

**ESNEK ATÖLYE TİPİ ÜRETİM ORTAMINDA
STOKASTİK TALEP İÇİN DİNAMİK ÇİZELGELEME**

Ü. Eraydın GENÇ



T.C.
BURSA ULUDAĞ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**ESNEK ATÖLYE TİPİ ÜRETİM ORTAMINDA STOKASTİK TALEP İÇİN
DİNAMİK ÇİZELGELEME**

Ü. Eraydın GENÇ

Prof. Dr. Erdal EMEL
(Danışman)

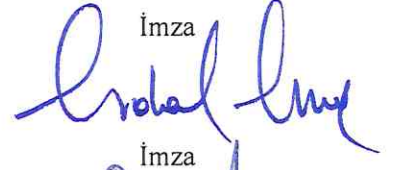
YÜKSEK LİSANS TEZİ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

TEZ ONAYI

Ü. Eraydın GENÇ tarafından hazırlanan “ESNEK ATÖLYE TİPİ ÜRETİM ORTAMINDA STOKASTİK TALEP İÇİN DİNAMİK ÇİZELGELEME” adlı tez çalışması aşağıdaki jüri tarafından oy birliği ile Bursa Uludağ Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı’nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman : Prof. Dr. Erdal EMEL

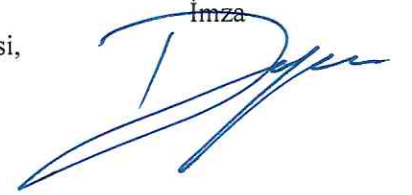
Başkan : Prof. Dr. Erdal EMEL
Uludağ Üniversitesi, Mühendislik Fakültesi,
Endüstri Mühendisliği Anabilim Dalı

İmza


Üye : Doç.Dr. Aytaç YILDIZ
Bursa Teknik Üniversitesi,
Mühendislik ve Doğa Bilimleri Fakültesi,
Endüstri Mühendisliği Anabilim Dalı

İmza


Üye : Dr. Duygu YILMAZ EROĞLU
Uludağ Üniversitesi, Mühendislik Fakültesi,
Endüstri Mühendisliği Anabilim Dalı

İmza


Yukarıdaki sonucu onaylarım

Prof.Dr.Hüseyin Aksel EREN
Enstitü Müdürü

.../.../....

U.Ü. Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmada;

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

25/10/2019..


Ü. Eraydın GENÇ

ÖZET

Yüksek Lisans Tezi

ESNEK ATÖLYE TİPİ ÜRETİM ORTAMINDA STOKASTİK TALEP İÇİN DİNAMİK ÇİZELGELEME

Ü. Eraydın GENÇ

Uludağ Üniversitesi

Fen Bilimleri Enstitüsü

Endüstri Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Erdal EMEL

Bu tez çalışmasının ana hedefi hat duruşları, iş kuyrukları ve parçaların aşırı üretimi nedeniyle kapasite kaybını önlemek için kullanımı kolay ama etkili bir atölye kontrol sistemi tasarlamaktır. Önerilen kontrol sistemi stokastik talebe cevap vermek amacıyla stok seviyelerini anlık güncelleyen çekme tip sistem kullanan bir dinamik çizelgeleme yaklaşımını içermektedir.

Dinamik çizelgeleme yaklaşımı olarak olay tabanlı algoritmalar esas alınmıştır. Önerilen algoritmalar siparişleri, stokları, En Erken Tamamlanma Zamanını ve En Kısa İşlem Süresini dikkate alan üç adımlı sınıflandırma ve sıralama prosedürüne sahiptir.

Çok modelli montaj aşamasının izlediği iki kademeli esnek akış atölyesi üretimini modelleyen bir benzetim ortamında önerilen Algoritma 1, 2 ve 3 ile İlk Gelen İlk İşlenir, Toplam En Küçük İşleme Süresi, En Kısa Dar Boğaz İşleme Süresi ve Tamamlanma Zamanı çizelgeleme yöntemleriyle karşılaştırılmıştır. En iyi çizelgeleme yöntemini belirlemek için Bonferroni yaklaşımı kullanılmıştır. Yapılan testlerde önerilen Algoritma 1, 2 ve 3, diğer klasik çizelgeleme yaklaşımlarına üstünlük sağlamıştır. Sipariş ve stok kontrolü yapan Algoritma 2 ise en iyi sonucu vermiştir.

Anahtar Kelimeler : Esnek İş Atölyesi, Dinamik Çizelgeleme, Stokastik Talep, En Erken Tamamlanma Zamanı, En Kısa İşlem Süresi , Olay Tabanlı Algoritma, Kanban, Çekme Sistemi.

ABSTRACT

MSc Thesis

DYNAMIC FLEXIBLE JOB SHOP SCHEDULING UNDER STOCHASTIC DEMAND

Eraydın GENÇ

Uludağ Üniversitesi

Graduate School of Natural and Applied Sciences

Department of Industrial Engineering

Supervisor: Prof. Dr. Erdal EMEL

The main goal of this thesis study is to devise an easy-to-use but effective shop floor control system to prevent capacity losses due to line stoppages, job queues and overproduction of parts. The thesis offers to use a dynamic scheduling approach responsive to stochastic demands as working under a pull type system using instant update of inventory levels.

An event-based algorithmic approach is proposed for dynamic scheduling. Consisting of a three step sorting and sequencing procedure that considers orders, inventories, Earliest Due Date (EDD) and Shortest Process Time (SPT).

A simulation model simulating a two-stage flexible flow shop production followed by a multi model assembly stage is used to compare the proposed Algorithms 1, 2 and 3 with First Come First Served (FCFS), Total Shortest Process Time (STPT), Shortest Bottleneck Process Time (SBPT) and Due Date (DD). Bonferroni approach was used to select best method. The proposed Algorithms 1,2 and 3 gave better results superior to other well-known sequencing approaches. Algorithm 2, which makes order and stock controls gave best result.

Key Words: Flexible Job Shop, Dynamic Scheduling, Stochastic Demand, Earliest Due Date (EDD), Shortest Process Time (SPT), Event-based algorithm, Kanban, Pull System.

ÖNSÖZ ve TEŞEKKÜR

Üretim işletmelerinin rekabet ortamında önde yer almaları üretim maliyetlerini kontrol edebilmeleriyle doğrudan ilişkilidir. Sanayi – Üniversite işbirliği maliyetleri kontrol altına alma ve azaltma yönünde işletmelere geniş fırsatlar sunabilmektedir. Bu fırsatlardan faydalanmanın yollarından biri çalışanların akademik eğitime devam etmesi ile güncel akademik bilginin işletme içine akmasına olanak sağlanmaktadır. Bu tez, konusunu gerçek bir üretim probleminden almış ve ürettiği çözümün kullanılmasıyla işletmelerin sağlayabileceği faydalara örnek oluşturmuştur.

Tez çalışması sonucu ortaya çıkarılan algoritma bir proje dahilinde yazılım haline getirilmiş ve gerçek üretim ortamında kullanılmaya başlanmıştır (2014). Yapılan çalışma YAD Uygulama Ödülleri (2014) kapsamında sunulmuştur. Uygulama ve sonuçları Uludağ Üniversitesi Endüstri Mühendisliği Bölümü adına EURO The Association of European Operational Research Societies konferansında (EURO 2015, 12-15 July 2015, Glasgow) sunulmuştur.

Yüksek Lisans eğitimi almama olanak sağlayan Ficosa Otomotiv'e ve gerçek üretim ortamında yapılan uygulamaya katkı sağlayan çalışanlarına teşekkürlerimi sunarım.

Yüksek Lisans eğitimi ve tez çalışmasının tüm gelişim aşamalarında, yapılan sunumlarda, bilgi ve yönlendirmeleri, sağladığı akademik imkanlar ve verdiği çok değerli zamanı ile hedefe ulaşılmasını sağlayan Sayın Prof. Dr. Erdal Emel'e özel teşekkürlerimi sunarım.

İÇİNDEKİLER

	Sayfa
ÖZET.....	i
ABSTRACT.....	ii
ÖNSÖZ.....	iii
SİMGELER VE KISALTMALAR.....	vi
ŞEKİLLER DİZİNİ.....	viii
ÇİZELGELER DİZİNİ.....	x
1. GİRİŞ	1
1.1. Çalışmanın Amacı.....	1
1.2. Kullanılan Metodoloji.....	3
1.3. Tezin Organizasyonu	4
2. KAYNAK ÖZETLERİ	5
3. MATERYAL ve YÖNTEM.....	9
3.1. Problemin Tanıtılması.....	9
3.2. Hedeflerin Tanımlanması.....	10
3.3. Matematiksel Tanımlamalar.....	11
3.3.1. Denklemler.....	11
3.3.2. Denklemlerin kartlı sistem ve kutu içi miktarlar ile düzenlenmesi.....	13
3.3.3. Kart adetlerinin hesaplanması	14
3.3.4. Bekleme sürelerinin hesaplanması.....	16
3.4. Önceliklendirme Yaklaşımı	17
3.4.1. Sıralama Yaklaşımları.....	17
3.4.2. SPT öncelik değerleri tablosu	18
3.4.3. Sınır katmanlarının belirlenmesi.....	18
3.4.4. Sıralama Algoritması	18
3.5. Uygulama.....	23
3.5.1. Üretim Ortamına Ait Veriler.....	23
3.5.2. Montaj Hattı Talep Verilerinin Analizi ve Kanban Kart Sayı Hesaplanması.....	26
3.5.3. Algoritmanın Test Edilmesi	33
3.6. Benzetim Modelleri.....	34
3.6.1. İlk Gelen İlk İşlenir (FCFS) Benzetim Modeli	34
3.6.2. En Küçük Kesme Süreli (SBPT) Benzetim Modeli.....	50

3.6.3. En Küçük Toplam İşlem Süraeli (Total SPT) Benzetim Modeli	50
3.6.4. Stok Bitiş Zamanı (Due Date) Benzetim Modeli	51
3.6.5. Algoritma 1, Sipariş Kontrollü Sıralama Benzetim Modeli.....	51
3.6.6. Algoritma 2, Sipariş ve Sıfır Stok Kontrollü Sıralama, Benzetim Modeli	52
3.6.7. Algoritma 3, Sıfır Stok Kontrollü Sıralama, Benzetim Modeli	53
3.7. Benzetimlerde Kullanılan Veriler	54
4. BULGULAR ve TARTIŞMA.....	57
4.1. Değişken Sipariş Miktarları ile Yapılan Test Sonuçları	57
4.2. Alt – Üst Sınır Diyagramı Üzerinde Karşılaştırma	59
4.3. Bonferroni En İyiyi Seçme Yaklaşımı	61
4.3.1. Prosedürün Benzetim Sonuçlarına Uygulanması.....	63
4.4. Sabit Sipariş ile Yapılan Test Sonuçları.....	66
5. TARTIŞMA VE SONUÇ	70
KAYNAKLAR.....	74
EKLER.....	75
ÖZGEÇMİŞ.....	100

SİMGELER ve KISALTMALAR DİZİNİ

Simgeler	Açıklama
i	Operasyon
j	Parça (Yarı mamul) referansı
m_i	i operasyonu makine adedi
P_{ij}	j . referansının i . operasyon işleme süresi (dakika)
τ_j	j . referansının geçiş süresi (dakika)
d_j	Teslim zamanı
C_j	j . referansının tamamlanma zamanı (gun-saat-dakika)
L_j	j . referansının gecikme süresi (dakika)
D_j	j . referansının talebi (adet)
D_{jh}	j . referansının h haftasındaki talebi (adet-hafta)
t_j^m	j referansının m . aşamaya girme/başlama zamanı (gun-dk-saniye)
t_j^{lm}	j referansının m . aşamaya son girme/başlama zamanı (gun-dk-saniye)
α	Her t_j^m oluşmasında bir artan hareket sıra numarası
$D'_j(\alpha)$	α anında j referansının haftalık kalan talep miktarı (adet)
$k_j(\alpha)$	α anında j referansından üretimi devam eden miktar (adet)
W_{ij}	j referansının i . operasyon kuyruğunda bekleme zamanı (dakika)
S_{ij}	j referansının i . operasyon setup zamanı (dakika)
V_j	j referansının montaj hattı tüketim hızı (adet/dakika)
Q_j	j referansının kutu içi miktarı (adet)
$Q'_j(\alpha)$	j referansının montaj hattında kutu içi kalan miktarı $\leq Q_j$ (adet)
$I_j(\alpha)$	α anında j referansının depodaki miktarı (adet)
I_j^{min}	j referansının en küçük stok miktarı (adet)
τ_j^{Imin}	j referansının en küçük stok miktarının tüketim süresi (dakika)
K_j	j referansının kanban kart adedi (adet)
$\tau_j^{WIP}(\alpha)$	α anında j . referansın WIP süreç içi stoklarının tükenme süresi (dakika)
τ_j^K	j referansının bir kart = kutu stok tükenme süresi (dakika)
$WIP_j(\alpha)$	α anında j . referansının süreç içi stok miktarı (adet)
$WIP_j^K(\alpha)$	Dolu kutular için, α anında j . referansının süreç içi stok miktarı
$K_j^S(\alpha)$	j referansının, α anında depolanmış kart = kutu adedi

$K_j^P(\alpha)$	j referansının, α anında üretimi devam eden kart = kutu adedi
l_j	j referansının ilk operasyona en geç başlama zamanı (dakika)
$WSPT_j^1$	j referansının Operasyon_1 SPT Öncelik Değeri
OP_j^α	j. referansının, α anında Operasyon_1'e girme Öncelik Değeri
OS_{layer}	Belirlenen her katman için Öncelik Sınır değeri.
SPT	En Kısa İşlem Süresi (Shortest Process Time)
FCFS	İlk Gelen İlk İşlenir (First Come First Served)
STPT	En Kısa Toplam İşlem Süresi (Total Shortest Process Time)
DD	Teslim Zamanı (Due Date)
SBPT	En Kısa Darboğaz İşlem Süresi (Shortest Bottleneck Process Time)
GA	Genetik Algoritma

ŞEKİLLER DİZİNİ

	Sayfa
Şekil 1.1. Uygulama Konusu Üretim Ortamı.....	3
Şekil 3.1. Üretim Ortamı.....	9
Şekil 3.2. Tasarlanan Üretim Ortamı	22
Şekil 3.3. Uygulama Konusu Üretim Ortamı.....	23
Şekil 3.4. Benzetim Modelinin Promodel Ekran Görüntüsü	34
Şekil 3.5. Makineler	35
Şekil 3.6. Kesme Kuyruğu Karar Kuralları.....	35
Şekil 3.7. Kesme Makinesi Karar Kuralları	36
Şekil 3.8. Elemanlar (Entities)	36
Şekil 3.9. Yol bağlantıları (path networks) tanımlamaları.....	37
Şekil 3.10. Net_Operator yolu (Path) tanımlaması	37
Şekil 3.11. Kaynaklar	38
Şekil 3.12. Operator kaynak kullanım tanımlaması (Specifications).....	38
Şekil 3.13. Process (İşlemler) ve Routing (Rotalama) pencereleri	39
Şekil 3.14. Kart – KesKuy Rotalama Penceresi.....	39
Şekil 3.15. Kart – KesKuy Operasyon Penceresi.....	40
Şekil 3.16. Kart – KesKuy Rotalama Penceresi.....	41
Şekil 3.17. Kart – KesmeMak Operasyon Penceresi	41
Şekil 3.18. Kart – KesmeMak Rotalama Penceresi	42
Şekil 3.19. Kart – CapKuy Rotalama Penceresi	42
Şekil 3.20. Kart – CapMak Operasyon Penceresi	43
Şekil 3.21. Kart – CapMak Rotalama Penceresi	43
Şekil 3.22. Kart – Raf Operasyon Penceresi	43
Şekil 3.22. Kart – Raf Operasyon Penceresi (Devam).....	44
Şekil 3.23. Kart – Raf Rotalama Penceresi	44
Şekil 3.24. Varış (Arrivals) penceresi	45
Şekil 3.25. Öznitelik Değişkenleri (Attributes) Penceresi	45
Şekil 3.26. Değişkenler (Variables) Penceresi.....	46
Şekil 3.27. Diziler (Arrays) Penceresi.....	47
Şekil 3.28. Dış Dosyalar (External Files) Penceresi	49

Şekil 4.1. Tüm Yöntemler Alt - Üst Sınır Diyagramı	60
Şekil 4.2. Algoritma 1,2 ve 3 Alt - Üst Sınır Diyagramı	60
Şekil 4.3. Algoritma 2 ve 3 Alt - Üst Sınır Diyagramı	61
Şekil 4.4. Algoritma 1, 2 ve 3 Alt - Üst Sınır Diyagramı (Sabit Sipariş)	69

ÇİZELGELER DİZİNİ

Sayfa

Çizelge 3.1. Setup süreleri (Dakika)	24
Çizelge 3.2. Operasyon ve Tüketim Süreleri(Dakika).....	25
Çizelge 3.3. Haftalık Yarı Mamul Sipariş Kasa Adetleri.....	26
Çizelge 3.4. Genel üretim planı verileri	27
Çizelge 3.5. Operasyon genel verileri	29
Çizelge 3.6. Operasyon 1, makine başına iş yükü analizi ve kuyruk uzunluğu tahmini.	30
Çizelge 3.7. Operasyon 2, makine başına iş yükü analizi ve kuyruk uzunluğu tahmini.	30
Çizelge 3.8. Kanban Kart Adetleri	31
Çizelge 3.9. Test Verileri Özeti	55
Çizelge 4.1. Benzetim Sonuçları (Değişken Sipariş).....	58
Çizelge 4.2. Montaj Hattı Duruş Oranları (%) (Değişken Sipariş).....	58
Çizelge 4.3. Montaj Hattı Duruş Süreleri (dak/10 000 dak) (Değişken Sipariş)	59
Çizelge 4.4. İlk Test İstatiksel Sonuçlar (Dakika) (Değişken Sipariş)	59
Çizelge 4.5. Bağlı Örneklem Sonuç Tablosu	63
Çizelge 4.6. Algortima 2 ve 3 İlave Benzetim Sonuçları (Değişken Sipariş).....	64
Çizelge 4.7. Algortima 2 ve 3 İlave Benzetim Sonuçları, Montaj Hattı Duruş Oran	65
Çizelge 4.8. 10 000 dk ile Genişletilmiş Montaj Hattı Duruş Süreleri	65
Çizelge 4.9. Benzetim Sonuçları (Sabit Sipariş).....	66
Çizelge 4.10. Montaj Hattı Duruş Oranları (%) (Sabit Sipariş).....	67
Çizelge 4.11. Montaj Hattı Duruş Süreleri (dak/10000 dak) (Sabit Sipariş)	68
Çizelge 4.12. İstatiksel Sonuçlar (Dakika) (Sabit Sipariş).....	69

1. GİRİŞ

1.1. Çalışmanın Amacı

Seri üretim ortamında faaliyet gösteren işletmelerde müşteri siparişlerini karşılamak üzere üretim ve malzeme ihtiyaç planlaması yapılmakta ve bu plana göre üretim sürecinin yürümesi beklenmektedir. Ürünün doğru zamanda, doğru miktarda ve doğru kalitede hazır bulunması ana hedefiyle oluşturulan bu plan üründen geriye doğru dallandırılarak yarımamül üretimlerini ve komponent tedariklerini şekillendirmektedir. Süreçlerin girdi ve çıktılarında oluşan birçok değişkenlikten dolayı yapılan planlamaya tam uymak genellikle mümkün olamamaktadır. Tedarik ve müşteri süreçleri şeklinde birçok sürecin arka arkaya işlev gördüğü seri üretim ortamında plana uyulamaması beraberinde kapasite ve kaynak kayıplarına, stok seviyelerinde yetersizlik ve aşırılık oluşturan geniş dalgalanmalara ve dolayısıyla ek maliyetlere neden olmaktadır. Başlangıçta oluşturulan üretim planına uyulamaması ilgili dönem içerisinde anlık olarak üretim planlarının yenilenmesi gerekliliğini ortaya çıkarır.

Üretim planı yapmak yerine üretim birimleri arasındaki malzeme akışını çekme sistemi prensiplerine göre düzenlemek tercih edilen uygulamalardan biridir. Öte yandan çekme sistemi talep değişkenliğinin yüksek olmadığı, öncel (tedarikçi) ile müşteri süreçleri arasında üretim hızı dengesinin iyi kurulduğu ortamlarda iyi sonuçlar vermektedir. Müşteri süreçlerinin değişken siparişleri ve vardiyalar arasında dengeli olmayan kapasite kullanımı nedeniyle üretim ve tüketim hızlarının değişkenlik gösterdiği üretim ortamlarında çekme sistemi FCFS (ilk gelen ilk işlenir) prensiplerine göre çalıştırıldığında stok yönetimi açısından olumsuz sonuçlar ortaya çıkmaktadır. Böyle bir ortamda FCFS kurallarına göre üretim sıralaması yapıldığında süreçler arası tedarik zincirinin kopmaması için yüksek ara stok seviyelerine ihtiyaç duyulacaktır.

Ana konularından biri Üretim ve Kaynak Planlaması olan Endüstri Mühendisliği, birçok araç ve yaklaşımıyla Üretim Sıralama ve Çizelgeleme konularıyla da ilgilenmektedir. Deterministik ve stokastik ortamlarda uyguladığı sıralama ve çizelgeleme yaklaşımlarıyla amaca uygun çözümler ortaya koymaktadır.

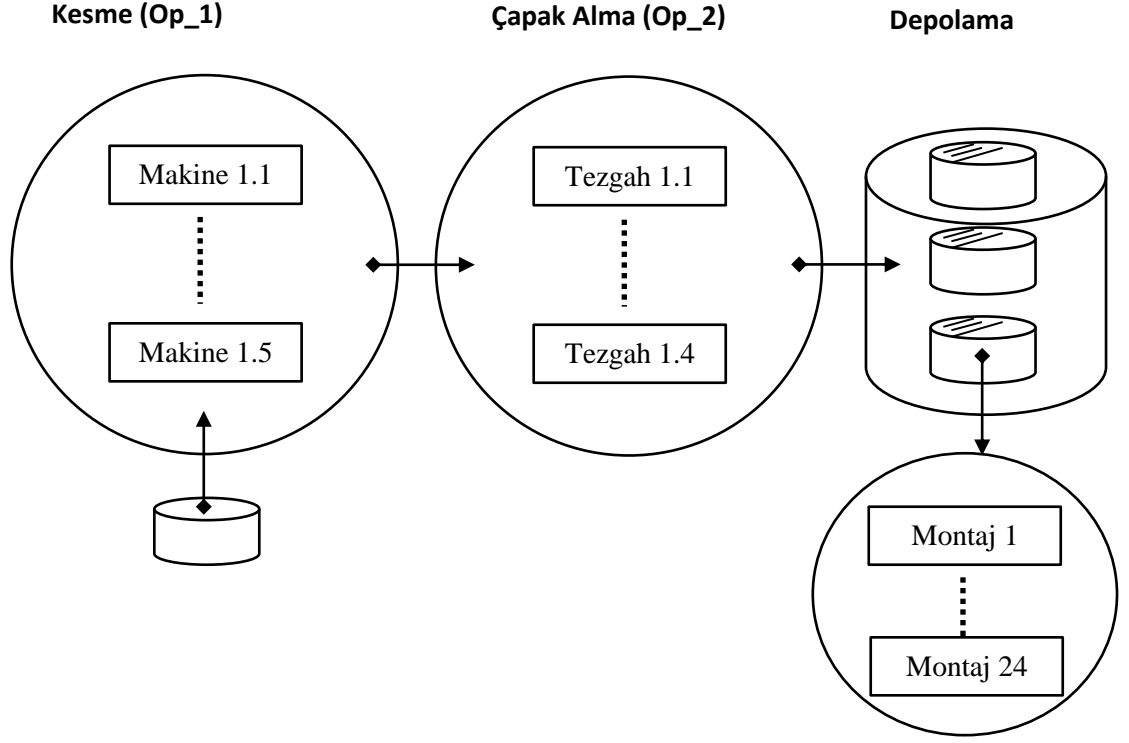
Birden çok operasyonun alternatif paralel makinelerden birinde işlenerek sırayla ilerlediği üretim şekli Esnek Atölye Tipi (Flexible Job Shop) olarak tanımlanmaktadır. Özellikle deterministik talep ortamında Esnek Atöyle Tipi üretimin çizelgelenmesi için birçok model önerilmiştir. Üretilen modellerin sağlıklı çalışması için öncelikle deterministik verilerin üretim esnasında değişmemesi gerekir. Bunun yanında çok aşamalı üretim ortamında her makineye işin çizelgelemeye göre yüklenmesi beklenir. Deterministik ortamın bazı nedenlerden dolayı bozulduğu durumlarda (örneğin bir komponentin eksikliği nedeniyle) ve/veya üretim çizelgelemenin uygulanmasının çeşitli nedenlerle aksadığı durumlarda (örneğin koordinasyon eksikliği) başlangıçta oluşturulan çizelge pratikte uygulanamaz.

Yukarıda ifade edilen olumsuzlukların gerçek üretim ortamında ortaya çıkması oldukça olası bir durumdur. Bu olumsuzlukların yaşandığı, kartlı çekme sistemi ile üretim ve tüketimlerin bildirildiği gerçek bir üretim ortamını uygulama sahası alan, stokastik talep ortamında anlık gerçekleşen tüketimlere tepki olarak üretim sıralamasını yeniden oluşturan bir modeli ortaya koymak tezin amacını oluşturmaktadır.

Tez kapsamına temel oluşturan problem gerçek bir üretim ortamından alınmıştır. İlgili üretim ortamında seri üretim yapan montaj hatlarında kullanılmak üzere yarı mamul üreten birimde, 5 adet özdeş makineden birinde Operasyon_1'den geçen malzeme 4 adet özdeş makinenin birinde Operasyon_2'den geçerek yarı mamul halini almakta ve aynı yerde bulunan depolama bölgesine konulmaktadır. Nihai müşteri siparişlerinin değişkenliği, vardiyalar arası dengeli olmayan kapasite kullanımı ve anlık üretim planı değişiklikleriyle stokastik hale gelen ihtiyaç doğrultusunda 64 çeşit yarımamul, 24 adet montaj hattına sevk edilmektedir. Şekil 1.1'de üretim ortamı temsil edilmiştir.

Mevcut durumda yarı mamul üreten birimde üretim sıralaması İlk Gelen İlk İşlenir (First Come First Served - FCFS) genel yaklaşımı içerisinde montaj hatlarıyla işçiler aracılığıyla kurulan iletişimin desteği ile yapılmaktadır. İletişimin gerekliliği FCFS yaklaşımının sıralama metodu olarak ihtiyacı karşılamamasından ileri gelmektedir. Yapılan analizlerde ilgili yarı mamul üretim biriminin montaj hattı duruşlarının %21'ine neden olarak montaj hatlarının en kötü tedarikçisi olduğu görülmektedir. Öte yandan

süreç içi genel stoğu 4 gün civarındadır. Sistem genel olarak yüksek stok bulundurduğu halde çok sıklıkla bazı yarı mamüllerde sıfır stok düzeyine düşülmekte, bu nedenle montaj hattı duruşları gerçekleşmektedir.



Şekil 1.1. Uygulama Konusu Üretim Ortamı

Stokastik talep ortamında üretim sıralamasını güncellerken, süreç içi stok miktarını yükseltmeyen ve üretim durum riskini en küçükleyen bir yöntem ihtiyacı vardır. Tez kapsamında böyle bir yöntem üretilmiş ve diğer bazı genel kabul görmüş sıralama yaklaşımlarıyla kıyaslanmıştır.

1.2. Kullanılan Metodoloji

Test yöntemi olarak benzetim modelleme kullanılmıştır. Simülasyon, teorik ya da gerçek bir sisteme ait neden sonuç ilişkilerinin bir bilgisayar modeline yansıtılmasıyla, değişik koşullar altında gerçek sisteme ait davranışların bilgisayar modelinde izlenmesini sağlayan bir yöntemdir. Tasarım halindeki sistemleri test etmek üzere gerçek deneyler yapmanın uygun olmadığı, çok maliyetli olduğu ve sistem davranışlarını analiz etmek amacıyla veri elde edilmesi gerektiği durumlarda benzetim yöntemleri ekonomik ve hızlı

bir çözüm oluşturmaktadır. Girdi, süreç ve çıktılarında yapılacak değişikliklerle bilgisayar ortamında hazırlanan benzetim modelleri sistem davranışını çok geniş bir alanda ve düşük maliyetle analiz etme imkanı sağlamaktadır. Tez kapsamında yeni bir sıralama yöntemi tasarlanmıştır. Tasarlanan yöntemin diğer sıralama yaklaşımlarıyla kıyaslanması, alternatif yöntemlerin sınanması ve testler esnasında veri elde edilebilmesi için benzetim modelleme yöntem olarak tercih edilmiştir.

1.3. Tezin Organizasyonu

Tez çalışması dört ana bölümden oluşmaktadır. Kaynak Özetleri bölümünde tez konusuna benzer özellikleri olan problemler üzerine yapılan çalışmalar ve çözüm yöntemleri incelenmiş ve özetler haline sunulmuştur. Tezin Materyal ve Yöntem bölümünde problem ve hedefler tanımlandıktan sonra çözüme temel oluşturacak olan matematiksel tanımlamalar yapılmıştır ve tezin ortaya koyduğu yöntemin denklemleri açıklanmıştır. Bölümün devamında çözüm yöntemi olan sıralama algoritması anlatılmıştır. Takip eden Uygulama bölümünde üretim ortamına ait veriler ortaya konduktan sonra matematiksel hesaplamalar yapılmıştır ve benzetim modelleri tanıtılmıştır. benzetim modellerinin çalıştırılması ile elde edilen sonuçlar ve değerlendirmeler Bulgular ve Tartışma bölümünde ele alınmıştır.

Tez çalışması sonucunda hangi yarı mamulün üretime (Operasyon_1) alınacağına anlık karar veren bir sıralama algoritması ortaya çıkarılmıştır. Algoritma temel olarak işlerin en geç başlama sürelerini kıyaslamakta ve En Kısa İşlem Süresi (Shortest Process Time - SPT)'ne göre sıralama yapmanın kuyruk azaltan özelliğinden faydalanmaktadır. Algoritmanın sipariş kontrolü yapan, sıfır stok kontrolü yapan ve hem sipariş hem sıfır stok kontrolü yapan üç türevi oluşturulmuştur. Algoritma türevleri, benzetim ortamında İlk Gelen İlk İşlenir (FCFS), En Kısa Toplam İşlem Süresi (STPT), Teslim Zamanı (DD) ve En Kısa Darboğaz İşlem Süresi (SBPT) sıralama yaklaşımlarıyla kıyaslanarak test edilmiştir. Diğer sıralama yaklaşımlarına üstünlük sağladığı görülen algoritma gerçek üretim ortamında kullanılmaya başlanmıştır.

2. KAYNAK ÖZETLERİ

Tez çalışması ile ulaşılmak istenen yöntem atölye tipi üretim ortamında sürekli gelen iş için dinamik bir üretim çizelgeleme yaklaşımıdır. Bu konu hakkında yapılan çalışmalar çözümün geliştirilmesinde önemli yer tutmaktadır. Üretim programının kesin olmaması nedeniyle yaklaşım stokastik bir model üzerine kurulmak durumundadır.

Bir çoğu statik çizelgeleme üzerine olmakla beraber, önerildiği 1985 yılından bu yana, Genetik Algoritma, atölye çizelgeleme problemlerinde sıkça kullanılır olmuştur (Lin ve ark. 1997). Lin ve arkadaşları (1997) ilgili makalede statik atölye çizelgeleme problemi için kullanılan GA yaklaşımını deterministik model için geliştirmişlerdir. Yazarlar, Raman'ın (Raman ve ark. 1989) önerdiği modeli kullanarak stokastik problemi deterministik problem serileri içerisinde eriterek çözüme ulaşmışlardır. Stokastik modelin deterministik seriler gibi düşünülmesi deterministik modellerin probleme uygulanmasının önünü açabilir (Koulamas 2010).

Süreçler arası üretim hızı farklılıkları, değişkenlik gösteren müşteri süreç siparişleri ve vardiyalar arası kapasite kullanımındaki farklılıkların bir sonucu olarak gerçek üretim ortamında bazı işler tam zamanında yapılamayacak ve gecikecektir. Gecikmelerin yenilenen sıralamaya nasıl yansıtılacağı diğer önemli bir konudur. C.Koulamas (2010) geciken işlerle tek makine teslim zamanı atamasını konu aldığı makalesinde ağırlıklandırılmış gecikme indikatörü kullanarak her iş için teslim zamanının optimum değerini bulmayı amaçlamıştır. Bu noktada ağırlıklandırmanın nasıl yapılacağı teslim zamanlarının atanmasında önemli bir konu olarak karşımızda durmaktadır.

Problemin bir kısıtı olarak değiştirilmez lot miktarları göz önünde bulundurulmaktadır. Lot miktarlarının çevrim zamanına tesiri, problemin önemli kısıtlayıcılarından. Filho ve Uzsoy (2011) bu ilişki üzerine yaptıkları çalışmada lot miktarı, çevrim zamanı ve üretim parametreleri arasındaki ilişkinin oldukça karmaşık ve doğrusal olmayan etkilere sahip olduğunu, lot miktarının azaltılmasının üretim parametrelerinin değiştirilmesiyle elde edilen çevrim zamanı iyileşmelerini artırdığını, buna karşılık lot miktarı değiştirme olanaklarının az olmasının iyileştirme çalışmalarının faydalarını olumsuz etkilediğini ortaya koymuşlardır.

Çözümüne ilişkin çalışmalar gözden geçirildiğinde Esnek Akış Atölyesi problemlerinin sezgisel çözümünde Genetik Algoritmanın yanı sıra Karınca Kolonisi, Tabu Araştırma, Geometrik Algoritma kullanılan çalışmalar da bulunmaktadır.

Yine benzer bir çalışmada Çoklu Ajan (Multiagent) Çizelgeleme Methodu esnek akış atölyesi probleminde erken ve geç tamamlanma zamanlarının optimizasyonu için kullanılmıştır (Wu ve Weng 2005). Çalışmada Tam Zamanında Üretim (JIT) prensiplerine uygun olarak stok seviyesinin düşürülmesi ve gecikme nedeniyle stoksuz kalma riski aynı amaç içerisinde birleştirilmiştir. Ele alınan problemde vardiyalar arası değişken kapasite kullanımının bir sonucu olarak tedarikçi ve müşteri süreçler arasında tüketim hızının üretim hızından yüksek ve düşük olduğu vardiyalar olacaktır. Bu durumda yüksek tüketimin olduğu döneme yüksek stok ile girme zorunluluğu ortaya çıkacaktır. Böyle bir zorunluluk altında erken tamamlamaları azaltmayı amaçlardan biri haline getirmek çok uygun görünmemektedir.

W.Xiao (ve ark. 2000) yayımladıkları makalelerinde hibrit akış atölyesi probleminde genetik algoritmayı tüm işlerin tamamlanma zamanının en küçüklenmesini performans kriteri olarak uygulamışlardır. Makalede ifade edilen hibrit akış atölyesi problemi herhangi kademesinde çoklu makinelerin olduğu akış atölyesindeki işlerin çizelgelenmesi olarak nitelenmiştir (Xiao ve ark. 2000). Algoritma temeli, iş sıralamalarını geliştirerek yapılan liste çizelgeleme prensibine dayanmaktadır (Xiao ve ark. 2000). Makalede iş sıralaması sadece ilk seviye üretim kademesinde yapılmakta, diğer kademelerde işler FIFO (ilk gelen ilk çıkar) prensibine göre ilerlemektedir. Başlangıç anında bir iş kuyruğu vardır ve bu işler ilk kademe görev için kullanılan M_1 adet makinenin birinde gerçekleştirilebilir (Xiao ve ark. 2000). İlgili makalede amaç tüm işlerin tamamlanma süresini (makespan) en küçükleme olarak verilmiştir. Tez kapsamında aynı amacın kullanılabilirliği şüphelidir. Amaç tüm işleri en az zamanda bitirmek değil, her işi teslim zamanından önce yani müşteri süreç/hat ihtiyaç duymadan bitirmektir. Ayrıca makale deterministik talep ortamında bir çözüm üretirken tez konusu problem stokastik talep ortamında dinamik bir çözüm gerektirmektedir.

Esnek atölye tipi üretim (FJc) atölye tipi üretim ve paralel makine ortamının genel ifadesidir. Her iş kendi rotasını izleyerek her istasyonda (iş merkezinde) herhangi makinenin yalnız birinde işlem görür (Pinedo 2008, s:16). Pinedo'nun Çizelgeleme kitabında çizelgeleme problemlerinin sınıflandırılmasında ilişkin tanımlamalar ve örnekler bulunmaktadır (Pinedo 2008 s:20). Tez konusu üretim ortamı; birden çok istasyonun (c) yer aldığı, her istasyonda paralel makinelerin (M) bulunduğu, her işin bir makinede yalnızca bir defa yapıldığı bir atölye olarak tanımlanabilir. Her işin bir teslim zamanı (d) vardır ve bağlı olarak bir başlama zamanı (r) olacaktır. İş sıralamasından bağımsız olarak makineye göre her işin setup zamanı (s) vardır. Amaç ağırlıklı (w) geç kalmaları (T) en küçüklemektir. Bu durumda standart notasyonumuz $FJc / r_j, s_{ijk}, M_j / \sum w_j T_j$ şeklinde tanımlanabilir.

M adet paralel makinenin bulunduğu istasyonda bir makine işini bitirdiğinde üretimi yönlendiren karar verici bu makineye hangi işi atayacağına veya makineyi bir süre için boş bırakıp bırakmayacağına karar verir. Bu kararı verirken mevcutta yapılan işlerin kalan işlem sürelerini ve kaç tane işin daha işlenmek üzere geleceğini bilmemektedir. Bu tip problemlerin çözümünde iyi bilinen yöntemlerden biri Liste Çizelgeleme Algoritmasıdır (List Scheduling Algorithm). Karar verici elindeki listeye göre sıradaki işi makineye atar (Pinedo 2008, s:139). Tezin konusunu oluşturan problemde ilk aşamada kuyrukta işler birikecektir. İkinci aşama FCFS (ilk gelen ilk işlenir) kuralına göre çizelgelenmektedir. İlk aşamada kuyrukta bekleyen işlerin oluşturduğu listenin bir kurala göre düzenlenerek üretim sırasında dönüştürülmesi gerekecektir. Bu kuralın her tedarikçi süreç çıktısı yarı mamul için belirlenen en büyük stok miktarından, yarı mamullerin tüketildiği müşteri sürecin/hattının üretim hızından ve talep değişimlerinden etkilenmemesi söz konusu değildir.

Çizelgeleme teorisinin temel ilkelerinden biri en küçük işlem zamanına (SPT) göre sıralamanın ortalama üretim hızını en büyükleyeceği ve süreç içi stokları en küçükleyeceğidir (Berkley, 1993). Berkley 1993 yılında yayımladığı makalesine benzetim metodları kullanarak tek kartlı kanban sisteminde FCFS sıralamasını SPT'den daha iyi ortalama üretim hızı oluşturduğunu göstermiştir. Yine aynı çalışmada periyodik

malzeme işlemeyen iki kartlı sistemde SPT'nin FCFS'e göre daha büyük ortalama stok değerleri ürettiği görülmüştür.

Parçanın ne zaman tükeneceğini dikkate almayan en küçük işlem süreli olanı sürekli en önce üretme (SPT) yaklaşımının nispeten uzun işlem süreli işleri sürekli öteleyeceğini ve dolayısıyla stok düzeylerini düzensizleştireceğini görmek zor değildir. Öte yandan SPT'nin küçük işlem süreli işleri önce işleme alarak kuyruk uzunluğunu azaltacağı gerçeği hala yerindedir. İşlerin stoksuzluk yaratacak düzeyde geç kalmasına neden olmayacak şekilde SPT yöntemini düzenleyen bir hesaplama yapısıyla SPT'nin kuyruk uzunluğunu azaltan gücü kullanılabilir.

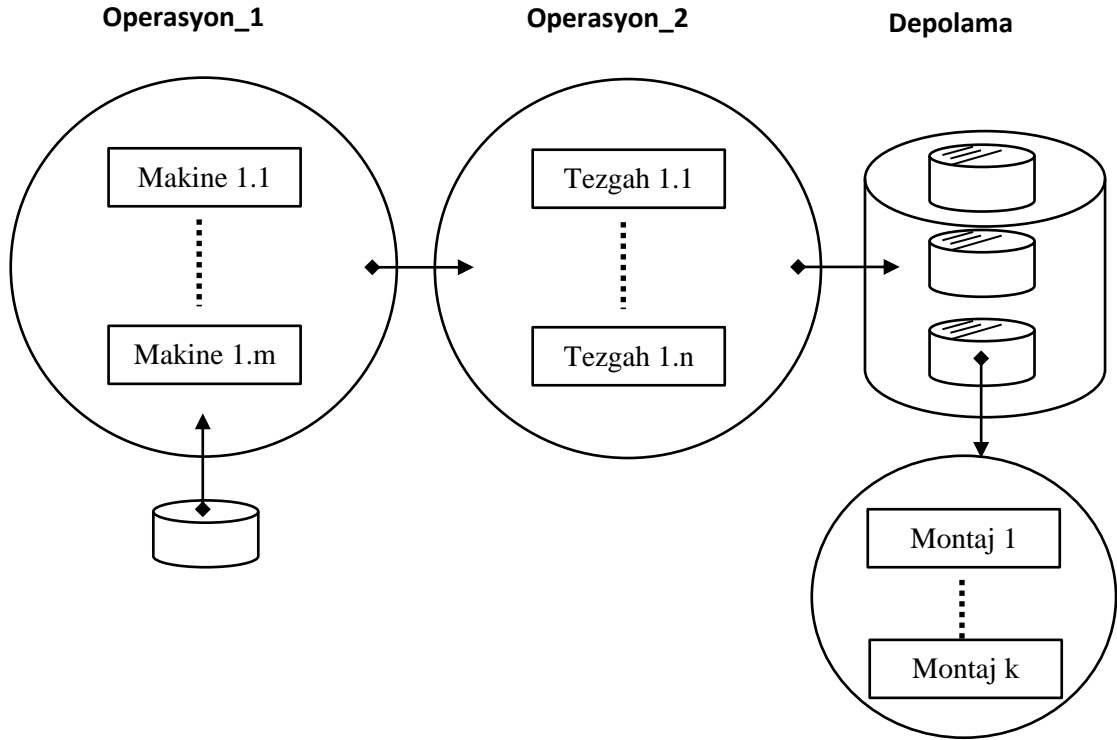
Bu düşünceyi destekler şekilde Lee 1987'de yayımladığı makalesinde SPT/LATE olarak tanımladığı sıralama kuralında; “kuyruktaki işlerden hiçbiri geç kalmadığı takdirde SPT kuralı uygulanabilir. Aksi takdirde önceliklendirme geç kalan işlerin miktarına göre yapılmalıdır.” demektedir. Aynı makalede yüksek çekme frekansına göre sıralamanın SPT/LATE kuralına göre daha zayıf sonuçlar ürettiği ifade edilmiştir.

Tez konusu olan üretim ortamında iş çeşitliği fazladır. Bu durum sistem performansı için önemli bir parametredir. SPT/LATE kuralı iş çeşitliliğinin arttığı noktada nispeten daha iyi sonuçlar vermiştir (Lee, 1987).

3. MATERYAL ve YÖNTEM

3.1. Problemin Tanıtılması

Seri üretim yapan montaj hatlarında kullanılmak üzere yarımamul üreten birimde parçalar iki operasyondan geçmektedir. Operasyon_1, m adet özdeş makineden birinde yapılmaktadır. Operasyon_1'den geçen parçalar n adet özdeş tezgahın birinde Operasyon_2'den geçerek yarımamul halini almakta ve aynı yerde bulunan depolama bölgesine konulmaktadır. Montaj hatlarının talebine göre depolama bölgesinden alınan yarımamul ilgili montaj hattına götürülmektedir (Şekil 3.1.).



Şekil 3.1. Üretim Ortamı

Nihai müşterinin ürün siparişleri geliş frekansı ve miktar olarak çeşitlilik göstermektedir. Sürekli üretilen ürünler olduğu gibi siparişleri kesikli biçimde haftalık veya aylık olarak gelen ürünlerde bulunmaktadır. Firmanın montaj hatlarının tamamı gündüz vardiyasında üretim yaparken akşam ve gece vardiyalarında müşteri siparişine göre gerekli olan montaj hatlarında üretim yapılmaktadır. Gündüz vardiyasında montaj hatlarının toplam yarı mamul tüketim hızı, toplam yarı mamul üretim hızından yüksek iken akşam ve gece

vardiyalarında düşüktür. Ortalama değerlere bakıldığında yarı mamul üretim biriminin kapasitesi günlük tüketimi karşılar düzeydedir. Değişken müşteri siparişleri, vardiyalar arasında dengeli olmayan kapasite kullanımı, yarı mamul üretim biriminin gündüz vardiyasında montaj biriminin tüketim hızından daha yavaş üretim yapabilmesi ve çeşitli nedenlerle montaj hatlarının plan dışı üretim değişimleri nedeniyle yarı mamul üreten birim için talep stokastik hale gelmekte ve ihtiyaç ile örtüşen üretim planlaması yapmak zorlaşmaktadır.

Üretim planlaması sadece montaj hattında üretilen son ürünler için haftalık olarak yapılmaktadır. Hafta içerisinde toplam üretilecek miktar aynı kalmakla beraber üretim zamanlaması değişebilmektedir. Yarımamul üretimleri için bir planlama yapılmamakta olup stok düzeyleri takip edilememektedir. Yarı mamul üreten birim, basit bir kartlı çekme sistemi mantığı kullanarak azalan veya biten yarı mamulleri tüketimi fazla olanı önce üretecek şekilde yapılan sıralamayla üretime almaktadır. Üretim politikası toplamda aşırı stok seviyesi oluşturmakla beraber bazı yarımamullerde sıklıkla sıfır stok seviyesine düşerek montaj hattı duruşlarına neden olmaktadır.

3.2 Hedeflerin Tanımlanması

Üretim sisteminden veri almak amacıyla depolanan, tüketilmek üzere montaj hattına giden ve Operasyon_1'e alınan yarımamullerin barkodlu Kanban kartları okutularak kayıtlar bilgisayar ortamına aktarılmaktadır. Depolama alanından montaj hattına her yarı mamul gittiğinde yarı mamul üretim biriminde üretim sıralamasının yenilenmesi / güncellenmesi gerekliliği vardır. Her yarı mamulün ilgili haftadaki toplam tüketim miktarı bellidir. Alan ve kasa adedi kısıtlaması nedeniyle en küçük kap adedinde depolama yaparak montaj hatlarının yarı mamul nedeniyle durmasına neden olmayacak üretim planlamasının / sıralamasının yapılması amaç olacaktır.

3.3 Matematiksel Tanımlamalar

3.3.1. Denklemler

- **Kalan sipariş miktarı:**

Sipariş döneminin başından sonuna kadar yapılan üretimlerle beraber Kalan sipariş miktarı düşer. Bir j referansının α anında kalan sipariş miktarı aşağıdaki gibi hesaplanır.

$$D_j'(\alpha) = D_j - \sum_{n=0}^{\alpha-1} k_j(n) \quad (3.1)$$

- **WIP (Work-in-Process) süreç içi stok miktarının hesaplanması:**

Bir α anında j referansına ait WIP stoğu o anda üretimi devam eden miktar, depoda bekleyen miktar ve montaj hattındaki miktarın toplamı olarak ifade edilmektedir. Montaj hattındaki miktar en fazla bir kutu içi miktar kadardır.

$$WIP_j(\alpha) = k_j(\alpha) + I_j(\alpha) + Q_j'(\alpha) \quad (3.2)$$

- **WIP süreç içi stoğun tükenme süresi:**

Her bir j parça referansı için montaj hatlarının tüketim hızı deterministik olarak bellidir. Bu durumda bir α anında WIP stoğun tükenme süresi şu şekilde hesaplanır.

$$\tau_j^{WIP}(\alpha) = \frac{WIP_j(\alpha)}{V_j} \quad (3.3)$$

- **En geç üretime başlama zamanının hesaplanması:**

Bir j parçası üretilmek üzere ilk işlemin kuyruğuna girdikten τ_j geçiş süresi sonra üretilmiş olur.

$$\tau_j = \sum_{\forall i} (W_{ij} + S_{ij} + P_{ij}) \quad (3.4)$$

Stoksuz kalınmaması için geçiş süresi WIP stoğun tükenme süresinden az olmalıdır.

$$\tau_j \leq \tau_j^{WIP}(\alpha) \quad (3.5)$$

Bununla beraber j referansı için bir en küçük stok seviyesi belirlenmiş olabilir. Bu durumda akış süresi ile WIP stoğun tükenme süresi arasındaki fark en küçük stok seviyesinin altına düşülmeyecek kadar olmalıdır.

$$\tau_j^{Imin} = \frac{I_j^{min}}{V_j} \quad (3.6)$$

$$\tau_j \leq \tau_j^{WIP}(\alpha) - \tau_j^{Imin} \quad (3.7)$$

$\tau_j^{WIP}(\alpha) - \tau_j^{Imin}$ değeri parça üretiminin tamamlanarak stoğa alınması için kullanılacak sürenin üst sınırıdır. İlgili üretim süreci için en geç teslim süresi olarak kabul edilebilir (Due Date).

Üretimin başlayacağı zaman en geç teslim zamanını aşmayacak şekilde teslim zamanından en az akış süresi kadar önce olmalıdır. En geç üretime başlama zamanını hesaplamak için ilk operasyon kuyruğunda geçecek süre akış zamanının dışında tutulmalıdır. Bir t gerçek zamanında montaj hattına gönderilecek j. referans için en geç üretime başlama zamanı aşağıdaki gibi tanımlanır:

$$l_j(\alpha) = t + \tau_j^{WIP}(\alpha) - \tau_j^{Imin} - \tau_j - W_{1j} \quad (3.8)$$

İşlem, bekleme ve kurulum (setup) sürelerini daha açık ifade edecek şekilde düzenlediğimizde aşağıdaki şekli alır.

$$l_j(\alpha) = t + \tau_j^{WIP}(\alpha) - \tau_j^{Imin} - \left[\sum_{\forall i} (W_{2j} + S_{ij} + P_{ij}) \right] \quad (3.9)$$

3.3.2. Denklemlerin kartlı sistem ve kutu içi miktarlar ile düzenlenmesi

Kalan sipariş miktarı denkleminde bir değişiklik yapılmasına gerek yoktur.

- **WIP (Work-in-Process), süreç içi stok miktar hesabının düzenlenmesi:**

Önceki bölümde bir α anında j referansına ait WIP stoğu o anda üretimi devam miktar, depoda bekleyen miktar ve montaj hattındaki miktarın toplamı olarak ifade edilmişti. Kartlı çekme sistemlerinde herbir kart bir kutu malzemeyi temsil etmektedir. Stok miktarlarını kart adetleriyle ifade eden denklem aşağıdaki şekilde oluşturulur.

$$WIP_j^K(\alpha) = (K_j^S(\alpha) + K_j^P(\alpha)) Q_j + Q_j'(\alpha) \quad (3.10)$$

- **WIP, süreç içi stoğun tükenme süresi hesabının düzenlenmesi:**

$$\tau_j^{WIP}(\alpha) = \frac{WIP_j^K(\alpha)}{V_j} \quad (3.11)$$

şeklinde kartlı sistem için ifade edilir.

$$\tau_j^K = \frac{Q_j}{V_j} \quad (3.12)$$

bir kartlık stoğun montaj hattında tükenme süresidir.

$$\tau_j^{WIP}(\alpha) = (K_j^S(\alpha) + K_j^P(\alpha)) \tau_j^K + \frac{Q_j'(\alpha)}{V_j} \quad (3.13)$$

herhangi bir anda bir j referansına ait stoğun tükenme süresidir.

t_j^1 j referansının depodan son çıkma zamanı olmak üzere bir t anında hat içerisindeki

stoğun tükenme zamanı $\frac{Q_j'}{V_j}$ aşağıdaki şekilde bulunabilir.

$$\frac{Q_j'}{V_j} = \max\{0; (t_j^1 + \tau_j^K) - t\} \quad (3.14)$$

Bu durumda bir t anında bir j referansının tükenme süresi aşağıdaki şekilde hesaplanabilir.

$$\tau_j^{WIP}(\alpha) = (K_j^S(\alpha) + K_j^P(\alpha)) \tau_j^K + \max\{0; (t_j^1 + \tau_j^K) - t\} \quad (3.15)$$

- **En geç üretime başlama zamanı hesabının düzenlenmesi:**

$\tau_j^{WIP}(\alpha)$ değeri kart adetlerine göre hesaplarak önceki bölümde belirtilen (3.15) denklemi aynen kullanılır. Bir t gerçek zamanında oluşan α hareketiyle en geç başlama zamanı tespit edilirse;

$$l_j(\alpha) = t + \tau_j^{WIP}(\alpha) - \tau_j^{Imin} - \sum_{\forall i} (W_{2j} + S_{ij} + P_{ij}) \quad (3.16)$$

3.3.3. Kart adetlerinin hesaplanması

Çekme sisteminin üretim-tüketim dengesini koruyacak şekilde düzenlenmesi gerekmektedir. Bu noktada kart adetlerinin doğru belirlenmesi oldukça önemlidir. Döngüye fazla kart konulduğunda WIP stok miktarı yüksek olacak, az kart konulduğunda ise tüketim talebinin zamanında karşılanamaması durumu ortaya çıkacaktır.

- **Kanban (kart adedi) miktarı belirleme denklemi**

Çekme sistemi için Kart Adedini belirlemek üzere oluşturulmuş denklem aşağıdaki gibidir. Kanban miktarı kart adedini ifade etmektedir. Kart adedi (Kanban) tam sayı olmalıdır.

$$\text{Kanban} \geq \frac{\text{Talep hızı} \cdot \text{Akış Zamanı} \cdot (1 + \text{Güvenlik Katsayısı})}{\text{Kutu içi miktar}}$$

$$K_j = \lceil D_j \tau_j (1 + G_j) / Q_j \rceil \quad (3.17)$$

Problemden talep oluşturan müşteri montaj hatlarıdır. Talep olarak montaj hatlarının tüketim hızı kullanılacaktır.

$$D_j = V_j$$

Akış Zamanı (Lead / Flow Time) ise parça üreten birimin τ_j , üretim akış zamanıdır. Problemin değişkenleriyle denklem aşağıdaki şekilde oluşur.

$$K_j = \left\lceil \frac{V_j \sum_{i \in I} (W_{ij} + S_{ij} + P_{ij}) (1 + G_j)}{Q_j} \right\rceil \quad (3.18)$$

V_j montaj hattının çalıştığı anda birim zamanda (adet/dk) tükettiği parça miktarıdır.

Kutu veya kasa başına bir kanban kartı verilecektir. Siparişleme periyotları haftalıktır. Bu durumda ilgili referans için haftalık en büyük kasa talebiden daha fazla karta ihtiyaç yoktur. Haftalık en büyük kasa talebini D_j olarak ifade edersek kanban kart adedi aşağıdaki şekilde sınırlandırılabilir.

$$K_j = \left\lceil \frac{(\sum_{i \in I} (W_{ij} + P_{ij} + S_{ij}) (1 + G_j) \min(D_j, V_j))}{Q_j} \right\rceil \quad (3.19)$$

- **En küçük stok miktarının güvenlik oranı olarak kanban miktar hesabına yansıtılması:**

G , güvenlik oranı kullanmanın amacı WIP stoğu yükselerek kart döngüsünün kırılma riskini azaltmaktır. Oran olarak belirlenen güvenlik ölçütü süre olarak belirlenebilir. En küçük stok süresi (τ_j^{min}) bu amaçla kullanılmaktadır. Denklem yeniden düzenlendiğinde aşağıdaki denklem elde edilir.

$$K_j = \left\lceil \frac{(\sum_{i \in I} (W_{ij} + P_{ij} + S_{ij}) + \tau_j^{min}) \min(D_j, V_j)}{Q_j} \right\rceil \quad (3.20)$$

Yukarıdaki denklem ile belirlenen en küçük stok süreleri Kanban hesabına yansıtılmış olur. I_j^{min} değerlerinin belirlenmesi birçok değişik yöntem ile yapılabilir. I_j^{min} değeri üretim sırası önceliklendirme kriteri olarak kullanılabilir. Ne kadar yüksekse o kadar öncelikli olur.

3.3.4. Bekleme sürelerinin (W_{ij}) hesaplanması

Üretim aşamaları arasında, işin geliş sıklığı ile işlem süresi arasındaki farklılığa bağlı olarak bekleme kuyrukları ve dolayısıyla bekleme süreleri oluşacaktır. Bu süreler akış zamanını etkilediğinden çekme sistemi hesaplarını etkileyecektir.

Tez kapsamında kuyrukta bekleme süreleri beklenen en yüksek ortalama değerler olarak hesaplanarak kart sayısı hesabına dahil edilmiştir. Öte yandan bilgisayar ile yönlendirilen bir üretim sıralama sisteminde malzeme hareketini temsil eden kart hareketlerinin kaydedilmesiyle oluşacak olan istatistiksel veriler bekleme süresi hesabında kullanılabilir.

Üç noktada kart hareketinin kaydedilmesi ile tüm bekleme sürelerinin tespit edilmesi düşünülmüştür.

- t_j^1 : j. referansın depodan montaj hattına götürülme zamanı
 t_j^2 : j. referansın Operasyon_1'e başlama zamanı
 t_j^3 : j. referansın işlemlerinin bitip depoya alınma zamanı

Bir kanban kartının Operasyon_1 kuyruğuna girmesi için ilgili parçanın depodan çekilip montaj hattına götürülmesi gerekmektedir. Yani ilk kayıt t_j^1 olarak gerçekleşecektir. Ancak Operasyon_1'e başlamak için, t_j^1 anından sonra Operasyon_1 için bekleme süresi ve kurulum süresi kadar zaman geçmiş olacaktır.

$$t_j^2 = t_j^1 + W_{1j} + S_{1j} \quad (3.21)$$

Bekleme süresini elde etmek için denklem aşağıdaki gibi düzenlenir.

$$W_{1j} = t_j^2 - t_j^1 - S_{1j} \quad (3.22)$$

Operasyon_2'ye başlamak için t_j^2 üzerinden Operasyon_1'in işlem süresi, Operasyon_2 için bekleme süresi ve Operasyon_2 için kurulum süresi kadar zaman geçecektir. Operasyon_2'nin tamamlanma süresi eklendiğinde ikinci kayıt süresine t_j^3 ulaşılabilecektir.

$$t_j^3 = t_j^2 + P_{1j} + W_{2j} + S_{2j} + P_{2j} \quad (3.23)$$

W_{1j} ile benzer şekilde W_{2j} için ifade aşağıdaki gibi düzenlenir.

$$W_{2j} = t_j^3 - t_j^2 - P_{1j} - S_{2j} - P_{2j} \quad (3.24)$$

t_j^m değerleri alınan kayıtlarla (örn: barkod tabancası ile okutma) oluşmaktadır. P_{ij} , S_{ij} değerleri deterministik olarak belirlenmiştir. Süreç içinde oluşan W_{ij} değerleri hesaplanarak belirli bir dönem için istatistiksel kayıtlar elde edilebilir. Bu dönemin uzunluğu bir saat gibi kısa bir süre olabileceği gibi bir ay gibi orta uzunlukta bir süre de olabilir. Elde edilen sonuçlar bir sonraki dönemin Çekme Sistemi hesaplarında veri olarak kullanılabilir gibi iyileştirme çalışmalarının belirlenmesi ve izlenmesinde de veri olarak kullanılabilir.

3.4. Önceliklendirme Yaklaşımı

Önceliklendirme yaklaşımımız SPT'nin kuyruk uzunluğunu azaltan özelliğinin en geç başlama zamanı ile birleştirilerek geç kalmaya neden olmayacak şekilde SPT'nin kullanılmasıdır. Algoritma, eğer varsa, öncelikle geç kalması muhtemel işleri SPT'ye göre sıralayarak önceliklendirir. Eğer geç kalan iş yoksa tüm işleri SPT'ye göre önceliklendirir. Eğer üretilmek üzere kartı gelen yarı mamulün ilgili sipariş dönemi için kalan siparişi yoksa, stok değeri yeterliyse, üretim planında sipariş döneminin sonuna atılır. Yine yeterli stoğu olan yarı mamuller SPT'ye göre kendi içlerinde sıralanır.

3.4.1. Sıralama Yaklaşımları

Aynı temel önceliklendirme prensibi ile hareket ederek, birbirinden sipariş ve/veya sıfır stok kontrolü içerip içermemesi özellikleriyle ayrılan sıralama algoritmaları üretilebilir.

1. Sipariş kontrolü tabanlı
2. Sipariş ve sıfır stok kontrolü tabanlı
3. Sıfır stok kontrolü tabanlı

3.4.2. SPT öncelik değerleri tablosu

Sıralama algoritmasında kullanılmak üzere Operasyon_1 SPT Öncelik Değerleri Tablosu hazırlanır. En büyük işlem süresine öncelik değeri olarak 1 değerini verdikten sonra büyük işlem süreliden küçük işlem süreliye doğru öncelik değerlerini 1 artırıp her işlem süresi için SPT Öncelik Değeri elde edilir ve $WSPT_j^1$ değerleri listelenir.

3.4.3. Sınır katmanlarının belirlenmesi

Sıralama algoritmasında 4 katman olacaktır. İlk katman en az öncelikli katmandır. Son katman ise en öncelikli katman olacaktır. OS: Öncelik Sınırı olmak üzere;

- Kalan siparişi olmayanlar (OS_1) (Sipariş kontrollü)
- En geç başlama süresine erişmemiş olanlar (OS_2)
- En geç başlama süresi geçmiş olanlar (OS_3)
- Sıfır stok durumunda olanlar (OS_4) (Sıfır stok kontrollü)

Bu katmanları ayıracak sınır başlangıçları belirlenmelidir. Belirlenen sınır başlangıçları bu 4 katmanın çakışmasını engellemelidir. Örneğin SPT Öncelik Değerlerinin en büyük değeri 100'den küçük ise sınır değerleri 100'ün katları şeklinde seçilebilir.

Örnek: $OS_1 = 100$, $OS_2 = 200$, $OS_3 = 300$, $OS_4 = 400$

3.4.4. Sıralama Algoritması

Süreç üretim ve tüketim döngüsü içerisinde üç tip olay üzerinde gerçekleşir. Üretim sıralaması bu üç tip olayın kendi alt döngüleri esnasında oluşur.

Olay Listesi:

Olay 1 : Talep edildiğinde j parçasını tüketilmek üzere montaj hattına götür.
Prosedür 1’i uygula.

Olay 2 : 1.operasyon makinelerinden biri boşaldığında yeni parçayı Operasyon_1’e al.
Prosedür 2’yi uygula.

Olay 3 : Üretimi tamamlanan parçayı depoya kaldır. **Prosedür 3’ü uygula.**

Prosedür 1:

Adım 1. Talep edildiğinde j parçası tüketilmek üzere son operasyona (montaj hattı) götürülür.

t_j^1 zaman kaydı oluşur ve α hareket sırası bir arttırılır.

Adım 2. Giden parça “STOKTAKİLER” listesinden çıkarılır, stok değeri güncellenir.

$$K_j^S(\alpha) = K_j^S(\alpha - 1) - 1 \quad (3.25)$$

Adım 3. j referansı, t_j^1 değeriyle beraber “OPERASYON_1 İÇİN BEKLEYENLER” listesine eklenir.

Prosedür 2:

Adım 1. Operasyon_1 makinelerinden biri boşaldığında “OPERASYON_1 İÇİN BEKLEYENLER” listesine bakılır. Eğer liste boş ise yeni kart gelene kadar beklenir.

Adım 2. Operasyon_1 başlama zamanı t_j^2 kaydı alınır.

Adım 3. OPERASYON_1 İÇİN BEKLEYENLER” listesindeki her bir referans için Öncelik Değeri Hesabı uygulanır.

Adım 4. Öncelik Değeri Hesabı sonuçlarına göre referanslar OD_j^α değerleri büyükten küçüğe, t_j^1 değerleri küçükten büyüğe olacak şekilde sıralanır ve ilk sıradaki referans Operasyon_1'e alınır.

Adım 5. İlgili referans “OPERASYON_1 İÇİN BEKLEYENLER” listesinden çıkarılarak “İŞLEM GÖRENLER” listesine eklenir, stok değeri güncellenir.

$$K_j^P(\alpha) = K_j^P(\alpha - 1) + 1 \quad (3.26)$$

- **Öncelik Değeri Hesabı:**

Adım 1. Tükenme süresi hesaplanır.

$$\tau_j^{WIP}(\alpha) = (K_j^S(\alpha) + K_j^P(\alpha)) \tau_j^K + \max\{0; (t_j^{11} + \tau_j^K) - t_j^2\} \quad (3.27)$$

Adım 2. En geç başlama zamanı hesaplanır.

$$l_j(\alpha) = t_j^2 + \tau_j^{WIP}(\alpha) - \tau_j^{Imin} - \sum_{\forall i} (W_{2j} + S_{ij} + P_{ij}) \quad (3.28)$$

Adım 3. Öncelik değeri hesaplanır.

$$OP_j^\alpha = \begin{cases} OS_1 + WSPT_j^1, & D_j'(\alpha) \leq 0, \text{ eğer sipariş kontrolü varsa} \\ OS_2 + WSPT_j^1, & l_j(\alpha) > t_j^2 \\ OS_3 + WSPT_j^1, & l_j(\alpha) \leq t_j^2 \\ OS_4 + WSPT_j^1, & K_j^S(\alpha) = 0, \text{ eğer sıfır stok kontrolü varsa} \end{cases} \quad (3.29)$$

Birbirine alternatif olarak sunulan üç algoritma tanımlanmıştır. Bu algortimalar sadece Öncelik Değeri Hesabındaki farklılık ile birbirinden ayrılırlar:

Algoritma 1’de Sipariş Kontrollü Sıralama için Öncelik Değer Hesabı:

$$OP_j^\alpha = \begin{cases} OS_1 + WSPT_j^1, & D_j'(\alpha) \leq 0 \\ OS_2 + WSPT_j^1, & l_j(\alpha) > t_j^2 \\ OS_3 + WSPT_j^1, & l_j(\alpha) \leq t_j^2 \end{cases} \quad (3.30)$$

Algoritma 2’de Sipariş ve Sıfır Stok Kontrollü Sıralama için Öncelik Değer Hesabı:

$$OP_j^\alpha = \begin{cases} OS_1 + WSPT_j^1, & D_j'(\alpha) \leq 0 \\ OS_2 + WSPT_j^1, & l_j(\alpha) > t_j^2 \\ OS_3 + WSPT_j^1, & l_j(\alpha) \leq t_j^2 \\ OS_4 + WSPT_j^1, & K_j^S(\alpha) = 0 \end{cases} \quad (3.31)$$

Algoritma 3’de Sıfır Stok Kontrollü Sıralama için Öncelik Değer Hesabı:

$$OP_j^\alpha = \begin{cases} OS_2 + WSPT_j^1, & l_j(\alpha) > t_j^2 \\ OS_3 + WSPT_j^1, & l_j(\alpha) \leq t_j^2 \\ OS_4 + WSPT_j^1, & K_j^S(\alpha) = 0 \end{cases} \quad (3.32)$$

Prosedür 3:

Adım 1. Operasyon_2 tamamlanan parça depoya alınır. t_j^3 zaman kaydı yapılır ve α hareket sırası bir arttırılır.

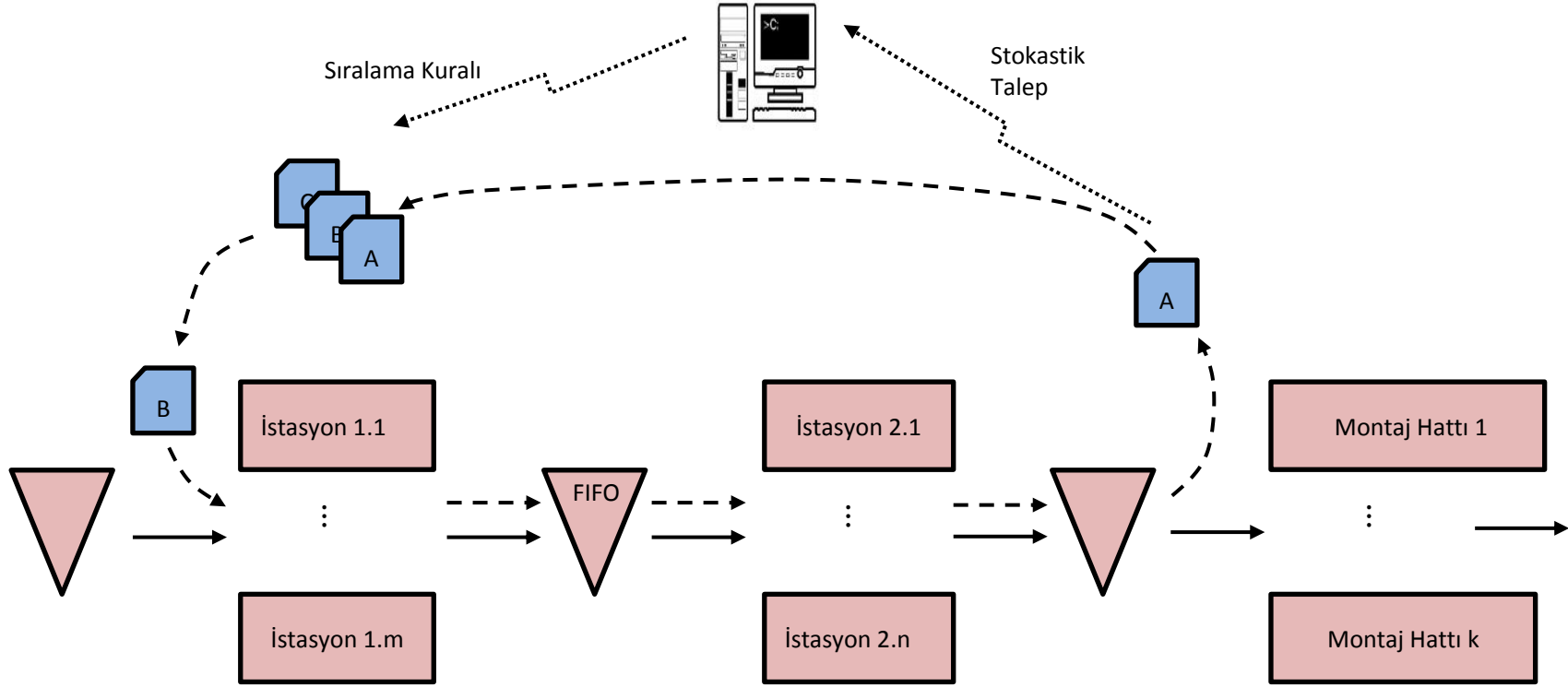
Adım 2. Depoya alınan parça “İŞLEM GÖRENLER” listesinden çıkarılarak “STOKTAKİLER” listesine eklenir, stok değeri güncellenir.

$$K_j^S(\alpha) = K_j^S(\alpha - 1) + 1; \quad K_j^P(\alpha) = K_j^P(\alpha - 1) - 1 \quad (3.33)$$

Adım 3. Adım 1 ve 2 üretim işlemi bitip depoya alınan her parça için uygulanır.

Algoritmaların son operasyonda (montaj hattı) neden olduğu duruş süreleri karşılaştırılacaktır. Karşılaştırmalar benzetim modellerinin bilgisayar ortamında çalıştırılması ile yapılmıştır.

Algoritmanın test edileceği üretim ortamı Tasarlanan Üretim Ortamı şeklinde ifade edilmiştir (Şekil 3.2.).

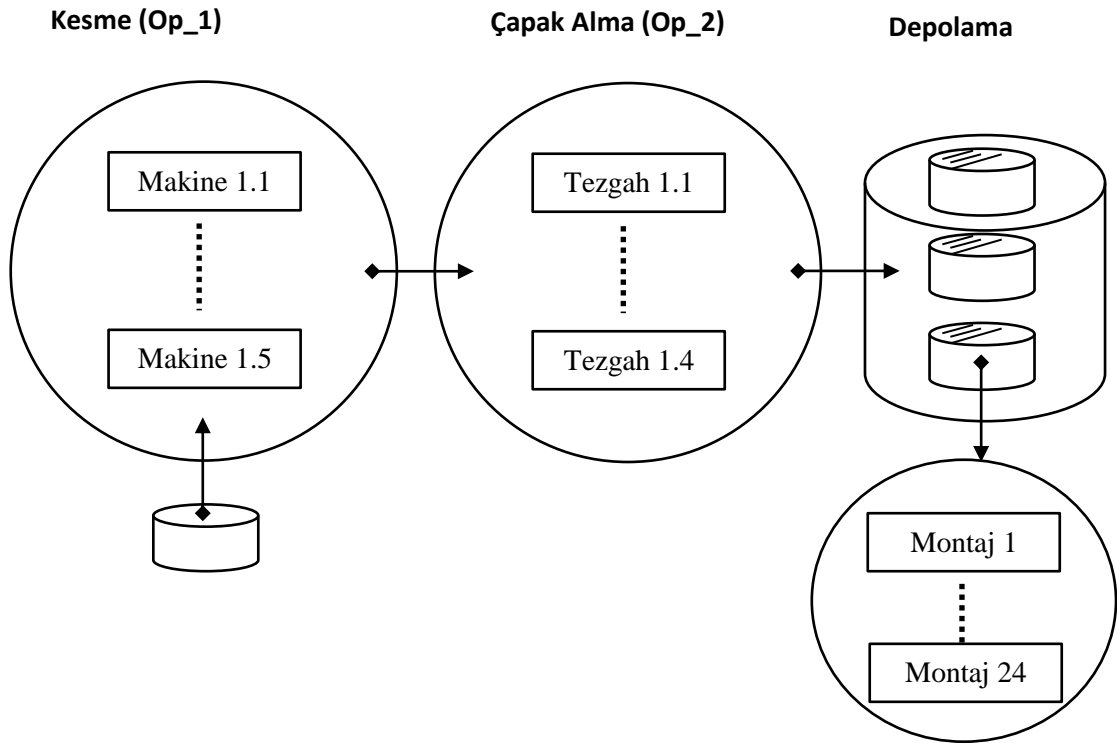


Şekil 3.2. Tasarlanan Üretim Ortamı

3.5. Uygulama

3.5.1. Üretim Ortamına Ait Veriler

Tez kapsamında gerçek bir üretim sisteminin mevcut durum analizinden elde edilen veriler tanımlanmıştır. Belirtilen gerçek üretim sisteminde, seri üretim yapan montaj hatlarında kullanılmak üzere yarı mamul üreten birimde, 5 adet özdeş makineden birinde kesme operasyonundan (Operasyon_1) geçen malzeme 4 adet özdeş tezgahın birinde çapak alınarak (Operasyon_2) yarı mamul halini almakta ve aynı yerde bulunan depolama bölgesine konulmaktadır. Montaj hatlarının talebine göre depolama bölgesinden alınan yarı mamul montaj hattına götürülmektedir (Şekil 3.3.).



Şekil 3.3. Uygulama Konusu Üretim Ortamı

Nihai müşterinin ürün siparişleri geliş frekansı ve miktar olarak çeşitlilik göstermektedir. Sürekli üretilen ürünler olduğu gibi siparişleri kesikli biçimde haftalık veya aylık olarak gelen ürünlerde bulunmaktadır. Firmanın montaj hatları gündüz vardiyasında tam kapasite üretim yaparken akşam ve gece vardiyalarında müşteri siparişine göre gerekli olan montaj hatlarında üretim yapılmaktadır. Gündüz vardiyasında montaj hatlarının yarı

mamul tüketim hızı, yarı mamul üretim hızından yüksek iken akşam ve gece vardiyalarında düşüktür. Toplamda yarı mamul üretim biriminin kapasitesi ortalama tüketimi karşılar düzeydedir. Değişken müşteri siparişleri, vardiyalar arasında dengeli olmayan kapasite kullanımı, yarı mamul üretim biriminin gündüz vardiyasında montaj biriminin tüketim hızından daha yavaş üretim yapabilmesi ve çeşitli nedenlerle montaj hatlarının plan dışı üretim değişimleri nedeniyle yarı mamul üreten birim için ihtiyaç ile örtüşen üretim planlaması yapmak zorlaşmaktadır.

Üretim planlaması sadece montaj hattında üretilen son ürünler için haftalık olarak yapılmaktadır. Hafta içerisinde toplam üretilecek miktar aynı kalmakla beraber üretim zamanlaması değişebilmektedir. Yarı mamul üretimleri için bir planlama yapılmamakta olup, stok düzeyleri ERP sisteminde takip edilememektedir. Yarı mamul üreten birim, basit bir çekme sistemi mantığı kullanarak azalan veya biten yarı mamulleri fazla tüketilenleri önce üretecek şekilde sırayla üretime almaktadır.

Üretim politikası toplamda aşırı stok seviyesi oluşturmakla beraber bazı parçalarda sıklıkla sıfır stok seviyesine düşerek montaj hattı duruşlarına neden olmaktadır. İlgili birim malzeme yokluğu duruşların %21'ine neden olmaktadır ve bu veri ile montaj hattının en kötü tedarikçisi konumundadır.

Gerçek üretim ortamından alınan veriler Çizelge 3.1. ve 3.2.'de özetlenmiştir. Tüm veriler tek bir yarı mamulden belli bir miktarda içeren kasa başıdır.

Bir kasanın montaj hattında tüketim süresi $1/V_j$ 'dir.

Çizelge 3.1. Setup süreleri (Dakika)

Setup 1	Setup 2	Setup 3
7	5	3

Setup 1 : Operasyon_1 hazırlık süresi.

Setup 2 : Operasyon_1 sonrası kasa düzenleme ve aktarma süresi.

Setup 3: Operasyon_2 hazırlık süresi.

Çizelge 3.2. Operasyon ve Tüketim Süreleri (Dakika)

j	P_{1j}	P_{2j}	I/V_j	j	P_{1j}	P_{2j}	I/V_j
1	34	0	200	17	137	80	267
2	34	0	200	18	103	60	300
3	100	0	357	19	100	75	167
4	80	0	286	20	100	75	167
5	45	35	109	21	100	67	185
6	45	35	109	22	100	67	170
7	126	80	314	23	45	35	129
8	126	80	270	24	45	35	129
9	113	137	293	25	45	35	122
10	75	58	190	26	45	35	122
11	75	58	190	27	126	86	300
12	86	86	500	28	126	86	300
13	86	86	500	29	72	30	144
14	111	67	169	30	60	30	144
15	111	67	169	31	160	80	348
16	137	80	267	32	160	80	348

j	P_{1j}	P_{2j}	I/V_j	j	P_{1j}	P_{2j}	I/V_j
33	137	80	185	49	160	80	185
34	160	80	185	50	90	45	110
35	71	86	250	51	90	45	110
36	100	75	204	52	45	36	165
37	100	75	204	53	45	36	165
38	160	80	429	54	86	67	275
39	160	80	384	55	86	67	275
40	77	45	169	56	90	45	104
41	77	45	169	57	77	45	104
42	45	36	119	58	60	45	110
43	45	36	119	59	60	45	137
44	15	12	52	60	60	45	137
45	15	12	52	61	60	45	110
46	137	80	185	62	137	80	267
47	137	80	185	63	90	70	180
48	137	80	185	64	90	70	180

3.5.2. Montaj Hattı Talep Verilerinin Analizi ve Kanban Kart Sayılarının Hesaplanması

Nihai müşterilerden gelen siparişler ve öngörüler neticesinde yarımamullerin hafta bazında talep öngörülerini Çizelge 3.3.'de verilmiştir.

Çizelge 3.3. Haftalık Yarı Mamul Sipariş Kasa Adetleri

Haftalar									
j	1	2	3	4	5	6	7	8	Toplam
1			2	4		3	2	3	14
2			2	4		3	2	3	14
3	3	3	2	3	3	4	2	3	23
4			2				1		3
5	13	9	2	8	7	7	8	8	62
6	17	9	7	1	4	7	11	11	67
7	7	7	8	6	8	7	10	6	59
8		1	1	1		1	1	1	6
9	5	5	5	2	4	2	5	4	32
10	6	7	4	3	8	8	9		45
11	6	7	4	3	8	8	9		45
12	3		4		2	1	1	2	13
13	4	4	1	5	4	7	4	5	34
14	6	4	6	3	6	3	10	5	43
15	3	6	6	5	8	5	7	7	47
16	9	5	3	1	7	7	9	8	49
17					1				1
18	3		2	3	3	1	3	2	17
19	4	5	5	4	3	3	13	9	46
20	18	12	15	13	22	24	19	18	141
21	6	7	5	5	7	4	4	8	46
22	5	8	6	5	5	7	2	4	42
23	1	5	4		2			1	13
24	1	5	4		2			1	13
25		12	16	16	18	23	19	20	124
26		12	16	16	18	23	19	20	124
27	5	4	4	4	2	2	7	2	30
28	5	4	4	4	2	2	7	2	30
29		4	1		2			2	9
30		4	2		1		2	1	10
31	4	2	4	3	3	3	4	2	25
32	4	3	4	5	2	3	3	4	28

Haftalar									
j	1	2	3	4	5	6	7	8	Toplam
33	5	9	10	4	8	6	8	5	55
34	5	9	8	5	7	4	9	5	52
35			1	1	1	1			4
36	4	7	5	6	6	9	3		40
37	4	3	4	4	8	3	4	5	35
38	1		1	2	2		3	1	10
39		2			2	2		2	8
40		1							1
41		1							1
42	50	44	44	43	28	30	24	46	309
43	50	44	44	43	28	30	24	46	309
44							3		3
45							3		3
46	3	2	2	3	1	3	2	5	21
47	2	2	2	2	2	2	2	5	19
48	8		4	8	4	7	2	4	37
49	5	4	5	7	4	3	3	2	33
50	11	3	8	8	11	10	12	12	75
51	7	8	6	14	7	10	12	16	80
52	1	4	2	1	3	4	4	3	22
53	1	4	2	1	3	4	4	3	22
54	4	5	5	1	2	2	3	4	26
55	4	5	5	1	2	2	3	4	26
56			1			1			2
57		1				1			2
58	3	4	6	3	6	3	7		32
59	3	4	7	5	2	3	4	4	32
60	4	3	7	4	3		6	7	34
61	4	3	5	5	3	5	3	3	31
62	3	4	4	1	3	3	6	5	29
63	27	18	15	8	11	8	13	11	111
64	27	18	15	8	11	8	13	11	111

Üretim ve kapasite kullanım planlaması yapılırken vardiyalar arası denge gözetilmemektedir. Üretim dışı sabit maliyetleri en küçükleme, enerji tasarrufu sağlamak ve üretim kontrolünün yükselmesi amacıyla beyaz yaka personelin mesaide olduğu dönemde mümkün olan en fazla üretim gerçekleştirilmek istenmektedir. Bu doğrultuda gündüz vardiyasında (1) olabilecek en yüksek üretim miktarı planlanmaktadır. Kalan üretimler sırasıyla akşam (2) ve gece (3) vardiyalarında yapılmaktadır.

8 haftalık üretim planına genel üretim planı verileri Çizelge 3.4.'de belirtilmiştir.

Çizelge 3.4. Genel üretim planı verileri

Tanım	Kısaltma	Miktar	Birim
Toplam çalışma günü	TG	49	Gün
Toplam çalışma vardiyası	TV	147	Vrd
Vardiya çalışma süresi	VS	430	Dakik
Toplam sipariş kasa miktarı	TKM	2830	Kasa (Kart)
Vardiya 1 üretim oranı	UO ₁	%55	
Vardiya 2 üretim oranı	UO ₂	%30	
Vardiya 3 üretim oranı	UO ₃	%15	

Operasyon ve tüketim süreleri tablosunda belirtildiği üzere talep edilen herbir yarı mamul kasınının montaj hattında kaç dakika tüketileceği bellidir. Bu değer montaj hatlarının üretim hızıyla oluşan değerdir. 8 haftalık tüketim öngörü belli olduğuna göre montaj hatlarının toplam üretim süresi aşağıdaki gibi hesaplanabilir.

D_{jh} : j referansının h haftası talep miktarı (kasa adedi)

V_j : j referansının montaj hattı tüketim hızı (kasa adet/dakika)

TMS : İlgili dönemde toplam montaj süresi olmak üzere

$$TKM = \sum_j \sum_h D_{jh} \quad (3.34)$$

$$TMS = \sum_j \sum_h D_{jh} / V_j \quad (3.35)$$

formülü ile TKM, 2830 adet-kasa ve TMS, 492881 dakika olarak hesaplanır.

Operasyon_1’de beş, Operasyon_2’de dört makine kullanımdadır.

MBS : İlgili dönemde makine başına çalışılabilir süre,

TOS_i : İlgili dönemde i operasyonu toplam operasyon (işlem + setup) süresi,

KSO_i : İlgili dönemde i operasyonu kullanılabilir süre,

OOS_i : İlgili dönemde i operasyonu ortalama operasyon (işlem + setup) süresi,

AS_i : İlgili dönemde i operasyonu atıl süresi olmak üzere;

$$MBS = TV * VS \quad (3.36)$$

$$KSO_i = m_i * MBS \quad (3.37)$$

Operasyon_1, toplam işlem ve setup süresi:

$$TOS_1 = \sum_{\forall h} \sum_{\forall j} D_{jh} * (P_{1j} + S_{1j} + S_{2j}) \quad (3.38)$$

Operasyon_2, toplam işlem ve setup süresi:

$$TOS_2 = \sum_{\forall h} \sum_{\forall j} D_{jh} * (P_{2j} + S_{3j}) \quad (3.39)$$

$$AS_i = KSO_i - TOS_i \quad (3.40)$$

$$OOS_i = TOS_i / TKM \quad (3.41)$$

Yukarıdaki formüller ile Çizelge 3.5’te gösterilen sonuçlar elde edilir. TMS 492881 dakikadır.

Çizelge 3.5. Operasyon genel verileri

Kısaltma	Operasyon_1	Operasyon_2
m_i (adet)	5	4
MBS (dk)	63 210	63 210
KSO _i (dk)	316 050	252 840
TSO _i (dk)	264 395	165 287
AS _i (dk)	51 655	150 763
OOS _i (dk / kasa)	93,43	58,41

Montaj tarafından tüketilmek üzere çekilen yarı mamullerin kasalarından alınan kanban kartları üretim emri olarak Operasyon_1 ve 2'yi geçerek yeniden dolu yarı mamul kasaları haline gelir.

OK_{iv} : V vardiyası, i. operasyonda makine başına ortalama kart adedi,

OTS_{iv} : V vardiyası ortalama toplam i. operasyon süresi,

W_{iv} : V vardiyası sonunda i. operasyon ortalama kuyruk süresi,

KW_{iv} : V vardiya sonunda i. operasyon kümülatif kuyruk süresi

W_D : İlgili dönemin en büyük ortalama kuyruk uzunluğu olmak üzere;

$$OK_{iv} = (UO_v * \sum_j \sum_h D_{jh}) / (m_i * TV) \quad (3.42)$$

$$OTS_{iv} = OOS_i * OK_{iv} \quad (3.43)$$

$$W_{iv} = \max\{0, OTS_{iv} - VS\} \quad (3.44)$$

$$KW_{iv} = \max\{0, OTS_{iv} - VS + KW_{i,v-1}\} \quad (3.45)$$

Tüketim hızının üretim hızından yüksek olduğu durumlarda kanban kartlarıyla gelen tüm iş emirleri tamamlanarak ürün haline getirilemez ve vardiya sonunda bir kuyruk oluşur.

W_{iv} , bu kuyruğun ortalama ne kadar sürede eriyeceğini gösterir. Verilen formüllerle Çizelge 3.6 ve 3.7’de gösterilen sonuçlar elde edilir.

Çizelge 3.6. Operasyon 1, makine başına iş yükü analizi ve kuyruk uzunluğu tahmini

Vardiya	OK _{1v} (Adet)	OTS _{1v} (dk)	W _{1v} (dk)	KW _{1v} (dk)
1.	6,36	593,7	163,7	163,7
2.	3,47	323,7	0,0	57,4
3.	1,73	162,1	0,0	0,0

Çizelge 3.7. Operasyon 2, makine başına iş yükü analizi ve kuyruk uzunluğu tahmini

Vardiya	OK _{2v} (Adet)	OTS _{2v} (dk)	W _{2v} (dk)	KW _{2v} (dk)
1.	7,94	464,0	34,0	34,0
2.	4,33	253,0	0,0	0,0
3.	2,17	126,6	0,0	0,0

İlgili dönemin en büyük ortalama kuyruk uzunluğu süresi, W_D , en büyük kümülatif kuyruk uzunluğu süresidir. (3.46) formülü ile $W_D = 163,7 + 34 = 197,7$ olarak hesaplanır.

$$W_D = \max\{KW_{1v} + KW_{2v}\} \quad (3.46)$$

Denklem (3.47) ile 8 haftalık üretim dönemi için herbir yarı mamül referansının kanban kart adetlerini hesaplamak üzere ilgili üretim dönemindeki herbir referansın haftalık en yüksek talebi esas alınır.

$$D_j = \max\{D_{jh}\} \quad (3.47)$$

Çizelge 3.2’de operasyon ve montaj hattı tüketim süre değerleri, Çizelge 3.3’te talep değerleri verilen tüm yarı mamuller için kanban kart adetleri denklem (3.19) ile hesaplanır. Tüm yarı mamuller için bekleme süresi toplamı $\sum_{vi}(W_{ij}) = W_D$ alınmıştır. Kart adedini sınırlayan haftalık talep değeri olarak herbir yarı mamülün en büyük haftalık talebi, $D_j = \max\{D_{jh}\}$, kullanılmıştır. Elde edilen kart adedi değerleri Çizelge 3.8’de gösterilmiştir.

Çizelge 3.8. Kanban Kart Adetleri

j	D_j	$\frac{1}{\bar{V}_j}$	Süreler (dk)						$G=0$	$G=0,3$
			P_{1j}	P_{2j}	S_{1j}	S_{2j}	S_{3j}	τ_j	K_j	K_j
1	4	200	34	0	7	5	0	244	2	2
2	4	200	34	0	7	5	0	244	2	2
3	4	357	100	0	7	5	0	310	1	2
4	2	286	80	0	7	5	0	290	2	2
5	13	109	45	35	7	5	3	293	3	4
6	17	109	45	35	7	5	3	293	3	4
7	10	314	126	80	7	5	3	419	2	2
8	1	270	126	80	7	5	3	419	1	1
9	5	293	113	137	7	5	3	463	2	3
10	9	190	75	58	7	5	3	346	2	3
11	9	190	75	58	7	5	3	346	2	3
12	4	500	86	86	7	5	3	385	1	2
13	7	500	86	86	7	5	3	385	1	2
14	10	169	111	67	7	5	3	391	3	4
15	8	169	111	67	7	5	3	391	2	4
16	9	267	137	80	7	5	3	430	2	3
17	1	267	137	80	7	5	3	430	1	1
18	3	300	103	60	7	5	3	376	2	2
19	13	167	100	75	7	5	3	388	3	4
20	24	167	100	75	7	5	3	388	3	4
21	8	185	100	67	7	5	3	380	3	3
22	8	170	100	67	7	5	3	380	3	3
23	5	129	45	35	7	5	3	293	3	3
24	5	129	45	35	7	5	3	293	3	3
25	23	122	45	35	7	5	3	293	3	4
26	23	122	45	35	7	5	3	293	3	4
27	7	300	126	86	7	5	3	425	2	2
28	7	300	126	86	7	5	3	425	2	2
29	4	144	72	30	7	5	3	315	3	3
30	4	144	60	30	7	5	3	303	3	3
31	4	348	160	80	7	5	3	453	2	2
32	5	348	160	80	7	5	3	453	2	2

Çizelge 3.8. Kanban Kart Adetleri (Devam)

j	D_j	$\frac{1}{\bar{V}_j}$	Süreler (dk)						$G=0$	$G=0,3$
			P_{1j}	P_{2j}	S_{1j}	S_{2j}	S_{3j}	τ_j	K_j	K_j
33	10	185	137	80	7	5	3	430	3	4
34	9	185	160	80	7	5	3	453	3	4
35	1	250	71	86	7	5	3	370	1	1
36	9	204	100	75	7	5	3	388	2	3
37	8	204	100	75	7	5	3	388	2	3
38	3	429	160	80	7	5	3	453	2	2
39	2	384	160	80	7	5	3	453	2	2
40	1	169	77	45	7	5	3	335	1	1
41	1	169	77	45	7	5	3	335	1	1
42	50	119	45	36	7	5	3	294	3	4
43	50	119	45	36	7	5	3	294	3	4
44	3	52	15	12	7	5	3	240	3	3
45	3	52	15	12	7	5	3	240	3	3
46	5	185	137	80	7	5	3	430	3	4
47	5	185	137	80	7	5	3	430	3	4
48	8	185	137	80	7	5	3	430	3	4
49	7	185	160	80	7	5	3	453	3	4
50	12	185	90	45	7	5	3	348	4	5
51	16	185	90	45	7	5	3	348	4	5
52	4	110	45	36	7	5	3	294	2	3
53	4	110	45	36	7	5	3	294	2	3
54	5	165	86	67	7	5	3	366	2	2
55	5	165	86	67	7	5	3	366	2	2
56	1	275	90	45	7	5	3	348	1	1
57	1	275	77	45	7	5	3	335	1	1
58	7	104	60	45	7	5	3	318	3	4
59	7	104	60	45	7	5	3	318	3	4
60	7	110	60	45	7	5	3	318	3	4
61	5	137	60	45	7	5	3	318	3	4
62	6	137	137	80	7	5	3	430	2	3
63	27	110	90	70	7	5	3	373	3	3
64	27	267	90	70	7	5	3	373	3	3

3.5.3. Algoritmanın Test Edilmesi

Oluşturulan algoritmalar benzetim yöntemi ile test edilmiştir. Benzetim programı olarak Promodel programı kullanılmıştır.

Gerçek üretim ortamından alınan veriler kullanılarak benzetim verileri elde edilmiş ve türetilmiştir. Makine adetleri, kapasiteleri, yarımamul üretim ve tüketim hızları değiştirilmediği için testlerde kullanılacak olan benzetim verilerinin teze veri sağlayan üretim ortamıyla kapasite kullanımını açısından uyumlu olmasına özen gösterilmiştir. Eğer temel şartlar aynı kalırken kapasitenin altında bir üretim adedi üzerinden benzetim yapılırsa sıralama yönteminden bağımsız olarak tüm işler büyük oranda yetişecek, kuyruk oluşumu azalacak, bu durumda seçilen sıralama yöntemlerinin etkisi anlaşılamayacaktır. Aynı şekilde eğer kapasitenin çok üzerinde iş yükü ile sistem benzetime tabi tutulursa seçilen sıralama yöntemi her ne olursa olsun iş kuyrukları aşırı uzayacak ve tüm sıralama yöntemleri yüksek oranda beklemelelere neden olacaktır. Sıralama yöntemi kapasiteyi artırıp azaltmayacak, sadece kapasitenin doğru şekilde kullanılmasına olanak sağlayacaktır.

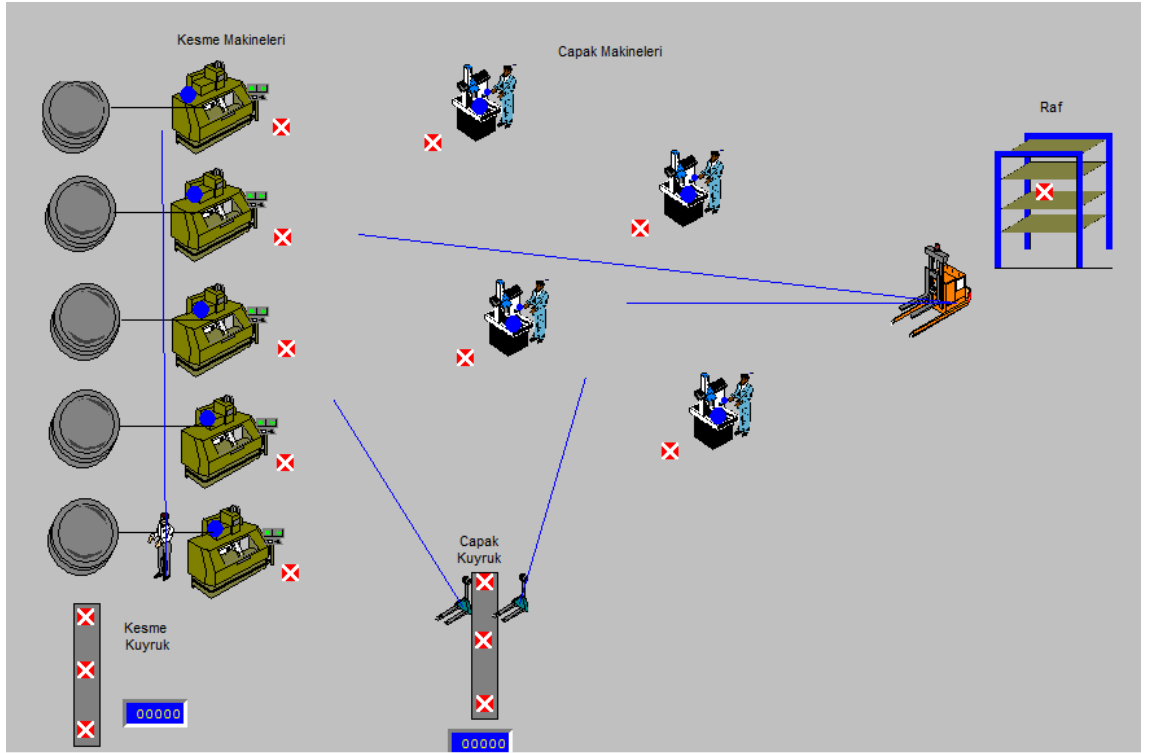
Benzetimde kullanılacak veriler aşağıdaki şekilde oluşturulmuştur.

1. Gerçek üretim ortamından alınan veriler Test 3 benzetimi için aynen kullanılmıştır.
2. Diğer benzetim verileri için yarı mamul tüketim sıraları rassal yöntemlerle oluşturulmuştur.
3. Benzetim verisi, kutu içi miktar bilgileri, üretim ve tüketim hızları kullanılarak her bir yarımamul için kanban kart adedi hesaplanmıştır.

Algoritmalar, benzetim ortamında İlk Gelen İlk İşlenir (FCFS), En Kısa Toplam İşlem Süresi (STPT), Teslim Zamanı (DD) ve En Kısa Darboğaz İşlem Süresi (SBPT) sıralama yaklaşımlarıyla kıyaslanarak test edilmektedir. Tüm sıralama yöntemleri benzetim ortamında hazırlanmış aynı üretim ve talep şartlarında, herbir yarımamul için belirlenmiş aynı sayıda kanban kartı kullanılarak test edilmektedir.

3.6. Benzetim Modelleri

Benzetim modeli olarak Promodel kullanılmıştır. İlk Gelen İlk İşlenir (FCFS), En Kısa Toplam İşlem Süresi (STPT), Teslim Zamanı (DD) ve En Kısa Darboğaz İşlem Süresi (SBPT) sıralama yaklaşımlarına yönelik benzetim modelleri üretilmiştir. Örnek problem gerçek üretim ortamından elde edildiğinden benzetim modellerinde gerçek operasyon isimleri kullanılmıştır. Kesme operasyonu Operasyon_1, çapak alma operasyonu Operasyon_2'dir. Depolama işlemi raflarda yapılmaktadır (Şekil 3.4).



Şekil 3.4. Benzetim Modelinin Promodel Ekran Görüntüsü

3.6.1. İlk Gelen İlk İşlenir (FCFS) Benzetim Modeli

Promodel'de modelin kurulduğu bölüm olan "Build" kısmında bulunan sıralama ile modelin bileşenlerini tanıtlacaktır.

- **Makine tanımlamaları (Locations)**

Kesme (1) operasyonu 5 özdeş makineden birinde, çapak alma (2) operasyonu 4 özdeş makineden birinde yapılabilmektedir. Her iki operasyon öncesi bekleme kuyrukları oluşabilir. Üretimi tamamlanan yarı mamul referansı depolanmak üzere rafa kaldırılmaktadır. Yarı mamul konduit adı verilen bir ilk maddeden üretilmektedir. Konduitler konduit balyaları şeklinde bulunmaktadır (Şekil 3.5).

Icon	Name	Cap.	Units	Dts...	Stats	Rules...
	KesmeMak	1	5	None	Time Series	Min(Once), First
	KesmeMak.1	1	1	None	Time Series	Min(Once)
	KesmeMak.2	1	1	None	Time Series	Min(Once)
	KesmeMak.3	1	1	None	Time Series	Min(Once)
	KesmeMak.4	1	1	None	Time Series	Min(Once)
	KesmeMak.5	1	1	None	Time Series	Min(Once)
	KesKuy	INFINITE	1	None	Time Series	Oldest, Min(Once)
	CapMak	1	4	None	Time Series	Oldest, First
	CapMak.1	1	1	None	Time Series	Oldest
	CapMak.2	1	1	None	Time Series	Oldest
	CapMak.3	1	1	None	Time Series	Oldest
	CapMak.4	1	1	None	Time Series	Oldest
	CapKuy	INFINITE	1	None	Time Series	Oldest, FIFO
	Raf	1000	1	None	Time Series	Oldest
	Konduit_Balya	INF	1	None	Time Series	Oldest, FIFO

Şekil 3.5. Makineler

Şekil 3.6.'da Kesme Kuyruğu ve Şekil 3.7.'de Kesme Makinesi için verilen karar kuralları kesme kuyruğundan çıkış kuralı ve kesme makinesine giriş kuralı olarak "Once" öznitelik değişkenin (attribute) en küçük değeri öncelikli olarak belirlenmiştir .

Decision Rules for KesKuy

Selecting Incoming Entities

Oldest by Priority

Random

Least Available Capacity

Last Selected Location

Highest Attribute Value

Lowest Attribute Value

Attributes

Queuing for Output

No Queuing

First In, First Out (FIFO)

Last In, First Out (LIFO)

By Type

Highest Attribute Value

Lowest Attribute Value

Attributes

Selecting a Unit

First Available

By Turn

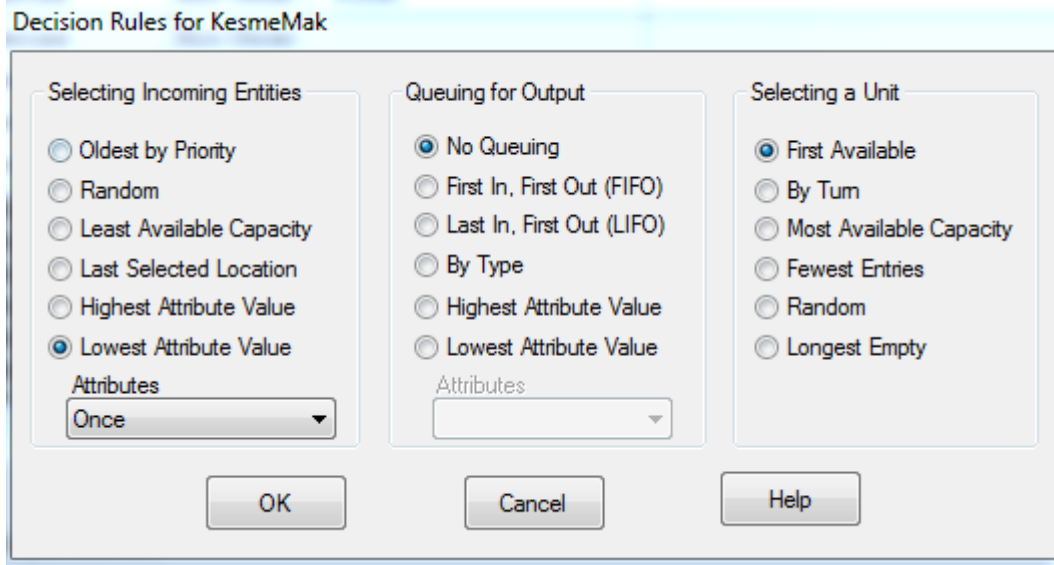
Most Available Capacity

Fewest Entries

Random

Longest Empty

Şekil 3.6. Kesme Kuyruğu Karar Kuralları



Şekil 3.7. Kesme Makinesi Karar Kuralları

- **Eleman tanımlamaları (Entities)**

Üretime giren konduit üretilen yarı mamullerin ilk maddesidir ve kanban kartı ile beraber operasyonları geçmektedir (Şekil 3.8).

Icon	Name	Speed (mpm)	Stats
	Kart	1	Time Series
	Konduit	1	Time Series

Şekil 3.8. Elemanlar (Entities)

- **Yol Bağlantıları (Path Networks)**

Kaynaklar, lokasyonlar arası yolculuk eden dinamik unsurlar olarak modellendiğinde yol bağlantılarını (path networks) takip ederler. Lokasyonlar arası hareket eden elemanlar (entities) eğer routing → move logic içerisinde tanımlanırsa yol bağlantılarını (path networks) takip edebilirler (Promodel 2011 Help System).

Net_Operator olarak tanımlanan yol bağlantısı üzerinde hareket eden operator kesme kuyruğunda bekleyen ilk iş emrini Operasyon_1'i gerçekleştirmek üzere ilk boşalan kesme makinesine götürür ve işi başlatır. Net_TPalet1 yolu üzerinde hareket ederen Tpalet1, kesme makinesinde Operasyon_1'i bitirilen malzemeyi Operasyon_2 olan çapak alma kuyruğuna taşıırken, Net_Tpalet2 yolu üzerinde hareket ederen Tpalet2 malzemeyi

çapak alma (Operasyon_2) operasyonuna götürmek üzere kullanılır. Net_Forklift yolunu kullanan forklift ise tüm işleri tamamlanmış yarı mamulü depolanmak üzere rafa taşınmasında kullanılır (Şekil 3.9).

Graphic...	Name	Type	T/S
	Net_Operator	Passing	Time
	Net_TPalet1	Passing	Speed & Distance
	Net_Forklift	Passing	Speed & Distance
	Net_TPalet2	Passing	Speed & Distance

Paths...	Interfaces...	Mapping...	Nodes
1	2	0	2
1	2	0	2
2	3	0	3
1	2	0	2

Şekil 3.9. Yol bağlantıları (path networks) tanımlamaları

Her bir yol bağlantısı (path network) için bir yol (path) tanımlanmalıdır (Şekil 3.10).

From	To	BI	Time
N1	N2	Bi	0

Şekil 3.10. Net_Operator yolu (Path) tanımlaması

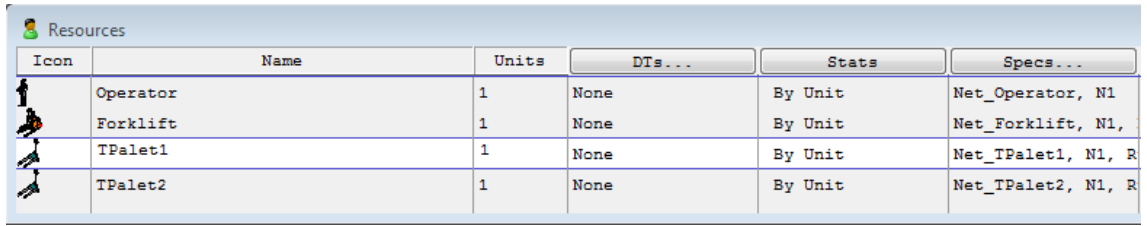
Benzer şekilde Net_Tpalet1, Net_Tpalet2 ve Net_Forklift için yol (path) tanımlamaları yapılır. Şekil 3.4. Benzetim Modelinin Promodel Ekran Görüntüsü'nde görünen mavi çizgiler yol bağlantılarını ifade etmektedir.





- **Kaynaklar (Resources)**

Kaynak bir insan, çeşitli işlerde veya fonksiyonlarda kullanılan ekipman veya araç parçalarıdır. Kaynaklar taşıma faaliyeti yapan elemanlar, makinelerdeki operasyonları gerçekleştiren elemanlar veya bakım işlerinde kullanılan elemanlar olarak tanımlanmış olabilirler. Hareketsiz, statik, kaynaklar olabileceği gibi bir yol (path network) üzerinde

hareket eden dinamik kaynaklarda olabilir. Kaynakların duruş zamanları (downtimes) olabilir (Promodel 2011 Help System).

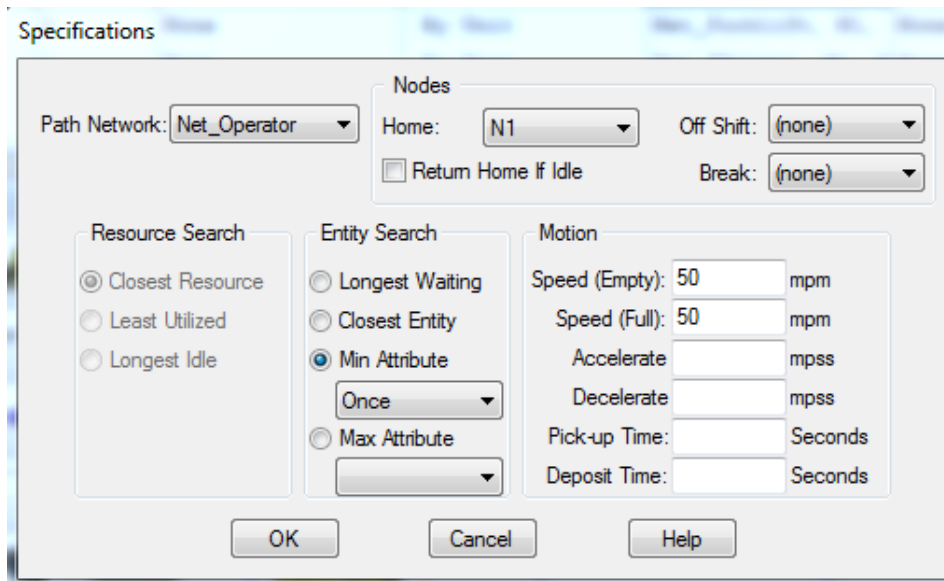
Üretim esnasında kullanacağımız kaynaklar Resources kısmında tanımlanır. Operasyon_1’i gerçekleştiren Kesme Makinelerine iş atayan Operator, Operasyon_2’den çıkan yarı mamulü depolanmak üzere raflara yerleştirmek için kullandığımız Forklift, Operasyon_1’de tamamlanan malzemeyi (konduit) Operasyon_2 kuyruğuna götüren transpalet (TPalet1) ve kuyruktan alarak Operasyon_2’ye götüren transpalet (TPalet2) kaynak olarak tanımlanmıştır (Şekil 3.11).



Icon	Name	Units	DTs...	Stats	Specs...
	Operator	1	None	By Unit	Net_Operator, N1
	Forklift	1	None	By Unit	Net_Forklift, N1, R
	TPalet1	1	None	By Unit	Net_TPalet1, N1, R
	TPalet2	1	None	By Unit	Net_TPalet2, N1, R

Şekil 3.11. Kaynaklar

“Specifications” kısmında kaynağın kullanımına ilişkin tanımlamalar yapılmıştır. Kaynağın hangi kurala göre kullanılacağı “Entity Search” kısmında belirlenir. FCFS kuralını uygulamak için oluşturulan ve yarı mamullerin depodan çıkma zamanlarını tutan “Once” attribute değişkeni “Min Attribute” olarak atanmıştır. Diğer kaynaklarda bu kısım “Longest Waiting” olarak işaretlenmiştir (Şekil 3.12).



Specifications

Path Network:

Nodes

Home: Off Shift:

Return Home If Idle Break:

Resource Search

Closest Resource

Least Utilized

Longest Idle

Entity Search

Longest Waiting

Closest Entity

Min Attribute

Max Attribute

Motion

Speed (Empty): mpm

Speed (Full): mpm

Accelerate: mpss

Decelerate: mpss

Pick-up Time: Seconds

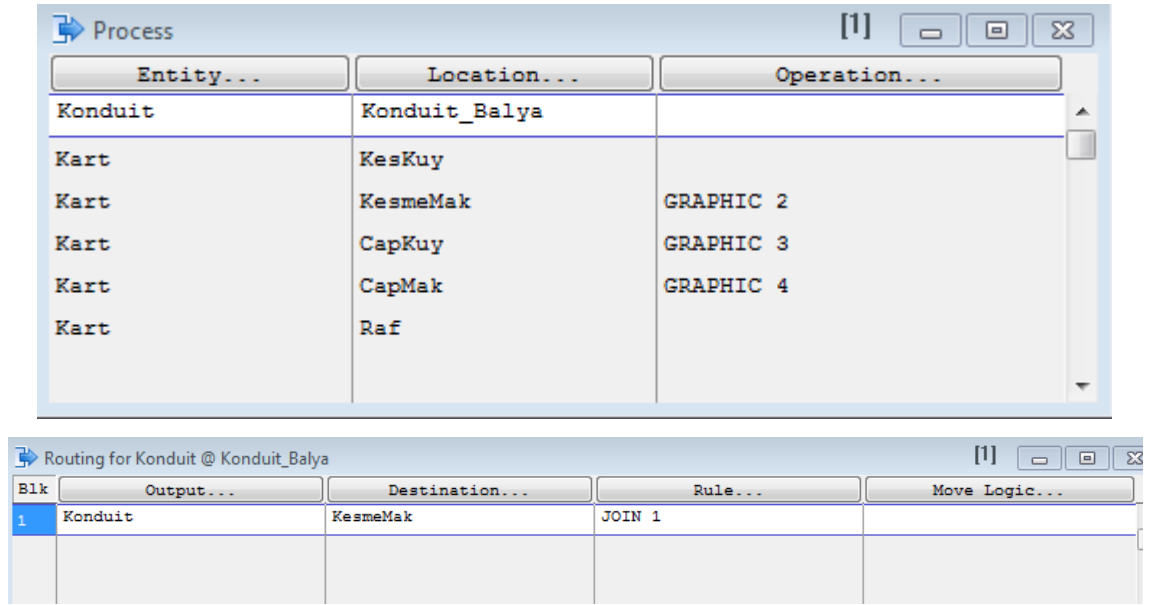
Deposit Time: Seconds

OK Cancel Help

Şekil 3.12. Operator kaynak kullanım tanımlaması (Specifications)

- **İşlemler (Processing)**

“Processing” kısmında elemanların (entity) sistem içindeki rotaları ve girdikleri her lokasyonda geçirdikleri operasyonlar tanımlanır. “Arrivals” tablolarında belirtildiği şekilde bir eleman bir sisteme girdikten çıkana kadar başına gelen herşey “Processing” kısmında tanımlanır (Promodel 2011 Help System). “Processing” kısmında “Process” ve “Routing” pencereleri bulunur. Bu pencerelerin içinde tanımlamalar yapılır (Şekil 3.13).

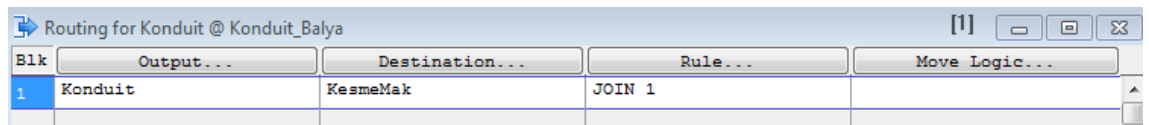


Şekil 3.13. Process (İşlemler) ve Routing (Rotalama) pencereleri

Process penceresindeki her eleman (entity) ve lokasyon için operasyonlar “Operation..” kısmında tanımlanmıştır. Herbir operasyonun açıklamaları Entity-Location eşleşmelerine göre aşağıda sıra ile yapılacaktır.

Konduit - Konduit-Balya:

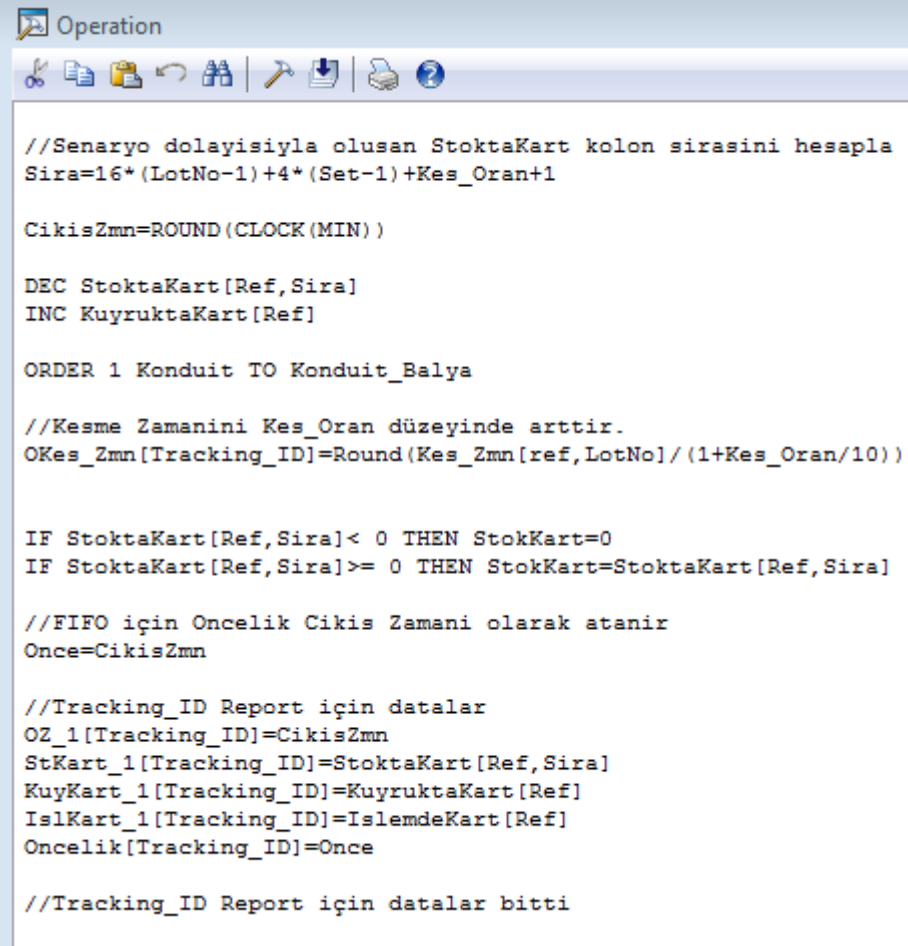
Herhangi bir operasyon yoktur. Yarı mamullerin üretiminde kullanılan “Konduit” lerin başlangıç durum ve konumlarını tanımlamak için oluşturulan bir eşleşmedir. Rotalama kısmında JOIN kuralı ile kesme makinesine (KesmeMak) hareketi sağlanır (Şekil 3.14).



Şekil 3.14. Kart – KesKuy Rotalama Penceresi

Kart – KesKuy :

Kanban kartının kesime girme önceliğini belirleyen algoritma Operation kısmında aşağıdaki gibi ifade edilmiştir (Şekil 3.15).



```
//Senaryo dolayisiyla olusan StoktaKart kolon sirasini hesapla
Sira=16*(LotNo-1)+4*(Set-1)+Kes_Oran+1

CikisZmn=ROUND(CLOCK(MIN))

DEC StoktaKart[Ref,Sira]
INC KuyruktaKart[Ref]

ORDER 1 Konduit TO Konduit_Balya

//Kesme Zamanini Kes_Oran düzeyinde arttir.
OKes_Zmn[Tracking_ID]=Round(Kes_Zmn[ref, LotNo]/(1+Kes_Oran/10))

IF StoktaKart[Ref,Sira]< 0 THEN StokKart=0
IF StoktaKart[Ref,Sira]>= 0 THEN StokKart=StoktaKart[Ref,Sira]

//FIFO için Oncelik Cikis Zamani olarak atanir
Once=CikisZmn

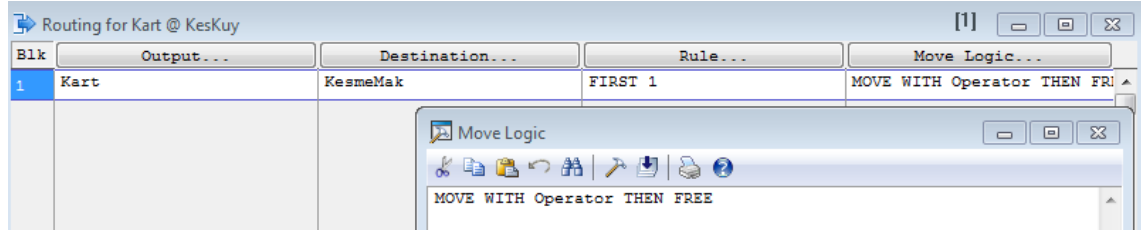
//Tracking_ID Report için datalar
OZ_1[Tracking_ID]=CikisZmn
StKart_1[Tracking_ID]=StoktaKart[Ref,Sira]
KuyKart_1[Tracking_ID]=KuyruktaKart[Ref]
IslKart_1[Tracking_ID]=IslemdeKart[Ref]
Oncelik[Tracking_ID]=Once

//Tracking_ID Report için datalar bitti
```

Şekil 3.15. Kart – KesKuy Operasyon Penceresi

Birden çok senaryo üzerinde denemeler yapılabilmesi için “Sira” değişkeni oluşturulmuştur. Bu değişken kanban kart adetlerinin excel içerisinde ilgili sayfanın hangi sütununda alınacağını belirtmektedir. Lot (LotNo), setup (Set) ve kesme süresi (Kes_Oran) alternatif senaryolarına göre belirlenen kart adetlerinin, ki ilk stok adedidir, yazıldığı sütunu belirtir. “CikisZmn” değişkeni kasanın depodan çıktığı yani kartın kesme kuyruğu sırasına geldiği zamanı kaydetmektedir. “OKes_Zmn” değişkeni, ilgili referansın (yarı mamul) kullanılan senaryo için kesme zamanını hesaplamaktadır. “StoktaKart” değişkeni ilgili referans için stokta ne kadar kart (=kasa) kaldığını hesaplayan değişkendir. “Once” değişkeni kesime alma önceliğini belirleyen

değişkendir ve FCFS kuralında “CikisZmn” değişkenine eşittir. Son kısımda işlem gören kasaların verileri izleme numarası olarak adlandırabileceğimiz Tracking_ID’lerine göre değişkenlere aktarılmaktadır. Sonra bu veriler text dosyalara kayıt edilerek analizler için saklanmaktadır.

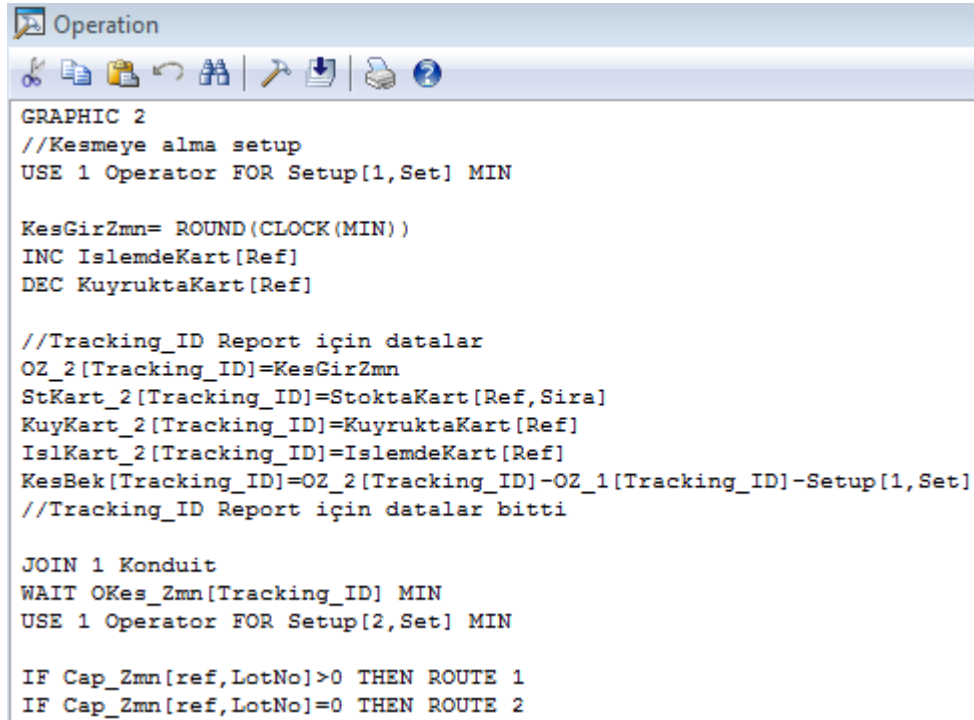


Şekil 3.16. Kart – KesKuy Rotalama Penceresi

Kart kesme kuyruğunda (KesKuy) bekledikten sonra kesme makinesine (KesmeMak) alınır. Kesme makinesine alınması esnasında Operator ile hareket eder (Şekil 3.16).

Kart – KesmeMak :

Kart, kesme makinesinde işleme alındığında oluşan operasyonlar belirtilmiştir (Şekil 3.17).



Şekil 3.17. Kart – KesmeMak Operasyon Penceresi

USE komutu ile 1 Operator 1. Setup'ı yapmaktadır. "KesGirZmn" deęiřkeni kartın (yarı mamulün) kesilmeye bařlandığı zamanı göstermektedir. Bu durum gerekleřtięinde "IslemdeKart" adedi bir adet artırılmakta, "KuyruktaKart" adedi bir adet azaltılmaktadır. Tracking_ID izleme numarası ile veriler kayıt edilmektedir. JOIN komutu ile "Konduit" kesilmek üzere retime alınmaktadır. WAIT komutu ile "OKes_Zmn" deęiřkeni ile ifade edilen kesme suresi kadar beklenilmekte, yani kesim iřlemi yapılmaktadır. Kesim iřlemi bittikten sonra kasanın son iřlemleri iin USE komutu ile bir operator 2. Setup'ı yapmaktadır.

Eęer "Cap_Zmn">0 ise ROUTE 1 zerinden apak alma kuyruęuna gitmekte, eęer "Cap_Zmn"=0 ise ROUTE 2 zerinden direkt depolama rafına konulmaktadır.

Blk	Output...	Destination...	Rule...	Move Logic...
1	Kart	CapKuy	FIRST 1	MOVE WITH TPalet1 THEN FREE
2	Kart	Raf	FIRST 1	MOVE WITH Forklift THEN FREE

řekil 3.18. Kart – KesmeMak Rotalama Penceresi

ROUTE 1 ve 2, rotalama (řekil 3.18) kuralında; apak kuyruęuna (CapKuy) gidecekse transpalet ile, rafa kaldırılacaksa forklift ile tařıma yapılır. Move Logic kısmında tanımlanmıştır.

Kart – CapKuy :

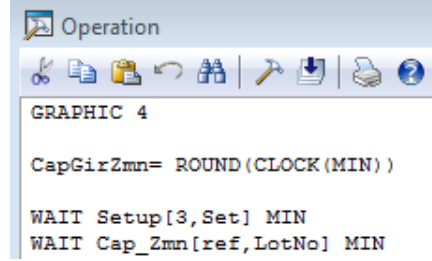
Kuyrukta bekleme operasyonudur ve Operasyon pencesinde grafik deęiřtirme dıřında (GRAPHIC 3) bir iřlemi yoktur. Rotalama (Routing) penceresinde gideceęi yer apak makinesi (CapMak) olarak tanımlanır ve First Available (FIRST 1) kuralına gre hareket saęlanır. Transpalet ile tařıma iin kullanılır (řekil 3.19).

Blk	Output...	Destination...	Rule...	Move Logic...
1	Kart	CapMak	FIRST 1	MOVE WITH TPalet2 THEN FREE

řekil 3.19. Kart – CapKuy Rotalama Penceresi

Kart – CapMak :

Çapak makinesinde çapak alma işleminin yapılmasıdır.



Şekil 3.20. Kart – CapMak Operasyon Penceresi

“CapGirZmn” değişkeni ile işleme başlama zamanı kayıt edilir. WAIT komutu ile hazırlık işlemlerinin tamamlanması için 3. Setup süresi kadar beklenir ve sonra yine WAIT komutu ile “Cap_Zmn” süresi kadar olan çapak alma işlemi yapılır (Şekil 3.20).

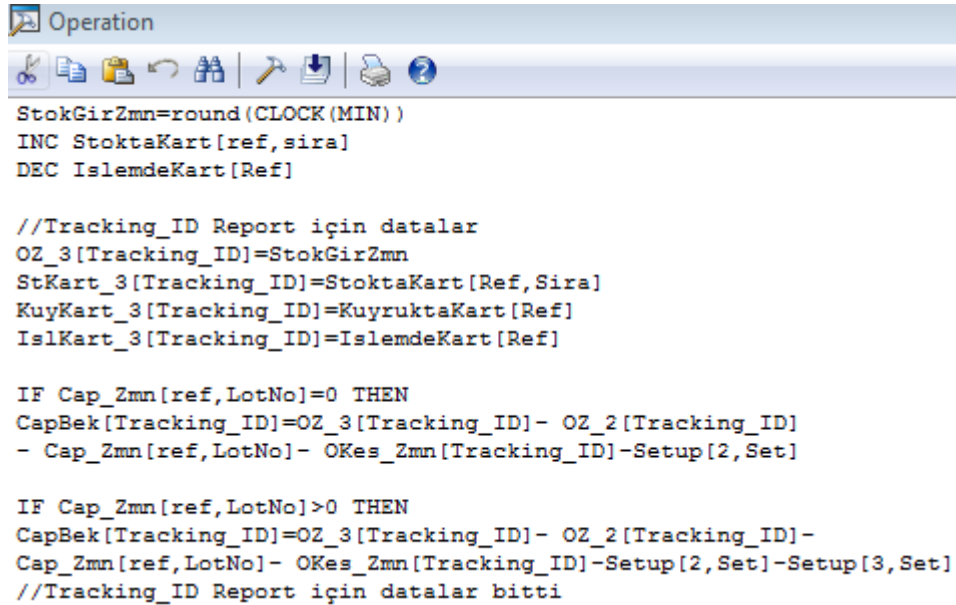


Şekil 3.21. Kart – CapMak Rotalama Penceresi

Çapak alma işlemi tamamlanan yarı mamul forklift ile rafa kaldırılır (Şekil 3.21).

Kart – Raf :

Bu kısımda üretim işlemleri tamamlanmış yarı mamul kanban kartı ile beraber rafa kaldırılır (Şekil 3.23).



Şekil 3.22. Kart – Raf Operasyon Penceresi

```

//Tracking_ID Report olustur
Senaryo=LotNo*100+Set*10+Kes_Oran
WRITE ID_Rapor, Senaryo
WRITE ID_Rapor, Tracking_ID
WRITE ID_Rapor, Ref
WRITE ID_Rapor, OZ_1[Tracking_ID]
WRITE ID_Rapor, StKart_1[Tracking_ID]
WRITE ID_Rapor, KuyKart_1[Tracking_ID]
WRITE ID_Rapor, IslKart_1[Tracking_ID]
WRITE ID_Rapor, LT[Tracking_ID]
WRITE ID_Rapor, Baslaengec[Tracking_ID]
WRITE ID_Rapor, Oncelik[Tracking_ID]
WRITE ID_Rapor, OZ_2[Tracking_ID]
WRITE ID_Rapor, StKart_2[Tracking_ID]
WRITE ID_Rapor, KuyKart_2[Tracking_ID]
WRITE ID_Rapor, IslKart_2[Tracking_ID]
WRITE ID_Rapor, KesBek[Tracking_ID]
WRITE ID_Rapor, OZ_3[Tracking_ID]
WRITE ID_Rapor, StKart_3[Tracking_ID]
WRITE ID_Rapor, KuyKart_3[Tracking_ID]
WRITE ID_Rapor, IslKart_3[Tracking_ID]
WRITE ID_Rapor, OKes_Zmn[Tracking_ID]
WRITE ID_Rapor, Cap_Zmn[ref, LotNo]
WRITELINE ID_Rapor, CapBek[Tracking_ID]

```

Şekil 3.22. Kart – Raf Operasyon Penceresi (Devam)

“StokGirZmn” değişkeni ile stoklama zamanı kayıt edilir. “StoktaKart” değişkeni ile stok miktarı bir artırılırken “İşlemdeKart” değişkeni bir azaltılarak işlemdeki stok değeri yenilenir. Raporlama kayıtları için Tracking_ID bazında değerler alınır veya hesaplanır.

Şekil 3.22 devamından görülen işlemler “ID_Rapor” isimli text dosyasına kaydedilen verilerin yazılmasıdır.

Blk	Output...	Destination...	Rule...	Move Logic...
1	Kart	EXIT	FIRST 1	

Şekil 3.23. Kart – Raf Rotalama Penceresi

- **Varişlar (Arrivals)**

Bir elemanın (entity) sisteme girmesi varış (arrival) olarak adlandırılır. Bir varış aşağıdaki kayıtların tanımlamasıyla oluşturulur:

- Variş başına gelen eleman adedi
- Varişların frekansı
- Variş yeri

- İlk varış zamanı
- Toplam varış adedi (Occurrences)

Entity...	Location...	Qty Each...	First Time...	Occurrences	Frequency	Logic...	isabl
Konduit	Konduit_Balya	1		1			No

Şekil 3.24. Varış (Arrivals) penceresi

- **Öznitelik Değişkenleri (Attributes):**

Attribute'ler değişkenlere benzemektedir ama özellikle bir lokasyona veya elemana (entity) eklidirler ve onlara ait bir bilgiyi içerirler. (Promodel 2011 Help System)

ID	Type	Classification	Notes...
Ref	Integer	Ent	
Once	Integer	Ent	
Tracking_ID	Integer	Ent	
Kalan_Sip	Integer	Ent	

Şekil 3.25. Öznitelik Değişkenleri (Attributes) Penceresi

Benzetim modelimizde 4 adet attribute değişken bulunmaktadır (Şekil 3.25).

Ref: Yarı mamullerin referans bilgilerini taşımaktadır.

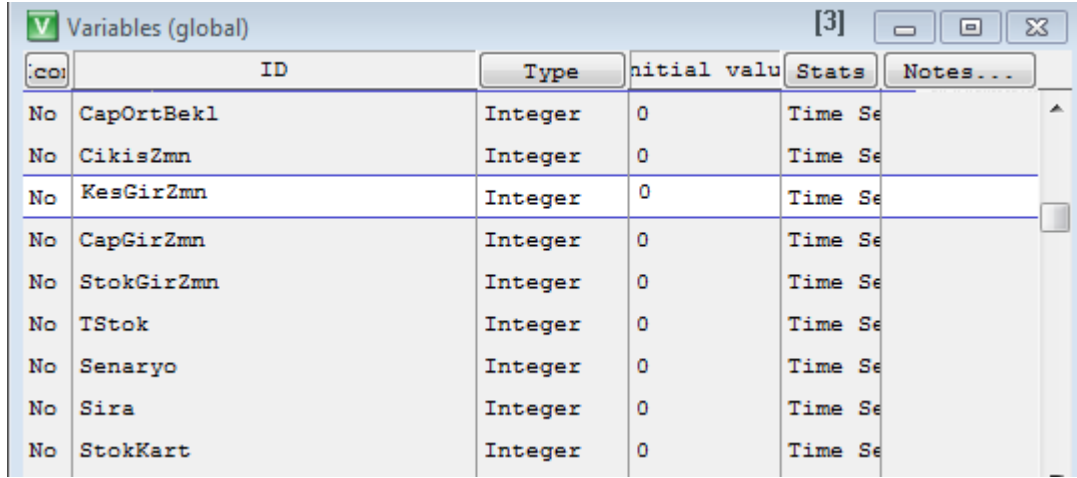
Once: Önceliklendirme değişkenidir.

Tracking_ID: Bir referans birden fazla kez sisteme girer ve çıkar. Dolayısıyla sistem içinde oluşan hareketleri izlemek için başka bir tekil bir numaraya ihtiyaç vardır.

Traking_ID gelen her karta verilen tekil numaradır.

Kalan_Sip: Bir referansa ait ilgili hafta kalan sipariş miktarıdır.

- **Değişkenler (Variables):**



ID	Type	Initial Value	Stats	Notes...
CapOrtBekl	Integer	0	Time Se	
CikisZmn	Integer	0	Time Se	
KesGirZmn	Integer	0	Time Se	
CapGirZmn	Integer	0	Time Se	
StokGirZmn	Integer	0	Time Se	
TStok	Integer	0	Time Se	
Senaryo	Integer	0	Time Se	
Sira	Integer	0	Time Se	
StokKart	Integer	0	Time Se	

Şekil 3.26. Değişkenler (Variables) Penceresi

CapOrtBekl: Dakika cinsinden çapak ortalama bekleme süresidir.

CikisZmn: Kanban kartının kesme makinesine geldiği yani yarı mamülün raftan çıkarak montaj hattına götürüldüğü zamandır.

KesGirZmn: Kanban kartının kesmeye (Operasyon_1) girdiği zamandır.

CapGirZmn: Kanban kartının çapak almaya (Operasyon_2) girdiği zamandır.

StokGirZmn: Yarı mamulün üretiminin tamamlandığı ve kanban kartı ile beraber rafa konulduğu zamandır.

TStok: Yarı mamule ait toplam kasa adedi = kart adedir.

Senaryo: Değişken lot miktarı, setup süresi ve kesme süresi oluşturulan senaryolara verilen numaradır.

Sira: Senaryoya uygun olarak hesaplanan kanban kart adetlerinin yazılı olduğu excel sütununun numarasıdır.

StokKart: Bir yarı mamüle ait rafta dolu olarak bulunan kanban kart = kasa adedi.

- **Diziler (Arrays):**

Diziler gerçek ve tam sayı değerler ihtiva eden hücre matrisleridir. Biz dizideki her hücre bir değişken gibi çalışır ve bir değişkenin kullanıldığı her yerde kullanılabilir. Tek boyutlu bir dizi tek sütunlu değerler halinde düşünülebilir. İki boyutlu diziler ise bir excel sayfası gibi çoklu sütunlardan oluşabilir. Promodel’de bir dizi en fazla 20 boyutlu olabilir. (Promodel 2011 Help System)

Diziler bir dosya içerisine yerleştirilerek dosya içerisinden çekilebilir. Eşitlik kurarak bir dizi değişkenine değer atanabilir. Örn: Arr1[5,2]=18. Benzetimde kullanılan dizileri Şekil 3.27’de gösterilmiştir.

ID	Dimensions	Type	Import File...
StoktaKart	100, 48	Integer	ProM_Dosyalar\Arrays
KuyruktaKart	100	Integer	
IslemdeKart	100	Integer	
StKart_1	7000	Integer	
StKart_2	7000	Integer	
StKart_3	7000	Integer	
KuyKart_1	7000	Integer	
KuyKart_2	7000	Integer	
KuyKart_3	7000	Integer	
IslKart_1	7000	Integer	
IslKart_2	7000	Integer	
IslKart_3	7000	Integer	
KesBek	7000	Integer	
CapBek	7000	Integer	
OZ_1	7000	Integer	
OZ_2	7000	Integer	
OZ_3	7000	Integer	
Oncelik	7000	Integer	
KartTukZmn	100, 3	Integer	ProM_Dosyalar\Arrays
Kes_Zmn	100, 3	Integer	ProM_Dosyalar\Arrays
Cap_Zmn	100, 3	Integer	ProM_Dosyalar\Arrays
Setup	3, 10	Integer	ProM_Dosyalar\Arrays
OKes_Zmn	7000	Integer	

Şekil 3.27. Diziler (Arrays) Penceresi

StoktaKart : Oluşturulan senaryolara uygun olarak sütun numarası ve referansa ait satır numarası içeren iki boyutlu bir dizidir. Bir yarı mamulün (referansın) kaç adet kanban kartına ait kasanın dolu olarak rafta bulunduğunu ifade eder.

KuyruktaKart: Bir referansa ait kaç adet kanban kartının kesme kuyruğunda (Operasyon_1) beklediğini gösterir.

IslemdeKart: Bir referansa ait kaç adet kanban kartının operasyona alındığını gösterir.

StKart_1 (2ve 3): Operasyon kademeleri boyunca 3 noktada bir referansa (yarı mamul) ait rafta bulunan dolu kasa (kanban kartlı) stok değerlerini tutan dizi değişkendir. 1; kesme kuyruğuna gelme, 2; kesmeye alma (Operasyon_1), 3; rafa kaldırılma yani stoğa girme durumunu ifade eder.

KuyKart_1 (2ve 3): Operasyon kademeleri boyunca 3 noktada bir referansa (yarı mamul) ait kesme kuyruğunda bekleyen kanban kartı adedini tutan dizi değişkendir.

IslKart_1 (2ve 3): Operasyon kademeleri boyunca 3 noktada bir referansa (yarı mamul) ait işlem gören kanban kartı adedini tutan dizi değişkendir.

KesBek: Bir Tracking _ID ile ifade edilen kanban kartının kesme kuyruğunda (Operasyon_1) bekleme süresidir.

CapBek: Bir Tracking _ID ile ifade edilen kanban kartının çapak alma kuyruğunda (Operasyon_2) bekleme süresidir.

OZ_1 (2 ve 3): Operasyona başlama zamanıdır. 1; kesme kuyruğuna gelme zamanı, 2; kesmeye alma zamanı (Operasyon_1), 3; rafa kaldırılma yani stoğa girme zamanıdır.

Oncelik: “Once” değişkeni ile belirlenen öncelik değerini ilgili kanban kartına ait tracking_ID’ye bağlı olarak saklayan dizi değişkendir.

KartTukZmn: Bir referansa (yarı mamul) ait kasanın, dolayısıyla kanban kartının, montaj hattında tüketilme süresidir.

Kes_Zmn: Bir referansa (yarı mamul) ait kasanın, dolayısıyla kanban kartının, kesim işleminin (Operasyon_1) tamamlanma süresidir.

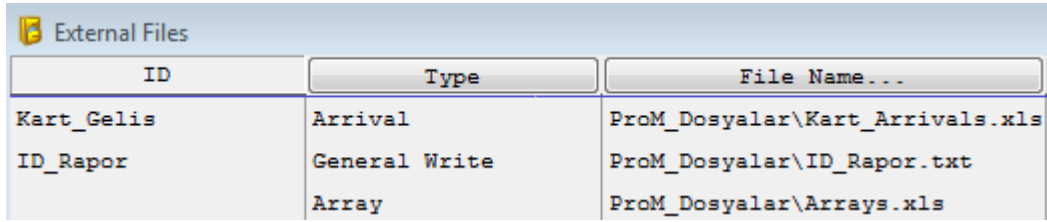
Cap_Zmn: Bir referansa (yarı mamul) ait kasanın, dolayısıyla kanban kartının, چاپک alma işleminin (Operasyon_2) tamamlanma süresidir.

Setup: Herhangi bir operasyon öncesi gereken setup süresidir.

OKes_Zmn: Senaryo oluşturmak için kesme süresine tesir eden çarpan.

- **Dış Dosyalar (External Files):**

Dış dosyalar benzetime veri alınması veya benzetimde oluşan verilerin yazılması için kullanılabilirler. Dış dosya içinden operasyon zamanları, varış çizelgeleri, vardiya çizelgeleri ve dış alt programlar (subroutine) alınabilir (Şekil 3.28).



ID	Type	File Name...
Kart_Gelis	Arrival	ProM_Dosyalar\Kart_Arrivals.xls
ID_Rapor	General Write	ProM_Dosyalar\ID_Rapor.txt
	Array	ProM_Dosyalar\Arrays.xls

Şekil 3.28. Dış Dosyalar (External Files) Penceresi

Kart_Gelis: Kanban kartlarının kesme kuyruğuna geliş zamanlarıdır. Rassal olarak belirlenip bir dosya üzerinden Promodel'e aktarılmıştır. İçeren dosya "Arrival" dosya tipindedir.

ID_Rapor: Benzetime sonuçlarının kayıt edildiği dosyadır.

Array dosyası: Dizi değişkenlerin başlangıç değerlerinin yer aldığı dosyadır. Özellikle StoktaKart değişkeni başlangıçtaki stok değerlerini benzetime aktarması açısından önemlidir.

FCFS sıralama yöntemine göre çalışan Promodel benzetim modelinin metni **EK 1'de** verilmiştir.

3.6.2. En Küçük Kesme Süreli (Shortest Bottleneck Process Time - SBPT) Benzetim Modeli

FCFS modelinden tek farkı önceliklendirmenin darboğaz operasyon olan kesme operasyonun süresine göre yapılmasıdır. Ek küçük kesme süreli referansa (yarı mamul) ait kart önce kesme işlemine alınır. Kesme süreleri eşit ise önce gelen işleme alınır. Processing bölümünün ilgili kısmında önceliklendirme “Once” değişkenine yapılan atama ile belirlenmiştir.

`Once=CikisZmn`

ifadesiyle FCFS modelinde “Once” değişkenine kasanın depodan montaj hattına götürüldüğü zamanı temsil eden “CikisZmn” değişkeni atanmıştır. (EK 1, s:77)

SBPT modelinde ise “Once” değişkenine birinci ve darboğaz operasyonu olan kesme operasyonun süresi atanmıştır.

`Once=OKes_Zmn[Tracking_ID]`

Şekil 3.6 ve Şekil 3.7’te görüldüğü üzere kesme kuyruğundan çıkarak kesme makinesine giriş kuralı olarak “Once” öznitelik değişkeninin (attribute) en küçük değeri öncelikli olarak belirlenmiştir.

3.6.3. En Küçük Toplam İşlem Süreli (Total Shortest Process Time – Total SPT) Benzetim Modeli

FCFS modelinden tek farkı önceliklendirmenin toplam işlem süresine göre yapılmasıdır. Ek küçük toplam işlem süreli referansa (yarı mamul) ait kart önce kesme işlemine alınır. İşlem süreleri eşit ise önce gelen işleme alınır. Processing bölümünün ilgili kısmında FCFS modelinde “Once=CikisZmn” iken

`Once=OKes_Zmn[Tracking_ID]+Cap_Zmn[Ref, LotNo]`

ifadesiyle Total SPT modelinde toplam işlem süresi olarak atanmıştır. Kesme kuyruğundan çıkarak kesme makinesine giriş kuralı olarak “Once” öznitelik değişkeninin (attribute) en küçük değeri öncelikli olarak belirlenmiştir (Şekil 3.6 ve 3.7).

3.6.4. Stok Bitiş Zamanı (Due Date) Benzetim Modeli

FCFS modelinden tek farkı önceliklendirmenin stok bitiş zamanına göre yapılmasıdır. İlk tükenen referansa (yarı mamul) ait kart önce kesme işlemine alınır. İşlem süreleri eşit ise önce gelen işleme alınır.

“DueDate” raftaki stok ve işlemdeki stoğun tükenme zamanı olara Promodel içerisinde aşağıdaki gibi hesaplanır ve “Once” değişkenine atanır.

$$\text{DueDate} = \text{CikisZmn} + (\text{StokKart} + \text{IslemdeKart}[\text{Ref}]) * \text{KartTukZmn}[\text{Ref}, \text{LotNo}]$$
$$\text{Once} = \text{DueDate}$$

FCFS modelinde “Once=CikisZmn” olarak atanan öncelik değeri DueDate modelinde “Once=DueDate” yani stok bitiş zamanı olarak atanmıştır. Bu sıralama şeklinde de kesme kuyruğundan çıkararak kesme makinesine giriş kuralı olarak “Once” öznitelik değişkeninin (attribute) en küçük değeri öncelikli olarak belirlenmiştir.

3.6.5. Algoritma 1, Sipariş Kontrollü Sıralama Benzetim Modeli

Algoritma 1 olarak adlandırdığımız sıralama yöntemini sınamak üzere oluşturulmuş olan benzetim modelidir. Algoritma 1 önceliklendirme değerini hesaplarken sipariş olup olmadığını kontrol ederek ilerler.

Algoritma 1, Sipariş Kontrollü Sıralama, Öncelik Değer Hesabı (3.30):

$$OP_j^\alpha = \begin{cases} OS_1 + WSPT_j^1, & D_j'(\alpha) \leq 0 \\ OS_2 + WSPT_j^1, & l_j(\alpha) > t_j^2 \\ OS_3 + WSPT_j^1, & l_j(\alpha) \leq t_j^2 \end{cases} \quad (3.30)$$

Promodel benzetim modelini açıklamak üzere Promodel Text dosyası EK 2’de verilmiştir.

Promodel Processing kısmında yapılan aşağıdaki işlemlerle öncelik değeri hesaplanır.

```
IF BeklZmn[RefNo] <= 0 THEN KesOnce[RefNo] = 400 + Prio[X]
IF BeklZmn[RefNo] > 0 THEN KesOnce[RefNo] = 300 + Prio[X]

IF KalanSip[RefNo] - StokKart - IslemdeKart[RefNo] <= 0
THEN KesOnce[RefNo] = DonemSonuna + Prio[X]

IF KesOnce[RefNo] > KesOnce[KesRef] THEN KesRef = RefNo
IF KesOnce[RefNo] = KesOnce[KesRef] THEN
{
IF BeklZmn[RefNo] < BeklZmn[KesRef] THEN KesRef = RefNo
}
}
```

Program satırında alınan aşağıdaki ifade ilgili hafta için kalan siparişlere karşılık mevcut stoğun yeterli olup olmadığını kontrol eder. Eğer stok yeterli ise öncelik değerini en alt düzeye düşürerek dönem sonuna atar.

```
IF KalanSip[RefNo] - StokKart - IslemdeKart[RefNo] <= 0 THEN
KesOnce[RefNo] = DonemSonuna + Prio[X]
```

3.6.6. Algoritma 2, Sipariş ve Sıfır Stok Kontrollü Sıralama, Benzetim Modeli

Algoritma 2 olarak adlandırdığımız sıralama yöntemini sınamak üzere oluşturulmuş olan benzetim modelidir. Algoritma 2 önceliklendirme değerini hesaplarken stok durumunu ve sipariş durumunu kontrol ederek ilerler.

Algoritma 2, Sipariş ve Sıfır Stok Kontrollü Sıralama, Öncelik Değer Hesabı (3.31):

$$OP_j^\alpha = \begin{cases} OS_1 + WSPT_j^1, & D_j'(\alpha) \leq 0 \\ OS_2 + WSPT_j^1, & l_j(\alpha) > t_j^2 \\ OS_3 + WSPT_j^1, & l_j(\alpha) \leq t_j^2 \\ OS_4 + WSPT_j^1, & K_j^S(\alpha) = 0 \end{cases} \quad (3.31)$$

Promodel Processing kısmında yapılan aşağıdaki işlemlerle öncelik değeri hesaplanır.

```
IF BeklZmn[RefNo]<=0 THEN KesOnce[RefNo]=400+Prio[X]
IF BeklZmn[RefNo]>0 THEN KesOnce[RefNo]=300+Prio[X]
IF KalanSip[RefNo]-StokKart-IslemdeKart[RefNo]<= 0 THEN
KesOnce[RefNo]=DonemSonuna+Prio[X]
```

```
IF StokKart+IslemdeKart[RefNo]<= 0 THEN
KesOnce[RefNo]=500+Prio[X]
```

```
IF KesOnce[RefNo]>KesOnce[KesRef] THEN KesRef=RefNo
IF KesOnce[RefNo]=KesOnce[KesRef] THEN
{
IF BeklZmn[RefNo]<BeklZmn[KesRef] THEN KesRef=RefNo
}
}
```

Öncelik değeri hesabı dışında Algoritma 1 için oluşturulan benzetim modelinden bir farkı yoktur.

```
IF StokKart+IslemdeKart[RefNo]<= 0 THEN
KesOnce[RefNo]=500+Prio[X]
```

ifadesinin eklenmesi ile Algoritma 2, Algoritma 1'den farklılaştırılmıştır. Bu ifade stok durumunu kontrol etmekte ve eğer stok miktarı "0" olursa öncelik düzeyini en öncelikli düzey olan $500+Prio[X]$ seviyesine taşımaktadır.

3.6.7. Algoritma 3, Sıfır Stok Kontrollü Sıralama, Benzetim Modeli

Algoritma 3 olarak adlandırdığımız sıralama yöntemini sınamak üzere oluşturulmuş olan benzetim modelidir. Algoritma 3 önceliklendirme değerini hesaplariken stok durumunu kontrol ederek ilerler.

Algoritma 3, Sıfır Stok Kontrollü Sıralama, Öncelik Değer Hesabı (3.32):

$$OP_j^\alpha = \begin{cases} OS_1 + WSPT_j^1, & D_j'(\alpha) \leq 0 \\ OS_2 + WSPT_j^1, & l_j(\alpha) > t_j^2 \\ OS_3 + WSPT_j^1, & l_j(\alpha) \leq t_j^2 \end{cases} \quad (3.32)$$

Promodel Processing kısmında yapılan aşağıdaki işlemlerle öncelik değeri hesaplanır.

```
IF BeklZmn[RefNo] <= 0 THEN KesOnce[RefNo] = 400 + Prio[X]
IF BeklZmn[RefNo] > 0 THEN KesOnce[RefNo] = 300 + Prio[X]
```

```
IF StokKart + IslemdeKart[RefNo] <= 0 THEN
KesOnce[RefNo] = 500 + Prio[X]
```

```
IF KesOnce[RefNo] > KesOnce[KesRef] THEN KesRef = RefNo
IF KesOnce[RefNo] = KesOnce[KesRef] THEN
{
IF BeklZmn[RefNo] < BeklZmn[KesRef] THEN KesRef = RefNo
}
}
```

Öncelik değeri hesabı dışında Algoritma 1 ve 2 için oluşturulan benzetim modelinden bir farkı yoktur. Algoritma 2'den farklı olarak aşağıdaki ifadenin yazılımdan çıkarılması ile sipariş kontrolü iptal edilmiştir.

```
IF KalanSip[RefNo] - StokKart - IslemdeKart[RefNo] <= 0 THEN
KesOnce[RefNo] = DonemSonuna + Prio[X]
```

3.7. Benzetimlerde Kullanılan Veriler

Bölüm 3.5.1. Üretim Ortamına Ait Veriler'de gerçek bir üretim ortamından alınan verilerin analizine yer verilmişti. İlgili üretim ortamına ait 8 hafta, 49 gün ve toplam 147 vardiyayı kapsayan, 492 881 dk montaj üretim süresi içeren gerçek veriler elimizde bulunmaktadır. Sıralama yöntemlerinin test edilmesi için Bonferroni En İyiyi Seçme Yaklaşımı kullanılmış, yaklaşımın iki kademeli yönteminde kullanılmak üzere kapasite kullanım oranları farklı, tüketilen yarı mamul adet ve sıraları rassal olan benzetim verileri oluşturulmuştur. Test yaklaşımının ilk kademe testleri için 10 adet, ikinci kademe testleri için ilave 20 adet ve toplamda 30 adet, 8 haftalık benzetim verisi kullanılmıştır. Kapasite kullanım oranı darboğaz operasyon olan Operasyon_1'e ait olup, şu şekilde hesaplanmıştır.

Operasyon 1 toplam işlem ve setup süresi, TOS_1 , (3.38) denklemiyle ifade edilmiştir.

Montaj hattına son kasa götürme zamanı $\max\{t_j^1\}$ 'dir.

Operasyon_1 (Darboğaz Operasyon) Kapasite Kullanım Oranı KKO_1 aşağıdaki şekilde hesaplanır.

$$KKO_1 = \frac{TOS_1 / m_1}{\max\{t_j^1\}} \quad (3.48)$$

Çizelge 3.9. Test Verileri Özeti

Test No	TMS (dk)	TKM (Adet)	$\sum K_j$ (Adet)	$\max\{t_j^1\}$ (dk)	KKO_1	G
1	477737	2674	133	63033	80,8%	0
2	492881	2830	152	63114	83,8%	0
3	508219	2959	152	62916	86,2%	0
4	502081	2896	152	61518	87,4%	0
5	525469	2872	147	63035	89,6%	0
6	525066	2999	168	61691	91,7%	0
7	544152	3117	181	62905	93,5%	0
8	553376	3257	189	63148	94,3%	0
9	554485	3220	240	61850	96,4%	0
10	565693	3230	243	61615	98,4%	0
11	480012	2742	142	63012	81,8%	0
12	487395	2843	150	62956	83,3%	0
13	491026	2882	148	63151	83,4%	0
14	489825	2806	147	62913	84,3%	0
15	498434	2918	148	62969	84,6%	0
16	506527	2979	155	63076	85,9%	0
17	503886	2886	186	62803	86,3%	0
18	518107	3000	156	62603	87,1%	0
19	508469	2965	162	63112	87,4%	0
20	513732	2979	159	62952	87,4%	0
21	520414	2952	163	62933	87,8%	0
22	518980	3047	165	61680	90,1%	0
23	532516	3081	163	63020	90,5%	0
24	536612	3074	175	61855	92,0%	0
25	545847	3136	177	63038	93,0%	0
26	543732	3121	179	62919	93,6%	0,3
27	544793	3178	189	61668	94,7%	0,3
28	561537	3290	240	61688	97,4%	0,3
29	577745	3299	242	61863	99,6%	0,3
30	572655	3287	243	61731	99,7%	0,3

Test Verileri Özetini içeren Çizelge 3.9'da görüldüğü gibi kullanılan test verilerinde darboğaz operasyonun (Operasyon_1) kapasite kullanım oranı %80,8'den başlayarak %99,7'ye kadar ulaşmaktadır. Hesaplanan kart adetleri 133 ile 243 adet arasında değişmektedir. Montaja giden, montajın talep ettiği kasa adedi 2674 ile 3299 adet arasında değişmektedir.

Benzetimlerde kullanılan kart varışları veri özetleri EK 3 - EK 12 arasındaki eklerde yer almaktadır.

4. BULGULAR ve TARTIŞMA

Önerilen algoritmanın üç alternatif yaklaşımı bulunmaktadır. Bu üç alternatif yaklaşımın temel işleyişi aynı olmakla beraber sahip oldukları ek kontroller bakımından ayrılmaktadırlar. Bu ayrışma aşağıdaki gibi özetlenebilir.

- Algoritma 1 : Üretim yapılan haftada ilgili yarı mamul için başka sipariş kalıp kalmadığını kontrol eder.
- Algoritma 2 : Üretim yapılan haftada ilgili yarı mamul için başka sipariş kalıp kalmadığını ve stok seviyesinin sıfır düzeyine düşüp düşmediğini kontrol eder.
- Algoritma 3 : İlgili yarı mamul için stok seviyesinin sıfır düzeyine düşüp düşmediğini kontrol eder.

Önerilen üç alternatifli algoritma genel kabul gören diğer dört sıralama yaklaşımına karşı test edilmiştir. Bu yaklaşımlar; İlk Gelen İlk İşlenir (FCFS), En Kısa Toplam İşlem Süresi (STPT), Teslim Zamanı (DD) ve En Kısa Darboğaz İşlem Süresi (SBPT) sıralama yaklaşımlarıdır.

Benzetimler kapasite kullanımının değişip değişmemesine bağlı olarak iki tip test verisi üzerinde yapılmıştır.

- Değişken sipariş miktarları ile değişken kapasite kullanım oranlarına sahip test verileri ve
- Sabit sipariş miktarları ile aynı kapasite kullanım oranına sahip farklı sıralamalı test verileri.

4.1. Değişken Sipariş Miktarları ile Yapılan Test Sonuçları

Üretim süreçlerinin kullandıkları kapasite müşteri siparişlerinin yoğunluğunun değişmesine göre farklılıklar gösterir. Kapasite kullanım oranı yükseldikçe süreç önü kuyruklar artar, işlerin gecikme olasılığı yükselir. Sıralama yöntemlerinin değişken sipariş ortamında oluşan farklı kapasite kullanım oranlarında göstereceği performans genel kullanımda göstereceği performans açısından gösterge olacaktır. Sipariş adedinin 2674 – 3299 kasa arasında, buna karşılık kapasite kullanım oranının %80 - %100

aralığında deđiřtiđi üretim ortamlarında sıralama yöntemlerinin oluşturacağı montaj hattı duruş süreleri tespit edilmek istenmiştir.

Yapılan ilk 10 adet benzetim sonucunda aşağıdaki sonuçlar elde edilmiştir (Çizelge 4.1 ve Çizelge 4.2) :

Çizelge 4.1. Benzetim Sonuçları (Deđişken Sipariş)

Test No	1	2	3	4	5
Toplam Montaj Süresi (dk)	477737	492881	508219	502081	525469
Benzetim Bitiş Zamanı (dk)	63033	63114	62916	61518	63035
Kapasite Kullanım Oranı (%)	80,8	83,8	86,2	87,4	89,6
Kart Adedi	133	152	152	152	147
Üretilen Kasa Adedi	2674	2830	2959	2896	2872
Test No	6	7	8	9	10
Toplam Montaj Süresi (dk)	525066	544152	553376	554485	565693
Benzetim Bitiş Zamanı (dk)	61691	62905	63148	61850	61615
Kapasite Kullanım Oranı (%)	91,7	93,5	94,3	96,4	98,4
Kart Adedi	168	181	189	240	243
Üretilen Kasa Adedi	2999	3117	3257	3220	3230

Çizelge 4.2. Montaj Hattı Duruş Oranları (%) (Deđişken Sipariş)

Yöntem	Test No	1	2	3	4	5
Algoritma 1		0,03	0,01	0,02	0,01	1,18
Algoritma 2		0,02	0,01	0,00	0,00	0,03
Algoritma 3		0,04	0,01	0,03	0,03	0,06
FCFS		1,32	3,79	7,12	4,13	5,03
STPT		1,12	1,55	5,53	3,68	4,67
SBPT		1,15	1,00	3,43	3,32	4,14
Due Date		0,84	1,09	5,58	3,09	5,02
Yöntem	Test No	6	7	8	9	10
Algoritma 1		0,27	0,16	0,91	2,79	6,52
Algoritma 2		0,00	0,00	0,05	0,02	0,08
Algoritma 3		0,04	0,01	0,01	0,00	0,05
FCFS		21,98	9,92	34,63	42,36	31,84
STPT		16,18	10,69	22,43	24,29	27,65
SBPT		13,81	6,21	19,22	29,73	26,26
Due Date		13,12	6,69	28,20	35,41	30,72

Benzetimlerin toplam montaj süreleri farklı olduğundan sonuçlar oran olarak ifade edilmiştir. Çizelge 4.3.'de, sonuçları daha anlaşılır kılmak amacıyla oran olarak belirtilen sonuçlar 10 000 dk süreli bir benztimin sonucu gibi genişletilerek gösterilmiştir.

Çizelge 4.3. Montaj Hattı Duruş Süreleri (dak/10 000 dak) (Değişken Sipariş)

Yöntem	Test No	1	2	3	4	5
Algoritma 1		2,95	0,75	1,75	0,70	118,18
Algoritma 2		1,76	0,53	0,41	0,00	2,59
Algoritma 3		3,66	0,99	2,75	3,17	5,52
FCFS		131,96	379,30	719,79	412,68	502,50
STPT		111,82	154,82	552,69	367,57	466,78
SBPT		114,50	100,27	342,82	331,60	413,63
Due Date		84,23	109,17	557,50	309,09	502,05

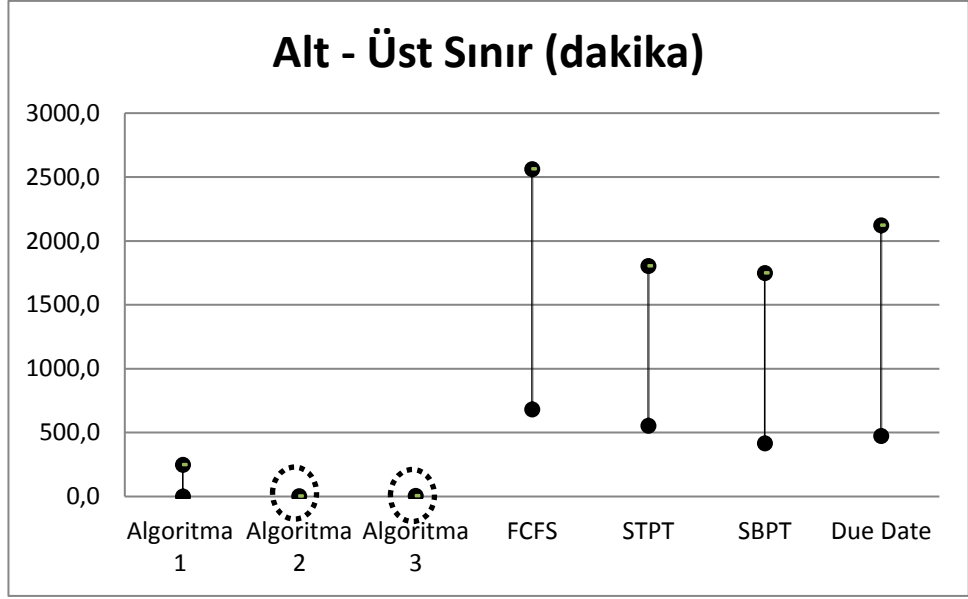
Yöntem	Test No	6	7	8	9	10
Algoritma 1		27,16	15,51	90,73	279,45	651,73
Algoritma 2		0,40	0,39	5,13	1,57	7,94
Algoritma 3		4,15	0,50	0,54	0,00	5,27
FCFS		2197,65	991,91	3463,40	4235,68	3184,06
STPT		1617,53	1068,75	2242,89	2429,41	2765,14
SBPT		1381,03	621,19	1921,95	2973,03	2625,49
Due Date		1311,63	668,93	2819,80	3540,73	3071,84

4.2. Alt - Üst Sınır Diyagramı Üzerinde Karşılaştırma

Örnek büyüklüğü (R) 10 ve güvenlik aralığı $(1-\alpha) \%95$ olmak üzere aşağıdaki istatistiksel sonuçlar elde edilmiştir (Çizelge 4.4).

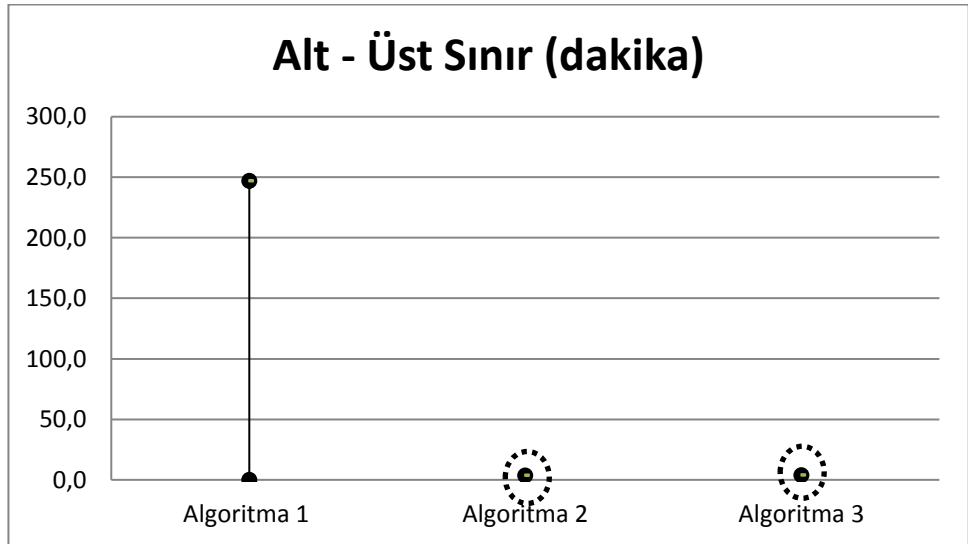
Çizelge 4.4. İlk Test İstatistiksel Sonuçlar (Dakika) (Değişken Sipariş)

Yöntem	Ortalama	Std Sapma	Güvenirlik Sınırı	Alt Sınır	Üst Sınır
Algoritma 1	119	207	128	0	247
Algoritma 2	2,1	2,6	1,6	0,5	3,7
Algoritma 3	2,7	2,0	1,3	1,4	3,9
FCFS	1622	1516	940	682	2562
STPT	1178	1009	625	552	1803
SBPT	1083	1076	667	416	1750
Due Date	1297	1331	825	473	2122

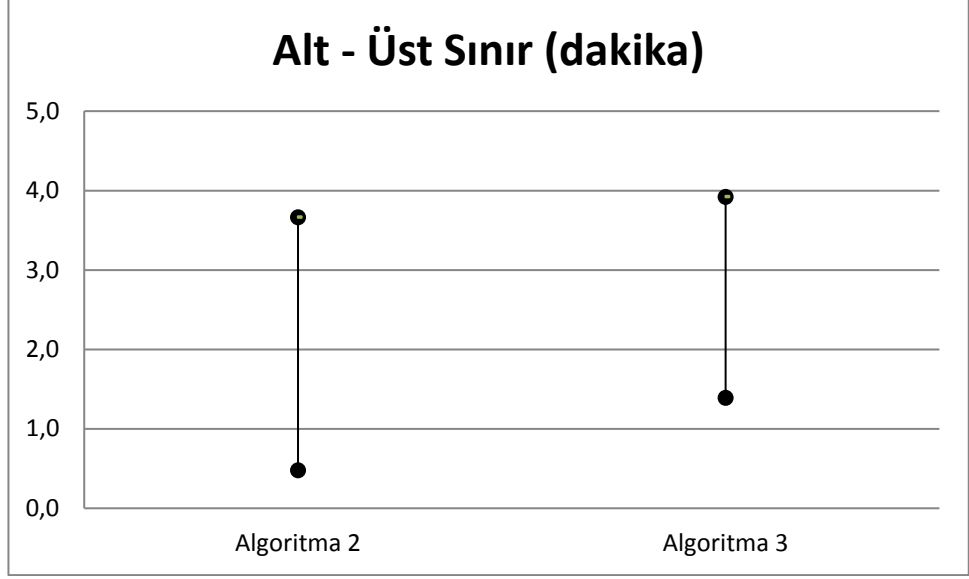


Şekil 4.1. Tüm Yöntemler Alt - Üst Sınır Diyagramı

Tüm Yöntemler Çizgi Diyagramında (Şekil 4.1.) görüldüğü üzere Algoritma 1, 2 ve 3 diğer yöntemlerden (FCFS, STPT, SBPT ve Due Date) sınırlar ve ortalama değerler olarak oldukça aşağıda kalmaktadırlar ve hiçbir noktada çakışmamaktadırlar. Algoritma 2 ve 3 grafik ölçeğinde nokta boyutunda kaldıklarından kesik çizgili daire ile işaretlemiştir (Şekil 4.2).



Şekil 4.2. Algoritma 1,2 ve 3 Alt - Üst Sınır Diyagramı



Şekil 4.3. Algoritma 2 ve 3 Alt - Üst Sınır Diyagramı

Şekil 4.2 ve Şekil 4.3'ü incelediğimizde Algoritma 2 ve 3'ün sınır değer aralığı 0-4 dk düzeyinde iken Algoritma 1'in sınır aralığı 0-250 dk arasında değişmektedir. Bu durumda Algoritma 1'in diğer iki alternatif algoritmaya göre daha iyi sonuçlar üretmediği söylenebilir.

Algoritma 2 ve 3'ün sınır değerleri ve ortalamaları birbirine yakındır. Bu nedenle ilave bir sınaama yöntemi ile hangi algoritmanın daha iyi olduğuna karar vermek gerekecektir.

4.3. Bonferroni En İyiyi Seçme Yaklaşımı

Algoritma 2 ve 3'ten hangisinin daha iyi olduğunu belirlemek için Bonferroni En İyiyi Seçme Yaklaşımı kullanılmıştır.

K adet sistem tasarımı olduğu varsayılınsın ve i . sistemin beklenen performans değeri θ_i olsun. En iyinin en büyük beklenen performans olarak tanımlandığı durumda, daha geniş bir seviyede en iyinin hangisi olduğu bulunmaya çalışılmaktadır. (Banks ve Ark. 1995, s: 497)

İki Seviyeli Bonferroni Prosedürü:

1. Anlamlı farklılık değeri ε , doğru seçim olasılık değeri olarak $1-\alpha$ ve ile seviye örneklem büyüklüğü olarak

$$R_0 \geq 10 \quad (4.1)$$

$$t_{\alpha/(K-1), R_0-1} \quad (4.2)$$

değerleri bulunur.

2. R_0 adet tekrar yapılarak tüm sistemler için ($i=1,2, \dots, K$) $Y_{1i}, Y_{2i}, \dots, Y_{R_0i}$ ortalama değerleri hesaplanır.
3. İlk seviye örneklem ortalaması hesaplanır. $\bar{Y}_i, i=1,2,\dots,K$.
Tüm $i \neq j$ için fark karelerinin örneklem varyansı hesaplanır.

$$S_{ij}^2 = \frac{1}{R_0-1} \sum_{r=1}^{R_0} (Y_{ri} - Y_{rj} - (\bar{Y}_i - \bar{Y}_j))^2 \quad (4.3)$$

4. İkinci seviye örneklem büyüklüğü hesaplanır.

$$R = \max \left\{ R_0, \max_{i \neq j} \left[\frac{t^2 S_{ij}^2}{\varepsilon^2} \right] \right\} \quad (4.4)$$

5. $R-R_0$ adet ek tekrar yapılır. Her bir sistem $i=1,2, \dots, K$ için ortalama değerler elde edilir.

$$Y_{R_0+1,i}, Y_{R_0+2,i}, \dots, Y_{R,i} \quad (4.5)$$

6. Tüm örneklerin ortalaması her bir sistem $i=1,2, \dots, K$ için hesaplanır.

$$\bar{Y}_i = \frac{1}{R} \sum_{r=1}^R Y_{ri} \quad (4.6)$$

7. En büyük Y_i ortalamaya sahip sistemi en iyi olarak seç. Ayrıca tüm $i=1,2, \dots, K$ için güvenlik aralığı aşağıdaki şekilde belirlenir.

$$\min\{0, \bar{Y}_i - \max_{i \neq j} \bar{Y}_j - \varepsilon\} \leq \theta_i - \max_{i \neq j} \bar{Y}_j \leq \max\{0, \bar{Y}_i - \max_{i \neq j} \bar{Y}_j + \varepsilon\} \quad (4.7)$$

4.3.1. Prosedürün Benzetim Sonuçlarına Uygulanması

Çizelge 4.3. Montaj Hattı Duruş Süreleri (dak/10 000 dak) (Değişken Sipariş)'den algoritma 2 ve 3 değerleri aşağıdaki çizelgeye taşınmıştır. Her benzetim için algoritmalar aynı rassal varyansları kullandığından bağlı (correlated) örneklem olarak değerlendirilir.

Çizelge 4.5. Bağlı Örneklem Sonuç Tablosu
Ortalama Süreler (dak)

Benzetim	Alg. 2	Alg. 3	Fark
	Y_{r1}	Y_{r2}	D_{12}
1	1,8	3,7	-2
2	0,5	1,0	0
3	0,4	2,8	-2
4	0,0	3,2	-3
5	2,6	5,5	-3
6	0,4	4,2	-4
7	0,4	0,5	0
8	5,1	0,5	5
9	1,6	0,0	2
10	7,9	5,3	3
\bar{D}_i	Fark Ortalaması		-0,58
S_{D_i}	Fark Standart Sapması		2,77
$S_{D_i}^2$	Fark Varyansı		7,70
$\frac{S_{D_i}}{\sqrt{R}}$	Fark Standart Hata		0,88

1. Anlamlı farklılık değeri $\varepsilon = 1$ dk, %95 olasılıklı en iyi seçebilmek için $1 - \alpha = 0.95$ alınır. $R_0 = 10$ 'dur. t dağılımı yüzde değerleri tablosunda $t_{0,05,9} = 1.83$ olarak elde edilir.
2. Bağlı Örneklem Sonuç Tablosu oluşturulur (Çizelge 4.5).
3. Bağlı Örneklem Sonuç Tablosundan $S_{D_i}^2 = 7.70$ hesaplanmış olarak görülmektedir.
4. Oluşturulması gereken örneklem adedi hesabı yapılır.

$$R = \max \left\{ 10, \left\lceil \frac{1.83^2 * 7.70}{1^2} \right\rceil \right\} = \max \{10, [25.78]\} = 26$$

5. $30 \geq 26$ adet toplam örneklem sayısına ulaşmak için sipariş adetleri ve talep sıraları rassal olarak değiştirilmiş $30 - 10 = 20$ adet daha benzetim uygulanmıştır. Sonuçlar Çizelge 4.6 ve 4.7'de gösterilmiştir.

Çizelge 4.6. Algortima 2 ve 3 İlave Benzetim Sonuçları (Değişken Sipariş)

Test No	11	12	13	14	15
Toplam Montaj Süresi (dak)	480012	487395	491026	489825	498434
Benzetim Bitiş Zamanı (dak)	63012	62956	63151	62913	62969
Kapasite Kullanım Oranı (%)	81,8	83,3	83,4	84,3	84,6
Kart Adedi	142	150	148	147	148
Üretilen Kasa Adedi	2742	2843	2882	2806	2918

Test No	16	17	18	19	20
Toplam Montaj Süresi (dak)	506527	503886	518107	508469	513732
Benzetim Bitiş Zamanı (dak)	63076	62803	62603	63112	62952
Kapasite Kullanım Oranı (%)	85,9	86,3	87,1	87,4	87,4
Kart Adedi	155	186	156	162	159
Üretilen Kasa Adedi	2979	2886	3000	2965	2979

Test No	21	22	23	24	25
Toplam Montaj Süresi (dk)	520414	518980	532516	536612	545847
Benzetim Bitiş Zamanı (dk)	62933	61680	63020	61855	63038
Kapasite Kullanım Oranı (%)	87,8	90,1	90,5	92,0	93,0
Kart Adedi	163	165	163	175	177
Üretilen Kasa Adedi	2952	3047	3081	3074	3136

Test No	26	27	28	29	30
Toplam Montaj Süresi (dk)	543732	544793	561537	577745	572655
Benzetim Bitiş Zamanı (dk)	62919	61668	61688	61863	61731
Kapasite Kullanım Oranı (%)	93,6	94,7	97,4	99,6	99,7
Kart Adedi	179	189	240	242	243
Üretilen Kasa Adedi	3121	3178	3290	3299	3287

Çizelge 4.7. Algoritma 2 ve 3 İlave Benzetim Sonuçları, Montaj Hattı Duruş Oran (%)

Yöntem	Test No	11	12	13	14	15
Algoritma 2		0,018	0,000	0,000	0,030	0,002
Algoritma 3		0,025	0,005	0,016	0,005	0,005

Yöntem	Test No	16	17	18	19	20
Algoritma 2		0,011	0,000	0,000	0,003	0,003
Algoritma 3		0,024	0,000	0,010	0,038	0,007

Yöntem	Test No	21	22	23	24	25
Algoritma 2		0,000	0,000	0,018	0,037	0,000
Algoritma 3		0,002	0,001	0,012	0,020	0,028

Yöntem	Test No	26	27	28	29	30
Algoritma 2		0,059	0,039	0,070	0,662	0,112
Algoritma 3		0,011	0,026	0,053	0,501	0,291

Çizelge 4.8. 10000 dk ile Genişletilmiş Montaj Hattı Duruş Süreleri

Yöntem	Test No	11	12	13	14	15
Algoritma 2		1,79	0,00	0,00	3,02	0,24
Algoritma 3		2,52	0,45	1,57	0,47	0,50

Yöntem	Test No	16	17	18	19	20
Algoritma 2		1,07	0,00	0,00	0,30	0,31
Algoritma 3		2,45	0,00	1,02	3,78	0,68

Yöntem	Test No	21	22	23	24	25
Algoritma 2		0,00	0,00	1,80	3,65	0,00
Algoritma 3		0,25	0,13	1,24	1,98	2,84

Yöntem	Test No	26	27	28	29	30
Algoritma 2		5,92	3,85	6,96	66,21	11,18
Algoritma 3		1,07	2,64	5,34	50,14	29,09

30 benzetimin sonucu olarak 10000 dk ile Genişletilmiş Montaj Hattı Duruş Sürelerinin (Çizelge 4.8.) ortalama değerleri hesaplandığında;

Algoritma 2 ortalama montaj hattı duruş süresi = 4,23 dk

Algoritma 3 ortalama montaj hattı duruş süresi = 4,49 dk olarak bulunur.

Bonferroni En İyiyi Seçme Yaklaşımına göre elde edilen sonuçlar doğrultusunda Algoritma 2'nin en iyi algoritma olduğu söylenebilir.

4.4. Sabit Sipariş ile Yapılan Test Sonuçları

Sipariş adetleri sabit tutularak kapasite kullanımının değişmediği, farklı tüketim sıralamalarıyla yarı mamüllerin montaj hatlarına verildiği üretim verileri için benzetim uygulanmıştır. Bu amaçla gerçek üretim verilerinin sıralamaları rassal olarak değiştirilmiş ve 20 farklı üretim sıralamasıyla aynı adette yarı mamül tüketiminin yapıldığı üretim verileri üretilmiştir. Oluşan sonuçlar Çizelge 4.9. ve 4.10.'da özetlenmiştir.

Çizelge 4.9. Benzetim Sonuçları (Sabit Sipariş)

Test No	1	2	3	4	5
Toplam Montaj Süresi (dk)	492881	492881	492881	492881	492881
Benzetim Bitiş Zamanı (dk)	62928	62874	62906	62938	62953
Kapasite Kullanım Oranı (%)	84,03	84,10	84,06	84,02	84,00
Kart Adedi	152	152	152	152	152
Üretilen Kasa Adedi	2830	2830	2830	2830	2830
Test No	6	7	8	9	10
Toplam Montaj Süresi (dk)	492881	492881	492881	492881	492881
Benzetim Bitiş Zamanı (dk)	62940	62932	62985	62934	63206
Kapasite Kullanım Oranı (%)	84,01	84,03	83,95	84,02	83,66
Kart Adedi	152	152	152	152	152
Üretilen Kasa Adedi	2830	2830	2830	2830	2830
Test No	11	12	13	14	15
Toplam Montaj Süresi (dk)	492881	492881	492881	492881	492881
Benzetim Bitiş Zamanı (dk)	63028	62913	63066	62914	63024
Kapasite Kullanım Oranı (%)	83,90	84,05	83,85	84,05	83,90
Kart Adedi	152	152	152	152	152
Üretilen Kasa Adedi	2830	2830	2830	2830	2830
Test No	16	17	18	19	20
Toplam Montaj Süresi (dk)	492881	492881	492881	492881	492881
Benzetim Bitiş Zamanı (dk)	62821	63108	63162	63021	63003
Kapasite Kullanım Oranı (%)	84,17	83,79	83,72	83,91	83,93
Kart Adedi	152	152	152	152	152
Üretilen Kasa Adedi	2830	2830	2830	2830	2830

Çizelge 4.10. Montaj Hattı Duruş Oranları (%) (Sabit Sipariş)

Yöntem	Test No	1	2	3	4	5
Algoritma 1		0,0065	0,0053	0,0095	0,0306	0,1073
Algoritma 2		0,0067	0,0065	0,0136	0,0189	0,0059
Algoritma 3		0,0237	0,0154	0,0450	0,0248	0,0351
FCFS		1,5781	2,5795	4,8519	2,0262	5,9087
STPT		1,9861	1,7388	3,2618	2,0924	4,1335
SBPT		1,7209	1,4616	2,4722	1,4358	3,8642
Due Date		1,0850	1,5706	2,6582	1,1725	3,7358

Yöntem	Test No	6	7	8	9	10
Algoritma 1		0,0394	0,0071	0,0341	0,0039	0,0183
Algoritma 2		0,0006	0,0410	0,0032	0,0049	0,0156
Algoritma 3		0,0101	0,0392	0,0172	0,0049	0,0511
FCFS		3,1440	5,2917	1,4959	1,2666	3,7382
STPT		1,7935	3,6037	2,3547	1,1889	2,5361
SBPT		1,2583	2,1244	1,7521	0,8901	2,6785
Due Date		1,6363	3,2158	0,9647	0,7075	2,1275

Yöntem	Test No	11	12	13	14	15
Algoritma 1		0,0063	0,0207	0,0349	0,0187	0,0566
Algoritma 2		0,0006	0,0030	0,0215	0,0000	0,0000
Algoritma 3		0,0116	0,0047	0,0160	0,0223	0,0006
FCFS		3,2274	1,9453	1,5657	2,4651	1,7862
STPT		2,9825	1,6231	1,3969	1,8043	1,7767
SBPT		2,4690	1,4423	1,0810	1,7720	1,2060
Due Date		1,7018	0,9830	0,6192	1,6458	1,4756

Yöntem	Test No	16	17	18	19	20
Algoritma 1		0,0069	0,0010	0,0006	0,0592	0,0438
Algoritma 2		0,0083	0,0000	0,0020	0,0274	0,0024
Algoritma 3		0,0219	0,0026	0,0071	0,0020	0,0077
FCFS		3,4672	2,3866	2,6913	3,4477	2,2358
STPT		2,5842	1,2959	2,0366	2,7459	2,1137
SBPT		1,9567	1,0717	1,5529	0,9181	1,3117
Due Date		2,2468	1,2674	1,6099	1,6059	1,3411

Benzetimlerin toplam montaj süreleri farklı olduğundan sonuçlar oran olarak ifade edilmiştir. Anlaşılmasını kolaylaştırmak amacıyla oran olarak belirtilen sonuçlar 10000 dak süreli bir benzetimin sonucu gibi genişletilmiştir (Çizelge 4.11).

Çizelge 4.11. Montaj Hattı Duruş Süreleri (dak/10000 dak) (Sabit Sipariş)

Yöntem	Test No	1	2	3	4	5
Algoritma 1		0,65	0,53	0,95	3,06	10,73
Algoritma 2		0,67	0,65	1,36	1,89	0,59
Algoritma 3		2,37	1,54	4,50	2,48	3,51
FCFS		157,81	257,95	485,19	202,62	590,87
STPT		198,61	173,88	326,18	209,24	413,35
SBPT		172,09	146,16	247,22	143,58	386,42
Due Date		108,50	157,06	265,82	117,25	373,58

Yöntem	Test No	6	7	8	9	10
Algoritma 1		3,94	0,71	3,41	0,39	1,83
Algoritma 2		0,06	4,10	0,32	0,49	1,56
Algoritma 3		1,01	3,92	1,72	0,49	5,11
FCFS		314,40	529,17	149,59	126,66	373,82
STPT		179,35	360,37	235,47	118,89	253,61
SBPT		125,83	212,44	175,21	89,01	267,85
Due Date		163,63	321,58	96,47	70,75	212,75

Yöntem	Test No	11	12	13	14	15
Algoritma 1		0,63	2,07	3,49	1,87	5,66
Algoritma 2		0,06	0,30	2,15	0,00	0,00
Algoritma 3		1,16	0,47	1,60	2,23	0,06
FCFS		322,74	194,53	156,57	246,51	178,62
STPT		298,25	162,31	139,69	180,43	177,67
SBPT		246,90	144,23	108,10	177,20	120,60
Due Date		170,18	98,30	61,92	164,58	147,56

Yöntem	Test No	16	17	18	19	20
Algoritma 1		0,69	0,10	0,06	5,92	4,38
Algoritma 2		0,83	0,00	0,20	2,74	0,24
Algoritma 3		2,19	0,26	0,71	0,20	0,77
FCFS		346,72	238,66	269,13	344,77	223,58
STPT		258,42	129,59	203,66	274,59	211,37
SBPT		195,67	107,17	155,29	91,81	131,17
Due Date		224,68	126,74	160,99	160,59	134,11

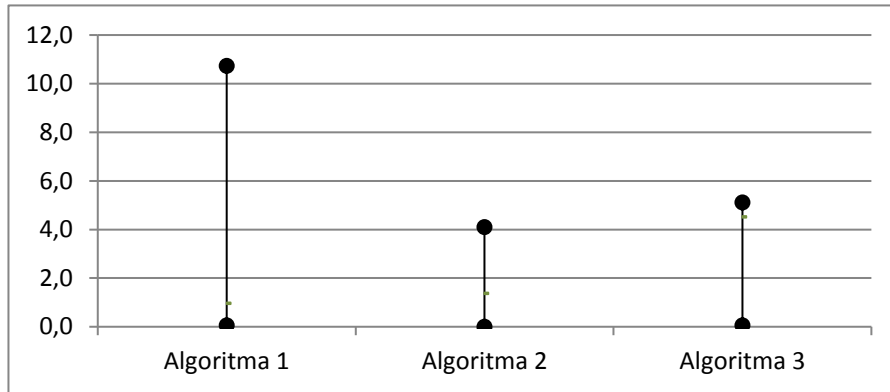
Örnek büyüklüğü (R) 20 ve güvenlik aralığı $(1-\alpha)$ %95 olmak üzere Çizelge 4.12'deki istatistiksel sonuçlar elde edilmiştir.

Çizelge 4.12. İstatiksel Sonuçlar (Dakika) (Sabit Sipariş)

Yöntem	Ortalama	Std Sapma	Güvenirlilik Sınırı	Alt Sınır	Üst Sınır
Algoritma 1	2,6	2,7	1,16	1,39	3,72
Algoritma 2	0,9	1,1	0,48	0,43	1,39
Algoritma 3	1,8	1,5	0,65	1,17	2,46
FCFS	285,5	130,1	57,01	228,48	342,51
STPT	225,2	78,1	34,24	191,00	259,49
SBPT	172,2	72,3	31,67	140,52	203,87
Due Date	166,9	79,8	34,96	131,89	201,82

İstatiksel sonuçlar incelendiğinde Algoritma 1, 2 ve 3'ün diğer sıralama yöntemlerinden oldukça farklı biçimde daha az durma süresi oluşturduğu görülmektedir. Algoritma 1, 2 ve 3'ü kendi aralarındaki farklılıklarını ortaya koymak için Şekil 4.4 Alt - Üst Sınır Diyagramı oluşturulmuştur.

Alt – Üst Sınır (Dakika)



Şekil 4.4. Algoritma 1, 2 ve 3 Alt - Üst Sınır Diyagramı (Sabit Sipariş)

Çizelge 4.12 ve Şekil 4.4 incelendiğinde Algoritma 2'nin diğer iki algoritmaya ortalama, sapma, alt ve üst sınır değerlerinde üstünlük kurduğu görülmektedir. Bu sonuç değişken siparişlerle yapılan benzetimlerin sonuçlarıyla tutarlılık göstermektedir. Her iki test sonucunda Algoritma 2'nin diğer sıralama yöntemlerine üstünlük kurduğu ifade edilebilir.

5. TARTIŞMA VE SONUÇ

Klasik anlamda FCFS temel prensibine göre kanban kartlarıyla yönetilen bir çekme sistemi uygulayarak başarılı üretim sıralama sonuçları elde etmek ancak süreçleri ve kapasite kullanımı dengeli, talep dalgalanması oldukça düşük üretim ortamlarında mümkündür. Tez konusu olarak ele alınan üretim ortamı stokastik talep kısıtına sahip, kapasite kullanımı vardiyalar arasında değişken ve akış hızları farklı süreçlerin oluşturduğu bir yapıdadır. Bu şartlar altında klasik bir çekme sisteminin üretimi müşteri süreçlerde duruş yaşanmadan yönlendirmesi ancak yüksek stok ortamında mümkündür. Yüksek stok günümüzde üretim sistemlerinin önlemeye çalıştığı bir maliyet unsurudur.

Tez kapsamında üretilen üç algoritma benzetim ortamında birbirleriyle ve kabul görmüş sıralama yaklaşımları olan FCFS, STPT, DD ve SBPT sıralama yaklaşımlarıyla kıyaslanmak üzere test edilmiştir.

Yeni oluşturulan üç algoritma birbirinin türevi algoritmalarıdır. Her üç algortimada kabul görmüş diğer sıralama yaklaşımlarına üstünlük sağlamıştır. Üç algortimada temel karşılaştırma kriteri olarak ilk operasyona en geç başlama zamanını (l_j) kullanır. l_j parametresine ek olarak stok ve/veya sipariş durumunu kontrol etmesine göre algoritmalar çeşitlendirilmiştir. Buna göre,

- Algoritma 1 : Üretim yapılan haftada ilgili yarı mamul için başka sipariş kalıp kalmadığını kontrol eder.
- Algoritma 2 : Üretim yapılan haftada ilgili yarı mamul için başka sipariş kalıp kalmadığını ve stok seviyesinin sıfır düşeyine düşüp düşmediğini kontrol eder.
- Algortima 3 : İlgili yarı mamul için stok seviyesinin sıfır düzeyine düşüp düşmediğini kontrol eder.

Sınır katmanlarının belirlenmesi Bölüm 3.4.3’de belirttiği gibi en büyük öncelik sıfır stok durumuna düşmeye verilmiş, en küçük öncelik ise kalan siparişin olmadığı duruma verilmiştir. İlk operasyona en geç başlama zamanının (l_j) mevcut andan önce bir zamanda yer alması bir geç kalmışlık durumu olarak değerlendirilmiş ve bu durumdaki malzemeler henüz geç kalmamış malzemelere göre öncelik kazanmıştır. Aynı sınır katmanı

dahilindeki malzemeler ise darboğaz süreç işlem süresi küçük olana öncelik tanınacak şekilde (SBPT) sıralanmıştır. Ancak tüm kriterlerde aynı ise ilk gelen ilk işlenir (FCFS) kuralı işletilmiştir.

Sıralama algoritması oluşturulurken kullanılan ilkeler:

- Üretim süreci sürekli bir süreçtir. Eğer bir malzemenin (yarı mamul) stoğu kalmadı ise en öncelikle üret.
- Siparişi olmayan ama en az bir kasa stoğu olan malzemenin üretimini erte.
- İşlem, setup ve kuyruk süreleri göz önüne alındığında geciken işe öncelik ver.
- İşlem süresi küçük olan iş önce işleme alınırsa kuyruk uzunluğu azalır. SPT'nin kuyruk azaltan yapısını kullanmak için aynı sınır katmanında yer alan malzemeler için darboğaz sürecin işlem süresi küçük olana öncelik ver.
- Tüm kriterleri aynı olan malzemelerde ilk geleni işleme al.

Bu sıralama şeklini kullanan Algoritma 2 diğer bütün algoritmalara üstünlük sağlamıştır. Teze konu olan üretim sisteminde kartlı kanban kullanılması planlanmıştır. Kartlı kanban kullanıldığında kart adetlerinin geçerli olduğu dönem içerisinde her bir malzemenin (referansın/yarı mamülün) en büyük stok değeri sınırlanmış olur. Üretim sistemi boşalan tüm kasalara ait kartları üretmek üzere sıralamasını günceller. Nihai amacı tüm kartların üretimini tamamlayarak tüm kanban kasalarını dolu hale getirmektir.

Alternatif bir yapı olarak sabit kanban kartlarının kullanılmadığı, her üretim yapıldığında bilgisayar çıktısı olarak alınacak malzeme tanıtım kartlarının kanban kartı yerine bildirimler için kullanıldığı bir sistem düşünülebilir.

$$K_j = \left\lceil \frac{(\sum_{i=1}^n (W_{ij} + P_{ij} + S_{ij})) (1 + G_j) \min(D_j, V_j)}{Q_j} \right\rceil \quad (3.19)$$

(3.19) kanban kart hesabı denkleminde bakıldığında haftalık talep miktarı (D_j)'nin kart sayısı için bir üst limit olduğu görülmektedir. Uzun bir dönem için sabit bir hesap yapılıyorsa ilgili dönem için en yüksek talep noktası baz alınarak stok seviyesi belirlemek gerekir. Stok seviyesinin sabit kart adedi ile sınırlanma zorunluluğu olmadığı durumda D_j 'nin en büyük değerini kullanma, $\max\{D_{jh}\}$, zorunluluğu ortadan kalkacaktır. Bu durumda D_{jh} 'nin artması veya azalması ilgili referansın en yüksek stok seviyesini etkileyecektir.

Bekleme süreleri (W_{ij}) kart sayısını etkileyen bir diğer etmendir. Uzun bir dönem için sabit bir hesap yapılıyorsa bekleme sürelerinin beklenen en büyük değerlerine göre hareket etmek, W_D , ve kanban adetlerini bu değere göre belirlemek gerekecektir. Teze konu olan üretim sıralama yaklaşımı bilgisayar ortamında yürütüldüğünde ilgili üretim sistemindeki tüm giriş ve çıkışların kayıtları alınabilecektir. (3.22) ve (3.24) bekleme süresi denklemleri hatırlandığında gerçekleşen bekleme süreleri anlık ve dönemlik olarak hesaplanabilecektir.

$$l_j(\alpha) = t + \tau_j^{WIP}(\alpha) - \tau_j^{min} - \sum_{vi} (W_{2j} + S_{ij} + P_{ij}) \quad (3.16)$$

(3.16) denkleminde görüldüğü gibi W_{2j} 'nin azalması en geç başlama zamanı $l_j(\alpha)$ değerini artıracaktır. Öncelik değeri hesaplandığında yükselen $l_j(\alpha)$ değeri geç kalan işler katmanı olan OS_3 'deki referans sayısını azaltacak, daha çok referans SPT sıralamasına göre üretime alınmaya başlayacaktır. SPT'ye göre üretim yapılması kuyruk uzunluğunun azalması demektir. Kuyruk uzunlukları azalan üretim sistemlerinin yüksek stok ihtiyacı da azalır.

Özetle, sistemde sabit sayıda kanban kartı olmayacağından referanslara ait en büyük stok seviyeleri değiştirilebilir olacaktır. Kanban kart hesabı ile belirlenen kart adedi başlangıç stok seviyesidir ve aynı zamanda sistemde bulunması gereken en yüksek stoktur. Bilgisayar ile yürütülen üretim sıralama sisteminde oluşan verilerin değerlendirilmesiyle elde edilen yeni bilgilerle stok seviyesi kendini sürekli güncelleyecek, malzemelerin en büyük stok seviyeleri malzemeye olan ihtiyaç ile artacak veya azalacaktır.

Oluşan esnek stok seviyelerinin ihtiyacın arttığı malzemelerin stoğunu yükselterek üretim duruş riskini daha da azaltacağı ve ihtiyacı azalan malzemelerin stoğunu azaltarak ortalama stok düzeyini düşüreceği iddia edilebilir. Bu iddia tezin bir sonucu ve ileri bir inceleme konusu olarak ortaya konulmuştur.

KAYNAKLAR

- Banks, J., Carson, J., Nelson, B.L. 1995** . Discrete-Event System Simulation, 2nd Ed. Prentice Hall International, London, UK, 1996, 548 pp.
- Berkley, B.J. 1993.** Simulation Test of FCFS and SPT Sequencing in Kanban System. *Desicion Science*. 24 : 218-227.
- Filho, M.G., Uzsoy, R. 2011.** The effect of shop floor continuous improvement programs on the lot size–cycle time relationship in a multi-product single-machine environment. *Int J Adv Manuf Technol*, 52 : 669–681.
- Koulamas, C. 2010.** A faster algorithm for a due date assignment problem with tardy jobs. *Operation Research Letters*, 38 : 127-128.
- Lee, L.C. 1987.** Parametric appraisal of the JIT system. *Int. J. Prod. Res.* 25(10) : 1415-1429.
- Lin, S.C., Goodman, E.D., Punch, W.F., 1997.** A Genetic Algorithm Approach to Dynamic Job Shop Scheduling Problems. Genetic Algorithms Research and Applications Group. 230 Engineering Building Michigan State University East Lansing, MI 48824.
- Pinedo, M.L. 2008.** Scheduling Theory, Algorithms, and Systems. Springer Science+Business Media, LLC., NewYork, NY, USA, 671 pp.
- Promodel 2011 Help System.** <http://www.promodel.com/onlinehelp>. (Erişim Tarihi:16.10.2015).
- Raman, N., Rachamadugu, R.V., Talbot, F.B. 1989.** Real-time scheduling of an automated manufacturing center. *European Journal of Operational Research*, 40: 222 - 242.
- Xiao, W., Hao, P., Zhang, S., Xu, X. 2000.** Hybrid Flow Job Scheduling Using Genetic Algorithms. 3. World Congress on Inteligent Control and Automotion, June 28 – July 2, 2000, Hefei, P.R. China.
- Wu, Z., Weng, M.X. 2005.** Multiagent Scheduling Method With Earliness and Tardiness Objectives in Flexible Job Shops. *IEEE transactions on systems, Man, and Cybernetics—Part B: Cybernetics*, 35(2) : 293-301.

EKLER

EK 1	FCFS Benzetim Modeli Promodel Text Metni
EK 2	Algoritma 1 Benzetim Modeli Promodel Text Metni
EK 3	Test 1 Benzetim Kart Varıřları Veri Özeti
EK 4	Test 2 Benzetim Kart Varıřları Veri Özeti
EK 5	Test 3 Benzetim Kart Varıřları Veri Özeti
EK 6	Test 4 Benzetim Kart Varıřları Veri Özeti
EK 7	Test 5 Benzetim Kart Varıřları Veri Özeti
EK 8	Test 6 Benzetim Kart Varıřları Veri Özeti
EK 9	Test 7 Benzetim Kart Varıřları Veri Özeti
EK 10	Test 8 Benzetim Kart Varıřları Veri Özeti
EK 11	Test 9 Benzetim Kart Varıřları Veri Özeti
EK 12	Test 10 Benzetim Kart Varıřları Veri Özeti

EK 1. FCFS Benzetim Modeli Promodel Text Metni

Locations ve Entities

```
*****
*
*                               Formatted Listing of Model:
*                               C:\E\PROM\Tez Modeller\FCFS.MOD
*
*****
```

```
Time Units:           Minutes
Distance Units:      Meters
```

```
*****
*                               Locations
*
*****
```

Name	Cap	Units	Stats	Rules	Cost
KesmeMak	1	5	Time Series	Min(Once), , First	
KesmeMak.1	1	1	Time Series	Min(Once), ,	
KesmeMak.2	1	1	Time Series	Min(Once), ,	
KesmeMak.3	1	1	Time Series	Min(Once), ,	
KesmeMak.4	1	1	Time Series	Min(Once), ,	
KesmeMak.5	1	1	Time Series	Min(Once), ,	
KesKuy	INFINITE	1	Time Series	oldest, Min(Once),	
CapMak	1	4	Time Series	oldest, , First	
CapMak.1	1	1	Time Series	oldest, ,	
CapMak.2	1	1	Time Series	oldest, ,	
CapMak.3	1	1	Time Series	oldest, ,	
CapMak.4	1	1	Time Series	oldest, ,	
CapKuy	INFINITE	1	Time Series	oldest, FIFO,	
Raf	1000	1	Time Series	oldest, ,	
Konduit_Balya	INF	1	Time Series	oldest, FIFO,	

Path Networks ve Interfaces

```
*****
*                               Entities
*
*****
```

Name	Speed (mpm)	Stats	Cost
Kart	1	Time Series	
Konduit	1	Time Series	

```
*****
*                               Path Networks
*
*****
```

Name	Type	T/S	From	To	BI	Dist/Time	Spe
Net_Operator	Passing	Time	N1	N2	Bi	0	1
Net_TPalet1	Passing	Speed & Distance	N1	N2	Bi	0	1
Net_Forklift	Passing	Speed & Distance	N1	N2	Bi	0	1
			N3	N1	Bi	0	1
Net_TPalet2	Passing	Speed & Distance	N1	N2	Bi	0	1

Resources

```
*****
*
*                               Resources
*
*****
```

Name	Units	Stats	Res Search	Ent Search	Path	Motion	Cost
Operator	1	By Unit	Closest	Min	Net_Operator Home: N1	Empty: 50 mpm Full: 50 mpm	
Forklift	1	By Unit	Closest	oldest	Net_Forklift Home: N1 (Return)	Empty: 50 mpm Full: 50 mpm	
TPalet1	1	By Unit	Closest	oldest	Net_TPalet1 Home: N1 (Return)	Empty: 50 mpm Full: 50 mpm	
TPalet2	1	By Unit	Closest	oldest	Net_TPalet2 Home: N1 (Return)	Empty: 50 mpm Full: 50 mpm	

Processing

```
*****
*
*                               Processing
*
*****
```

Entity	Location	Process		Routing			Move Logic
		Operation	Blk	Output	Destination	Rule	
Konduit Kart	Konduit_Balya KesKuy		1	Konduit	KesmeMak	JOIN 1	

```
//Senaryo dolayisiyla olusan stoktakart kolon sirasini hesapla
Sira=16*(LotNo-1)+4*(Set-1)+Kes_Oran+1
```

```
CikisZmn=ROUND(CLOCK(MIN))
```

```
DEC Stoktakart[Ref,Sira]
INC Kuyruktakart[Ref]
```

```
ORDER 1 Konduit TO Konduit_Balya
```

```
//Kesme Zamanini Kes_Oran düzeyinde arttir.
OKes_Zmn[Tracking_ID]=Round(Kes_Zmn[ref,LotNo]/(1+Kes_Oran/10))
```

```
IF Stoktakart[Ref,Sira]< 0 THEN Stokkart=0
IF Stoktakart[Ref,Sira]>= 0 THEN Stokkart=Stoktakart[Ref,Sira]
```

```
//FIFO için öncelik cikis Zamani olarak atanir
Once=CikisZmn
```

```
//Tracking_ID Report için datalar
OZ_1[Tracking_ID]=CikisZmn
Stkart_1[Tracking_ID]=Stoktakart[Ref,Sira]
Kuykart_1[Tracking_ID]=Kuyruktakart[Ref]
Islkart_1[Tracking_ID]=Islemdekart[Ref]
Oncelik[Tracking_ID]=Once
```

```
//Tracking_ID Report için datalar bitti
```

```
1 Kart KesmeMak FIRST 1 MOVE WITH Operator THEN FREE
```

Processing (Devam)

Kart KesmeMak GRAPHIC 2

```
//Kesmeye alma setup
USE 1 operator FOR Setup[1,Set] MIN

KesGirZmn= ROUND(CLOCK(MIN))
INC IslemdeKart[Ref]
DEC Kuyruktakart[Ref]

//Tracking_ID Report için datalar
OZ_2[Tracking_ID]=KesGirZmn
StKart_2[Tracking_ID]=Stoktakart[Ref,Sira]
Kuykart_2[Tracking_ID]=Kuyruktakart[Ref]
Islkart_2[Tracking_ID]=IslemdeKart[Ref]
KesBek[Tracking_ID]=OZ_2[Tracking_ID]-OZ_1[Tracking_ID]-Setup[1,Set]
//Tracking_ID Report için datalar bitti

JOIN 1 Konduit
WAIT OKes_Zmn[Tracking_ID] MIN
USE 1 operator FOR Setup[2,Set] MIN

IF Cap_Zmn[ref,LotNo]>0 THEN ROUTE 1
IF Cap_Zmn[ref,LotNo]=0 THEN ROUTE 2
```

Kart CapKuy GRAPHIC 3

```
1 Kart CapKuy FIRST 1 MOVE WITH TPalet1 THEN FREE
2 Kart Raf FIRST 1 MOVE WITH Forklift THEN FREE
```

Kart CapMak GRAPHIC 4

```
1 Kart CapMak FIRST 1 MOVE WITH TPalet2 THEN FREE
```

```
CapGirZmn= ROUND(CLOCK(MIN))
```

```
WAIT Setup[3,Set] MIN
WAIT Cap_Zmn[ref,LotNo] MIN
```

```
1 Kart Raf FIRST 1 MOVE WITH Forklift THEN FREE
```

Processing (Devam)

Kart Raf

```
StokGirZmn=round(CLOCK(MIN))
INC Stoktakart[ref,sira]
DEC Islemdekart[Ref]

//Tracking_ID Report için datalar
OZ_3[Tracking_ID]=StokGirZmn
StKart_3[Tracking_ID]=Stoktakart[Ref,sira]
KuyKart_3[Tracking_ID]=Kuyruktakart[Ref]
IslKart_3[Tracking_ID]=Islemdekart[Ref]

IF Cap_Zmn[ref,lotNo]=0 THEN
CapBek[Tracking_ID]=OZ_3[Tracking_ID]- OZ_2[Tracking_ID]- Cap_Zmn[ref,lotNo]
- OKes_Zmn[Tracking_ID]-Setup[2,Set]

IF Cap_Zmn[ref,lotNo]>0 THEN
CapBek[Tracking_ID]=OZ_3[Tracking_ID]- OZ_2[Tracking_ID]- Cap_Zmn[ref,lotNo]
- OKes_Zmn[Tracking_ID]-Setup[2,Set]-Setup[3,Set]
//Tracking_ID Report için datalar bitti

//Tracking_ID Report olustur
Senaryo=lotNo*100+Set*10+Kes_Oran
WRITE ID_Rapor, Senaryo
WRITE ID_Rapor, Tracking_ID
WRITE ID_Rapor, Ref
WRITE ID_Rapor, OZ_1[Tracking_ID]
WRITE ID_Rapor, StKart_1[Tracking_ID]
WRITE ID_Rapor, KuyKart_1[Tracking_ID]
WRITE ID_Rapor, IslKart_1[Tracking_ID]
WRITE ID_Rapor, Baslaengec[Tracking_ID]
WRITE ID_Rapor, Oncelik[Tracking_ID]
WRITE ID_Rapor, OZ_2[Tracking_ID]
WRITE ID_Rapor, StKart_2[Tracking_ID]
WRITE ID_Rapor, KuyKart_2[Tracking_ID]
WRITE ID_Rapor, IslKart_2[Tracking_ID]
WRITE ID_Rapor, KesBek[Tracking_ID]
WRITE ID_Rapor, OZ_3[Tracking_ID]
WRITE ID_Rapor, StKart_3[Tracking_ID]
WRITE ID_Rapor, KuyKart_3[Tracking_ID]
WRITE ID_Rapor, IslKart_3[Tracking_ID]
WRITE ID_Rapor, OKes_Zmn[Tracking_ID]
WRITE ID_Rapor, Cap_Zmn[ref,lotNo]
WRITELINE ID_Rapor, CapBek[Tracking_ID]
1 Kart EXIT FIRST 1
```

Arrivals

```
*****
* Arrivals *
*****

Entity Location Qty Each First Time Occurrences Frequency Logic
-----
Konduit Konduit_Balya 1 1
```

Attributes ve Variables

```
*****
*                               Attributes
*****
```

ID	Type	Classification
Ref	Integer	Entity
Once	Integer	Entity
Tracking_ID	Integer	Entity
Kalan_sip	Integer	Entity

```
*****
*                               Variables (global)
*****
```

ID	Type	Initial value	Stats
CaportBek1	Integer	0	Time Series
CikisZmn	Integer	0	Time Series
KesGirZmn	Integer	0	Time Series
CapGirZmn	Integer	0	Time Series
StokGirZmn	Integer	0	Time Series
TStok	Integer	0	Time Series
Senaryo	Integer	0	Time Series
Sira	Integer	0	Time Series
Stokkart	Integer	0	Time Series

Arrays

```
*****
*                               Arrays
*****
```

ID	Dimensions	Type	Import File	Export File
Stoktakart	100,48	Integer		
Kuyruktakart	100	Integer		
Islemdekart	100	Integer		
StKart_1	7000	Integer		
StKart_2	7000	Integer		
StKart_3	7000	Integer		
KuyKart_1	7000	Integer		
KuyKart_2	7000	Integer		
KuyKart_3	7000	Integer		
IslKart_1	7000	Integer		
IslKart_2	7000	Integer		
IslKart_3	7000	Integer		
KesBek	7000	Integer		
CapBek	7000	Integer		
StBek	7000	Integer		
OZ_1	7000	Integer		
OZ_2	7000	Integer		
OZ_3	7000	Integer		
Oncelik	7000	Integer		
KartTukZmn	100,3	Integer		PROM_Dosyalar\Arrays.xls
Baslaengec	7000	Integer		
Kes_Zmn	100,3	Integer		PROM_Dosyalar\Arrays.xls
Cap_Zmn	100,3	Integer		PROM_Dosyalar\Arrays.xls
MinStokZmn	100	Integer		PROM_Dosyalar\Arrays.xls
Setup	3,10	Integer		PROM_Dosyalar\Arrays.xls
OKes_Zmn	7000	Integer		
DS	7000	Integer		

Macros, Subroutines ve External Files

```
*****
*                                     MACROS                                     *
*****

ID          Text
-----
LotNo       1
Set         1
Kes_Oran    0

*****
*                                     Subroutines                             *
*****

ID          Type      Parameter  Type      Logic
-----
Sub1        None

*****
*                                     External Files                             *
*****

ID          Type      File Name      Prompt
-----
Kart_Gelis Arrival
ID_Rapor   General write  PROM_Dosyalar\ID_Rapor.txt
(null)     PROM_Dosyalar\Arrays.xls
```

EK 2. ALGORİTMA 1 Benzetim Modeli Promodel Text Metni

Locations

```
*****
*
*                               Formatted Listing of Model:
*                               C:\E\Prom\Modeller\Alg1.mod
*
*****
```

```
Time Units:           Minutes
Distance Units:      Meters
```

```
*****
*                               Locations
*
*****
```

Name	Cap	Units	Stats	Rules	Cost
KesmeMak	1	5	Time Series	Oldest, , First	
KesmeMak.1	1	1	Time Series	Oldest, ,	
KesmeMak.2	1	1	Time Series	Oldest, ,	
KesmeMak.3	1	1	Time Series	Oldest, ,	
KesmeMak.4	1	1	Time Series	Oldest, ,	
KesmeMak.5	1	1	Time Series	Oldest, ,	
KesKuy	INFINITE	1	Time Series	Oldest, FIFO,	
CapMak	1	4	Time Series	Oldest, FIFO, First	
CapMak.1	1	1	Time Series	Oldest, FIFO,	
CapMak.2	1	1	Time Series	Oldest, FIFO,	
CapMak.3	1	1	Time Series	Oldest, FIFO,	
CapMak.4	1	1	Time Series	Oldest, FIFO,	
CapKuy	INFINITE	1	Time Series	Oldest, FIFO,	
Raf	1000	1	Time Series	Oldest, ,	
Konduit_Balya	INF	1	Time Series	Oldest, FIFO,	
Bilgisayar	1	1	Time Series	Oldest, ,	

Entities ve Path Networks

```
*****
*                               Entities
*
*****
```

Name	Speed (mpm)	Stats	Cost
Kart	1	Time Series	
Konduit	1	Time Series	
Kartislemde	1	Time Series	

```
*****
*                               Path Networks
*
*****
```

Name	Type	T/S	From	To	BI	Dist/Time	Sp
Net_Operator	Passing	Speed & Distance	N1	N2	Bi	0	1
Net_TPalet1	Passing	Speed & Distance	N1	N2	Bi	0	1
Net_Forklift	Passing	Speed & Distance	N1	N2	Bi	0	1
			N3	N1	Bi	0	1
Net_TPalet2	Passing	Speed & Distance	N1	N2	Bi	0	1

Interfaces ve Resources

```
*****
*                                     Interfaces                                     *
*****
```

Net	Node	Location
Net_Operator	N1	Bilgisayar
	N2	KesmeMak
Net_TPalet1	N1	CapKuy
	N2	KesmeMak
Net_Forklift	N1	Raf
	N2	CapMak
	N3	KesmeMak
Net_TPalet2	N1	CapKuy
	N2	CapMak

```
*****
*                                     Resources                                     *
*****
```

Name	Units	Stats	Res Search	Ent Search	Path	Motion	Cost
Operator	1	By Unit	Closest	oldest	Net_Operator Home: N1	Empty: 50 mpm Full: 50 mpm	
Forklift	1	By Unit	Closest	oldest	Net_Forklift Home: N1 (Return)	Empty: 50 mpm Full: 50 mpm	
TPalet1	1	By Unit	Closest	oldest	Net_TPalet1 Home: N1 (Return)	Empty: 50 mpm Full: 50 mpm	
TPalet2	1	By Unit	Closest	oldest	Net_TPalet2 Home: N1 (Return)	Empty: 50 mpm Full: 50 mpm	

Processing

```
*****
*                                     Processing                                     *
*****
```

Entity	Location	Process		Routing			
		Operation	Blk	output	Destination	Rule	Move Logic
Konduit Kart	Konduit_Balya Keskuy		1	Konduit	KesmeMak	JOIN 1	
<pre>//Senaryo dolayisiyla olusan Stoktakart kolon sirasini hesapla Sira=16*(LotNo-1)+4*(Set-1)+Kes_Oran+1 CikisZmn=ROUND(CLOCK(MIN)) IF StoktaKart[Ref,Sira]>0 THEN sonCikisZmn[Ref]=CikisZmn DEC StoktaKart[Ref,Sira] INC Kuyruktakart[Ref] KalanSip[Ref]=Kalan_Sip //DS: Donem sonu hesabi IF CikisZmn<=HafZmn[1] THEN DonemSonuna=240 IF(CikisZmn<=HafZmn[2]) AND (CikisZmn>HafZmn[1]) THEN DonemSonuna=220 IF(CikisZmn<=HafZmn[3]) AND (CikisZmn>HafZmn[2]) THEN DonemSonuna=200 IF(CikisZmn<=HafZmn[4]) AND (CikisZmn>HafZmn[3]) THEN DonemSonuna=180 IF(CikisZmn<=HafZmn[5]) AND (CikisZmn>HafZmn[4]) THEN DonemSonuna=160 IF(CikisZmn<=HafZmn[6]) AND (CikisZmn>HafZmn[5]) THEN DonemSonuna=140 IF(CikisZmn<=HafZmn[7]) AND (CikisZmn>HafZmn[6]) THEN DonemSonuna=120 IF(CikisZmn<=HafZmn[8]) AND (CikisZmn>HafZmn[7]) THEN DonemSonuna=100</pre>							

Processing (Devam)

```
ORDER 1 Konduit TO Konduit_Balya

//Kesme Zamanini Kes_Oran düzeyinde arttir.
OKes_Zmn[Ref]=Round(Kes_Zmn[ref, LotNo]/(1+Kes_Oran/10))

CREATE 1 AS Kartislemde

WRITE Stok_Rapor, KesmeyeA1
WRITE Stok_Rapor, CikisZmn
WRITE Stok_Rapor, Ref
WRITE Stok_Rapor, Stoktakart[Ref,Sira]
WRITE Stok_Rapor, Kuyruktakart[Ref]
WRITE Stok_Rapor, Islemdekart[Ref]
WRITELINE Stok_Rapor, "Cikis Ani"

kartislemde keskuy          1   Kart          EXIT          FIRST 1
kartislemde bilgisayar    1   Kartislemde Bilgisayar FIRST 1
kartislemde kesmemak      GRAPHIC 2 1   Kartislemde Kesmemak   FIRST 1

KesGirZmn= ROUND(CLOCK(MIN))

//Siralama Yenileme
WHILE RefNo<=64 DO
BEGIN

IF Kuyruktakart[RefNo]>0 THEN
{

IF Stoktakart[RefNo,Sira]< 0 THEN StokKart=0
IF Stoktakart[RefNo,Sira]>= 0 THEN StokKart=Stoktakart[RefNo,Sira]

YLT[RefNo]= (StokKart+Islemdekart[RefNo])*KartTukZmn[RefNo, LotNo]
BeklZmn[RefNo]= YLT[RefNo]-(OKes_Zmn[RefNo]+ Cap_Zmn[RefNo, LotNo]
+ CapOrtBekl+MinStokZmn[RefNo]+Setup[1, Set]+Setup[2, Set]+Setup[3, Set])

EkZmn=SonCikZmn[RefNo]+KartTukZmn[RefNo, LotNo]-KesGirZmn
IF EkZmn<0 THEN EkZmn=0
IF SonCikZmn[RefNo]=0 THEN EkZmn=0

BeklZmn[RefNo]=BeklZmn[RefNo]+EkZmn

//Kesme süresine göre öncelik indisi belirleme

//subroutine çalıştır
X=Kes_Zmn[RefNo, LotNo]
Kesim_Öncelik(X)

IF BeklZmn[RefNo]<=0 THEN KesOnce[RefNo]=400+Prio[X]
IF BeklZmn[RefNo]>0 THEN KesOnce[RefNo]=300+Prio[X]

IF Kalansip[RefNo]-StokKart-Islemdekart[RefNo]<= 0
THEN KesOnce[RefNo]=DonemSonuna+Prio[X]

IF KesOnce[RefNo]>KesOnce[KesRef] THEN KesRef=RefNo

IF KesOnce[RefNo]=KesOnce[KesRef] THEN
{
IF BeklZmn[RefNo]<BeklZmn[KesRef] THEN KesRef=RefNo
} //KesOnce[RefNo]=KesOnce[KesRef]
} //Kuyrukta[RefNo]>0
```

Processing (Devam)

```
WRITE Stok_Rapor, KesmeyeA1
WRITE Stok_Rapor, KesGirZmn
WRITE Stok_Rapor, RefNo
WRITE Stok_Rapor, Stoktakart[RefNo,Sira]
WRITE Stok_Rapor, Kuyruktakart[RefNo]
WRITE Stok_Rapor, Islemdekart[RefNo]
WRITE Stok_Rapor, BeklZmn[RefNo]
WRITE Stok_Rapor, KesBek1
WRITELINE Stok_Rapor, KesOnce[RefNo]

INC RefNo

END
INC KesmeyeAL

//Siralama Yenileme bitti

Ref=KesRef
INC Islemdekart[Ref]
DEC Kuyruktakart[Ref]

KesMakinasi=LOCATION()
KesMakkesimBitis[KesMakinasi]=KesGirZmn+OKes_Zmn[Ref]+Setup[1,Set]+Setup[2,Set]

KesOnce[Ref]=0
RefNo=1

//Kesmeye alma setup
USE 1 Operator FOR Setup[1,Set] MIN

KesOnce[Ref]=0
RefNo=1

//Kesmeye alma setup
USE 1 Operator FOR Setup[1,Set] MIN

JOIN 1 Konduit

WAIT OKes_Zmn[Ref] MIN
USE 1 Operator FOR Setup[2,Set] MIN

IF Cap_Zmn[ref,LotNo]>0 THEN ROUTE 1
IF Cap_Zmn[ref,LotNo]=0 THEN ROUTE 2

1      kartislemde CapKuy      FIRST 1  MOVE WITH TPalet1 THEN FREE
2      kartislemde Raf        FIRST 1  MOVE WITH Forklift THEN FREE

kartislemde CapKuy          GRAPHIC 3
```

```
1 Kartislemde CapMak FIRST 1 MOVE WITH TPalet2 THEN FREE
```

Kartislemde CapMak

GRAPHIC 4

```
CapGirZmn= ROUND(CLOCK(MIN))
CapMakinası=LOCATION()

WAIT Setup[3,Set] MIN
WAIT Cap_Zmn[ref,LotNo] MIN
```

Processing (Devam)

```
1 Kartislemde Raf FIRST 1 MOVE WITH Forklift THEN FREE
```

Kartislemde Raf

```
StokGirZmn=round(CLOCK(MIN))

INC StoktaKart[ref,sıra]
DEC IslemdeKart[Ref]

WRITE Stok_Rapor, KesmeyeA1
WRITE Stok_Rapor, StokGirZmn
WRITE Stok_Rapor, Ref
WRITE Stok_Rapor, StoktaKart[Ref,Sıra]
WRITE Stok_Rapor, KuyruktaKart[Ref]
WRITE Stok_Rapor, IslemdeKart[Ref]
WRITE Stok_Rapor, "Depolama Ani"
WRITELINE Stok_Rapor, CapMakinası
1 Kartislemde EXIT FIRST 1
```

Arrivals ve Attributes

```
*****
* Arrivals *
*****

Entity Location Qty Each First Time Occurrences Frequency Logic
-----
Konduit Konduit_Balya 1 1
```

```
*****
* Attributes *
*****

ID Type Classification
-----
Ref Integer Entity
Tracking_ID Integer Entity
Kalan_Sip Integer Entity
```

Variables

```
*****
*                                     variables (global)                               *
*****
```

ID	Type	Initial value	stats
CapOrtBek1	Integer	0	Time Series
CikisZmn	Integer	0	Time Series
KesGirZmn	Integer	0	Time Series
CapGirZmn	Integer	0	Time Series
StokGirZmn	Integer	0	Time Series
Senaryo	Integer	0	Time Series
Sira	Integer	0	Time Series
StokKart	Integer	0	Time Series
KesmeyeA1	Integer	0	Time Series
RefNo	Integer	1	Time Series
EkZmn	Integer	0	Time Series
KesBek1	Integer	0	Time Series
DonemSonuna	Integer	0	Time Series
X	Integer	1	Time Series
KesRef	Integer	1	Time Series
KesMakinasi	Integer	0	Time Series
CapMakinasi	Integer	0	Time Series
KesMakIndex	Integer	2	Time Series

Arrays

```
*****
*                                     Arrays                                           *
*****
```

ID	Dimensions	Type	Import File	Export File
Stoktakart	100,48	Integer		
Kuyruktakart	100	Integer		
Islendekart	100	Integer		
Stkart_1	7000	Integer		
Stkart_2	7000	Integer		
Stkart_3	7000	Integer		
Kuykart_1	7000	Integer		
Kuykart_2	7000	Integer		
Kuykart_3	7000	Integer		
Islkart_1	7000	Integer		
Islkart_2	7000	Integer		
Islkart_3	7000	Integer		
KesBek	7000	Integer		
CapBek	7000	Integer		
StBek	7000	Integer		
OZ_1	7000	Integer		
OZ_2	7000	Integer		
OZ_3	7000	Integer		
Oncelik	7000	Integer		
KartTukZmn	100,3	Integer	ProM_Dosyalar\Arrays.xls	
Baslaengec	7000	Integer		
Kes_Zmn	100,3	Integer	ProM_Dosyalar\Arrays.xls	
Cap_Zmn	100,3	Integer	ProM_Dosyalar\Arrays.xls	
MinStokZmn	100	Integer	ProM_Dosyalar\Arrays.xls	
Setup	3,10	Integer	ProM_Dosyalar\Arrays.xls	
OKes_Zmn	7000	Integer		
DS	7000	Integer		
Kesonce	7000	Integer		
YLT	7000	Integer		
Prio	999	Integer		
SonCikZmn	100	Integer		
Bek1Zmn	100	Integer		
Kalansip	100	Integer		
KesMakKesimBitis	10	Integer		
HafZmn	8	Integer	ProM_Dosyalar\Arrays.xls	

Macros

```

*****
*                                     MACROS                                     *
*****

```

ID	Text
LotNo	1
Set	1
Kes_Oran	0

Subroutines

```

*****
*                                     subroutines                               *
*****

```

ID	Type	Parameter	Type	Logic
Kesim_Oncelik	None	X	Integer	IF X=160 THEN Prio[X]=1 IF X=137 THEN Prio[X]=2 IF X=126 THEN Prio[X]=3 IF X=113 THEN Prio[X]=4 IF X=111 THEN Prio[X]=5 IF X=103 THEN Prio[X]=6 IF X=100 THEN Prio[X]=7 IF X=90 THEN Prio[X]=8 IF X=86 THEN Prio[X]=9 IF X=80 THEN Prio[X]=10 IF X=77 THEN Prio[X]=11 IF X=75 THEN Prio[X]=12 IF X=72 THEN Prio[X]=13 IF X=71 THEN Prio[X]=14 IF X=60 THEN Prio[X]=15 IF X=45 THEN Prio[X]=16 IF X=34 THEN Prio[X]=17 IF X=15 THEN Prio[X]=18 RETURN
Sub1	None			

External Files

```

*****
*                                     External Files                               *
*****

```

ID	Type	File Name	Prompt
Kart_Gelis	Arrival	Prom_Dosyalar\Kart_Arrivals.xls	
Stok_Rapor	General write	Prom_Dosyalar\Stok_Rapor.txt	
Kesme_Giris_Rapor	General write	Prom_Dosyalar\Kesme_Giris_Rapor.txt	
Siralama_Rapor	General write	Prom_Dosyalar\Siralama_Rapor.txt	
Capak_Giris_Rapor	General write	Prom_Dosyalar\Capak_Giris_Rapor.txt	
ID_Rapor	General write	Prom_Dosyalar\ID_Rapor.txt	
(null)		Prom_Dosyalar\Arrays.xls	

EK 3. Test 1 Benzetim Kart Varışları Veri Özeti

Test Verisinin Özeti

Test No	TMS (dk)	TKM (Adet)	$\sum K_j$ (Adet)	$\max\{t_j^1\}$ (dk)	KKO_1	G
1	477737	2674	133	63033	80,8%	0

Kart Varış Sırası

Entity	Loc	Qty	Time	Number	Freq	Ref	Tracking_ID
Kart	KesKuy	1	11	1		3	1
Kart	KesKuy	1	41	1		16	2
Kart	KesKuy	1	42	1		20	3
Kart	KesKuy	1	43	1		6	4
Kart	KesKuy	1	43	1		47	5
Kart	KesKuy	1	46	1		61	6
Kart	KesKuy	1	56	1		43	7
Kart	KesKuy	1	87	1		7	8
Kart	KesKuy	1	105	1		64	9
Kart	KesKuy	1	110	1		18	10
.....							
.....							
.....							
Kart	KesKuy	1	62673	1		59	2665
Kart	KesKuy	1	62712	1		51	2666
Kart	KesKuy	1	62724	1		20	2667
Kart	KesKuy	1	62839	1		38	2668
Kart	KesKuy	1	62877	1		42	2669
Kart	KesKuy	1	62891	1		51	2670
Kart	KesKuy	1	62895	1		13	2671
Kart	KesKuy	1	62939	1		7	2672
Kart	KesKuy	1	62996	1		50	2673
Kart	KesKuy	1	63033	1		48	2674

EK 4. Test 2 Benzetim Kart Varışları Veri Özeti

Test Verisinin Özeti

Test No	TMS (dk)	TKM (Adet)	$\sum K_j$ (Adet)	$\max\{t_j^1\}$ (dk)	KKO_1	G
2	492881	2830	152	63114	83,8%	0

Kart Varış Sırası

Entity	Loc	Qty	Time	Number	Freq	Ref	Tracking_ID
Kart	KesKuy	1	120	1		42	1
Kart	KesKuy	1	120	1		43	2
Kart	KesKuy	1	145	1		6	3
Kart	KesKuy	1	167	1		20	4
Kart	KesKuy	1	180	1		63	5
Kart	KesKuy	1	180	1		64	6
Kart	KesKuy	1	185	1		48	7
Kart	KesKuy	1	205	1		15	8
Kart	KesKuy	1	205	1		37	9
Kart	KesKuy	1	240	1		42	10
.....							
.....							
.....							
Kart	KesKuy	1	62532	1		46	2821
Kart	KesKuy	1	62553	1		51	2822
Kart	KesKuy	1	62646	1		42	2823
Kart	KesKuy	1	62646	1		43	2824
Kart	KesKuy	1	62647	1		19	2825
Kart	KesKuy	1	62664	1		51	2826
Kart	KesKuy	1	62775	1		51	2827
Kart	KesKuy	1	62894	1		51	2828
Kart	KesKuy	1	63004	1		51	2829
Kart	KesKuy	1	63114	1		51	2830

EK 5. Test 3 Benzetim Kart Varışları Veri Özeti

Test Verisinin Özeti

Test No	TMS (dk)	TKM (Adet)	$\sum K_j$ (Adet)	$\max\{t_j^1\}$ (dk)	KKO_1	G
3	508219	2959	152	62916	86,2%	0

Kart Varış Sırası

Entity	Loc	Qty	Time	Number	Freq	Ref	Tracking_ID
Kart	KesKuy	1	2	1		5	1
Kart	KesKuy	1	4	1		50	2
Kart	KesKuy	1	8	1		15	3
Kart	KesKuy	1	13	1		42	4
Kart	KesKuy	1	13	1		43	5
Kart	KesKuy	1	22	1		6	6
Kart	KesKuy	1	23	1		20	7
Kart	KesKuy	1	46	1		33	8
Kart	KesKuy	1	49	1		63	9
Kart	KesKuy	1	49	1		64	10
.....							
.....							
.....							
Kart	KesKuy	1	62782	1		28	2950
Kart	KesKuy	1	62787	1		50	2951
Kart	KesKuy	1	62797	1		42	2952
Kart	KesKuy	1	62797	1		43	2953
Kart	KesKuy	1	62803	1		10	2954
Kart	KesKuy	1	62803	1		11	2955
Kart	KesKuy	1	62808	1		25	2956
Kart	KesKuy	1	62808	1		26	2957
Kart	KesKuy	1	62916	1		42	2958
Kart	KesKuy	1	62916	1		43	2959

EK 6. Test 4 Benzetim Kart Varışları Veri Özeti

Test Verisinin Özeti

Test No	TMS (dk)	TKM (Adet)	$\sum K_j$ (Adet)	$\max\{t_j^1\}$ (dk)	KKO_1	G
4	502081	2896	152	61518	87,4%	0

Kart Varış Sırası

Entity	Loc	Qty	Time	Number	Freq	Ref	Tracking_ID
Kart	KesKuy	1	1	1		6	1
Kart	KesKuy	1	1	1		7	2
Kart	KesKuy	1	2	1		25	3
Kart	KesKuy	1	2	1		26	4
Kart	KesKuy	1	4	1		36	5
Kart	KesKuy	1	6	1		20	6
Kart	KesKuy	1	15	1		63	7
Kart	KesKuy	1	15	1		64	8
Kart	KesKuy	1	19	1		19	9
Kart	KesKuy	1	25	1		1	10
.....							
.....							
.....							
Kart	KesKuy	1	61174	1		63	2887
Kart	KesKuy	1	61174	1		64	2888
Kart	KesKuy	1	61200	1		58	2889
Kart	KesKuy	1	61235	1		42	2890
Kart	KesKuy	1	61235	1		43	2891
Kart	KesKuy	1	61245	1		20	2892
Kart	KesKuy	1	61314	1		13	2893
Kart	KesKuy	1	61471	1		39	2894
Kart	KesKuy	1	61492	1		37	2895
Kart	KesKuy	1	61518	1		36	2896

EK 7. Test 5 Benzetim Kart Varışları Veri Özeti

Test Verisinin Özeti

Test No	TMS (dk)	TKM (Adet)	$\sum K_j$ (Adet)	$\max\{t_j^1\}$ (dk)	KKO_1	G
5	525469	2872	147	63035	89,6%	0

Kart Varış Sırası

Entity	Loc	Qty	Time	Number	Freq	Ref	Tracking_ID
Kart	KesKuy	1	3	1		63	1
Kart	KesKuy	1	4	1		22	2
Kart	KesKuy	1	7	1		16	3
Kart	KesKuy	1	9	1		7	4
Kart	KesKuy	1	18	1		25	5
Kart	KesKuy	1	26	1		20	6
Kart	KesKuy	1	44	1		12	7
Kart	KesKuy	1	51	1		13	8
Kart	KesKuy	1	62	1		6	9
Kart	KesKuy	1	79	1		55	10
.....							
.....							
.....							
Kart	KesKuy	1	62556	1		63	2863
Kart	KesKuy	1	62562	1		33	2864
Kart	KesKuy	1	62621	1		42	2865
Kart	KesKuy	1	62692	1		20	2866
Kart	KesKuy	1	62746	1		30	2867
Kart	KesKuy	1	62747	1		33	2868
Kart	KesKuy	1	62859	1		20	2869
Kart	KesKuy	1	63026	1		20	2870
Kart	KesKuy	1	63027	1		50	2871
Kart	KesKuy	1	63035	1		48	2872

EK 8. Test 6 Benzetim Kart Varışları Veri Özeti

Test Verisinin Özeti

Test No	TMS (dk)	TKM (Adet)	$\sum K_j$ (Adet)	$\max\{t_j^1\}$ (dk)	KKO_1	G
6	525066	2999	168	61691	91,7%	0

Kart Varış Sırası

Entity	Loc	Qty	Time	Number	Freq	Ref	Tracking_ID
Kart	KesKuy	1	1	1		25	1
Kart	KesKuy	1	1	1		26	2
Kart	KesKuy	1	2	1		42	3
Kart	KesKuy	1	2	1		43	4
Kart	KesKuy	1	9	1		7	5
Kart	KesKuy	1	11	1		63	6
Kart	KesKuy	1	11	1		64	7
Kart	KesKuy	1	12	1		5	8
Kart	KesKuy	1	13	1		60	9
Kart	KesKuy	1	14	1		16	10
.....							
.....							
.....							
Kart	KesKuy	1	61165	1		11	2990
Kart	KesKuy	1	61177	1		50	2991
Kart	KesKuy	1	61203	1		59	2992
Kart	KesKuy	1	61286	1		13	2993
Kart	KesKuy	1	61493	1		42	2994
Kart	KesKuy	1	61493	1		43	2995
Kart	KesKuy	1	61506	1		58	2996
Kart	KesKuy	1	61612	1		42	2997
Kart	KesKuy	1	61612	1		43	2998
Kart	KesKuy	1	61691	1		29	2999

EK 9. Test 7 Benzetim Kart Varışları Veri Özeti

Test Verisinin Özeti

Test No	TMS (dk)	TKM (Adet)	$\sum K_j$ (Adet)	$\max\{t_j^1\}$ (dk)	KKO_1	G
7	544152	3117	181	62905	93,5%	0

Kart Varış Sırası

Entity	Loc	Qty	Time	Number	Freq	Ref	Tracking_ID
Kart	KesKuy	1	2	1		21	1
Kart	KesKuy	1	2	1		50	2
Kart	KesKuy	1	4	1		15	3
Kart	KesKuy	1	5	1		20	4
Kart	KesKuy	1	11	1		7	5
Kart	KesKuy	1	11	1		42	6
Kart	KesKuy	1	11	1		43	7
Kart	KesKuy	1	12	1		52	8
Kart	KesKuy	1	12	1		53	9
Kart	KesKuy	1	13	1		6	10
.....							
.....							
.....							
Kart	KesKuy	1	62739	1		61	3108
Kart	KesKuy	1	62783	1		25	3109
Kart	KesKuy	1	62783	1		26	3110
Kart	KesKuy	1	62801	1		51	3111
Kart	KesKuy	1	62805	1		42	3112
Kart	KesKuy	1	62805	1		43	3113
Kart	KesKuy	1	62814	1		50	3114
Kart	KesKuy	1	62857	1		60	3115
Kart	KesKuy	1	62905	1		25	3116
Kart	KesKuy	1	62905	1		26	3117

EK 10. Test 8 Benzetim Kart Varışları Veri Özeti

Test Verisinin Özeti

Test No	TMS (dk)	TKM (Adet)	$\sum K_j$ (Adet)	$\max\{t_j^1\}$ (dk)	KKO_1	G
8	553376	3257	189	63148	94,3%	0

Kart Varış Sırası

Entity	Loc	Qty	Time	Number	Freq	Ref	Tracking_ID
Kart	KesKuy	1	2	1		58	1
Kart	KesKuy	1	4	1		16	2
Kart	KesKuy	1	4	1		42	3
Kart	KesKuy	1	4	1		43	4
Kart	KesKuy	1	4	1		51	5
Kart	KesKuy	1	7	1		62	6
Kart	KesKuy	1	10	1		5	7
Kart	KesKuy	1	14	1		50	8
Kart	KesKuy	1	23	1		20	9
Kart	KesKuy	1	24	1		15	10
.....							
.....							
.....							
Kart	KesKuy	1	62797	1		26	3248
Kart	KesKuy	1	62831	1		42	3249
Kart	KesKuy	1	62831	1		43	3250
Kart	KesKuy	1	62892	1		48	3251
Kart	KesKuy	1	62914	1		36	3252
Kart	KesKuy	1	62919	1		25	3253
Kart	KesKuy	1	62919	1		26	3254
Kart	KesKuy	1	62950	1		42	3255
Kart	KesKuy	1	62950	1		43	3256
Kart	KesKuy	1	63148	1		47	3257

EK 11. Test 9 Benzetim Kart Varışları Veri Özeti

Test Verisinin Özeti

Test No	TMS (dk)	TKM (Adet)	$\sum K_j$ (Adet)	$\max\{t_j^1\}$ (dk)	KKO_1	G
9	554485	3220	240	61850	96,4%	0

Kart Varış Sırası

Entity	Loc	Qty	Time	Number	Freq	Ref	Tracking_ID
Kart	KesKuy	1	1	1		22	1
Kart	KesKuy	1	1	1		42	2
Kart	KesKuy	1	1	1		43	3
Kart	KesKuy	1	1	1		50	4
Kart	KesKuy	1	1	1		62	5
Kart	KesKuy	1	2	1		52	6
Kart	KesKuy	1	2	1		53	7
Kart	KesKuy	1	5	1		48	8
Kart	KesKuy	1	8	1		20	9
Kart	KesKuy	1	9	1		6	10
.....							
.....							
.....							
Kart	KesKuy	1	61522	1		33	3211
Kart	KesKuy	1	61612	1		42	3212
Kart	KesKuy	1	61612	1		43	3213
Kart	KesKuy	1	61673	1		48	3214
Kart	KesKuy	1	61679	1		60	3215
Kart	KesKuy	1	61680	1		59	3216
Kart	KesKuy	1	61731	1		42	3217
Kart	KesKuy	1	61731	1		43	3218
Kart	KesKuy	1	61850	1		42	3219
Kart	KesKuy	1	61850	1		43	3220

EK 12. Test 10 Benzetim Kart Varışları Veri Özeti

Test Verisinin Özeti

Test No	TMS (dk)	TKM (Adet)	$\sum K_j$ (Adet)	$\max\{t_j^1\}$ (dk)	KKO_1	G
10	565693	3230	243	61615	98,4%	0

Kart Varış Sırası

Entity	Loc	Qty	Time	Number	Freq	Ref	Tracking_ID
Kart	KesKuy	1	4	1		51	1
Kart	KesKuy	1	7	1		63	2
Kart	KesKuy	1	7	1		64	3
Kart	KesKuy	1	10	1		42	4
Kart	KesKuy	1	10	1		43	5
Kart	KesKuy	1	12	1		50	6
Kart	KesKuy	1	17	1		5	7
Kart	KesKuy	1	17	1		20	8
Kart	KesKuy	1	30	1		25	9
Kart	KesKuy	1	30	1		26	10
.....							
.....							
.....							
Kart	KesKuy	1	61345	1		43	3221
Kart	KesKuy	1	61390	1		36	3222
Kart	KesKuy	1	61464	1		42	3223
Kart	KesKuy	1	61464	1		43	3224
Kart	KesKuy	1	61516	1		34	3225
Kart	KesKuy	1	61582	1		7	3226
Kart	KesKuy	1	61583	1		42	3227
Kart	KesKuy	1	61583	1		43	3228
Kart	KesKuy	1	61615	1		27	3229
Kart	KesKuy	1	61615	1		28	3230

ÖZGEÇMİŞ

Adı Soyadı : Ü. Eraydın GENÇ

Doğum Yeri ve Tarihi : Ordu, 01.03.1981

Yabancı Dili : İngilizce

Eğitim Durumu

Lise : Ordu Fen Lisesi, 1995-1998

Lisans : Uludağ Üniversitesi Endüstri Mühendisliği 1999-2003

Yüksek Lisans : Uludağ Üniversitesi Endüstri Mühendisliği 2008-2019

Çalıştığı Kurumlar

Matay Otomotiv A.Ş. 2005-2007

Ficosa Otomotiv A.Ş. 2007-2014

Matay Otomotiv A.Ş. 2014-2018

Nobel Otomotiv A.Ş. 2018-....

İletişim : eraydin_genc@yahoo.com

BURSA ULUDAĞ ÜNİVERSİTESİ

TEZ ÇOĞALTMA VE ELEKTRONİK YAYIMLAMA İZİN FORMU

Yazar Adı Soyadı	Ü. Eraydın GENÇ
Tez Adı	ESNEK ATÖLYE TİPİ ÜRETİM ORTAMINDA STOKASTİK TALEP İÇİN DİNAMİK ÇİZELGELEME
Enstitü	Fen Bilimleri
Anabilim Dalı	Endüstri Mühendisliği
Tez Türü	Yüksek Lisans
Tez Danışman(lar)ı	Prof. Dr. Erdal EMEL
Çoğaltma (Fotokopi Çekim) İzni Kısıtlama	<input type="checkbox"/> Patent Kısıt (2 yıl) <input type="checkbox"/> Genel Kısıt (6 ay) <input checked="" type="checkbox"/> Tezimin elektronik ortamda yayımlanmasına izin veriyorum.

Hazırlamış olduğum tezimin belirttiğim hususlar dikkate alınarak, fikri mülkiyet haklarım saklı kalmak üzere Bursa Uludağ Üniversitesi Kütüphane ve Dokümantasyon Daire Başkanlığı tarafından hizmete sunulmasına izin verdiğimi beyan ederim.

Tarih : 31.10.2019

İmza :

