

**ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

YÜKSEK LİSANS TEZİ

112531

B. Gülsün K. SARIOĞLU

112531

Sipariş Tipi Üretim Sistemlerinde Değişken Darboğazların
Çizelgelenmesi İçin Yeni Bir Yaklaşım

**T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ**

ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

ADANA, 2001

**ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**SİPARİŞ TİPİ ÜRETİM SİSTEMLERİNDE OLUŞAN DARBOĞAZLA
ÇİZELGELENMESİNDE KISITLAR TEORİSİ YAKLAŞIMI**

GÜLSÜN K. SARIOĞLU

**YÜKSEK LİSANS TEZİ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI**

Bu tez 18/07/2001 tarihinde aşağıdaki jüri üyeleri tarafından Oybirliği/~~Oyçokluğu~~ ile kabul edilmiştir.

İmza: 

Doç.Dr. Rızvan EROL
DANIŞMAN

İmza: 

Yrd.Doç.Dr.Funda YILDIRIM
ÜYE

İmza: 

Yrd.Doç.Dr. Arzu GÜÇRAY
ÜYE

Bu tez Estitümüz Endüstri Mühendisliği Anabilim Dalı'nda hazırlanmıştır.

Kod No: 1845

**T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANİTASYON MERKEZİ**





Prof.Dr. Melih BORAL
Enstitü Müdürü

Bu çalışma Çukurova Üniversitesi Araştırma Fonu tarafından desteklenmiştir.

Proje No: FBE.2000.YL.21

Not: Bu tezde kullanılan özgün ve başka kaynaktan yapılan bildirişlerin, çizelge, şekil ve fotoğrafların kaynak gösterilmeden kullanımı, 5846 sayılı Fikir ve Sanat Eserleri Kanunundaki hükümlere tabidir.

İÇİNDEKİLER

ÖZ.....	I
ABSTRACT.....	II
ÖNSÖZ.....	III
ŞEKİLLER DİZİNİ.....	V
TABLolar DİZİNİ.....	VI
1.GİRİŞ.....	1
1.1. Problemin Tanımı.....	1
1.1.1. Üretim Çizelgeleme	4
1.1.2. İncelenen Çizelgeleme Problemi.....	6
1.2. Çalışmanın Amacı.....	7
1.3. Çalışmanın Kapsamı ve Varsayımlar.....	9
1.3.1. Kapsam.....	9
1.3.2. Varsayımlar.....	10
1.4. Çalışmanın Adımları.....	10
1.5. Tezin Organizasyonu.....	11
2. ÖNCEKİ ÇALIŞMALAR.....	12
2.1. Çizelgeleme Algoritmaları	12
2.2. Kısıtlarla Yönetim Felsefesi	13
2.3. Darboğaz Kaynağın Yönetimi	16
2.4. Darboğaz Kaynağın Değişimi	20
2.5. Literatürün Genel Değerlendirmesi.....	23
3. MATERYAL VE METOD.....	24
3.1. Materyal.....	24
3.2. Metod.....	25
3.2.1. Sistemin Karakteristikleri.....	25
3.2.2. Performans Değişkenleri.....	26
3.2.3. Sezgisel Çizelgeleme Algoritmaları.....	27
3.2.4. Darboğaz Kaynaklar.....	30
3.2.5. Değişken Darboğaz Olması Durumu.....	33

3.2.6. Geliştirilen Çizelgeleme Algoritması.....	36
3.2.6.1. Darboğaz Kaynağın Tanımlanması ve Darboğaz Değişiminin Ölçülmesi.....	37
3.2.6.2. Çizelgeleme Algoritması.....	40
3.2.6.3. Örnek Problem Algoritması.....	43
3.2.7. Simülasyon Deneylerinin Tasarımı.....	45
3.2.8. Simülasyon Deneyleri.....	47
4. ARAŞTIRMA BULGULARI VE TARTIŞMA.....	54
4.1. Simülasyon Sonuçları.....	54
4.2. Sipariş Geliş Hızının Etkileri.....	63
4.2.1. Sipariş Geliş Hızı - Ortalama Akış Zamanı İlişkisi.....	63
4.2.2. Sipariş Geliş Hızı - Ortalama Ara Stok Düzeyi İlişkisi.....	66
4.2.3. Sipariş Geliş Hızı - Geç Kalan Sipariş Sayısı İlişkisi.....	66
4.2.4. Sipariş Geliş Hızı - Pozitif Gecikme İlişkisi.....	71
4.2.5. Sipariş Geliş Hızı - Geç Kalma Yüzdesi İlişkisi.....	71
4.2.6. Sipariş Geliş Hızı - Toplam Geç Kalma Süresi İlişkisi.....	76
4.3. Sipariş Termin Zamanının Etkileri.....	76
5. SONUÇLAR VE ÖNERİLER.....	83
5.1. Çalışmanın Özetlenmesi.....	83
5.2. Sonuçlar.....	84
5.3. Sonraki Çalışmalar İçin Öneriler.....	86
KAYNAKLAR.....	87
ÖZGEÇMİŞ.....	91
EKLER.....	92
EK 1. Sipariş Akış Şeması.....	92
EK 2. Sezgisel Çizelgeleme Kuralları.....	93
EK 3. Darboğaz Kaynakların Tanımlanması.....	120
EK 4. Algoritma 1 İçin Simülasyon Uygulaması.....	132
EK 5. Algoritma 2 İçin Simülasyon Uygulaması.....	145

ÖZ

YÜKSEK LİSANS TEZİ

SİPARİŞ TİPİ ÜRETİM SİSTEMLERİNDE OLUŞAN DARBOĞAZLARIN
ÇİZELGELENMESİNDE KISITLAR TEORİSİ YAKLAŞIMI

B. GÜLSÜN K. SARIOĞLU

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

Danışman: Doç. Dr. Rızvan EROL

Yıl: 2001, **Sayfa:** 158

Jüri: Doç. Dr. Rızvan EROL

Yrd. Doç. Dr. Funda YILDIRIM

Yrd. Doç. Dr. Arzu GÜÇRAY

Bu çalışmada, özellikle farklı sipariş tipleri için çalışan veya değişkenliklere maruz kalan üretim ortamlarında ortaya çıkan değişken darboğaz olma durumunun çözümü için dinamik bir çizelgeleme ve kontrol yaklaşımı geliştirilmiştir.

Üretim ortamında ortaya çıkması olası görünen darboğaz kaynakların, kısıtlarla yönetim mantığına dayanarak, öncelikli olarak kontrolü ve bu kaynaklara dayanarak da tüm üretim ortamının çizelgelenmesi amaçlanmıştır. Değişken darboğaz kaynak durumu için darboğaz kaynağı dinamik olarak kontrol eden ve DBR(drum buffer rope) tekniğinin uyarlanmasıyla elde edilen yeni bir çizelgeleme algoritması geliştirilmiştir.

Önerilen yeni dinamik çizelgeleme algoritması özellikle ara stok düzeylerinin, siparişlerin geç kalmalarının ve akış süresinin azaltılmasını amaçlanmaktadır. Algoritma, değişik sezgisel çizelgeleme kuralları ile (EDD, FCFS, STR, MOPNR, MWKR, S/RO, SD, LWKR, SPT) simülasyon deneyleri kullanılarak karşılaştırılmış ve özellikle de ara stok düzeyleri ve gecikmeler ile ilgili tüm ölçütlerde üstünlük sağladığı görülmüştür. Simülasyon deneylerinde büyük ölçekli bir tekstil işletmesinin verileri esas alınmış ve mevcut duruma ilave olarak, farklı iş geliş oranları ve termin zamanları için de deneyler gerçekleştirilmiştir.

Anahtar Kelimeler: Sipariş tipi üretim, Dinamik çizelgeleme, Darboğaz çizelgeleme, Darboğaz değişimi

ABSTRACT

MSc THESIS

A THEORY OF CONSTRAINTS APPROACH TO BOTTLENECK SCHEDULING IN JOB SHOP PRODUCTION SYSTEMS

B. GÜLSÜN K. SARIOĞLU

DEPARTMENT OF INDUSTRIAL ENGINEERING
INSTITUTE OF NATURAL AND APPLIED SCIENCES
UNIVERSITY OF ÇUKUROVA

Supervisor: Assoc. Prof. Rızvan EROL
Year:2001, **Pages:** 158

Jury: Assoc. Prof. Rızvan EROL
Assistant Prof. Funda YILDIRIM
Assistant Prof. Arzu GÜÇRAY

In this study, the bottleneck shiftiness problem faced especially in job shop production systems with product-mix changes or variations in production environment is investigated and a dynamic production scheduling and control algorithm is developed.

Based on the Management with Constraints approach, it is proposed to control the possible bottleneck resources and schedule the production around these resources. In case of bottleneck shiftiness, a new scheduling algorithm, a modified version of the DBR (drum buffer rope) technique, is developed to control the production dynamically.

The proposed new dynamic scheduling algorithm aims to reduce work in process levels, the ratio of positively late jobs, and the average flow time. This developed algorithm are compared with the well-known dispatching rules (EDD, FCFS, STR, MOPNR, MWKR, S/RO, SD, LWKR, SPT) using simulation experiments. Simulation results indicate that the developed algorithm outperform the stated dispatching rules especially in terms of ratio of positively late jobs and work in process levels. Simulation experiments use the production data collected at a large textile firm. In addition to the current production data, simulation experiments are conducted for varying job arrival rates and due dates.

Key Words: Job-shop production, Dynamic scheduling, Bottleneck scheduling, Bottleneck shiftiness

ÖNSÖZ

Güçlü olmayı sürdürmenin, güçlü olmak kadar hatta daha da zor olduğu, iletişim araçlarıyla birlikte küçülen, bununla birlikte güçlü olanın sınırsız olarak büyüebildiği bir ekonomik ortam, 21. yüzyılın eşiğinde kendini göstermektedir. Artan rakipler ve zorlaşan rekabet koşulları, firmalar için ayakta kalmanın koşulu olarak 'rekabet üstünlüğü' sağlamak zorunluluğunu doğurmuştur.

Yaşanan bu gelişmeler esnasında bilim sürekli olarak rekabet üstünlüğü sağlamak amacıyla malzeme, makine, işçilik vb. üretim kaynaklarından oluşan ve sürekli olarak karmaşıklaşan üretim sistemlerinin daha etkin ve verimli yönetilebilmesi için yeni yöntem ve ilkeler geliştirmeye çalışmıştır ve bu çalışmalar günümüzde de, artan bir hızla devam etmektedir.

Firmaların ayakta kalabilmeleri ve varlıklarını sürdürebilmeleri, ancak değişen şartlara ve isteklere hızlı tepkiler verebilmesi ve müşteri memnuniyetini sağlaması ile mümkün olabilecektir. Bunun için de firmaların kendi kısıtlarını farketmesi gerekmektedir. Kısıtlar firmada her aşamada olabilir, ancak en fazla üzerinde durulan, üretim aşamasında ortaya çıkan kısıtlardır. Üretim ortamında kısıt, yoğunlukla kapasitesi, talep karşısında yetersiz kalan ve tüm üretimin darboğazı haline gelen kaynaklardır.

Etkin bir yönetim için firmanın, tüm üretim düzeyini kontrol eden bu darboğaz kaynaklara odaklanması gerekmektedir. Darboğaz kaynaklarını tespit ederek ve bu kaynaklara üretim çizelgelemede odaklanarak firmanın, üretim (çıktı) düzeyini ve oluşan ara stok miktarlarını kontrol etmesi mümkündür.

Üretim ortamlarında çoğu zaman, üretim ortamında birden fazla darboğaz kaynak bulunur. Gelen siparişin özelliğine veya diğer değişkenliklere bağlı olarak, bir gün bir kaynak darboğazken, başka bir gün darboğazın değiştiği durumlarla, sipariş tipi üretim ortamlarında sıkça rastlanır. Böyle durumlarda tek bir kaynağı darboğaz kabul ederek, darboğaz değişse bile o kaynağa odaklanarak çizelgeleme yapılmaya devam edilmesi, etkin sonuçlar vermeyecektir. Değişken darboğazlı durumlarda, güncellenen iş sıralamaları ve değişken çizelgeler yapılmalıdır.

Bu çalışmada kısıtlarla yönetim felsefesine dayanarak, darboğaz kaynağına nasıl odaklanılacağı ve değişken darboğazlı ortamlarda dinamik gözlem yapan çizelgelerin nasıl yapılabileceği ile ilgili algoritmalar geliştirilmiş ve değişken darboğaz durumunun iki kaynak arasında olması durumu örnek seçilerek, algoritmanın verdiği sonuçlara bakılmıştır.

Değişken darboğaz kaynakları dinamik olarak kontrol etmeyi amaçlayan, geliştirilen algoritmalar, gerçek uygulamalarda da kullanılan değişik sezgisel çizelgeleme kuralları ile, simülasyon deneyleri yardımıyla karşılaştırılmış ve belirlenen performans ölçütleri için, bir tekstil işletmesinin verileri kullanılarak etkinliği ölçülmüştür. Geliştirilen algoritmaların, ortalama akış zamanı, ortalama ara stok düzeyi ve gecikmeler ile ilgili tüm ölçütlerde etkin olduğu görülmüştür.

Bu çalışma ile, farklı ürün karmaları ile çalışan ve üretim ortamlarında değişkenliklere maruz kalan işletmelerde darboğaz kaynağın ve buna bağlı olarak tüm üretim ortamının etkin bir şekilde çizelgelenmesi sağlanabilecektir.

Çalışmam süresince her aşamada yardımlarını esirgemeyen hocalarım sayın Doç. Dr. Rızvan EROL, Yrd. Doç. Dr. Arzu GÜÇRAY ve Yrd. Doç. Dr. Funda YILDIRIM'a teşekkür ederim.

ŞEKİLLER DİZİNİ

Şekil 1.1. Bir Üretim Planlama Sistemi.....	3
Şekil 4.1. Sipariş Geliş Hızı – Ortalama Akış Zamanı İlişkisi (1. Algoritma).....	64
Şekil 4.2. Sipariş Geliş Hızı – Ortalama Akış Zamanı İlişkisi (2. Algoritma).....	65
Şekil 4.3. Sipariş Geliş Hızı – Ortalama Ara Stok İlişkisi (1. Algoritma).....	67
Şekil 4.4. Sipariş Geliş Hızı – Ortalama Ara Stok İlişkisi (2. Algoritma).....	68
Şekil 4.5. Sipariş Geliş Hızı – Geç Kalan Sipariş Sayısı İlişkisi (1. Algoritma).....	69
Şekil 4.6. Sipariş Geliş Hızı – Geç Kalan Sipariş Sayısı İlişkisi (2. Algoritma).....	70
Şekil 4.7. Sipariş Geliş Hızı – Pozitif Gecikme İlişkisi (1. Algoritma).....	72
Şekil 4.8. Sipariş Geliş Hızı – Pozitif Gecikme İlişkisi (2. Algoritma).....	73
Şekil 4.9. Sipariş Geliş Hızı – Geç Kalma Yüzdesi İlişkisi (1. Algoritma).....	74
Şekil 4.10. Sipariş Geliş Hızı – Geç Kalma Yüzdesi İlişkisi (2. Algoritma).....	75
Şekil 4.11. Sipariş Geliş Hızı–Toplam Geç Kalma Süresi İlişkisi (1. Algoritma)....	77
Şekil 4.12. Sipariş Geliş Hızı–Toplam Geç Kalma Süresi İlişkisi (2. Algoritma)....	78
Şekil 4.13. Termin Zamanı–Geç Kalan Sipariş Sayısı İlişkisi (1. ve 2. Algoritma).	81
Şekil 4.14. Termin Zamanı – Geç Kalma Yüzdesi İlişkisi (1. ve 2. Algoritma).....	82

TABLolar DİZİNİ

Tablo 3.1. Sezgisel Çizelgeleme Kuralları.....	29
Tablo 3.2. Darboğaz Kaynakların Tespiti.....	38
Tablo 4.1. Exp(2.2) Geliş Oranı İçin Sezgisel Kuralların Performans Ölçütleri Sonuçları.....	55
Tablo 4.2. Simülasyon Deneylerinde Kullanılan Zaman Stoğu Alt ve Üst Sınırları..	56
Tablo 4.3.Exp(2.2) Geliş Oranı İçin 1. Algoritma Performans Ölçütleri Sonuçları...	57
Tablo 4.4.Exp(2.2) Geliş Oranı İçin 2. Algoritma Performans Ölçütleri Sonuçları...	58
Tablo 4.5. Sipariş Geliş Hızı – Ortalama Akış Zamanı İlişkisi (1. Algoritma).....	64
Tablo 4.6. Sipariş Geliş Hızı – Ortalama Akış Zamanı İlişkisi (2. Algoritma).....	65
Tablo 4.7. Sipariş Geliş Hızı – Ortalama Ara Stok İlişkisi (1. Algoritma).....	67
Tablo 4.8. Sipariş Geliş Hızı – Ortalama Ara Stok İlişkisi (2. Algoritma).....	68
Tablo 4.9. Sipariş Geliş Hızı – Geç Kalan Sipariş Sayısı İlişkisi (1. Algoritma).....	69
Tablo 4.10. Sipariş Geliş Hızı – Geç Kalan Sipariş Sayısı İlişkisi (2. Algoritma).....	70
Tablo 4.11. Sipariş Geliş Hızı – Pozitif Gecikme İlişkisi (1. Algoritma).....	72
Tablo 4.12. Sipariş Geliş Hızı – Pozitif Gecikme İlişkisi (2. Algoritma).....	73
Tablo 4.13. Sipariş Geliş Hızı – Geç Kalma Yüzdesi İlişkisi (1. Algoritma).....	74
Tablo 4.14. Sipariş Geliş Hızı – Geç Kalma Yüzdesi İlişkisi (2. Algoritma).....	75
Tablo 4.15. Sipariş Geliş Hızı – Toplam Geç Kalma Süresi İlişkisi (1. Algoritma)..	77
Tablo 4.16. Sipariş Geliş Hızı – Toplam Geç Kalma Süresi İlişkisi (2. Algoritma)..	78
Tablo 4.17. Termin Zamanı – Ortalama Akış Zamanı İlişkisi (1. ve 2. Algoritma)...	80
Tablo 4.18. Termin Zamanı – Ortalama Ara Stok İlişkisi (1. ve 2. Algoritma).....	80
Tablo 4.19. Termin Zamanı–Geç Kalan Sipariş Sayısı İlişkisi (1. ve 2. Algoritma).	81
Tablo 4.20. Termin Zamanı – Pozitif Gecikme İlişkisi (1. ve 2. Algoritma).....	80
Tablo 4.21. Termin Zamanı – Geç Kalma Yüzdesi İlişkisi (1. ve 2. Algoritma).....	82
Tablo 4.22.Termin Zamanı–Toplam Geç Kalma Süresi İlişkisi (1. ve 2. Algoritma)	80

1. GİRİŞ**1.1.Problemin Tanımı**

Bugüne kadar mühendislik bilimleri kapsamında yapılan çalışmaların önemli bir kısmı, üretim tekniklerinin geliştirilmesi ve üretimin kısıtlı kaynaklarının etkin kullanılması amacına yönelik olmuştur. Zaman içinde, kullanılan yöntem, araç ve teknoloji değişikliklerine bağlı olarak gelişmeler ve yeni yaklaşımlar ortaya çıkmış, firmalar ve ülkeler bu alandaki başarılarıyla ve uyum yetenekleriyle doğru orantılı olacak şekilde faydalar sağlamış ya da kaybetmiştir.

Üretim, mühendisler tarafından, bir fiziksel varlık üzerinde onun değerini artırıcı bir değişiklik yapmak veya hammadde ve yarı mamulleri mamule dönüştürmek olarak tanımlanır. Ekonomistler ise üretimi, bir fayda sağlamak şeklinde düşünürler. En yalın anlamıyla üretimi, yaratılan değer olarak tanımlamak mümkündür.

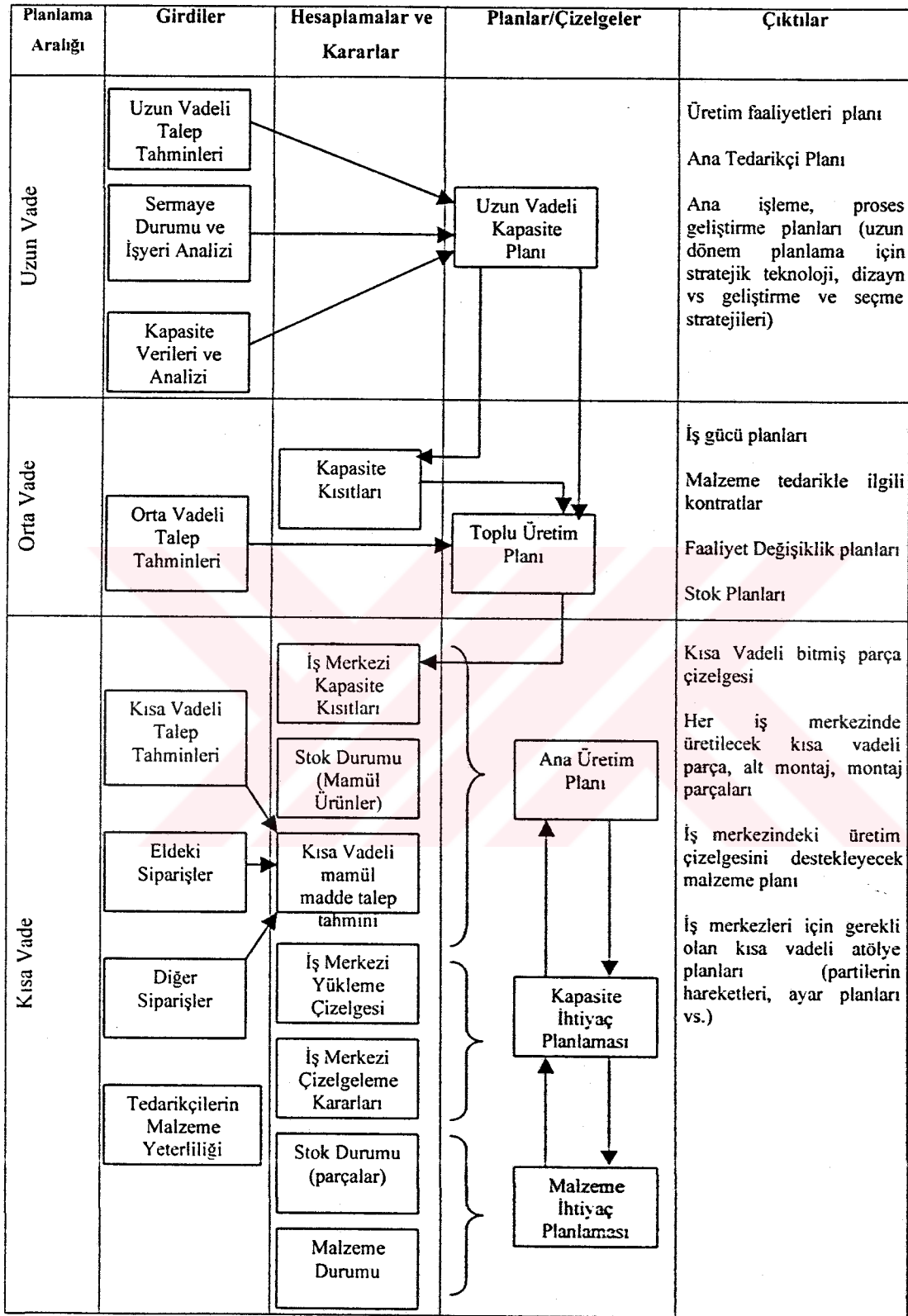
Üretim planlama ve kontrol ise, bir üretim yönetimi etkinliği olarak üretilecek ürünü belirlemek, üretim için donanım gereğini saptamak ve ürünlerin istenen kalite ve maliyette, istenen sürede, doğru zamanlarda ve istenen miktarlarda üretimini sağlayacak çizelgeleme, programlama çalışmalarını kapsar. Diğer bir ifadeyle gelecekteki faaliyetlerin (veya miktarlarının) düzeylerini veya limitlerini belirleyen ve gerekli zamanlarda önlem alan fonksiyona üretim planlama ve kontrol denir. Üretim planlama ve kontrolünün hedefi, kaynak kayıplarını en aza indirmek ve üretimde en yüksek verimliliği sağlamaktır. En yüksek verimlilik ise istenilen miktarda ürünü, istenilen zamanda ve kalitede en iyi ve en ucuz yöntemlerle üretmekle sağlanır (Acar,1989).

Üretim planlama farklı organizasyonel düzeylerde ve değişik zaman aralıklarını içerecek şekilde oluşur (Şekil 1.1). Firmanın üst yönetimi uzun vadeli kapasite planlarını, bir aşağı düzeyde operasyondan sorumlu yöneticiler orta vadeli planları oluştururken, kısa vadeli planlar (çizelgeler) fabrika düzeyinde oluşturulur ve ayrıntılı olarak ürünlerin üretim miktarlarını ve üretilecekleri zamanları içerir.

Uzun dönemli planlama: Bu planlama süreci, fabrikaların tasarımı ve yerleşimi, dağıtım kanallarının seçimi, üretim ve depo kapasiteleri, bölüm ve transfer hatlarının tasarımı, müşteri hizmet politikasının seçimi gibi konuları içerir. Bu kararlar 2 ila 5 yıllık dönemler için verilir ve çoğunlukla tahminler ile senaryolar hazırlanarak alternatifler gözden geçirilir. Beş yıllık talep ne olacak? Hangi teknoloji kullanılmalı? Üç yıl içinde yüksek teknoloji montaj hattını kurmanın riski nedir? gibi sorulara cevap aranır (Morton ve Pentico, 1993). Bu kararların verilebilmesi için pazar araştırması, uzun dönemlik tahminler ve kaynak planlaması gibi ön çalışmalar yapılır.

Orta dönemli planlama: Uzun dönemli planlama süreci sonunda işletmenin genel politikası ve kaynak kısıtları belirlenir. Bu genel politika ve kısıtlar çerçevesinde birkaç aydan iki yıla kadar bir planlama dönemi göz önünde tutularak, olası mevsimsel talep dalgalanmalarında üretimin düzgünleştirilmesi, dış kaynak değişimleri, örneğin işçi kiralama veya fason üretim ile, iç kaynak değişimleri, fazla mesai, yeni üretim hattı açılması, vardiya değişimi, kaynakların paylaşımı gibi, orta dönemli kararlar verilir. Bu kararların verilebilmesi için tahminler, işgücü planlaması, üretim planlaması gibi ön çalışmaların yapılması gerekir.

Kısa dönemli planlama: Bu planlama süreci üretim çizelgeleri ve iş programlarının hazırlanması ve üretim kontrolü gibi faaliyetleri içerir. Genellikle birkaç haftalık bir planlama dönemi gözönüne alınır. Kısa dönemli planlamada malzeme ihtiyaç planlaması, işyeri yükleme, termin tarihlerinin saptanması gibi konularla ilgilenilir. Alınmış siparişler ve istatistiksel tahminler, ana çizelgeyi oluşturmak için kullanılır. Ardından her ürün alt bileşenlerine indirgenerek her bir ara kademe üretimin ayrıntıları oluşturulur. Kısa dönemli planlama süreci, üretim miktarlarının, belirlenen hedeflere ulaşmak üzere sürekli kontrolü ve gerekirse yeniden ayarlanması, malzeme eksikliği, makine bozulmaları gibi aksaklıkların giderilmesi, işçilerin üretim merkezlerine tahsisi, önceliklerin belirlenmesi, imalat ara stok seviyelerinin tespiti gibi kararları içerir (Acar, 1989).



Şekil 1.1. Bir Üretim Planlama Sistemi (Kaynak: Yılmaz, M. İTÜ E.M. sayfası)

1.1.1 Üretim Çizelgeleme

Üretim çizelgeleme, üretim planlamaya göre daha ayrıntılı ve kısa dönemlidir. Çizelgeleme, en yakın zamandaki üretim amaçlarına ulaşabilmek için mevcut koşulları göz önüne alarak ayrıntılı bir plan çıkarma işlemidir.

Çizelgeleme (scheduling) ve sıralama (sequencing, dispatching) kavramları çoğunlukla birbirleri yerine kullanılsa da, çizelgelemenin daha geniş bir anlama sahip olduğu söylenebilir. Sıralama, bir makinede işlenmek için bekleyen işler kuyruğundan bir sonraki işin belirlenmesi ve seçimini sağlayan karar kuralı iken çizelgeleme, aynı zamanda üretim parti büyüklüğünü ve kaynakların işlere dağılımını da içeren sıralama aktivitesidir (Melnik ve diğerleri, 1986). Çizelgeleme, istenen miktarda çıktıyı, istenen zamanda, aktiviteler ve kaynaklar arasındaki çok sayıdaki zaman ve ilişki kısıtlarını da sağlayarak üretebilmek için gerekli tüm aktivitelerin yerine getirilebilmesi amacıyla kaynakların kullanımlarının organizasyonu, seçimi ve zamanlaması olarak tanımlanabilir (Morton ve Pentico, 1993).

Çizelgeleme problemleri çoğu zaman aktiviteleri ve kaynakları birbirine ve sistemdeki olaylara bağlayan çok sayıda kısıtla karmaşık hale gelmiştir. Sistemde üstesinden gelinmesi gereken faaliyetler 'iş (job)' olarak adlandırılır ve bunlar birçok bileşenden yani 'iş elemanı' ndan (activity, operation) oluşmuştur. Her operasyon, belirli bir kaynakta, belirli bir süreyi 'işlem zamanı (processing time)' olarak gerektirir. İşler 'kaynaklar (resources)'da gerçekleştirilir.

Çizelgeleme problemleri çözüm yaklaşımları aşağıdaki gibi gruplandırılabilir:

Basit sezgisel yaklaşımlar: Bu yaklaşım, karar noktalarında sezgisel öncelik kurallarını kullanır. İşler ortama bu kurala göre aktarılır ve işlenirler, örneğin 'bu makine için (değer)/makine işlem zamanı oranı en yüksek olan işi seç' gibi. Burada parti büyüklükleri çoğunlukla sabittir.

Bilgisayar simülasyonu yaklaşımları: Bilgisayardaki gelişmeler sonucunda iş ortamının yapısı, aktiviteler, işler ve diğer bazı kısıtların detaylarıyla verilmesi, uygun verilerle ve basit sezgisel kurallarla, verilen dönem için bilgisayarların

çizelgelemeyi gerçekleştirmesi olası hale gelmiştir. Simülasyon, sistemin gerçeğe oldukça yakın varsayımlarla çizelgelenmesini sağlar. Dezavantajı ise matematiksel modellerdeki gibi optimal çözüm değil, yaklaşık çözümler vermesi ve mevcut çözümün ne kadar iyi olduğu veya daha iyisinin nasıl elde edileceğine dair iyi karşılaştırma yöntemlerinin olmamasıdır.

Matematiksel yaklaşımlar: Balas ve Gomory, 1960’larda, bilgisayarların artan güçlerini, çizelgeleme problemlerinin tam olarak çözülmesini sağlayacak teorik yöntemlerin formüle edilmesinde kullanmıştır. Problemlerin dal-sınır (branch and bound) algoritmaları ile çözümü gerçekleştirilebilir. 1960 ve 1970’lerde çizelgeleme problemlerinde dinamik programlama (dynamic programming) yöntemleri kullanılmaya başlanmıştır. Öte yandan, dinamik programlamada küçük problemler, en iyi öncelikleri, rotaları ve diğer istenenleri bulmak için çözülebilirken, daha büyük ve gerçekçi problemlerde çözüm büyüklüğü kontrol edilemez durumdadır, ve bu problemler ‘problem büyüklüğüyle üstel olarak büyümeyen bir çözüm metodu bulunamayan’ anlamına gelen “np-complete” olarak adlandırılırlar (Brucker, 1995). Bu yüzden bu büyük matematiksel programları yaklaşık olarak çözebilecek değişik sezgisel metotlar geliştirilmeye başlanmıştır, örneğin genel bir teknik olan ve çizelgeleme için Wilkerson ve Irwin (1971) tarafından kullanılan komşu bölge tarama (neighborhood search) yöntemi ve diğerleri gibi. Bu yöntem diğer bazı modern metotların da temelini oluşturmuştur. Lagrange serbest bırakma (Lagrangian relaxion) yöntemi bazı kısıtları, cezalarını da ödeyerek bırakmaya yönelik olarak daha basit tamsayılı programlama problemlerini çözer (Morton ve Pentico, 1993).

Daha yeni yaklaşımlar: Bazı pozisyonların aktif listede tekrar yer almayacağı (tabu olması) gerçeğinden ismini alan tabu arama (tabu search), komşu bölge tarama yönteminin temeline dayanır. Aynı temele dayanan ancak bazı eklemeler de içeren benzetim tavlama (simulated annealing), bahsedilebilecek bir diğer yöntemdir. Doğal gelişim prosesini deneyen araştırma prosesi olan genetik algoritma (genetic algorithm) de bir diğer yöntemdir. Beam search (Partial

Enumeration) da yapay zeka yazılım mühendisleri tarafından geliştirilmiş, karar ağaçlarının araştırılmasına dayanan bir yöntemdir.

Darboğaz metotları: Creative Output tarafından oluşturulan OPT®, darboğaz kaynakları bulmaya yöneliktir. Sistemin kalanı da darboğazı besleyecek şekilde yönlendirilirler. OPT tekniğinin amacı, darboğaz sisteme odaklanmasıdır.

Adams ve diğerleri tarafından geliştirilen Değişen Darboğaz (shifting bottleneck) prosedürü, makespan minimizasyonu amacıyla sınırlı olsa da, uygulamada başarı sağlayan bir yöntemdir. Metod, biri hariç tüm kaynakların her ilerlemede belirlenmiş sabit karar kuralına sahip olduğu daha basit bir sistemin çözümüyle uğraşır.

Çizelgeleme problemlerinin çözümünde önerilen matematiksel metodların optimal çözüme ulaşıyor olmaları avantaj iken, bunun için çok fazla hesaplama çabası ve zaman gerektirmeleri ve büyük problemlerde bu durumun daha da belirgin olması, oldukça karmaşık olan ve birçok kısıtın dikkate alınmasını gerektiren gerçek üretim problemlerinde, bilgisayarların da yardımıyla, sezgisel algoritmaların daha çok tercih edilir olmasına sebep olmaktadır.

1.1.2.İncelenen Çizelgeleme Problemi

Sipariş tipi üretim ortamlarının çizelgelenmesi, akış tipi üretim ortamlarının çizelgelenmesinden daha zordur. Çünkü sipariş tipi üretim ortamlarında iş merkezleri arasındaki akış, akış tipi üretim ortamlarının aksine, ürün tipine göre değişecektir, yani her ürünün kendine ait bir rotası bulunacaktır. Akış tipi üretim ortamlarında ekipmanlar sadece tek tip ürün için kullanılırken, sipariş tipi üretim ortamlarında ekipmanlar, sistemdeki birçok iş tarafından paylaşılacaktır. Bu ortamda, iş miktarlarının değişken olması ve farklı bileşenlerin farklı önceliklere sahip olması da sipariş tipi üretim ortamının çizelgelenmesinin karmaşık olmasına sebep olmaktadır (Selladuari, 1995).

Sistemde darboğaz niteliğinde kaynakların olması da problemi farklı boyutlara taşır. Kullanılan çizelgeleme yönteminin bu hususu dikkate alması sistemin toplam performansı açısından zorunludur.

Talep yapısına bağlı olarak, farklı kaynakların darboğaz haline gelmesi de özellikle sipariş tipi üretim ortamlarında rastlanan bir durumdur. Akış tipi üretimlerde de problem benzer nitelikler gösterir. Çizelgeleme, dikkate alacağı darboğazın değişmesi sebebiyle daha karmaşık bir yapıya bürünmektedir.

Bu çalışmada incelenecek çizelgeleme problemi, “değişken darboğaz kaynakları içeren sipariş tipi sistemlerde çizelgeleme” olarak tanımlanabilir.

1.2. Çalışmanın Amacı

Üretim ortamlarında, kısa dönemli kararlar arasında belki de en önemlisi, üretim çizelgelemenin gerçekleştirilmesidir. Sistem büyüdükçe ve karmaşıklaştıkça, bu işlemin gerçekleştirilmesi de zorlaşacak, daha fazla zaman alacak ve önemi de artacaktır.

Sipariş tipi üretim ortamlarında değişkenliklerin olması ve önceliklerin tanımlanması sebebiyle çizelgeleme işlemi önem kazanmaktadır. Sipariş tipi üretim ortamlarının genel özelliklerine bakacak olursak, n tane işin m adet iş merkezinde işlenmesi sözkonusudur. i işinin k makinesindeki işlem zamanı p_{ik} olarak, siparişin işlem görmesi için ortama bırakılma zamanı r_i , termin tarihi de d_i olarak belirlidir. Buna göre, sistemin bileşenleri aşağıdaki gibi gösterilebilir (Sipper ve Bulfin, 1997):

- C_i : i işinin tamamlanma zamanı
- F_i : i işinin akış zamanı, $C_i - r_i$
- L_i : i işinin gecikme (lateness) zamanı $C_i - d_i$

Gecikme, çizelgenin verilen termin tarihine uygun hareket edip etmediğini gösterir. Gecikme, eğer iş termin tarihinden önce bitirilmişse negatif değer alacaktır. Negatif değer, beklenenden daha iyi performans gösterdiğini, pozitif değer ise işlerin geç kaldıklarını gösterir. Birçok durumda cezalar ve diğer maliyetler pozitif değer ile

ilişkilendirildiklerinden, negatif gecikme elde edilmesi herhangi bir fayda sağlamaz.

Bu yüzden sadece pozitif gecikme zamanını ölçen bir ölçüt (tardiness) kullanılır.

T_i : i işinin pozitif gecikme (tardiness) değeri, $\max \{0, L_i\}$

E_i : i işinin negatif gecikme (earliness) değeri, $\max \{0, -L_i\}$

$\delta_i = 1$,eğer i işi pozitif gecikmiş ($T_i > 0$) ise

$\delta_i = 0$,eğer i işi zamanında veya erken bitirilmişse

$C_{\max}$: $\max \{C_i\}$ tüm işlerin tamamlanma zamanının maksimumu (*makespan* olarak da adlandırılır)

$L_{\max}$: $\max \{L_i\}$ tüm işlerin maksimum gecikme değeri

$T_{\max}$: $\max \{T_i\}$ tüm işlerin maksimum pozitif gecikme değeri

Bu tanımlamalara göre, üretim ortamının performans ölçütleri de aşağıdaki gibi olacaktır (Baker, 1974):

Ortalama Akış Zamanı : $\bar{F} = \frac{1}{n} \sum_{j=1}^n F_j$

Ortalama Gecikme : $\bar{T} = \frac{1}{n} \sum_{j=1}^n T_j$

Maksimum Akış Süresi : $F_{\max} = \max \{F_j\} \quad 1 \leq j \leq n$

Maksimum Gecikme : $T_{\max} = \max \{T_j\} \quad 1 \leq j \leq n$

Geciken İş Sayısı : $N_T = \sum_{j=1}^n \delta(T_j)$

$$\delta(x) = 1 \quad x > 0$$

$$\delta(x) = 0 \quad \text{diğer}$$

Her üretim ortamında gerçekleştirilmesi istenen performans ölçütü farklı olabilir. Genel olarak,

- gecikme süresinin minimizasyonu
- geciken iş sayısının minimizasyonu
- ara stok (üretim içi stok) düzeyi minimizasyonu

- akış zamanı azaltılması
- makine kullanım oranlarının ayarlanması
- hazırlık zamanlarının azaltılması

ve bunların maliyetleriyle ilgili ölçütler, performans ölçütü olarak alınabilir.

“Değişken darboğaz kaynakları içeren sipariş tipi sistemlerde çizelgeleme” de bir çözüm arayan bu çalışmada, geliştirilen ve test edilen algoritma ile gerçekleştirilmek istenen amaçlar şu şekilde sıralanabilir:

- Ortalama akış zamanlarının azaltılması
- Geciken sipariş sayısının minimize edilmesi
- Siparişlerin toplam gecikme sürelerinin minimize edilmesi
- Ara stok seviyelerinin azaltılması
- Makinelerin kullanım düzeylerinin ayarlanması

1.3.Çalışmanın Kapsamı ve Varsayımlar

1.3.1. Kapsam

İncelenmek istenen sistem, değişken darboğaz kaynakları içeren bir sistemdir.

Temel özellikleri aşağıdaki gibi sıralanabilir:

- Ürün çeşidi birden fazladır.
- Her ürünün kendine özgü bir rotası vardır ve bilinmektedir.
- Ürünlerin gelişler arası süresi deterministiktir.
- Ürünler sipariş büyüklükleri farklı olarak gelmektedir.
- Benzer işlemi gerçekleştiren birden fazla makine vardır.
- Her bir işlemin o makinedeki işlem süresi, birim başına olmak üzere, belirlidir.
- İşlerin termin zamanları belirlidir.
- Bazı makinelerde üretimler arasında hazırlık zamanları oluşmaktadır.

Sistem incelendiğinde bazı makinelerin zaman içinde darboğaz kaynak durumuna geldiği gözlenmektedir. Öncelikle darboğaz kaynak durumuna gelen makineler tespit edilmiş, bu makineler esas alınarak, geliştirilen algoritma çerçevesinde etkin bir çizelgeleme yapılmaya çalışılmıştır. Elde edilen sonuçlar, benzer üretim sistemleri için yaygın olarak kullanılan sıralama kuralları ile yapılan çizelgeleme sonuçları ile karşılaştırılmıştır.

1.3.2. Varsayımlar

İncelenen çizelgeleme probleminde, sistem ile ilgili olarak aşağıdaki temel varsayımlar yapılmıştır:

- Makinelerin her an üretime hazır oldukları yani makine arızaları söz konusu değildir
- Makineler arasındaki taşıma süreleri ihmal edilmiştir
- Gelen siparişler, tek bir parti halinde, bölünmeden üretilmektedirler
- Her makine belli bir anda sadece tek bir iş için kullanılabilir
- Her bir işlem sadece bir kez yapılmaktadır, kalite uygunsuzluğu gibi sebepler yüzünden geri dönüşler söz konusu değildir
- Sipariş, rotası gereği daha önce uğradığı bir iş merkezine ikincil bir işlem görmek üzere gönderilebilir
- Bir makinede üretim başladıktan sonra kesilemez ve durdurulamaz
- Sisteme gelen her siparişin sisteme girişine izin verilir, sipariş reddedilmesi söz konusu değildir.

1.4. Çalışmanın Adımları

Çalışma, aşağıdaki sırayı izlemektedir:

- (1) Problemin amacı, kısıtları, kriterler, varsayımlar ve performans değişkenleri belirlenecektir.
- (2) Probleme ilgili bilimsel yazın incelemesi gerçekleştirilecek, çalışmanın katkıları belirlenecektir.

- (3) İncelenen problem için geliştirilen çözüm algoritması ifade edilecektir.
- (4) Geliştirilen algoritmayı test etmek üzere sistemin performans kriterleri seçilen diğer çizelgeleme kuralları ile karşılaştırılacaktır.
- (5) Simülasyon deneyinde kullanılacak kontrol değişkenleri belirlenecek ve model tasarlanacaktır.
- (6) Her algoritma için simülasyon modeli SIMAN dili kullanılarak hazırlanacak ve modelin doğrulanması gerçekleştirilecektir.
- (7) Her algoritma için simülasyon modeli çalıştırılacak ve sonuçlar tablo ve grafikler halinde özetlenecektir.
- (8) Geliştirilen algoritmanın diğer sezgisel algoritmalarla karşılaştırması yapılacak, iyi sonuç verdiği veya zayıf olduğu durumlar belirlenecektir.
- (9) Bu çalışmada ele alınmayan varsayımlara ve ek problemlere uygun çözüm yaklaşımlarıyla, sonraki çalışmalar için önerilerde bulunulacaktır.

1.5. Tezin Organizasyonu

Tezin bundan sonraki bölümlerinde sırasıyla, ikinci bölümde literatürün genel değerlendirilmesi yapılacak, çalışmanın orijinal katkısından bahsedilecektir. Üçüncü bölümde, sistemin genel özellikleri belirlenecek, sistem tanımlanacaktır. Geliştirilen algoritma ve simülasyon deneylerinin oluşturulmasından bahsedilecektir. Karşılaştırma yapmak amacıyla sezgisel algoritmaların seçimi de bu bölümde gerçekleştirilecektir. Dördüncü bölümde geliştirilen algoritma ve diğer sezgisel algoritmaların simülasyon sonuçları sunulacak ve yorumlanacaktır. Son bölümde ise, sonraki çalışmalar için önerilerde bulunulacaktır.

2. ÖNCEKİ ÇALIŞMALAR

2.1. Çizelgeleme Algoritmaları

Panwalkar ve Iskander (1977), çizelgeleme kuralını bir veya daha fazla öncelik kuralı veya sezgisel algoritmanın bir kombinasyonu olarak tanımlayan çalışmada, kuralların sınıflandırması ve tanımlaması yapılmıştır. Jackson kuralları *statik* veya *dinamik* olarak sınıflandırırken, Conway ve Maxwell *lokal* ve *global* tanımlamasını yapmış, Moore ve Wilson ise, bir kuralı statik veya dinamik ve lokal veya global olmasını gösteren iki boyutlu ölçüt kullanmışlardır. Bu çalışmada ise çizelgeleme kuralları basit öncelik kuralları (SPT, LPT, EDD, MINSEQ, RANDOM gibi), basit öncelik kurallarının kombinasyonları, ağırlıklandırılmış öncelik indeksleri, sezgisel çizelgeleme kuralları ve bu sınıflara girmeyen diğer kurallar olarak gruplandırılarak açıklanmış ve literatürdeki kullanımları, problem tipi ve performans ölçütleri temel alınarak gösterilmiştir. Sınıflandırılan ve incelenen farklı çizelgeleme kuralları altında yatan genel kavramı açıklamanın, bazı açılardan oldukça zor olduğu görülmüştür. Bu durumun da, endüstriyel uygulamalarda performans kriterlerinin oldukça farklılaşması, işlem şartlarının ve varsayımların farklı oluşu gibi etkenlerden kaynaklandığı belirtilmiştir.

Melnyk ve diğerleri (1986), literatürde iş ortamlarında operasyonların ve siparişlerin en iyi nasıl yönetileceğine ilişkin çalışmalar oldukça yaygın olmasına karşın, pratikte bu kuralların geçerliliğine ilişkin çalışmalar oldukça az sayıdadır. Araştırmacılar ve uygulayıcılar arasındaki boşluk, çoğunlukla, araştırmacılar hakkındaki şu görüşlerden kaynaklanır: araştırmacılar, genellikle doğru probleme ulaşmada başarısızdırlar; önerdikleri çözümlerin uygulaması zor ve karmaşıktır; geliştirdikleri çözümler, uygulayıcılara yabancı gelecek terimlerle ifade edilmektedir; üretim ortamının gerçekleri çoğunlukla gözardı edilmiştir. Bu çalışma, çizelgeleme yöntemlerinin üretim ortamı üzerindeki etkilerini ve rollerini, uygulayıcıların bakış açısından tanımlamış, uygulayıcılar tarafından kritik olarak görülen ancak literatürde daha az önem görmüş kritik faktörler belirlenmiştir. Uygulayıcılara göre çizelgeleme ve sıralama, diğer üretim faaliyetlerinden ayrı tutulamayacak bir sistem aktivitesidir

ve beraberce ele alınmalıdır. Etkin çizelgeleme uygulanabilmesi için, etkin kapasite planlamasının da gerçekleştirilmesi gereklidir. Çizelgeleme yönteminin üretim ortamının özelliklerine de uygun olması uygulayıcılar tarafından istenen ve kritik görünen faktörlerdir. Kullanım kolaylığı, anlaşılır, açık, anlamlı ve geçerli olması uygulayıcıların üretim çizelgeleme ve sıralama tekniklerinden bekledikleri özellikler olmaktadır.

2.2. Kısıtlarla Yönetim Felsefesi

Schragwnheim ve Ronen (1990) yöneticilerin en kritik konulara konsantre olmalarını sağlayan ve birçok üretim organizasyonunda uygulanmaya başlayan DBR (Drum-Buffer-Rope) tekniğinin, TOC (Theory of Constraints) mantığına dayandığını söyleyen bu çalışmada, yedi tip hammaddenin işlenerek iki farklı ürüne dönüştüğü, 6 üretim kaynağının bulunduğu bir üretim ortamı tanımlanmış, talepler ve teslim zamanları belirliyken, bu ortamın DBR tekniğiyle nasıl çizelgelenebileceği gösterilmiş ve model simülasyonla çalıştırılarak denenmiştir.

Fawcett ve Pearson (1991) çalışmalarında kısıtlarla yönetimin temel amacının sürekli gelişen bir süreç oluşturmak olduğunu belirtmişlerdir. Eşzamanlı (senkronize) üretim, talebi karşılamak amacıyla malzemelerin üretim kaynaklarından çabukça ve düzgün bir şekilde hareket etmesi için sistematik bir metod olarak tanımlanabilir. Kısıtlarla yönetim, kısıtlı kaynakların yönetiminin geliştirilmesi ve bu kritik kaynakların operasyonlarının çizelgelenmesiyle senkronizasyona ulaşılmasını amaçlar. Burada kısıtlı kaynaklar belirlenip, yönetimin odaklanma mekanizması olarak kullanıldığında, yönetim kararları (ürün maliyetlendirme, performans ölçümü, ürün ve proses tasarımı, yap-satınal kararları) değişir. Bu değişimlere göre kısıtlarla yönetim, üretim kaynaklarının koordinasyonuna ulaşmada firmaya yardımcı olur.

Kısıtlı kaynak atılken çıktı kaybı olacaktır. Benzer şekilde kısıtlı olmayan kaynaktaki fazla kapasite, kısıtlı kaynağın üretebileceğinden daha fazla üretim yapıyorsa, stoklara eklenecektir. Bu da, stokta ve maliyette artış, daha uzun üretim zamanları demektir. Kısıtlarla yönetimin uygulanması için sistematik bir yaklaşım izlenmeli, işletme analizi yapılmalı, amaçlar, problemler ve politikalar

belirlenmelidir. Bu yöntemler, tüm kaynaklarda etkinlik sağlayacak, firmayı sınırlayan tüm engellerin farkedilmesini ve çözümlenmesini sağlayacak ve firmaya rekabet avantajı sağlayacaktır.

Spencer (1991) , OPT ve TOC felsefelerinden bahsederek, MRP sisteminin kurulu olduğu bir işletmede, MRP paketinin normal çıktı olarak verdiği raporların değerlendirilerek Kısıtlar Teorisi yaklaşımları ile DBR tekniğinin ve elemanlarının nasıl uygulanabileceğini anlatmıştır. Sipariş tipi ortamlarda DBR tekniğinin uygulanabildiği gibi, tekrarlı üretimde de bu mantığın uygulanmasının olası olduğu söylenmiştir. Bu tip çevrelerde senkronize üretimin avantajlarına ulaşılmasının, sistemdeki birkaç değişiklik ve yönetim felsefesindeki değişimle mümkün olabileceği belirtilmiştir.

Zapfel ve Missbauer (1993) özellikle son otuz yıldır bilgisayar yardımına başvuru ve üretim firmalarının rekabet pozisyonları için büyük bir önemi bulunan üretim planlama ve kontrol yöntemlerinin en yaygın bilinenlerinin karşılaştırılmasını yapmışlardır. Üretim planlama ve kontrol sistemlerinin temel fonksiyonları,

- Müşteri siparişlerine veya talep tahminlerine dayanarak ne kadar nihai ürüne ihtiyaç duyulduğunun belirlenmesi,
- Malzeme ihtiyaçlarının, üretim parti büyüklüklerinin, bileşen ve hammadde için geçecek zamanın belirlenmesi,
- Envanter muhasebesi,
- İşlerin çizelgelenmesi ve sıralanması,
- Kapasitelerin planlanması ve dengelenmesi,
- Siparişlerin bırakılma zamanları (order release),
- Amaç performansın kontrolü ve eğer sapmalar varsa önlemler alınması

olarak sıralanabilir.

Uzun üretim zamanı, yarı mamül ve nihai ürün stoklarının azaltılma gerekliliği, daha iyi ve daha yeni üretim planlama ve kontrol sistemlerinin geliştirilmesini gerekli kılmıştır. Geleneksel üretim planlama ve kontrol sistemlerinden sonra, işyükünü de dikkate alan sistemler; tüm malzeme akışı kontrolünü pazarlama, finans ve operasyonlar planı ile birleşmesine dayanan hiyerarşik yapıda bir bilişim sistemi olan MRP II sistemi; olası tüm stoklardan

kaçınmayı ve üretim prosesinin sürekli iyileşmesini amaçlayan JIT sistemi ve malzeme akışının yani üretim sisteminin performansının sistemdeki darboğazlar tarafından belirlendiği mantığına dayanan OPT sistemi gibi yeni üretim planlama sistemleri geliştirilmiştir. Kısıtlar Teorisi felsefesini temel alan OPT sistemi, darboğaz kaynaklar için DBR tekniğini kullanarak çizelge oluşturan ve bu çizelgeye dayanarak siparişlerin de çizelgesini oluşturan sınırlı yükleme prosedürüdür. Tüm üretim planlama ve kontrol sistemlerinin geldiği nokta, hangisinin en iyi olduğu sorusudur. Ancak bu, üretim ortamına göre değişecektir. Ürün yapısı, ürün değişkenliği, üretim hacmi, talep dalgalanmaları, üretim teknolojisi gibi faktörler, üretim ortamlarını birbirinden farklı kılar. Üretim ortamlarının sınıflandırılması pratik deneyimlere ve teorik temellere dayanır. Farklı yöntemlerin performans kriterleri açısından değerlendirilmesi, analitik veya simülasyon modelleri yardımıyla gerçekleştirilebilir. Ancak bu deneyler de üretim ortamının tüm karmaşık yapısını yansıtmayacağından ve sınırlı varsayımlara dayanacağından, en iyi sistemin veya olası kombinasyonların ve sonuçların gözlenmesinin tam olarak sağlanamadığı söylenmiştir.

Rodrigues ve Mackness (1998), üretim yönetimi amacıyla geliştirilen tüm yazılımların amacının fabrikadaki iş akışının senkronize olarak yönetiminin sağlanması olarak belirtilmişlerdir. Senkronizeleşmiş bir sistemde bileşenler iş merkezlerine üretim için gerektiği zamanda ulaşırlar. Böylece siparişlerin gecikmesine, yüksek stok maliyetlerine sebep olan süreç içi stokların oluşması da engellenmiş olur. Stok azaltılmasının bir diğer faydası da ürün kalitesinde artış sağlanmasıdır. Böylece üretim senkronizasyonu, rekabet avantajı kazandıracak bir kaynak olarak görülebilir. JIC (Just in Case) olarak da bilinen klasik üretim yönetiminde işler bir istasyondan diğerine itilirler. Burada, bileşenin sonraki istasyonda ne zaman kullanılacağı dikkate alınmaz. Bu tip yönetimin sonucu, yüksek oranda ara stok, maliyet ve geciken siparişlerdir.

Bu tip problemleri çözmek amacıyla üretim aktivitelerini iyileştirecek ve üretim sistemini senkronize edecek yaklaşımlar geliştirilmiştir. Bu yaklaşımlardan biri olan JIT (Just in Time), gereksiz stoğu ve gereksiz üretim zamanını sıfırlamaya, işçi verimliliğini artırmaya yönelik bir talep odaklı çekme sistemidir. Bileşenler,

üretim için tam gerekli oldukları zamanda üretilir ve pazardan, hammaddeye doğru bir çekme sistemiyle iletilirler. Stok düzeyi ve üretim zamanı düşüktür, ancak çıktı miktarı makine bozulmaları gibi durumlarda risk altındadır.

TOC, firmanın amacına ulaşmasını sağlayacak bir dizi prensip ve prosedür içerir. Senkronize üretimi amaçlayan TOC, itme ve çekme sistemlerinin bir bileşimi olan DBR yöntemini kullanır. Her işletmede darboğaz kaynaklar vardır ve bunlar belirlendikten sonra bu kaynakların maksimum oranda kullanılabilmesi için diğer kaynaklarla iletişim sağlanır.

Bu çalışmada, bahsedilen üç farklı yaklaşımın, JIC, JIT, DBR, karşılaştırılması amacıyla VIMS (visual interactive modelling system) kullanılmıştır. Herbir yöntem için beşinci makinesi darboğaz olan, işlem zamanları belirli, 7 aşamalı bir montaj hattı oluşturulmuştur. Performans ölçütleri olarak da 100 ürün için toplam üretim zamanı, ortalama bekleme zamanı, ortalama üretim zamanı, darboğaz kullanım oranı ve girdi/çıktı kontrolü değerlendirilmiştir. En yüksek üretim oranı, düşük bekleme zamanı ve en düşük ortalama üretim zamanı açısından en iyi çözüm JIT olduğu görülür. Ardından, JIT'a benzer sonuçlarla DBR iyi bir sonuç verirken, JIC yöntemi girdi/çıktı kontrolünde, yüksek stok oranıyla en kötü çözümü vermiştir. Sistemde kullanılan kanban sayıları değişmesi veya varsayımlara makine bozulmaları eklenmesi durumlarında, JIT sisteminin bunlardan direkt etkilendiği ve çıktı miktarında değişimler olduğu görülür. Bunun sebebi darboğaz makinenin etkin kullanılamamasıdır. Bu olası durumlarda DBR sisteminin makine kullanım oranı ve bekleme zamanları açısından en etkin yöntem olduğu görülmüştür.

2.3. Darboğaz Kaynağın Yönetimi

Wassenhove ve Vanderhenst (1983) çalışmalarında, bir dizi darboğaz makine için üretim planı geliştirmeye çalışmışlardır. Ürün talebi zaman bağlı ve deterministik, makine başlama/bitiş maliyetleri çok yüksek ve hazırlık zamanları da sıralamaya bağlıdır. Problemin çözümünde ilk olarak başlama/bitirme ve stok bulundurma maliyetleri arasındaki ödünleşim ile, sezgisel algoritma yardımıyla üretim hacmi belirlenir. Her makinede gezgin satıcı problemi çözülerek optimum

üretim sıralaması bulunarak, maliyet açısından optimize edici olmayan ancak duyarlılık analizi için kullanılabilir bir model oluşturulmuştur.

Neely ve Byrne (1992), çalışmalarında bilgisayar simülasyonu kullanarak OPT felsefesini denemiş ve sonuçlarını göstermişlerdir. Çizelgeleme yapılırken darboğazların dikkate alınmasına dayalı bu sistemin, performans üzerinde olumlu etkileri vardır. OPT felsefesi her işletmede en az bir darboğazın bulunacağı varsayımına dayanır. Darboğaz kaynak ise, “ürünü üretmek için kullanılan kaynak zincirinde en yavaş makine veya kaynaktır. Çıktı miktarı bu tek makinayla belirlenir. Bunun üretimini artırmak, ürünün çıktısını artıracaktır” (Goldratt ve Cox) şeklinde tanımlanmıştır.

Çalışmada, darboğaz kaynağın bulunduğu bir üretim ortamında değişik çizelgeleme algoritmalarının nasıl çalışacağı denenmektedir. Sistem, 5 nolu makine darboğaz olmak üzere 9 makineden oluşmaktadır. Herbiri farklı ürünler olan 20 işlik yığınlar, hafta başında iş ortamına yüklenmektedir. Veriler deterministiktir, taşıma zamanları ve makine bozulmaları dikkate alınmamıştır. Araştırma için 6 kural seçilmiştir:

- FIFO,
- Toplam işlem zamanı,
- Yığın işlem zamanı,
- Yığın işlem+hazırlık zamanı,
- Toplam işlem zamanı/yığın işlem zamanı,
- Toplam işlem zamanı/ yığın işlem+hazırlık zamanı

Algoritmaların etkinliklerinin ölçülmesi için, OPT felsefesinin ölçütlerine benzer ölçütler kullanılmıştır: İşleri tamamlamak için geçen süre (akış zamanı), kuyruk uzunluğu ve makine kullanım oranı.

Simülasyon deneylerinde en iyi sonuçlar, darboğazdaki iş yükü en az olan yığının yüklenmesi yöntemiyle elde edilmiştir. Bu yöntem etkindir çünkü darboğazın önünde kuyruk oluşturacak işlerin hızla itilmesi sözkonusudur. Bu durumda stok düzeyi ve işleri bitirmek için gereken süre azalacak ve gecikmeler engellenmiş olacaktır.

Cook (1994), bu çalışmada, darboğaz kaynak içeren akış tipi üretim ortamında, geleneksel yöntemler, JIT ve TOC sistemlerinin etkinlikleri simülasyon deneyleri kullanılarak karşılaştırmıştır. Tek ürünlü ve hazırlık zamanının olmadığı, akış tipi 5 istasyonlu bir sistemde, üretim zamanları belirli dağılımlarla alınmış ve hammadde tedarik sorununun olmadığı varsayılmıştır. Performans ölçütleri olarak sistemin toplam çıktısı, akış zamanı, akış zamanının değişkenliği ve ortalama ara stok düzeyi alınmıştır. Değişken üretim hacimleri için yöntemler karşılaştırılmıştır. TOC, JIT'dan daha fazla çıktı elde etmiş ve üretim hacmi büyüdükçe ortalama ara stok düzeyi de artmıştır. Ancak üretim planlama sistemleri arasında çok önemli bir değişkenlik sergilenmemiştir. Tüm sistemlerde stok arttıkça, toplam işlem zamanının arttığı ve, TOC kullanan sistemde akış zamanı değişkenliğinin en az olduğu gözlenmiştir. Sonuçta TOC ve JIT sistemlerinin bazı açılardan yapısal olarak benzerliği belirtilmiştir.

Bassett ve Todd (1994): Birçok üretim planlama müdürü tarafından SPT öncelik kuralı, sipariş tipi ortamlarda en etkin çizelgeleme kuralı olarak görülür. Bazı çalışmalar da, iş başına düşen ortalama akış zamanı açısından SPT kuralının diğer kurallardan daha etkin olduğunu göstermiştir. Sipariş tipi ortamlarda, akış tipi ortamlarda olduğu gibi, kapasite dengelenemez. Mümkün olduğunca akışın dengelenmesi konusunda çalışılmalıdır. Kısıtlı kaynağın bulunduğu ortamlarda Goldratt'ın da söylediği gibi, darboğaz kaynakların kısıtlı kapasitelerinin doldurulması gereklidir. Hipotezi test etmek amacıyla da, üç istasyonluk bir üretim ortamı alınmış ve simülasyon yöntemiyle denenmiştir. İşler, her üç istasyondan da sırasıyla geçmektedirler. İşlem süreleri, sürekli düzgün dağılıma uygun olarak, ilk istasyonda 1 ± 0.5 ; ikinci istasyonda 0.5 ± 0.25 olarak değişirken, darboğaz olan üçüncü istasyonda işlem süresi 2.7 ± 1.35 ile uzun veya 1 ± 0.5 ile kısa süreler arasında sürekli düzgün dağılıma göre değişmekte ve kullanım oranları da %62.5, %52 ve %100 olarak gözlemlenmektedir. Bu ortamda, darboğaz olan üçüncü makine için SPT kuralı ile darboğazdaki işlem süresi en uzun olana öncelik verme kuralları uygulanmış ve sistemler, ortalama akış zamanı, tamamlanmamış iş oranı, darboğaz istasyonun kullanım oranları açısından karşılaştırılmıştır. Ortalama akış zamanı SPT için iyi iken, aynı zamanda tamamlanmamış iş sayısı daha fazladır. SPT

kullanılmadığı durumda darboğaz istasyonunun kullanım oranının daha yüksek olduğu görülmüştür. SPT kuralının kronik bir darboğaz durumu yarattığı söylenebilir.

Grosfeld-Nir. A (1995) yaptığı çalışma, biri haricinde tüm makinelerin hazırlık maliyetinin 0 olduğu çok aşamalı bir üretim sistemini gözönüne almakta ve hazırlık maliyetinin sıfır olmadığı aşama darboğaz olarak tanımlanmaktadır. Eğer üretim maliyetleri hazırlık ve üretim zamanına bağlıysa bu darboğaz, kritik kaynak (tüm kapasitesiyle kullanılan kaynak) olarak düşünülebilir. Bu da, tüm sistemin çıktı miktarını etkileyecek kaynaktır. Böyle bir durumda, darboğaz olmayan kaynağın kazanılan veya kaybedilen hazırlık zamanı sistemin karını önemli bir şekilde etkilemeyecektir. Böylece tüm hazırlık zamanları -kritik kaynağına hariç- ihmal edilebilir maliyet olarak düşünülebilir. Böylece tek bir kritik kaynağa sahip sistem, bu şekilde modellenebilir.

Gunn ve Nahavandi (2000), çalışmalarında, birçok firmanın yüzyüze kaldıkları ikilemin, çıktı miktarı ve üretim zamanı arasındaki ödünleşim olduğunu belirtmişlerdir. Yüksek oranda çıktı elde edebilmek için makinelerin aç kalmasını önleyecek yüksek oranda ara stok (WIP) bulundurulmalıdır. Kısıtlar teorisinde Goldratt ve Cox (1986), bunun üretim koordinasyonu zayıflığından kaynaklandığını belirtirler. Little kuralı olarak adlandırılan kural da,

$$WIP = \text{çıkıtı} \times \text{üretim zamanı}$$

eşitliğini doğrular. Bu da çıktı düzeyinin uzun üretim zamanları ve yüksek ara stok düzeyi veya kısa üretim zamanları ve düşük ara stok düzeyi ile elde edilebileceğini gösterir. Burada problem, üretim zamanını azaltırken çıktı düzeyinin en iyi nasıl korunacağıdır. Bu noktada ara stok düzeyi optimum veya optimuma yakın düzeyde olacaktır. Bu sonucu elde etmek için darboğaz makinenin önünde bekleyen ürünlerin yönetimi kritik olmaktadır. Kritik kaynaklarda arastok düzeyinin minimum olmasını sağlamak, tüm fabrikada ara stok düzeyinin düşük olmasını, böylece üretim zamanının da minimum olmasını sağlayacaktır. Kısa üretim süresi firmaya esneklik sağlayacak, rekabet gücünü artıracaktır. Bunun için öncelikle darboğaz makinelerin yeri belirlenmelidir. Bu darboğazların arastokları, fabrikanın ara stok düzeyini sınırlamak için belirli bir limitte tutulmalıdır. Bu limit, çıktı düzeyi önceden

belirlenmiş bir değerin altına düşene dek azaltılabilir. Bu noktada üretim zamanı minimuma yakın, ara stok düzeyi de optimumdur.

Bu çalışmada ele alınan sistem, birçok farklı tipte ürünün üretildiği, yığın üretimin yapıldığı 7 aşamalı bir üretim ortamıdır. Bu ortamda simülasyon modeli oluşturarak çıktı düzeyini kontrol eden ve buna bağlı olarak ara stok düzeyini düşüren algoritma yardımıyla, darboğaz makinelerdeki ara stok düzeyleri kullanılarak optimum ara stok düzeyi bulunmuş ve karşılaştırma yöntemi olarak alınan CONWIP sisteminden daha etkin sonuçlar verdiği görülmüştür. ara stok düzeyi %40 oranında ve akış zamanı da neredeyse yarısı kadar azaltılmıştır.

2.4. Darboğaz Kaynağın Değişimi

Moss ve Yu (1999): Bu çalışma, belirli üretim parametreleriyle darboğaz değişiminin (bottleneck shiftiness) tahmin edilmesine yöneliktir. Bazı üretim ortamlarında, yapılan işin özelliğine göre özel bir operasyon birgün darboğazken, bir başka gün darboğaz olmayabilir. Burada darboğazın belirlenmesi ve kaynakların buna göre dağıtılması, yöneticiler için oldukça önemlidir. Bu çalışmanın amacı, bazı belirli sistem parametrelerine (darboğaz ve darboğaz olmayan kaynaklarda üretim zamanları, iş geliş oranı, ortamın tipi ve büyüklüğü, darboğazın pozisyonu gibi) dayanarak darboğaz değişiminin tahmininde yardımcı olmaktır. Çeşitli sistem parametrelerinin darboğaz değişimi üzerindeki etkilerini ölçmek ve denemek için simülasyon çalışması yapılmıştır.

Simülasyon modeli, geliş oranı markov proses, işlem zamanları poisson dağılımı olacak şekilde tasarlanmış, üç farklı geliş oranı alınmış ve, 3, 5 ve 7 iş merkezli üç iş ortamı kullanılmıştır. Darboğazların yeri de üç farklı yerde alınmış. üretim ortamı için de iki farklı tasarım yapılarak, sonuçta oluşan 54 kombinasyon denenerek simülasyon modeli çalıştırılmıştır. Her bir çalıştırmanın sonunda da darboğaz değişimi değeri hesaplanmıştır.

Elde edilen sonuçlar, korelasyon analizi ve çoklu regrasyon kullanılarak analiz edilmiş ve matematiksel model oluşturulmuştur. Darboğaz değişiminin, darboğaz kaynaktaki ortalama işlem süresi ile negatif; iş merkezlerinin sayısı, iş geliş

oranı ve darboğaz olmayan kaynaktaki işlem süresi ile pozitif ilişkili olduğu görülmüştür. Darboğaz değişiminin küçük olması, daha sabit darboğazlar olduğunu gösterecektir.

Lawrance ve Buss (1994) yaptıkları çalışmada, iş merkezlerindeki darboğazın zaman içinde değişmesi olarak tanımlanan darboğaz değişimi (bottleneck shiftiness) durumunun, Jackson ağı modeli kullanılarak olasılıklarının bulunması ve simülasyon deneyi kullanılarak da, yapılan tanımlamaların yansızlığını gösterilmişlerdir. Darboğaz değişim ölçütü için hem kısa, hem de uzun dönemlik özellikler dikkate alınmıştır. Ele alınan Jackson ağında n adet tek aşamalı üretim merkezi vardır. Her iş merkezi μ_k oranlı bir üstel servis zamanı dağılımına sahiptir. Sisteme dışarıdan gelişler bağımsız poisson prosesine göre, γ_k oranlıdır. i merkezinde işini tamamlayan iş p_{ij} olasılıkla j iş merkezine gider veya $1 - \sum p_{ij}$ olasılığı ile sistemi terkeder. k işmerkezi için toplam varış oranı, $\lambda_k = \gamma_k + \sum p_{ik} * \lambda_i$ şeklindedir. Bu varsayımlar altında bir merkezin o andaki darboğaz olma olasılığı, şu şekilde belirtilir:

$L_{q1}, \dots, L_{qn}$, durağan durumda, ağın her düğüm noktasında kuyruktaki müşteri sayısı vektörü ve $\rho_i = \lambda_i / \mu_i$ olsun. Bu durumda k makinesinin özel (exclusive) darboğaz olma olasılığı,

$$\Pr\{L_{qk} > L_{qj}, j \neq k\} = \rho_k^2 - \sum_{j \neq k} \frac{(1 - \rho_k) \rho_j^2 \rho_k^2}{1 - \rho_j \rho_k} + \sum_{i \neq j \neq k} \frac{(1 - \rho_k) \rho_j^2 \rho_i^2 \rho_k^2}{1 - \rho_i \rho_j \rho_k} + \dots + (-1)^{n-1} \frac{\prod_{j=1}^n (1 - \rho_k) \rho_j^2}{1 - \prod_{j=1}^n \rho_j}$$

Ve kapsayan (inclusive) darboğaz olma olasılığı da,

$$\Pr\{L_{qk} \geq L_{qj}, j \neq k, L_{qk} > 0\} = \rho_k^2 - \sum_{j \neq k} \frac{(1 - \rho_k) \rho_k^2 \rho_j^2}{1 - \rho_j \rho_k} + \sum_{i \neq j \neq k} \frac{(1 - \rho_k) \rho_k^2 \rho_i^2 \rho_j^2}{1 - \rho_i \rho_j \rho_k} + \dots + (-1)^{n-1} \frac{(1 - \rho_k) \rho_k^2 \prod_{j \neq k} \rho_j^2}{1 - \prod_{j=1}^n \rho_j}$$

olarak verilebilir. Buna göre, bir iş merkezinin kullanım oranı düşük olsa da, bazı zamanlarda darboğaz olacaktır denilebilir. Darboğaz değişimini ölçebilmek için skalar bir ölçüt geliştirilmiştir. Bu ölçüt, tüm iş merkezleri arasından kapsayan darboğaz olasılığının değişim katsayısı ile, eğer bir istasyon 1 olasılıkla kapsayan darboğaz ise, darboğaz olma olasılığının değişim katsayısını karşılaştırır. Bu durumda darboğaz değişim ölçütü β , şu şekildedir:

$$\beta = 1 - \frac{cv}{\sqrt{n}}$$

burada,

- n iş merkezleri sayısı
- cv n iş merkezinin kapsayan darboğaz olasılığının değişim katsayısı
- $\sqrt{n}$ 1, 1 ve $(n-1)0$ ' in vektörünün değişim katsayısı ve sonuçta da,
- β varsayımlar altındaki ortamın ve tek baskın darboğazlı ideal durum arasındaki göreceli farkı ölçer.

β , 0 ile 1 arasında değer alabilir. 0, sistemde tek bir baskın darboğazın olduğu ve bunun değişmediği durumu ifade ederken, 1, tüm iş merkezlerinin darboğaz olma olasılığının eşit olduğu durumdur.

β ölçütü, ortalama akış zamanı veya ara stok düzeyi gibi sistem performansı için bir ölçüt değildir, ancak tüm üretim sisteminin planlama ve kontrol kompleksliği için iyi bir ölçüttür.

Bulunan darboğaz olasılıklarının geçerli ölçütler olup olmadığını test etmek amacıyla, üç üretim ortamında simülasyon deneyi yapılmıştır. İşlerin sırasıyla geçtiği dört makineli, işlerin üstel 0.8 dağılımına göre sisteme girdikleri, işlem zamanları şekil (shape) parametresi $\alpha = 4$, ve ortalaması da, darboğaz için 1.1 ve diğerleri için 1.0 olan gama dağılımına uyan akış tipi üretim ortamında, makine kullanım oranları darboğaz için %88 ve diğerleri için %80 düzeyindedir. Yine dört makineden oluşan ve işin diğer makineye gitme olasılığının $p_{ij} = 1/4$, $j = 1, 2, 3, 4$, $i \neq j$ ise, ve $p_{ii} = 0$, aksi halde olarak tanımlanabilecek sipariş tipi üretim ortamında, iş geliş oranları üstel 0.8 ve işlem zamanları da akış tipi üretim ortamında olduğu gibi gama

dağılımındadır. Ele alınan üçüncü sistem, 10 ürünli ve 13 istasyonlu, işlerin rotalarının bilindiği bir sistemdir. İşlem zamanları, işlem kolaylığı açısından gama dağılımına çevrilmiştir. Bu üç örnek için simülasyon programının çalıştırılması sonucunda, darboğaz değişiminin belirlenmesinde, modelde tanımlanan ve simülasyonla elde edilen β ölçütlerine oldukça yakın sonuçlar verdiği görülmüştür. Bu durumda modelin geçerli olduğu söylenebilir. Bu adımlardan sonra çalışmada, yöneticilerin darboğaz problemi için uyguladıkları yöntemlerin darboğaz değişimine etkilerinden ve yapılması gerekenlerden bahsedilmiştir.

2.5. Literatürün Genel Değerlendirmesi

Literatür genel olarak incelendiğinde, özellikle sipariş tipi üretim ortamlarında karşılaşılan darboğaz iş merkezlerini veya kaynakları çizergelemek için çeşitli yöntemler önerildiğini görülebilir. Ancak bu çalışmalar, çoğunlukla, tek ve belirli (sabit) darboğazların olduğu ve bunların değişmediği sistemler için sunulmuştur. Bahsedilen çalışmalar da, daha çok bu tip sistemlerde hangi tip sezgisel algoritmaların daha etkin çözümler vereceği konusunda yoğunlaşmış durumdadır.

Darboğaz kaynakların yönetiminde, işletmedeki darboğaza odaklanmaya dayanan 'Kısıtlar Teorisi' yaklaşımı, literatürde oldukça ayrıntılı açıklanmış ve uygulama örnekleri anlatılmıştır. Bu felsefenin mantığına dayanan ve bazı büyük firmalar tarafından da kullanılan ve başarılı sonuçlar elde edilen OPT paketi ve darboğazları yöneten DBR tekniği hakkında, literatürde yapılmış çalışmalar bulunmaktadır. Ancak her iki yöntem de, sistemde baskın ve tek bir darboğaz olduğunda etkin çalışmaktadır. Birçok işletmede, sistemdeki değişkenlere bağlı olarak, baskın darboğazdan çok, farklı zamanlarda, farklı iş merkezlerinin darboğaz olması durumu oluşmaktadır. Değişken darboğazlı sistemler ve bunların yönetilmesine ait, literatürde yeterli çalışma bulunmamaktadır.

Bu çalışmada da yapılmak istenen, değişken darboğazlı sipariş tipi bir üretim sisteminde, DBR mantığını temel alan geliştirilmiş bir algoritma ile, kısıtlara odaklanma yaklaşımının etkin kullanımının sağlanmasıdır.

3. MATERYAL VE METOD**3.1. Materyal**

Darboğaz kaynakların değiştiği sipariş tipi üretim ortamlarında, darboğaz kaynağa dayanarak çizelgelemenin araştırıldığı bu çalışmada, çizelgeleme için uygun algoritmaların geliştirilmesine çalışılmıştır. Geliştirilen algoritmaların test edilmesi amacıyla, belirlenen performans değişkenleri için seçilen sezgisel çizelgeleme kurallarıyla karşılaştırması yapılmıştır.

Algoritma ve diğer sezgisel algoritmaları simülasyon yöntemiyle karşılaştırılması aşamalarında, sonuçların gerçek sistemlere de uygulanabilirliğini sağlamak amacıyla, mevcut bir sistemin verilerinin kullanılmasının daha uygun olacağı düşünülmüştür. Bu amaçla da, sipariş tipi üretim sistemi özelliği taşıyan bir tekstil işletmesinin terbiye bölümünün verileri kullanılmıştır (NAS, 1998). Üretim ortamının özelliği, çalışmanın amaçlarına oldukça uygundur. Değişik tipte ürünler üretilmektedir, üretim aşamasında ürünler çok sayıda iş merkezinden geçmekte ve rotaları gereği bir iş merkezini birden fazla defa kullanmak durumunda kalmaktadırlar (Ek 1). Her iş merkezinde, makine sayıları değişkendir. Üretim zamanları, her iş merkezinde ürün cinsine göre değişmektedir ve gelen siparişlerin büyüklükleri de değişkendir. Bu sistemde, ortalama iş akış zamanının, geç kalan iş sayısının, toplam geç kalma süresinin, ortalama ara stok düzeyinin düşürülmesini amaçlayan çalışmada, aşağıda sıralanan veriler kullanılmıştır:

- Siparişlerin geliş sıklıkları,
- Sipariş tipleri ve geliş olasılıkları,
- Sipariş büyüklükleri ve geliş olasılıkları,
- Her siparişin termin zamanı,
- Her tip ürünün izlemesi gereken rota,
- Her iş merkezindeki makine sayısı,
- Her tip ürün için her makinedeki birim işlem süresi,
- Siparişlerin özelliklerine bağlı olarak, belirli makinelerde gereksinim duyulan hazırlık zamanları.

Bu verilere Bölüm 3.2.8.1 'de detaylı olarak yer verilmiştir.

3.2. Metod

3.2.1. Sistemin Karakteristikleri

Çalışmada kullanılan sistem, m adet iş merkezi bulunan, d_i termin zamanı ve R_i rotası belirli n kadar iş gelişlerinin rastgele olduğu sipariş tipi bir üretim sistemidir. Ele alınan sistemin özellikleri ve varsayımları aşağıdaki gibi sıralanabilir:

- Siparişlerin sisteme gelişleri rastgeledir.
- Siparişlerin büyüklükleri değişken olabilmektedir.
- Siparişlerin reddedilmesi sözkonusu değildir.
- Siparişlerin rotaları belirlidir.
- Siparişler rotası gereği aynı iş merkezine tekrar uğrayabilirler.
- Siparişler, sisteme girdikten sonra termin zamanları belirlenir.
- Her iş merkezinde benzer işlemi yapan birden fazla makine bulunmaktadır.
- Belirli makinelerdeki hazırlık zamanları, ürünün rengine ve tipine göre değişir.
- Bir iş merkezindeki işlemi tamamlanan sipariş, rotasının gerektirdiği bir sonraki iş merkezine gönderilmektedir.
- Bir işlem, ondan önce gelmesi gereken işlemler tamamlanmadan başlayamaz.
- Her makine belli bir anda sadece tek bir iş yapabilir.
- Makinede üretim başladıktan sonra kesilemez.
- Her tip ürün için her makinedeki birim üretim zamanları belirlidir.
- İş merkezleri arasındaki taşıma süresi ve makine bozulmaları ihmal edilebilir.

3.2.2. Performans Değişkenleri

Hem literatürdeki çalışmalarda, hem de gerçek üretim ortamlarında çizelgelemede amaçlanan çok sayıda performans değişkeni vardır. Literatürde bu amaçlar,

(i) tamamlanma zamanına dayalı kriterler

- ortalama akış zamanının (F) azaltılması
- maksimum akış zamanının ($F_{\max}$) azaltılması
- ortalama tamamlanma zamanının (C) azaltılması
- maksimum tamamlanma zamanının ($C_{\max}$, makespan) azaltılması

(ii) işin termin zamanına dayalı kriterler

- ortalama pozitif gecikme değerinin (T) azaltılması
- maksimum pozitif gecikme değerinin ($T_{\max}$) azaltılması
- ortalama gecikme zamanının (L) azaltılması
- maksimum gecikme zamanının ($L_{\max}$) azaltılması
- geç kalmış iş sayısının (N_T) azaltılması

(iii) stok ve kullanım oranına dayalı kriterler,

- tamamlanmamış veya bekleyen iş sayısının azaltılması
- ortalama veya maksimum makine atıl zamanının azaltılması

şeklinde sıralanabilir ve eğer işler eşit ağırlıkta değilse, sayılan değerlerin ağırlıklandırılmış ölçütleri veya maliyet değerleri de ele alınabilir (Baker, 1974; French, 1982; Brucker, 1995).

Uygulamada gerçek sistemlerde ise, asıl amaç müşteri tatmini şeklinde görülmektedir. Burada gelen siparişlerin reddedilmemesi, istenen termin tarihlerinin sağlanabilmesi önem kazanmaktadır. Bunun için de üretim ortamında makinelerin etkin kullanılabilmesi, akış zamanlarının düşürülmesi, ara stokların olabildiğince azaltılması gerekmektedir.

Bu çalışmada ele alınan performans değişkenleri;

- ortalama akış zamanı,
- ortalama ara stok düzeyi,
- geç kalan iş sayısı,

- geç kalma süresi,
- pozitif gecikme,
- geciken iş oranı

olarak sıralanabilir.

3.2.3. Sezgisel Çizelgeleme Algoritmaları

Çizelgeleme problemlerinde uygulamada ve literatürde kullanılan sezgisel çizelgeleme kuralları, farklı özellikleri temel alınarak sınıflandırılabilirler. Sınıflandırma, kuralın özelliklerine dayanarak yapılacağı gibi, üretim ortamının özelliklerine dayanarak da gerçekleştirilebilir (Melnyk ve diğerleri, 1986):

(1) *Kural Odaklı Çerçeve*: Çizelgeleme literatürünü sınıflandırırken Moore ve Wilson (1962), gruplandırmayı aşağıdaki şekilde gerçekleştirmişlerdir:

Statik ve Dinamik: Statik kurallarda üretim ortamındaki koşullar değişse bile öncelikler değişmez. Dinamik kurallarda ise, üretim ortamındaki koşullar değiştikçe, örneğin yeni siparişlerin gelmesi, stok veya kapasite düzeyindeki değişiklikler gibi, siparişlerin öncelikleri de değişecektir.

Lokal ve Global: Lokal kurallar miyopiktir ve sadece verilen bir iş merkezinde kuyruktaki işler hakkındaki bilgileri gözönüne alır. Üretim ortamındaki diğer kuyrukların durumlarının bilgilerini gözardı eder. Ancak global kurallar iş önceliğini belirlerken, üretim ortamındaki diğer kaynaklardan, iş merkezinden önce gelen veya izleyen merkezlerdeki siparişlerin durumları hakkında gelen bilgileri de gözönüne alır.

Bir diğer kural odaklı çerçeve, Panwalker ve Iskander (1977) tarafından çizilmiştir. Çizelgeleme kurallarını sınıflandırırken aşağıdaki kategorileri kullanmışlardır:

Basit öncelik kuralları: Bunlar genellikle işin özel karakteristiğine (termin zamanı, kalan bolluk zamanı veya işlem süresi, kalan operasyon sayısı gibi) dayanır. Diğer bazı bilgiler, örneğin kuyruk uzunluğu, bu sınıfta alındığı gibi, rassal seçim kuralları da bu grupta alınır. Bu sınıflamada kullanılan bilgiler, işlem zamanı, işlem sayısı, maliyetler, hazırlık zamanı, geliş zamanı ve bolluk zamanı olarak sıralanabilir.

Basit öncelik kurallarının kombinasyonu: Verilen bir iş kuyruğunda farklı kümelere veya kuyruktaki tüm işlere farklı zamanlarda uygulanan kurallardır. Örneğin SPT kuralı, belirli bir maksimum periyot boyunca bekleyen iş haricindeki tüm işlere uygulanabilir.

Ağırlıklandırılmış öncelik indisleri: İşin önceliği belirlenirken ilk iki gruptaki kuralların ağırlıklandırılmış kombinasyonu kullanılır.

Sezgisel çizelgeleme kuralları: Özel bir işle ilgili olmayan bilgileri kullanarak oluşturulan daha karmaşık kurallardır, örneğin alternatif rota ulaşılabilirliği, makine atıl zamanı gibi.

Diğer kurallar: Özel bir üretim operasyonu için tasarlanan ve ilk dört gruba girmeyen kurallardır.

(2) *Üretim Ortamı Odaklı Çerçeve:* Üretim ortamının işlem karakteristikleri ile ilgilidir. Bu yaklaşım farklı üretim ortamlarının farklı çizelgeleme yaklaşımlarına ihtiyaç duyacağı gerçeğinden kaynaklanır. Day ve Hottenstein (1970) iki temel karakteristiğe dayanarak çizelgeleme kuralları için sekiz kategori tanımlamışlardır:

İş gelişlerinin yapısı: İş gelişleri kümeler, yığınlar veya sürekli gelişler halinde olabilir.

Rota veya üretim ortamının yerleşim yapısı: Tek iş, tek makinede çizelgeleniyor olabilir. Paralel çoklu makinelerde, her iş bir makineden geçer ancak paralel yerleştirilmiş birden fazla makine bulunur. Seri makinelerin olduğu ortam, her işin aynı dizi makineyi izlediği durumdur ve akış tipi üretim ortamı olarak da tanımlanabilir. Her işin kendi rotasına sahip olduğu çoklu rassal makineli ortam, sipariş tipi üretim ortamı olarak da adlandırılır.

Bu grup için bir başka sınıflandırma da Graves (1981) tarafından yapılmıştır:

İhtiyaçlar: Üretim emirlerinin ve siparişlerin müşteri tarafından belirlendiği ve stoğun olmadığı ortam açık üretim ortamı (open shop) olarak adlandırılırken, tüm müşteri isteklerinin doğrudan stoktan karşılandığı ortamlar da, kapalı ortam (closed shop) olarak adlandırılır.

İşleme kompleksliği: Tek aşama ve tek işlemci, tek aşama ve paralel işlemler, çok aşama ve akış tipi üretim ortamı, çok aşama ve sipariş tipi üretim ortamı olarak gruplandırılabilir.

izelgeleme kriteri: izelgeleme, izelgeleme maliyeti ve izelgeleme performansı temeline dayalı olabilir. Bu  gruplamaya gre Graves sekiz kategori tanımlamıştır.

Tm bu tanımlamalara gre en sık kullanılan sezgisel izelgeleme kuralları ve zellikleri kısaca řu řekilde zetlenebilir (Baker, 1974; Askin ve Standridge, 1993; Sipper ve Bulfin, 1997):

Tablo 3.1. Sezgisel izelgeleme Kuralları

SPT (Shortest Processing Time)	En kk iřlem zamanlı iři se
EDD (Earliest Due Date)	Termin tarihi ilk olan (en yakın olanı) se
FCFS (First Come First Served)	İř istasyonu kuyruđuna ilk gelen iři se
LCFS (Last Come First Served)	Son gelen iři se
FISFS (First In System First Served)	Sisteme ilk gelen iři se
STR (Slack Time Remaining)	Termin zamanından nce kalan sreyle, kalan iřlem zamanı arasındaki fark (slack) en kk olan iři se
S/RO (Slack per Remaining Operation)	Bolluk zamanı, kalan iřlem sayısına oranı en kk olanı se
LTWK (Least Total Work)	Toplam iřlem zamanı minimum olan iři se
MOPNR (Most Operation Remaining)	İřlem sırasında (rotasında) en fazla iři kalan iři se
MWKR (Most Work Remaining)	Kalan toplam iřlem zamanı en fazla olan iři se
LWKR (Least Work Remaining)	Tamamlanmamıř iřleri iin toplam iřlem zamanı en az olan iři se
RANDOM	İři rassal olarak se
WINQ (Work In Next Queue)	Takip eden makinedeki iři kuyruđu en kısa olan iři se
SD (Start Date)	Termin zamanı–normal retim zamanı en kk olan iři se
CR (Critical Ratio)	Termin zamanı ve řu andaki zaman arasındaki farkın kalan iřlere blm deđeri minimum olan iři se

Burada LTWK ve EDD statik, LWKR ve bolluk zamanı temelli kurallar dinamiktir. SPT miyopik, ancak WINQ global bir kuraldır

retim sisteminin zelliiđine gre sezgisel her kuralın, iyi sonu verdiđi farklı performans ltleri vardır. rneđin tek makine probleminde eđer tm iřlerin bırakılma zamanı aynı ve 0 ise, SPT ortalama akıř zamanı minimizasyonu, bekleyen

iş sayısı minimizasyonu, yani ara stok minimizasyonu sağlar (Sipper ve Bulfin, 1997). Ancak bu kural işlerin termin zamanlarını dikkate almadığından, $T_{\max}$ ve $L_{\max}$ minimizasyonu EDD ile sağlanabilir (Baker, 1974).

Çalışmada denemelerde ilk olarak, sisteme müdahale etmeyen ve çoğu zaman karşılaştırma kuralı (benchmark) olarak kullanılan FCFS, termin zamanlarını dikkate alan ve gerçek sistemlerde de sıkça kullanılan EDD, dinamik bir algoritma olan ve uygulamalarda birçok performans ölçütü, özellikle ortalama akış zamanı dolayısıyla da yarı mamül stokları ve gecikmeler için en iyi sonuçları veren LWKR ve SPT kuralları ile, yine dinamik algoritmalar olan STR, MOPNR, MWKR, S/RO ile, statik SD olmak üzere dokuz sezgisel algoritmanın bu sistem için modellenmesi gerçekleştirilmiştir. Hem statik, hem de dinamik algoritmalar alınmış, gerçek sistemlerde sıkça kullanılan algoritmalara bakılmış ve sistemin her türlü özelliğini dikkate alabilmesi için bu dokuz sezgisel algoritmanın seçimi yapılmıştır.

3.2.4. Darboğaz Kaynaklar

Hemen tüm üretim ortamlarında, bir veya birkaç tane kısıtlı kaynak veya darboğaz bulunur (Porter ve diğerleri, 1999). Darboğazı eğer üretim kaynağı olarak düşünecek olursak, darboğaz kaynak, kapasitesi, istenen veya gereken kapasiteden daha az olan kaynaktır. Darboğazlar, bazı ürünlerin veya bileşenlerin üretim hızını sınırlayan kaynaklardır (Dilworth, 1996).

Darboğazların tanımlanması birkaç yolla gerçekleştirilebilir. Darboğaz kaynak, eğer talep makinenin kapasitesini aşıyorsa üretim ortamındaki en uzun kuyruklara sahip olacaktır (Dilworth, 1996; Sipper ve Bulfin, 1997). Kapasitenin aşılmadığı durumlarda, makinelerin iş yüklerine bakılabilir (Morton ve Pentico, 1993). İş yükü oranı diğerlerinden yüksek olan kaynak darboğaz olarak tanımlanabilir (Dilworth, 1996; Sipper ve Bulfin, 1997).

Eğer darboğaz kaynaklar etkin bir şekilde çizelgelenebilir ise, diğer makineler de bu çizelgeye göre planlanabilir. Darboğazdan önceki makinelerin, yapılan darboğaz çizelgelemesine göre, geriye doğru (backward) çizelgelemesi yapılırken, darboğazdan sonra gelen makinelerin, ileriye doğru (forward) çizelgelemesi

gereklidir (Sipper ve Bulfin,1997).

Darboğaz kaynakları, daha doğrusu sistemdeki kısıtları farketmeyi, tanımlamayı ve çözümlemeyi amaçlayan yönetim felsefesi, 'Kısıtlar Teorisi (Theory Of Constraints, TOC)' olarak adlandırılır. Teori, Eliyahu Goldratt tarafından geliştirilmiş ve tanıtılmıştır. 1980'lerde Goldratt, üreticilerin kaynaklarını ve stoklarını kontrol etmede yetersiz kaldıkları görüşüne dayanarak, bunu çözmek için işletmelerin sınırlı imkanlarını, makine, personel, alet, malzeme ve diğer kısıtları göz önüne alan, üretim prosesini çizelgeleyecek bir yazılım geliştirmiştir (Chase, 1998). Bu yazılım Optimize Üretim Teknolojisi (Optimized Production Technology, OPT) olarak adlandırılır. Bu çizelgeler, mümkün ve doğru idi, dahası bilgisayarda, MRP'nin gerektirdiği zamanın ancak bir kısmını kullanıyordu. Bunun sebebi, darboğaz ve darboğaz olmayan operasyonların ayrılmasına dayanan çizelgeleme mantığıdır. OPT çizelgelemesinin mantığının anlaşılmasına yardımcı olmak için Goldratt, 9 üretim çizelgeleme kuralı tanımlamıştır:

- Kapasiteyi değil, akışı dengele.
- Darboğaz olmayan bir kaynağın kullanım düzeyi kendi potansiyeli ile değil, sistemdeki diğer bazı kısıtlar ile belirlenir.
- Bir kaynağın kullanımı (utilization) ile, çalışması (activation) aynı değildir.
- Darboğaz kaynakta bir saat kayıp, tüm sistemde bir saat kayıp demektir.
- Darboğaz olmayan kaynakta bir saat kazanılması, sadece aldatıcı bir görünüştür.
- Darboğaz kaynaklar sistemin çıktı düzeyini ve stoğunu yönetirler.
- Taşıma parti büyüklüğü ve üretim parti büyüklüğü eşit olmayabilir, ve çoğu zaman da eşit olmamalıdır.
- Üretim parti büyüklüğü rotası ve zamanı konusunda değişken olmalıdır.
- Öncelikler ancak sistemin kısıtları incelenerek konulabilir.

OPT, bir işletmedeki tüm iş merkezleri için öncelik ve kapasite kısıtlarını göz önüne alarak optimuma yakın iş çizelgelerini hazırlar. Bu sistemde amaç, kritik

(darboğaz) kaynakların kullanımını en büyükleyerek üretim miktarını arttırmak, buna karşılık ara stok düzeyleri ile kaynak hazırlık zamanlarını en aza indirmektir.

Yaklaşık 100 büyük firmaya bu yazılım kurulduktan sonra, Goldratt yazılımdan çok yaklaşımın mantığı üzerinde durmuştur. Ve birçok işletme alanlarına uygulanabilecek popüler bir problem çözme yaklaşımı olan TOC'i geliştirmiştir.

Kısıtlar Teorisine göre görünüşte farklı olsa da, firmaların asıl hedefi 'faaliyet gösterdiği anda ve gelecekte para kazanmak'tır. Firmaların amaç olarak gördüğü diğer ölçütler, yüksek kaliteli ürünler üretmek, daha iyi müşteri hizmetleri sunmak, istihdam sağlamak, teknoloji geliştirmek, pazar payı kazanmak gibi, ancak eğer firmanın parası varsa ve varlığını sürdürebiliyorsa olabilecek amaçlardır. Kısıtlar Teorisi, firmanın amaçlarına ulaşmada varolan kısıtlarını farketmeye yöneliktir. Firmanın amaçlarına ulaşmasını engelleyen, onu sınırlayan faktörler 'kısıt' olarak adlandırılır. Bu kısıtlar yönetimsel ve davranışsal kısıtlar, lojistik kısıtlar, pazar kısıtları, malzeme veya kapasite kısıtı olarak sıralanabilir. Fabrikanın üretim akışını düzgün ve zamanında sağlama yeteneğini doğrudan etkileyen ve belki de en çok gözlenen anahtar kısıtlardan biri de, kapasite kısıtlarıdır. Kapasite kısıtı, eldeki ulaşılabilir kaynağın kapasitesi istenen çıktıyı sağlamada yetersiz kaldığı durumda oluşur. Kısıtlar Teorisi, gerçek kısıtların, dolayısıyla da organizasyonun performansını artırmak için kıt kaynaklara odaklanan bir yönetim felsefesidir. Bu felsefenin sürekli gelişme sürecinde tanımladığı beş adımı vardır:

1. Sistemdeki kısıtların tespiti ve tanımlanması
2. Bu kısıtların nasıl düzeltileceğine karar verilmesi
3. İlgili her şeyin bu kararın uygulanması için seferber edilmesi, kısıtı desteklemek için sistemin diğer tüm parçalarının düzenlenmesi
4. Sistem kısıtlarının ortadan kaldırılması
5. İlk adıma dönülerek diğer kısıtların tanımlanması

Üretim ortamlarındaki üretim/durdurma kararları 'davul çalışı' olarak adlandırılır (Goldratt ve Fox, 1986; Umble ve Srikanth, 1995). Geleneksel sistemlerde (Just In Case -JIC) davul, hammadde sisteme girdiğinde çalarken, JIT (Just In Time) sistemlerinde davul, pazar talebi tarafından vurulur. Kısıtlar Teorisi

temelli DBR (Drum-Buffer-Rope) tekniğinde davul vuruşu, kısıtlı kaynağa bağlıdır. Bütün sistemin üretim oranı bu kaynağa bağlıdır. Kısıtlı kaynağın önünde ara stok bulunur. Bu stok, kısıtlı kaynağı belirli bir süre korumaya yetecek kadardır. Sistemin stok düzeyinin korunabilmesi için, sisteme girecek hammadde miktarının da sınırlandırılması gereklidir. Kısıtlı kaynakla ilk giriş istasyonu arasında kurulacak bir iletişim (rope) bu sınırlamayı sağlayacaktır. Kısıtlı kaynakların ve kritik noktaların korunmasında kullanılan ara stok, zaman stoğu (time buffer) olarak adlandırılır ve kısıtlı kaynağın ne kadar süre korunmaya ihtiyacı olduğuna göre, büyüklüğü belirlenir.

OPT, sistemdeki darboğaz kaynakları bulmaya yöneliktir. Sistemin kalanı da darboğazı besleyecek şekilde yönlendirilirler. OPT'nin amacı, darboğaz sisteme odaklanmasıdır, tek ve sabit darboğazlı sistemlerde performansı oldukça iyi olan bir modeldir. Ancak ideal sistem olmasından alıkoyacak bazı sakıncaları vardır: çoklu ve değişken darboğazlı sistemlerde bu yöntem işlemeyecektir (Morton ve Pentico, 1993).

Özetle darboğaz, üretim yapısında bir yerlerde, her zaman vardır. Planlama tekniği darboğazın belirlenmesi, çıktının maksimize edilmesini sağlayacak şekilde termin tarihlerine dikkat ederek operasyonların çizelgelenmesi ve diğer operasyonların akışının korunması için çizelgelenmesi şeklindedir. (Perter ve diğerleri, 1999). Diğer operasyonlar, darboğaz kaynağın yerleşimine göre geriye veya ileriye doğru çizelgelenirler.

3.2.5. Değişken Darboğaz Olması Durumu

Darboğaz değişimi (shifting bottleneck) iş merkezlerinde darboğazın yerinin zaman içinde değişmesi durumudur. Darboğaz değişimi, üretim sisteminin rastsallık özelliklerinden kaynaklı doğal bir sonuçtur.

Lawrance ve Buss (1991) darboğaz için,

1. Geçici olarak kapasitesi yetmeyen iş merkezi,
2. O anda en çok işe veya müşteriye sahip iş merkezi,
3. Uzun dönemde kullanım oranı en yüksek olan iş merkezi tanımlarını

yapmışlardır.

Bu tanımlar birbirleriyle ilişkilidir, ancak zaman ufukları farklıdır: sırasıyla, kısa, orta ve uzun dönemlik tanımlardır. İlk tanımda, iş merkezinin kapasite altı kalmasının ne kadar süreceği sorulabilir. Bir iş merkezi, kapasitesinden daha fazla iş aldığı anda bunun sonucu, uzun kuyruklar ve ara stok düzeyinde artışlardır. Kısa dönemde bu tip kapasite yetmeme durumlarının göstergesi, iş merkezi önündeki uzun kuyruklardır. Bu bizi ikinci tanıma götürür. Üçüncü tanım, uzun dönemlik kapasite kısıtlarının tanımlanması için uygundur, ancak kısa dönemde kısıtların tanımlanmasında uzun dönemlik kullanım oranı yüksek olan iş merkezlerinde bile göreceli olarak tıkanıklık olmayabileceğinden, daha az uygun olduğu söylenebilir. Dahası, uzun dönemlik kapasite kullanım oranına dayalı tanımlar, kısa dönemlik dalgalanmaları da dikkate almazlar.

Darboğaz değişimi ölçütü için hem kısa, hem de uzun dönemlik özellikler dikkate alınmalıdır. Bunun için, bir iş merkezinin diğerlerinden daha fazla işe sahip olduğu zamanın oranının uzun dönemde ölçümü dikkate alınabilir. Bu durumda ölçüt, darboğaz değişiminin hem dinamik özelliğini, hem de uzun dönemdeki durağan özelliklerini dikkate alacaktır.

Lawrance ve Buss (1994)'un belirttiği darboğaz olma olasılıklarını gösteren formülasyona göre, her iş merkezi, atıl kapasitesi büyük veya kullanım oranı düşük olsa da, zamanının belirli bir oranında darboğaz olacaktır.

Darboğaz değişimi problemi, eğer sistem dengeli ise, yani istasyonların kullanım oranları birbirleriyle eşit durumda ise, daha da kötü olacaktır. Bu istasyonların darboğaz olma olasılıkları birbirlerine yakındır, bu durumda da darboğaz değişimi (shiftiness) maksimize olmuştur. Diğer yandan, tek bir baskın iş merkezi varsa, bunun darboğaz olma olasılığı 1'e yaklaşır ve diğer merkezlerin olasılıkları da düşer.

Daha önce de bahsedilen ve darboğaz değişimini ölçen β parametresi, sistem performansı için, ara stok düzeyi veya ortalama akış zamanları gibi bir ölçüt değil, üretim planlamanın kompleksliğinin bir ölçütü olacaktır. β , 0 ve 1 arasında değer alabilir. Bu değer 0'a yaklaştıkça değişmeyen tek bir darboğazın olduğu ve yönetimin buna odaklanması gerektiği söylenebilir. β değeri yükseldikçe,

darboğazların zaman içinde değiştiğinden söz edilebilir.

Sistem parametrelerinin birçoğunun β üzerinde etkileri vardır (Moss ve Yu, 1999). İş geliş oranlarının β üzerinde pozitif etkiye sahip olması beklenir. İşler daha sık olarak geldikçe, işlenmek için daha çok iş sistemde olacak, tüm iş merkezlerinin kullanım oranları artacaktır. Kullanım oranı arttıkça tek bir iş merkezinin baskın darboğaz olma olasılıkları düşecektir. Bu da darboğaz değişiminde artış demektir. Darboğaz kaynakta ortalama işlem süresi, β üzerinde negatif etkiye sahiptir. İşlem zamanı arttıkça, darboğaz, darboğaz olarak kalmaya devam edecek ve darboğaz değişimi oranı da düşecektir. Darboğaz olmayan kaynakta ortalama işlem süresi arttıkça, bu iş merkezleri darboğaz kaynak olmaya daha çok yaklaşacaklardır. Dolayısıyla bu değişkenle darboğaz değişimi arasında pozitif bir ilişkiden söz edilebilir. İş ortamı sipariş tipi olduğunda rotalar değiştiğinden, akış tipi üretim sistemine göre darboğaz değişiminin daha yüksek olması beklenir. İş ortamında iş merkezi sayısının değişimi de β üzerinde etkilidir. İş merkezleri arasında kuyruk oluşumları doğaldır, iş merkezi sayısı arttıkça kuyruk oluşma potansiyeli artar, daha çok ara stok oluşur, kullanım oranları artar, ve β bu durumda artar.

Özetle, darboğaz değişiminin, darboğazdaki ortalama işlem süresi ile negatif, iş geliş oranı, iş merkezlerinin sayısı ve darboğaz olmayan kaynaklardaki işlem süresi ile de pozitif ilişkili olduğu söylenebilir.

Üretim ortamlarında ortaya çıkan darboğaz değişim problemi, yöneticiler tarafından çoğunlukla geçici olarak kapasiteyi artırma veya kalıcı olarak kapasite ekleme ile çözümlenir. Darboğaz değişimi probleminin çözülmesi için önlemler alınırken, üretim ortamının diğer bazı karakteristikleri üzerinde yıkıcı etkileri olabilir. Özellikle ortalama akış süresine dikkat edilmelidir.

Kapasitenin geçici olarak artırılması fazla mesai uygulaması, işçi ekleme gibi yollarla gerçekleştirilir. Ancak bu önlemler darboğaz değişimini olumsuz etkileyecektir. Bir iş merkezinde kuyruğun azaltılması ve o iş merkezinin darboğaz olmasının engellenmesiyle başka bir iş merkezinin çok daha çabuk olarak darboğaz merkez haline gelmesine sebep olur, dolayısıyla darboğaz değişimini artırır. Ancak bu önlem, ortalama akış süresinin azaltılmasında etkindir. Dolayısıyla yöneticiler, bir ikilemle karşı karşıyadırlar.

Uzun dönemlik darboğazlarda kapasiteyi kalıcı olarak artırmak, darboğaz değişimini artıracaktır. Tersine, kullanım oranı düşük olan iş merkezlerinde kapasite artışı bu değişimi azaltacaktır (Lawrance ve Buss, 1994). Burada da, uzun dönemlik darboğaz kapasitesinin artırılması, ortalama akış süresini azaltacaktır. Bu sonuçlar, ilk bakışta sezgilerle çelişebilir. Uzun dönemlik darboğaz azaltılmasının darboğaz değişim problemini de azaltacağı varsayılır. Ancak uzun dönemlik darboğaz kapasitesi arttıkça, bekleyen iş kuyrukları azalacak, merkezlerin darboğaz olma sıklıkları artacaktır. Sistemin bütünsel değişimi de artacaktır. Tersine, düşük kullanım oranına sahip iş merkezlerine kapasite eklenmesi, o iş merkezinin darboğaz olma olasılığını daha da düşürecek, dolayısıyla sistemin darboğaz değişimi de azalacaktır.

Görüldüğü gibi, üretim ortamının performansını iyileştirme çabaları, darboğaz değişimi üzerinde istenmeyen etkilere sebep olabilir. Yöneticiler, örneğin ortalama akış zamanı gibi üretim ortamı performansı iyileştirmeye yönelik politikaları tasarlarken, darboğaz değişiminin artmasına sebep olacak diğer istenmeyen etkileri de gözönünde bulundurmalıdırlar.

3.2.6. Geliştirilen Çizelgeleme Algoritması

Daha önce Bölüm 3.2.4'te bahsedilen kısıtlarla yönetim felsefesini uygulayan OPT programında temel yaklaşım, belirli bir darboğaz için, bu iş merkezi hiç boş kalmayacak şekilde işlerin sıralanması ve bu iş merkezine gönderilmesidir. Diğer iş merkezleri ise bu iş merkezinin önceliklerine göre planlanır.

Darboğazların değişken olması durumunda ise bu yaklaşım iyi sonuç vermemektedir. (Morton ve Pentico, 1993). Bir iş merkezi için çalışma yapıp, diğer iş merkezlerini buna göre düzenledikten sonra, başka bir iş merkezinin darboğaz duruma geliyor olması yapılan çizelgelemenin etkin olmaması sonucunu doğurmaktadır.

Bu nitelikteki üretim ve sipariş yapılarına, özellikle sipariş tipi üretim yapan işletmelerde sık rastlanmaktadır. Genelde en belirgin darboğaz halde olan iş merkezi dikkate alınarak çizelgeleme yapılmakta, farklı iş merkezlerinin de darboğaz duruma

geldiĐi anlarda geici kapasite artırımları (mmkn olduĐu zaman fazla mesai ya da ilave iřĐc gibi) yapılarak plana uyulmaya alıřılmaktadır. Bunun mmkn olmadığı zaman ise izelgelemenin kayıpları mecburen kabul edilmektedir.

DeĐiřken darboĐazlı sistemlerde izelgeleme probleminin zm iin iki tip algoritma geliřtirilmiřtir. Bu algoritmalar darboĐaz niteliĐindeki iř merkezlerinin boř kalmalarını nleyecek řekilde iřlerin ncelik sıralarına sahip olması temel mantıĐından yola ıkılarak hazırlanmıřtır. Aslında OPT ile aynı temel yaklařımı kullanmakla beraber farkı, bu izelgelemenin dinamik olarak retim sistemine yn vermesidir.

Her yeni sipariř geliřinde ve her iř merkezi iin yeni bir iřleme bařlama ncesinde, sistemde tanımlanan ncelik kurallarına bakılarak sıralama belirlenmektedir. DarboĐaz iř merkezinin deĐiřmesi, bu sistemde dikkate alınabilmekte ve sistemin incelenen andaki resmi, aktif darboĐazı belirlediĐi iin, istenen performans deĐiřkenlerini optimize edecek řekilde ncelikler belirlenebilmektedir.

ncelik kurallarının oluřturulmasında iki tip algoritma geliřtirilmiřtir.

DarboĐaz iř merkezlerinin iř yklerine bakılarak ve belirlenmiř zaman stoĐunun alt ve st limitleri ile karřılařtırılarak, sipariřlerin;

- Birinci algoritmada, iřlem sreleri
- İkinci algoritmada, darboĐaz iř merkezine ulařma sreleri

kullanılarak, sistem istenen zellikleri itibariyle dengelenir.

Algoritmaya ait detay bilgiler Blm 3.2.6.2.'de verilmiřtir.

3.2.6.1. DarboĐaz KaynaĐın Tanımlanması ve DarboĐaz DeĐiřiminin llmesi

Bu alıřmada, Lawrance ve Buss (1994) tarafından yapılan darboĐaz olma tanımlarından ikinci tanım kabul edilerek darboĐaz tespiti yapılmıřtır. Bu tanıma gre, gzlem anında en ok iř ykne sahip olan iř merkezi darboĐazdır.

Sistem mevcut varsayımları altında alıřtırılırken ve sisteme giren sipariřler geliř sırasına gre iřlenirken, sisteme her yeni sipariř giriřinde, btn sistemin bir

resmi çekilmekte, tüm makinelerin iş yüküne, yani önündeki toplam kuyruklara zaman stoğu birimiyle bakılmakta (her p_{ij} ($j = 1, \dots, m$)) ve her seferinde en uzun zaman kuyruğu yani en yüksek işyükü olan kaynak belirlenerek, bu kaynağa ait darboğaz olma durumuna 1 değeri eklenmektedir. Simülasyon zamanı sonunda elde edilen darboğaz olma durum sayılarına bakılmaktadır.

Çalışmada ele alınan örnek problemde bu yaklaşım;

S_j : j makinesinin darboğaz duruma düşme sayısı

p_{ij} : j makinesi önündeki i işinin j makinesinde gerektirdiği işlem süresi olmak üzere,

$t=0$ dan simülasyon sonuna kadar;

$t = t +$ (sipariş gelişler arası süre) anında (sisteme her yeni sipariş gelişinde)

her p_{ij} ($j = 1, \dots, m$) için (tüm kaynakların işyüklerinin saptanması)

$\max \{p_{ij}\}$ ($j = 1, \dots, m$) değerini sağlayan j için $S_j = S_j + 1$

şeklinde uygulanmıştır.

Sonuçta, iki makinenin diğerlerine oranla, daha fazla sayıda darboğaz durumuna geldikleri ve darboğaz olma özelliğinin bu ikisi arasında değiştiği görülmüştür. Elde edilen sonuçlar Tablo 3.2'dedir.

Tablo 3.2. Darboğaz Kaynakların Tespiti

Deneme #	1	2	3	4	5	Ortalama	Olasılık
HASILSOKUMU	0	0	0	0	0	0	0,00
BEKLETME	0	0	0	0	0	0	0,00
KOMBINKASAR	280	7	148	150	383	193,6	0,01
MERSERIZE	0	0	0	0	0	0	0,00
YIKAMA	5	2	1	1	11	4	0,00
KURUTMA	18	0	0	99	4	24,2	0,00
ZIMPARA	1048	217	571	924	953	742,6	0,05
BOYAMA	7728	10415	8991	7129	7685	8389,6	0,53
APRE	547	105	385	257	259	310,6	0,02
SANFOR	0	0	0	0	0	0	0,00
FIKSE	40	35	8	46	27	31,2	0,00
JETBOYAMA	74	7	78	196	68	84,6	0,01
YAKMA	0	5	0	1	4	2	0,00
BASKI	5395	5174	5431	6432	5498	5586	0,35
KONDANSE	570	55	228	598	750	440,2	0,03

Görüleceği üzere boyama istasyonu toplam darboğaz gözlemi yapıldığı anların %53'ünde en uzun kuyruğa sahip istasyon, baskı istasyonu ise % 35'inde en uzun kuyruğa sahip istasyon olarak tespit edilmiştir.

İki gözlem arasında geçen sürenin, tüm gözlemler dikkate alındığında, her kaynağı eşit etkilediği varsayılırsa, bu kaynakların zamanın %53'ünde ve %35'inde darboğaz oldukları söylenebilir.

Bu tespit, tüm önceliklerde FCFS kuralı kullanılarak yapılmıştır. Sistem, ilk anda farklı bir sıralama kuralı ile çalıştırılarak da darboğaz oluşumları incelenebilir. Bu durum (uygulanan kural makine öncelikleri getirmediği takdirde) darboğaz olma olasılıklarını etkilemeyecektir.

Örnek problemde elde edilen sonuçlara göre, darboğazların değişim ölçütü (shiftiness) hesaplanırsa;

$$\beta = 1 - cv/\sqrt{n}$$

formülü kullanılarak;

$$cv = 2,349188$$

$$\sqrt{n} = 3,872983 \quad (n = 15 \text{ iş merkezi için}) \text{ ve,}$$

$$\beta = 0,393$$

bulunur.

Daha önce 2.4 alt bölümünde anlatıldığı üzere β değerinin sıfır olması sistemde tek bir baskın darboğazın olduğu ve bunun değişmediği durumunu ifade ederken, bir olması tüm iş merkezlerinin darboğaz olma olasılığının eşit olduğu durumu gösterir.

Örneğimizde $\beta = 0,393$ olarak bulunması birden fazla darboğaz olduğu ve bunların zaman içinde değiştiği anlamına gelir. Darboğaz olabilen iş merkezi sayısının çok olmaması sebebiyle, bu değer 0,39 seviyesindedir. Eğer iki iş merkezi yerine daha fazla sayıda iş merkezi darboğaz olsaydı bu değer daha yüksek çıkacaktı.

3.2.6.2. Çizelgeleme Algoritması

Geliştirilen algoritmalar için matematiksel gösterim aşağıdaki gibidir:

- i iş numarası
- i^* belirlenen her makinede öncelikli olan işin numarası
- t bir sonraki çizelgeleme zamanı
- B darboğaz makineler kümesi
- k darboğaz olan makineleri belirtir ($k \in B$)
- u_{it} t anında sitemde geçerli öncelik
- ϕ_k k darboğaz makinesinde tanımlanan zaman stoğu üst sınırı
- ϑ_k k darboğaz makinesinde tanımlanan zaman stoğu alt sınırı
- f_k $\Sigma p_{ik} / \vartheta_k$ ($k \in B$) oranı
- p_{ij} i işinin j makinesindeki işlem süresi
- a_{ik} i işinin k darboğaz makineye kadar olan iş merkezlerindeki toplam işlem süresi
- U_k k darboğaz makinesinde işlenecek ($k \in B$) işler kümesi
- $k \notin A$ $\{1, 2, \dots, m\}$ sistemdeki darboğaz dışındaki makineler
- H_k k makinesinde ($k \in B$) şu anda işlenen işin tamamlanma zamanı (k makinesinin bir sonraki iş için hazır olduğu an)
- A_j darboğazdan önceki makineler ($j = 1, \dots, k-1$)
- A_ℓ darboğazdan sonraki makineler ($\ell = k+1, \dots, m$)

Adım 0: Sıfırlama Adımı

$t = t_0$, $U_k = \emptyset$, $k \in B$, $B \subset A = \{1, 2, \dots, m\}$

Adım 1: Bir Sonraki Çizelgeleme Algoritmasının Seçimi

Zamanı t kadar artır $t = \min H_k$ ($k \in B$)

U_k belirle.

Σp_{ik} , $k \in B$ hesapla. $i = k$ makinesini kullanan ve önceki makinelerde operasyonlarını tamamlamış işler.

Adım 2: Güncelleme

t zamanında biten işleri bırak.

U_k kümesini ve Σp_{ik} ($k \in B$) değerlerini güncelle.

Adım 3: İş Seçimi

Durumlara karşılık gelen kullanılacak çizelgeleme kuralları ve öncelikleri dinamik olarak belirle.

u_{it} önceliklerini ve i^* işini belirle.

Bu adımda, sistemde geçerli önceliği belirlemek amacıyla iki farklı algoritma kullanılmıştır:

1. Algoritma

Darboğaz kaynağın iş yükünü dengelemeye yönelik olarak, uygun işlem zamanlı işin öncelikli olması sağlanmaktadır.

Darboğaz kaynağın zaman stoğu az ise, bu kaynakta işlem süresi uzun olan işlere darboğazdan önce gelen tüm iş merkezlerinde öncelik verilerek, kaynağın önündeki zaman stoğu artırılmakta ve boş kalması olasılığı azaltılmaktadır.

Gözlem anında zaman stoğu fazla ise, bu kaynakta işlem süresi kısa olan işlere darboğazdan önce gelen tüm iş merkezlerinde öncelik verilerek, kaynağın bekleyen zaman stoğu azaltılmakta ve diğer kaynakların öncelikli kullanımı sağlanarak sistemin toplam ara stoğunun artması engellenmekte, diğer kaynakların kullanım oranları artırılmaktadır. Diğer sistem performans değişkenleri de olumlu etkilenmektedir.

Bu algoritma için darboğaz kaynakların durumunu ve sistemde geçerli öncelikleri belirleyecek üç farklı durumdan sözedilebilir:

1. En az bir k ($k \in B$) için; $\Sigma p_{ik} < \theta_k$ ise,

şartı sağlayan k iş merkezleri içinde; $u_{it} = \min f_k$ koşulunu sağlayan $k \in B$ için

$i^* = \max p_{ik}$

2. $\forall k \in B$ için; $\Sigma p_{ik} > \theta_k$ ise; ve en az bir k için;

$\Sigma p_{ik} < \phi_k$ ise, sağlayan $k \in B$ için

$u_{it} = \min f_k$ koşulunu sağlayan $k \in B$ için

$i^* = \max p_{ik}$

3. $\forall k \in B$ için; $\Sigma p_{ik} > \phi_k$ ise;

$u_{it} = \max f_k$ koşulunu sağlayan $k \in B$ için

$i^* = \min p_{ik}$

2. Algoritma

Darboğaz kaynağın iş yükünü dengelemek için, işlerin darboğaz kaynağa ulaşma hızlarına bağlı olarak öncelikler belirlenir.

Darboğaz kaynağın zaman stoğu az ise, bu kaynağa en kısa sürede ulaşacak işlere darboğazdan önce gelen tüm iş merkezlerinde öncelik verilerek kaynağın bekleyen zaman stoğu artırılmakta ve boş kalması olasılığı azaltılmaktadır.

Gözlem anında zaman stoğu fazla ise, bu kaynağa ulaşma süresi en uzun olan işlere darboğazdan önce gelen tüm iş merkezlerinde öncelik verilerek kaynağın bekleyen zaman stoğu azaltılmaktadır.

Bu algoritma için darboğaz kaynakların durumunu ve sistemde geçerli öncelikleri belirleyecek üç farklı durumdan sözedilebilir:

1. En az bir k ($k \in B$) için; $\Sigma p_{ik} < \theta_k$ ise,

şartı sağlayan k iş merkezleri içinde; $u_{it} = \min f_k$ koşulunu sağlayan $k \in B$ için

$i^* = \min a_{ik}$

2. $\forall k \in B$ için; $\Sigma p_{ik} > \theta_k$ ise; ve en az bir k için;

$\Sigma p_{ik} < \phi_k$ ise, sağlayan $k \in B$ için

$u_{it} = \min f_k$ koşulun sağlayan $k \in B$ için

$i^* = \min a_{ik}$

3. $\forall k \in B$ için; $\Sigma p_{ik} > \phi_k$ ise;

$u_{it} = \max f_k$ koşulun sağlayan $k \in B$ için

$i^* = \max a_{ik}$

Adım 4:

Algoritmaya göre u_{it} önceliklerini ve i^* işini belirle. A_j ($j = 1, \dots, k-1$) makinelerinde u_{it} önceliği geçerlidir ve i^* işi önceliklidir. A_ℓ ($\ell = k+1, \dots, m$) makinelerinde

$i^* = \min p_{it}$ geçerlidir. Adım 1' e git.

3.2.6.3. Örnek Problem Algoritması

Genel özellikleri Bölüm 3.2.8.1'de verilen örnek problem için yukarıda tanımlanan algoritmalar aşağıda detaylandırılmıştır.

Adım 0: Sıfırlama Adımı

$t = t_0$, $U_k = \emptyset$, $k \in B$, $B = \{x, y\}$ $B \subset A = \{1, 2, \dots, x, \dots, y, \dots, m\}$

Adım 1: Bir Sonraki Çizelgeleme Algoritmasının Seçimi

Zamanı t kadar artır. $t = \min H_k$, $k \in B$

U_k belirle.

Σp_{ik} , $k \in B$, hesapla. i : k makinesini kullanan ve önceki makinelerde operasyonlarını tamamlamış işler.

Adım 2: Güncelleme

t zamanında biten işleri bırak.

U_k kümesini ve Σp_{ik} , $k \in B$ değerlerini güncelle.

Adım 3: İş Seçimi

Durumlara karşılık gelen kullanılacak çizelgeleme kuralları ve öncelikleri dinamik olarak belirlenir.

u_{it} önceliklerini ve i^* işini belirle.

1. Algoritma

(1) $\Sigma p_{ik} < \vartheta_k$ ($k=x, y$) ve $u_{it} = \min f_k$ ($k=x, y$) için $i^* = \max p_{ik}$

(2) $\Sigma p_{ix} < \vartheta_x$ ve $\Sigma p_{iy} > \vartheta_y$ için $i^* = \max p_{ix}$

- (3) $\Sigma p_{ix} > \vartheta_x$ ve $\Sigma p_{iy} < \vartheta_y$ için $i^* = \max p_{iy}$
- (4) $\Sigma p_{ik} > \vartheta_k$ ve $\Sigma p_{ik} < \phi_k$ ($k = x, y$) için $u_{it} = \min f_k$ ve $i^* = \max p_{ik}$
- (5) $\Sigma p_{ik} > \vartheta_k$ ve $\Sigma p_{ix} > \phi_x$ ve $\Sigma p_{iy} < \phi_y$ için $i^* = \max p_{iy}$
- (6) $\Sigma p_{ik} > \vartheta_k$ ve $\Sigma p_{ix} < \phi_x$ ve $\Sigma p_{iy} > \phi_y$ için $i^* = \max p_{ix}$
- (7) $\Sigma p_{ik} > \vartheta_k$ ve $\Sigma p_{ik} > \phi_k$ için $u_{it} = \max f_k$ ve $i^* = \min p_{ik}$

2.Algoritma

- (1) $\Sigma p_{ik} < \vartheta_k$ ($k=x, y$) ve $u_{it} = \min f_k$ ($k=x, y$) için $i^* = \min a_{ik}$
- (2) $\Sigma p_{ix} < \vartheta_x$ ve $\Sigma p_{iy} > \vartheta_y$ için $i^* = \min a_{ix}$
- (3) $\Sigma p_{ix} > \vartheta_x$ ve $\Sigma p_{iy} < \vartheta_y$ için $i^* = \min a_{iy}$
- (4) $\Sigma p_{ik} > \vartheta_k$ ve $\Sigma p_{ik} < \phi_k$ ($k = x, y$) için $u_{it} = \min f_k$ ve $i^* = \min a_{ik}$
- (5) $\Sigma p_{ik} > \vartheta_k$ ve $\Sigma p_{ix} > \phi_x$ ve $\Sigma p_{iy} < \phi_y$ için $i^* = \min a_{iy}$
- (6) $\Sigma p_{ik} > \vartheta_k$ ve $\Sigma p_{ix} < \phi_x$ ve $\Sigma p_{iy} > \phi_y$ için $i^* = \min a_{ix}$
- (7) $\Sigma p_{ik} > \vartheta_k$ ve $\Sigma p_{ik} > \phi_k$ için $u_{it} = \max f_k$ ve $i^* = \max a_{ik}$

Adım 4:

Algoritmaya göre u_{it} önceliklerini ve i^* işini belirle. A_j ($j = 1, \dots, k-1$) makinelerinde u_{it} önceliği geçerlidir ve i^* işi önceliklidir. A_ℓ ($\ell = k+1, \dots, m$) makinelerinde $i^* = \min p_{it}$ geçerlidir. Adım 1'e git.

3.2.7. Simülasyon Deneylerinin Tasarımı

Simülasyon, karmaşık proses ve sistemlerin tasarımı ve işletiminden sorumlu karar vericiler için en güçlü araçlardan birisidir. Olası olmayan durumların çalışma, analiz ve geliştirilmesi işlemlerini mümkün hale getirir. Gittikçe rekabetçi olan ortamda simülasyon, mühendis, tasarımcı ve yöneticiler için vazgeçilmez bir problem çözme yöntemidir (Shannon, 1998). Simülasyonu, ‘gerçek bir sistemin bir modelini tasarlama ve model ile sistemin davranışını anlama ve/veya bu sistemin işletilmesinde değişik stratejiler geliştirilmesi amacıyla, deneyler düzenlenmesi işlemi’ olarak tanımlamak mümkündür.

Bir simülasyon modelinin elemanları sistemin bileşenleri, kontrol edilebilen değişkenler ve parametreler, kontrol edilemeyen değişkenler ve parametreler, fonksiyonel ilişkiler, amaç fonksiyonu ve kısıtlardır (Halaç, 1982).

Simülasyon tekniği kullanmanın faydaları, model kurulduktan sonra sistemin farklı durumlardaki analizi için tekrar kullanılabilir oluşu, veriler detaylı olmasa da kullanılabilir oluşu, sistemdeki karmaşık etkileşimleri deneme imkanı vermesi, varolmayan sistemlerin de denenmesini sağlaması, analitik çözümlerin doğruluğunun test edilmesi için de kullanılabilir olması, dinamik sistemler için uygun oluşu olarak sıralanabilirken, dezavantajları da, çoğu zaman maliyetin yüksek olması, modeli oluşturmanın çok zaman gerektirmesi, her sistem için modelin tekrar kurulması gereği şeklinde tanımlanabilir.

Yapılan simülasyon modelinde, sırasıyla şu işlemler gerçekleşmektedir:

1. Siparişlerin belirlenen geliş hızına göre sisteme gelmeleri, sipariş büyüklüklerinin, termin zamanlarının ve diğer özelliklerinin belirlenmesi,
2. Siparişlerin, rotasının gerektirdiği iş merkezlerine uğraması, sipariş büyüklüklerine ve makineye göre işlemin gerçekleştirilmesi,
3. İşlemler sırasında, verilen çizelgeleme algoritmasına göre önceliklerinin belirlenerek işlenmesi,
4. Sistem için belirlenen performans değişkenlerinin hesaplanması.

Burada amaç, geliştirilen çizelgeleme algoritmalarının, mevcut sezgisel algoritmalar karşısında performansının gözlenmesidir. Simülasyon deneylerinde, algoritmaları değişen durumlarda da test edebilmek amacıyla, sistem değişkenlerinin farklılaştırılması gerçekleştirilmiştir. Simülasyon deneylerinde genellikle, farklı durumların test edilmesi amacıyla, sistemin iş geliş oranları, sipariş büyüklükleri ve termin zamanları gibi parametrelerin değiştirilmesi gerçekleştirilir.

Simülasyon deneylerinde istatistiksel tahminlerin doğruluğu ve yansızlığı için aşağıdaki değişkenlerde uygun değerlerin alınması gereklidir (Law ve McComas, 1999):

- Her simülasyon deneyinin çalışma uzunluğu,
- Bağımsız simülasyon deneylerinin çalıştırılma sayısı,
- Isınma (warm up) periyodu uzunluğu,

Simülasyon deneylerinde en az 3 bağımsız çalıştırılmanın yapılması gereklidir. Her bir bağımsız çalışmada performans ölçütlerinin ortalamalarını kullanarak tamamının genel ortalamasına bakılır ve performans ölçütü değeri bu şekilde hesaplanır. Birçok simülasyon çalışmasında sistemin uzun dönemlik (steady state) davranışı ile ilgilenilir. Çalışmaya sistem atıl halde iken başlandığından, performans ölçütleri hesaplanırken başlangıç dönemindeki verilerin dikkate alınmaması, sistem normal durumunda çalışırken verilerin toplanması gerekir, bu da ısınma periyodunun kullanılmasıyla sağlanır.

Bu çalışmada, ısınma periyodu kullanılmış, model yaklaşık dört yıllık bir dönem için çalıştırılmış ve beş replikasyonun ortalama değerleri dikkate alınmıştır.

Daha önce de bahsedildiği gibi, sistemin birçok performans ölçütü ile tanımlamak mümkündür. Bu çalışmada, gerçek üretim ortamlarında gözlenen ve öncelikli alınan aşağıdaki performans ölçütleri gözönüne alınmıştır:

- Ortalama akış zamanı,
- Geciken sipariş sayısı,
- Ortalama gecikme süresi,
- Sistemin ortalama ara stok düzeyi,
- Geciken sipariş sayısının yüzde oranı,
- Pozitif gecikme oranı.

3.2.8. Simülasyon Deneyleri

Simülasyon çalışmasında kullanılan sistem, çok ürünlü bir sipariş tipi üretim sistemidir.

Siparişler sisteme poisson geliş sürecine göre ulaşmaktadırlar. Siparişlerin gelişler arası süresi, temel başlangıç olarak ortalaması 2.2 olan üstel dağılıma uymaktadır.

Sisteme yedi tip ürünün gelmesi olasıdır. Gelen siparişin tipi baskılı, jet boya malı, ipliği boyalı gömlek, pamuklu, kord, ipliği boyalı perilen, veya düz boyalı pe/vis olabilir. Sayılan her bir tip siparişin sisteme gelme olasılıkları da sırasıyla 0.9, 0.23, 0.6, 0.18, 0.16, 0.2 ve 0.8 şeklindedir.

Sipariş büyüklükleri de benzer şekilde belirlenmektedir. Gelen siparişin büyüklüğü,

- %20 olasılıkla 2500 metre,
- %40 olasılıkla 4000 metre,
- %25 olasılıkla 6000 metre,
- %10 olasılıkla 10000 metre
- %5 olasılıkla da 25000 metre

olarak gerçekleşmektedir.

Gelen sipariş, iç veya dış pazara ait olabilir. Bu duruma göre siparişlerin termin zamanları belirlenmektedir. Dış pazara ait siparişlerin termin zamanı sürekli düzgün dağılıma göre 2 ile 3 hafta arasındayken, iç pazar için siparişin termin zamanı 3 ile 4 hafta arasında, yine sürekli düzgün dağılıma göre belirlenecektir. Başlangıçta siparişlerin %30 olasılıkla dış pazara ve %70 olasılıkla iç pazara ait olduğu varsayılmıştır.

Gelen sipariş tiplerinden herbirinin rotası belirlidir. Ürünler, rotalarındaki iş merkezlerinden sırasıyla geçmek durumundadırlar. Bu rotalar ürünler için aşağıdaki gibidir:

Pamuklu tipte siparişler için:

Haşlısökümü, Bekletme, Kombinkasar, Merserize, Yıkama, Kurutma, Zımpara, Yıkama, Kurutma, Boyama, Bekletme, Yıkama, Kurutma, Apre, Sanfor, Katlama

Kord tipi sipariřler iin:

Kombinkasar, Yıkama, Kurutma, Boyama, Bekletme, Yıkama, Kurutma, Apre, Sanfor, Katlama

Jet boyama sipariřleri iin:

Yıkama, Kurutma, Yakma, Fikse, Jetboyama, Kurutma, Apre, Sanfor, Katlama

Baskılı tipte sipariřler iin:

Hařılskm, Bekletme, Kombinkasar, Merserize, Yıkama, Kurutma, Baskı, Kondanse, Apre, Sanfor, Katlama

Pe/vis sipariři iin:

Hařılskm, Bekletme, Yıkama, Kurutma, Boyama, Kurutma, Apre, Katlama

Gmlek sipariřleri iin:

Yıkama, Kurutma, Yakma, Merserize, Kurutma, Apre, Sanfor, Katlama

Perilen sipariřleri iin:

Yıkama, Fikse, Apre, Sanfor, Katlama

Sisteme gelen sipariřin rn cinsine gre renk aısından belirli zellikleri olacaktır ve bu zellikler bazı makine hazırlık zamanları zerinde etkilidir. Gelen pamuklu gabardin, kord, jet boya malı ve dz boyalı pe/vis sipariřlerinin %30'u aık, %30'u orta ve %40'ı koyu renklidir. Bu rnler, sistemde boyama veya baskı iřlemlerine gre beyaz olarak iřlem grrlerken, iplięi boyalı olarak gelen gmlek ve perilen tipleri, sisteme girdikten sonra renkli olarak iřlem greceklendir.

retim ařamasında yıkama, merserize, boyama ve jet boyama iřlemlerinde renk farklılıęı sebebiyle hazırlık zamanları oluřmaktadır. Yıkama iřleminde renkli sipariřten beyaza veya beyaz sipariřten renkliye geiřte iki saatlik hazırlık sresi gerekmektedir. Merserize iřleminde ise bu sre drt saattir. Boyama ve jet boyama iřlemlerinde en son iřlem gren sipariřin renk tonuna gre deęiřen hazırlık sresi olacaktır. En son olarak aık renkli sipariřin iřlemi yapılmıřsa, orta veya koyu renk sipariře geiřte hazırlık sresi gerekmeyecektir. Ancak eęer son olarak koyu renkli sipariř iřlenmiřse, aık renkli bir sipariře geiř iin  saat, ve eęer orta renkli bir sipariř son olarak iřlenmiřse aık veya koyu renkli sipariře geilebilmesi iin de makine temizlik sebebiyle bir saatlik hazırlık sresi oluřacaktır.

Siparişler, yukarıda belirtilen şekilde sisteme girdikten ve özellikleri belirtildikten sonra rotaları gereği gitmeleri gereken iş merkezlerinden, 3.2.6. bölümünde, algorithmada belirtilen önceliklere göre sıralanarak üretimleri gerçekleştirilir. Bu sırada, her sipariş için istenen performans değişkenleri de hesaplanacaktır.

Çalışmada,

- üretimi tamamlanan siparişin sistemde ne kadar kaldığını belirten *ortalama akış zamanı*, tüm siparişler için toplanan toplam akış zamanının, tamamlanan sipariş sayısına bölünmesiyle elde edilir.
- Tamamlanan sipariş miktarı düşülerek, yeni giren sipariş miktarları da eklenerek, sistemde o andaki toplam miktar hesaplanır. Bu değerin zaman ağırlıklı ortalaması, metre cinsinde *ara stok düzeyini* verecektir.
- Siparişin üretimi tamamlandığında, o sipariş için belirlenmiş olan termin zamanı ile arasındaki farka bakılarak, eğer iş termin tarihinden önce bitmiş ise, erken tamamlanan iş; eğer termin tarihinden sonra bitmiş ise geç kalmış iş olarak sayılır.
- Geç kalan siparişlerin hem sayısı, hem de ne kadar geç kaldıklarını ölçmek amacıyla *ortalama geç kalma süreleri* hesaplanır.
- Toplam geç kalma süreleri, tamamlana işlerin sayısına bölünerek, *pozitif gecikme* bulunur.
- *Geç kalan işlerin sayısı*, üretilen sipariş sayısına bölünerek de *geçiken işlerin oranı (geç kalma yüzdesi)* hesaplanır.

Geliştirilen algoritmaların etkinliklerini ölçmek amacıyla, karşılaştırma için kullanılacak çizelgeleme kuralları ve seçim kriterleri aşağıda sıralanmıştır:

▪ En Yakın Termin Zamanı (EDD) Algoritması

Siparişlerin sisteme gelişi ve özellikleri ile ilgili kısımlar, yukarıda anlatılanlarla aynı olmakla beraber, üretim aşamasında siparişlerin önceliklendirilmesi ve sıralanması EDD kuralına göre yapılmaktadır.

Siparişler, rotalarının gerektirdiği işlem için ilgili iş merkezinin önünde

kuyruğa girerler. Her bir iş merkezinde, sonraki işin seçiminde kuyruktaki işlerin termin zamanlarına bakılarak, termin zamanı en yakın iş öncelikli olur ve bir sonraki işlenecek iş olarak kuyruktan seçilerek alınır. Bu kural her makinede, siparişler tamamlanana kadar devam eder.

EDD kuralı, daha önce de bahsedildiği gibi, statik bir çizelgeleme kuralıdır. İşlerin diğer özelliklerini (işlem süresi, operasyon sayısı gibi) değil, termin zamanını dikkate alır. Gecikmeleri minimize etmede genellikle başarılı olan bu kural, uygulamada gerçek sistemlerde de en çok kullanılan kurallardan biridir.

▪ İlk Giren İlk Çıkar (FCFS) Algoritması

Siparişlerin sisteme gelişi ve özellikleri ile ilgili kısımlar, yukarıda anlatılanlarla aynı olmakla beraber, üretim aşamasında siparişlerin önceliklendirilmesi ve sıralanması FCFS kuralına göre yapılmaktadır.

Siparişler, rotalarının gerektirdiği işlem için ilgili iş merkezinin önünde kuyruğa girerler. Her bir iş merkezinde, sonraki işlemin seçiminde kuyruktaki işlerin özelliklerine (işlem zamanı, termin zamanı gibi) bakılmaksızın, sisteme müdahale etmeden, işlerin kuyruğa geliş sırasına göre önceliklendirilmesi ve seçilerek işlenmesi sözkonusudur.

Bu kural, statik bir kuraldır ve sistemin herhangi bir düzeltme etkisi almadan nasıl çalışacağını görmek amacıyla karşılaştırma algoritmalarından biri olarak seçilmiştir.

▪ Başlangıç Zamanı (SD) Algoritması

Siparişlerin sisteme gelişi ve özellikleri ile ilgili kısımlar, yukarıda anlatılanlarla aynı olmakla beraber, üretim aşamasında siparişlerin önceliklendirilmesi ve sıralanması SD kuralına göre yapılmaktadır.

Siparişler, rotalarının gerektirdiği işlem için ilgili iş merkezinin önünde kuyruğa girerler. İlk iş merkezinde, sonraki işlemin seçiminde, termin zamanına ve normalde gerektirdiği üretim zamanına göre, en önce başlaması gereken iş, öncelikli

olarak seçilir ve iş merkezine alınarak üretime başlanır.

SD, statik bir kuraldır ve siparişlerin üretim zamanı ve termin zamanı gibi özelliklerini dikkate alarak seçimi gerçekleştirir.

- **Kalan Bolluk Zamanı (STR) Algoritması**

Siparişlerin sisteme gelişi ve özellikleri ile ilgili kısımlar, yukarıda anlatılanlarla aynı olmakla beraber, üretim aşamasında siparişlerin önceliklendirilmesi ve sıralanması STR kuralına göre yapılmaktadır.

Siparişler, rotalarının gerektirdiği işlem için ilgili iş merkezinin önünde kuyruğa girerler. Her bir iş merkezinde, sonraki işlemin seçiminde, kuyruktaki işler için bolluk zamanı (slack time) hesaplaması gerçekleştirilir. Bolluk zamanı, her iş için termin zamanı ve işin kalan operasyonlarının gerektirdiği işlem zamanı arasındaki farktır, yani tüm işlemler eğer varsayıldığı işlem zamanında bitirilecek olursa, termin zamanından önce kalacak bolluk zamanını ifade eder. Bolluk zamanı en az olan işe öncelik verilerek bu seçim gerçekleştirilir.

STR, dinamik bir kuraldır. Siparişin rotasındaki her işlem aşamasında, siparişin termin zamanını ve kalan işlem zamanlarını dikkate alarak bir seçim gerçekleştirir.

- **Operasyon Başına Kalan Bolluk Zamanı (S/RO) Algoritması**

Siparişlerin sisteme gelişi ve özellikleri ile ilgili kısımlar, yukarıda anlatılanlarla aynı olmakla beraber, üretim aşamasında siparişlerin önceliklendirilmesi ve sıralanması S/RO kuralına göre yapılmaktadır.

Siparişler, rotalarının gerektirdiği işlem için ilgili iş merkezinin önünde kuyruğa girerler. Her bir iş merkezinde, sonraki işlemin seçiminde, kuyruktaki işler için yukarıda da anlatıldığı şekilde bolluk zamanı hesaplaması gerçekleştirilir. Her iş için bulunan bolluk zamanı, işin kalan işlem sayısına bölünerek, operasyon başına işin kalan bolluk zamanı hesaplanır. Operasyon başına kalan bolluk zamanı en az olan işe öncelik verilerek işleme alınır.

S/RO kuralı, dinamik bir seim kuralıdır. retim her seim aamasında iler iin termin zamanı, bolluk zamanı ve kalan operasyon sayılarını dikkate alarak seimi gerekletirir.

▪ **En Az İlemi Kalan (LWKR) Algoritması**

Siparilerin sisteme gelii ve zellikleri ile ilgili kısımlar, yukarıda anlatılanlarla aynı olmakla beraber, retim aamasında siparilerin nceliklendirilmesi ve sıralanması LWKR kuralına gre yapılmaktadır.

Sipariler, rotalarının gerektirdii ilem iin ilgili i merkezinin nnde kuyrua girerler. Her bir i merkezinde, sonraki iin seiminde kuyruktaki ilerin kalan toplam ilem zamanlarına bakılarak, gerektirdii ilem zamanı en az olan i, ncelikli olur ve bir sonraki ilenecek i olarak kuyruktan seilerek alınır. Bu kural her makinede, sipariler tamamlanana kadar uygulanır.

LWKR kuralı, dinamik bir kuraldır. Kuyruklardan ilenmek zere ilerin seiminin her aamasında, her bir i iin ilem zamanları verileri kullanılarak, kalan ilem zamanları hesaplanır ve seim gerekletirilir. Uygulamada olduka baarılı sonular verdii grlmtr.

▪ **En Fazla İlemi Kalan (MWKR) Algoritması**

Siparilerin sisteme gelii ve zellikleri ile ilgili kısımlar, yukarıda anlatılanlarla aynı olmakla beraber, retim aamasında siparilerin nceliklendirilmesi ve sıralanması MWKR kuralına gre yapılmaktadır.

Sipariler, rotalarının gerektirdii ilem iin ilgili i merkezinin nnde kuyrua girerler. Her bir i merkezinde, sonraki ilemin seiminde, yukarıda da anlatıldıı ekilde kuyruktaki iler iin kalan ilem zamanları hesaplanır. Gerektirdii toplam ilem zamanı halen en fazla olan i seilerek, ilenmek zere i merkezine alınır.

En fazla ilem sresi kalan ii seme kuralı, MWKR dinamik bir kuraldır. Seim sırasında, kalan ilem sreleri toplamına bakarak karar verir. Bu kural

uygulamada, oldukça iyi bir performans gösteren bir kuraldır.

▪ **En Fazla Operasyonu Kalan (MOPNR) Algoritması**

Siparişlerin sisteme gelişi ve özellikleri ile ilgili kısımlar, yukarıda anlatılanlarla aynı olmakla beraber, üretim aşamasında siparişlerin önceliklendirilmesi ve sıralanması MOPNR kuralına göre yapılmaktadır.

Siparişler, rotalarının gerektirdiği işlem için ilgili iş merkezinin önünde kuyruğa girerler. Her bir iş merkezinde, sonraki işlemin seçiminde kuyruktaki işlerin diğer özellikleri (işlem zamanı, termin zamanı gibi) dikkate alınmaksızın, işin tamamlanmasına kadar kalan operasyon sayılarına bakılır. Kalan operasyon sayısı en fazla olan iş, bir sonraki işlenecek iş olarak belirlenir.

Bu kural dinamik bir çizelgeleme kuralıdır, ancak işin sadece operasyon sayılarını dikkate alır.

▪ **En Kısa İşlem Süresi (SPT) Algoritması**

Siparişlerin sisteme gelişi ve özellikleri ile ilgili kısımlar, yukarıda anlatılanlarla aynı olmakla beraber, üretim aşamasında siparişlerin önceliklendirilmesi ve sıralanması SPT kuralına göre yapılmaktadır.

Siparişler, rotalarının gerektirdiği işlem için ilgili iş merkezinin önünde kuyruğa girerler. Her bir iş merkezinde, sonraki işlemin seçiminde kuyruktaki işlerin o iş merkezinde gerektirdiği işlem zamanına bakılır. O iş merkezindeki işi en çabuk bitecek olan, yani işlem süresi en az olan iş seçilerek işleme alınır. Bu kural her makinede, siparişler tamamlanana kadar uygulanır.

SPT, statik, ve sadece bulunduğu iş merkezi kuyruğunu dikkate aldığından miyopik bir kuraldır. İşin sadece, her makinede gerektirdiği işlem zamanı bilgilerini kullanır. Bulunduğu iş merkezindeki işlerin çabukça hareket etmesini ve işleme girmesini sağladığından, ortalama akış zamanı ve bekleyen iş sayısı minimizasyonları açısından oldukça iyi performans gösteren ve uygulamada da sıkça kullanılan bir çizelgeleme kuralıdır.

4. ARASTIRMA BULGULARI VE TARTISMA

4.1. Simlasyon Sonuları

Deęişken darboęazlı sistemlerde izelgeleme yapmak amacıyla geliřtirilen algoritmaların performansının gzlenmesi iin, karřılařtırma yapılacak dokuz adet sezgisel izelgeleme kuralı kullanılmıřtır.

Seilen kurallar, oęunluęu dinamik zellikte ve uygulamada sıka kullanılan ve performans ltlerinin optimizasyonunda olduka iyi sonular veren kurallardır. zellikle SPT ve LWKR kurallarının ortalama akıř sresi minimizasyonunda, dolayısıyla da ara stok dzeyi ve gecikmelerde olduka iyi sonular vermesi beklenir. EDD, termin zamanına gre hareket eden bir kural olduęundan, gecikmeleri minimize etmesi beklenir. FCFS, sistemi varolan řartlılarıyla alıřtırdıęından, iyi bir karřılařtırma kuralı olabilir. Seilen dięer kurallar da, STR, MOPNR, MWKR, S/RO, SD olarak sıralanabilir.

alıřmada bařlangı adımında, ortalaması 2.2 olan stel daęılıma gre sipariř geliř hızının belirlendięi bir sistem modellenmiřtir. Bu sistemin, yukarıda sayılan sezgisel izelgeleme kuralları iin simlasyon modelleri yazılmıř (EK 2) ve modeller alıřtırılarak performans ltleri deęerlerine bakılmıřtır (Tablo 4.1.).

nceki blmde bahsedilen darboęaz kaynaęa odaklanarak sistemi izelgeleme mantıęını arařtırılırken, rnek olması ve uygulamayı kolaylařtırması amacıyla, darboęaz deęiřiminin sadece iki iř merkezi arasında gerekleřtięi durum gznne alınmıřtır. Bu iř merkezlerinin tespit edilmesi amacıyla, sistemi normal kořulları altında alıřtıran ve iř merkezlerinin anlık olarak darboęaz olma durumlarına bakan bir model geliřtirilmiřtir (EK 3). Ardından, tespit edilen iř merkezlerine odaklanan, darboęaz kaynaęın iř ykn dengelemeye ynelik olarak, uygun iřlem zamanlı iřin ncelikli olması mantıęına dayanan Algoritma 1 (EK 4) ile, darboęaz kaynaęın iř ykn dengelemek iin, iřlerin darboęaz kaynaęa ulařma hızlarına baęlı olarak nceliklerinin belirlendięi Algoritma 2 (EK 5) iin simlasyon modelleri oluřturulmuřtur. Her iki algoritma iin, darboęaz kaynakların zaman stoęu alt ve st sınırlarını deęiřtirerek modeller birok farklı durumda yeniden denenmiřtir.

Tablo 4.1. exp(2.2) Geliş Oranı İçin Sezgisel Kuralların Performans Ölçütleri Sonuçları

	EDD	FCFS	SD	STR	S/OP	LWKR	MWKR	MOPNR	SPT
Ak Zaman Ortalaması	284,03	278,584	304,272	257,77	303,506	127,49	355,648	317,884	134,582
Ak Zaman Maksimum De eri	2873,44	2828,22	2949,82	21085	22100,2	17456,96	23537,8	3261,8	11513,86
Gelen Sipari Say s	15808,6	15808,6	15808,6	15808,6	15808,6	15808,6	15808,6	15808,6	15808,6
Tamamlanan Sipari Say s	15751	15745,8	15750	15718,2	15706,2	15802,4	15645,6	15736,6	15801
Toplam WIP	587,008	570,98	637,982	745,16	870,068	214,6	1138,108	676,962	214,812
Toplam WIP Maksimum De eri	5180	4965	5495	6320	7180	1940	9515	5905	1900
Boyama Merkezi Kullan m Oran	94,023	94,035	93,942	93,977	93,954	93,981	93,993	93,950	93,966
Bask Merkezi Kullan m Oran	99,513	99,500	99,505	99,500	99,322	99,626	99,377	99,241	99,548
Geç Kalan Sipari Say s	1193,8	1429	1232,6	814,6	1401	338	1901,6	2022,8	350,4
Erken Biten Sipari Say s	14557,2	14316,8	14517,4	14903,6	14305,2	15464,4	13744	13713,8	15450,6
Ortalama + Gecikme	18,07	18,71	18,92	15,3	18,32	4,198	32,75	26,36	5,873
Ortalama - Gecikme	2,723	2,835	2,603	2,7317	2,605	3,182	2,846	2,9299	3,192
Geciken İşlerin Yüzde Oranı	7,579	9,075	7,826	5,1825	8,92	2,138	12,154	12,85	2,217
Siparişlerin Toplam Geç Kalma Süresi	1422801	1472755	1489880,1	1202383,7	1438807,1	331730,4	2562107	2074134	463961,7

Algoritma 1 ve Algoritma 2'ye ait modellerin her biri 11 farklı alt ve üst sınır değerleriyle yeniden çalıştırılmıştır. Tesadüfi olarak belirlenen alt ve üst sınırlar, aşağıda, Tablo 4.2.'de gösterilmiştir.

Tablo 4.2. Simülasyon Deneylerinde Kullanılan Zaman Stoğu Alt ve Üst Sınırları

Algoritma 1	Algoritma 2	Boyama Alt Sınır	Boyama Üst Sınır	Baskı Alt Sınır	Baskı Üst Sınır
DBR20	DBR30	20	40	20	40
DBR21	DBR31	30	80	30	80
DBR22	DBR32	20	90	20	90
DBR23	DBR33	40	70	40	70
DBR24	DBR34	50	100	50	100
DBR25	DBR35	100	200	100	200
DBR26	DBR36	24	48	36	72
DBR27	DBR37	36	72	48	96
DBR28	DBR38	48	96	36	72
DBR29	DBR39	72	144	48	96
DBR201	DBR301	30	50	30	50

Sistemin başlangıç koşullarında bu iki farklı algoritmaya ait modeller için performans ölçütleri değerleri, Tablo 4.3. ve Tablo 4.4.'de özetlenmektedir.

Akış Zamanı:

Sezgisel çizelgeleme kurallarına ait modeller ile, geliştirilen iki algoritmaya ait modeller incelendiğinde, akış zamanı açısından en kötü sonucu, MWKR kuralının verdiği söylenebilir. Termin zamanını dikkate alan ve sıkça kullanılan EDD kuralının, MWKR kadar olmasa da, sisteme müdahale etmeyen FCFS kuralından bile daha kötü sonuç verdiği görülmektedir. Akış zamanı ortalaması açısından en iyi sonuçlar, beklendiği şekilde SPT ve en iyi olarak da, LWKR kuralı ile elde edilmektedir. Bu kuralların, siparişlerin kaynaklar tarafından işlenmesini çabuklaştırarak, beklemeyi en aza indirdikleri bilinmektedir.

Tablo 4.3. exp(2.2) Geliş Oranı İçin 1. Algoritma Performans Ölçütleri Sonuçları

Zaman Stoğu Alt-Üst Sınırları

Boyama i merkezi

Bask i merkezi

20-40 30-80 20-90 40-70 50-100 100-200 24-48 36-72 48-96 72-144 30-50

20-40 30-80 20-90 40-70 50-100 100-200 36-72 48-96 36-72 48-96 30-50

DBR20 DBR21 DBR22 DBR23 DBR24 DBR25 DBR26 DBR27 DBR28 DBR29 DBR201

Ak Zaman Ortalaması	141,13	136,91	136,758	137,082	137,446	136,078	137,65	138,102	137,796	137,38	136,732
Ak Zaman Maksimum De eri	4025,46	3685,74	3690,02	3652,3	3688,46	3661,68	3657,9	3686,92	3662,42	3705,62	3700,42
Gelen Sipari Say s	15808,6	15808,6	15808,6	15808,6	15808,6	15808,6	15808,6	15808,6	15808,6	15808,6	15808,6
Tamamlanan Sipari Say s	15833	15822	15822,6	15824,2	15823,2	15825	15823,2	15821	15817,4	15824,4	15822,2
Toplam WIP	150,22	150,434	149,928	149,872	151,342	149,926	150,11	151,552	151,862	151,778	149,786
Toplam WIP Maksimum De eri	2040	1965	2025	1935	2000	1985	1960	2050	2100	2135	2020
Boyama Merkezi Kullan m Oran	95,987	95,977	95,999	96,082	96,015	96,050	95,964	96,009	96,031	96,018	96,002
Bask Merkezi Kullan m Oran	91,953	91,981	91,992	91,946	91,961	91,993	91,977	91,961	91,913	91,970	91,958
Geç Kalan Sipari Say s	299	298,8	296,2	300,8	297,6	291,4	301,8	305,8	300,6	304,4	295,4
Erken Biten Sipari Say s	15524	15523,2	15526,4	15523,4	15525,6	15533,6	15521,4	15515,2	15552,8	15520	15526,8
Ortalama + Gecikme	2,22	2,056	2,061	2,019	2,054	2,02	2,022	2,081	2,027	2,067	2,05
Ortalama - Gecikme	3,016	3,023	3,022	3,02	3,017	3,024	3,016	3,01	3,018	3,021	3,023
Geciken İşlerin Yüzde Oranı	1,889	1,88	1,872	1,9	1,88	1,841	1,907	1,932	1,9	1,923	1,86
Siparişlerin Toplam Geç Kalma Süresi	159933	182655	183017,02	159780,92	162514,16	159870,4	159975,2	164638,3	160303,5	163541,3	162168,8

Tablo 4.4 exp(2.2) Geliş Oranı İçin 2. Algoritma Performans Ölçütleri Sonuçları

Zaman Stoğu Alt-Üst Sınırları

Boyama i merkezi

Bask i merkezi

Boyama i merkezi	20-40	30-80	20-90	40-70	50-100	100-200	24-48	36-72	48-96	72-144	30-50
Bask i merkezi	20-40	30-80	20-90	40-70	50-100	100-200	36-72	48-96	36-72	48-96	30-50
	DBR30	DBR31	DBR32	DBR33	DBR34	DBR35	DBR36	DBR37	DBR38	DBR39	DBR301
Ak Zaman Ortalamas	136,208	136,808	136,648	137,458	136,646	136,794	136,138	136,19	137,484	136,528	136,646
Ak Zaman Maksimum De eri	3705,78	3595,12	3634,08	3647,32	3633,02	3687,42	3691,76	3625,8	3661,52	3635,48	3593,28
Gelen Siparİ Say s	15808,6	15808,6	15808,6	15808,6	15808,6	15808,6	15808,6	15808,6	15808,6	15808,6	15808,6
Tamamlanan Siparİ Say s	15823,6	15823,8	15825	15826,6	15824,8	15827	15821	15828,6	15822,4	15821,8	15821,6
Toplam WİP	152,298	153,024	152,966	153,972	152,794	153,218	152,058	152,404	154,296	152,174	153,048
Toplam WİP Maksimum De eri	2010	1950	2100	2075	2010	2045	1965	1965	2080	2000	2020
Boyama Merkezi Kullan m Oran	96,015	96,019	96,019	96,029	96,038	96,053	96,006	96,003	96,011	96,022	96,015
Bask Merkezi Kullan m Oran	91,929	91,943	91,960	91,946	91,970	91,934	91,936	91,961	91,925	91,922	91,952
Geç Kalan Siparİ Say s	303,6	299,6	300,4	304	303,2	308,4	297,8	305,4	306,4	301,6	305,2
Erken Biten Siparİ Say s	15520	15524,2	15524,6	15522,6	15521,6	15518,6	15523,2	15523,2	15516	15520,2	15516,4
Ortalama + Gecikme	2,06	2,032	2,079	2,056	2,048	2,049	2,055	2,094	2,071	2,067	2,047
Ortalama - Gecikme	3,03	3,024	3,027	3,02	3,025	3,026	3,029	3,029	3,023	3,029	3,027
Geciken İřlerin Yüzde Oranı	1,91	1,89	1,89	1,92	1,91	1,948	1,882	1,92	1,936	1,906	1,92
Sipariřlerin Toplam Geç Kalma Süresi	162948	160735,28	164518,66	162686,82	162054,2	162153,02	162600,02	165738,94	163865,34	163549,3	161933,1

Her ne kadar, geerli bir performans lt sayılmasa da, akıř zamanının her kural iin gerekleřen maksimum deėerine de bakılabilir. Bu deėerler, tm alıřma sresi boyunca belki de tek bir kez gzlendiklerinden, karřılařtırma iin iyi bir lt deėillerdir, ancak sistemin akıř dengesinin tahmin edilmesi aısından, deėerlendirilebilecek bir lttr. Burada, akıř zamanı ortalaması aısından en iyi sonuları verdiėi sylenen SPT ve LWKR kurallarının, en yksek akıř zamanını da gerekleřtiren kurallar olduėu grlmektedir. Bu kurallar, en kısa iři kalan veya o iři merkezindeki iřlemi en kısa srecek iře ncelik verdiklerinden, bu zellikleri saėlamayan iřlerin, bazen uzun bir sre arka sıralara itilmesine ve iřlenmemesine sebep olmakta ve bu yzden akıř zamanı maksimum olarak gzlenmektedir. Gerek hayatta, bunun bazı mřteri sipariřlerinin ok uzun sreler sonrasında karřılanabilmesi olarak ortaya ıkacaėı sylenebilir ki, bu da kabul edilemez bir durumdur.

Geliřtirilen algoritmalarla elde edilen akıř zamanları ortalamalarının, zellikle de bazı limitler iin, SPT kuralına olduka yakın deėerler verdiėi sylenebilir. En iyi sonu, DBR25 ve DBR36 modelleri ile elde edilmiřtir.

Geliřtirilen algoritmalar iin akıř zamanı maksimum deėerine bakıldıėında, ok aykırı deėerlere rastlanmamakta, bu da sistemin dengeli bir řekilde alıřtığını ve hibir sipariřin gereėinden fazla bekletilmediėini gstermektedir.

Akıř zamanları aısından incelendiėinde; geliřtirilen algoritmaların, en iyi sonu veren sezgisel izelgeleme kuralları ile neredeyse bařabař sonular verdiėi gzlenmektedir. Bu sonular, geliřtirilen algoritmaların kullanılması halinde, en dřk ortalama akıř sresine yakın deėerlerin elde edilebileceėini, buna ilave olarak hibir sipariř ok uzun sre beklemeyeceėini gstermektedir.

Gelen ve Tamamlanan Sipariř Sayıları:

Simlasyon modelinin alıřtırıldıėı sre boyunca, modellenen her algorithmada sisteme giren sipariř sayısı aynıdır. Bunun byle olması karřılařtırma yapabilmek aısından da bir gerekliliktir.

Tamamlanan sipariř sayıları arasında byk farklar gzlenmemektedir. Bunun sebebi, sıralama kuralı ne olursa olsun kaynak kapasitelerinin

değişmemesidir. Meydana gelen küçük farklar ise sezgisel sıralama kurallarında ya da geliştirilen algoritmalarda kaynakların boş kalma sürelerinin azaltılması ile elde edilen bazı avantajlar ile oluşmaktadır.

Üretimi tamamlanan sipariş sayısında ise, geliştirilen algoritmalarla elde edilen sonuçlar, sezgisel algoritmalara göre göreceli olarak daha iyidir. Tamamlanan sipariş sayısı için elde edilen en iyi değer 1. Algoritma ile 15833 sipariş iken, sezgisel çizelgeleme kurallarında, LWKR ile 15802' dir. Diğer sezgisel çizelgeleme kurallarında bu sayı daha da azdır.

Bu durumda, geliştirilen algoritmaların, sipariş sayısı açısından, sistemin toplam çıktı düzeyini artırdığı da söylenebilir, bu da istenen bir durumdur.

Ortalama Arastok Düzeyi:

Tüm sistemin ortalama ara stok düzeyine baktığımızda, karşılaştırma için kullanılan sezgisel algoritmalarda en kötü sonucun yine MWKR, en iyi sonucun da beklendiği gibi LWKR ve SPT kurallarıyla elde edildiği görülmektedir.

Geliştirilen çizelgeleme algoritmalarında, her limit için ve her iki algoritma için ortalama ara stok düzeyinin çok daha düşük olduğu da gözlenebilir. Ortalama ara stok düzeyinde, DBR201 algoritmasıyla, EDD kuralına göre %75, ve en iyi sezgisel çizelgeleme kuralına (LWKR) göre de %31, iyileştirme sağlanmıştır.

Bu başarılı sonucun elde edilmesi, geliştirilen algoritmalarla, sistemde dengeli bir akışın meydana getirilmesi, darboğaz kaynaklardan optimum yararlanacak şekilde sipariş akışlarının düzenlenmesi ile mümkün olmuştur. En çok ara stoğun darboğaz kaynaklar önünde oluşması doğaldır. Geliştirilen algoritmalarda bu stoğun belirlenen bir seviyenin üzerine çıkması halinde, bu seviyeyi düşürücü tedbirler alınmıştır. Geliştirilen 1. Algoritmada, bu kaynaklarda işlenmeyen siparişlere öncelik verilmiş ya da en kısa işlem süreli olanlar gönderilmiştir. Bu da zamanla darboğaz kaynaklar önündeki ara stok düzeyinin düşmesini sağlamıştır. Geliştirilen 2. Algoritmada ise, darboğaz kaynaklara ulaşma süresi en uzun olan siparişlere öncelik verilmiş, dolayısıyla diğer kaynakların daha aktif kullanımı sağlanırken darboğaz kaynaklar önündeki ara stoklar da azaltılmıştır.

Darboaz İř Merkezleri Kullanım Dzeyleri:

Simlasyon sresinin ok uzun olması, iř merkezi kullanım oranlarının yakın olması sonucunu dourmuřtur.

Sezgisel izelgeleme kurallarında yksek kullanım oranları vardır ve kurallar bazında byk farklılıklar gstermemektedir. Boyama iř merkezinin kullanım oranı %94 iken Baskı iř merkezinde bu deer %99.5 civarındadır. Bu oranların tatmin edici dzeyde olduđu sylenebilir, ancak aynı zamanda yıılmalar sebebiyle darboaz kaynakların yksek kullanım oranlarına ramen dier kaynakların etkin kullanılmadıı, performans ltlerinden de grlebilmektedir.

Geliřtirilen algoritmalar sipariř akıřını kaynak kullanımlarını dengeleyecek řekilde dzenlemiř, Boyama iř merkezinde kullanım oranını artırırken (%96), Baskı iř merkezinde bu oran dřmřtr (%92). Sistemin tm performans deiřkenleri incelendiinde; Boyama iř merkezindeki geliřmenin toplam sistem performansına ciddi oranda olumlu etkisi varken, Baskı iř merkezinde kullanım oranının dřmesi toplam sistem performansını olumsuz etkilememektedir.

Geliřtirilen algoritmalar herhangi bir kaynaın kullanım oranını artırmak yerine, dinamik yapısı ile ihtiya duyulan zamanda ihtiya duyulan kaynaın kullanım oranını artıracak řekilde alıřmaktadır. Kullanılacak alt ve st zaman stou deerleri ile daha etkin bir sonu elde edilebilmesi de muhtemeldir.

Ge Kalan Sipariř Sayısı:

Ge kalan sipariř sayılarına bakıldıında, hemen tm sezgisel izelgeleme algoritmalarının kt sonu verdii, ve bunların arasında en iyi sonucun da yine LWKR ve SPT ile elde edildii grlmektedir. Geliřtirilen izelgeleme algoritması DBR201 ile, sezgisel kuralların arasında en iyi sonucu veren LWKR kuralına gre %12.6 ve SPT kuralına gre %15.6 daha iyi sonu verdii grlmektedir. Termin zamanlarına gre izelgeleme yapan ve iyi sonu vermesi beklenen EDD kuralına gre ise bu iyileřtirme %75 dzeyindedir.

Bu iyileřtirme, dier iyileřtirmelere paraleldir. Dier performans deiřkenlerini etkileyen iyileřtirmeler bu parametrenin de olumlu olması sonucunu dourmuřtur. Akıř zamanı azalıdıından ve sipariřlerin iřleme giriřleri, darboaz

kaynaklar tersini gerektirmediği sürece, sisteme girdiği sıralamada olduğu için siparişlerin gecikmesi asgari düzeyde tutulabilmektedir.

Ortalama Pozitif Gecikme:

Ortalama pozitif gecikmede, daha önceki performans ölçütlerinde iyi sonuç veren SPT kuralının etkin olmadığı görülmektedir. Bunun sebebi, en kısa işlem süreli işlere öncelik verilirken bazı siparişlerin çok uzun süreler bitirilemeden beklemesidir.

En iyi sonuç, LWKR kuralı ile elde edilmiştir. Geliştirilen çizelgeleme algoritmalarında ise, bahsedilen oranın yarı yarıya düştüğü, yani en iyi sezgisel çizelgeleme algoritmasının sonucuna göre %48.3 iyileştirme sağlamaktadır. EDD kuralına karşı bu iyileştirme %56.5 düzeyine kadar çıkmaktadır.

Geliştirilen algoritmaların önceki kısımlarda da belirtilen, etkin kaynak kullanım özelliği ve siparişlerin olabildiğince sıralı olarak işlenmesine imkan tanınması bu olumlu sonuçların elde edilmesini sağlamaktadır.

Geciken İşlerin Oranı:

Geciken işlerin yüzde oranlarında, sezgisel algoritmalarda en iyi sonuçlar LWKR ve SPT, en kötü sonuçlar ise FCFS kuralları ile elde edilmiştir.

Geliştirilen algoritma kurallarında, tüm zaman stoğu dilimlerinde daha düşük geç kalma oranları elde edilmektedir. EDD kuralına göre iyileştirme %75.7, SPT kuralına göre ise %17, FCFS kuralına göre ise %80 düzeyindedir.

Toplam Geç Kalma Süreleri:

Siparişlerin toplam geç kalma süreleri karşılaştırıldığında, en kötü sonuç MWKR kuralı ile elde edilirken, LWKR ve SPT, diğerlerine göre oldukça iyi sonuçlar vermişlerdir.

Geliştirilen algoritmalarla toplam geç kalma süresi, oldukça düşürülmüş ve en iyi sonucu veren LWKR kuralına göre %81 kadar iyileştirme sağlarken, EDD kuralına göre de %96 iyileştirme sağlamıştır.

4.2. Sipariř Geliř Hızının Etkileri

Geliřtirilen algoritmalar, rnek olarak alınan sistemde, nceki kısımda anlatıldıđı gibi, performans deđiřkenlerini iyileřtiren sonular vermiř ve genelde bařarılı olmuřtur. Performans deđiřkenleri zerinde etkili olabilecek eřitli sistem parametreleri de mevcuttur. Etkili parametrelerin deđiřimi durumunda geliřtirilen modellerin nasıl sonu vereceđi de incelenmelidir.

Bu amala, sıka karřılařılan talep geliř hızı deđiřimlerinin, sonulara etkisi test edilmiř ve deđiřimlerin performans ltleri zerindeki etkileri arařtırılmıřtır.

Sistem, farklı sipariř geliř hızları iin alıřtırılmıřtır. Bu parametrenin deđiřmesinde sistemin performans deđiřkeninin gzlenebilir bir řekilde deđiřmesi beklenir. nk sipariř geliř hızının deđiřmesiyle, kaynakların kullanım oranları, iřykleri oranları, dolayısıyla bekleyen sipariř sayıları, akıř zamanları ve ge kalma durumları da deđiřecektir.

Karřılařtırma yapmak amaıyla, sezgisel izelgeleme kurallarından, sistemin farklı zelliklerinin (termin zamanı, iřleme zamanı, operasyon sayıları gibi) dikkate alınabilmesini sađlamak amaıyla, iřletmelerde sıka kullanılan EDD, sisteme mdahale etmeyen FCFS, dinamik bir kural olan S/RO, ve tm performans ltlerinde en iyi sonuları veren LWKR ve SPT kuralları alınmıř, geliřtirilen her iki algoritmadan da seilen altıřar model alıřtırılarak sonular tablolar ve grafikler halinde gsterilmiřtir.

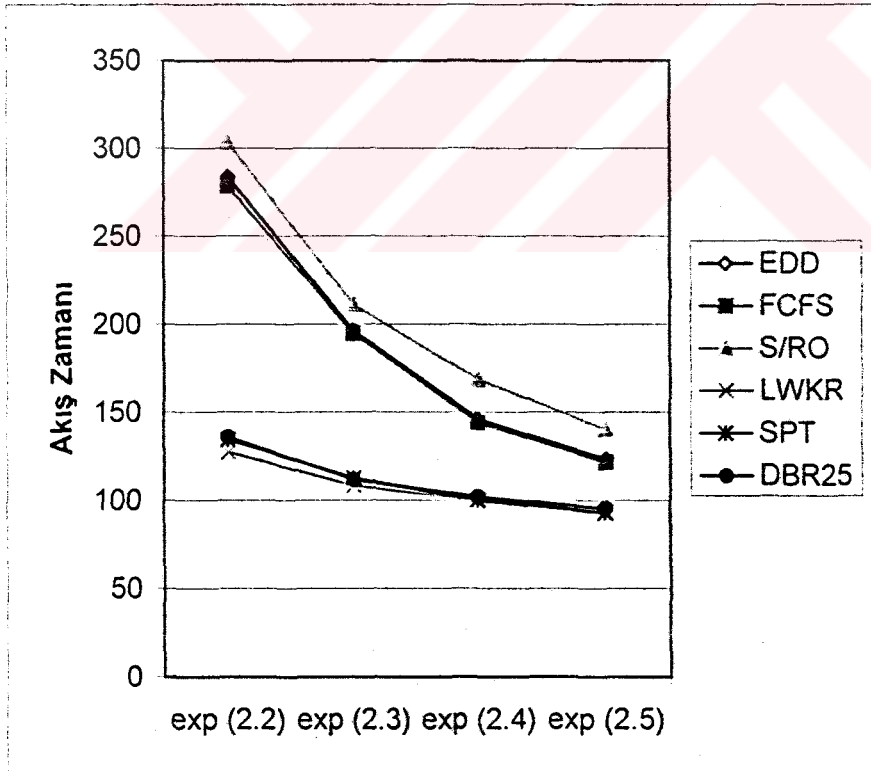
4.2.1. Sipariř Geliř Hızı – Ortalama Akıř Zamanı İliřkisi

Tablo 4.5. ile řekil 4.1.'de Algoritma 1, ve Tablo 4.6. ile řekil 4.2.'de Algoritma 2 yntemlerine ait rnek modellerin. seilen izelgeleme kuralları ile birlikte, Ortalama Akıř Zamanı deđiřimleri grlmektedir.

Sonulara gre, sipariř geliř hızları azaldıka akıř sreleri azalmakta, ancak bu azalma eđilimi her algoritmada aynı řekilde gerekleřmemektedir. Geliř sıklıklarının azalması, sistem kaynaklarının kullanımındaki yođunluđun azalmasına,

Tablo 4.5. Sipariř Geliř Hızı-Ortalama Akıř Zamanı İliřkisi (1. Algoritma)

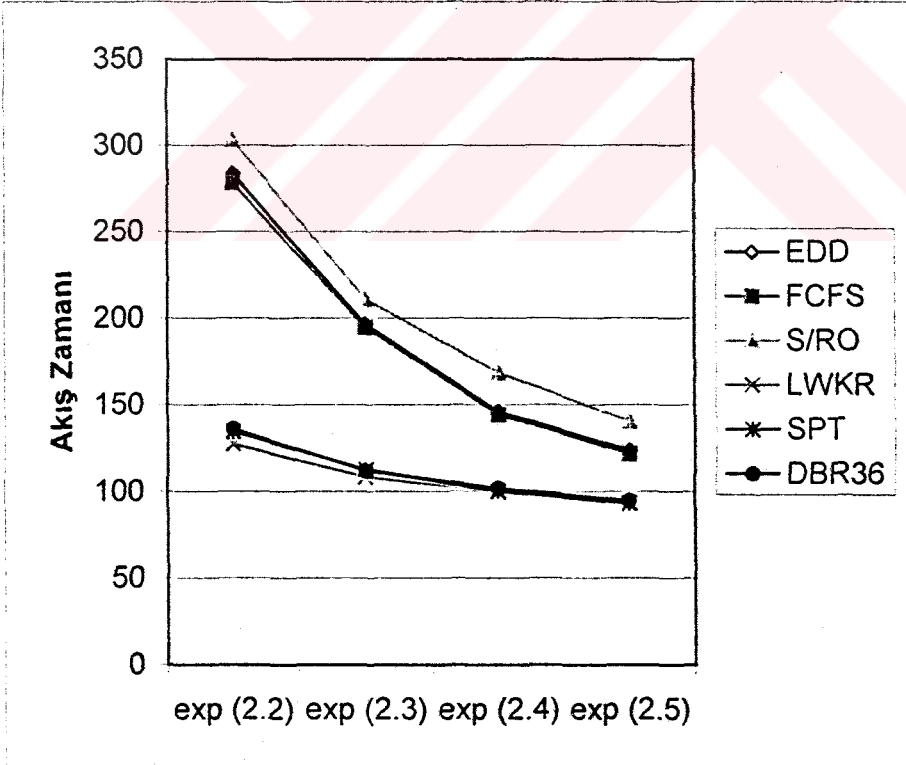
	exp (2.2)	exp (2.3)	exp (2.4)	exp (2.5)
EDD	284,030	196,618	146,142	123,668
FCFS	278,584	194,774	144,320	121,774
S/RO	303,506	211,270	168,674	140,604
LWKR	127,490	108,380	99,776	92,886
SPT	134,582	112,830	100,342	93,378
DBR21	136,910	112,114	102,082	94,622
DBR22	136,758	112,092	102,025	94,821
DBR23	137,082	112,622	102,311	94,859
DBR25	136,078	111,888	102,196	95,607
DBR29	137,380	112,082	101,992	94,520
DBR201	136,732	112,474	102,158	94,796



řekil 4.1. Sipariř Geliř Hızı-Ortalama Akıř Zamanı İliřkisi (1. Algoritma)

Tablo 4.6. Sipariş Geliş Hızı-Ortalama Akış Zamanı İlişkisi (2. Algoritma)

	exp (2.2)	exp (2.3)	exp (2.4)	exp (2.5)
EDD	284,030	196,618	146,142	123,668
FCFS	278,584	194,774	144,320	121,774
S/RO	303,506	211,270	168,674	140,604
LWKR	127,490	108,380	99,776	92,886
SPT	134,582	112,830	100,342	93,378
DBR31	136,808	112,450	102,489	95,011
DBR33	137,458	112,086	102,508	94,775
DBR36	136,138	111,938	102,002	94,746
DBR37	136,190	112,282	102,210	94,836
DBR39	136,528	112,206	102,435	94,895
DBR301	136,646	112,250	102,239	94,874



Şekil 4.2. Sipariş Geliş Hızı-Ortalama Akış Zamanı İlişkisi (2. Algoritma)

daha önce etkin sonuç vermeyen çizelgeleme kurallarının zaaflarının örtülmesine neden olmaktadır. Bu çizelgeleme kurallarında (EDD, FCFS, S/RO) Ortalama Akış Zamanı iyileşmesi diğer kurallara ve geliştirilen iki algoritmaya nazaran daha belirgindir.

Geliş sıklıkları azaldıkça, geliştirilen algoritmalarla, daha önce iyi sonuç veren çizelgeleme kuralları arasındaki performans farkı, sistemdeki darboğaz kaynağın iş geliş sıklığı azaldıkça etkinliğini kaybetmesi sebebiyle, azalmıştır.

4.2.2. Sipariş Geliş Hızı – Ortalama Ara Stok Düzeyi İlişkisi

Tablo 4.7. ile Şekil 4.3.'de Algoritma 1, ve Tablo 4.8. ile Şekil 4.4'te Algoritma 2 yöntemlerine ait örnek modellerin, seçilen çizelgeleme kuralları ile birlikte, Ortalama ara stok düzeyi değişimleri görülmektedir.

Sipariş geliş hızlarının azaltılmasıyla, sistemdeki ortalama ara stok düzeyi (metre cinsinden) beklendiği şekilde azalmıştır.

Ortalama ara stok azalması, her algoritmada eşit oranda olmamasına karşın, kuralların birbirine karşı olan üstünlüklerini ve sıralamalarını değiştirmemiştir.

En iyi sonuç veren yine geliştirilen algoritmalarlardır. Ancak geliş sıklığı azaldıkça, aradaki performans farkı da azalmaktadır. Daha sık gelişlerin olduğu sistemlerde, darboğaz kaynağın etkin kullanımının önemi de arttığı için geliştirilen algoritmaların etkisi de daha fazla olacaktır.

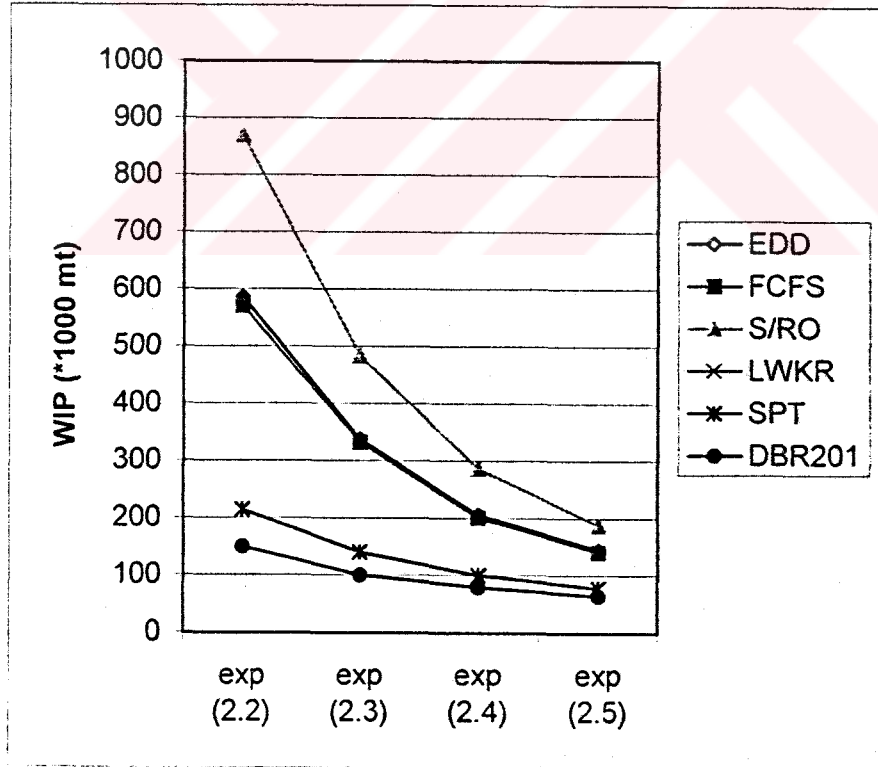
4.2.3. Sipariş Geliş Hızı – Geç Kalan Sipariş Sayısı İlişkisi

Tablo 4.9. ile Şekil 4.5.'te Algoritma 1, ve Tablo 4.10. ile Şekil 4.6.'da Algoritma 2 yöntemlerine ait örnek modellerin, seçilen çizelgeleme kuralları ile birlikte, Geç Kalan Sipariş Sayısı değişimleri görülmektedir.

Tüm kural ve algoritmalarda geç kalan sipariş sayıları azalmasına rağmen, bu azalma aynı oranda gerçekleşmemiştir. S/RO çizelgeleme kuralı bu değişimlerden en fazla etkilenen çizelgeleme kuralı olmuş ve geliş sıklıkları azalırken daha iyi sonuçlar vermiştir.

Tablo 4.7. Sipariř Geliř Hızı-Ortalama Ara Stok İliřkisi (1. Algoritma)

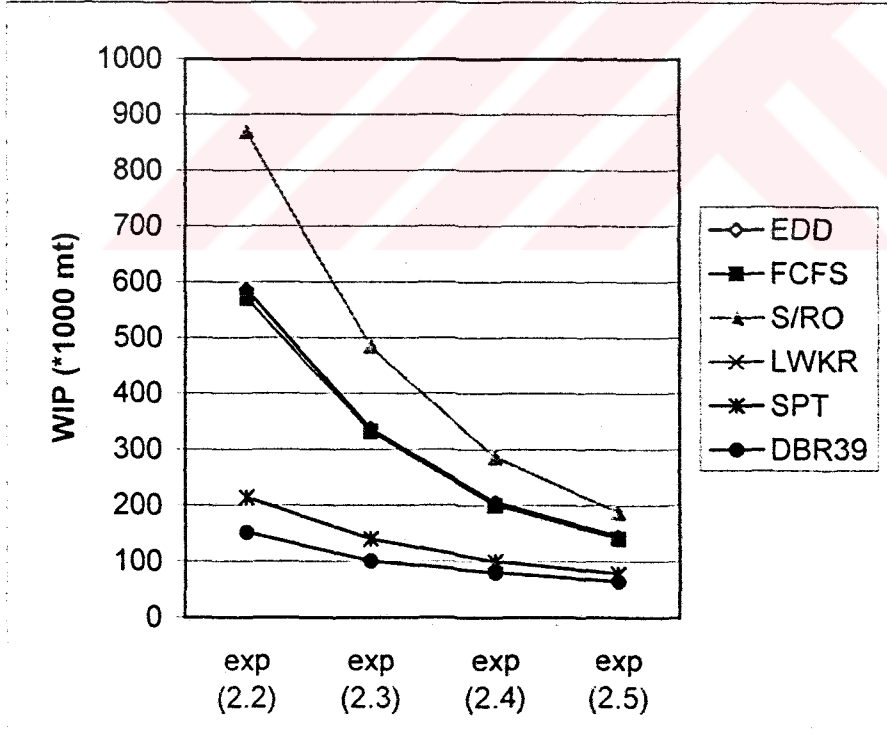
	exp (2.2)	exp (2.3)	exp (2.4)	exp (2.5)
EDD	587,008	337,568	206,100	145,260
FCFS	570,980	332,434	200,948	140,744
S/RO	870,068	483,924	286,666	188,076
LWKR	214,600	139,552	100,181	77,923
SPT	214,812	140,438	100,509	78,703
DBR21	150,434	100,524	80,251	64,796
DBR22	149,928	100,800	80,193	65,053
DBR23	149,872	100,717	80,468	65,042
DBR25	149,926	100,336	80,576	64,605
DBR29	151,778	100,676	80,485	64,572
DBR201	149,786	100,522	79,913	64,726



Şekil 4.3. Sipariř Geliř Hızı-Ortalama Ara Stok İliřkisi (1. Algoritma)

Tablo 4.8. Sipariş Geliş Hızı-Ortalama Ara Stok İlişkisi (2. Algoritma)

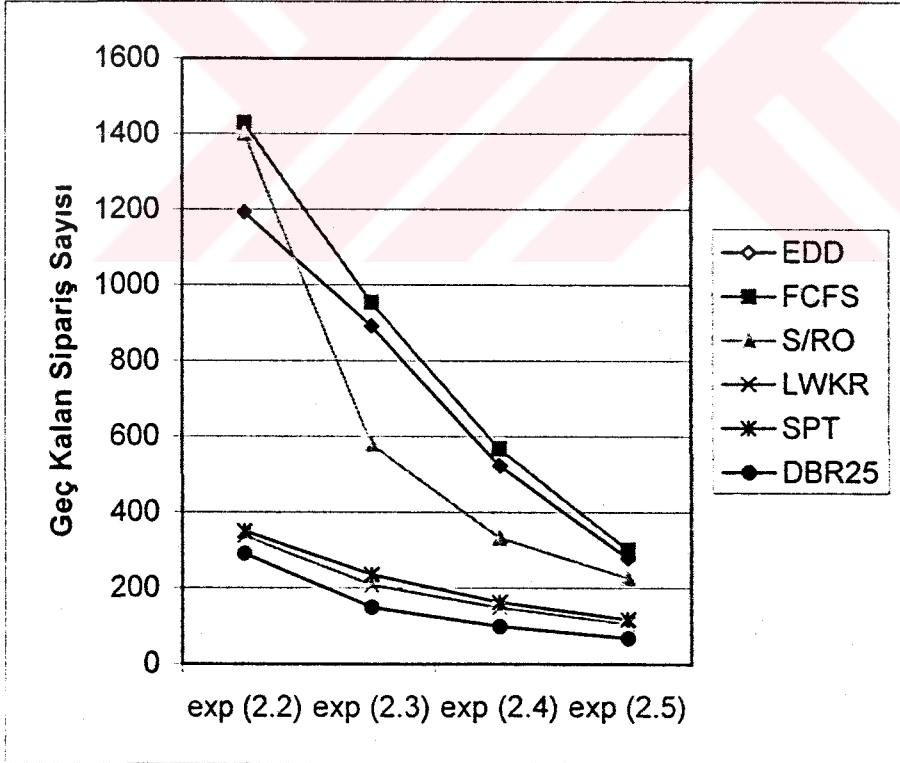
	exp (2.2)	exp (2.3)	exp (2.4)	exp (2.5)
EDD	587,008	337,568	206,100	145,260
FCFS	570,980	332,434	200,948	140,744
S/RO	870,068	483,924	286,666	188,076
LWKR	214,600	139,552	100,181	77,923
SPT	214,812	140,438	100,509	78,703
DBR31	153,024	101,733	80,969	65,381
DBR33	153,972	101,051	81,067	64,768
DBR36	152,058	101,136	80,367	65,179
DBR37	152,404	101,428	80,917	65,181
DBR39	152,174	100,915	80,686	64,887
DBR301	153,048	101,065	80,471	64,999



Şekil 4.4. Sipariş Geliş Hızı-Ortalama Ara Stok İlişkisi (2. Algoritma)

Tablo 4.9. Sipariř Geliř Hızı-Geç Kalan Sipariř Sayısı İliřkisi (1. Algoritma)

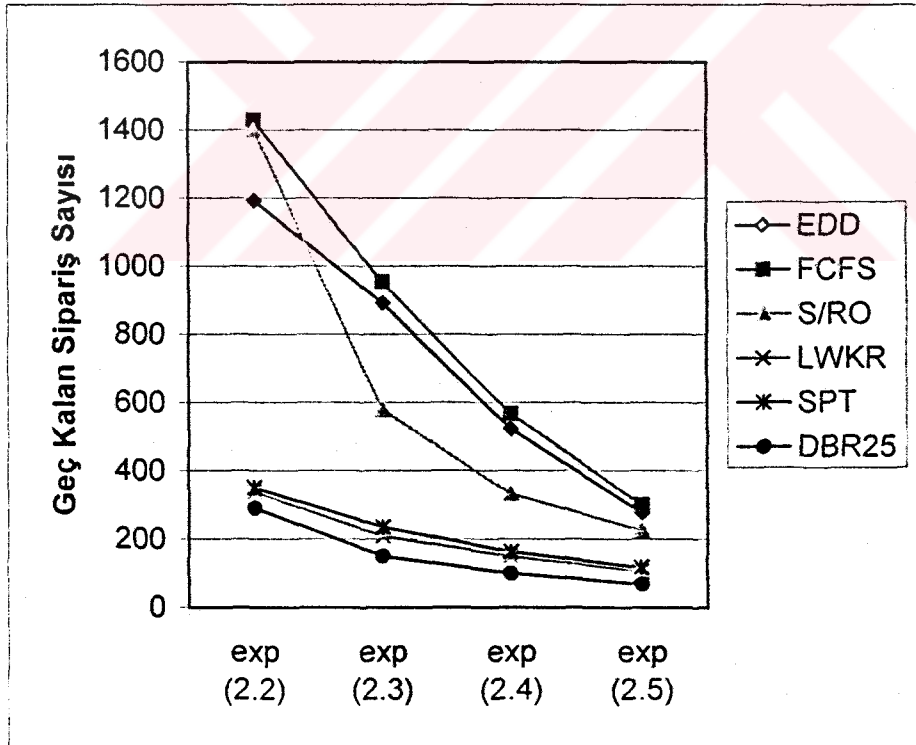
	exp (2.2)	exp (2.3)	exp (2.4)	exp (2.5)
EDD	1193,8	892,4	523,4	278,4
FCFS	1429,0	954,2	567,6	301,2
S/RO	1401,0	579,4	332,8	226,4
LWKR	338,0	208,4	149,0	106,2
SPT	350,4	235,8	163,0	118,0
DBR21	298,8	152,0	100,8	69,8
DBR22	296,2	155,2	100,6	67,0
DBR23	300,8	152,8	103,8	68,6
DBR25	291,4	150,0	99,8	68,8
DBR29	304,4	153,6	102,4	67,0
DBR201	295,4	153,8	102,6	69,0



řekil 4.5. Sipariř Geliř Hızı-Geç Kalan Sipariř Sayısı İliřkisi (1. Algoritma)

Tablo 4.10. Sipariş Geliş Hızı-Geç Kalan Sipariş Sayısı İlişkisi (2. Algoritma)

	exp (2.2)	exp (2.3)	exp (2.4)	exp (2.5)
EDD	1193,8	892,4	523,4	278,4
FCFS	1429,0	954,2	567,6	301,2
S/RO	1401,0	579,4	332,8	226,4
LWKR	338,0	208,4	149,0	106,2
SPT	350,4	235,8	163,0	118,0
DBR21	298,8	152,0	100,8	69,8
DBR22	296,2	155,2	100,6	67,0
DBR23	300,8	152,8	103,8	68,6
DBR25	291,4	150,0	99,8	68,8
DBR29	304,4	153,6	102,4	67,0
DBR201	295,4	153,8	102,6	69,0



Şekil 4.6. Sipariş Geliş Hızı-Geç Kalan Sipariş Sayısı İlişkisi (2. Algoritma)

Geliş sıklıkları azaldıkça geç kalan sipariş sayılarının düşmesi beklenen bir durumdur. Geliş sıklığının, kaynak kullanım değerlerini etkilemeyecek kadar azalması halinde, çizelgeleme kuralları arasında bu performans kriteri açısından fark kalmayacaktır.

Geliştirilen çizelgeleme kuralları tüm geliş sıklıklarında, bu performans kriteri açısından yine en iyi sonuçları vermişlerdir.

4.2.4. Sipariş Geliş Hızı – Pozitif Gecikme İlişkisi

Tablo 4.11. ile Şekil 4.7.'de Algoritma 1, ve Tablo 4.12. ile Şekil 4.8.'de Algoritma 2 yöntemlerine ait örnek modellerin, seçilen çizelgeleme kuralları ile birlikte, Pozitif Gecikme değişimleri görülmektedir.

Geliş sıklıkları azaldıkça Pozitif Gecikme oranının da azalması beklenmektedir. Bu beklenti, S/RO çizelgeleme kuralı haricindeki diğer kural ve algoritmalarda gerçekleşmiştir. S/RO kuralında, exp(2.3) ve exp(2.4) gelişler arası sürelerinde, geciken iş sayısı azalırken, bu işlerin gecikme sürelerinin artması, bu durumun ortaya çıkmasına neden olmuştur.

Geliştirilen çizelgeleme kuralları tüm geliş sıklıklarında, bu performans kriteri açısından yine en iyi sonuçları vermişlerdir. Değişimleri de dengeli olmuştur. Diğer çizelgeleme kurallarında, farklı geliş sıklıklarında beklenmedik oranlarda değişimler gözlenir iken, geliştirilen algoritmaların elde edilen sonuçlarında, değişim kontrollü bir eğilim göstermiştir.

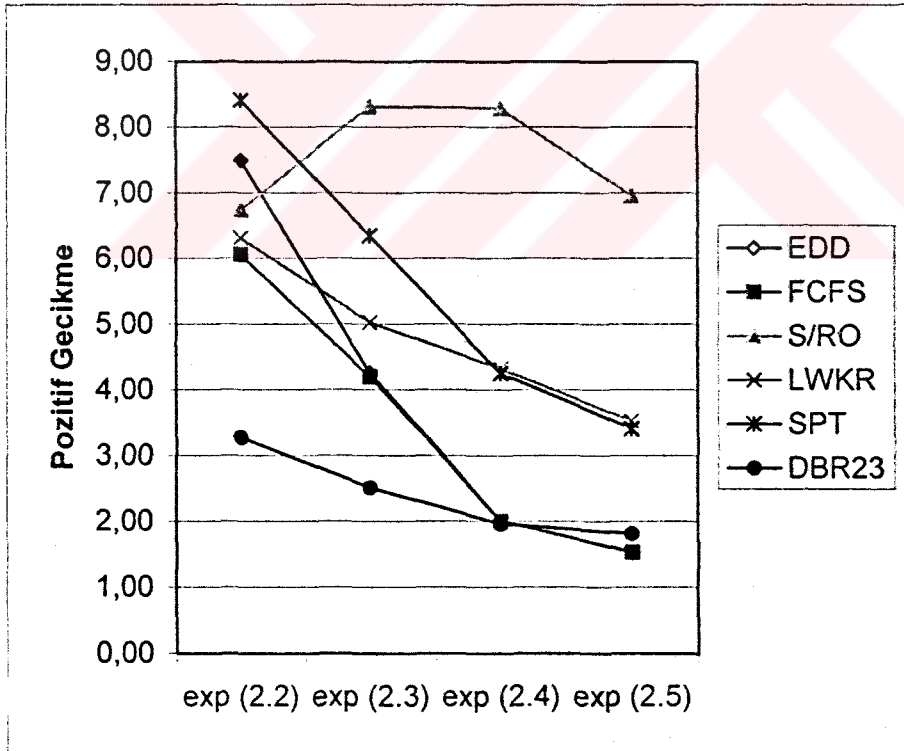
4.2.5. Sipariş Geliş Hızı – Geç Kalma Yüzdesi İlişkisi

Tablo 4.13. ile Şekil 4.9 da Algoritma 1, ve Tablo 4.14. ile Şekil 4.10.'da Algoritma 2 yöntemlerine ait örnek modellerin, seçilen çizelgeleme kuralları ile birlikte, Geç Kalma Yüzdesi değişimleri görülmektedir.

Tüm kural ve algoritmalarda geç kalma yüzdesi azalırken, S/RO çizelgeleme kuralı en fazla değişiklik gösteren çizelgeleme kuralı olmuş ve geliş sıklıkları azalırken daha iyi sonuçlar vermiştir.

Tablo 4.11. Sipariř Geliř Hızı-Pozitif Gecikme İliřkisi (1. Algoritma)

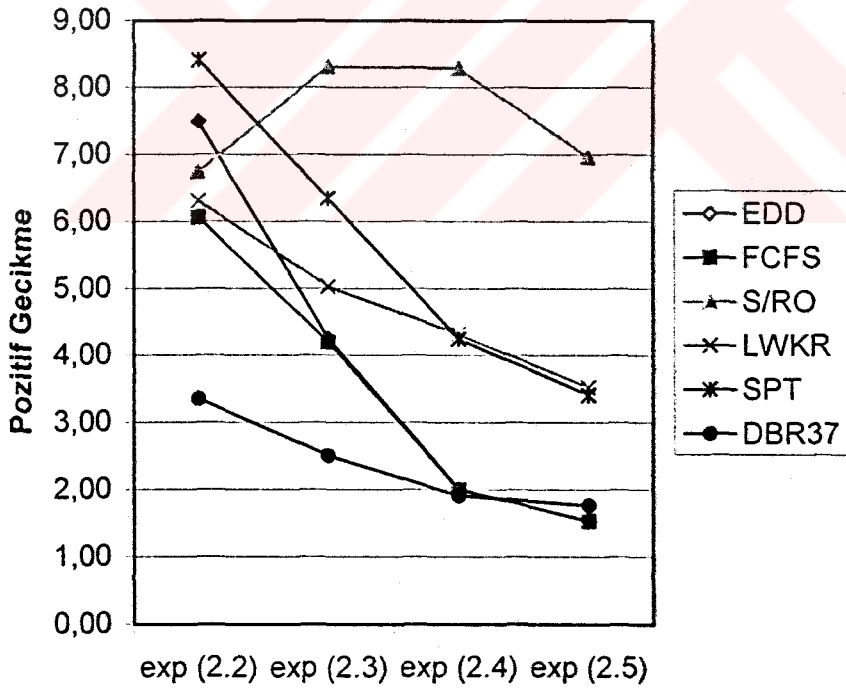
	exp (2.2)	exp (2.3)	exp (2.4)	exp (2.5)
EDD	7,494	4,258	2,013	1,533
FCFS	6,061	4,202	2,006	1,538
S/RO	6,742	8,313	8,294	6,959
LWKR	6,309	5,029	4,325	3,527
SPT	8,410	6,348	4,253	3,414
DBR21	3,357	2,560	2,086	1,843
DBR22	3,400	2,673	2,047	1,873
DBR23	3,279	2,519	1,970	1,821
DBR25	3,391	2,558	2,046	1,834
DBR29	3,324	2,535	2,031	1,852
DBR201	3,388	2,506	2,002	1,811



Şekil 4.7. Sipariř Geliř Hızı-Pozitif Gecikme İliřkisi (1. Algoritma)

Tablo 4.12. Sipariř Geliř Hızı-Pozitif Gecikme İliřkisi (2. Algoritma)

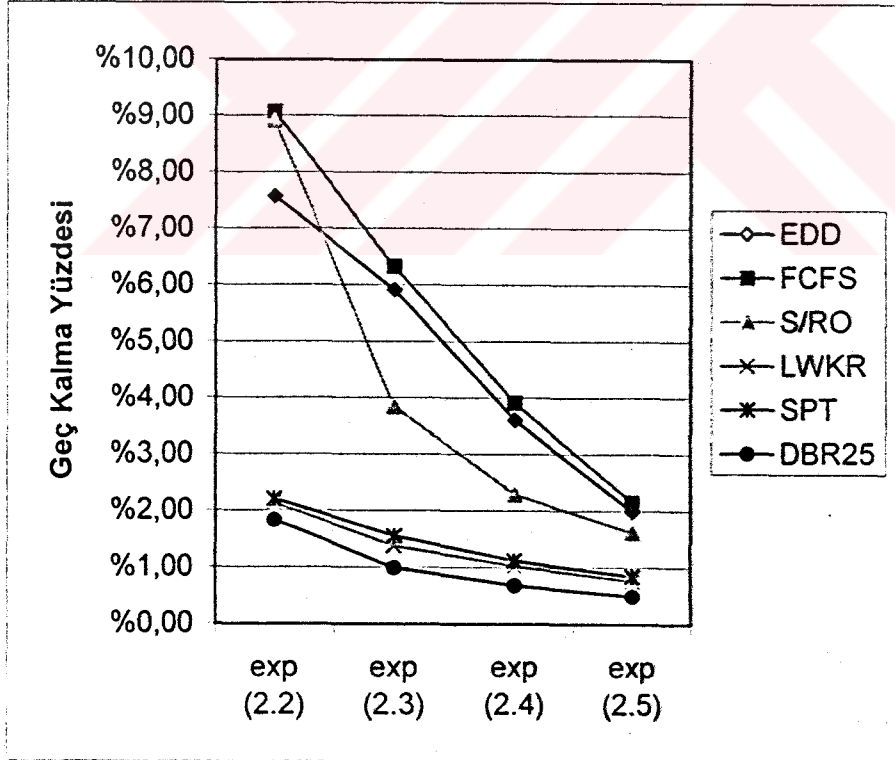
	exp (2.2)	exp (2.3)	exp (2.4)	exp (2.5)
EDD	7,494	4,258	2,013	1,533
FCFS	6,061	4,202	2,006	1,538
S/RO	6,742	8,313	8,294	6,959
LWKR	6,309	5,029	4,325	3,527
SPT	8,410	6,348	4,253	3,414
DBR31	3,321	2,663	2,025	1,799
DBR33	3,320	2,592	2,054	1,784
DBR36	3,374	2,535	1,994	1,855
DBR37	3,357	2,512	1,917	1,768
DBR39	3,352	2,647	2,032	1,857
DBR301	3,295	2,519	2,086	1,835



řekil 4.8. Sipariř Geliř Hızı-Pozitif Gecikme İliřkisi (2. Algoritma)

Tablo 4.13. Sipariř Geliř Hızı-Geç Kalma Yzdesi İliřkisi (1. Algoritma)

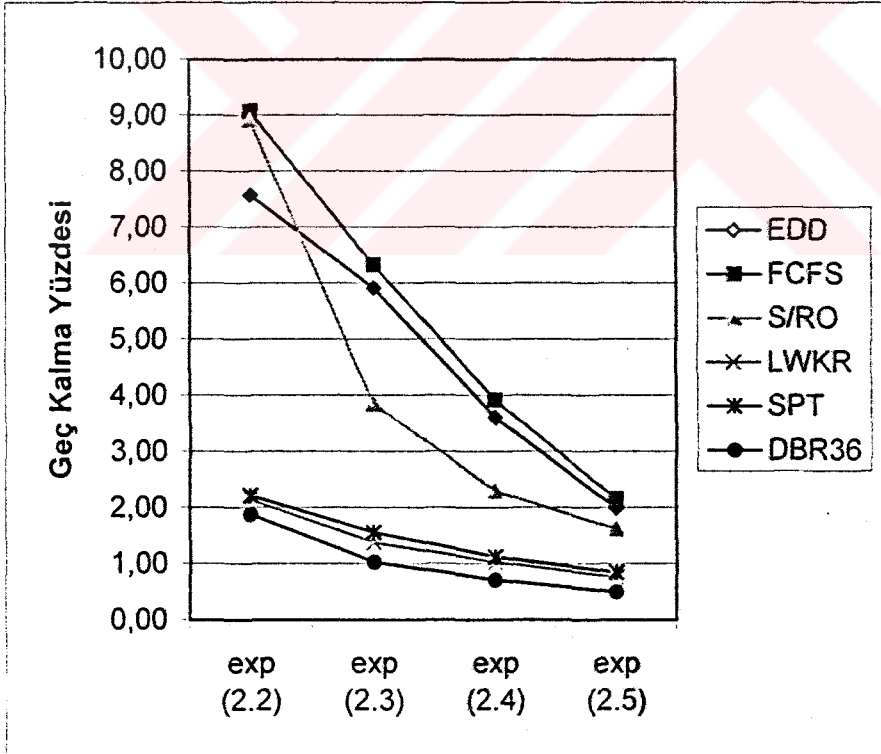
	exp (2.2)	exp (2.3)	exp (2.4)	exp (2.5)
EDD	7,579	5,920	3,610	1,994
FCFS	9,075	6,328	3,916	2,158
S/RO	8,920	3,844	2,299	1,622
LWKR	2,139	1,380	1,026	0,761
SPT	2,217	1,560	1,122	0,845
DBR21	1,880	1,005	0,693	0,500
DBR22	1,872	1,026	0,692	0,480
DBR23	1,900	1,100	0,714	0,491
DBR25	1,841	0,992	0,687	0,492
DBR29	1,923	1,016	0,704	0,480
DBR201	1,866	1,017	0,706	0,494



řekil 4.9. Sipariř Geliř Hızı-Geç Kalma Yzdesi İliřkisi (1. Algoritma)

Tablo 4.14. Sipariř Geliř Hızı-Geç Kalma Yzdesi İliřkisi (2. Algoritma)

	exp (2.2)	exp (2.3)	exp (2.4)	exp (2.5)
EDD	7,579	5,920	3,610	1,994
FCFS	9,075	6,328	3,916	2,158
S/RO	8,920	3,844	2,299	1,622
LWKR	2,139	1,380	1,026	0,761
SPT	2,217	1,560	1,122	0,845
DBR31	1,893	1,017	0,713	0,517
DBR33	1,920	1,030	0,720	0,530
DBR36	1,882	1,030	0,711	0,498
DBR37	1,929	1,075	0,737	0,530
DBR39	1,906	1,022	0,728	0,515
DBR301	1,929	1,063	0,718	0,510



řekil 4.10. Sipariř Geliř Hızı-Geç Kalma Yzdesi İliřkisi (2. Algoritma)

Geliş sıklıkları azaldıkça, geç kalma yüzdesinin de, geç kalan sipariş sayılarında olduğu gibi düşmesi beklenen bir durumdur.

Geliştirilen çizelgeleme kuralları tüm geliş sıklıklarında, bu performans kriteri açısından yine en iyi sonuçları vermişlerdir.

4.2.6. Sipariş Geliş Hızı – Toplam Geç Kalma Süresi İlişkisi

Tablo 4.15. ile Şekil 4.11.'de Algoritma 1, ve Tablo 4.16. ile Şekil 4.12.'de Algoritma 2 yöntemlerine ait örnek modellerin, seçilen çizelgeleme kuralları ile birlikte, Toplam Geç Kalma Süresi değişimleri görülmektedir.

Diğer bağlantılı performans kriterlerinin değişiminde bahsedildiği üzere S/RO haricinde tüm kurallarda ve algoritmalarda, geliş sıklıklarının azalması ile birlikte, Toplam Geç Kalma Sürelerinin de azaldığı görülmektedir.

Geliştirilen algoritmalar yine diğerlerinden daha iyi sonuçlar vermiştir.

4.3. Sipariş Termin Zamanının Etkileri

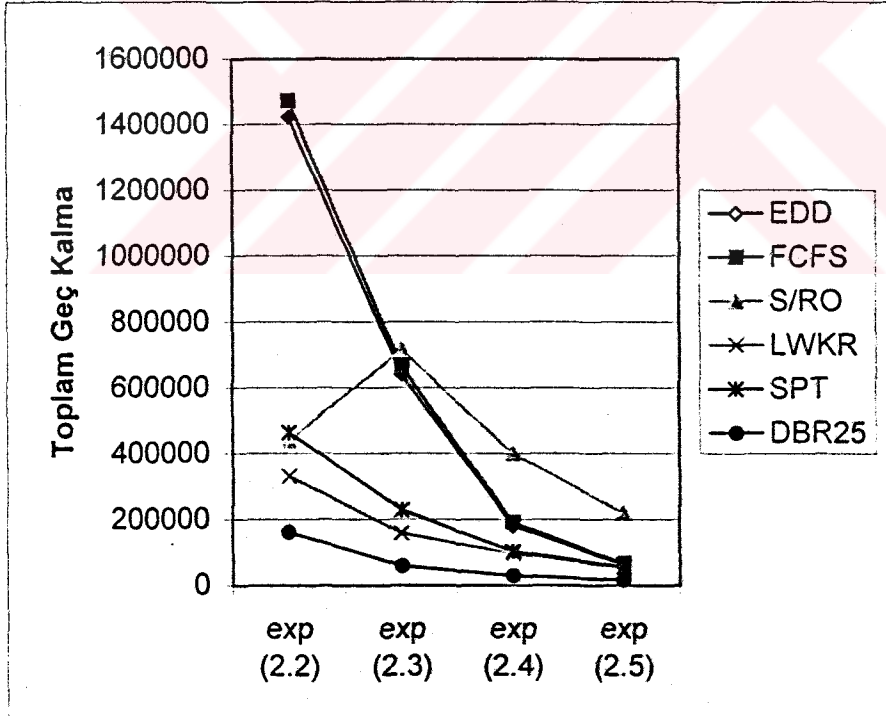
Performans kriterleri üzerinde etkili olabilecek bir başka sistem parametresi de termin zamanlarıdır. Sisteme gelen siparişlerin iç veya dış pazara ait olmalarına göre termin zamanları belirlenmektedir. Sezgisel kurallar ve geliştirilen algoritmaların performanslarını test etmek amacıyla, termin zamanlarının farklı değerlerinde sistem çıktıları incelenebilir.

Mevcut uygulamada, termin zamanı değişimi, sisteme gelen siparişlerin, iç veya dış pazara ait olma durumlarını değiştirerek elde edilebilmektedir. İç ve dış pazar için belirlenen termin zamanları arasında büyük bir fark olmadığından, bu oranlar için performans değerlerinin yüksek oranda bir farklılık göstermesi beklenmemektedir. Ancak, sezgisel algoritmalar ile, geliştirilen algoritmaların farklılığını göstermek amacıyla üç farklı iç ve dış pazar oranı, yani farklı termin zamanları ile model çalıştırılmıştır.

Termin zamanındaki değişimleri test ederken, karşılaştırma yapılacak sezgisel kuralların seçiminde, termin zamanına göre çizelgelemeyi yapan EDD kuralı ile,

Tablo 4.15. Sipariř Geliř Hızı-Toplam Geç Kalma Sresi İliřkisi (1. Algoritma)

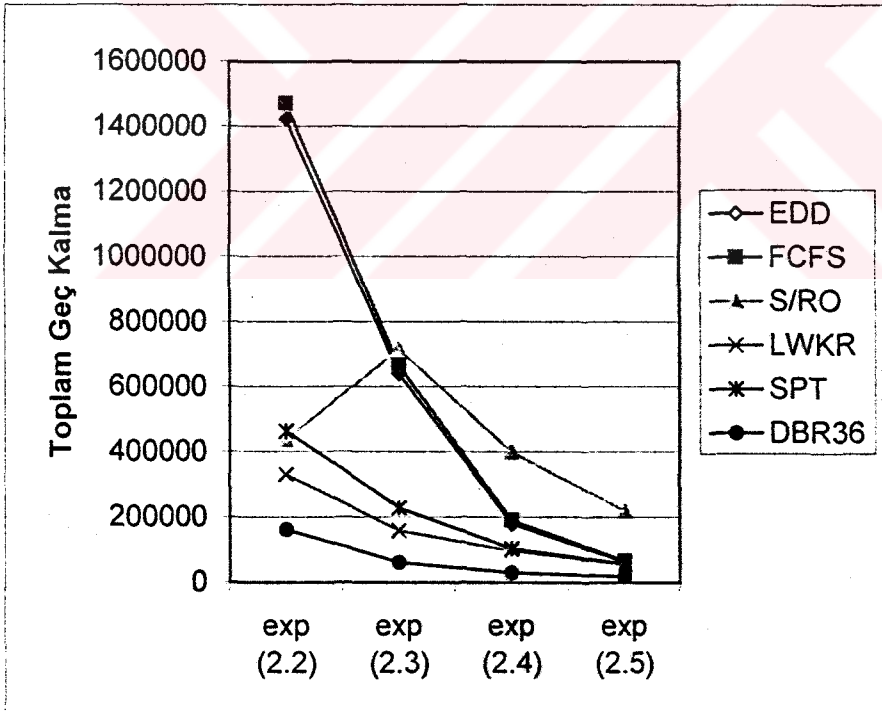
	exp (2.2)	exp (2.3)	exp (2.4)	exp (2.5)
EDD	1422801	640755	181167	63043
FCFS	1472755	667143	192046	67298
S/RO	438807	716031	400250	220327
LWKR	331730	159459	96414	52991
SPT	463962	229641	103939	57004
DBR21	159933	59689	30957	18058
DBR22	162655	61227	30514	17778
DBR23	163017	59739	30359	17666
DBR25	162514	59701	30121	17760
DBR29	160303	59658	30702	17408
DBR201	162169	59671	30212	17617



Şekil 4.11. Sipariř Geliř Hızı-Toplam Geç Kalma Sresi İliřkisi (1. Algoritma)

Tablo 4.16. Sipariř Geliř Hızı-Toplam Geç Kalma Süresi İliřkisi (2. Algoritma)

	exp (2.2)	exp (2.3)	exp (2.4)	exp (2.5)
EDD	1422801	640755	181167	63043
FCFS	1472755	667143	192046	67298
S/RO	438807	716031	400250	220327
LWKR	331730	159459	96414	52991
SPT	463962	229641	103939	57004
DBR31	160735	63172	30778	18333
DBR33	162687	62218	31738	18564
DBR36	162600	60864	30241	18099
DBR37	165739	62557	30362	18318
DBR39	163549	62289	31757	18548
DBR301	161933	61946	32133	18218



Şekil 4.12. Sipariř Geliř Hızı-Toplam Geç Kalma Süresi İliřkisi (2. Algoritma)

hemen tüm performans ölçütlerinde en iyi sonuçları veren LWKR kuralları seçilmiştir. Geliştirilen her iki algoritmadan da ikişer farklı model için karşılaştırma tabloları oluşturulmuştur.

Tablo 4.17.'den görülebileceği gibi Ortalama Akış Zamanlarında, termin zamanlarının etkisi olmamıştır. Termin zamanının kısalması ya da uzaması, işlem süresini etkilememekte, sadece bazı çizelgeleme kurallarında önceliklerin değişmesine neden olmaktadır. Bunun sistem performansına etkisi ise yok denecek kadar azdır.

Tablo 4.18., termin zamanı değişimlerine bağlı olarak Ortalama ara stok düzeyi değişimini göstermektedir. Farklılık yok denecek kadar azdır. Termin zamanı değişimi, özellikle EDD çizelgeleme kuralını etkilediği için, bir miktar değişim olmuş, ancak bu da dikkate değer bulunmamıştır.

Tablo 4.19. ve Şekil 4.13.'de Geç Kalan Sipariş Sayısının, termin zamanlarının değişimine bağlı olarak değişimleri gösterilmiştir. Termin zamanlarının değişimi ile beklendiği gibi geç kalan sipariş sayısında da değişme olmuştur. Dış pazar oranının artırılması ile termin süreleri kısaldığı için, geç kalan sipariş sayısı artmaktadır. Tüm çizelgeleme kuralları ve geliştirilen algoritmalarda paralel sonuçlar gözlenmiştir.

Tablo 4.20.'den görülebileceği gibi, Pozitif Gecikme ilişkisi termin zamanlarının değişimine bağlı olarak etkilenecek göstermiştir. Dış ve iç pazar oranı ve buna bağlı termin değişimi gecikmeleri etkilemiş, bu da pozitif gecikme değeri üzerinde etkili olmuştur.

Tablo 4.21. ve Şekil 4.14.'den da görülebileceği gibi, Geç Kalma Yüzdesi, daha kısa termin zamanına sahip olan dış pazara ait ürünlerin oranının artmasıyla, tüm sezgisel çizelgeleme algoritmalarında ve geliştirilen algoritmalarda neredeyse paralel bir artışa sebep olmuştur, bu da beklenen bir durumdur.

Tablo 4.22.'de, Toplam Geç Kalma Sürelerine ait değişimin, dış pazara ait ürünlerin oranının artırılmasıyla paralel olarak artış gösterdiği görülmektedir. Diğer geç kalma ölçütlerinde olduğu gibi, bu ölçütteki artışın da sebebi, termin tarihi kısa olan işlerin, sisteme giren toplam siparişlerin büyük oranını teşkil etmesinden kaynaklanmaktadır.

Tablo 4.17.Termin Zamanı-Ortalama Akıř Zamanı İliřkisi (1. ve 2. Algoritma)

	30/70	50/50	70/30
EDD	284,03	283,347	283,463
LWKR	127,49	127,495	127,495
DBR21	136,91	136,915	136,915
DBR25	136,078	136,082	136,082
DBR31	136,808	136,812	136,812
DBR36	136,138	136,261	136,261

Tablo 4.18.Termin Zamanı-Ortalama Ara Stok İliřkisi (1. ve 2. Algoritma)

	30/70	50/50	70/30
EDD	587,008	583,927	584,785
LWKR	214,6	214,606	214,605
DBR21	150,434	150,44	150,44
DBR25	149,926	149,932	149,932
DBR31	153,024	153,028	153,082
DBR36	152,058	152,062	152,062

Tablo 4.20.Termin Zamanı-Pozitif Gecikme İliřkisi (1. ve 2. Algoritma)

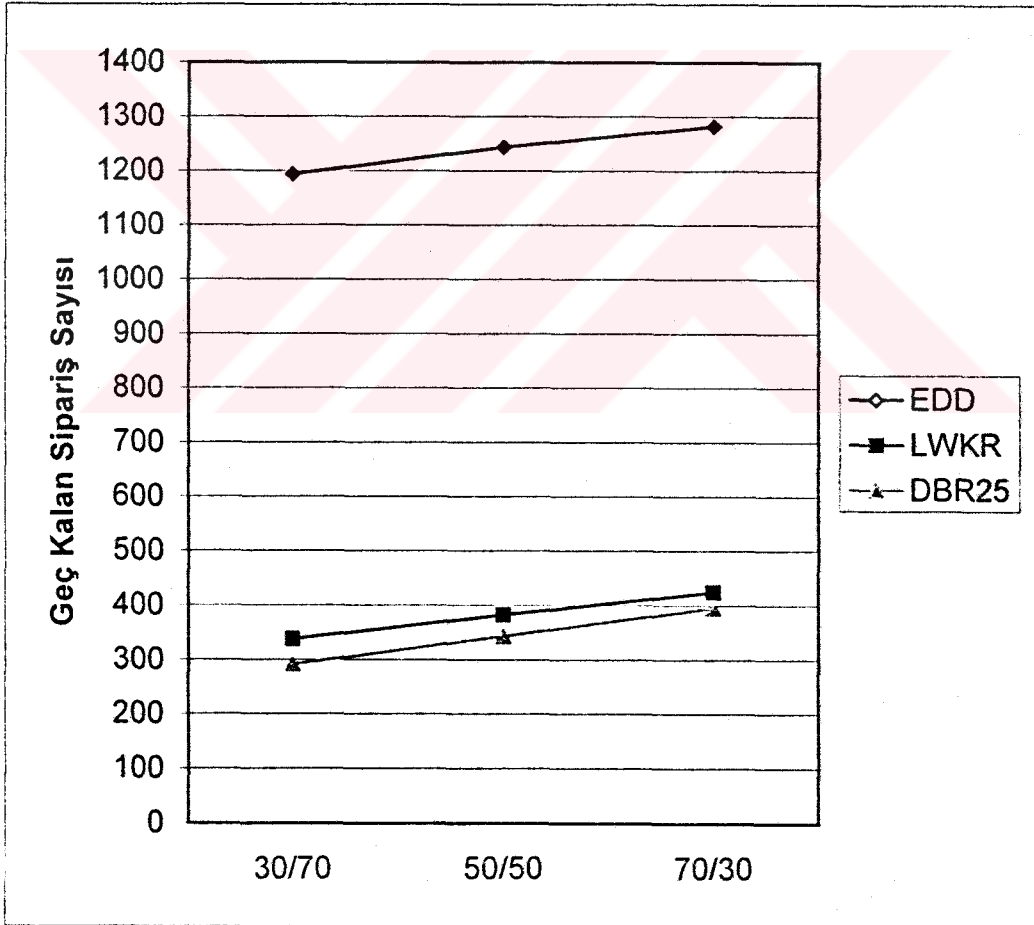
	30/70	50/50	70/30
EDD	18,07	18,65	19,28
LWKR	4,198	4,36	4,53
DBR21	2,056	2,199	2,34
DBR25	2,02	2,16	2,29
DBR31	2,032	2,177	2,32
DBR36	2,055	2,198	2,33

Tablo 4.22.Termin Zamanı-Toplam Geç Kalma Sresi İliřkisi (1. ve 2. Algoritma)

	30/70	50/50	70/30
EDD	1422801	1468650	1518407
LWKR	331730	345044	358196
DBR21	159933	174029	185280
DBR25	162514	171119	181822
DBR31	160735	172313	183856
DBR36	162600	173943	185094

Tablo 4.19.Termin Zamanı-Geç Kalan Sipariř Sayısı İliřkisi (1. ve 2. Algoritma)

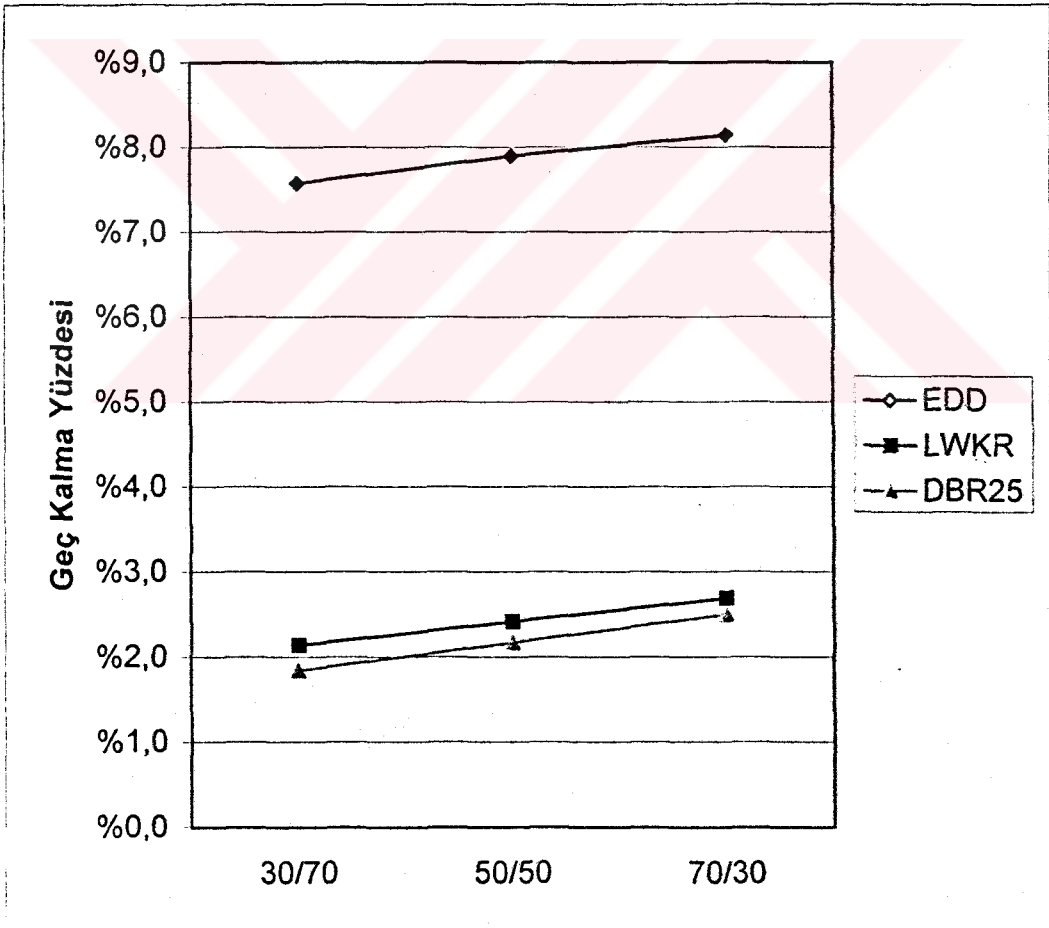
	30/70	50/50	70/30
EDD	1193,8	1243,8	1282,4
LWKR	338	383,2	425,2
DBR21	298,8	349,4	406,4
DBR25	291,4	343,8	395,8
DBR31	299,6	351	405
DBR36	297,8	352,6	404,4



řekil 4.13.Termin Zamanı-Geç Kalan Sipariř Sayısı İliřkisi (1. ve 2. Algoritma)

Tablo 4.21.Termin Zamanı-Geç Kalma Yzdesi İlişkisi (1. ve 2. Algoritma)

	30/70	50/50	70/30
EDD	7,579	7,897	8,1438
LWKR	2,1389	2,42	2,69
DBR21	1,88	2,208	2,56
DBR25	1,841	2,172	2,501
DBR31	1,893	2,218	2,55
DBR36	1,882	2,228	2,55



Şekil 4.14.Termin Zamanı-Geç Kalma Yzdesi İlişkisi (1. ve 2. Algoritma)

5.SONUÇLAR VE ÖNERİLER**5.1. Çalışmanın Özetlenmesi**

Üretim çizelgelemeyi daha iyi yapabilmek, oldukça uzun süredir iş ve bilim dünyasının üzerinde çalıştığı bir konudur. Pek çok matematiksel çalışma ile işleri en iyi şekilde sıralamanın yolları aranmış, pek çok farklı yöntem ve yaklaşım geliştirilmiştir. Bilgi sistemlerinin hızla gelişmesi, çizelgeleme alanında da kendini göstermiş ve özellikle de simülasyon esaslı çizelgeleme yöntemleri, optimum olmasa da iyi sonuçlar veren çizelgeler oluşturulmasını sağlamıştır.

Üretim çizelgelemede önemli olan, üretim ortamının özelliğine uygun olarak, öncelikli görülen amaçların optimize edilmesini sağlayıcı sistemi geliştirmektir. Gözetilecek amaçlar çoğunlukla, müşteri tatminini sağlayacak olan, termin zamanlarının sağlanması, üretim kaynaklarının etkin kullanımı ve ara stok düzeylerinin ve akış zamanlarının azaltılmasıdır.

Literatürde çizelgeleme konusunda oldukça fazla sayıda çalışma bulunmaktadır. Bununla beraber, sipariş tipi üretim ortamlarında, üretim ortamın ve siparişlerin belirsizlik özelliğinden kaynaklanan, darboğaz kaynağın değiştiği durumlar için yapılmış çalışmalar oldukça az sayıdadır. Gerçek sistemlerde bu tip durumlarda darboğaz haline gelen kaynaklar için ya geçici önlemlerle ya da kalıcı kapasite artırımlarıyla çözüm bulunmaya çalışılmaktadır. Darboğaz kaynağın tek ve baskın olduğu durumlarda, Kısıtlar Teorisi temelli OPT yönteminin DBR tekniği, darboğaz kaynağı çizelgelemede iyi sonuçlar vermektedir. Ancak darboğaz kaynağın birden fazla olduğu ve değiştiği durumlarda, bu yöntemler etkisiz kalmaktadır. Bu çalışmada, değişken ve çoklu darboğazlı sistemlerde sistemi dinamik olarak kontrol eden ve darboğaz kaynaklara odaklanarak sistemin kalanını da çizelgeleyen yeni bir yaklaşım geliştirilmiştir.

Ele alınan örnek sistem, çoklu sipariş ve çoklu makineli bir sipariş tipi üretim ortamıdır. Alınan siparişlerin rotaları, işlem zamanları, termin tarihleri gibi özellikleri belirlidir. Çalışmada, akış zamanı, sistemin ara stok düzeyi, geciken

sipariş sayısı, gecikme süresi, pozitif gecikme, yüzde gecikme oranı, performans değişkenleri olarak alınmışlardır.

Çalışmada, sayılan özelliklere sahip bir üretim ortamında DBR tekniğinin mantığına dayanan ve onu geliştiren dinamik bir çizelgeleme tekniği geliştirilmiştir. Değişken darboğazlı sistemde optimum çizelgelemeyi gerçekleştirmeyi amaçlayan çalışmada, aşağıdaki adımlar uygulanmaktadır:

1. Ortamın ve siparişlerin özelliğine göre darboğaz olan iş merkezlerinin belirlenmesi,
2. Belirlenen darboğaz iş merkezleri için zaman stoğu kurulması,
3. Darboğaz önündeki zaman stoğunun değişimine göre sistemin kalanında, gerekli işlere öncelik verilmesi,
 - 3.1. Darboğaz kaynakların iş yükünü dengelemeye yönelik olarak, uygun işlem zamanları için öncelikli olması,
 - 3.2. Darboğaz kaynakların iş yüklerini dengelemek için, işlerin darboğaz kaynağa ulaşma hızlarına bağlı olarak önceliklerinin olması,
4. Belirlenen önceliklerin, siparişlerin rotalarında, darboğazdan önce gelen tüm iş merkezlerinde geçerli olması ve darboğazdan sonra gelen iş merkezlerinde ise, SPT kuralının uygulanması.

5.2. Sonuçlar

Geliştirilen algoritmalar, seçilen dokuz sezgisel çizelgeleme kuralı ile karşılaştırılmıştır (FCFS, EDD, LWKR, SPT, STR, MOPNR, MWKR, S/RO, SD). Sistemin mevcut durumu için geliştirilen yöntemler ve sezgisel çizelgeleme kuralları ile denemeler yapılmış ve belirlenen performans kriterlerinde gösterdikleri etkinlikler karşılaştırılmıştır (Tablo 4.1., Tablo 4.3., Tablo 4.4.). Seçilen dağılım kuralları çoğunluğu dinamik olan, oldukça iyi sonuçlar veren ve uygulamada gerçek sistemlerde kullanılan sezgisel kurallardır.

Geliştirilen algoritmalar, özellikle de ara stok düzeylerinin azaltılması ve gecikmeler ile ilgili kriterlerde oldukça iyi sonuçlar vermişlerdir.

- Sistemin ara stok düzeyi, EDD kuralına göre %75 ve LWKR kuralına göre de %31 daha azdır.
- Geç kalan sipariş sayısı, SPT kuralına göre %15.6 daha azken, EDD kuralına göre bu oran %75 düzeyindedir.
- Ortalama pozitif gecikmede iyileşme EDD için %56.5 iken, LWKR için de %48.3 düzeyindedir.
- Geciken işlerin oranı, geliştirilen algoritmalarda EDD için %75.7, SPT için %17 ve FCFS kuralı için ise, %80 daha azdır.
- Toplam geç kalma süreleri için de benzer sonuçlar alınmıştır. LWKR kuralına göre %81 oranında, EDD kuralına göre %96 oranında iyileştirme sağlanmıştır.

Tüm bu performans değişkenlerinin iyileştirilmesi sırasında siparişlerin ortalama akış zamanları için, sezgisel çizelgeleme kuralları arasında en iyi sonuç veren kurallarla başabaş sonuçlar elde edilirken, sistemin çıktı düzeyinde de herhangi bir azalmaya sebep olmamakta, aksine, tüm kaynakların etkin kullanımının sağlanmasıyla çıktı düzeyinde bir artış sağlanmakta ve tamamlanan sipariş sayısının arttığı görülmektedir.

Değişen durumlar için, örneğin sipariş geliş hızlarının değişmesi veya termin zamanlarının farklı olması gibi, geliştirilen algoritmalar ve sezgisel çizelgeleme kurallarının etkinlikleri yeniden değerlendirilmiş, farklı durumlarda sezgisel kuralların farklı davranabildiği görülmüştür. Geliştirilen algoritmalar denenerek tüm bu durumlarda, en iyi sonuçları verdikleri belirlenmiştir.

Bu sonuçların alınmasının nedeni, geliştirilen algoritmaların sisteme dinamik olarak bakması, üretim ortamında, üretimi kısıtlayan ve tüm sistemi etkileyen darboğaz kaynaklara odaklanarak çizelgeleme yapması, ara stok düzeylerini kontrol etmesi, ve tüm sistemde geçerli önceliklerin dinamik olarak değişen durumlara göre belirlenmesidir.

5.3. Sonraki Çalışmalar İçin Öneriler

Yapılan çalışmada geliştirilen algoritmaların daha da iyileştirilebilecekleri düşünülmektedir. Bu çalışma, algoritmanın çalışma mantığının gösterilmesi açısından, değişken darboğazın iki iş merkezi arasında olması durumunu dikkate almaktadır. Çizelgeleme mantığının daha fazla iş merkezine de uygulanması mümkündür. Böyle bir durumda, darboğaz olma olasılığı bulunan iş merkezleri için zaman stokları kullanılacak, simülasyon modellerinde uygun kontroller eklenerek, dinamik olarak, gelen siparişin özelliğine göre darboğaz haline gelecek kaynaklara odaklanan çizelgeleme gerçekleştirilmiş olacaktır. Bu çizelgeleme mantığının kullanılmasıyla, ürün karmasının değişimi, talep dalgalanmaları gibi olaylarda sistem dinamik olarak, kendini kontrol edecek ve etkin olmaya devam edecektir.

Çalışmada, kurulan zaman stoğunun düzeyinin daha etkin çalışacak şekilde belirlenmesi, gerçekleştirilecek bir başka öneridir. Stok düzeylerinin yönetimi ve kontrolü, ayrıca incelenebilecek bir konu olduğundan, bu çalışmada değinilmemiştir. Ancak üretim ortamının çıktı düzeyini azaltmayacak şekilde, darboğaz kaynakların daha etkin kullanılmasını sağlayacak zaman stoğu büyüklüğü belirlenmesi de olasıdır.

Darboğaz iş merkezinden sonra gelen kaynaklarda, bu çalışmada, işlerin daha çabuk ilerlemesini sağlayan SPT kuralı uygulanmıştır. EDD veya uygun görülecek diğer sezgisel kuralların denenmesi, hatta duruma göre bunların da dinamik olarak değiştirilmeleri gerçekleştirilebilir.

Son olarak, her simülasyon çalışmasında yapıldığı gibi, bu üretim ortamı için de bazı varsayımlarda bulunulduğu hatırlanmalıdır. Bu varsayımların dışında, makine bozulmaları, taşıma zamanları, siparişlerin iptal olması veya tekrar işlenmek üzere, daha önce uğradığı iş merkezlerine geri dönmesi gibi durumlar da modele eklenebilir.

KAYNAKLAR

- ACAR, N. 1989. Üretim Yönetimi Yöntem ve Uygulamaları. Milli Prodüktivite Merkezi Yayınları:280, Ankara, 209s.
- ASKIN, R.G., STANDRIDGE, C.R. 1993. Modeling And Analysis Of Manufacturing Systems. John Wiley And Sons Inc., NY, 461p.
- BAKER, K.R., 1974. Introduction To Sequencing And Scheduling, John Wiley And Sons, Inc., NY., 305 p.
- BASSETT, G., TODD,R., 1994. The SPT Priority Sequence Rule. International Journal Of Operations And Production Management, 14(12): 70-78.
- BRUCKER, P., 1995. Scheduling Algorithms. Springer-Verlag, Berlin, 327 p.
- CHASE, R.B., AQUILANO, N.J., JACOBS, F.R., 2001. Operations Management For Competitive Advantage. McGraw Hill/Irwin, Ninth Edition, NY, 763 p.
- COOK, D.P.,1994. A Simulation Comparison Of Traditional, JIT, And TOC Manufacturing Systems In A Flow Shop With Bottlenecks. Production And Inventory Management Journal, First Quarter:73-78.
- DILWORTH, J.B., 1996. Operations Management. Mc-Graw Hill, NY., 758p.
- FAWCET, S.E., PEARSON, J.N., 1991. Understanding And Applying Constraint Management In Today's Manufacturing Environments. Production And Inventory Management, Third Quarter:46-55.
- FRENCH, S.,1982. Sequencing And Scheduling, An Introduction To The Mathematics of The Job Shop. Ellis Horwood Ltd., England, 245 p.

GOLDRATT, E.M., FOX, R.E., 1986. The Race. North River Press, NY, 180 p.

GROSFELD-NIR A., 1995. Single Bottleneck Systems With Proportional Expected Yields And Rigid Demand. European Journal of Operational Research, 80(2):297-307.

GUNN, B., NAHAVANDI, S., 2000. Determining The Optimum Level Of Work In Progress Using Constraint Analysis And Computer Simulation. Assembly Automation, 20 (4) :305-312.

HALAÇ, O., 1982. İşletmelerde Simülasyon Teknikleri. İstanbul Üniversitesi Yayınları:2936, İstanbul, 278 s.

HURLEY, S.F., KADIPASAOĞLU, S., 1998. Wandering Bottlenecks: Speculating On The True Causes. Production And Inventory Management Journal, Fourth Quarter: 1-4.

KAN, A.H.G.R., 1976. Machine Scheduling Problems. The Hague, 180 p.

LAW, A.M., McCOMAS, M.G., 1999. Simulation Of Manufacturing Systems. Proceedings of The 1999 Winter Simulation Conference: 56-59.

LAWRENCE S.R., BUSS A.H., 1994. Shifting Production Bottlenecks: Causes, Cures And Conundrums. Production And Operations Management, 3 (1): 21-37.

MELNYK, S.A., VICKERY, S.K., CARTER, P.L., 1986. Scheduling, Sequencing And Dispatching: Alternative Perspectives. Production And Inventory Management, Second Quarter: 58-67.

- MORTON, T.E., PENTICO, D.W., 1993. Heuristic Scheduling Systems, With Applications To Production Systems And Project Management. Wiley Series, NY., 695p.
- MOSS, H.K., WEN BIN YU, 1999. Toward The Estimation of Bottleneck Shiftiness In A Manufacturing Operation. Production And Inventory Management, Second Quarter: 53-58.
- NAS, S., 1998. Çok Aşamalı Sipariş Tipi Üretim Sistemleri İçin Yeni Bir Dinamik Çizelgeleme Yaklaşımı. Ç.Ü. Yüksek Lisans Tezi, Adana, 112s. (yayınlanmamış)
- NEELY, A.D., BYRNE, M.D., 1992. A Simulation Study Of Bottleneck Scheduling. International Journal of Production Economics, 26(1-3): 187-192.
- PANWALKAR, S.S., ISKANDER, W., 1977. A Survey of Scheduling Rules. Operations Research, 25 (1): 45-61.
- PORTER, K., LITTLE, D., PECK, M., ROLLINS, R., 1999. Manufacturing Classifications: Relationships With Production Control Systems. Integrated Manufacturing Systems 10(4): 189-198.
- RODRIGUES L.H., MACKNESS J.R., 1998. Teaching The Meaning Of Manufacturing Synchronisation Using Simple Simulation Models. International Journal Of Operations And Production Management, 18(3): 246-259.
- SCHRAGENHEIM, E., RONEN, B., 1990. Drum-Buffer-Rope Floor Control. Production And Inventory Management Journal, Third Quarter: 18-22.

SHANNON, R.E., 1998. Introduction To The Art And Science Of Simulation. Proceedings Of The 1998 Winter Simulation Conference: 7-14.

SIPPER, D., BULFIN, R.L., 1997. Production, Planning, Control And Integration. Mc-Graw Hill, NY., 630 p.

SPENCER, M.S., 1991. Using 'The Goal' In An MRP System. Production And Inventory Management Journal, Fourth Quarter: 22-27.

Systems Modeling Corporation, 1989. The SIMAN IV Reference Guide. System Modeling And Corp., Pennsylvania, 426 p.

UMBLE M., SRIKANTH, M., 1995. Principles For World-Class Excellence: Synchronous Manufacturing. The Spectrum Publishing Company, Inc., Connecticut, 269p.

VAN WASSENHOVE L.N., VANDERHENST, P., 1983. Planning Production In A Bottleneck Department. European Journal of Operational Research, 12: 127-137.

YILMAZ, M. Üretim Sistemleri. Murat Yılmaz İTÜ Endüstri Mühendisliği Sayfası. <http://www.mylmz.cjb.net>.

ZAPFEL, G., MISSBAUER, H., 1993. New Concepts For Production Planning And Control. European Journal of Operational Research, 67(3): 297-320.

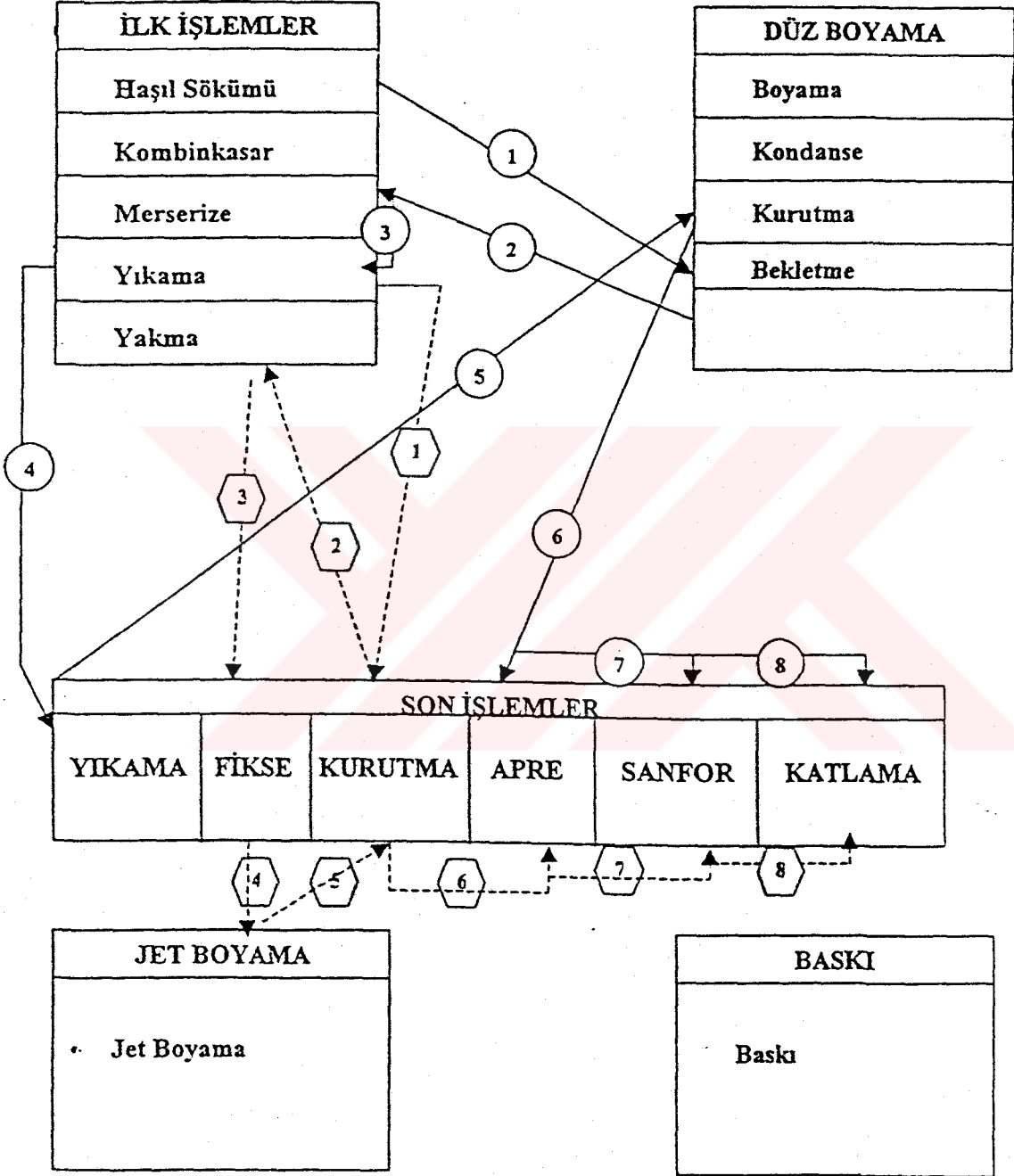
ÖZGEÇMİŞ

1976 yılı Mersin doğumlu. İlk ve orta eğitimini Mersin’de tamamladı. 1998 yılında Çukurova Üniversitesi Endüstri Mühendisliği bölümünden mezun oldu. Aynı yıl Çukurova Üniversitesi Endüstri Mühendisliği Anabilim Dalı’nda Yüksek Lisans programına başladı. Halen Çukurova Üniversitesi İktisadi ve İdari Bilimler Fakültesi İşletme Bölümü’nde Araştırma Görevlisi olarak çalışmaktadır.



EKLER

EK 1 SİPARİŞ AKIŞ ŞEMASI



(Kaynak: NAS, S., 1998)

EK 2 SEZGİSEL ÇİZELGELEME KURALLARI

EDD kuralına ait simülasyon modeli

BEGIN;

CREATE:exponential(2.5,1):MARK(TimeIn);
COUNT:ENTNUM;

ASSIGN:NS=discrete(0.18,1,0.34,2,0.57,3,0.65,4,
0.74,5,0.80,6,1.0,7):
SipBuy=discrete(0.2,2500,0.6,4000,0.85,6000,0.95,10000,
1.0,25000);

BRANCH,1:WITH,0.3,export:
WITH,0.7,yerli;

export ASSIGN:DueDate=TNOW+UNIF(336,504):NEXT(PA1);
yerli ASSIGN:DueDate=TNOW+UNIF(504,672);

PA1 branch, 1:
if,NS==1,PA2:
if,NS==2,PA2:
if,NS==3,PA2:
if,NS==4,PA3:
if,NS==5,PA2:
if,NS==6,PA3:
else, PA3;

PA2 assign:renklibeyaz=1;
PA21 assign:renktonu=discrete(0.3,1,0.6,2,1.0,3):NEXT(PA4);

PA3 assign:renklibeyaz=2;
PA31 assign:renktonu=1;

PA4 ROUTE;

STATION,HASILSOKUMU;
QUEUE,HASILSOKUMUQ;
SEIZE:HASILSOKUMUR;
ASSIGN:TIMEHASILSOKUMU=OptTime*SipBuy;
DELAY:TIMEHASILSOKUMU;
RELEASE:HASILSOKUMUR;
ROUTE;
STATION,BEKLETME;
QUEUE,BEKLETMEQ;
SEIZE:BEKLETMER;
ASSIGN:TIMEBEKLETME=OptTime;
DELAY:TIMEBEKLETME;
RELEASE:BEKLETMER;
ROUTE;

STATION,KOMBINKASAR;
QUEUE,KOMBINKASARQ;
SEIZE:KOMBINKASARR;
ASSIGN:TIMEKOMBINKASAR=OptTime*SipBuy;
DELAY:TIMEKOMBINKASAR;
RELEASE:KOMBINKASARR;
ROUTE;

STATION,MERSERIZE;
QUEUE,MERSERIZEQ;
SEIZE:MERSERIZER;

assign:acttypemerserize=renklibeyaz;

PA5 ASSIGN:TIMEMERSERIZEILK=OptTime*SipBuy;

branch,1:
if, acttypemerserize.NE.pretypemerserize, PA51:
else, PA6;

PA51 ASSIGN:TIMEMERSERIZESON=TIMEMERSERIZEILK+4:NEXT(PA61);

PA6 ASSIGN:TIMEMERSERIZESON=TIMEMERSERIZEILK;

PA61 DELAY:TIMEMERSERIZESON;
assign:pretypemerserize=acttypemerserize;

RELEASE:MERSERIZER;
ROUTE;

STATION,YIKAMA;
QUEUE,YIKAMAQ;
SEIZE:YIKAMAR;

assign:acttypeyikama=renklibeyaz;

PA7 ASSIGN:TIMEYIKAMAILK=OptTime*SipBuy;

branch,1:
if, acttypeyikama.NE.pretypeyikama, PA71:
else, PA8;

PA71 ASSIGN:TIMEYIKAMASON=TIMEYIKAMAILK+2:NEXT(PA81);

PA8 ASSIGN:TIMEYIKAMASON=TIMEYIKAMAILK;

PA81 DELAY:TIMEYIKAMASON;
assign:pretypeyikama=acttypeyikama;

RELEASE:YIKAMAR;
ROUTE;

STATION,KURUTMA;
QUEUE,KURUTMAQ;
SEIZE:KURUTMAR;
ASSIGN:TIMEKURUTMA=OptTime*SipBuy;

DELAY:TIMEKURUTMA;
RELEASE:KURUTMAR;
ROUTE;

STATION,ZIMPARA;
QUEUE,ZIMPARAQ;
SEIZE:ZIMPARAR;
ASSIGN:TIMEZIMPARA=OptTime*SipBuy;
DELAY:TIMEZIMPARA;
RELEASE:ZIMPARAR;
ROUTE;

STATION,BOYAMA;
QUEUE,BOYAMAQ;
SEIZE:BOYAMAR;
assign:acttypeboyama=renktonu;
ASSIGN:TIMEBOYAMA=OptTime*SipBuy+TF(prettypeboyama, acttypeboyama);
DELAY:TIMEBOYAMA;
assign:prettypeboyama=acttypeboyama;
assign:renklibeyaz=2;
RELEASE:BOYAMAR;
ROUTE;

STATION,APRE;
QUEUE,APREQ;
SEIZE:APRER;
ASSIGN:TIMEAPRE=OptTime*SipBuy;
DELAY:TIMEAPRE;
RELEASE:APRER;
ROUTE;

STATION,SANFOR;
QUEUE,SANFORQ;
SEIZE:SANFORR;
ASSIGN:TIMESANFOR=OptTime*SipBuy;
DELAY:TIMESANFOR;
RELEASE:SANFORR;
ROUTE;

STATION,FIKSE;
QUEUE,FIKSEQ;
SEIZE:FIKSER;
ASSIGN:TIMEFIKSE=OptTime*SipBuy;
DELAY:TIMEFIKSE;
RELEASE:FIKSER;
ROUTE;

STATION,JETBOYAMA;
QUEUE,JETBOYAMAQ;
SEIZE:JETBOYAMAR;
assign:acttypejetboyama=renktonu;
ASSIGN:TIMEJETBOYAMA=OptTime*SipBuy/1500+
TF(prettypejetboyama, acttypejetboyama);
DELAY:TIMEJETBOYAMA;
assign:prettypejetboyama=acttypejetboyama;
assign:renklibeyaz=2;

RELEASE:JETBOYAMAR;
ROUTE;

STATION,YAKMA;
QUEUE,YAKMAQ;
SEIZE:YAKMAR;
ASSIGN:TIMEYAKMA=OptTime*SipBuy;
DELAY:TIMEYAKMA;
RELEASE:YAKMAR;
ROUTE;

STATION,BASKI;
QUEUE,BASKIQ;
SEIZE:BASKIR;
ASSIGN:TIMEBASKI=OptTime*SipBuy;
DELAY:TIMEBASKI;
RELEASE:BASKIR;
ROUTE;

STATION,KONDANSE;
QUEUE,KONDANSEQ;
SEIZE:KONDANSER;
ASSIGN:TIMEKONDANSE=OptTime*SipBuy;
DELAY:TIMEKONDANSE;
RELEASE:KONDANSER;
ROUTE;

STATION,KATLAMA;
QUEUE,KATLAMAQ;
SEIZE:KATLAMAR;
RELEASE:KATLAMAR;

ASSIGN:GECIKME=DUEDATE - TNOW;

IF: GECIKME<0;
COUNT:GECKALANLAR;
ASSIGN:TOPLAMGECIKME=TOPLAMGECIKME+ABS (GECIKME) ;
ELSE;
COUNT:ERKENBITENLER;
ASSIGN:TOPLAMERKEN=TOPLAMERKEN+ABS (GECIKME) ;
ENDIF;

TALLY:FlowTime,INT (TimeIn):DISPOSE;

create,1,simson:simson+1,1;
ASSIGN:ORTGECIKME=TOPLAMGECIKME/NC (GECKALANLAR) ;
ASSIGN:ORTERKENBIT=TOPLAMERKEN/NC (ERKENBITENLER) ;
WRITE, SONUCDOSYA:NC (GECKALANLAR),ORTGECIKME,
NC (ERKENBITENLER),ORTERKENBIT:dispose;

END;

EDD kuralına ait simülasyon modelinin experiment dosyası

BEGIN;

PROJECT, Gulsun EDD;

VARIABLES: secim:

acttypeyikama:
 pretypeyikama, 1:
acttypemerserize:
 pretypemerserize, 1:
acttypeboyama:
 pretypeboyama, 1:
acttypejetboyama:
 pretypejetboyama, 1:
TOPLAMGECIKME, 0:
TOPLAMERKEN, 0:
ORTGECIKME:
ORTERKENBIT:
simson, 40000;

ATTRIBUTES: TimeIn: SipBuy: OptTime: DueDate: renklibeyaz: renktonu:

TIMEHASILSOKUMU:
TIMEBEKLETME: TIMEKOMBINKASAR: TIMEMERSERIZEILK:
TIMEMERSERIZESON: TIMEYIKAMAILK: TIMEYIKAMASON:
TIMEKURUTMA: TIMEZIMPARA: TIMEBOYAMA: TIMEAPRE:
TIMESANFOR: TIMEFIKSE: TIMEJETBOYAMA:
TIMEYAKMA: TIMEBASKI: TIMEKONDANSE: GECIKME;

COUNTERS: ENTNUM:

GECKALANLAR:
ERKENBITENLER;

QUEUES: HASILSOKUMUQ, LVF (DueDate):

BEKLETMEQ, LVF (DueDate):
KOMBINKASARQ, LVF (DueDate):
MERSERIZEQ, LVF (DueDate):
YIKAMAQ, LVF (DueDate):
KURUTMAQ, LVF (DueDate):
ZIMPARAQ, LVF (DueDate):
BOYAMAQ, LVF (DueDate):
APREQ, LVF (DueDate):
SANFORQ, LVF (DueDate):
FIKSEQ, LVF (DueDate):
JETBOYAMAQ, LVF (DueDate):
YAKMAQ, LVF (DueDate):
BASKIQ, LVF (DueDate):
KONDANSEQ, LVF (DueDate):
KATLAMAQ, LVF (DueDate);

RESOURCES: HASILSOKUMUR:

BEKLETMER, 60:
KOMBINKASARR:
MERSERIZER:
YIKAMAR, 4:

KURUTMAR, 3:
ZIMPARAR:
BOYAMAR, 3:
APRER, 2:
SANFORR, 2:
FIKSER:
JETBOYAMAR, 5:
YAKMAR:
BASKIR:
KONDANSER:
KATLAMAR;

STATIONS: HASILSOKUMU:

BEKLETME:
KOMBINKASAR:
MERSERIZE:
YIKAMA:
KURUTMA:
ZIMPARA:
BOYAMA:
APRE:
SANFOR:
FIKSE:
JETBOYAMA:
YAKMA:
BASKI:
KONDANSE:
KATLAMA;

SEQUENCES: 1, PAM,
HASILSOKUMU, OptTime=0.00033 & BEKLETME, OptTime=12 &
KOMBINKASAR, OptTime=0.00080 & MERSERIZE, OptTime=0.00033 &
YIKAMA, OptTime=0.00059 & KURUTMA, OptTime=0.00050 &
ZIMPARA, OptTime=0.00166 & YIKAMA, OptTime=0.00058 &
KURUTMA, OptTime=0.00060 & BOYAMA, OptTime=0.00235 &
BEKLETME, OptTime=24 & YIKAMA, OptTime=0.00095 &
KURUTMA, OptTime=0.00088 & APRE, OptTime=0.00078 &
SANFOR, OptTime=0.00060 & KATLAMA:

2, KORD,
KOMBINKASAR, OptTime=0.00080 & YIKAMA, OptTime=0.00058 &
KURUTMA, OptTime=0.00060 & BOYAMA, OptTime=0.00250 &
BEKLETME, OptTime=24 & YIKAMA, OptTime=0.00195 &
KURUTMA, OptTime=0.00088 & APRE, OptTime=0.00078 &
SANFOR, OptTime=0.00060 & KATLAMA:

3, JET,
YIKAMA, OptTime=0.00062 & KURUTMA, OptTime=0.00048 &
YAKMA, OptTime=0.00062 & FIKSE, OptTime=0.00066 &
JETBOYAMA, OptTime=5.40 & KURUTMA, OptTime=0.00085 &
APRE, OptTime=0.00061 & SANFOR, OptTime=0.00062 &
KATLAMA:

4, BASKILI,
HASILSOKUMU, OptTime=0.00033 & BEKLETME, OptTime=12 &
KOMBINKASAR, OptTime=0.00066 & MERSERIZE, OptTime=0.00033 &

YIKAMA,OptTime=0.00063 & KURUTMA,OptTime=0.00046 &
BASKI,OptTime=0.00435 & KONDANSE,OptTime=0.0033 &
APRE,OptTime=0.00052 & SANFOR,OptTime=0.00048 &
KATLAMA:

5, PEVIS,
HASILSOKUMU,OptTime=0.00043 & BEKLETME,OptTime=12 &
YIKAMA,OptTime=0.00058 & KURUTMA,OptTime=0.00041 &
BOYAMA,OptTime=0.00215 & KURUTMA,OptTime=0.00045 &
APRE,OptTime=0.00050 & KATLAMA:

6, GOMLEK,
YIKAMA,OptTime=0.00055 & KURUTMA,OptTime=0.00041 &
YAKMA,OptTime=0.00062 & MERSEIZE,OptTime=0.00033 &
KURUTMA,OptTime=0.00041 & APRE,OptTime=0.00065 &
SANFOR,OptTime=0.00052 & KATLAMA:

7, PERILEN,
YIKAMA,OptTime=0.00037 & FIKSE,OptTime=0.00058 &
APRE,OptTime=0.00095 & SANFOR,OptTime=0.00041 &
KATLAMA;

TALLIES:FlowTime;

DSTATS:NQ(HASILSOKUMUQ)*SipBuy/1000:

NQ(BEKLETMEQ)*SipBuy/1000:

NQ(KOMBINKASARQ)*SipBuy/1000:

NQ(MERSEIZEQ)*SipBuy/1000:

NQ(YIKAMAQ)*SipBuy/1000:

NQ(KURUTMAQ)*SipBuy/1000:

NQ(ZIMPARAQ)*SipBuy/1000:

NQ(BOYAMAQ)*SipBuy/1000:

NQ(APREQ)*SipBuy/1000:

NQ(SANFORQ)*SipBuy/1000:

NQ(FIKSEQ)*SipBuy/1000:

NQ(JETBOYAMAQ)*SipBuy/1000:

NQ(YAKMAQ)*SipBuy/1000:

NQ(BASKIQ)*SipBuy/1000:

NQ(KONDANSEQ)*SipBuy/1000:

NQ(HASILSOKUMUQ)*SipBuy/1000+NQ(BEKLETMEQ)*SipBuy/1000+
NQ(KOMBINKASARQ)*SipBuy/1000+NQ(MERSEIZEQ)*SipBuy/1000+
NQ(YIKAMAQ)*SipBuy/1000+ NQ(KURUTMAQ)*SipBuy/1000+
NQ(ZIMPARAQ)*SipBuy/1000+ NQ(BOYAMAQ)*SipBuy/1000+
NQ(APREQ)*SipBuy/1000+ NQ(SANFORQ)*SipBuy/1000+
NQ(FIKSEQ)*SipBuy/1000+NQ(JETBOYAMAQ)*SipBuy/1000+
NQ(YAKMAQ)*SipBuy/1000+ NQ(BASKIQ)*SipBuy/1000+
NQ(KONDANSEQ)*SipBuy/1000,TOTALWIP:

NR(HASILSOKUMUR)*100:

NR(BEKLETMER)/60*100:

NR(KOMBINKASARR)*100:

NR(MERSEIZER)*100:

NR(YIKAMAR)/4*100:

NR(KURUTMAR)/3*100:

```
NR(ZIMPARAR)*100:
NR(BOYAMAR)/3*100:
NR(APRER)/2*100:
NR(SANFORR)/2*100:
NR(FIKSER)*100:
NR(JETBOYAMAR)/5*100:
NR(YAKMAR)*100:
NR(BASKIR)*100:
NR(KONDANSER)*100;
```

```
FILES:SONUCDOSYA,"disp21(2.5).DAT",SEQ,WKS,DIS;
```

```
TABLES: 1, 1,1, 0,0,0:
        2, 1,1, 2,0,2:
        3, 1,1, 3,2,0;
```

```
REPLICATE,5,0,40000,,,5000;
```

```
END;
```



LWKR kuralına ait simülasyon modeli

BEGIN;

CREATE:exponential(2.2,1):MARK(TimeIn);
COUNT:ENTNUM;

ASSIGN:NS=discrete(0.18,1,0.34,2,0.57,3,0.65,4,
0.74,5,0.80,6,1.0,7):
SipBuy=discrete(0.2,2500,0.6,4000,0.85,6000,0.95,10000,
1.0,25000);

BRANCH,1:WITH,0.3,export:
WITH,0.7,yerli;

export ASSIGN:DueDate=TNOW+UNIF(336,504):NEXT(PA1);
yerli ASSIGN:DueDate=TNOW+UNIF(504,672);

PA1 IF: NS==1;
ASSIGN:TTIME=SipBuy*0.01095+36:
Next(PA2);
ELSEIF:
NS==2;
ASSIGN:TTIME=SipBuy*0.00889+24:
Next(PA2);
ELSEIF:
NS==3;
ASSIGN:TTIME=((SipBuy*0.00446)+(SipBuy/1500)*5.4):
Next(PA2);
ELSEIF:
NS==4;
ASSIGN:TTIME=(SipBuy*0.01106+12):
Next(PA3);
ELSEIF:
NS==5;
ASSIGN:TTIME=(SipBuy*0.00452+12):
Next(PA2);
ELSEIF:
NS==6;
ASSIGN:TTIME=(SipBuy*0.00309):
Next(PA3);
ELSE;
ASSIGN:TTIME=(SipBuy*0.00231):
Next(PA3);
ENDIF;

PA2 assign:renklibeyaz=1;
PA21 assign:renktonu=discrete(0.3,1,0.6,2,1.0,3):NEXT(PA4);

PA3 assign:renklibeyaz=2;
PA31 assign:renktonu=1;

PA4 ROUTE;

STATION, HASILSOKUMU;
 QUEUE, HASILSOKUMUQ;
 SEIZE: HASILSOKUMUR;
 ASSIGN: TIMEHASILSOKUMU=OptTime*SipBuy;
 DELAY: TIMEHASILSOKUMU;
 RELEASE: HASILSOKUMUR;
 ASSIGN: TTIME=TTIME-TIMEHASILSOKUMU;
 ROUTE;

STATION, BEKLETME;
 QUEUE, BEKLETMEQ;
 SEIZE: BEKLETMER;
 ASSIGN: TIMEBEKLETME=OptTime;
 DELAY: TIMEBEKLETME;
 RELEASE: BEKLETMER;
 ASSIGN: TTIME=TTIME-TIMEBEKLETME;
 ROUTE;

STATION, KOMBINKASAR;
 QUEUE, KOMBINKASARQ;
 SEIZE: KOMBINKASARR;
 ASSIGN: TIMEKOMBINKASAR=OptTime*SipBuy;
 DELAY: TIMEKOMBINKASAR;
 RELEASE: KOMBINKASARR;
 ASSIGN: TTIME=TTIME-TIMEKOMBINKASAR;
 ROUTE;

STATION, MERSERIZE;
 QUEUE, MERSERIZEQ;
 SEIZE: MERSERIZER;

assign: acttypemerserize=renklibeyaz;

PA5 ASSIGN: TIMEMERSERIZEILK=OptTime*SipBuy;

branch, 1:

if, acttypemerserize.NE.pretypemerserize, PA51;
 else, PA6;

PA51 ASSIGN: TIMEMERSERIZESON=TIMEMERSERIZEILK+4:NEXT(PA61);

PA6 ASSIGN: TIMEMERSERIZESON=TIMEMERSERIZEILK;

PA61 DELAY: TIMEMERSERIZESON;
 assign: pretypemerserize=acttypemerserize;

RELEASE: MERSERIZER;
 ASSIGN: TTIME=TTIME-TIMEMERSERIZESON;
 ROUTE;

STATION, YIKAMA;
 QUEUE, YIKAMAQ;
 SEIZE: YIKAMAR;
 assign: acttypeyikama=renklibeyaz;

```

PA7      ASSIGN:TIMEYIKAMAILK=OptTime*SipBuy;
branch,1:
        if, acttypeyikama.NE.prettypeyikama, PA71:
        else, PA8;

PA71     ASSIGN:TIMEYIKAMASON=TIMEYIKAMAILK+2:NEXT(PA81);

PA8      ASSIGN:TIMEYIKAMASON=TIMEYIKAMAILK;

PA81     DELAY:TIMEYIKAMASON;
        assign:prettypeyikama=acttypeyikama;

RELEASE:YIKAMAR;
ASSIGN:TTIME=TTIME-TIMEYIKAMASON;
ROUTE;

STATION, KURUTMA;
QUEUE, KURUTMAQ;
SEIZE:KURUTMAR;
ASSIGN:TIMEKURUTMA=OptTime*SipBuy;
DELAY:TIMEKURUTMA;
RELEASE:KURUTMAR;
ASSIGN:TTIME=TTIME-TIMEKURUTMA;
ROUTE;

STATION, ZIMPARA;
QUEUE, ZIMPARAQ;
SEIZE:ZIMPARAR;
ASSIGN:TIMEZIMPARA=OptTime*SipBuy;
DELAY:TIMEZIMPARA;
RELEASE:ZIMPARAR;
ASSIGN:TTIME=TTIME-TIMEZIMPARA;
ROUTE;

STATION, BOYAMA;
QUEUE, BOYAMAQ;
SEIZE:BOYAMAR;
assign:acttypeboyama=renktonu;
ASSIGN:TIMEBOYAMA=OptTime*SipBuy+TF(prettypeboyama, acttypeboyama);
DELAY:TIMEBOYAMA;
assign:prettypeboyama=acttypeboyama;
assign:renklibeyaz=2;
RELEASE:BOYAMAR;
ASSIGN:TTIME=TTIME-TIMEBOYAMA;
ROUTE;

STATION, APRE;
QUEUE, APREQ;
SEIZE:APRER;
ASSIGN:TIMEAPRE=OptTime*SipBuy;
DELAY:TIMEAPRE;
RELEASE:APRER;
ASSIGN:TTIME=TTIME-TIMEAPRE;
ROUTE;

```

STATION, SANFOR;
QUEUE, SANFORQ;
SEIZE: SANFORR;
ASSIGN: TIMESANFOR=OptTime*SipBuy;
DELAY: TIMESANFOR;
RELEASE: SANFORR;
ASSIGN: TTIME=TTIME-TIMESANFOR;
ROUTE;

STATION, FIKSE;
QUEUE, FIKSEQ;
SEIZE: FIKSER;
ASSIGN: TIMEFIKSE=OptTime*SipBuy;
DELAY: TIMEFIKSE;
RELEASE: FIKSER;
ASSIGN: TTIME=TTIME-TIMEFIKSE;
ROUTE;

STATION, JETBOYAMA;
QUEUE, JETBOYAMAQ;
SEIZE: JETBOYAMAR;
assign: acttypejetboyama=renktonu;
ASSIGN: TIMEJETBOYAMA=OptTime*SipBuy/1500+
 TF(pretypejetboyama, acttypejetboyama);
DELAY: TIMEJETBOYAMA;
assign: pretypejetboyama=acttypejetboyama;
assign: renklibeyaz=2;
RELEASE: JETBOYAMAR;
ASSIGN: TTIME=TTIME-TIMEJETBOYAMA;
ROUTE;

STATION, YAKMA;
QUEUE, YAKMAQ;
SEIZE: YAKMAR;
ASSIGN: TIMEYAKMA=OptTime*SipBuy;
DELAY: TIMEYAKMA;
RELEASE: YAKMAR;
ASSIGN: TTIME=TTIME-TIMEYAKMA;
ROUTE;

STATION, BASKI;
QUEUE, BASKIQ;
SEIZE: BASKIR;
ASSIGN: TIMEBASKI=OptTime*SipBuy;
DELAY: TIMEBASKI;
RELEASE: BASKIR;
ASSIGN: TTIME=TTIME-TIMEBASKI;
ROUTE;

STATION, KONDANSE;
QUEUE, KONDANSEQ;
SEIZE: KONDANSER;
ASSIGN: TIMEKONDANSE=OptTime*SipBuy;
DELAY: TIMEKONDANSE;
RELEASE: KONDANSER;
ASSIGN: TTIME=TTIME-TIMEKONDANSE;

```
ROUTE;

STATION, KATLAMA;
QUEUE, KATLAMAQ;
SEIZE: KATLAMAR;
RELEASE: KATLAMAR;

ASSIGN: GECIKME = DUE DATE - TNOW;

IF:      GECIKME < 0;
        COUNT: GECKALANLAR;
        ASSIGN: TOPLAMGECIKME = TOPLAMGECIKME + ABS (GECIKME);
ELSE;
        COUNT: ERKENBITENLER;
        ASSIGN: TOPLAMERKEN = TOPLAMERKEN + ABS (GECIKME);
ENDIF;

TALLY: FlowTime, INT (TimeIn) : DISPOSE;

create, 1, simson: simson + 1, 1;
        ASSIGN: ORTGECIKME = TOPLAMGECIKME / NC (GECKALANLAR);
        ASSIGN: ORTERKENBIT = TOPLAMERKEN / NC (ERKENBITENLER);
        WRITE, SONUCDOSYA: NC (GECKALANLAR), ORTGECIKME,
                NC (ERKENBITENLER), ORTERKENBIT: dispose;

END;
```

LWKR kuralına ait simülasyon modelinin experiment dosyası

BEGIN;
PROJECT,gulsun LWR;

VARIABLES:secim:
 acttypeyikama:
 pretypeyikama,1:
 acttypemerserize:
 pretypemerserize,1:
 acttypeboyama:
 pretypeboyama,1:
 acttypejetboyama:
 pretypejetboyama,1:
 TOPLAMGECIKME,0:
 TOPLAMERKEN,0:
 ORTGECIKME:
 ORTERKENBIT:
 simson,40000;

ATTRIBUTES:TimeIn:SipBuy:OptTime:DueDate:TTIME:
 renklibeyaz:renktonu:
 TIMEHASILSOKUMU:
 TIMEBEKLETME:TIMEKOMBINKASAR:TIMEMERSERIZEILK:
 TIMEMERSERIZERSON:TIMEYIKAMAILK:TIMEYIKAMASON:
 TIMEKURUTMA:TIMEZIMPARA:TIMEBOYAMA:TIMEAPRE:
 TIMESANFOR:TIMEFIKSE:TIMEJETBOYAMA:
 TIMEYAKMA:TIMEBASKI:TIMEKONDANSE:GECIKME;

COUNTERS:ENTNUM:
 GECKALANLAR:
 ERKENBITENLER;

QUEUES:HASILSOKUMUQ,LVF(TTIME):
 BEKLETMEQ,LVF(TTIME):
 KOMBINKASARQ,LVF(TTIME):
 MERSERIZEQ,LVF(TTIME):
 YIKAMAQ,LVF(TTIME):
 KURUTMAQ,LVF(TTIME):
 ZIMPARAQ,LVF(TTIME):
 BOYAMAQ,LVF(TTIME):
 APREQ,LVF(TTIME):
 SANFORQ,LVF(TTIME):
 FIKSEQ,LVF(TTIME):
 JETBOYAMAQ,LVF(TTIME):
 YAKMAQ,LVF(TTIME):
 BASKIQ,LVF(TTIME):
 KONDANSEQ,LVF(TTIME):
 KATLAMAQ,LVF(TTIME);

RESOURCES:HASILSOKUMUR:
 BEKLETMER,60:
 KOMBINKASARR:
 MERSERIZER:

YIKAMAR, 4:
KURUTMAR, 3:
ZIMPARAR:
BOYAMAR, 3:
APRER, 2:
SANFORR, 2:
FIKSER:
JETBOYAMAR, 5:
YAKMAR:
BASKIR:
KONDANSER:
KATLAMAR;

STATIONS: HASILSOKUMU:

BEKLETME:
KOMBINKASAR:
MERSERIZE:
YIKAMA:
KURUTMA:
ZIMPARA:
BOYAMA:
APRE:
SANFOR:
FIKSE:
JETBOYAMA:
YAKMA:
BASKI:
KONDANSE:
KATLAMA;

SEQUENCES: 1, PAM,
HASILSOKUMU, OptTime=0.00033 & BEKLETME, OptTime=12 &
KOMBINKASAR, OptTime=0.00080 & MERSERIZE, OptTime=0.00033 &
YIKAMA, OptTime=0.00059 & KURUTMA, OptTime=0.00050 &
ZIMPARA, OptTime=0.00166 & YIKAMA, OptTime=0.00058 &
KURUTMA, OptTime=0.00060 & BOYAMA, OptTime=0.00235 &
BEKLETME, OptTime=24 & YIKAMA, OptTime=0.00095 &
KURUTMA, OptTime=0.00088 & APRE, OptTime=0.00078 &
SANFOR, OptTime=0.00060 & KATLAMA:

2, KORD,
KOMBINKASAR, OptTime=0.00080 & YIKAMA, OptTime=0.00058 &
KURUTMA, OptTime=0.00060 & BOYAMA, OptTime=0.00250 &
BEKLETME, OptTime=24 & YIKAMA, OptTime=0.00195 &
KURUTMA, OptTime=0.00088 & APRE, OptTime=0.00078 &
SANFOR, OptTime=0.00060 & KATLAMA:

3, JET,
YIKAMA, OptTime=0.00062 & KURUTMA, OptTime=0.00048 &
YAKMA, OptTime=0.00062 & FIKSE, OptTime=0.00066 &
JETBOYAMA, OptTime=5.40 & KURUTMA, OptTime=0.00085 &
APRE, OptTime=0.00061 & SANFOR, OptTime=0.00062 &
KATLAMA:

4, BASKILI,
HASILSOKUMU, OptTime=0.00033 & BEKLETME, OptTime=12 &

KOMBINKASAR,OptTime=0.00066 & MERSERIZE,OptTime=0.00033 &
 YIKAMA,OptTime=0.00063 & KURUTMA,OptTime=0.00046 &
 BASKI,OptTime=0.00435 & KONDANSE,OptTime=0.0033 &
 APRE,OptTime=0.00052 & SANFOR,OptTime=0.00048 &
 KATLAMA:

5, PEVIS,
 HASILSOKUMU,OptTime=0.00043 & BEKLETME,OptTime=12 &
 YIKAMA,OptTime=0.00058 & KURUTMA,OptTime=0.00041 &
 BOYAMA,OptTime=0.00215 & KURUTMA,OptTime=0.00045 &
 APRE,OptTime=0.00050 & KATLAMA:

6, GOMLEK,
 YIKAMA,OptTime=0.00055 & KURUTMA,OptTime=0.00041 &
 YAKMA,OptTime=0.00062 & MERSERIZE,OptTime=0.00033 &
 KURUTMA,OptTime=0.00041 & APRE,OptTime=0.00065 &
 SANFOR,OptTime=0.00052 & KATLAMA:

7, PERILEN,
 YIKAMA,OptTime=0.00037 & FIKSE,OptTime=0.00058 &
 APRE,OptTime=0.00095 & SANFOR,OptTime=0.00041 &
 KATLAMA;

TALLIES:FlowTime;

DSTATS:NQ(HASILSOKUMUQ)*SipBuy/1000:
 NQ(BEKLETMEQ)*SipBuy/1000:
 NQ(KOMBINKASARQ)*SipBuy/1000:
 NQ(MERSERIZEQ)*SipBuy/1000:
 NQ(YIKAMAQ)*SipBuy/1000:
 NQ(KURUTMAQ)*SipBuy/1000:
 NQ(ZIMPARAQ)*SipBuy/1000:
 NQ(BOYAMAQ)*SipBuy/1000:
 NQ(APREQ)*SipBuy/1000:
 NQ(SANFORQ)*SipBuy/1000:
 NQ(FIKSEQ)*SipBuy/1000:
 NQ(JETBOYAMAQ)*SipBuy/1000:
 NQ(YAKMAQ)*SipBuy/1000:
 NQ(BASKIQ)*SipBuy/1000:
 NQ(KONDANSEQ)*SipBuy/1000:

NQ(HASILSOKUMUQ)*SipBuy/1000+ NQ(BEKLETMEQ)*SipBuy/1000+
 NQ(KOMBINKASARQ)*SipBuy/1000+ NQ(MERSERIZEQ)*SipBuy/1000+
 NQ(YIKAMAQ)*SipBuy/1000+ NQ(KURUTMAQ)*SipBuy/1000+
 NQ(ZIMPARAQ)*SipBuy/1000+ NQ(BOYAMAQ)*SipBuy/1000+
 NQ(APREQ)*SipBuy/1000+ NQ(SANFORQ)*SipBuy/1000+
 NQ(FIKSEQ)*SipBuy/1000+ NQ(JETBOYAMAQ)*SipBuy/1000+
 NQ(YAKMAQ)*SipBuy/1000+ NQ(BASKIQ)*SipBuy/1000+
 NQ(KONDANSEQ)*SipBuy/1000, TOTALWIP:

NR(HASILSOKUMUR)*100:
 NR(BEKLETMER)/60*100:
 NR(KOMBINKASARR)*100:
 NR(MERSERIZER)*100:
 NR(YIKAMAR)/4*100:
 NR(KURUTMAR)/3*100:

```
NR(ZIMPARAR)*100:
NR(BOYAMAR)/3*100:
NR(APRER)/2*100:
NR(SANFORR)/2*100:
NR(FIKSER)*100:
NR(JETBOYAMAR)/5*100:
NR(YAKMAR)*100:
NR(BASKIR)*100:
NR(KONDANSER)*100;
```

```
FILES:SONUCDOSYA,"disp26(2.5).DAT",SEQ,WKS,DIS;
```

```
TABLES: 1, 1,1, 0,0,0:
        2, 1,1, 2,0,2:
        3, 1,1, 3,2,0;
```

```
REPLICATE,5,0,40000,,,5000;
```

```
END;
```

```
Experiment compilation completed with 0 error(s) and 0 warning(s).
```



S/OP kuralına ait simülasyon modeli

BEGIN;

CREATE:exponential(2.5,1):MARK(TimeIn);

COUNT:ENTNUM;

ASSIGN:NS=discrete(0.18,1,0.34,2,0.57,3,0.65,4,
0.74,5,0.80,6,1.0,7):

SipBuy=discrete(0.2,2500,0.6,4000,0.85,6000,0.95,10000,
1.0,25000);

BRANCH,1:WITH,0.3,export:

WITH,0.7,yerli;

export ASSIGN:DueDate=TNOW+UNIF(336,504):NEXT(PA1);

yerli ASSIGN:DueDate=TNOW+UNIF(504,672);

PA1 IF: NS==1;

ASSIGN:TTIME=SipBuy*0.01095+36;

ASSIGN:NUMOPR=16;

ASSIGN:STRM=DueDate-TNOW-TTIME;

ASSIGN:STRMOP=STRM/NUMOPR:

Next(PA2);

ELSEIF:

NS==2;

ASSIGN:TTIME=SipBuy*0.00889+24;

ASSIGN:NUMOPR=12;

ASSIGN:STRM=DueDate-TNOW-TTIME;

ASSIGN:STRMOP=STRM/NUMOPR:

Next(PA2);

ELSEIF:

NS==3;

ASSIGN:TTIME=((SipBuy*0.00446)+(SipBuy/1500)*5.4);

ASSIGN:NUMOPR=9;

ASSIGN:STRM=DueDate-TNOW-TTIME;

ASSIGN:STRMOP=STRM/NUMOPR:

Next(PA2);

ELSEIF:

NS==4;

ASSIGN:TTIME=(SipBuy*0.01106+12);

ASSIGN:NUMOPR=11;

ASSIGN:STRM=DueDate-TNOW-TTIME;

ASSIGN:STRMOP=STRM/NUMOPR:

Next(PA3);

ELSEIF:

NS==5;

ASSIGN:TTIME=(SipBuy*0.00452+12);

ASSIGN:NUMOPR=8;

ASSIGN:STRM=DueDate-TNOW-TTIME;

ASSIGN:STRMOP=STRM/NUMOPR:

Next(PA2);

ELSEIF:

```

        NS==6;
        ASSIGN:TTIME=(SipBuy*0.00309);
        ASSIGN:NUMOPR=8;
        ASSIGN:STRM=DueDate-TNOW-TTIME;
        ASSIGN:STRMOP=STRM/NUMOPR;
        Next (PA3);
ELSE;
        ASSIGN:TTIME=(SipBuy*0.00231);
        ASSIGN:NUMOPR=5;
        ASSIGN:STRM=DueDate-TNOW-TTIME;
        ASSIGN:STRMOP=STRM/NUMOPR;
        Next (PA3);
ENDIF;

PA2    assign:renklibeyaz=1;
PA21   assign:renktonu=discrete(0.3,1,0.6,2,1.0,3):NEXT(PA4);

PA3    assign:renklibeyaz=2;
PA31   assign:renktonu=1;

PA4    ROUTE;

STATION,HASILSOKUMU;
QUEUE,HASILSOKUMUQ;
SEIZE:HASILSOKUMUR;
ASSIGN:TIMEHASILSOKUMU=OptTime*SipBuy;
DELAY:TIMEHASILSOKUMU;
RELEASE:HASILSOKUMUR;
ASSIGN:TTIME=TTIME-TIMEHASILSOKUMU;
ASSIGN:STRM=DueDate-TNOW-TTIME;
ASSIGN:NUMOPR=NUMOPR-1;
ASSIGN:STRMOP=STRM/NUMOPR;
ROUTE;

STATION,BEKLETME;
QUEUE,BEKLETMEQ;
SEIZE:BEKLETMER;
ASSIGN:TIMEBEKLETME=OptTime;
DELAY:TIMEBEKLETME;
RELEASE:BEKLETMER;
ASSIGN:TTIME=TTIME-TIMEBEKLETME;
ASSIGN:STRM=DueDate-TNOW-TTIME;
ASSIGN:NUMOPR=NUMOPR-1;
ASSIGN:STRMOP=STRM/NUMOPR;
ROUTE;

STATION,KOMBINKASAR;
QUEUE,KOMBINKASARQ;
SEIZE:KOMBINKASARR;
ASSIGN:TIMEKOMBINKASAR=OptTime*SipBuy;
DELAY:TIMEKOMBINKASAR;
RELEASE:KOMBINKASARR;
ASSIGN:TTIME=TTIME-TIMEKOMBINKASAR;

```

ASSIGN:STRM=DueDate-TNOW-TTIME;
ASSIGN:NUMOPR=NUMOPR-1;
ASSIGN:STRMOP=STRM/NUMOPR;
ROUTE;

STATION,MERSERIZE;
QUEUE,MERSERIZEQ;
SEIZE:MERSERIZER;

assign:acttypemerserize=renklibeyaz;

PA5 ASSIGN:TIMEMERSERIZEILK=OptTime*SipBuy;

branch,1:
 if, acttypemerserize.NE.pretypemerserize, PA51:
 else, PA6;

PA51 ASSIGN:TIMEMERSERIZESON=TIMEMERSERIZEILK+4:NEXT(PA61);

PA6 ASSIGN:TIMEMERSERIZESON=TIMEMERSERIZEILK;

PA61 DELAY:TIMEMERSERIZESON;
 assign:pretypemerserize=acttypemerserize;

RELEASE:MERSERIZER;
ASSIGN:TTIME=TTIME-TIMEMERSERIZESON;
ASSIGN:STRM=DueDate-TNOW-TTIME;
ASSIGN:NUMOPR=NUMOPR-1;
ASSIGN:STRMOP=STRM/NUMOPR;
ROUTE;

STATION,YIKAMA;
QUEUE,YIKAMAQ;
SEIZE:YIKAMAR;
assign:acttypeyikama=renklibeyaz;

PA7 ASSIGN:TIMEYIKAMAILK=OptTime*SipBuy;

branch,1:
 if, acttypeyikama.NE.pretypeyikama, PA71:
 else, PA8;

PA71 ASSIGN:TIMEYIKAMASON=TIMEYIKAMAILK+2:NEXT(PA81);

PA8 ASSIGN:TIMEYIKAMASON=TIMEYIKAMAILK;

PA81 DELAY:TIMEYIKAMASON;
 assign:pretypeyikama=acttypeyikama;

RELEASE:YIKAMAR;
ASSIGN:TTIME=TTIME-TIMEYIKAMASON;
ASSIGN:STRM=DueDate-TNOW-TTIME;
ASSIGN:NUMOPR=NUMOPR-1;
ASSIGN:STRMOP=STRM/NUMOPR;
ROUTE;

STATION, KURUTMA;
QUEUE, KURUTMAQ;
SEIZE: KURUTMAR;
ASSIGN: TIMEKURUTMA=OptTime*SipBuy;
DELAY: TIMEKURUTMA;
RELEASE: KURUTMAR;
ASSIGN: TTIME=TTIME-TIMEKURUTMA;
ASSIGN: STRM=DueDate-TNOW-TTIME;
ASSIGN: NUMOPR=NUMOPR-1;
ASSIGN: STRMOP=STRM/NUMOPR;
ROUTE;

STATION, ZIMPARA;
QUEUE, ZIMPARAQ;
SEIZE: ZIMPARAR;
ASSIGN: TIMEZIMPARA=OptTime*SipBuy;
DELAY: TIMEZIMPARA;
RELEASE: ZIMPARAR;
ASSIGN: TTIME=TTIME-TIMEZIMPARA;
ASSIGN: STRM=DueDate-TNOW-TTIME;
ASSIGN: NUMOPR=NUMOPR-1;
ASSIGN: STRMOP=STRM/NUMOPR;
ROUTE;

STATION, BOYAMA;
QUEUE, BOYAMAQ;
SEIZE: BOYAMAR;
assign: acttypeboyama=renktonu;
ASSIGN: TIMEBOYAMA=OptTime*SipBuy+TF(pretypeboyama, acttypeboyama);
DELAY: TIMEBOYAMA;
assign: pretypeboyama=acttypeboyama;
assign: renklibeyaz=2;
RELEASE: BOYAMAR;
ASSIGN: TTIME=TTIME-TIMEBOYAMA;
ASSIGN: STRM=DueDate-TNOW-TTIME;
ASSIGN: NUMOPR=NUMOPR-1;
ASSIGN: STRMOP=STRM/NUMOPR;
ROUTE;

STATION, APRE;
QUEUE, APREQ;
SEIZE: APRER;
ASSIGN: TIMEAPRE=OptTime*SipBuy;
DELAY: TIMEAPRE;
RELEASE: APRER;
ASSIGN: TTIME=TTIME-TIMEAPRE;
ASSIGN: STRM=DueDate-TNOW-TTIME;
ASSIGN: NUMOPR=NUMOPR-1;
ASSIGN: STRMOP=STRM/NUMOPR;
ROUTE;

STATION, SANFOR;
QUEUE, SANFORQ;
SEIZE: SANFORR;
ASSIGN: TIMESANFOR=OptTime*SipBuy;
DELAY: TIMESANFOR;

RELEASE: SANFORR;
ASSIGN: TTIME=TTIME-TIMESANFOR;
ASSIGN: STRM=DueDate-TNOW-TTIME;
ASSIGN: NUMOPR=NUMOPR-1;
ASSIGN: STRMOP=STRM/NUMOPR;
ROUTE;

STATION, FIKSE;
QUEUE, FIKSEQ;
SEIZE: FIKSER;
ASSIGN: TIMEFIKSE=OptTime*SipBuy;
DELAY: TIMEFIKSE;
RELEASE: FIKSER;
ASSIGN: TTIME=TTIME-TIMEFIKSE;
ASSIGN: STRM=DueDate-TNOW-TTIME;
ASSIGN: NUMOPR=NUMOPR-1;
ASSIGN: STRMOP=STRM/NUMOPR;
ROUTE;

STATION, JETBOYAMA;
QUEUE, JETBOYAMAQ;
SEIZE: JETBOYAMAR;
assign: acttypejetboyama=renktonu;
ASSIGN: TIMEJETBOYAMA=OptTime*SipBuy/1500+
 TF(pretypejetboyama, acttypejetboyama);
DELAY: TIMEJETBOYAMA;
assign: pretypejetboyama=acttypejetboyama;
assign: renklibeyaz=2;
RELEASE: JETBOYAMAR;
ASSIGN: TTIME=TTIME-TIMEJETBOYAMA;
ASSIGN: STRM=DueDate-TNOW-TTIME;
ASSIGN: NUMOPR=NUMOPR-1;
ASSIGN: STRMOP=STRM/NUMOPR;
ROUTE;

STATION, YAKMA;
QUEUE, YAKMAQ;
SEIZE: YAKMAR;
ASSIGN: TIMEYAKMA=OptTime*SipBuy;
DELAY: TIMEYAKMA;
RELEASE: YAKMAR;
ASSIGN: TTIME=TTIME-TIMEYAKMA;
ASSIGN: STRM=DueDate-TNOW-TTIME;
ASSIGN: NUMOPR=NUMOPR-1;
ASSIGN: STRMOP=STRM/NUMOPR;
ROUTE;

STATION, BASKI;
QUEUE, BASKIQ;
SEIZE: BASKIR;
ASSIGN: TIMEBASKI=OptTime*SipBuy;
DELAY: TIMEBASKI;
RELEASE: BASKIR;
ASSIGN: TTIME=TTIME-TIMEBASKI;
ASSIGN: STRM=DueDate-TNOW-TTIME;
ASSIGN: NUMOPR=NUMOPR-1;

ASSIGN:STRMOP=STRM/NUMOPR;
ROUTE;

STATION,KONDANSE;
QUEUE,KONDANSEQ;
SEIZE:KONDANSER;
ASSIGN:TIMEKONDANSE=OptTime*SipBuy;
DELAY:TIMEKONDANSE;
RELEASE:KONDANSER;
ASSIGN:TTIME=TTIME-TIMEKONDANSE;
ASSIGN:STRM=DueDate-TNOW-TTIME;
ASSIGN:NUMOPR=NUMOPR-1;
ASSIGN:STRMOP=STRM/NUMOPR;
ROUTE;

STATION,KATLAMA;
QUEUE,KATLAMAQ;
SEIZE:KATLAMAR;
RELEASE:KATLAMAR;

ASSIGN:GECIKME=DUEDATE - TNOW;

IF: GECIKME<0;
COUNT:GECKALANLAR;
ASSIGN:TOPLAMGECIKME=TOPLAMGECIKME+ABS (GECIKME) ;
ELSE;
COUNT:ERKENBITENLER;
ASSIGN:TOPLAMERKEN=TOPLAMERKEN+ABS (GECIKME) ;
ENDIF;

TALLY:FlowTime,INT (TimeIn) :DISPOSE;

create,1,simson:simson+1,1;
ASSIGN:ORTGECIKME=TOPLAMGECIKME/NC (GECKALANLAR) ;
ASSIGN:ORTERKENBIT=TOPLAMERKEN/NC (ERKENBITENLER) ;
WRITE, SONUCDOSYA:NC (GECKALANLAR),ORTGECIKME,
NC (ERKENBITENLER),ORTERKENBIT:dispose;

END;

S/OP kuralına ait simülasyon modelinin experiment dosyası

BEGIN;

PROJECT,gulsun S/RO;

VARIABLES:secim:

acttypeyikama:
 pretypeyikama,1:
acttypemerserize:
 pretypemerserize,1:
acttypeboyama:
 pretypeboyama,1:
acttypejetboyama:
 pretypejetboyama,1:
TOPLAMGECIKME,0:
TOPLAMERKEN,0:
ORTGECIKME:
ORTERKENBIT:
simson,40000;

ATTRIBUTES:TimeIn:SipBuy:OptTime:DueDate:STRM:TTIME:NUMOPR:STRMOP:

renklibeyaz:renktonu:
TIMEHASILSOKUMU:
TIMEBEKLETME:TIMEKOMBINKASAR:TIMEMERSERIZEILK:
TIMEMERSERIZESON:TIMEYIKAMAILK:TIMEYIKAMASON:
TIMEKURUTMA:TIMEZIMPARA:TIMEBOYAMA:TIMEAPRE:
TIMESANFOR:TIMEFIKSE:TIMEJETBOYAMA:
TIMEYAKMA:TIMEBASKI:TIMEKONDANSE:GECIKME;

COUNTERS:ENTNUM:

GECKALANLAR:
ERKENBITENLER;

QUEUES:HASILSOKUMUQ,LVF(STRMOP):

BEKLETMEQ,LVF(STRMOP):
KOMBINKASARQ,LVF(STRMOP):
MERSERIZEQ,LVF(STRMOP):
YIKAMAQ,LVF(STRMOP):
KURUTMAQ,LVF(STRMOP):
ZIMPARAQ,LVF(STRMOP):
BOYAMAQ,LVF(STRMOP):
APREQ,LVF(STRMOP):
SANFORQ,LVF(STRMOP):
FIKSEQ,LVF(STRMOP):
JETBOYAMAQ,LVF(STRMOP):
YAKMAQ,LVF(STRMOP):
BASKIQ,LVF(STRMOP):
KONDANSEQ,LVF(STRMOP):
KATLAMAQ,LVF(STRMOP);

RESOURCES:HASILSOKUMUR:

BEKLETMER,60:
KOMBINKASARR:
MERSERIZER:

YIKAMAR, 4:
KURUTMAR, 3:
ZIMPARAR:
BOYAMAR, 3:
APRER, 2:
SANFORR, 2:
FIKSER:
JETBOYAMAR, 5:
YAKMAR:
BASKIR:
KONDANSER:
KATLAMAR;

STATIONS:HASILSOKUMU:

BEKLETME:
KOMBINKASAR:
MERSERIZE:
YIKAMA:
KURUTMA:
ZIMPARA:
BOYAMA:
APRE:
SANFOR:
FIKSE:
JETBOYAMA:
YAKMA:
BASKI:
KONDANSE:
KATLAMA;

SEQUENCES:1, PAM,
HASILSOKUMU, OptTime=0.00033 & BEKLETME, OptTime=12 &
KOMBINKASAR, OptTime=0.00080 & MERSERIZE, OptTime=0.00033 &
YIKAMA, OptTime=0.00059 & KURUTMA, OptTime=0.00050 &
ZIMPARA, OptTime=0.00166 & YIKAMA, OptTime=0.00058 &
KURUTMA, OptTime=0.00060 & BOYAMA, OptTime=0.00235 &
BEKLETME, OptTime=24 & YIKAMA, OptTime=0.00095 &
KURUTMA, OptTime=0.00088 & APRE, OptTime=0.00078 &
SANFOR, OptTime=0.00060 & KATLAMA:

2, KORD,
KOMBINKASAR, OptTime=0.00080 & YIKAMA, OptTime=0.00058 &
KURUTMA, OptTime=0.00060 & BOYAMA, OptTime=0.00250 &
BEKLETME, OptTime=24 & YIKAMA, OptTime=0.00195 &
KURUTMA, OptTime=0.00088 & APRE, OptTime=0.00078 &
SANFOR, OptTime=0.00060 & KATLAMA:

3, JET,
YIKAMA, OptTime=0.00062 & KURUTMA, OptTime=0.00043 &
YAKMA, OptTime=0.00062 & FIKSE, OptTime=0.00066 &
JETBOYAMA, OptTime=5.40 & KURUTMA, OptTime=0.00085 &
APRE, OptTime=0.00061 & SANFOR, OptTime=0.00062 &
KATLAMA:

4, BASKILI,
HASILSOKUMU, OptTime=0.00033 & BEKLETME, OptTime=12 &

KOMBINKASAR,OptTime=0.00066 & MERSEIZE,OptTime=0.00033 &
YIKAMA,OptTime=0.00063 & KURUTMA,OptTime=0.00046 &
BASKI,OptTime=0.00435 & KONDANSE,OptTime=0.0033 &
APRE,OptTime=0.00052 & SANFOR,OptTime=0.00048 &
KATLAMA:

5, PEVIS,
HASILSOKUMU,OptTime=0.00043 & BEKLETME,OptTime=12 &
YIKAMA,OptTime=0.00058 & KURUTMA,OptTime=0.00041 &
BOYAMA,OptTime=0.00215 & KURUTMA,OptTime=0.00045 &
APRE,OptTime=0.00050 & KATLAMA:

6, GOMLEK,
YIKAMA,OptTime=0.00055 & KURUTMA,OptTime=0.00041 &
YAKMA,OptTime=0.00062 & MERSEIZE,OptTime=0.00033 &
KURUTMA,OptTime=0.00041 & APRE,OptTime=0.00065 &
SANFOR,OptTime=0.00052 & KATLAMA:

7, PERILEN,
YIKAMA,OptTime=0.00037 & FIKSE,OptTime=0.00058 &
APRE,OptTime=0.00095 & SANFOR,OptTime=0.00041 &
KATLAMA;

TALLIES:FlowTime;

DSTATS:NQ(HASILSOKUMUQ)*SipBuy/1000:
NQ(BEKLETMEQ)*SipBuy/1000:
NQ(KOMBINKASARQ)*SipBuy/1000:
NQ(MERSEIZEQ)*SipBuy/1000:
NQ(YIKAMAQ)*SipBuy/1000:
NQ(KURUTMAQ)*SipBuy/1000:
NQ(ZIMPARAQ)*SipBuy/1000:
NQ(BOYAMAQ)*SipBuy/1000:
NQ(APREQ)*SipBuy/1000:
NQ(SANFORQ)*SipBuy/1000:
NQ(FIKSEQ)*SipBuy/1000:
NQ(JETBOYAMAQ)*SipBuy/1000:
NQ(YAKMAQ)*SipBuy/1000:
NQ(BASKIQ)*SipBuy/1000:
NQ(KONDANSEQ)*SipBuy/1000:

NQ(HASILSOKUMUQ)*SipBuy/1000+ NQ(BEKLETMEQ)*SipBuy/1000+
NQ(KOMBINKASARQ)*SipBuy/1000+ NQ(MERSEIZEQ)*SipBuy/1000+
NQ(YIKAMAQ)*SipBuy/1000+ NQ(KURUTMAQ)*SipBuy/1000+
NQ(ZIMPARAQ)*SipBuy/1000+ NQ(BOYAMAQ)*SipBuy/1000+
NQ(APREQ)*SipBuy/1000+ NQ(SANFORQ)*SipBuy/1000+
NQ(FIKSEQ)*SipBuy/1000+ NQ(JETBOYAMAQ)*SipBuy/1000+
NQ(YAKMAQ)*SipBuy/1000+ NQ(BASKIQ)*SipBuy/1000+
NQ(KONDANSEQ)*SipBuy/1000,TOTALWIP:

NR(HASILSOKUMUR)*100:
NR(BEKLETMER)/60*100:
NR(KOMBINKASARR)*100:
NR(MERSEIZER)*100:
NR(YIKAMAR)/4*100:

```
NR(KURUTMAR)/3*100:
NR(ZIMPARAR)*100:
NR(BOYAMAR)/3*100:
NR(APRER)/2*100:
NR(SANFORR)/2*100:
NR(FIKSER)*100:
NR(JETBOYAMAR)/5*100:
NR(YAKMAR)*100:
NR(BASKIR)*100:
NR(KONDANSER)*100;
```

```
FILES:SONUCDOSYA,"disp25(2.2).DAT",SEQ,WKS,DIS;
```

```
TABLES: 1, 1,1, 0,0,0:
        2, 1,1, 2,0,2:
        3, 1,1, 3,2,0;
```

```
REPLICATE,5,0,40000,,,5000;
```

```
END;
```

EK 3 DARBOĞAZ KAYNAKLARIN TANIMLANMASI

Darboğaz Kaynağın Tespiti İçin Simülasyon Modeli

```
BEGIN;

CREATE:exponential(2.2,1):MARK(TimeIn);
COUNT:ENTNUM;

ASSIGN:ENBKUYRUK=TOPLAMTIMEHASILSOKUMU;
ASSIGN:KUYRUKSIRASI=1;

ASSIGN:HTIMEBEKLETME=TOPLAMTIMEBEKLETME/60;
ASSIGN:HTIMEYIKAMA=TOPLAMTIMEYIKAMA/4;
ASSIGN:HTIMEKURUTMA=TOPLAMTIMEKURUTMA/3;
ASSIGN:HTIMEBOYAMA=TOPLAMTIMEBOYAMA/2;
ASSIGN:HTIMEAPRE=TOPLAMTIMEAPRE/2;
ASSIGN:HTIMESANFOR=TOPLAMTIMESANFOR/2;
ASSIGN:HTIMEJETBOYAMA=TOPLAMTIMEJETBOYAMA/5;

IF:
    HTIMEBEKLETME.GE.ENBKUYRUK;
    ASSIGN:ENBKUYRUK=HTIMEBEKLETME;
    ASSIGN:KUYRUKSIRASI=2;
ENDIF;
IF:
    TOPLAMTIMEKOMBINKASAR.GE.ENBKUYRUK;
    ASSIGN:ENBKUYRUK=TOPLAMTIMEKOMBINKASAR;
    ASSIGN:KUYRUKSIRASI=3;
ENDIF;
IF:
    TOPLAMTIMEMERSERIZE.GE.ENBKUYRUK;
    ASSIGN:ENBKUYRUK=TOPLAMTIMEMERSERIZE;
    ASSIGN:KUYRUKSIRASI=4;
ENDIF;
IF:
    HTIMEYIKAMA.GE.ENBKUYRUK;
    ASSIGN:ENBKUYRUK=HTIMEYIKAMA;
    ASSIGN:KUYRUKSIRASI=5;
ENDIF;
IF:
    HTIMEKURUTMA.GE.ENBKUYRUK;
    ASSIGN:ENBKUYRUK=HTIMEKURUTMA;
    ASSIGN:KUYRUKSIRASI=6;
ENDIF;
IF:
    TOPLAMTIMEZIMPARA.GE.ENBKUYRUK;
    ASSIGN:ENBKUYRUK=TOPLAMTIMEZIMPARA;
    ASSIGN:KUYRUKSIRASI=7;
ENDIF;
IF:
    HTIMEBOYAMA.GE.ENBKUYRUK;
    ASSIGN:ENBKUYRUK=HTIMEBOYAMA;
    ASSIGN:KUYRUKSIRASI=8;
ENDIF;
IF:
    HTIMEAPRE.GE.ENBKUYRUK;
    ASSIGN:ENBKUYRUK=HTIMEAPRE;
    ASSIGN:KUYRUKSIRASI=9;
ENDIF;
```

```

IF:      HTIMESANFOR.GE.ENBKUYRUK;
        ASSIGN:ENBKUYRUK=HTIMESANFOR;
        ASSIGN:KUYRUKSIRASI=10;
ENDIF;
IF:      TOPLAMTIMEFIKSE.GE.ENBKUYRUK;
        ASSIGN:ENBKUYRUK=TOPLAMTIMEFIKSE;
        ASSIGN:KUYRUKSIRASI=11;
ENDIF;
IF:      HTIMEJETBOYAMA.GE.ENBKUYRUK;
        ASSIGN:ENBKUYRUK=HTIMEJETBOYAMA;
        ASSIGN:KUYRUKSIRASI=12;
ENDIF;
IF:      TOPLAMTIMEYAKMA.GE.ENBKUYRUK;
        ASSIGN:ENBKUYRUK=TOPLAMTIMEYAKMA;
        ASSIGN:KUYRUKSIRASI=13;
ENDIF;
IF:      TOPLAMTIMEBASKI.GE.ENBKUYRUK;
        ASSIGN:ENBKUYRUK=TOPLAMTIMEBASKI;
        ASSIGN:KUYRUKSIRASI=14;
ENDIF;
IF:      TOPLAMTIMEKONDANSE.GE.ENBKUYRUK;
        ASSIGN:ENBKUYRUK=TOPLAMTIMEKONDANSE;
        ASSIGN:KUYRUKSIRASI=15;
ENDIF;
IF:      TOPLAMTIMEKATLAMA.GE.ENBKUYRUK;
        ASSIGN:ENBKUYRUK=TOPLAMTIMEKATLAMA;
        ASSIGN:KUYRUKSIRASI=16;
ENDIF;

IF:      KUYRUKSIRASI==1;
        COUNT:HASILSOKUMUC;
ELSEIF:  KUYRUKSIRASI==2;
        COUNT:BEKLETMEC;
ELSEIF:  KUYRUKSIRASI==3;
        COUNT:KOMBINKASARC;
ELSEIF:  KUYRUKSIRASI==4;
        COUNT:MERSERIZEC;
ELSEIF:  KUYRUKSIRASI==5;
        COUNT:YIKAMAC;
ELSEIF:  KUYRUKSIRASI==6;
        COUNT:KURUTMAC;
ELSEIF:  KUYRUKSIRASI==7;
        COUNT:ZIMPARAC;
ELSEIF:  KUYRUKSIRASI==8;
        COUNT:BOYAMAC;
ELSEIF:  KUYRUKSIRASI==9;
        COUNT:APREC;
ELSEIF:  KUYRUKSIRASI==10;
        COUNT:SANFORC;
ELSEIF:  KUYRUKSIRASI==11;
        COUNT:FIKSEC;
ELSEIF:  KUYRUKSIRASI==12;
        COUNT:JETBOYAMAC;
ELSEIF:  KUYRUKSIRASI==13;
        COUNT:YAKMAC;

```

```

ELSEIF: KUYRUKSIRASI==14;
        COUNT:BASKIC;
ELSEIF: KUYRUKSIRASI==15;
        COUNT:KONDANSEC;
ELSEIF: KUYRUKSIRASI==16;
        COUNT:KATLAMAC;
ENDIF;

```

```

ASSIGN:NS=discrete(0.18,1,0.34,2,0.57,3,0.65,4,
                  0.74,5,0.80,6,1.0,7);
SipBuy=discrete(0.2,2500,0.6,4000,0.85,6000,0.95,10000,
                1.0,25000);

```

```

BRANCH,1:WITH,0.3,export:
        WITH,0.7,yerli;

```

```

export  ASSIGN:DueDate=TNOW+UNIF(336,504):NEXT(PA1);
yerli   ASSIGN:DueDate=TNOW+UNIF(504,672);

```

```

PA1 branch, 1:
        if,NS==1,PA2:
        if,NS==2,PA2:
        if,NS==3,PA2:
        if,NS==4,PA3:
        if,NS==5,PA2:
        if,NS==6,PA3:
        else, PA3;

```

```

PA2  assign:renklibeyaz=1;
PA21 assign:renktonu=discrete(0.3,1,0.6,2,1.0,3):NEXT(PA4);

```

```

PA3  assign:renklibeyaz=2;
PA31 assign:renktonu=1;

```

```

PA4  ROUTE;

```

```

STATION,HASILSOKUMU;
ASSIGN:TOPLAMTIMEHASILSOKUMU=TOPLAMTIMEHASILSOKUMU+
        OptTime*SipBuy;
QUEUE,HASILSOKUMUQ;
SEIZE:HASILSOKUMUR;
ASSIGN:TIMEHASILSOKUMU=OptTime*SipBuy;
DELAY:TIMEHASILSOKUMU;
RELEASE:HASILSOKUMUR;
ASSIGN:TOPLAMTIMEHASILSOKUMU=TOPLAMTIMEHASILSOKUMU-
        OptTime*SipBuy;
ROUTE;

```

```

STATION,BEKLETME;
ASSIGN:TOPLAMTIMEBEKLETME=TOPLAMTIMEBEKLETME+OptTime;
QUEUE,BEKLETMEQ;

```

```

SEIZE:BEKLETMER;
ASSIGN:TIMEBEKLETME=OptTime;
DELAY:TIMEBEKLETME;
RELEASE:BEKLETMER;
ASSIGN:TOPLAMTIMEBEKLETME=TOPLAMTIMEBEKLETME-OptTime;
ROUTE;

```

```

STATION,KOMBINKASAR;
ASSIGN:TOPLAMTIMEKOMBINKASAR=TOPLAMTIMEKOMBINKASAR+
    OptTime*SipBuy;
QUEUE,KOMBINKASARQ;
SEIZE:KOMBINKASARR;
ASSIGN:TIMEKOMBINKASAR=OptTime*SipBuy;
DELAY:TIMEKOMBINKASAR;
RELEASE:KOMBINKASARR;
ASSIGN:TOPLAMTIMEKOMBINKASAR=TOPLAMTIMEKOMBINKASAR-
    OptTime*SipBuy;
ROUTE;

```

```

STATION,MERSERIZE;
ASSIGN:TOPLAMTIMEMERSERIZE=TOPLAMTIMEMERSERIZE+
    OptTime*SipBuy;
QUEUE,MERSERIZEQ;
SEIZE:MERSERIZER;

```

```

assign:acttypemerserize=renklibeyaz;

```

```

PA5    ASSIGN:TIMEMERSERIZEILK=OptTime*SipBuy;

```

```

branch,1:
    if, acttypemerserize.NE.pretypemerserize, PA51:
    else, PA6;

```

```

PA51    ASSIGN:TIMEMERSERIZESON=TIMEMERSERIZEILK+4:NEXT(PA61);

```

```

PA6     ASSIGN:TIMEMERSERIZESON=TIMEMERSERIZEILK;

```

```

PA61    DELAY:TIMEMERSERIZESON;
assign:pretypemerserize=acttypemerserize;

```

```

RELEASE:MERSERIZER;
ASSIGN:TOPLAMTIMEMERSERIZE=TOPLAMTIMEMERSERIZE-
    OptTime*SipBuy;
ROUTE;

```

```

STATION,YIKAMA;
ASSIGN:TOPLAMTIMEYIKAMA=TOPLAMTIMEYIKAMA+OptTime*SipBuy;
QUEUE,YIKAMAQ;
SEIZE:YIKAMAR;
assign:acttypeyikama=renklibeyaz;

```

```

PA7     ASSIGN:TIMEYIKAMAILK=CptTime*SipBuy;

```

```

branch,1:
    if, acttypeyikama.NE.prettypeyikama, PA71:
    else, PA8;

```

```

PA71    ASSIGN:TIMEYIKAMASON=TIMEYIKAMAILK+2:NEXT(PA81);

PA8      ASSIGN:TIMEYIKAMASON=TIMEYIKAMAILK;

PA81     DELAY:TIMEYIKAMASON;
          assign:pretypeyikama=acttypeyikama;

RELEASE:YIKAMAR;
ASSIGN:TOPLAMTIMEYIKAMA=TOPLAMTIMEYIKAMA-OptTime*SipBuy;
ROUTE;

STATION,KURUTMA;
ASSIGN:TOPLAMTIMEKURUTMA=TOPLAMTIMEKURUTMA+
      OptTime*SipBuy;
QUEUE,KURUTMAQ;
SEIZE:KURUTMAR;
ASSIGN:TIMEKURUTMA=OptTime*SipBuy;
DELAY:TIMEKURUTMA;
RELEASE:KURUTMAR;
ASSIGN:TOPLAMTIMEKURUTMA=TOPLAMTIMEKURUTMA-OptTime*SipBuy;
ROUTE;

STATION,ZIMPARA;
ASSIGN:TOPLAMTIMEZIMPARA=TOPLAMTIMEZIMPARA+
      OptTime*SipBuy;
QUEUE,ZIMPARAQ;
SEIZE:ZIMPARAR;
ASSIGN:TIMEZIMPARA=OptTime*SipBuy;
DELAY:TIMEZIMPARA;
RELEASE:ZIMPARAR;
ASSIGN:TOPLAMTIMEZIMPARA=TOPLAMTIMEZIMPARA-OptTime*SipBuy;
ROUTE;

STATION,BOYAMA;
ASSIGN:TOPLAMTIMEBOYAMA=TOPLAMTIMEBOYAMA+OptTime*SipBuy;
QUEUE,BOYAMAQ;
SEIZE:BOYAMAR;
assign:acttypeboyama=renktonu;
ASSIGN:TIMEBOYAMA=OptTime*SipBuy+TF(pretypeboyama, acttypeboyama);
DELAY:TIMEBOYAMA;
assign:pretypeboyama=acttypeboyama;
assign:renklibeyaz=2;
RELEASE:BOYAMAR;
ASSIGN:TOPLAMTIMEBOYAMA=TOPLAMTIMEBOYAMA-OptTime*SipBuy;
ROUTE;

STATION,APRE;
ASSIGN:TOPLAMTIMEAPRE=TOPLAMTIMEAPRE+OptTime*SipBuy;
QUEUE,APREQ;
SEIZE:APRER;
ASSIGN:TIMEAPRE=OptTime*SipBuy;
DELAY:TIMEAPRE;
RELEASE:APRER;
ASSIGN:TOPLAMTIMEAPRE=TOPLAMTIMEAPRE-OptTime*SipBuy;
ROUTE;

```

STATION, SANFOR;
ASSIGN: TOPLAMTIMESANFOR=TOPLAMTIMESANFOR+OptTime*SipBuy;
QUEUE, SANFORQ;
SEIZE: SANFORR;
ASSIGN: TIMESANFOR=OptTime*SipBuy;
DELAY: TIMESANFOR;
RELEASE: SANFORR;
ASSIGN: TOPLAMTIMESANFOR=TOPLAMTIMESANFOR-OptTime*SipBuy;
ROUTE;

STATION, FIKSE;
ASSIGN: TOPLAMTIMEFIKSE=TOPLAMTIMEFIKSE+OptTime*SipBuy;
QUEUE, FIKSEQ;
SEIZE: FIKSER;
ASSIGN: TIMEFIKSE=OptTime*SipBuy;
DELAY: TIMEFIKSE;
RELEASE: FIKSER;
ASSIGN: TOPLAMTIMEFIKSE=TOPLAMTIMEFIKSE-OptTime*SipBuy;
ROUTE;

STATION, JETBOYAMA;
ASSIGN: TOPLAMTIMEJETBOYAMA=TOPLAMTIMEJETBOYAMA+OptTime*SipBuy/1500;
QUEUE, JETBOYAMAQ;
SEIZE: JETBOYAMAR;
assign: acttypejetboyama=renktonu;
ASSIGN: TIMEJETBOYAMA=OptTime*SipBuy/1500+
TF(pretypejetboyama, acttypejetboyama);
DELAY: TIMEJETBOYAMA;
assign: pretypejetboyama=acttypejetboyama;
assign: renklibeyaz=2;
RELEASE: JETBOYAMAR;
ASSIGN: TOPLAMTIMEJETBOYAMA=TOPLAMTIMEJETBOYAMA-OptTime*SipBuy/1500;
ROUTE;

STATION, YAKMA;
ASSIGN: TOPLAMTIMEYAKMA=TOPLAMTIMEYAKMA+OptTime*SipBuy;
QUEUE, YAKMAQ;
SEIZE: YAKMAR;
ASSIGN: TIMEYAKMA=OptTime*SipBuy;
DELAY: TIMEYAKMA;
RELEASE: YAKMAR;
ASSIGN: TOPLAMTIMEYAKMA=TOPLAMTIMEYAKMA-OptTime*SipBuy;
ROUTE;

STATION, BASKI;
ASSIGN: TOPLAMTIMEBASKI=TOPLAMTIMEBASKI+OptTime*SipBuy;
QUEUE, BASKIQ;
SEIZE: BASKIR;
ASSIGN: TIMEBASKI=OptTime*SipBuy;
DELAY: TIMEBASKI;
RELEASE: BASKIR;
ASSIGN: TOPLAMTIMEBASKI=TOPLAMTIMEBASKI-OptTime*SipBuy;
ROUTE;

```

STATION, KONDANSE;
ASSIGN: TOPLAMTIMEKONDANSE=TOPLAMTIMEKONDANSE+
      OptTime*SipBuy;
QUEUE, KONDANSEQ;
SEIZE: KONDANSER;
ASSIGN: TIMEKONDANSE=OptTime*SipBuy;
DELAY: TIMEKONDANSE;
RELEASE: KONDANSER;
ASSIGN: TOPLAMTIMEKONDANSE=TOPLAMTIMEKONDANSE-OptTime*SipBuy;
ROUTE;

STATION, KATLAMA;
QUEUE, KATLAMAQ;
SEIZE: KATLAMAR;
RELEASE: KATLAMAR;

ASSIGN: GECIKME=DUE DATE - TNOW;

IF:      GECIKME<0;
        COUNT: GECKALANLAR;
        ASSIGN: TOPLAMGECIKME=TOPLAMGECIKME+ABS (GECIKME) ;
ELSE;
        COUNT: ERKENBITENLER;
        ASSIGN: TOPLAMERKEN=TOPLAMERKEN+ABS (GECIKME) ;
ENDIF;

TALLY: FlowTime, INT (TimeIn) :DISPOSE;

CREATE, 1, SIMSON: SIMSON+1, 1;
        ASSIGN: ORTGECIKME=TOPLAMGECIKME/NC (GECKALANLAR) ;
        ASSIGN: ORTERKENBIT=TOPLAMERKEN/NC (ERKENBITENLER) ;
        WRITE, SONUCDOSYA: NC (GECKALANLAR) , ORTGECIKME,
              NC (ERKENBITENLER) , ORTERKENBIT:DISPOSE;

END;

```

Darboğaz Tespitine Ait Simülasyon Modelinin Experiment Dosyası

BEGIN;

PROJECT,gulsun, darboğaz tespiti;

VARIABLES:secim:

acttypeyikama:
pretypeyikama,1:
acttypemerserize:
pretypemerserize,1:
acttypeboyama:
pretypeboyama,1:
acttypejetboyama:
pretypejetboyama,1:

SIMSON:

ENBKUYRUK:

KUYRUKSIRASI:

TOPLAMGECIKME,0:

TOPLAMERKEN,0:

ORTGECIKME:

ORTERKENBIT:

TOPLAMTIMEHASILSOKUMU,0:

TOPLAMTIMEBEKLETME,0:

TOPLAMTIMEKOMBINKASAR,0:

TOPLAMTIMEMERSERIZE,0:

TOPLAMTIMEYIKAMA,0:

TOPLAMTIMEKURUTMA,0:

TOPLAMTIMEZIMPARA,0:

TOPLAMTIMEBOYAMA,0:

TOPLAMTIMEAPRE,0:

TOPLAMTIMESANFOR,0:

TOPLAMTIMEFIKSE,0:

TOPLAMTIMEJETBOYAMA,0:

TOPLAMTIMEYAKMA,0:

TOPLAMTIMEBASKI,0:

TOPLAMTIMEKONDANSE,0:

TOPLAMTIMEKATLAMA,0:

HTIMEBEKLETME,0:

HTIMEYIKAMA,0:

HTIMEKURUTMA,0:

HTIMEBOYAMA,0:

HTIMEAPRE,0:

HTIMESANFOR,0:

HTIMEJETBOYAMA,0;

ATTRIBUTES:TimeIn:SipBuy:OptTime:DueDate:renklibeyaz:renk

tonu:

TIMEHASILSOKUMU:

TIMEBEKLETME:TIMEKOMBINKASAR:TIMEMERSERIZEILK:

TIMEMERSERIZESON:TIMEYIKAMAILK:TIMEYIKAMASON:

TIMEKURUTMA:TIMEZIMPARA:TIMEBOYAMA:TIMEAPRE:

TIMESANFOR:TIMEFIKSE:TIMEJETBOYAMA:

TIMEYAKMA:TIMEBASKI:TIMEKONDANSE:GECIKME;

COUNTERS:ENTNUM:

GECKALANLAR:
ERKENBITENLER:
HASILSOKUMUC:
BEKLETMEC:
KOMBINKASARC:
MERSERIZEC:
YIKAMAC:
KURUTMAC:
ZIMPARAC:
BOYAMAC:
APREC:
SANFORC:
FIKSEC:
JETBOYAMAC:
YAKMAC:
BASKIC:
KONDANSEC:
KATLAMAC;

QUEUES:HASILSOKUMUQ:

BEKLETMEQ:
KOMBINKASARQ:
MERSERIZEQ:
YIKAMAQ:
KURUTMAQ:
ZIMPARAQ:
BOYAMAQ:
APREQ:
SANFORQ:
FIKSEQ:
JETBOYAMAQ:
YAKMAQ:
BASKIQ:
KONDANSEQ:
KATLAMAQ;

RESOURCES:HASILSOKUMUR:

BEKLETMER, 60:
KOMBINKASARR:
MERSERIZER:
YIKAMAR, 4:
KURUTMAR, 3:
ZIMPARAR:
BOYAMAR, 3:
APRER, 2:
SANFORR, 2:
FIKSER:
JETBOYAMAR, 5:
YAKMAR:
BASKIR:
KONDANSER:
KATLAMAR;

STATIONS:HASILSOKUMU:

BEKLETME:
KOMBINKASAR:
MERSERIZE:
YIKAMA:
KURUTMA:
ZIMPARA:
BOYAMA:
APRE:
SANFOR:
FIKSE:
JETBOYAMA:
YAKMA:
BASKI:
KONDANSE:
KATLAMA;

SEQUENCES:1, PAM,

HASILSOKUMU,OptTime=0.00033 & BEKLETME,OptTime=12 &
KOMBINKASAR,OptTime=0.00080 & MERSERIZE,OptTime=0.00033 &
YIKAMA,OptTime=0.00059 & KURUTMA,OptTime=0.00050 &
ZIMPARA,OptTime=0.00166 & YIKAMA,OptTime=0.00058 &
KURUTMA,OptTime=0.00060 & BOYAMA,OptTime=0.00235 &
BEKLETME,OptTime=24 & YIKAMA,OptTime=0.00095 &
KURUTMA,OptTime=0.00088 & APRE,OptTime=0.00078 &
SANFOR,OptTime=0.00060 & KATLAMA:

2, KORD,

KOMBINKASAR,OptTime=0.00080 & YIKAMA,OptTime=0.00058 &
KURUTMA,OptTime=0.00060 & BOYAMA,OptTime=0.00250 &
BEKLETME,OptTime=24 & YIKAMA,OptTime=0.00195 &
KURUTMA,OptTime=0.00088 & APRE,OptTime=0.00078 &
SANFOR,OptTime=0.00060 & KATLAMA:

3, JET,

YIKAMA,OptTime=0.00062 & KURUTMA,OptTime=0.00048 &
YAKMA,OptTime=0.00062 & FIKSE,OptTime=0.00066 &
JETBOYAMA,OptTime=5.40 & KURUTMA,OptTime=0.00085 &
APRE,OptTime=0.00061 & SANFOR,OptTime=0.00062 &
KATLAMA:

4, BASKILI,

HASILSOKUMU,OptTime=0.00033 & BEKLETME,OptTime=12 &
KOMBINKASAR,OptTime=0.00066 & MERSERIZE,OptTime=0.00033 &
YIKAMA,OptTime=0.00063 & KURUTMA,OptTime=0.00046 &
BASKI,OptTime=0.00435 & KONDANSE,OptTime=0.0033 &
APRE,OptTime=0.00052 & SANFOR,OptTime=0.00048 &
KATLAMA:

5, PEVIS,

HASILSOKUMU,OptTime=0.00043 & BEKLETME,OptTime=12 &
YIKAMA,OptTime=0.00058 & KURUTMA,OptTime=0.00041 &
BOYAMA,OptTime=0.00215 & KURUTMA,OptTime=0.00045 &
APRE,OptTime=0.00050 & KATLAMA:

6, GOMLEK,
 YIKAMA, OptTime=0.00055 & KURUTMA, OptTime=0.00041 &
 YAKMA, OptTime=0.00062 & MERSEIZE, OptTime=0.00033 &
 KURUTMA, OptTime=0.00041 & APRE, OptTime=0.00065 &
 SANFOR, OptTime=0.00052 & KATLAMA:

7, PERILEN,
 YIKAMA, OptTime=0.00037 & FIKSE, OptTime=0.00058 &
 APRE, OptTime=0.00095 & SANFOR, OptTime=0.00041 &
 KATLAMA;

TALLIES: FlowTime;

DSTATS: NQ(HASILSOKUMUQ)*SipBuy/1000:
 NQ(BEKLETMEQ)*SipBuy/1000:
 NQ(KOMBINKASARQ)*SipBuy/1000:
 NQ(MERSEIZEQ)*SipBuy/1000:
 NQ(YIKAMAQ)*SipBuy/1000:
 NQ(KURUTMAQ)*SipBuy/1000:
 NQ(ZIMPARAQ)*SipBuy/1000:
 NQ(BOYAMAQ)*SipBuy/1000:
 NQ(APREQ)*SipBuy/1000:
 NQ(SANFORQ)*SipBuy/1000:
 NQ(FIKSEQ)*SipBuy/1000:
 NQ(JETBOYAMAQ)*SipBuy/1000:
 NQ(YAKMAQ)*SipBuy/1000:
 NQ(BASKIQ)*SipBuy/1000:
 NQ(KONDANSEQ)*SipBuy/1000:

NQ(HASILSOKUMUQ)*SipBuy/1000+ NQ(BEKLETMEQ)*SipBuy/1000+
 NQ(KOMBINKASARQ)*SipBuy/1000+ NQ(MERSEIZEQ)*SipBuy/1000+
 NQ(YIKAMAQ)*SipBuy/1000+ NQ(KURUTMAQ)*SipBuy/1000+
 NQ(ZIMPARAQ)*SipBuy/1000+ NQ(BOYAMAQ)*SipBuy/1000+
 NQ(APREQ)*SipBuy/1000+ NQ(SANFORQ)*SipBuy/1000+
 NQ(FIKSEQ)*SipBuy/1000+ NQ(JETBOYAMAQ)*SipBuy/1000+
 NQ(YAKMAQ)*SipBuy/1000+ NQ(BASKIQ)*SipBuy/1000+
 NQ(KONDANSEQ)*SipBuy/1000, TOTALWIP:

NR(HASILSOKUMUR)*100:
 NR(BEKLETMER)/60*100:
 NR(KOMBINKASARR)*100:
 NR(MERSEIZER)*100:
 NR(YIKAMAR)/4*100:
 NR(KURUTMAR)/3*100:
 NR(ZIMPARAR)*100:
 NR(BOYAMAR)/3*100:
 NR(APRER)/2*100:
 NR(SANFORR)/2*100:
 NR(FIKSER)*100:
 NR(JETBOYAMAR)/5*100:
 NR(YAKMAR)*100:
 NR(BASKIR)*100:
 NR(KONDANSER)*100;

FILES: SONUCDOSYA, "BND2.DAT", SEQ, WKS, DIS;

```
TABLES: 1, 1,1, 0,0,0:  
        2, 1,1, 2,0,2:  
        3, 1,1, 3,2,0;
```

```
REPLICATE,5,0,40000,,,5000;
```

```
END;
```

```
Experiment compilation completed with 0 error(s) and 0 warning(s).
```



EK 4 ALGORİTMA 1 İÇİN SİMÜLASYON UYGULAMASI

1. Algoritmaya Ait Örnek Simülasyon Modeli

```
BEGIN;

CREATE:exponential(2.2,1):MARK(TimeIn);
COUNT:ENTNUM;

ASSIGN:NS=discrete(0.18,1,0.34,2,0.57,3,0.65,4,
                  0.74,5,0.80,6,1.0,7);
      SipBuy=discrete(0.2,2500,0.6,4000,0.85,6000,0.95,10000,
                  1.0,25000);

BRANCH,1:WITH,0.3,export:
      WITH,0.7,yerli;

export  ASSIGN:DueDate=TNOW+UNIF(336,504):NEXT(PA1);
yerli   ASSIGN:DueDate=TNOW+UNIF(504,672);

PA1 IF: NS==1;
      ASSIGN:APRETIME=SipBuy*0.00078;
      ASSIGN:SANFTIME=SipBuy*0.00060;
      ASSIGN:BOYAMATIME=SipBuy*0.00235:
      Next(PA2);
ELSEIF:
      NS==2;
      ASSIGN:APRETIME=SipBuy*0.00078;
      ASSIGN:SANFTIME=SipBuy*0.00060;
      ASSIGN:BOYAMATIME=SipBuy*0.00250:
      Next(PA2);
ELSEIF:
      NS==3;
      ASSIGN:APRETIME=SipBuy*0.00061;
      ASSIGN:SANFTIME=SipBuy*0.00062;
      ASSIGN:JETBOYAMATIME=SipBuy*0.00250/1500;
      ASSIGN:YAKMATIME=SipBuy*0.00062;
      ASSIGN:FIKSETIME=SipBuy*0.00066:
      Next(PA2);
ELSEIF:
      NS==4;
      ASSIGN:APRETIME=SipBuy*0.00052;
      ASSIGN:SANFTIME=SipBuy*0.00048;
      ASSIGN:BASKITIME=SipBuy*0.00435;
      ASSIGN:KONDANSETIME=SipBuy*0.0033:
      Next(PA3);
ELSEIF:
      NS==5;
      ASSIGN:APRETIME=SipBuy*0.00050;
      ASSIGN:BOYAMATIME=SipBuy*0.00215;
```

```

        Next (PA2) ;
ELSEIF:
    NS==6;
    ASSIGN:APRETIME=SipBuy*0.00065;
    ASSIGN:SANFTIME=SipBuy*0.00052;
    ASSIGN:YAKMATIME=SipBuy*0.00062;
    Next (PA3) ;
ELSE;
    ASSIGN:APRETIME=SipBuy*0.00095;
    ASSIGN:SANFTIME=SipBuy*0.00041;
    ASSIGN:FIKSETIME=SipBuy*0.00058;
    Next (PA3) ;
ENDIF;

PA2    assign:renklibeyaz=1;
PA21   assign:renktonu=discrete(0.3,1,0.6,2,1.0,3):NEXT(PA4);

PA3    assign:renklibeyaz=2;
PA31   assign:renktonu=1;

PA4    ROUTE;

STATION,HASILSOKUMU;
QHSS1  QUEUE,HASILSOKUMUQ;
        SEIZE:HASILSOKUMUR;

BRANCH, 2:
    ALWAYS, QHSS1, YES:
        IF, SIRALAMA==1,HSS1:
        IF, SIRALAMA==2,HSS2:
        IF, SIRALAMA==3,HSS3:
        IF, SIRALAMA==4,HSS4;
HSS1    SEARCH,HASILSOKUMUQ:MAX(BOYAMATIME):NEXT(QHSS2);
HSS2    SEARCH,HASILSOKUMUQ:MAX(BASKITIME):NEXT(QHSS2);
HSS3    SEARCH,HASILSOKUMUQ:MIN(BASKITIME):NEXT(QHSS2);
HSS4    SEARCH,HASILSOKUMUQ:MIN(BOYAMATIME);
QHSS2   REMOVE:J,HASILSOKUMUQ,QHSS3;
        RELEASE:HASILSOKUMUR;
        DELAY:0.0:DISPOSE;

QHSS3   SEIZE:HASILSOKUMUR;
        ASSIGN:TIMEHASILSOKUMU=OptTime*SipBuy;
        DELAY:TIMEHASILSOKUMU;
        RELEASE:HASILSOKUMUR;

ROUTE;

STATION,BEKLETME;
QBK1    QUEUE,BEKLETMEQ;
        SEIZE:BEKLETMER;

BRANCH, 2:
    ALWAYS, QBK1, YES:

```

```

        IF, SIRALAMA==1,BEK1:
        IF, SIRALAMA==2,BEK2:
        IF, SIRALAMA==3,BEK3:
        IF, SIRALAMA==4,BEK4;
BEK1    SEARCH,BEKLETMEQ:MAX(BOYAMATIME):NEXT(QBEK2);
BEK2    SEARCH,BEKLETMEQ:MAX(BASKITIME):NEXT(QBEK2);
BEK3    SEARCH,BEKLETMEQ:MIN(BASKITIME):NEXT(QBEK2);
BEK4    SEARCH,BEKLETMEQ:MIN(BOYAMATIME);
QBEK2   REMOVE:J,BEKLETMEQ,QBEK3;
        RELEASE:BEKLETMER;
        DELAY:0.0:DISPOSE;

QBEK3   SEIZE:BEKLETMER;
        ASSIGN:TIMEBEKLETME=OptTime;
        DELAY:TIMEBEKLETME;
        RELEASE:BEKLETMER;
ROUTE;

STATION,KOMBINKASAR;
QKKS1   QUEUE,KOMBINKASARQ;
        SEIZE:KOMBINKASARR;
BRANCH, 2:
        ALWAYS, QKKS1, YES:
        IF, SIRALAMA==1,KKS1:
        IF, SIRALAMA==2,KKS2:
        IF, SIRALAMA==3,KKS3:
        IF, SIRALAMA==4,KKS4;
KKS1    SEARCH,KOMBINKASARQ:MAX(BOYAMATIME):NEXT(QKKS2);
KKS2    SEARCH,KOMBINKASARQ:MAX(BASKITIME):NEXT(QKKS2);
KKS3    SEARCH,KOMBINKASARQ:MIN(BASKITIME):NEXT(QKKS2);
KKS4    SEARCH,KOMBINKASARQ:MIN(BOYAMATIME);
QKKS2   REMOVE:J,KOMBINKASARQ,QKKS3;
        RELEASE:KOMBINKASARR;
        DELAY:0.0:DISPOSE;

QKKS3   SEIZE:KOMBINKASARR;
        ASSIGN:TIMEKOMBINKASAR=OptTime*SipBuy;
        DELAY:TIMEKOMBINKASAR;
        RELEASE:KOMBINKASARR;
ROUTE;

STATION,MERSERIZE;
QMRS1   QUEUE,MERSERIZEQ;
        SEIZE:MERSERIZER;

BRANCH, 2:
        ALWAYS, QMRS1, YES:
        IF, SIRALAMA==1,MRS1:
        IF, SIRALAMA==2,MRS2:
        IF, SIRALAMA==3,MRS3:
        IF, SIRALAMA==4,MRS4;
MRS1    SEARCH,MERSERIZEQ:MAX(BOYAMATIME):NEXT(QMRS2);
MRS2    SEARCH,MERSERIZEQ:MAX(BASKITIME):NEXT(QMRS2);
MRS3    SEARCH,MERSERIZEQ:MIN(BASKITIME):NEXT(QMRS2);

```

```

MRS4      SEARCH,MERSERIZEQ:MIN(BOYAMATIME);
QMRS2     REMOVE:J,MERSERIZEQ,QMRS3;
          RELEASE:MERSERIZER;
          DELAY:0.0:DISPOSE;

QMRS3     SEIZE:MERSERIZER;
          assign:acttypemerserize=renklibeyaz;
PA5       ASSIGN:TIMEMERSERIZEILK=OptTime*SipBuy;
branch,1:
          if, acttypemerserize.NE.pretypemerserize, PA51:
          else, PA6;
PA51      ASSIGN:TIMEMERSERIZESON=TIMEMERSERIZEILK+4:NEXT(PA61);
PA6       ASSIGN:TIMEMERSERIZESON=TIMEMERSERIZEILK;
PA61      DELAY:TIMEMERSERIZESON;
          assign:pretypemerserize=acttypemerserize;
          RELEASE:MERSERIZER;

ROUTE;

```

```

STATION,YIKAMA;
QYKS1     QUEUE,YIKAMAQ;
          SEIZE:YIKAMAR;

```

```

BRANCH, 2:
          ALWAYS, QYKS1, YES:
          IF, SIRALAMA==1,YKS1:
          IF, SIRALAMA==2,YKS2:
          IF, SIRALAMA==3,YKS3:
          IF, SIRALAMA==4,YKS4;
YKS1      SEARCH,YIKAMAQ:MAX(BOYAMATIME):NEXT(QYKS2);
YKS2      SEARCH,YIKAMAQ:MAX(BASKITIME):NEXT(QYKS2);
YKS3      SEARCH,YIKAMAQ:MIN(BASKITIME):NEXT(QYKS2);
YKS4      SEARCH,YIKAMAQ:MIN(BOYAMATIME);
QYKS2     REMOVE:J,YIKAMAQ,QYKS3;
          RELEASE:YIKAMAR;
          DELAY:0.0:DISPOSE;

```

```

QYKS3     SEIZE:YIKAMAR;
          assign:acttypeyikama=renklibeyaz;
PA7       ASSIGN:TIMEYIKAMAILK=OptTime*SipBuy;
branch,1:
          if, acttypeyikama.NE.pretypeyikama, PA71:
          else, PA8;
PA71      ASSIGN:TIMEYIKAMASON=TIMEYIKAMAILK+2:NEXT(PA81);
PA8       ASSIGN:TIMEYIKAMASON=TIMEYIKAMAILK;
PA81      DELAY:TIMEYIKAMASON;
          assign:pretypeyikama=acttypeyikama;
          RELEASE:YIKAMAR;

ROUTE;

```

```

STATION,KURUTMA;
QKRS1     QUEUE,KURUTMAQ;
          SEIZE:KURUTMAR;

```

```

BRANCH, 2:
    ALWAYS, QKRS1, YES:
        IF, SIRALAMA==1, KRS1:
        IF, SIRALAMA==2, KRS2:
        IF, SIRALAMA==3, KRS3:
        IF, SIRALAMA==4, KRS4;
KRS1    SEARCH, KURUTMAQ:MAX(BOYAMATIME):NEXT(QKRS2);
KRS2    SEARCH, KURUTMAQ:MAX(BASKITIME):NEXT(QKRS2);
KRS3    SEARCH, KURUTMAQ:MIN(BASKITIME):NEXT(QKRS2);
KRS4    SEARCH, KURUTMAQ:MIN(BOYAMATIME);
QKRS2    REMOVE:J, KURUTMAQ, QKRS3;
        RELEASE:KURUTMAR;
        DELAY:0.0:DISPOSE;

QKRS3    SEIZE:KURUTMAR;
        ASSIGN:TIMEKURUTMA=OptTime*SipBuy;
        DELAY:TIMEKURUTMA;
        RELEASE:KURUTMAR;

ROUTE;

```

STATION, ZIMPARA;

```

QZMS1    QUEUE, ZIMPARAQ;
        SEIZE:ZIMPARAR;

```

```

BRANCH, 2:
    ALWAYS, QZMS1, YES:
        IF, SIRALAMA==1, ZMS1:
        IF, SIRALAMA==2, ZMS2:
        IF, SIRALAMA==3, ZMS3:
        IF, SIRALAMA==4, ZMS4;
ZMS1    SEARCH, ZIMPARAQ:MAX(BOYAMATIME):NEXT(QZMS2);
ZMS2    SEARCH, ZIMPARAQ:MAX(BASKITIME):NEXT(QZMS2);
ZMS3    SEARCH, ZIMPARAQ:MIN(BASKITIME):NEXT(QZMS2);
ZMS4    SEARCH, ZIMPARAQ:MIN(BOYAMATIME);
QZMS2    REMOVE:J, ZIMPARAQ, QZMS3;
        RELEASE:ZIMPARAR;
        DELAY:0.0:DISPOSE;

QZMS3    SEIZE:ZIMPARAR;
        ASSIGN:TIMEZIMPARA=OptTime*SipBuy;
        DELAY:TIMEZIMPARA;
        RELEASE:ZIMPARAR;

ROUTE;

```

STATION, BOYAMA;

ASSIGN:TOTALTIMEBOYAMA=TOTALTIMEBOYAMA+BOYAMATIME;

```

IF:      TOTALTIMEBOYAMA.LT.BOYAMAALTSURE.AND.TOTALTIMEBASKI.LT.
        BASKIALTSURE;

```

```

        ASSIGN:BASKIORAN=TOTALTIMEBASKI/BASKIALTSURE;
        ASSIGN:BOYAMAORAN=TOTALTIMEBOYAMA/BOYAMAALTSURE;

```

```

IF: BASKIORAN.LT.BOYAMAORAN;
    ASSIGN:SIRALAMA=2;
    ELSE; ASSIGN:SIRALAMA=1;
ENDIF;

ELSEIF:
    TOTALTIMEBOYAMA.LT.BOYAMAALTSURE.AND.TOTALTIMEBASKI.GT.
    BASKIALTSURE;
    ASSIGN:SIRALAMA=1;

ELSEIF:
    TOTALTIMEBOYAMA.GT.BOYAMAALTSURE.AND.TOTALTIMEBASKI.LT.
    BASKIALTSURE;
    ASSIGN:SIRALAMA=2;

ELSEIF:
    TOTALTIMEBOYAMA.GT.BOYAMAALTSURE.AND.TOTALTIMEBASKI.GT.
    BASKIALTSURE;

IF: TOTALTIMEBOYAMA.LT.BOYAMAUSTSURE.AND.TOTALTIMEBASKI.LT.
    BASKIUUSTSURE;
    ASSIGN:BASKIORAN=TOTALTIMEBASKI/BASKIALTSURE;
    ASSIGN:BOYAMAORAN=TOTALTIMEBOYAMA/BOYAMAALTSURE;
    IF: BASKIORAN.LT.BOYAMAORAN;
        ASSIGN:SIRALAMA=2;
        ELSE; ASSIGN:SIRALAMA=1;
    ENDIF;

ELSEIF: TOTALTIMEBOYAMA.GT.BOYAMAUSTSURE.AND.
    TOTALTIMEBASKI.LT.BASKIUUSTSURE;
    ASSIGN:SIRALAMA=2;

ELSEIF: TOTALTIMEBOYAMA.LT.BOYAMAUSTSURE.AND.
    TOTALTIMEBASKI.GT.BASKIUUSTSURE;
    ASSIGN:SIRALAMA=1;

ELSEIF: TOTALTIMEBOYAMA.GT.BOYAMAUSTSURE.AND.
    TOTALTIMEBASKI.GT.BASKIUUSTSURE;
    ASSIGN:BASKIORAN=TOTALTIMEBASKI/BASKIALTSURE;
    ASSIGN:BOYAMAORAN=TOTALTIMEBOYAMA/BOYAMAALTSURE;
    IF: BASKIORAN.LT.BOYAMAORAN;
        ASSIGN:SIRALAMA=4;
        ELSE; ASSIGN:SIRALAMA=3;
    ENDIF;
ENDIF;

ENDIF;

QUEUE,BOYAMAQ;
SEIZE:BOYAMAR;

ASSIGN:TOTALTIMEBOYAMA=TOTALTIMEBOYAMA-BOYAMATIME;
assign:acttypeboyama=renktonu;
ASSIGN:TIMEBOYAMA=OptTime*SipBuy+Tf(prettypeboyama, acttypeboyama);
DELAY:TIMEBOYAMA;
assign:prettypeboyama=acttypeboyama;
assign:renklibeyaz=2;
RELEASE:BOYAMAR;
ASSIGN:BOYAMATIME=0;
ROUTE;

```

STATION, APRE;
QUEUE, APREQ;
SEIZE: APRER;
ASSIGN: TIMEAPRE=OptTime*SipBuy;
DELAY: TIMEAPRE;
RELEASE: APRER;
ROUTE;

STATION, SANFOR;
QUEUE, SANFORQ;
SEIZE: SANFORR;
ASSIGN: TIMESANFOR=OptTime*SipBuy;
DELAY: TIMESANFOR;
RELEASE: SANFORR;
ROUTE;

STATION, FIKSE;
QUEUE, FIKSEQ;
SEIZE: FIKSER;
ASSIGN: TIMEFIKSE=OptTime*SipBuy;
DELAY: TIMEFIKSE;
RELEASE: FIKSER;
ROUTE;

STATION, JETBOYAMA;
QUEUE, JETBOYAMAQ;
SEIZE: JETBOYAMAR;
assign: acttypejetboyama=renktonu;
ASSIGN: TIMEJETBOYAMA=OptTime*SipBuy/1500+
TF(pretypejetboyama, acttypejetboyama);
DELAY: TIMEJETBOYAMA;
assign: pretypejetboyama=acttypejetboyama;
assign: renklibeyaz=2;
RELEASE: JETBOYAMAR;
ROUTE;

STATION, YAKMA;
QUEUE, YAKMAQ;
SEIZE: YAKMAR;
ASSIGN: TIMEYAKMA=OptTime*SipBuy;
DELAY: TIMEYAKMA;
RELEASE: YAKMAR;
ROUTE;

STATION, BASKI;

ASSIGN: TOTALTIMEBASKI=TOTALTIMEBASKI+BASKITIME;

IF: TOTALTIMEBOYAMA.LT.BOYAMAALTSURE.AND.TOTALTIMEBASKI.LT.
BASKIALTSURE;

```

ASSIGN: BASKIORAN = TOTALTIMEBASKI / BASKIALTSURE;
ASSIGN: BOYAMAORAN = TOTALTIMEBOYAMA / BOYAMAALTSURE;

IF: BASKIORAN.LT.BOYAMAORAN;
    ASSIGN: SIRALAMA = 2;
    ELSE; ASSIGN: SIRALAMA = 1;
ENDIF;

ELSEIF:
    TOTALTIMEBOYAMA.LT.BOYAMAALTSURE.AND.TOTALTIMEBASKI.GT.
    BASKIALTSURE;
    ASSIGN: SIRALAMA = 1;

ELSEIF:
    TOTALTIMEBOYAMA.GT.BOYAMAALTSURE.AND.TOTALTIMEBASKI.LT.
    BASKIALTSURE;
    ASSIGN: SIRALAMA = 2;

ELSEIF:
    TOTALTIMEBOYAMA.GT.BOYAMAALTSURE.AND.TOTALTIMEBASKI.GT.
    BASKIALTSURE;

    IF: TOTALTIMEBOYAMA.LT.BOYAMAALTSURE.AND.TOTALTIMEBASKI.LT.
    BASKIALTSURE;
        ASSIGN: BASKIORAN = TOTALTIMEBASKI / BASKIALTSURE;
        ASSIGN: BOYAMAORAN = TOTALTIMEBOYAMA / BOYAMAALTSURE;
        IF: BASKIORAN.LT.BOYAMAORAN;
            ASSIGN: SIRALAMA = 2;
            ELSE; ASSIGN: SIRALAMA = 1;
        ENDIF;

    ELSEIF: TOTALTIMEBOYAMA.GT.BOYAMAALTSURE.AND.
    TOTALTIMEBASKI.LT.BASKIALTSURE;
        ASSIGN: SIRALAMA = 2;

    ELSEIF: TOTALTIMEBOYAMA.LT.BOYAMAALTSURE.AND.
    TOTALTIMEBASKI.GT.BASKIALTSURE;
        ASSIGN: SIRALAMA = 1;

    ELSEIF: TOTALTIMEBOYAMA.GT.BOYAMAALTSURE.AND.
    TOTALTIMEBASKI.GT.BASKIALTSURE;
        ASSIGN: BASKIORAN = TOTALTIMEBASKI / BASKIALTSURE;
        ASSIGN: BOYAMAORAN = TOTALTIMEBOYAMA / BOYAMAALTSURE;
        IF: BASKIORAN.LT.BOYAMAORAN;
            ASSIGN: SIRALAMA = 4;
            ELSE; ASSIGN: SIRALAMA = 3;
        ENDIF;
    ENDIF;
ENDIF;

ENDIF;

QUEUE, BASKIQ;
SEIZE: BASKIR;
ASSIGN: TOTALTIMEBASKI = TOTALTIMEBASKI - BASKITIME;
ASSIGN: TIMEBASKI = OptTime * SipBuy;
DELAY: TIMEBASKI;
RELEASE: BASKIR;
ASSIGN: BASKITIME = 0;
ROUTE;

```

```
STATION, KONDANSE;  
QUEUE, KONDANSEQ;  
SEIZE: KONDANSER;  
ASSIGN: TIMEKONDANSE=OptTime*SipBuy;  
DELAY: TIMEKONDANSE;  
RELEASE: KONDANSER;  
ROUTE;
```

```
STATION, KATLAMA;  
QUEUE, KATLAMAQ;  
SEIZE: KATLAMAR;  
RELEASE: KATLAMAR;
```

```
ASSIGN: GECIKME=DUEDATE - TNOW;
```

```
IF:      GECIKME<0;  
        COUNT: GECKALANLAR;  
        ASSIGN: TOPLAMGECIKME=TOPLAMGECIKME+ABS (GECIKME) ;  
ELSE;  
        COUNT: ERKENBITENLER;  
        ASSIGN: TOPLAMERKEN=TOPLAMERKEN+ABS (GECIKME) ;  
ENDIF;
```

```
TALLY: FlowTime, INT (TimeIn) :DISPOSE;
```

```
create, 1, simson: simson+1, 1;  
        ASSIGN: ORTGECIKME=TOPLAMGECIKME/NC (GECKALANLAR) ;  
        ASSIGN: ORTERKENBIT=TOPLAMERKEN/NC (ERKENBITENLER) ;  
        WRITE, SONUCDOSYA: NC (GECKALANLAR) , ORTGECIKME,  
                NC (ERKENBITENLER) , ORTERKENBIT: dispose;
```

```
END;
```

1.Algoritmaya Ait Simülasyon Modelinin Experiment Dosyası

```
BEGIN;
PROJECT,gulsun, DBR25;

VARIABLES:secim:
    acttypeyikama:
        pretypeyikama,1:
    acttypemerserize:
        pretypemerserize,1:
    acttypeboyama:
        pretypeboyama,1:
    acttypejetboyama:
        pretypejetboyama,1:
    SIRALAMA,1:
    BOYAMAALTSURE,100:
    BASKIALTSURE,100:
    BOYAMAUSTSURE,200:
    BASKIUUSTSURE,200:
    BASKIORAN:
    BOYAMAORAN:
    TOTALTIMEBOYAMA,0:
    TOTALTIMEBASKI,0:
    TOPLAMGECIKME,0:
    TOPLAMERKEN,0:
    ORTGECIKME:
    ORTERKENBIT:
    simson,40000;

ATTRIBUTES:TimeIn:SipBuy:OptTime:DueDate:
    renklibeyaz:renktonu:
    TIMEHASILSOKUMU:
    TIMEBEKLETME:TIMEKOMBINKASAR:TIMEMERSERIZEILK:
    TIMEMERSERIZESON:TIMEYIKAMAILK:TIMEYIKAMASON:
    TIMEKURUTMA:TIMEZIMPARA:TIMEBOYAMA:TIMEAPRE:
    TIMESANFOR:TIMEFIKSE:TIMEJETBOYAMA:
    TIMEYAKMA:TIMEBASKI:TIMEKONDANSE:GECIKME:
    APRETIME:SANFTIME:BOYAMATIME:YAKMATIME:FIKSETIME:
    JETBOYAMATIME:BASKITIME:KONDANSETIME;

COUNTERS:ENTNUM:
    GECKALANLAR:
    ERKENBITENLER;

QUEUES:HASILSOKUMUQ:
    BEKLETMEQ:
    KOMBINKASARQ:
    MERSERIZEQ:
    YIKAMAQ:
    KURUTMAQ:
    ZIMPARAQ:
    BOYAMAQ,LVF(BOYAMATIME):
    APREQ,LVF(APRETIME):
    SANFORQ,LVF(SANFTIME):
```

FIKSEQ, LVF (FIKSETIME) :
JETBOYAMAQ, LVF (JETBOYAMATIME) :
YAKMAQ, LVF (YAKMATIME) :
BASKIQ, LVF (BASKITIME) :
KONDANSEQ, LVF (KONDANSETIME) :
KATLAMAQ, LVF (DUEDATE) ;

RESOURCES:HASILSOKUMUR:

BEKLETMER, 60 :
KOMBINKASARR :
MERSERIZER :
YIKAMAR, 4 :
KURUTMAR, 3 :
ZIMPARAR :
BOYAMAR, 3 :
APRER, 2 :
SANFORR, 2 :
FIKSER :
JETBOYAMAR, 5 :
YAKMAR :
BASKIR :
KONDANSER :
KATLAMAR ;

STATIONS:HASILSOKUMU:

BEKLETME :
KOMBINKASAR :
MERSERIZE :
YIKAMA :
KURUTMA :
ZIMPARA :
BOYAMA :
APRE :
SANFOR :
FIKSE :
JETBOYAMA :
YAKMA :
BASKI :
KONDANSE :
KATLAMA ;

SEQUENCES:1, PAM,
HASILSOKUMU, OptTime=0.00033 & BEKLETME, OptTime=12 &
KOMBINKASAR, OptTime=0.00080 & MERSERIZE, OptTime=0.00033 &
YIKAMA, OptTime=0.00059 & KURUTMA, OptTime=0.00050 &
ZIMPARA, OptTime=0.00166 & YIKAMA, OptTime=0.00058 &
KURUTMA, OptTime=0.00060 & BOYAMA, OptTime=0.00235 &
BEKLETME, OptTime=24 & YIKAMA, OptTime=0.00095 &
KURUTMA, OptTime=0.00088 & APRE, OptTime=0.00078 &
SANFOR, OptTime=0.00060 & KATLAMA:

2, KORD,
KOMBINKASAR, OptTime=0.00080 & YIKAMA, OptTime=0.00058 &

KURUTMA,OptTime=0.00060 & BOYAMA,OptTime=0.00250 &
BEKLETME,OptTime=24 & YIKAMA,OptTime=0.00195 &
KURUTMA,OptTime=0.00088 & APRE,OptTime=0.00078 &
SANFOR,OptTime=0.00060 & KATLAMA:

3, JET,

YIKAMA,OptTime=0.00062 & KURUTMA,OptTime=0.00048 &
YAKMA,OptTime=0.00062 & FIKSE,OptTime=0.00066 &
JETBOYAMA,OptTime=5.40 & KURUTMA,OptTime=0.00085 &
APRE,OptTime=0.00061 & SANFOR,OptTime=0.00062 &
KATLAMA:

4, BASKILI,

HASILSOKUMU,OptTime=0.00033 & BEKLETME,OptTime=12 &
KOMBINKASAR,OptTime=0.00066 & MERSERIZE,OptTime=0.00033 &
YIKAMA,OptTime=0.00063 & KURUTMA,OptTime=0.00046 &
BASKI,OptTime=0.00435 & KONDANSE,OptTime=0.0033 &
APRE,OptTime=0.00052 & SANFOR,OptTime=0.00048 &
KATLAMA:

5, PEVIS,

HASILSOKUMU,OptTime=0.00043 & BEKLETME,OptTime=12 &
YIKAMA,OptTime=0.00058 & KURUTMA,OptTime=0.00041 &
BOYAMA,OptTime=0.00215 & KURUTMA,OptTime=0.00045 &
APRE,OptTime=0.00050 & KATLAMA:

6, GOMLEK,

YIKAMA,OptTime=0.00055 & KURUTMA,OptTime=0.00041 &
YAKMA,OptTime=0.00062 & MERSERIZE,OptTime=0.00033 &
KURUTMA,OptTime=0.00041 & APRE,OptTime=0.00065 &
SANFOR,OptTime=0.00052 & KATLAMA:

7, PERILEN,

YIKAMA,OptTime=0.00037 & FIKSE,OptTime=0.00058 &
APRE,OptTime=0.00095 & SANFOR,OptTime=0.00041 &
KATLAMA;

TALLIES:FlowTime;

DSTATS:NQ(HASILSOKUMUQ)*SipBuy/1000:
NQ(BEKLETMEQ)*SipBuy/1000:
NQ(KOMBINKASARQ)*SipBuy/1000:
NQ(MERSERIZEQ)*SipBuy/1000:
NQ(YIKAMAQ)*SipBuy/1000:
NQ(KURUTMAQ)*SipBuy/1000:
NQ(ZIMPARAQ)*SipBuy/1000:
NQ(BOYAMAQ)*SipBuy/1000:
NQ(APREQ)*SipBuy/1000:
NQ(SANFORQ)*SipBuy/1000:
NQ(FIKSEQ)*SipBuy/1000:
NQ(JETBOYAMAQ)*SipBuy/1000:
NQ(YAKMAQ)*SipBuy/1000:
NQ(BASKIQ)*SipBuy/1000:
NQ(KONDANSEQ)*SipBuy/1000:

NQ(HASILSOKUMUQ)*SipBuy/1000+ NQ(BEKLETMEQ)*SipBuy/1000+
 NQ(KOMBINKASARQ)*SipBuy/1000+ NQ(MERSERIZEQ)*SipBuy/1000+
 NQ(YIKAMAQ)*SipBuy/1000+ NQ(KURUTMAQ)*SipBuy/1000+
 NQ(ZIMPARAQ)*SipBuy/1000+ NQ(BOYAMAQ)*SipBuy/1000+
 NQ(APREQ)*SipBuy/1000+ NQ(SANFORQ)*SipBuy/1000+
 NQ(FIKSEQ)*SipBuy/1000+ NQ(JETBOYAMAQ)*SipBuy/1000+
 NQ(YAKMAQ)*SipBuy/1000+ NQ(BASKIQ)*SipBuy/1000+
 NQ(KONDANSEQ)*SipBuy/1000,TOTALWIP:

NR(HASILSOKUMUR)*100:
 NR(BEKLETMER)/60*100:
 NR(KOMBINKASARR)*100:
 NR(MERSERIZER)*100:
 NR(YIKAMAR)/4*100:
 NR(KURUTMAR)/3*100:
 NR(ZIMPARAR)*100:
 NR(BOYAMAR)/3*100:
 NR(APRER)/2*100:
 NR(SANFORR)/2*100:
 NR(FIKSER)*100:
 NR(JETBOYAMAR)/5*100:
 NR(YAKMAR)*100:
 NR(BASKIR)*100:
 NR(KONDANSER)*100;

FILES:SONUCDOSYA,"DBR25(2.2).DAT",SEQ,WKS,DIS;

TABLES: 1, 1,1, 0,0,0:
 2, 1,1, 2,0,2:
 3, 1,1, 3,2,0;

REPLICATE,5,0,40000,,,5000;

END;

EK 5 ALGORİTMA 2 İÇİN SİMÜLASYON UYGULAMASI

2. Algoritmaya Ait Örnek Simülasyon Modeli

```
BEGIN;

CREATE:exponential(2.2,1):MARK(TimeIn);
COUNT:ENTNUM;

ASSIGN:NS=discrete(0.18,1,0.34,2,0.57,3,0.65,4,
                  0.74,5,0.80,6,1.0,7);
SipBuy=discrete(0.2,2500,0.6,4000,0.85,6000,0.95,10000,
                1.0,25000);

BRANCH,1:WITH,0.3,export:
        WITH,0.7,yerli;

export  ASSIGN:DueDate=TNOW+UNIF(336,504):NEXT(PA1);
yerli   ASSIGN:DueDate=TNOW+UNIF(504,672);

PA1 IF: NS==1;
      ASSIGN:TIMETOBASKI=BS;
      ASSIGN:TIMETOBOYAMA=SipBuy*0.00539+12;
      ASSIGN:APRETIME=SipBuy*0.00078;
      ASSIGN:SANFTIME=SipBuy*0.00060;
      ASSIGN:BOYAMATIME=SipBuy*0.00235:
      Next(PA2);

ELSEIF:
      NS==2;
      ASSIGN:TIMETOBASKI=BS;
      ASSIGN:TIMETOBOYAMA=SipBuy*0.00198;
      ASSIGN:APRETIME=SipBuy*0.00078;
      ASSIGN:SANFTIME=SipBuy*0.00060;
      ASSIGN:BOYAMATIME=SipBuy*0.00250:
      Next(PA2);

ELSEIF:
      NS==3;
      ASSIGN:TIMETOBASKI=BS;
      ASSIGN:TIMETOBOYAMA=BS;
      ASSIGN:APRETIME=SipBuy*0.00061;
      ASSIGN:SANFTIME=SipBuy*0.00062;
      ASSIGN:JETBOYAMATIME=SipBuy*0.00250/1500;
      ASSIGN:YAKMATIME=SipBuy*0.00062;
      ASSIGN:FIKSETIME=SipBuy*0.00066:
      Next(PA2);

ELSEIF:
      NS==4;
      ASSIGN:TIMETOBASKI=SipBuy*0.00241+12;
      ASSIGN:TIMETOBOYAMA=BS;
      ASSIGN:APRETIME=SipBuy*0.00052;
      ASSIGN:SANFTIME=SipBuy*0.00048;
      ASSIGN:BASKITIME=SipBuy*0.00435;
```

```

        ASSIGN:KONDANSETIME=SipBuy*0.0033:
        Next (PA3);
ELSEIF:
    NS==5;
    ASSIGN:TIMETOBASKI=BS;
    ASSIGN:TIMETOBOYAMA=SipBuy*0.00142+12;
    ASSIGN:APRETIME=SipBuy*0.00050;
    ASSIGN:BOYAMATIME=SipBuy*0.00215:
    Next (PA2);
ELSEIF:
    NS==6;
    ASSIGN:TIMETOBASKI=BS;
    ASSIGN:TIMETOBOYAMA=BS;
    ASSIGN:APRETIME=SipBuy*0.00065;
    ASSIGN:SANFTIME=SipBuy*0.00052;
    ASSIGN:YAKMATIME=SipBuy*0.00062:
    Next (PA3);
ELSE;
    ASSIGN:TIMETOBASKI=BS;
    ASSIGN:TIMETOBOYAMA=BS;
    ASSIGN:APRETIME=SipBuy*0.00095;
    ASSIGN:SANFTIME=SipBuy*0.00041;
    ASSIGN:FIKSETIME=SipBuy*0.00058:
    Next (PA3);
ENDIF;

PA2    assign:renklibeyaz=1;
PA21   assign:renktonu=discrete(0.3,1,0.6,2,1.0,3):NEXT(PA4);

PA3    assign:renklibeyaz=2;
PA31   assign:renktonu=1;

PA4    ROUTE;

STATION,HASILSOKUMU;
QHSS1  QUEUE,HASILSOKUMUQ;
        SEIZE:HASILSOKUMUR;

BRANCH, 2:
    ALWAYS, QHSS1, YES:
        IF, SIRALAMA==1,HSS1:
        IF, SIRALAMA==2,HSS2:
        IF, SIRALAMA==3,HSS3:
        IF, SIRALAMA==4,HSS4;
HSS1    SEARCH,HASILSOKUMUQ:MIN(TIMETOBOYAMA):NEXT(QHSS2);
HSS2    SEARCH,HASILSOKUMUQ:MIN(TIMETOBASKI):NEXT(QHSS2);
HSS3    SEARCH,HASILSOKUMUQ:MAX(TIMETOBASKI):NEXT(QHSS2);
HSS4    SEARCH,HASILSOKUMUQ:MAX(TIMETOBOYAMA);
QHSS2   REMOVE:J,HASILSOKUMUQ,QHSS3;
        RELEASE:HASILSOKUMUR;
        DELAY:0.0:DISPOSE;

QHSS3   SEIZE:HASILSOKUMUR;

```

```

        ASSIGN:TIMEHASILSOKUMU=OptTime*SipBuy;
        DELAY:TIMEHASILSOKUMU;
        ASSIGN:TIMETOBASKI=TIMETOBASKI-TIMEHASILSOKUMU;
        ASSIGN:TIMETOBOYAMA=TIMETOBOYAMA-TIMEHASILSOKUMU;
        RELEASE:HASILSOKUMUR;

ROUTE;

STATION,BEKLETME;
QBK1  QUEUE,BEKLETMEQ;
      SEIZE:BEKLETMER;

BRANCH, 2:
      ALWAYS, QBK1, YES:
        IF, SIRALAMA==1,BK1:
        IF, SIRALAMA==2,BK2:
        IF, SIRALAMA==3,BK3:
        IF, SIRALAMA==4,BK4;
BK1  SEARCH,BEKLETMEQ:MIN(TIMETOBOYAMA):NEXT(QBK2);
BK2  SEARCH,BEKLETMEQ:MIN(TIMETOBASKI):NEXT(QBK2);
BK3  SEARCH,BEKLETMEQ:MAX(TIMETOBASKI):NEXT(QBK2);
BK4  SEARCH,BEKLETMEQ:MAX(TIMETOBOYAMA);
QBK2  REMOVE:J,BEKLETMEQ,QBK3;
      RELEASE:BEKLETMER;
      DELAY:0.0:DISPOSE;

QBK3  SEIZE:BEKLETMER;
      ASSIGN:TIMEBEKLETME=OptTime;
      DELAY:TIMEBEKLETME;
      ASSIGN:TIMETOBASKI=TIMETOBASKI-TIMEBEKLETME;
      ASSIGN:TIMETOBOYAMA=TIMETOBOYAMA-TIMEBEKLETME;
      RELEASE:BEKLETMER;

ROUTE;

STATION,KOMBINKASAR;
QKKS1  QUEUE,KOMBINKASARQ;
      SEIZE:KOMBINKASARR;

BRANCH, 2:
      ALWAYS, QKKS1, YES:
        IF, SIRALAMA==1,KKS1:
        IF, SIRALAMA==2,KKS2:
        IF, SIRALAMA==3,KKS3:
        IF, SIRALAMA==4,KKS4;
KKS1  SEARCH,KOMBINKASARQ:MIN(TIMETOBOYAMA):NEXT(QKKS2);
KKS2  SEARCH,KOMBINKASARQ:MIN(TIMETOBASKI):NEXT(QKKS2);
KKS3  SEARCH,KOMBINKASARQ:MAX(TIMETOBASKI):NEXT(QKKS2);
KKS4  SEARCH,KOMBINKASARQ:MAX(TIMETOBOYAMA);
QKKS2  REMOVE:J,KOMBINKASARQ,QKKS3;
      RELEASE:KOMBINKASARR;
      DELAY:0.0:DISPOSE;

QKKS3  SEIZE:KOMBINKASARR;
      ASSIGN:TIMEKOMBINKASAR=OptTime*SipBuy;
      DELAY:TIMEKOMBINKASAR;
      ASSIGN:TIMETOBASKI=TIMETOBASKI-TIMEKOMBINKASAR;

```

```

ASSIGN:TIMETOBOYAMA=TIMETOBOYAMA-TIMEKOMBINKASAR;
RELEASE:KOMBINKASARR;

ROUTE;

STATION,MERSERIZE;
QMRS1  QUEUE,MERSERIZEQ;
      SEIZE:MERSERIZER;

BRANCH, 2:
      ALWAYS, QMRS1, YES:
      IF, SIRALAMA==1,MRS1:
      IF, SIRALAMA==2,MRS2:
      IF, SIRALAMA==3,MRS3:
      IF, SIRALAMA==4,MRS4;
MRS1   SEARCH,MERSERIZEQ:MIN(TIMETOBOYAMA):NEXT(QMRS2);
MRS2   SEARCH,MERSERIZEQ:MIN(TIMETOBASKI):NEXT(QMRS2);
MRS3   SEARCH,MERSERIZEQ:MAX(TIMETOBASKI):NEXT(QMRS2);
MRS4   SEARCH,MERSERIZEQ:MAX(TIMETOBOYAMA);
QMRS2  REMOVE:J,MERSERIZEQ,QMRS3;
      RELEASE:MERSERIZER;
      DELAY:0.0:DISPOSE;

QMRS3  SEIZE:MERSERIZER;
      assign:acttypemerserize=renklibeyaz;
PA5    ASSIGN:TIMEMERSERIZEILK=OptTime*SipBuy;
branch,1:
      if, acttypemerserize.NE.pretypemerserize, PA51:
      else, PA6;
PA51   ASSIGN:TIMEMERSERIZESON=TIMEMERSERIZEILK+4:NEXT(PA61);
PA6    ASSIGN:TIMEMERSERIZESON=TIMEMERSERIZEILK;
PA61   DELAY:TIMEMERSERIZESON;
      assign:prettypemerserize=acttypemerserize;
      ASSIGN:TIMETOBASKI=TIMETOBASKI-TIMEMERSERIZESON;
      ASSIGN:TIMETOBOYAMA=TIMETOBOYAMA-TIMEMERSERIZESON;
      RELEASE:MERSERIZER;

ROUTE;

STATION,YIKAMA;
QYKS1  QUEUE,YIKAMAQ;
      SEIZE:YIKAMAR;

BRANCH, 2:
      ALWAYS, QYKS1, YES:
      IF, SIRALAMA==1,YKS1:
      IF, SIRALAMA==2,YKS2:
      IF, SIRALAMA==3,YKS3:
      IF, SIRALAMA==4,YKS4;
YKS1   SEARCH,YIKAMAQ:MIN(TIMETOBOYAMA):NEXT(QYKS2);
YKS2   SEARCH,YIKAMAQ:MIN(TIMETOBASKI):NEXT(QYKS2);
YKS3   SEARCH,YIKAMAQ:MAX(TIMETOBASKI):NEXT(QYKS2);
YKS4   SEARCH,YIKAMAQ:MAX(TIMETOBOYAMA);
QYKS2  REMOVE:J,YIKAMAQ,QYKS3;
      RELEASE:YIKAMAR;
      DELAY:0.0:DISPOSE;

```

```

QYKS3  SEIZE:YIKAMAR;
        assign:acttypeyikama=renklibeyaz;
PA7     ASSIGN:TIMEYIKAMAILK=OptTime*SipBuy;
branch,1:
        if, acttypeyikama.NE.prettypeyikama, PA71:
        else, PA8;
PA71    ASSIGN:TIMEYIKAMASON=TIMEYIKAMAILK+2:NEXT(PA81);
PA8     ASSIGN:TIMEYIKAMASON=TIMEYIKAMAILK;
PA81    DELAY:TIMEYIKAMASON;
        assign:prettypeyikama=acttypeyikama;
        ASSIGN:TIMETOBASKI=TIMETOBASKI-TIMEYIKAMASON;
        ASSIGN:TIMETOBOYAMA=TIMETOBOYAMA-TIMEYIKAMASON;
        RELEASE:YIKAMAR;
ROUTE;

```

```

STATION, KURUTMA;
QKRS1   QUEUE, KURUTMAQ;
        SEIZE:KURUTMAR;

```

```

BRANCH, 2:
        ALWAYS, QKRS1, YES:
        IF, SIRALAMA==1, KRS1:
        IF, SIRALAMA==2, KRS2:
        IF, SIRALAMA==3, KRS3:
        IF, SIRALAMA==4, KRS4;
KRS1    SEARCH, KURUTMAQ:MIN(TIMETOBOYAMA):NEXT(QKRS2);
KRS2    SEARCH, KURUTMAQ:MIN(TIMETOBASKI):NEXT(QKRS2);
KRS3    SEARCH, KURUTMAQ:MAX(TIMETOBASKI):NEXT(QKRS2);
KRS4    SEARCH, KURUTMAQ:MAX(TIMETOBOYAMA);
QKRS2   REMOVE:J, KURUTMAQ, QKRS3;
        RELEASE:KURUTMAR;
        DELAY:0.0:DISPOSE;

```

```

QKRS3   SEIZE:KURUTMAR;
        ASSIGN:TIMEKURUTMA=OptTime*SipBuy;
        DELAY:TIMEKURUTMA;
        ASSIGN:TIMETOBASKI=TIMETOBASKI-TIMEKURUTMA;
        ASSIGN:TIMETOBOYAMA=TIMETOBOYAMA-TIMEKURUTMA;
        RELEASE:KURUTMAR;
ROUTE;

```

```

STATION, ZIMPARA;
QZMS1   QUEUE, ZIMPARAQ;
        SEIZE:ZIMPARAR;

```

```

BRANCH, 2:
        ALWAYS, QZMS1, YES:
        IF, SIRALAMA==1, ZMS1:
        IF, SIRALAMA==2, ZMS2:
        IF, SIRALAMA==3, ZMS3:
        IF, SIRALAMA==4, ZMS4;
ZMS1    SEARCH, ZIMPARAQ:MIN(TIMETOBOYAMA):NEXT(QZMS2);
ZMS2    SEARCH, ZIMPARAQ:MIN(TIMETOBASKI):NEXT(QZMS2);

```

ZMS3 SEARCH, ZIMPARAQ:MAX (TIMETOBASKI) :NEXT (QZMS2) ;
ZMS4 SEARCH, ZIMPARAQ:MAX (TIMETOBOYAMA) ;
QZMS2 REMOVE:J, ZIMPARAQ, QZMS3;
 RELEASE: ZIMPARAR;
 DELAY:0.0:DISPOSE;

QZMS3 SEIZE: ZIMPARAR;
 ASSIGN:TIMEZIMPARA=OptTime*SipBuy;
 DELAY:TIMEZIMPARA;
 ASSIGN:TIMETOBASKI=TIMETOBASKI-TIMEZIMPARA;
 ASSIGN:TIMETOBOYAMA=TIMETOBOYAMA-TIMEZIMPARA;
 RELEASE: ZIMPARAR;

ROUTE;

STATION, BOYAMA;
ASSIGN: TOTALTIMEBOYAMA=TOTALTIMEBOYAMA+BOYAMATIME;

IF: TOTALTIMEBOYAMA.LT.BOYAMAALTSURE.AND.TOTALTIMEBASKI.LT.
 BASKIALTSURE;

 ASSIGN:BASKIORAN=TOTALTIMEBASKI/BASKIALTSURE;
 ASSIGN:BOYAMAORAN=TOTALTIMEBOYAMA/BOYAMAALTSURE;

 IF: BASKIORAN.LT.BOYAMAORAN;
 ASSIGN:SIRALAMA=2;
 ELSE; ASSIGN:SIRALAMA=1;
 ENDIF;

ELSEIF: TOTALTIMEBOYAMA.LT.BOYAMAALTSURE.AND.TOTALTIMEBASKI.GT.
 BASKIALTSURE;
 ASSIGN:SIRALAMA=1;

ELSEIF: TOTALTIMEBOYAMA.GT.BOYAMAALTSURE.AND.TOTALTIMEBASKI.LT.
 BASKIALTSURE;
 ASSIGN:SIRALAMA=2;

ELSEIF: TOTALTIMEBOYAMA.GT.BOYAMAALTSURE.AND.TOTALTIMEBASKI.GT.
 BASKIALTSURE;

 IF: TOTALTIMEBOYAMA.LT.BOYAMAALTSURE.AND.TOTALTIMEBASKI.LT.
 BASKIALTSURE;
 ASSIGN:BASKIORAN=TOTALTIMEBASKI/BASKIALTSURE;
 ASSIGN:BOYAMAORAN=TOTALTIMEBOYAMA/BOYAMAALTSURE;
 IF: BASKIORAN.LT.BOYAMAORAN;
 ASSIGN:SIRALAMA=2;
 ELSE; ASSIGN:SIRALAMA=1;
 ENDIF;

 ELSEIF: TOTALTIMEBOYAMA.GT.BOYAMAALTSURE.AND.
 TOTALTIMEBASKI.LT.BASKIALTSURE;
 ASSIGN:SIRALAMA=2;

 ELSEIF: TOTALTIMEBOYAMA.LT.BOYAMAALTSURE.AND.
 TOTALTIMEBASKI.GT.BASKIALTSURE;

```

        ASSIGN:SIRALAMA=1;
    ELSEIF: TOTALTIMEBOYAMA.GT.BOYAMAUSTSURE.AND.
        TOTALTIMEBASKI.GT.BASKIUSTSURE;
        ASSIGN:BASKIORAN=TOTALTIMEBASKI/BASKIALTSURE;
        ASSIGN:BOYAMAORAN=TOTALTIMEBOYAMA/BOYAMAALTSURE;
    IF: BASKIORAN.LT.BOYAMAORAN;
        ASSIGN:SIRALAMA=4;
    ELSE; ASSIGN:SIRALAMA=3;
    ENDIF;
ENDIF;
ENDIF;

```

```

QUEUE,BOYAMAQ;
SEIZE:BOYAMAR;
ASSIGN:TOTALTIMEBOYAMA=TOTALTIMEBOYAMA-BOYAMATIME;
assign:acttypeboyama=renktonu;
ASSIGN:TIMEBOYAMA=OptTime*SipBuy+TF(prettypeboyama, acttypeboyama);
DELAY:TIMEBOYAMA;
assign:prettypeboyama=acttypeboyama;
assign:renklibeyaz=2;
RELEASE:BOYAMAR;
ASSIGN:BOYAMATIME=0;
ROUTE;

```

```

STATION,APRE;
QUEUE,APREQ;
SEIZE:APRER;
ASSIGN:TIMEAPRE=OptTime*SipBuy;
DELAY:TIMEAPRE;
RELEASE:APRER;
ROUTE;

```

```

STATION,SANFOR;
QUEUE,SANFORQ;
SEIZE:SANFORR;
ASSIGN:TIMESANFOR=OptTime*SipBuy;
DELAY:TIMESANFOR;
RELEASE:SANFORR;
ROUTE;

```

```

STATION,FIKSE;
QUEUE,FIKSEQ;
SEIZE:FIKSER;
ASSIGN:TIMEFIKSE=OptTime*SipBuy;
DELAY:TIMEFIKSE;
RELEASE:FIKSER;
ROUTE;

```

```

STATION,JETBOYAMA;
QUEUE,JETBOYAMAQ;
SEIZE:JETBOYAMAR;

```

```
assign:acttypejetboyama=renktonu;
ASSIGN:TIMEJETBOYAMA=OptTime*SipBuy/1500+
    TF(pretypejetboyama, acttypejetboyama);
DELAY:TIMEJETBOYAMA;
assign:pretypejetboyama=acttypejetboyama;
assign:renklibeyaz=2;
RELEASE:JETBOYAMAR;
ROUTE;
```

```
STATION, YAKMA;
QUEUE, YAKMAQ;
SEIZE:YAKMAR;
ASSIGN:TIMEYAKMA=OptTime*SipBuy;
DELAY:TIMEYAKMA;
RELEASE:YAKMAR;
ROUTE;
```

```
STATION, BASKI;
```

```
ASSIGN:TOTALTIMEBASKI=TOTALTIMEBASKI+BASKITIME;
```

```
IF:    TOTALTIMEBOYAMA.LT.BOYAMAALTSURE.AND.TOTALTIMEBASKI.LT.
        BASKIALTSURE;
```

```
    ASSIGN:BASKIORAN=TOTALTIMEBASKI/BASKIALTSURE;
    ASSIGN:BOYAMAORAN=TOTALTIMEBOYAMA/BOYAMAALTSURE;
```

```
    IF: BASKIORAN.LT.BOYAMAORAN;
        ASSIGN:SIRALAMA=2;
    ELSE; ASSIGN:SIRALAMA=1;
    ENDIF;
```

```
ELSEIF: TOTALTIMEBOYAMA.LT.BOYAMAALTSURE.AND.TOTALTIMEBASKI.GT.
        BASKIALTSURE;
        ASSIGN:SIRALAMA=1;
```

```
ELSEIF: TOTALTIMEBOYAMA.GT.BOYAMAALTSURE.AND.TOTALTIMEBASKI.LT.
        BASKIALTSURE;
        ASSIGN:SIRALAMA=2;
```

```
ELSEIF: TOTALTIMEBOYAMA.GT.BOYAMAALTSURE.AND.TOTALTIMEBASKI.GT.
        BASKIALTSURE;
```

```
    IF: TOTALTIMEBOYAMA.LT.BOYAMAALTSURE.AND.TOTALTIMEBASKI.LT.
        BASKIALTSURE;
        ASSIGN:BASKIORAN=TOTALTIMEBASKI/BASKIALTSURE;
        ASSIGN:BOYAMAORAN=TOTALTIMEBOYAMA/BOYAMAALTSURE;
    IF: BASKIORAN.LT.BOYAMAORAN;
        ASSIGN:SIRALAMA=2;
    ELSE; ASSIGN:SIRALAMA=1;
    ENDIF;
```

```
ELSEIF: TOTALTIMEBOYAMA.GT.BOYAMAALTSURE.AND.
        TOTALTIMEBASKI.LT.BASKIALTSURE;
```

```

        ASSIGN:SIRALAMA=2;
ELSEIF: TOTALTIMEBOYAMA.LT.BOYAMAUSTSURE.AND.
        TOTALTIMEBASKI.GT.BASKIUSTSURE;
        ASSIGN:SIRALAMA=1;
ELSEIF: TOTALTIMEBOYAMA.GT.BOYAMAUSTSURE.AND.
        TOTALTIMEBASKI.GT.BASKIUSTSURE;
        ASSIGN:BASKIORAN=TOTALTIMEBASKI/BASKIALTSURE;
        ASSIGN:BOYAMAORAN=TOTALTIMEBOYAMA/BOYAMAALTSURE;
        IF: BASKIORAN.LT.BOYAMAORAN;
            ASSIGN:SIRALAMA=4;
        ELSE; ASSIGN:SIRALAMA=3;
        ENDIF;
    ENDIF;
ENDIF;

```

```

QUEUE, BASKIQ;
SEIZE: BASKIR;
ASSIGN: TOTALTIMEBASKI = TOTALTIMEBASKI - BASKITIME;
ASSIGN: TIMEBASKI = OptTime * SipBuy;
DELAY: TIMEBASKI;
RELEASE: BASKIR;
ASSIGN: BASKITIME = 0;
ROUTE;

```

```

STATION, KONDANSE;
QUEUE, KONDANSEQ;
SEIZE: KONDANSER;
ASSIGN: TIMEKONDANSE = OptTime * SipBuy;
DELAY: TIMEKONDANSE;
RELEASE: KONDANSER;
ROUTE;

```

```

STATION, KATLAMA;
QUEUE, KATLAMAQ;
SEIZE: KATLAMAR;
RELEASE: KATLAMAR;

```

```

ASSIGN: GECIKME = DUE DATE - TNOW;

```

```

IF:      GECIKME < 0;
        COUNT: GECKALANLAR;
        ASSIGN: TOPLAMGECIKME = TOPLAMGECIKME + ABS (GECIKME) ;
ELSE;
        COUNT: ERKENBITENLER;
        ASSIGN: TOPLAMERKEN = TOPLAMERKEN + ABS (GECIKME) ;
ENDIF;

```

```

TALLY: FlowTime, INT (TimeIn) : DISPOSE;

```

```
create,1,simson:simson+1,1;  
  ASSIGN:ORTGECIKME=TOPLAMGECIKME/NC(GECKALANLAR);  
  ASSIGN:ORTERKENBIT=TOPLAMERKEN/NC(ERKENBITENLER);  
  WRITE, SONUCDOSYA:NC(GECKALANLAR),ORTGECIKME,  
    NC(ERKENBITENLER),ORTERKENBIT:dispose;  
  
END;
```

2.Algoritmaya Ait Simülasyon Modelinin Experiment Dosyası

```
BEGIN;  
PROJECT,gulsun DBR33;
```

```
VARIABLES:secim:  
    acttypeyikama:  
        pretypeyikama,1:  
    acttypemerserize:  
        pretypemerserize,1:  
    acttypeboyama:  
        pretypeboyama,1:  
    acttypejetboyama:  
        pretypejetboyama,1:  
    SIRALAMA,1:  
    BOYAMAALTSURE,40:  
    BASKIALTSURE,40:  
    BOYAMAUSTSURE,70:  
    BASKIUUSTSURE,70:  
    BASKIORAN:  
    BOYAMAORAN:  
    TOTALTIMEBOYAMA,0:  
    TOTALTIMEBASKI,0:  
    TOPLAMGECIKME,0:  
    TOPLAMERKEN,0:  
    ORTGECIKME:  
    ORTERKENBIT:  
    BS,2000:  
    simson,40000;
```

```
ATTRIBUTES:TimeIn:SipBuy:OptTime:DueDate:  
    renklibeyaz:renktonu:  
    TIMEHASILSOKUMU:  
    TIMEBEKLETME:TIMEKOMBINKASAR:TIMEMERSERIZEILK:  
    TIMEMERSERIZESON:TIMEYIKAMAILK:TIMEYIKAMASON:  
    TIMEKURUTMA:TIMEZIMPARA:TIMEBOYAMA:TIMEAPRE:  
    TIMESANFOR:TIMEFIKSE:TIMEJETBOYAMA:  
    TIMEYAKMA:TIMEBASKI:TIMEKONDANSE:GECIKME:  
    APRETIME:SANFTIME:BOYAMATIME:YAKMATIME:FIKSETIME:  
    JETBOYAMATIME:BASKITIME:KONDANSETIME:  
    TIMETOBOYAMA:  
    TIMETOBASKI;
```

```
COUNTERS:ENTNUM:  
    GECKALANLAR:  
    ERKENBITENLER;
```

```
QUEUES:HASILSOKUMUQ:  
    BEKLETMEQ:  
    KOMBINKASARQ:  
    MERSERIZEQ:  
    YIKAMAQ:
```

KURUTMAQ:
ZIMPARAQ:
BOYAMAQ, LVF (BOYAMATIME) :
APREQ, LVF (APRETIME) :
SANFORQ, LVF (SANFTIME) :
FIKSEQ, LVF (FIKSETIME) :
JETBOYAMAQ, LVF (JETBOYAMATIME) :
YAKMAQ, LVF (YAKMATIME) :
BASKIQ, LVF (BASKITIME) :
KONDANSEQ, LVF (KONDANSETIME) :
KATLAMAQ, LVF (DUEDATE) ;

RESOURCES:HASILSOKUMUR:

BEKLETMER, 60:
KOMBINKASARR:
MERSERIZER:
YIKAMAR, 4:
KURUTMAR, 3:
ZIMPARAR:
BOYAMAR, 3:
APRER, 2:
SANFORR, 2:
FIKSER:
JETBOYAMAR, 5:
YAKMAR:
BASKIR:
KONDANSER:
KATLAMAR;

STATIONS:HASILSOKUMU:

BEKLETME:
KOMBINKASAR:
MERSERIZE:
YIKAMA:
KURUTMA:
ZIMPARA:
BOYAMA:
APRE:
SANFOR:
FIKSE:
JETBOYAMA:
YAKMA:
BASKI:
KONDANSE:
KATLAMA;

SEQUENCES:1, PAM,
HASILSOKUMU, OptTime=0.00033 & BEKLETME, OptTime=12 &
KOMBINKASAR, OptTime=0.00080 & MERSERIZE, OptTime=0.00033 &
YIKAMA, OptTime=0.00059 & KURUTMA, OptTime=0.00050 &
ZIMPARA, OptTime=0.00166 & YIKAMA, OptTime=0.00058 &
KURUTMA, OptTime=0.00060 & BOYAMA, OptTime=0.00235 &
BEKLETME, OptTime=24 & YIKAMA, OptTime=0.00095 &

KURUTMA,OptTime=0.00088 & APRE,OptTime=0.00078 &
SANFOR,OptTime=0.00060 & KATLAMA:

2, KORD,
KOMBINKASAR,OptTime=0.00080 & YIKAMA,OptTime=0.00058 &
KURUTMA,OptTime=0.00060 & BOYAMA,OptTime=0.00250 &
BEKLETME,OptTime=24 & YIKAMA,OptTime=0.00195 &
KURUTMA,OptTime=0.00088 & APRE,OptTime=0.00078 &
SANFOR,OptTime=0.00060 & KATLAMA:

3, JET,
YIKAMA,OptTime=0.00062 & KURUTMA,OptTime=0.00048 &
YAKMA,OptTime=0.00062 & FIKSE,OptTime=0.00066 &
JETBOYAMA,OptTime=5.40 & KURUTMA,OptTime=0.00085 &
APRE,OptTime=0.00061 & SANFOR,OptTime=0.00062 &
KATLAMA:

4, BASKILI,
HASILSOKUMU,OptTime=0.00033 & BEKLETME,OptTime=12 &
KOMBINKASAR,OptTime=0.00066 & MERSEIZE,OptTime=0.00033 &
YIKAMA,OptTime=0.00063 & KURUTMA,OptTime=0.00046 &
BASKI,OptTime=0.00435 & KONDANSE,OptTime=0.0033 &
APRE,OptTime=0.00052 & SANFOR,OptTime=0.00048 &
KATLAMA:

5, PEVIS,
HASILSOKUMU,OptTime=0.00043 & BEKLETME,OptTime=12 &
YIKAMA,OptTime=0.00058 & KURUTMA,OptTime=0.00041 &
BOYAMA,OptTime=0.00215 & KURUTMA,OptTime=0.00045 &
APRE,OptTime=0.00050 & KATLAMA:

6, GOMLEK,
YIKAMA,OptTime=0.00055 & KURUTMA,OptTime=0.00041 &
YAKMA,OptTime=0.00062 & MERSEIZE,OptTime=0.00033 &
KURUTMA,OptTime=0.00041 & APRE,OptTime=0.00065 &
SANFOR,OptTime=0.00052 & KATLAMA:

7, PERILEN,
YIKAMA,OptTime=0.00037 & FIKSE,OptTime=0.00058 &
APRE,OptTime=0.00095 & SANFOR,OptTime=0.00041 &
KATLAMA;

TALLIES:FlowTime;

DSTATS:NQ(HASILSOKUMUQ)*SipBuy/1000:
NQ(BEKLETMEQ)*SipBuy/1000:
NQ(KOMBINKASARQ)*SipBuy/1000:
NQ(MERSEIZEQ)*SipBuy/1000:
NQ(YIKAMAQ)*SipBuy/1000:
NQ(KURUTMAQ)*SipBuy/1000:
NQ(ZIMPARAQ)*SipBuy/1000:
NQ(BOYAMAQ)*SipBuy/1000:
NQ(APREQ)*SipBuy/1000:
NQ(SANFORQ)*SipBuy/1000:

NQ(FIKSEQ)*SipBuy/1000:
NQ(JETBOYAMAQ)*SipBuy/1000:
NQ(YAKMAQ)*SipBuy/1000:
NQ(BASKIQ)*SipBuy/1000:
NQ(KONDANSEQ)*SipBuy/1000:

NQ(HASILSOKUMUQ)*SipBuy/1000+ NQ(BEKLETMEQ)*SipBuy/1000+
NQ(KOMBINKASARQ)*SipBuy/1000+ NQ(MERSERIZEQ)*SipBuy/1000+
NQ(YIKAMAQ)*SipBuy/1000+ NQ(KURUTMAQ)*SipBuy/1000+
NQ(ZIMPARAQ)*SipBuy/1000+ NQ(BOYAMAQ)*SipBuy/1000+
NQ(APREQ)*SipBuy/1000+ NQ(SANFORQ)*SipBuy/1000+
NQ(FIKSEQ)*SipBuy/1000+ NQ(JETBOYAMAQ)*SipBuy/1000+
NQ(YAKMAQ)*SipBuy/1000+ NQ(BASKIQ)*SipBuy/1000+
NQ(KONDANSEQ)*SipBuy/1000,TOTALWIP:

NR(HASILSOKUMUR)*100:
NR(BEKLETMER)/60*100:
NR(KOMBINKASARR)*100:
NR(MERSERIZER)*100:
NR(YIKAMAR)/4*100:
NR(KURUTMAR)/3*100:
NR(ZIMPARAR)*100:
NR(BOYAMAR)/3*100:
NR(APRER)/2*100:
NR(SANFORR)/2*100:
NR(FIKSER)*100:
NR(JETBOYAMAR)/5*100:
NR(YAKMAR)*100:
NR(BASKIR)*100:
NR(KONDANSER)*100;

FILES:SONUCDOSYA,"DBR33(2.5).DAT",SEQ,WKS,DIS;

TABLES: 1, 1,1, 0,0,0:
2, 1,1, 2,0,2:
3, 1,1, 3,2,0;

REPLICATE,5,0,40000,,,5000;

END;