4.11.3 : 97
ST1.9.3 : 40 C1.9.3 : 58
ST2.9.3 : 58 C2.9.3 : 64
ST3.9.1 : 64 C3.9.1 : 82
ST4.9.1 : 82 C4.9.1 : 91
ST1.10.1 : 41 C1.10.1 : 49
ST2.10.4 : 49 C2.10.4 : 65
ST3.10.3 : 65 C3.10.3 : 72
ST4.10.2 : 83 C4.10.2 : 88
STA1.7.1 : 97 F1.7.1 : 118
STA2.7.1 : 131 F2.7.1 : 165
STA1.8.2 : 97 F1.8.2 : 118
STA2.8.2 : 131 F2.8.2 : 165
ST1.4.4 : 49 C1.4.4 : 65
ST2.4.5 : 65 C2.4.5 : 84
ST4.4.2 : 88 C4.4.2 : 104
ST3.5.2 : 57 C3.5.2 : 67
ST4.5.1 : 91 C4.5.1 : 99
STA1.3.3 : 105 F1.3.3 : 155
STA2.3.5 : 155 F2.3.5 : 184
STA1.4.4 : 105 F1.4.4 : 155
STA2.4.6 : 155 F2.4.6 : 184

FFS6J30 En Uzun İşlem Zamanı Sonuçları

ST1.10.1 : 0 C1.10.1 : 20
ST2.10.3 : 20 C2.10.3 : 35
ST3.10.1 : 35 C3.10.1 : 45
ST4.10.1 : 45 C4.10.1 : 59
ST1.9.2 : 0 C1.9.2 : 19
ST2.9.3 : 35 C2.9.3 : 55
ST4.9.2 : 55 C4.9.2 : 74
ST1.11.3 : 0 C1.11.3 : 12
ST2.11.3 : 55 C2.11.3 : 74
ST3.11.2 : 74 C3.11.2 : 81
ST4.11.3 : 81 C4.11.3 : 101
STA1.7.1 : 101 F1.7.1 : 134
STA2.7.1 : 134 F2.7.1 : 172
STA1.8.2 : 101 F1.8.2 : 134
STA2.8.2 : 134 F2.8.2 : 172
ST1.23.3 : 20 C1.23.3 : 39
ST2.23.2 : 39 C2.23.2 : 49
ST3.23.3 : 49 C3.23.3 : 63
ST4.23.4 : 63 C4.23.4 : 76
ST1.22.2 : 27 C1.22.2 : 39
ST2.22.2 : 49 C2.22.2 : 67
ST3.22.4 : 67 C3.22.4 : 81
ST4.22.1 : 81 C4.22.1 : 88
ST2.21.1 : 0 C2.21.1 : 11
ST3.21.1 : 45 C3.21.1 : 55
ST4.21.2 : 74 C4.21.2 : 82
STA1.15.3 : 88 F1.15.3 : 122
STA2.15.3 : 122 F2.15.3 : 170
STA1.16.4 : 88 F1.16.4 : 122
STA2.16.4 : 122 F2.16.4 : 170
ST1.17.1 : 28 C1.17.1 : 36
ST2.17.2 : 67 C2.17.2 : 79
ST3.17.1 : 79 C3.17.1 : 91
ST4.17.4 : 91 C4.17.4 : 104
ST1.15.1 : 38 C1.15.1 : 45
ST2.15.2 : 79 C2.15.2 : 95
ST3.15.3 : 95 C3.15.3 : 102
ST4.15.2 : 102 C4.15.2 : 113
ST1.16.1 : 47 C1.16.1 : 54
ST3.16.2 : 81 C3.16.2 : 89
ST4.16.1 : 89 C4.16.1 : 95
STA1.11.5 : 113 F1.11.5 : 162
STA2.11.5 : 162 F2.11.5 : 199
STA1.12.6 : 113 F1.12.6 : 162
STA2.12.6 : 162 F2.12.6 : 199
ST1.2.2 : 47 C1.2.2 : 56
ST2.2.2 : 95 C2.2.2 : 110
ST3.2.4 : 110 C3.2.4 : 128
ST4.2.1 : 128 C4.2.1 : 142
ST1.3.3 : 47 C1.3.3 : 55
ST2.3.3 : 74 C2.3.3 : 88
ST3.3.2 : 89 C3.3.2 : 102
ST4.3.3 : 102 C4.3.3 : 112
ST2.1.2 : 110 C2.1.2 : 130
ST4.1.4 : 130 C4.1.4 : 144
STA1.1.3 : 144 F1.1.3 : 179
STA2.1.3 : 179 F2.1.3 : 209
STA1.2.4 : 144 F1.2.4 : 179
STA2.2.4 : 179 F2.2.4 : 209
ST1.29.1 : 62 C1.29.1 : 69
ST2.29.3 : 88 C2.29.3 : 99
ST3.29.1 : 99 C3.29.1 : 113
ST4.29.3 : 113 C4.29.3 : 126
ST1.27.3 : 63 C1.27.3 : 79
ST2.27.1 : 79 C2.27.1 : 88
ST4.27.2 : 113 C4.27.2 : 132
ST2.28.2 : 130 C2.28.2 : 135
ST3.28.2 : 135 C3.28.2 : 149
ST4.28.3 : 149 C4.28.3 : 156
ST2.30.3 : 99 C2.30.3 : 109
ST3.30.3 : 109 C3.30.3 : 115
ST4.30.2 : 132 C4.30.2 : 142
STA1.19.1 : 156 F1.19.1 : 186
STA2.19.1 : 186 F2.19.1 : 215
STA1.20.2 : 156 F1.20.2 : 186
STA2.20.2 : 186 F2.20.2 : 215
ST1.8.2 : 64 C1.8.2 : 81
ST2.8.1 : 88 C2.8.1 : 99
ST3.8.1 : 113 C3.8.1 : 119
ST4.8.1 : 142 C4.8.1 : 150
ST1.6.1 : 77 C1.6.1 : 94
ST2.6.3 : 109 C2.6.3 : 124
ST4.6.2 : 142 C4.6.2 : 149
ST2.7.1 : 99 C2.7.1 : 108
ST3.7.3 : 115 C3.7.3 : 127
ST4.7.4 : 144 C4.7.4 : 150
STA1.5.5 : 162 F1.5.5 : 203
STA2.5.5 : 203 F2.5.5 : 234
STA1.6.6 : 162 F1.6.6 : 203
STA2.6.6 : 203 F2.6.6 : 234
ST1.14.3 : 87 C1.14.3 : 105
ST2.14.1 : 108 C2.14.1 : 122
ST3.14.1 : 122 C3.14.1 : 132
ST1.13.2 : 89 C1.13.2 : 98
ST2.13.1 : 122 C2.13.1 : 135
ST4.13.2 : 149 C4.13.2 : 162
ST1.12.2 : 100 C1.12.2 : 106
ST2.12.2 : 135 C2.12.2 : 149
ST3.12.3 : 149 C3.12.3 : 157
ST4.12.1 : 157 C4.12.1 : 162
STA1.9.3 : 179 F1.9.3 : 202
STA2.9.3 : 209 F2.9.3 : 257
STA1.10.4 : 179 F1.10.4 : 202
STA2.10.4 : 209 F2.10.4 : 257
ST1.19.1 : 102 C1.19.1 : 121
ST2.19.2 : 149 C2.19.2 : 167
ST3.19.4 : 167 C3.19.4 : 187
ST4.19.4 : 187 C4.19.4 : 199
ST1.20.3 : 113 C1.20.3 : 129
ST2.20.3 : 129 C2.20.3 : 141
ST3.20.1 : 141 C3.20.1 : 146
ST4.20.3 : 156 C4.20.3 : 163
ST1.18.2 : 114 C1.18.2 : 129
ST3.18.1 : 146 C3.18.1 : 153
ST4.18.1 : 162 C4.18.1 : 178
STA1.13.1 : 199 F1.13.1 : 224
STA2.13.1 : 224 F2.13.1 : 245
STA1.14.2 : 199 F1.14.2 : 224
STA2.14.2 : 224 F2.14.2 : 245
ST2.25.1 : 135 C2.25.1 : 148
ST3.25.2 : 149 C3.25.2 : 167
ST4.25.2 : 167 C4.25.2 : 186
ST2.24.1 : 148 C2.24.1 : 167
ST3.24.1 : 167 C3.24.1 : 176
ST4.24.3 : 176 C4.24.3 : 188
ST1.26.1 : 129 C1.26.1 : 138
ST2.26.2 : 167 C2.26.2 : 177
ST4.26.1 : 178 C4.26.1 : 192
STA1.17.3 : 202 F1.17.3 : 233
STA2.17.5 : 234 F2.17.5 : 255
STA1.18.4 : 202 F1.18.4 : 233
STA2.18.6 : 234 F2.18.6 : 255
ST1.5.2 : 137 C1.5.2 : 145
ST2.5.3 : 145 C2.5.3 : 163
ST3.5.3 : 163 C3.5.3 : 168
ST4.5.2 : 186 C4.5.2 : 200
ST1.4.3 : 137 C1.4.3 : 149
ST2.4.3 : 163 C2.4.3 : 170
ST3.4.2 : 170 C3.4.2 : 186
ST4.4.3 : 188 C4.4.3 : 193
STA1.3.5 : 203 F1.3.5 : 225
STA2.3.1 : 245 F2.3.1 : 278
STA1.4.6 : 203 F1.4.6 : 225
STA2.4.2 : 245 F2.4.2 : 278

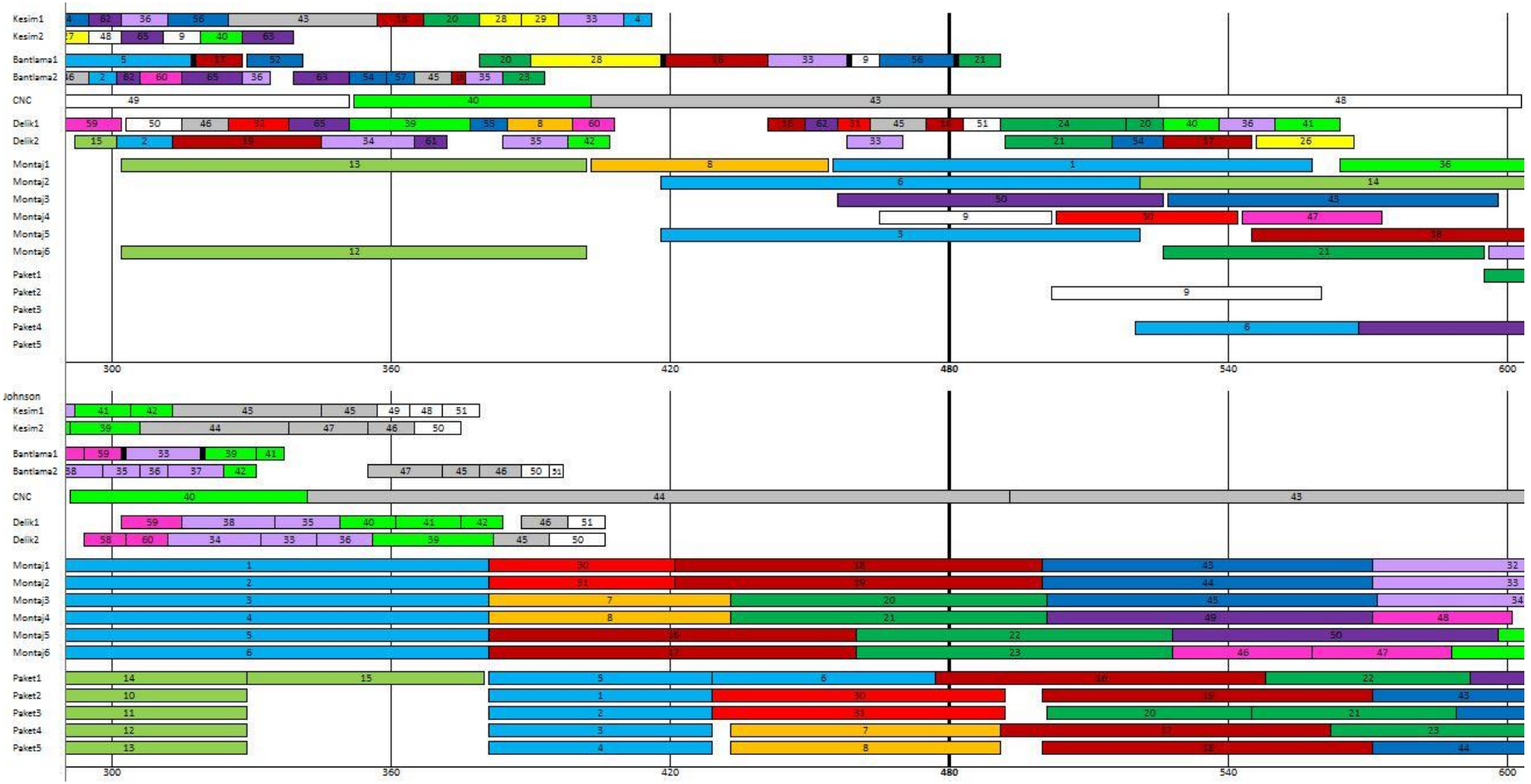
FSS6J66 En Uzun İşlem Zamanı Sonuçları

ST1.26.1 : 0 C1.26.1 : 11,9
ST2.26.2 : 11,9 C2.26.2 : 24,4
ST4.26.1 : 24,4 C4.26.1 : 45,7
ST1.28.2 : 0 C1.28.2 : 8,9
ST2.28.1 : 8,9 C2.28.1 : 36,7
ST1.25.2 : 8,9 C1.25.2 : 17,9
ST2.25.1 : 37,2 C2.25.1 : 48,9
ST4.25.2 : 48,9 C4.25.2 : 59,4
ST1.30.1 : 11,9 C1.30.1 : 21
ST2.30.2 : 24,4 C2.30.2 : 35,9
ST4.30.1 : 45,7 C4.30.1 : 55
ST1.27.2 : 17,9 C1.27.2 : 26,4
ST1.29.1 : 21 C1.29.1 : 28,6
STA1.24.1 : 59,4 F1.24.1 : 170,3
STA2.24.1 : 170,3 F2.24.1 : 220,3
STA1.25.2 : 59,4 F1.25.2 : 170,3
STA2.25.2 : 170,3 F2.25.2 : 220,3
STA1.26.3 : 59,4 F1.26.3 : 170,3
STA2.26.3 : 170,3 F2.26.3 : 220,3
STA1.27.4 : 59,4 F1.27.4 : 170,3
STA2.27.4 : 170,3 F2.27.4 : 220,3
STA1.28.5 : 59,4 F1.28.5 : 170,3
STA2.28.5 : 170,3 F2.28.5 : 220,3
STA1.29.6 : 59,4 F1.29.6 : 170,3
STA2.29.1 : 220,3 F2.29.1 : 270,3
ST1.1.2 : 26,4 C1.1.2 : 41,6
ST2.1.2 : 41,6 C2.1.2 : 56,6
ST4.1.1 : 56,6 C4.1.1 : 75,5
ST1.5.1 : 28,6 C1.5.1 : 43,7
ST2.5.1 : 49,4 C2.5.1 : 78,2
ST1.3.2 : 41,6 C1.3.2 : 51,4
ST2.3.2 : 56,6 C2.3.2 : 68,4
ST4.3.2 : 68,4 C4.3.2 : 78,9
ST1.2.1 : 43,7 C1.2.1 : 53,4
ST2.2.2 : 68,4 C2.2.2 : 74,6
ST4.2.1 : 75,5 C4.2.1 : 87,2
ST1.6.2 : 51,4 C1.6.2 : 58,7
ST1.4.1 : 53,4 C1.4.1 : 59,2
STA1.1.1 : 170,3 F1.1.1 : 273,9
STA2.1.2 : 273,9 F2.1.2 : 321,6
STA1.2.2 : 170,3 F1.2.2 : 273,9
STA2.2.3 : 273,9 F2.2.3 : 321,6
STA1.3.3 : 170,3 F1.3.3 : 273,9
STA2.3.4 : 273,9 F2.3.4 : 321,6
STA1.4.4 : 170,3 F1.4.4 : 273,9

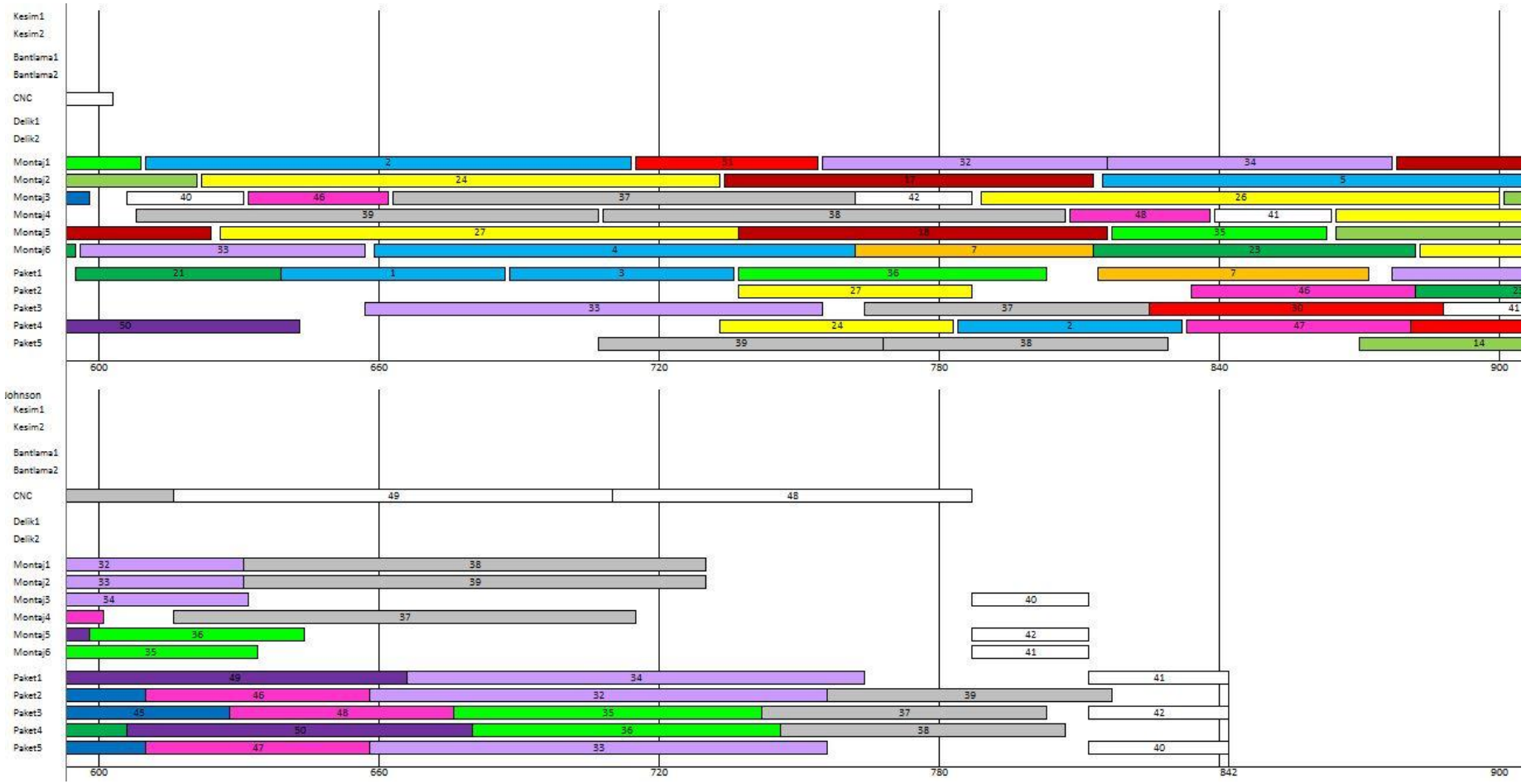
STA2.4.5 : 273,9 F2.4.5 : 321,6
STA1.5.5 : 170,3 F1.5.5 : 273,9
STA2.5.1 : 273,9 F2.5.1 : 321,6
STA1.6.6 : 170,3 F1.6.6 : 273,9
STA2.6.1 : 321,6 F2.6.1 : 369,3
ST1.11.2 : 58,7 C1.11.2 : 70,3
ST2.11.2 : 74,6 C2.11.2 : 87,1
ST4.11.2 : 87,1 C4.11.2 : 108,4
ST1.13.1 : 59,2 C1.13.1 : 68,4
ST2.13.1 : 78,2 C2.13.1 : 106
ST1.10.1 : 68,4 C1.10.1 : 77,4
ST2.10.1 : 106,5 C2.10.1 : 118,2
ST4.10.1 : 118,2 C4.10.1 : 128,7
ST1.15.2 : 70,3 C1.15.2 : 79,8
ST2.15.2 : 87,1 C2.15.2 : 98,8
ST4.15.2 : 108,4 C4.15.2 : 117,7
ST1.12.1 : 77,4 C1.12.1 : 86,1
ST1.14.2 : 79,8 C1.14.2 : 87,4
STA1.10.1 : 273,9 F1.10.1 : 374,5
STA2.10.2 : 374,5 F2.10.2 : 425,7
STA1.11.2 : 273,9 F1.11.2 : 374,5
STA2.11.3 : 374,5 F2.11.3 : 425,7
STA1.12.3 : 273,9 F1.12.3 : 374,5
STA2.12.4 : 374,5 F2.12.4 : 425,7
STA1.13.4 : 273,9 F1.13.4 : 374,5
STA2.13.5 : 374,5 F2.13.5 : 425,7
STA1.14.5 : 273,9 F1.14.5 : 374,5
STA2.14.1 : 374,5 F2.14.1 : 425,7
STA1.15.6 : 273,9 F1.15.6 : 374,5
STA2.15.1 : 425,7 F2.15.1 : 476,9
ST1.44.1 : 86,1 C1.44.1 : 118,3
ST3.44.1 : 118,3 C3.44.1 : 269,2
ST1.43.2 : 87,4 C1.43.2 : 119,3
ST3.43.1 : 269,2 C3.43.1 : 391,9
ST1.47.1 : 118,3 C1.47.1 : 135,8
ST2.47.2 : 135,8 C2.47.2 : 151,8
ST1.45.2 : 119,3 C1.45.2 : 130,8
ST2.45.2 : 151,8 C2.45.2 : 159,8
ST4.45.2 : 159,8 C4.45.2 : 171,5
ST1.46.2 : 130,8 C1.46.2 : 141
ST2.46.2 : 159,8 C2.46.2 : 168,7
ST4.46.1 : 168,7 C4.46.1 : 179,2
STA1.37.1 : 391,9 F1.37.1 : 490,8
STA2.37.2 : 490,8 F2.37.2 : 551,8
STA1.38.2 : 391,9 F1.38.2 : 490,8
STA2.38.3 : 490,8 F2.38.3 : 551,8
STA1.39.3 : 391,9 F1.39.3 : 490,8
STA2.39.4 : 490,8 F2.39.4 : 551,8
ST1.19.1 : 135,8 C1.19.1 : 153,1
ST2.19.2 : 168,7 C2.19.2 : 177,7
ST4.19.2 : 177,7 C4.19.2 : 209,8
ST1.16.2 : 141 C1.16.2 : 161,7
ST2.16.1 : 161,7 C2.16.1 : 183,6
ST4.16.1 : 183,6 C4.16.1 : 191,7
ST1.17.1 : 153,1 C1.17.1 : 168,3
ST2.17.1 : 183,6 C2.17.1 : 194,3
ST4.17.1 : 194,3 C4.17.1 : 213,6
ST1.18.2 : 161,7 C1.18.2 : 171,2
ST2.18.2 : 177,7 C2.18.2 : 180,7
ST4.18.2 : 209,8 C4.18.2 : 217,9
STA1.16.4 : 374,5 F1.16.4 : 453,5
STA2.16.5 : 453,5 F2.16.5 : 525,1
STA1.17.5 : 374,5 F1.17.5 : 453,5
STA2.17.1 : 476,9 F2.17.1 : 548,5
STA1.18.6 : 374,5 F1.18.6 : 453,5
STA2.18.5 : 525,1 F2.18.5 : 596,7
STA1.19.4 : 453,5 F1.19.4 : 532,5
STA2.19.1 : 548,5 F2.19.1 : 620,1
ST1.34.1 : 168,3 C1.34.1 : 192,9
ST2.34.2 : 192,9 C2.34.2 : 209,1
ST4.34.1 : 213,6 C4.34.1 : 233,7
ST1.38.2 : 171,2 C1.38.2 : 182,4
ST2.38.2 : 209,1 C2.38.2 : 223
ST4.38.2 : 223 C4.38.2 : 243,1
ST1.33.2 : 182,4 C1.33.2 : 196,9
ST2.33.1 : 196,9 C2.33.1 : 213,7
ST4.33.1 : 233,7 C4.33.1 : 246
ST1.35.1 : 192,9 C1.35.1 : 205,8
ST2.35.2 : 223 C2.35.2 : 231,1
ST4.35.2 : 243,1 C4.35.2 : 257,2
ST1.36.2 : 196,9 C1.36.2 : 207,2
ST2.36.2 : 231,1 C2.36.2 : 237,5
ST4.36.1 : 246 C4.36.1 : 257,7
ST1.37.1 : 205,8 C1.37.1 : 216,6
ST2.37.2 : 237,5 C2.37.2 : 249,5
STA1.32.5 : 453,5 F1.32.5 : 513,9
STA2.32.2 : 551,8 F2.32.2 : 650
STA1.33.6 : 453,5 F1.33.6 : 513,9
STA2.33.3 : 551,8 F2.33.3 : 650
STA1.34.1 : 490,8 F1.34.1 : 551,2
STA2.34.4 : 551,8 F2.34.4 : 650
ST1.21.2 : 207,2 C1.21.2 : 219,7
ST2.21.1 : 219,7 C2.21.1 : 229,3
ST4.21.2 : 257,2 C4.21.2 : 280,5
ST1.24.1 : 216,6 C1.24.1 : 225,4
ST2.24.1 : 229,3 C2.24.1 : 237,3
ST4.24.1 : 257,7 C4.24.1 : 285
ST1.20.2 : 219,7 C1.20.2 : 231,6
ST2.20.1 : 237,3 C2.20.1 : 248,3
ST4.20.2 : 280,5 C4.20.2 : 288,6
ST1.22.1 : 225,4 C1.22.1 : 233,4
ST2.22.2 : 249,5 C2.22.2 : 252,2
ST4.22.1 : 285 C4.22.1 : 294,3
ST1.23.2 : 231,6 C1.23.2 : 239,7
ST2.23.2 : 252,2 C2.23.2 : 261,5
STA1.20.2 : 490,8 F1.20.2 : 559,4
STA2.20.5 : 596,7 F2.20.5 : 640,8
STA1.21.3 : 490,8 F1.21.3 : 559,4
STA2.21.1 : 620,1 F2.21.1 : 664,2
STA1.22.5 : 513,9 F1.22.5 : 582,5
STA2.22.5 : 640,8 F2.22.5 : 684,9
STA1.23.6 : 513,9 F1.23.6 : 582,5
STA2.23.2 : 650 F2.23.2 : 694,1
ST1.56.1 : 233,4 C1.56.1 : 246,9
ST2.56.1 : 248,8 C2.56.1 : 264,8
ST1.54.2 : 239,7 C1.54.2 : 249,2
ST2.54.2 : 261,5 C2.54.2 : 269,5
ST4.54.2 : 288,6 C4.54.2 : 300,3
ST1.53.1 : 246,9 C1.53.1 : 256
ST2.53.1 : 264,8 C2.53.1 : 276,8
ST4.53.1 : 294,3 C4.53.1 : 301,8
ST1.52.2 : 249,2 C1.52.2 : 258,1
ST2.52.1 : 276,8 C2.52.1 : 288,8
ST1.55.1 : 256 C1.55.1 : 264,3
ST2.55.2 : 269,5 C2.55.2 : 273,6
ST4.55.2 : 300,3 C4.55.2 : 308,4
ST1.57.2 : 258,1 C1.57.2 : 266,3
ST2.57.2 : 273,6 C2.57.2 : 280,2
STA1.43.4 : 532,5 F1.43.4 : 603,7
STA2.43.3 : 650 F2.43.3 : 689,1
STA1.44.1 : 551,2 F1.44.1 : 622,4
STA2.44.4 : 650 F2.44.4 : 689,1
STA1.45.2 : 559,4 F1.45.2 : 630,6
STA2.45.1 : 664,2 F2.45.1 : 703,3
ST1.65.1 : 264,3 C1.65.1 : 272,6
ST2.65.2 : 280,2 C2.65.2 : 293,3
ST4.65.1 : 301,8 C4.65.1 : 315,1
ST1.64.2 : 266,3 C1.64.2 : 274,9
ST2.64.1 : 288,8 C2.64.1 : 298
ST4.64.2 : 308,4 C4.64.2 : 315,7
ST1.63.1 : 272,6 C1.63.1 : 283,8
ST2.63.2 : 293,3 C2.63.2 : 304,7
ST1.61.2 : 274,9 C1.61.2 : 283,3
ST2.61.1 : 298,5 C2.61.1 : 303,3
ST4.61.1 : 315,1 C4.61.1 : 321,6
ST1.62.2 : 283,3 C1.62.2 : 290,8
ST2.62.2 : 304,7 C2.62.2 : 309,4
ST4.62.2 : 315,7 C4.62.2 : 322,6
ST1.66.1 : 283,8 C1.66.1 : 292,2
STA1.49.3 : 559,4 F1.49.3 : 629
STA2.49.5 : 684,9 F2.49.5 : 759,4
STA1.50.5 : 582,5 F1.50.5 : 652,1
STA2.50.3 : 689,1 F2.50.3 : 763,6
ST1.40.2 : 290,8 C1.40.2 : 300,6
ST3.40.1 : 391,9 C3.40.1 : 443,1
ST4.40.1 : 443,1 C4.40.1 : 454,8
ST1.39.1 : 292,2 C1.39.1 : 307,3
ST2.39.1 : 307,3 C2.39.1 : 318,8
ST4.39.2 : 322,6 C4.39.2 : 348,7
ST1.41.2 : 300,6 C1.41.2 : 312,6
ST2.41.1 : 318,8 C2.41.1 : 325,2
ST4.41.2 : 348,7 C4.41.2 : 362,6
ST1.42.1 : 307,3 C1.42.1 : 316,8
ST2.42.2 : 316,8 C2.42.2 : 323,7
ST4.42.2 : 362,6 C4.42.2 : 371,6
STA1.35.6 : 582,5 F1.35.6 : 628
STA2.35.4 : 689,1 F2.35.4 : 755
STA1.36.4 : 603,7 F1.36.4 : 649,2
STA2.36.2 : 694,1 F2.36.2 : 760
ST1.49.2 : 312,6 C1.49.2 : 319,7
ST3.49.1 : 443,1 C3.49.1 : 536,7
ST1.48.1 : 316,8 C1.48.1 : 323,7
ST3.48.1 : 536,7 C3.48.1 : 614,4
ST1.50.2 : 319,7 C1.50.2 : 330,1
ST2.50.2 : 330,1 C2.50.2 : 335,6
ST4.50.2 : 371,6 C4.50.2 : 383,9
ST1.51.1 : 323,7 C1.51.1 : 331,3
ST2.51.2 : 335,6 C2.51.2 : 338,7
ST4.51.2 : 383,9 C4.51.2 : 392
STA1.40.1 : 622,4 F1.40.1 : 647,4
STA2.40.1 : 703,3 F2.40.1 : 733,5
STA1.41.6 : 628 F1.41.6 : 653
STA2.41.1 : 733,5 F2.41.1 : 763,7
STA1.42.3 : 629 F1.42.3 : 654
STA2.42.4 : 755 F2.42.4 : 785,2
ST1.58.2 : 330,1 C1.58.2 : 352,3
ST2.58.1 : 352,3 C2.58.1 : 367,5
ST4.58.2 : 392 C4.58.2 : 401,3
ST1.59.1 : 331,3 C1.59.1 : 343,1
ST2.59.1 : 367,5 C2.59.1 : 375,5
ST4.59.2 : 401,3 C4.59.2 : 414,2
ST1.60.1 : 343,1 C1.60.1 : 352,8
ST2.60.2 : 352,8 C2.60.2 : 361,4
ST4.60.2 : 414,2 C4.60.2 : 422,9
STA1.46.2 : 630,6 F1.46.2 : 660,2
STA2.46.5 : 759,4 F2.46.5 : 807,5
STA1.47.1 : 647,4 F1.47.1 : 677
STA2.47.2 : 760 F2.47.2 : 808,1
STA1.48.4 : 649,2 F1.48.4 : 678,8
STA2.48.3 : 763,6 F2.48.3 : 811,7
ST1.8.2 : 352,3 C1.8.2 : 362,8
ST2.8.1 : 375,5 C2.8.1 : 390
ST4.8.2 : 422,9 C4.8.2 : 436,6
ST1.7.1 : 352,8 C1.7.1 : 364,8
ST2.7.1 : 390 C2.7.1 : 394,3
ST4.7.2 : 436,6 C4.7.2 : 444,3
STA1.7.5 : 652,1 F1.7.5 : 703,1
STA2.7.1 : 763,7 F2.7.1 : 822,1
STA1.8.6 : 653 F1.8.6 : 704
STA2.8.4 : 785,2 F2.8.4 : 843,6
ST1.32.2 : 362,8 C1.32.2 : 374,6
ST2.32.1 : 394,3 C2.32.1 : 401,8
ST4.32.2 : 444,3 C4.32.2 : 457,2
ST1.31.1 : 364,8 C1.31.1 : 373,1
ST2.31.1 : 401,8 C2.31.1 : 410,9
ST4.31.1 : 454,8 C4.31.1 : 462,1
STA1.30.3 : 654 F1.30.3 : 693,2
STA2.30.5 : 807,5 F2.30.5 : 870,3
STA1.31.2 : 660,2 F1.31.2 : 699,4
STA2.31.2 : 808,1 F2.31.2 : 870,9
ST1.9.1 : 373,1 C1.9.1 : 380,8
ST2.9.1 : 410,9 C2.9.1 : 417
STA1.9.1 : 677 F1.9.1 : 714,2
STA2.9.3 : 811,7 F2.9.3 : 870,3

EK-6: FFS6J66 PROBLEMİNİN MATEMATİKSEL (C_{MAX}), UJK, EABK VE PEABK YÖNTEMLERİNE GÖRE ÇÖZÜMLERİNİN GANNT ŞEMALARI

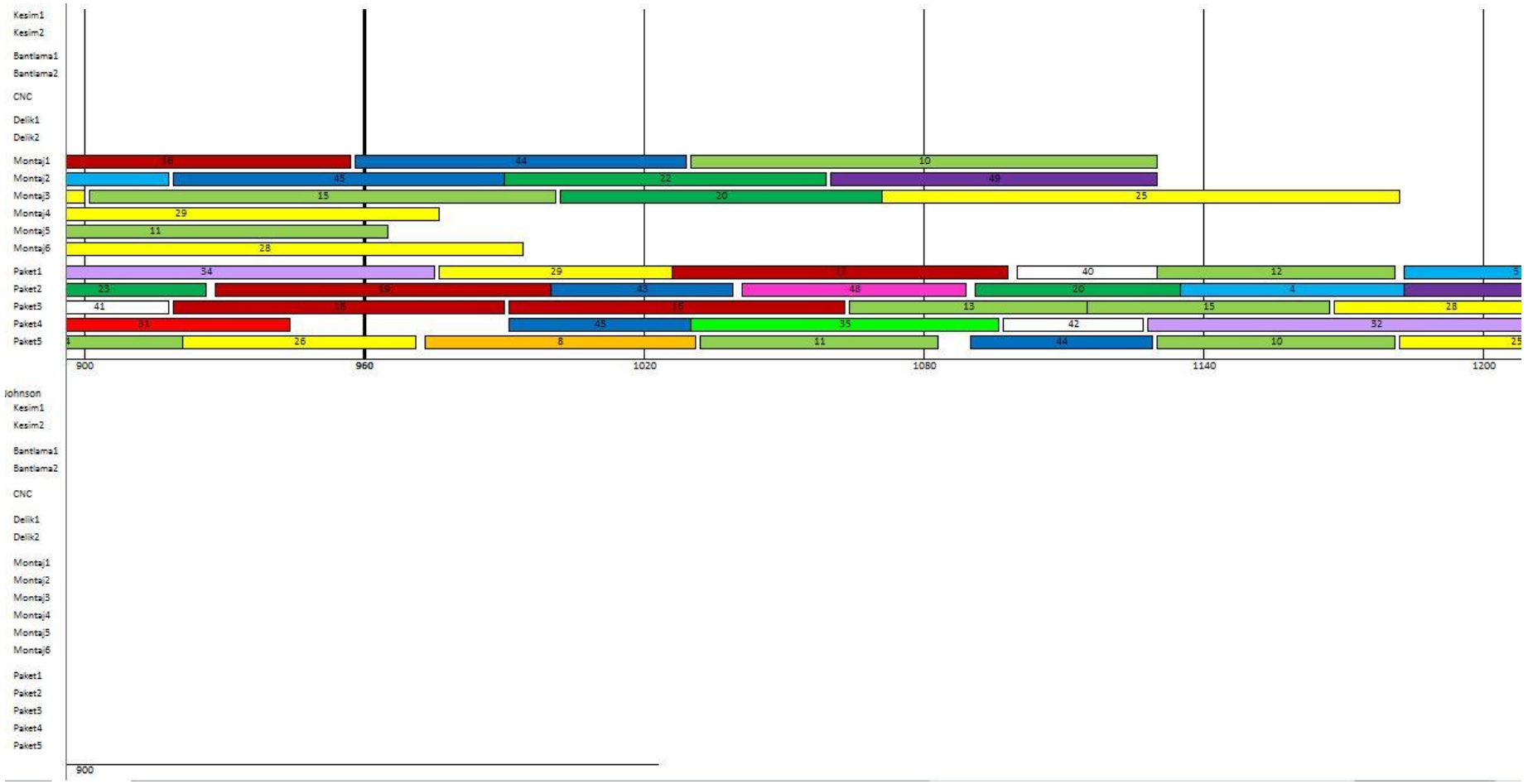
Şekil 29: FFS6J66 Probleminin Matematiksel Model (C_{max}) ve UJK yöntemine göre çözümlerinin Gannt Şemaları (0-300 dk.)



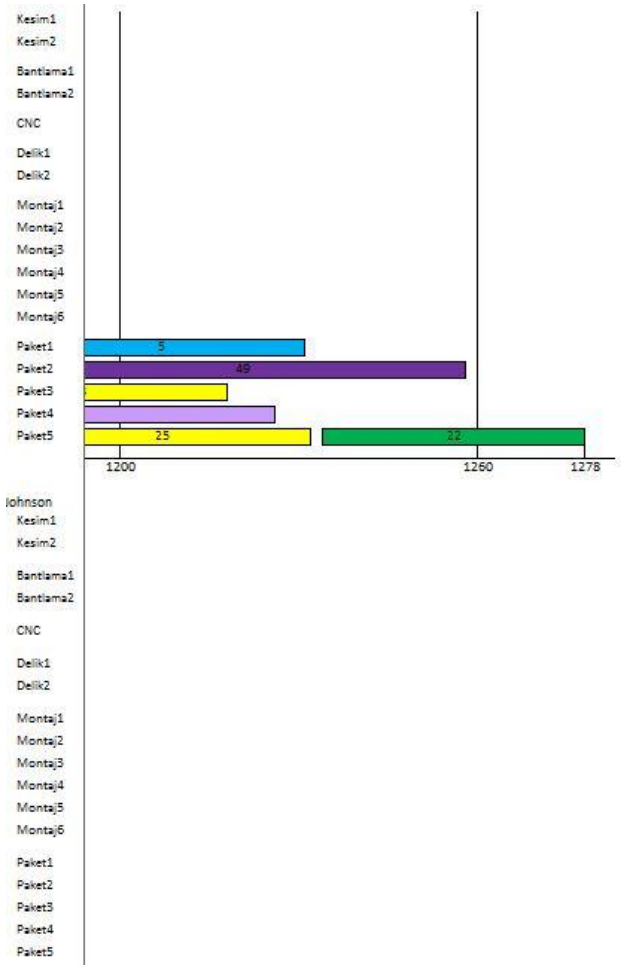
Şekil 30: FFS6J66 Probleminin Matematiksel Model (C_{max}) ve UJK yöntemine göre çözümlerinin Gannt Şemaları (300-600 dk.)



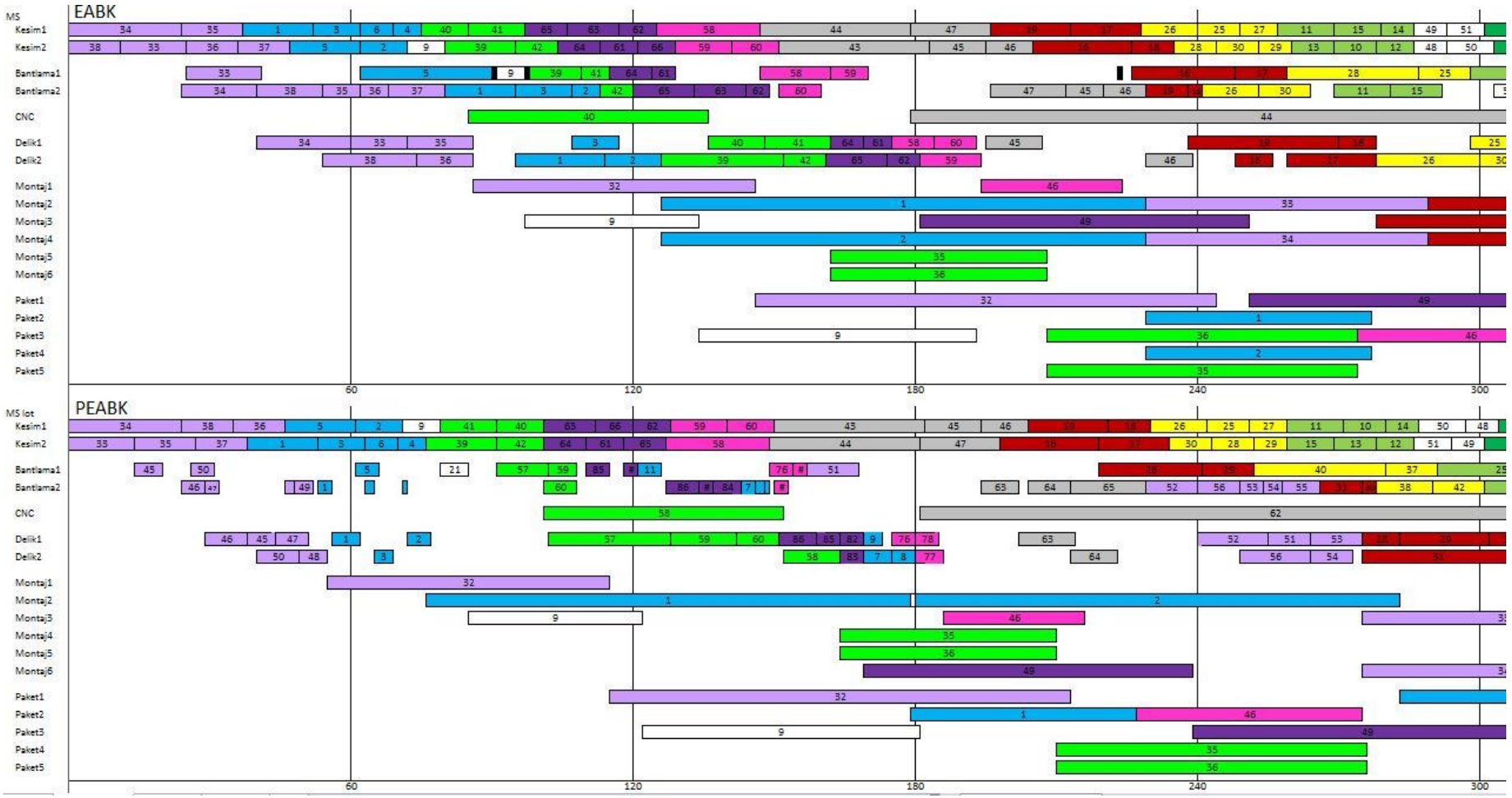
Şekil 31: FFS6J66 Probleminin Matematiksel Model (C_{max}) ve UJK yöntemine göre çözümlerinin Gannt Şemaları (600-900 dk.)



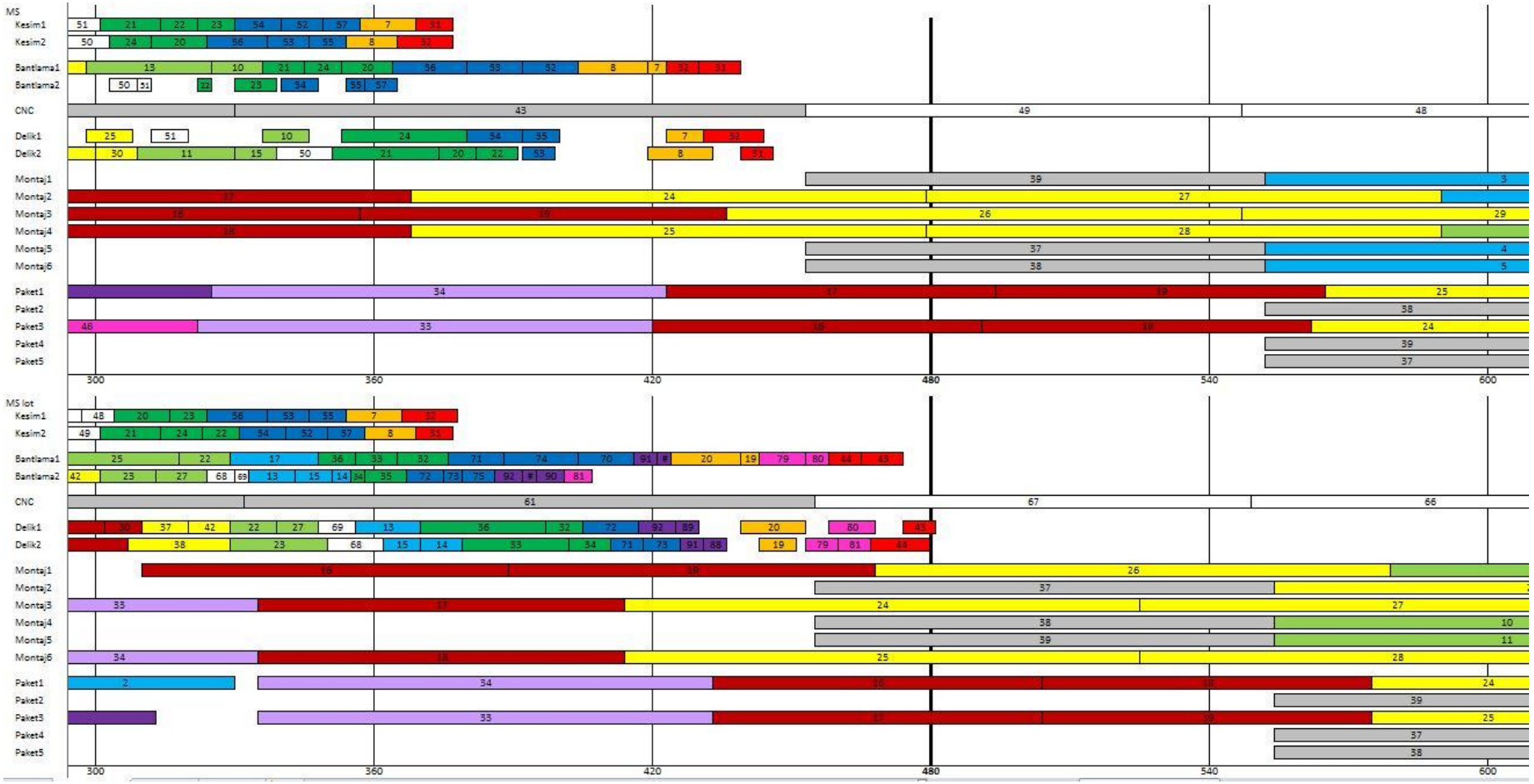
Şekil 32: FFS6J66 Probleminin Matematiksel Model (C_{max}) ve UJK yöntemine göre çözümlerinin Gannt Şemaları (900-1200 dk.)



Şekil 33: FFS6J66 Probleminin Matematiksel Model (C_{max}) ve UJK yöntemine göre çözümlerinin Gannt Şemaları (1200-1278 dk.)

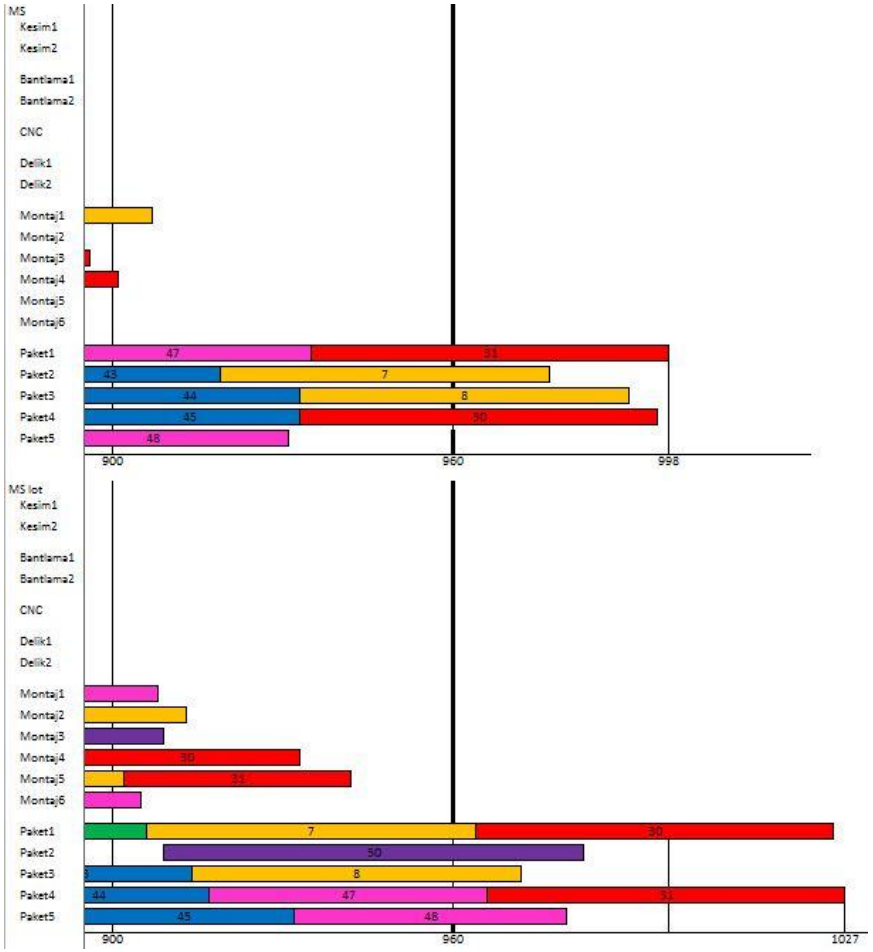


Şekil 34: FFS6J66 Probleminin EABK ve PEABK yöntemlerine göre çözümlerinin Gannt Şemaları (0-300 dk.)



Şekil 35: FFS6J66 Probleminin EABK ve PEABK yöntemlerine göre çözümlerinin Gannt Şemaları (300-600 dk.)

Şekil 36: FFS6J66 Probleminin EABK ve PEABK yöntemlerine göre çözümlerinin Gannt Şemaları (600-900 dk.)



Şekil 37: FFS6J66 Probleminin EABK ve PEABK yöntemlerine göre çözümlerinin Gannt Şemaları (900-1027 dk.)

KAYNAKLAR

Adler, L., N. Fraiman, E. Kobacker, M. Pinedo, J.C. Plotnicoff and T.P. Wu. (1993). "BPSS: A scheduling support system for packing industry", *Operations Research*, 41(4), 641-648.

Agnetis, A., A. Pacifici, F. Rossi, M. Lucertini, S. Nicoletti, F. Nicolo, G. Oriolo, D. Pacciarelli and E. Pesaro. (1997). "Scheduling of flexible flow lines in an automobile assembly plant", *European Journal of Operational Research*, 97, 348-362.

Aker, S.B. (1956). "A graphical approach to production scheduling problems", *Operations Research*, 4(2), 244-245.

Akrami, B., B. Karimi and S.M.M. Hosseini. (2006). "Two metaheuristic methods for the common cycle economic lot sizing and scheduling in flexible flow shops with limited intermediate buffers: The finite horizon case", *Applied Mathematics & Computation*, 183, 634-645.

Alisantoso, D., L.P. Khoo and P.Y. Jiang. (2003). "An immune algorithm approach to the scheduling of a flexible PCB flow shop", *International Journal of Advanced Manufacturing Technologies*, 22, 819-827.

Allaoui, H. and A. Artiba. (2006). "Scheduling two-stage hybrid flow shop with availability constraints", *Computers & Operations Research*, 33, 1399-1419.

Amin-Naseri, M.R. and M.A. Beheshti-Nia. (2009). "Hybrid flow shop scheduling with parallel batching", *International Journal of Production Economics*, 117, 185-196.

Artigues, C. and D. Feiller. (2008). "A branch and bound method for the job-shop problem with sequence-dependent setup times", *Annals of Operations Research*, 159, 135-159.

Azizoğlu, M., E. Çakmak and S. Kondakci. (2001). "A flexible flowshop problem with total flow time minimization", *European Journal of Operational Research*, 132, 528-538.

Baker, K.R. and D. Trietsch. (2009). *Principles of Sequencing and Scheduling*. New Jersey: Wiley.

Behnamian, J., S.M.T. Fatemi Ghomi and M. Zandieh. (2009). "A multi-phase covering Pareto-optimal front method to multi-objective scheduling in a realistic hybrid flowshop using a hybrid metaheuristic", *Expert Systems with Applications*, 36, 11057-11069.

Behnamian, J., S.M.T. Fatemi Ghomi. (2011). "Hybrid flowshop scheduling with machine and resource-dependent processing times", *Applied Mathematical Modelling*, 35, 1107-1123.

Behnamian, J. and M. Zandieh. (2011). "A discrete colonial competitive algorithm for hybrid flowshop scheduling to minimize earliness and quadratic tardiness penalties", *Expert Systems with Applications*, 38, 14490-14498.

Botta-Genoulaz, V. (2000). "Hybrid flow shop scheduling with precedence constraints and time lags to minimize maximum lateness", *International Journal of Production Economics*, 64, 101-111.

Bowman, E.H. (1959). "The Schedule-Sequencing problem", *Operations Research*, 7(5), 621-624.

Brah, S.A. and J.L. Hunsucker. (1991). "Branch and bound algorithm for the flow shop with multiple processors", *European Journal of Operational Research*, 51, 88-99.

Brah, S.A. and L.L. Loo. (1999). "Heuristics for scheduling in a flow shop with multiple processors", *European Journal of Operational Research*, 113, 113-122.

Brucker, P. (2007). *Scheduling Algorithms* (5th Ed.). Berlin: Springer.

Brucker, P., B. Jurisch and B. Sievers. (1994). "A branch and bound algorithm for the job shop scheduling problem", *Discrete Applied Mathematics*, 49, 107-127.

Brucker, P., K.A. Kravchenko and Y.N. Sotskov. (1997). "On the complexity of the two machine job-shop scheduling with regular objective functions", *OR Spektrum*, 19, 5-10.

Caricato, P., A. Grieco and D. Serino. (2007). "Tsp-based scheduling in a batch-wise hybrid flow-shop", *Robotics & Computer-Integrated Manufacturing*, 23, 234-241.

Carlier, J. and E. Pinson. (1989). "An algorithm for solving job-shop problem", *Management Science*, 35(2), 164-176.

Carlier, J. and E. Neron. (2000). "An exact method for solving the multi-processor flow-shop", *RAIRO Operations Research*, 34, 1-25.

Carpov, S., J. Carlier, D. Nace and R. Sirdey. (2012). "Two-stage hybrid flow shop with precedence sonstraints and parallel machines at second stage", *Computers & Operations Research*, 39, 736-745.

Chen, B. (1995). "Analysis of classes of heuristics for scheduling a two-stage flow shop with parallel machines at one stage", *Journal of Operations Research Society*, 46, 234-244.

Chang, S. and D. Liao. (1994). "Scheduling flexible flow shops with no setup effects", *IEEE Transactions on Robotics & Automation*, 10(2), 112-122

Cheng, J., Y. Karuno and H. Kise. (2001). "A shifting bottleneck approach for a parallel-machine flowshop scheduling problem", *Journal of the Operations Research Society of Japan*, 44(2), 140-156.

Choi, H. and D. Lee. (2009). "Scheduling algorithms to minimize the number of tardy jobs in two-stage hybrid flow shops", *Computers & Industrial Engineering*, 56, 113-120.

Choong, F., S. Phon-Amnuaisuk and M.Y. Alias. (2011). "Metaheuristic methods in hybrid flow shop scheduling problem", *Expert Systems with Applications*, 38, 10787-10793.

Crowder, B. (2006). *Minimizing the Makespan in a Flexible Flowshop with Sequence Dependent Setup Times, Uniform Machines, and Limited Buffers*, Thesis of Master of Science in Industrial Engineering, College of Engineering and Mineral Resources, West Virginia University. West Virginia

Cura, T. (2008). *Modern Sezgisel Teknikler ve Uygulamaları*, İstanbul: Papatya Yayıncılık.

Ding, F. and D. Kittichartttphayak. (1994). "Heuristics for scheduling flexible flow lines", *Computers & Industrial Engineering*, 26(1), 27-34.

Dudek, R.A., S.S. Panwalkar and M.L. Smith. (1992). "The lessons of flowshop scheduling research", *Operations Research*, 40(1), 7-13.

Dyer, M.E. and L.A. Wolsey. (1990). "Formulating the single machine sequencing problem with release dates as a mixed integer program", *Discrete Applied Mathematics*, 26(2-3), 255-270.

Engin, O. and A. Döyen. (2004). "A new approach to solve hybrid flow shop scheduling problems by artificial immune system", *Future Generation Computer Systems*, 20, 1083-1095.

Figielska, E. (2009). "A genetic algorithm and a simulated annealing algorithm combined with column generation technique for solving the problem of scheduling in the hybrid flowshop with additional resources", *Computers & Industrial Engineering*, 56, 142-151.

Fisher, M.L. (1973). "Optimal Solution of Scheduling Problems Using Lagrange Multipliers: Part I", *Operations Research*, 21(5), 1114-1127.

French, S. (1982). *Sequencing and Scheduling: An Introduction to the Mathematics of the Job-Shop*, Chichester, UK, Ellis Horwood.

Garey, M.R., D.S. Johnson and R. Sethi. (1976). “The complexity of flowshop and jobshop scheduling”, *Mathematics of Operations Research,* 1(2), 117-129.

Gicquel, C., L. Hege, M. Minoux and W. Canneyt. (2012). “A discrete time exact solution approach for a complex hybrid flow-shop scheduling problem with limited-wait constraints”, *Computers & Operations Research,* 39, 629-636.

Greenberg, H.H. (1968). “A branch-bound solution to the general scheduling problem”, *Operations Research,* 16(2), 353-361.

Guinet, A.G.P. and M.M. Solomon (1996). “Scheduling hybrid flowshops to minimize maximum tardiness or maximum completion time”, *International Journal of Production Research,* 34(6), 1643-1654.

Guinet, A., M.M. Solomon, P.K. Kedia and A. Dussauchoy. (1996). “A computational study of heuristics for two-stage flexible flowshops”, *International Journal of Production Research,* 34(5), 1399-1415.

Gupta, J.N.D. (1988). “Two-stage, hybrid flowshop scheduling problem”, *The Journal of Operational Research Society,* 39(4), 359-364.

Gupta, J.N.D. and E.A. Tunc. (1991). “Schedules for a two-stage hynrid flowshop with parallel machines at the second stage”, *International Journal of Production Research,* 29(7), 1489-1502.

Gupta, J.N.D. and E.A. Tunc. (1994). “Scheduling a two-stage hybrid flowshop with seperable setup and removal times”, *European Journal of Operational Research,* 77, 415-428.

Gupta, J.N.D., A.M.A. Hariri and C.N. Potts. (1997). “Scheduling a two-stage hybrid flow shop with parallel machines at the first stage”, *Annals of Operations Research,* 69, 171-191.

Gupta, J.N.D., K. Krüger, V. Lauff, F. Werner and Y.N. Sotskov. (2002). “Heuristics for hybrid flow shops with controllable processing times and assignable due dates”, *Computers & Operations Research,* 29, 1417-1439.

Haouari, M., L. Hidri and A. Gharbi. (2006). "Optimal scheduling of a two-stage hybrid flow shop", *Mathematical Methods of Operations Research*, 64, 107-124.

Hefetz, N. and I. Adiri. (1982). "An efficient optimal algorithm for the two-machine unit-time jobshop schedule-length problem", *Mathematics of Operations Research*, 7(3), 354-360.

Heizer, J. and B. Render. (2006). *Operations Management*, New Jersey, Prentice Hall.

Hidri, L. and M. Haouari. (2011). "Bounding strategies for the hybrid flow shop scheduling problem", *Applied Mathematics & Computation*, 217, 8248-8263.

Hmida, A.B, M. Haouari, M. Huguet and P. Lopez. (2011). "Solving two-stage hybrid flow shop using climbing depth-bounded discrepancy search", *Computers & Industrial Engineering*, 60, 320-327.

Hong, T. and T. Wang. (1999). "A heuristic palmer-based fuzzy flexible flow-shop scheduling algorithm", *1999 IEEE International Fuzzy Systems Conference*, Seoul, Korea.

Hong, T., and T. Wang. (2000). "Fuzzy flexible flow shops at two machine centers for continuous fuzzy domains", *Information Sciences*, 129, 227-237.

Hong, T., T. Wang and S. Wang. (2001). "A palmer-based continuous fuzzy flexible flow-shop scheduling algorithm", *Soft Computing*, 6, 426-433.

Hoogeveen, J.A., J.K. Lenstra and B. Veltman. (1996). "Preemptive scheduling in a two-stage multiprocessor flow shop is NP-hard", *European Journal of Operational Research*, 89, 172-175.

Huang, W. and S. Li. (1998). "A two-stage hybrid flowshop with uniform machines and setup times", *Mathematical & Computer Modelling*, 27(2), 27-45.

Hunsucker, J.L. and J.R. Shah. (1994). "Comparative performance analysis of priority rules in a constrained flow shop with multiple processors environment", *European Journal of Operational Research*, 72, 102-114.

Ignall, E. and L. Schrage. (1965). “Application of the branch and bound technique to some flow-shop scheduling problems”, *Operations Research*, 13(3), 400-412.

Jain, A.S. and S. Meeran. (1999). “Deteministic job-shop scheduling: Past, present and future”, *European Journal of Operational Research*, 113, 390-434.

Janiak, A., E. Kozan, M. Lichtenstein and C. Oğuz. (2007). “Metaheuristic approach to the hybrid flow shop scheduling problem with a cost-related criterion”, *International Journal of Production Economics*, 105, 407-424.

Jayamohan, M.S. and C. Rajendran. (2000). “A comparative analysis of two different approaches to scheduling in flexible flow shops”, *Production Planning & Control,* 11(6), 572-580.

Jenabi, M., S.M.T. Fatemi Ghomi, S.A. Torabi and B. Karimi. (2007). “Two hybrid meta-heuristics for the finite horizon ELSP in flexible flow lines with unrelated parallel machines”, *Applied Mathematics & Computation*, 186, 230-245.

Jin, Z.H., K. Ohno, T. Ito and S.E. Elmaghraby. (2002). “Scheduling hybrid flowshops in printed circuit board assembly lines”, *Production and Operations Management,* 11(2), 216-230.

Johnson, S.M. (1954). “Optimal two- and three-stage production schedules with setup times included”, *Naval Research Logistics Quarterly*, 1(1), 61-68.

Jouglet, A., C. Oğuz and M. Sevaux. (2009). “Hybrid flow-shop: A memetic algorithm using constraint-based scheduling for efficient search”, *Journal of Mathematical Modelling & Algorithms*, 8, 271-292.

Jungwattanakit, J., M. Reodecha, P. Chaovalitwongse and F. Werner. (2009). “A comparison of scheduling algorithms for flexible flow shop problems with unrelated parallel machines, setup times and dual criteria”, *Computers & Operations Research*, 36, 358-378.

Karaboğa, D. (2004). *Yapay Zeka Optimizasyon Algoritmaları*, İstanbul: Atlas Yayın Dağıtım.

Karimi, N., M. Zandieh and R. Kar-Amooz. (2009). “Bi-objective group scheduling in hybrid flexible flowshop: A multi-phase approach”, *Expert Systems with Applications*, 37(6), 4024-4032.

Kim, Y., B. Joo and J. Shin. (2009). “Heuristics for a two-stage hybrid flowshop scheduling problem with ready times and a product mix ratio constraint”, *Journal of Heuristisc*, 15, 19-42.

Kis, T. and E. Pesch. (2005). “A review of exact solution methods for the non-preemptive multiprocessor flowshop problem”, *European Journal of Operational Research*, 164, 592-608.

Khalouli, S., F. Ghedjati and A. Hamzaoui. (2010). “A meta-heuristic approach to solve a JIT scheduling problem in hybrid flow shop”, *Engineerign Applications of Artificial Intelligence*, 23, 765-771.

Kochhar, S. and R.J.T. Morris. (1987). “Heuristic methods for flexible flow line scheduling”, *Journal of Manufacturing Systems*, 6(4), 299-314.

Koulamas C. and G.J. Kyparisis. (2000). “Asymtotically optimal linear time algorithms for two-stage and three-stage flexible flow shops”, *Naval Research Logistics*, 47(3), 259-268.

Kurz, M.E. and R.G. Askin. (2003). “Comparing scheduling rules for flexible flow lines”, *International Journal of Production Economics*, 85, 371-388.

Kurz, M.E. and R.G. Askin. (2004). “Scheduling flexible flow lines with sequence-dependent setup times”, *European Journal of Operational Research*, 159, 66-82.

Kyparisis, G.J. and C. Koulamas. (2001). “A note on weighted comptetion time minimization in a flexible flow shop”, *Operations Research Letters*, 29, 5-11.

Kyparisis, G.J. and C. Koulamas. (2006a). “Flexible flow shop scheduling with uniform parallel machines”, *European Journal of Operational Research*, 168, 985-997.

Kyparisis, G.J., and C. Koulamas. (2006b). "A note on makespan minimization in two-stage flexible flowshops with uniform machines", *European Journal of Operational Research*, 175, 1321-1327.

Lageweg, B.J., J.K. Lenstra and A.H.G. Rinnooy Kan. (1977). "Job-shop Scheduling by Implicit Enumeration", *Management Science*, 24(4), 441-450.

Lawler, E.L., J.K. Lenstra, A.H.G Rinnooy Kan and D.B. Shmoys. (1993). Sequencing and scheduling: Algorithms and complexity. S.C. Graves, A.H.G. Rinnooy Kan and P.H. Zipkin (Ed.), *Handbooks in Operations Research and Management Science, Vol.4: Logistics of Production and Inventory* içinde (ss.445-522). New York: North Holland.

Lee, C. and G.L. Vairaktarakis. (1994). "Minimizing makespan in hybrid flowshops", *Operations Research Letters*, 16, 149-158.

Lee, C. and G.L. Vairaktarakis. (1998). "Performance comparison of some classes of flexible flow shops and job shops", *The International Journal of Flexible Mnufacturing Systems*, 10, 379-405.

Li, S. (1997). "A hybrid two-stage flowshop with part family, batch production, major and minor setups", *European Journal of Operational Research*, 102, 142-156.

Lin, H. and C. Liao, (2003). "A case study in a two-stage hybrid flow shop with setup time and dedicated machines", *International Journal of Production Economics*, 86, 133-143.

Linn, R. and W. Zhang, (1999). "Hybrid Flow Shop Scheduling: A Survey", *Computers & Industrial Engineering*, 37, 57-61.

Liu, C. and S. Chang, (2000). "Scheduling flexible flow shops with sequence-dependent setup effects", *IEEE Transactions on Robotics & Automation*, 16(4), 408-419.

Logendran, R., S. Carson and E. Hanson. (2005). "Group scheduling in flexible flow shops", *International Journal of Production Economics*, 96, 143-155.

Logendran, R., P. deSzoeke and F. Barnard. (2006). "Sequence-dependent group scheduling problems in flexible flow shops", *International Journal of Production Economics*, 102, 66-86.

Luss, H. and M.B. Rosenwein. (1993). "A due date assignment algorithm for multiproduct manufacturing facilities", *European Journal of Operational Research*, 65, 187-198.

Moursli, O. and Y. Pochet. (2000). "A branch-and-bound algorithm for the hybrid flowshop", *International Journal of Production Economics*, 64, 113-125.

Naderi, B., : Zandieh and M.A.H.A. Shirazi. (2009). "Modeling and scheduling a case of flexible flowshops: Total weighted tardiness minimization", *Computers & Industrial Engineering*, 57(4), 1258-1267.

Narasimhan, S.L. and S.S. Panwalkar. (1984). "Scheduling in a two-stage manufacturing process", *International Journal of Production Research*, 22(4), 555-564.

Narasimhan, S.L. and P.M. Mangiameli. (1987). "A comparison of sequencing rules for a two-stage hybrid flow shop", *Decision Sciences*, 18(2), 250-265.

Negenman, E.G. (2001). "Local search algorithms for the multiprocessor flow shop scheduling problem", *European Journal of Operational Research*, 128, 147-158.

Neron, E., P. Baptiste and J.N.D. Gupta. (2001). "Solving hybrid flowshop problem using energetic reasoning and global operations", *Omega The International Journal of Management Science*, 29, 501-511.

Niu, Q., T. Zhou, M. Fei and B. Wang. (2012). "An efficient quantum immune algorithm to minimize mean flow time for hybrid flow shop problems", *Mathematics & Computers in Simulation*, 84, 1-25.

Oğuz, C. and M.F. Ercan. (1997). "Scheduling multiprocessor tasks in a two-stage flow-shop environment", *Computers & Industrial Engineering*, 33(1-2), 269-272.

Oğuz, C. and M.F. Ercan. (2005). "A genetic algorithm for hybrid flow-shop scheduling with multiprocessor tasks", *Journal of Scheduling*, 8, 323-351.

Özbakır, L. (2004). *Çok Objektifli Esnek Atölye Çizelgeleme Problemlerinin Sezgisel Yöntemlerle Modellenmesi, Analizi ve Çözümü*, Erciyes Üniversitesi Sosyal Bilimler Enstitüsü, Yayımlanmamış Doktora Tezi. Kayseri.

Pan, J.C-H. and J-S. Chen. (2005). "Mixed binary integer programming formulations for the reentrant job shop scheduling problem", *Computers & Operations Research*, 32, 1197-1212.

Panwalkar, S.S. and W Iskander. (1977). "A survey of scheduling rules", *Operations Research*, 25(1), 45-61.

Paternina-Arboleda, C.D., J.R. Montoya-Torres, M.J. Acero-Dominguez and M.C. Herrera-Hernandez. (2008). "Scheduling jobs on k-stage flexible flow-shops", *Annals of Operations Research*, 164, 29-40.

Pinedo, M.L. (2008). *Scheduling: Theory, Algorithms and Systems* (3rd Ed.). New York: Springer.

Pinedo, M.L. (2005). *Planning and Scheduling in Manufacturing and Services*, New York: Springer.

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

Quadt, D. and H. Kuhn. (2007a). "Batch scheduling of jobs with identical process times on flexible flow lines", *International Journal of Production Economics*, 105, 385-401.

Quadt, D. and H. Kuhn. (2007b). "A taxonomy of flexible flow line scheduling", *European Journal of Operational Research*, 178, 686-698.

Rajendran, C. and D. Chaudhuri. (1992a). “A multi-stage parallel processor flowshop problem with minimum flowtime”, *European Journal of Operational Research*, 57, 111-122.

Rajendran, C. and D. Chaudhuri.(1992b). “Scheduling in n-job, m-stage flowshop with parallel processors to minimize makespan”, *International Journal of Production Economics*, 27, 137-143.

Randhawa, S.U., C. Jowono and S. Burhanuddin. (1994).”Scheduling in multistage flowshop systems: An application in the food processing industry”, *Industrial Management & Data Systems*, 94(5), 16-24.

Ribas, I., R. Leisten and J.M. Framinan. (2010). “Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective”, *Computers & Operations Research*, 37, 1439-1454.

Ruiz, R. and C. Maroto. (2006). “A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility”, *European Journal of Operational Research*, 169, 781-800.

Ruiz, R., F.S. Şerifoğlu and T. Urlings. (2008). “Modeling realistic hybrid flowshop scheduling problems”, *Computers & Operations Research*, 35, 1151-1175.

Ruiz, R. and J.A. Vazquez-Rodriguez. (2010). “The hybrid flow shop scheduling problem”, *European Journal of Operational Research*, 205(1), 1-18.

Russell, R.S. and B.W. Taylor. (2003). *Operations Management*, New Jersey, Prentice Hall.

Santos, D.L., J.L. Hunsucker and D.E. Deal. (1995). “Global lower bounds for flow shops with multiple processors”, *European Journal of Operational Research*, 80, 112-120.

Santos, D.L., J.L. Hunsucker and D.E. Deal. (1996). “An evaluation of sequencing heuristics in flow shops with multiple processors”, *Computers & Industrial Engineering*, 30(4), 681-692.

Sawik, T. (1995). “Scheduling flexible flow lines with no in-process buffers”, *International Journal of Production Research*, 33(5), 1357-1367.

Sawik, T. (2000). “Mixed integer programming for scheduling flexible flow lines with limited intermediate buffers”, *Mathematical & Computer Modelling*, 31, 39-52.

Sawik, T. (2002). “An exact approach for batch scheduling in flexible flow lines with limited intermediate buffers”, *Mathematical & Computer Modelling*, 36, 461-471.

Shiau, D., S. Cheng and Y. Huang. (2008). “Proportionate flexible flow shop scheduling via a hybrid constructive genetic algorithm”, *Expert Systems with Applications*, 34, 1133-1143.

Soewandi, H. and S.E. Elmaghraby. (2001). “Sequencing three-stage flexible flowshops with identical machines to minimize makespan”, *IIE Transactions*, 33, 985-993.

Soewandi, H. And S.E. Elmaghraby. (2003). “Sequencing on two-stage hybrid flowshops with uniform machines to minimize makespan”, *IIE Transactions*, 35, 467-477.

Sotskov, Y.N. and N.V. Shakhlevich. (1995). “NP-hardness of shop-scheduling problems with three jobs”, *Discrete Applied Mathematics*, 59(3), 237-266.

Sriskandarajah, C. and S.P. Sethi. (1989). “Scheduling algorithms for flexible flowshops: Worst and average case performance”, *European Journal of Operational Research*, 43, 143-160.

Szwarc, W. (1960). “Solution of the Akers-Friedman scheduling problem”, *Operations Research*, 8(6), 782-788.

Şevkli, M. (2005). *Atölye Tipi Çizelgeleme Problemlerinde Paçacık Sürü Optimizasyonu Yaklaşımı ve Genetik Algoritma Modeli ile Karşılaştırılması*, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, Yayımlanmamış Doktora Tezi.

Tavakkoli-Moghaddam, R., N. Safaesi and F. Sassani. (2009). "A memetic algorithm for the flexible flow line scheduling problem with processor blocking", *Computers & Operations Research*, 36, 402-414.

T'kindt, V., J-C. Billaut. (2006). *Multicriteria Scheduling: Theory, Models and Algorithms* (Second Edition), Berlin: Springer.

Tseng, C., C. Liao and T. Liao. (2008). "A note on two-stage hybrid flowshop scheduling with missing operations", *Computers & Industrial Engineering*, 54, 605-704.

Tsubone, H., M. Ohba, H. Takamuki and Y. Miyake. (1993). "A production scheduling system for a hybrid flow shop – a case study", *Omega International Journal of Management Science*, 21(2), 205-214.

Uetake, T., H. Tsubone and M. Ohba. (1995). "A production scheduling system in a hybrid flow shop", *International Journal of Production Economics*, 41, 395-398.

Urlings, T., R. Ruiz and T. Stützle. (2010). "Shifting representation search for hybrid flexible flowline problems", *European Journal of Operational Research*, 207, 1086-1095.

Verma, S. and M. Dessouky. (1999). "Multistage hybrid flowshop scheduling with identical jobs and uniform parallel machines", *Journal of Scheduling*, 2, 135-150.

Vignier, A., J. Billaut, C. Proust and V. T'kindt. (1996). "Resolution of some 2-stage hybrid flowshop scheduling problems", *IEEE International Conference on Systems, Man and Cybernetics*, 4, 2934-2941.

Vignier, A., D. Dardilhac, D. Dezalay and C. Proust. (1996). "A branch and bound approach to minimize the total completion time in a k-stage hybrid flowshop", *IEEE Conference on Emerging Technologies and Factory Automation*, 1, 215-220.

Vignier, A., P. Commandeur and C. Proust. (1997). “New lower bound for the hybrid flowshop scheduling problem”, *6th International Conference on Emerging Technologies and Factory Automation*, 446-451.

Wang, L. and D. Li. (2002). “A scheduling algorithm for flexible flowshop problem”, *Proceedings of the 4th World Congress on Intelligent Control and Automation*, Shangai, China.

Wang, H. (2005). “Flexible flowshop scheduling: optimum, heuristics and artificial intelligence solutions”, *Expert Systems*, 22(2), 78-85.

Wang, Z., W. Xing and F. Bai. (2005). “No-wait flexible flowshop with no-idle machines”, *Operations Research Letters*, 33, 609-614.

Wang, S. and M. Liu. (2013a). “A heuristic method for two-stage hybrid flow shop with dedicated machines”, *Computers & Operations Research*, 40, 438-450.

Wang, S. and M. Liu. (2013b). “A genetic algorithm for tow-stage no-wait hybrid flow shop scheduling problem”, *Computers & Operations Research*, 40, 1064-1075.

Wittrock, R.J. (1985). “Scheduling algorithms for flexible flow lines”, *IBM Journal of Research & Development*, 29(4), 401-412.

Wittrock, R.J. (1988). “An adaptable scheduling algorithm for flexible flow lines”, *Operations Research*, 36(3), 445-453.

Xie, J. and X. Wang. (2005). “Complexity and algorithms for two-stage flexible flowshop scheduling with availability constraints", *Computers & Mathematics with Applications*, 50, 1629-1638.

Yang, Y. (1998). *Optimization and Heuristic Algorithms for flexible Flow Shop Scheduling*, Thesis of PhD in the Graduate School of Arts and Science, Columbia University. Columbia.

Yang, Y., S. Kreipl and M. Pinedo. (2000). “Heuristics for minimizing total weighted tardiness in flexible flow shops”, *Journal of Scheduling*, 3, 89-108.

Yang, J. (2011). “Minimizing total completion time in two-stage hybrid flow shop with dedicated machines”, *Computers & Operations Research*, 38, 1045-1053.

Yao, F.S., M. Zhao and H. Zhang. (2012). “Two-stage hybrid flow shop scheduling with dynamic job arrivals”, *Computers & Operations Research*, 39, 1701-1712.

Yu, L., A. Ohsato, T. Kawakami and T. Sekiguchi. (1999). “COBRA-based design and development of distributed scheduling systems: An application to flexible flowshop scheduling systems”, *IEEE International Conference on Systems, Man & Cybernetics*, Niigata.

Zandieh, M., S.M.T. Fatemi Ghomi and S.M. Moattar Husseini. (2006). “An immune algorithm approach to hybrid flow shop scheduling with sequence-dependent setup times”, *Applied Mathematics & Computation*, 180, 111-127.

Zandieh, M. and M. Gholami. (2009). “An immune algorithm for scheduling a hybrid flow shop with sequence-dependent setup times and machines with random breakdowns”, *International Journal of Production Research*, 47(24), 6999-7027.

Zobolas, G.I., C.D. Tarantilis and G. Ioannou. (2008). Exact, Heuristic and Meta-heuristic Algorithms for Solving Shop Scheduling Problems. F. Xhafa ve A. Abraham (Ed.), *Metaheuristics for Scheduling in Industrial and Manufacturing Applications* içinde (ss.1-40). Berlin: Springer.

ÖZ GEÇMİŞ

KİŞİSEL BİLGİLER

Adı, Soyadı: Emre ASLAN
Uyruğu: Türkiye (TC)
Doğum Tarihi ve Yeri: 8 Eylül 1981, Kayseri
Medeni Durumu: Evli
Tel: +90 356 252 16 16 / 2441
Email: emre.aslan@gop.edu.tr
Yazışma Adresi: Gaziosmanpaşa Üniversitesi İİBF İşletme Bölümü TOKAT

EĞİTİM

Derece	**Kurum**	**Mezuniyet Tarihi**
Yüksek Lisans	Erciyes Üni. Sosyal Bilimler Ens.	2007
Lisans	Ege Üni. İİBF İşletme	2004
Lise	Talas FKTAL, Kayseri	1999

İŞ DENEYİMLERİ

Yıl	**Kurum**	**Görev**
2004 – Halen	Gaziosmanpaşa Üni.	Araştırma Görevlisi

YABANCI DİL

İngilizce

YAYINLAR

Aslan, E.; O. Unutulmaz (2013). Esnek Akış Tipi Çizelgeleme Problemi için Karma Tamsayılı Programlama Modeli, *13. Üretim Araştırmaları Sempozyumu*, 25-27 Eylül 2013, Sakarya.

İlkay, M.S.; E. Aslan (2012). The Effect of the ISO 9001 Quality Management System on the Performance of SMEs, *International Journal of Quality & Reliability Management*, 29(7), 753-778.

Özdemir, A.İ.; E. Aslan (2011). Supply Chain Integration, Competition Capability and Business Performance: A Study on Turkish SMEs, *Asian Journal of Business Management*, 3(4), 325-332.

Aslan, E.; A. Yiğit; O. Karkacıer (2011). Sıra Bağımlı Hazırlık Zamanlı Paralel Makinelerde Çizelgeleme: Bir Tekstil İşletmesi Uygulaması, *11. Üretim Araştırmaları Sempozyumu*, 23-24 Haziran 2011, İstanbul.

Yiğit, A.; E. Aslan, O. Karkacıer (2010). Ofis Mobilyaları Üreten Bir Tesisin Yerleşiminin Yeniden Düzenlenmesi, *10. Ulusal Üretim Araştırmaları Sempozyumu*, 16-18 Eylül 2010, Girne-KKTC.

Aslan, E.; M.S. İlkay, A.İ. Özdemir (2009). ISO 9001 Belgeli KOBİ'lerin Performans Farklılıklarının Bazı Faktörler Bakımından Analizi, *Dumlupınar Üniversitesi Sosyal Bilimler Dergisi,* 25, 35-40.

İlkay, M.S.; E. Aslan (2008) Erciyes Üniversitesi Yeni Öğrenci Kayıt Sisteminin Simülasyonu, *8. Ulusal Üretim Araştırmaları Sempozyumu.* 24-25 Ekim 2008, İstanbul.

İlkay, M. S.; E. Aslan (2006). ISO 9001: 2000 Kalite Yönetim Sisteminin Kayseri Bölgesindeki KOBİ'lerin Performanslarına Etkileri Üzerine Bir Araştırma, *Hacettepe Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*. 24(2), 67-83.