Bilkent Üniversitesi
Mühendislik Fakültesi
Endüstri Mühendisliği Bölümü

Üniversite-Sanayi İşbirliği Öğrenci Projeleri 2023

Derleyenler

Prof. Dr. Savaş Dayanık, Doç. Dr. A. Selin Kocaman,
Dr. Emre Uzun
Endüstri Mühendisliği Bölümü
Sistem Tasarımı Ders Koordinatörleri

Yeşim Gülseren
Üniversite-Sanayi İşbirliği Öğrenci Projeleri
Koordinatörü

ISBN: 978-605-9788-51-9
BASKI: Meteksan Matbaacılık, Haziran 2023
YAYINCI SERTİFİKA NO: 51344
MATBAA SERTİFİKA NO: 46519

Bandrol Uygulamasına İlişkin Usul ve Esaslar Hakkında Yönetmeliğin 5. maddesinin 2. fıkrası çerçevesinde bandrol taşıması zorunlu değildir.

According to the second paragraph of Article 5 of the Regulation on Banderole Application Procedures and Principles, the use of banderole for this publication is not compulsory.

Para ile satılamaz.
For free distribution only.

Bilkent University
Faculty of Engineering
Department of Industrial Engineering

University-Industry Collaboration Student Projects 2023

Editors

Prof. Dr. Savaş Dayanık, Assoc. Prof. A. Selin Kocaman,
Dr. Emre Uzun
Department of Industrial Engineering
Systems Design Course Coordinators

Yeşim Gülseren
University-Industry Collaboration Student Projects
Coordinator

Düzenleme Kurulu

Organizing Committee

Atahan Bayır
Yunus Emre Çakır
Prof. Dr. Savaş Dayanık
Aslı Eroğlu
Yeşim Gülseren
Doç. Dr. A. Selin Kocaman
Dr. Emre Uzun

Değerlendirme Kurulu

Review Committee

Bilkent Üniversitesi Endüstri Mühendisliği Öğretim Üyeleri

Bilkent University Industrial Engineering Faculty Members

Doç. Dr. Arnab Basu
Prof. Dr. Savaş Dayanık
Prof. Dr. Ülkü Gürler
Doç. Dr. Özlem Çavuş İyigün
Prof. Dr. Bahar Yetiş Kara
Prof. Dr. Oya Ekin Karaşan
Doç. Dr. Özlem Karsu
Dr. Öğr. Üyesi Taghi Khaniyev
Doç. Dr. Ayşe Selin Kocaman
Dr. Öğr. Üyesi Emre Nadar
Prof. Dr. Mustafa Çelebi Pınar
Dr. Öğr. Üyesi Firdevs Ulus
Dr. Emre Uzun

Bilkent Üniversitesi İşletme Fakültesi Öğretim Üyesi

Bilkent University Faculty of Business Administration Faculty Member

Doç. Dr. Emre Berk

Doç. Dr., *Assoc. Prof.*; Dr. Öğr. Üyesi, *Asst. Prof.*

Önsöz

Bu kitap, 2022-2023 öğretim yılında Bilkent Üniversitesi Endüstri Mühendisliği Bölümü tarafından gerçekleştirilen *Üniversite-Sanayi İşbirliği Bitirme Projeleri* özetlerini kapsamaktadır. Programımız 29 yıl önce sistem tasarımı derslerinin sanayi projelerine dönüştürülmesi ile başlamıştır. Bu süre içerisinde farklı sektör ve büyüklükte 122 iş, sanayi, ve kâr amacı gütmeyen kuruluşlarla toplam 534 proje gerçekleştirilmiştir.

Endüstri Mühendisliği Bölümü dördüncü sınıf öğrencilerinden oluşan proje ekipleri, akademik ve iş dünyasından danışmanların gözetiminde firmanın gündemine girmiş olan ve çözüm bekleyen gerçek problemlerini çözmektedirler. Yapılan projeler sonucunda ortaya çıkan ürün, yöntem veya hizmet, ilgili firmaya önemli yarar ve katma değer sağlamaktadır.

Endüstri Mühendisliği Proje Fuarı ve Yarışması, 2003 yılında yapılan projelerin ilgili tüm firma, kuruluş ve üniversitelerle paylaşılması, iş dünyasının seçkin kuruluşlarının birbirleriyle ve üniversite ile olan etkileşiminin artırılması ve öğrencilerimizin iş hayatına daha donanımlı hazırlanmasını sağlamak amacıyla başlatılmıştır. Her yıl sistematik ve etkin bir şekilde yapılan bu çalışmaların daha kalıcı ve yaygın olarak paylaşılması amacıyla da "Endüstri Projeleri" kitabı serisi hazırlanmış ve bu dönemde gerçekleştirilen projeler gizlilik ilkesine bağlı kalınarak özet halinde sizlere sunulmuştur.

Kitapta yer alan proje özetlerinin doğru ve okunaklı olması için desteklerini esirgemeyen *Değerlendirme Kurulu*'muza, fuar ve yarışma jürimizde görev alan Mustafa Bora Dilik (Nevzat Ecza), Erdinç Mert (BeNova Danışmanlık), Özgür Sarhan (Dünya Bankası), Dilek Şen (Prosis Danışmanlık) ve Prof. Dr. M. Selim Aktürk'e (Bilkent Üniversitesi) teşekkür ederiz.

Prof. Dr. Savaş Dayanık
Doç. Dr. A. Selin Kocaman
Dr. Emre Uzun

Bilkent Üniversitesi Endüstri Mühendisliği Bölümü
Sistem Tasarımı Dersi Koordinatörleri

Preface

This booklet contains 2022-2023 academic year *University-Industry Collaboration Student Project* summaries done by the senior students of the Industrial Engineering Department at Bilkent University in collaboration with industrial companies, businesses, and non-profit organizations. This program started when senior design courses were reorganized as industrial projects 29 years ago. Since then, 534 projects have been completed, with 122 companies operating in various sectors.

Senior student groups of the Industrial Engineering Department solve companies' real problems under the guidance of academic and industrial advisors. The project outcomes provide companies with many operational benefits and add value to their services and products.

Since 2003 *Industrial Engineering Project Fair and Competition* has been held to disseminate the project outcomes to firms and universities, boost the synergy, encourage collaboration between industry and university, and help senior students get better equipped before they take full industrial positions. Every year the project summaries are edited in a project booklet with care given not to disclose firm-specific sensitive information and shared with the community to spread the word and impact of projects.

We thank the *Review Committee* for their efforts that improved the correctness and readability of project summaries in the book. We also thank Mustafa Bora Dilik (Nevzat Pharmaceutical), Erdinç Mert (BeNova Consulting), Özgür Sarhan (The World Bank), Dilek Şen (Prosis Consultancy) and Prof. Dr. M. Selim Aktürk (Bilkent University) for serving on the project competition jury this year.

Prof. Dr. Savaş Dayanık
Assoc. Prof. A. Selin Kocaman
Dr. Emre Uzun

Bilkent University Industrial Engineering Department
Systems Design Course Coordinators

İçindekiler
Table of Contents

Bugüne kadar öğrenci projelerimize destek veren kuruluşlar

Companies participated in the student projects so far

Düzenleme kurulu, 2022-2023 programına değerli katkıları için aşağıda adı geçen Bilkent Üniversitesi mensuplarına teşekkür eder.

The organizing committee thanks Bilkent University members named below for their invaluable help to run 2022-2023 program.

Bilkent Üniversitesi Endüstri Mühendisliği Öğretim Üyeleri

Bilkent University Industrial Engineering Faculty Members

Doç. Dr. Arnab Basu
Prof. Dr. Savaş Dayanık
Prof. Dr. Ülkü Gürler
Doç. Dr. Özlem Çavuş İyigün
Prof. Dr. Bahar Yetiş Kara
Prof. Dr. Oya Ekin Karaşan
Doç. Dr. Özlem Karsu
Dr. Öğr. Üyesi Taghi Khaniyev
Doç. Dr. Ayşe Selin Kocaman
Dr. Öğr. Üyesi Emre Nadar
Prof. Dr. Mustafa Çelebi Pınar
Dr. Öğr. Üyesi Firdevs Ulus
Dr. Emre Uzun

Bilkent Üniversitesi İşletme Fakültesi Öğretim Üyesi

Bilkent University Faculty of Business Administration Faculty Member

Doç. Dr. Emre Berk

Üniversite Sanayi İşbirliği Öğrenci Projeleri Koordinatörü

University & Industry Cooperation Student Projects Coordinator

Yeşim Gülseren

Bilkent Üniversitesi Endüstri Mühendisliği Bölümü Sekreteri

Bilkent University Industrial Engineering Department Secretary

Vegün Ekmekçi

Düzenleme kurulu, 2022-2023 programına sağladıkları işbirliği için aşağıda yer alan iş dünyasının değerli mensuplarına teşekkür eder.

The organizing committee thanks the esteemed company representatives listed below for their cooperation to run 2022-2023 program.

A101
Dilara Fatma Erke

Arçelik Bulaşık İşletmesi
Aysen Gümüş
Muhammet Erol

Arçelik Buzdolabı İşletmesi
Cansu Demir
Özlem Deviren
Yaşar Semih Tolukan

Arçelik Elektronik İşletmesi
Hüseyin Artun
Burak Beşer
Tuğçe Pınar Öztürk
Orhan Sancaklı

Arçelik Global
Nihat Bayız
Cansu Musaoğlu
Evrim Özgül
Tuğba Uyar

Bakioğlu Holding
Sabahattin Bilgen
Beste Yıldız

Doğadan
Ferat Atabek
Yüce Cankur
Anıl Dertli
Tuğba Ekşi
Gözde Şili
Hayrullah Üstün

Demir Export
Özge Göksu Başer
Haydar Çınar
Pınar Tekin

Eti Gıda
Dr. Cemal Akyel
Ezgi Çetinel
Enes Talha Doğanlı
Sinem Eryılmaz
İlay Tümer
Zafer Ünlüer

Meteksan
Okan Akay
Eda Şenol

FNSS
Başak Demirkol
Gülçin Özden
Ayşegül Öztürk

ReklamUp
Ahmet Karaoğlan
Ayşegül Yıldırım

Norm Fasteners
Fatma Akdemir
Dila Nart

Oplog
Can Ayhan
Yaren Yılmaz

Oyak Renault
Eyüphan Altunbay
İlker Civa
Elif Çimen
Özgür Durdu
Elif Tayan
Turhan Yamaç

Solvoyo
Eylül Erkan
Emre Haliloglu
İshak Memigüven
Mehmet Buğra Unalan

Supply Chain Wizard
Yunus Emre Yurdagül

Tepe Betopan
Burak Bilgiç
Çağatay Çaparlı
Cavit Mocan

UNDP
Gökhan Dikmener

Unilever
Sumru Ergök Eker
Ömür Göçer

Bölüm Başkanı'ndan

Bilkent Üniversitesi Endüstri Mühendisliği Bölümü, öğrencilerinin teknolojik ve sosyal değişikliklere uyum sağlayan, yaşam boyu öğrenen ve sorgulayan iyi endüstri mühendisleri olarak mezun olmalarını amaçlamaktadır. Karmaşık sistemlere ve problemlere bütün olarak bakabilme ve analitik düşünebilme, eğitim programının önemli amaçlarındandır. Bölüm, 2007 yılında *Accreditation Board for Engineering and Technology (ABET)* adlı bağımsız kuruluş tarafından eğitim kalitesini belgeleyen tam akreditasyonu Türkiye'de ilk alan mühendislik bölümüdür.

Eğitimde dünya çapında kalite standartlarını kullanan Endüstri Mühendisliği Bölümü, ülkemizde örnek gösterilen *Üniversite-Sanayi İşbirliği Programı*'nı 29 yıldır başarıyla uygulamaktadır. Programın hedefi mezuniyet aşamasındaki öğrencilerin kapsamlı mesleki deneyim kazandırmaktır. Altı-yedi kişilik proje ekipleri, akademik ve endüstriyel danışmanların gözetiminde firmaların çözüm bekleyen gerçek problemlerini çözmektedirler.

Bu yıl, *21. Endüstri Mühendisliği Proje Fuarı ve Yarışması*'nda 21 proje bulunmaktadır. Fuarda öğrencilerimiz, yıl boyunca projeleri üzerinde yaptıkları çalışmalarını sunmaktadırlar. Onları özverili çalışmaları için kutluyor, programa büyük katkıları olan firma yetkililerine ve danışmanlarımıza teşekkür ediyorum.

Bütün süreç boyunca yoğun ve özverili çalışmalarıyla programın hedeflerine ulaşması için büyük çaba gösteren program koordinatörleri Prof. Dr. Savaş Dayanık, Doç. Dr. A. Selin Kocaman ve Dr. Emre Uzun'a, Üniversite-Sanayi İşbirliği Öğrenci Projeleri Koordinatörü'müz Yeşim Gülseren'e, asistanlarımız, Atahan Bayır, Yunus Emre Çakır, Aslı Eroğlu ve emeği geçen herkese çok teşekkür ediyorum.

Prof. Dr. Bahar Y. Kara
Endüstri Mühendisliği Bölüm Başkanı

Chairperson's Message

Bilkent University Industrial Engineering Department strives for its students to grasp changes in technology and society and be lifelong learners and inquirers. One of the department's educational goals is that our students hold a holistic view of systems and problems backed up with analytical thinking. The department is the first engineering department in Turkey, the quality of whose education program was fully accredited by *the Accreditation Board for Engineering and Technology (ABET)* back in 2007.

For 29 years, the Industrial Engineering Department has been successfully running its exemplary *University-Industry Collaboration Program.* The program's objective is to have the department's senior students gain full-fledged industrial experience before getting full industrial positions. Six-to-seven member student groups attack real open problems of companies under the supervision of academic and industrial advisors.

Twenty-one projects are present at the *21th Industrial Engineering Project Fair and Competition.* At the fair, student groups present their year-long work and the outcomes of their projects. I congratulate them for their tireless and heart-whole hard work. I also thank the company representatives and academic and industrial advisors for their support and collaboration.

Finally, I thank course coordinators Prof. Dr. Savaş Dayanık, Assoc. Prof. A. Selin Kocaman and Dr. Emre Uzun, University-Industry Collaboration Student Projects Coordinator Yeşim Gülseren, graduate assistants Atahan Bayır, Yunus Emre Çakır, Aslı Eroğlu for their relentless efforts to ensure that the program succeeds.

Prof. Dr. Bahar Y. Kara
Industrial Engineering Department Chairperson

Teşekkür Mektupları

Appreciation Letters

Arçelik, 1955 yılında kurulan bir şirkettir ve dayanıklı tüketim ve tüketici elektroniği sektörlerinde üretim, pazarlama ve satış sonrası destek hizmetleri sunmaktadır. Arçelik, 45.000 çalışanı, 12 markası (Arçelik, Beko, Grundig, Blomberg, ElektraBregenz, Arctic, Leisure, Flavel, Defy, Altus, Dawlance, Voltas Beko), 9 ülkede, 30 üretim tesisi ve 53 ülkedeki iştirakleriyle global olarak faaliyet göstermektedir.

Arçelik, 1991 yılında Ar-Ge birimini kurarak kendi patentli teknolojilerini geliştirmeye başlamıştır. Bugün Türkiye ve dünyada 29 Ar-Ge merkezinde 2000'den fazla personelle çalışmaktadır. Arçelik, üniversitelerden gelen bilimsel bilginin sanayiye aktarılmasının önemini fark etmiş, bu doğrultuda üniversitelerle çeşitli işbirliği süreçleri yürütmektedir. Lisans tez çalışmaları, gerçek sorunlara çözümler bulmayı ve yenilikçi ürünler ile süreçler geliştirmeyi sağlamaktadır. Aynı zamanda mühendislik öğrencilerinin sanayi deneyimi kazanmalarına da katkı sağlamaktadır. Başarıyla tamamlanan projeler, mühendislik adaylarına ileride yapacakları çalışmalarda yol gösterici faydalar sunmaktadır. Bu nedenle, Bilkent Üniversitesi'nin Sanayi Odaklı Bitirme Projeleri'ni önemli bir katma değer olarak görmekteyiz.

Arçelik olarak, Bilkent Üniversitesi Rektörlüğü ve Mühendislik Fakültesi yönetici ve akademisyenlerine, ÜSİ Mezuniyet Projeleri Koordinatörü Sayın Yeşim Gülseren Hanım'a, üniversite-sanayi işbirliği projelerimizde bize destek olan değerli mühendis adaylarımıza ve projelerin uygulanabilirliğine yönelik geri bildirimler sağlayarak projelerimizi başarılı sonuçlara taşıyan Makina Mühendisliği, Endüstri Mühendisliği ve Elektrik-Elektronik Mühendisliği Departmanlarındaki saygıdeğer akademisyenlerimize teşekkür ederiz.

Evrim ÖZGÜL
Arçelik A.Ş.
Global Ar-Ge Teşvikleri ve Üniversite İlişkileri Kıdemli Yöneticisi

Arçelik Buzdolabı İşletmesi Metot Mühendisliği Yöneticiliği'nden,

1955 yılında Sütlüce'de kurulan Arçelik A.Ş. Türk beyaz eşya sektörüne ilk adımı çamaşır makinesi ile atmıştır. 1970'li ve 80'li yıllarda ürün gamı genişletilerek buzdolabı, bulaşık makinesi, pişirici cihazlar, elektronik ve küçük ev aletleri segmentlerinde tüketicilere ürünler sunmaktadır.

Arçelik, 40.000'i aşkın çalışanı, 12 markasıyla (Arçelik, Beko, Grundig, Blomberg, ElektraBregenz, Arctic, Leisure, Flavel, Defy, Altus, Dawlance, Voltas Beko), 9 ülkede, 28 üretim tesisi, 30 ARGE merkezi ve 49 ülkedeki iştirakleriyle global olarak faaliyet göstermektedir.

Bilkent Üniversitesi Endüstri Mühendisliği Akademisyenleri, Üniversite-Sanayi İşbirliği Koordinatörleri ve öğrencilerinin katkıları ile Eskişehir Buzdolabı İşletmesi'nde "Buzdolabı Taleplerinin Üretim Tesislerine Dağılımı" projesini yürütmekteyiz. Bu proje ile şirketimizin yapmakta olduğu "Uzun Vadeli Planlama" prosesi içeriğine temel oluşturan 5 yıllık global buzdolabı ürün taleplerinin mevcut üretim tesisleri arasında dağılımı için bir matematiksel model oluşturulmuştur. Bu model ile önümüzdeki 5 yılın lojistik maliyeti enazlamak amaçlanmıştır. Hazırlanan arayüz programı ile kapasite kullanım oranları, kazanılacak lojistik maliyeti avantajları gibi gösterge ve öngörüler tablo ve grafik yöntemleri ile tarafımıza sunulacaktır.

Gerçekleştirdiğimiz projenin hem şirketimiz hem de öğrenci arkadaşlarımız için son derece yararlı olduğuna inanıyoruz. Proje kapsamında emek harcamış olan öğrencilerimize, görüşleri ile projeye yön veren Bilkent Üniversitesi Akademisyenleri'ne ve süreç boyunca her zaman destek veren Üniversite-Sanayi İşbirliği Koordinatörleri'ne çok teşekkür ediyoruz. Söz konusu projelerde beraber çalıştığımız mühendis adaylarına bundan sonraki iş ve akademik hayatlarında başarılar dileriz.

Özlem DEVİREN USLU
Metot Mühendisliği Yöneticisi

Sayın Bilkent Üniversitesi Öğrencileri,

"Damperli Kamyon Kestirimci Bakım Tahmini" projesi için birlikte çalıştığımız süre boyunca ortaya koyduğunuz özverili çalışma ve emek için sizlere teşekkür ederiz. Bu projede gerçekleştirdiğimiz çalışmaların, işletmenin bakım süreçlerine olası bir katkı sağlayacağını düşünüyoruz. Proje ekibi olarak düzenli toplantılar gerçekleştirerek iş birimlerinin görüşlerini dikkate aldık. Sonuçlar, bazı geliştirme alanları olmakla birlikte, işletme ihtiyaçlarına cevap verebilecek bir çözüm sundu.

Gelecekteki projelerinizde edindiğiniz deneyimleri kullanarak daha da ilerleme kaydedeceğinize inanıyoruz. Ortaya koyduğunuz çabalara teşekkür eder ve başarılarınızın devamını dileriz.

Saygılarımızla,

Kürşad UĞUR
Pınar TEKİN
Haydar ÇINAR
Özge Göksu BAŞER

Meteksan Savunma Sanayi A.Ş., 2006 yılında Bilkent Holding bünyesinde faaliyet gösteren yüksek teknoloji şirketlerinin savunma sanayiine yönelik proje ve aktivitelerini tek bir çatıda birleştirmek ve koordine etmek üzere kurulmuştur. Şirketimizin kuruluş amacı savunma sanayi sektöründe, Savunma Sanayii Başkanlığı'nın da vizyonu doğrultusunda, Türk Silahlı Kuvvetleri ve diğer güvenlik güçlerimize yerli, milli, bağımsız ve özgün, yüksek teknoloji ürünleri ve alt sistemler geliştirmek ve üretmektir.

Meteksan Savunma Sanayii A.Ş. olarak 2022-2023 akademik döneminde gerçekleştirdiğimiz Üniversite-Sanayi İşbirliği Projeleri kapsamında sizlerle birlikte çalışma fırsatı bulduğumuz için büyük bir memnuniyet duyuyoruz. Bu süre zarfında gerçekleştirdiğimiz "Üretim Planlaması, Çizelgeleme ve Süreç Eniyileme" projesi, üretim hattımız için önemli geliştirmelerin sağlanmasına yönelik bir adım olmuştur.

Savunma sanayii alanında faaliyet gösteren ve yüksek teknoloji üretimi yapan bir şirket olarak, yenilikçi çözümler üretmek ve sektörde öncü bir konumda bulunmak hedeflerimizin temelidir. Bu doğrultuda, üniversite-sanayi işbirliklerinin önemi büyük bir öncelik taşımaktadır. Yetenekli ve vizyon sahibi genç mühendis adaylarıyla bir araya gelerek gerçekleştirdiğimiz bu projede, hem sizlere pratik deneyimler sunma fırsatı bulduk, hem de iş süreçlerimizi daha verimli hale getirmek adına değerli katkılarınızdan faydalandık. İlgili projede kullanılan karar destek sistemi sayesinde, üretim planlama süreçlerimizde de belirgin iyileştirmeler sağlanacağını öngördük. Ayrıca, anlık gelen değişikliklere hızlı ve sorunsuz bir şekilde uyum sağlayarak ileriye dönük planlamalar yapabileceğimiz bir yapı oluşturma hedefimizi de belirledik. Sizlerin analitik becerileri, özverili çalışmaları ve yenilikçi bakış açınız, projenin başarıyla tamamlanmasını sağladı ve bizlere yeni fikirler ve perspektifler kazandırdı. Sizlerle birlikte çalışmak, öğrencilerimize gerçek dünya deneyimleri kazandırma amacımızı gerçekleştirmemize de yardımcı oldu. Bu değerli işbirliği sayesinde, öğrencilerimizin mezuniyet sonrası kariyerlerine daha donanımlı ve sektör ihtiyaçlarına uyum sağlayabilecek şekilde hazırlanmalarına katkıda bulunduğumuza inanıyoruz. Bilkent Üniversitesi Endüstri Mühendisliği Bölümü olarak, bu işbirliği fırsatını bize sunarak projeye katılan öğrencilere ve bize rehberlik eden öğretim kadrosuna içtenlikle teşekkür ederiz.

M. Can AKSOY
Üretim ve Lojistik Direktörü

OPLOG

Bu sene onuncu yılını yurt içi ve yurt dışı açılımlarıyla kutlayan OPLOG, hem robotik geliştirmeleri hem de sipariş karşılama hizmetiyle ilgileri üzerine çekmektedir. Ülkemizde ve yurt dışında birçok depoda hizmet veren şirketimiz, müşterilerine özelleştirilmiş hizmetler sunmaktadır. Müşterilerimizin mal kabul aşamasında bize teslim ettiği ürünleri, gerekli koşullarda özenle saklamakta ve siparişleri özelleştirilmiş hizmetlerimizle taleplerine göre hazırlamaktayız.

Her deposunda aynı kalitede hizmet veren OPLOG, Bilkent Cyberpark bünyesinde yer almaktan ve Bilkent Üniversitesi Endüstri Mühendisliği ile ortak projeler yapmaktan gurur duymaktadır. Bu sene iki farklı projede öğrenciler ile bir araya gelen ekiplerimiz, depo içi ürün toplama süresini minimize etmek için "Kaotik Depolu Sipariş Karşılama Merkezlerinde Ürün Toplama Sürelerinin Enazlanması" isimli projede ve farklı bölgelerdeki depolarda talep analizleriyle desteklenerek bulundurulması gereken stok miktarını belirlemek için "Sipariş Karşılama Merkezlerine Envanter Paylaştıran Proaktif Karar Destek Sistemi" isimli projede birlikte çalıştılar.

Her zaman gelişimin ve yeniliklerin yanında yer alan şirketimiz adına iki projenin de farklı yönlerden iyileştirmelere ve verimliliğe katkıları, öğrencilerin azimleri ve çalışkanlıkları, akademisyenlerin ilgileri ve destekleri için teşekkür ederiz. Öğrencilerin bu merak ve heveslerinin hiç solmamasını diler, hayatlarının bir sonraki adımında başarılar dileriz. Akademisyenlerimizin aynı özenle yetiştirecekleri daha birçok nesli heyecanla beklemekteyiz.

Yaren YILMAZ
OPLOG AR-GE Mühendisi

Unilever olarak, 190'dan fazla ülkede satışı ve her gün 3,4 milyar insan tarafından kullanılan ürünleriyle Güzellik ve Sağlık, Kişisel Bakım, Ev Bakımı, Gıda ve Dondurma ürünlerinde dünyanın önde gelen tedarikçilerinden biriyiz. Vizyonumuz, sürdürülebilir iş dünyasında dünya lideri olmak ve amaç odaklı, geleceğe uygun iş modelimizin üstün performansımızı nasıl desteklediğini ortaya koymak üzerine inşa edilmiştir. İlerici, sorumlu bir işletme olma konusunda uzun bir geleneğe sahibiz. Sürdürülebilir iş stratejimiz olan Unilever Compass ile de gezegenin sağlığının iyileştirilmesi, insanların sağlığını ve esenliğini iyileştirmek ve daha adil ve sosyal açıdan daha kapsayıcı bir dünyaya katkıda bulunmayı hedefliyoruz.

Türkiye'de yüz yılı aşkın süredir inovasyonu ve veri odaklı çalışmayı önceliklendirerek faaliyet gösteriyoruz. Sektörün önde gelen liderlerinden biri olmamızdaki en büyük faktör şüphesiz geçmişten gelen birikimimizle birlikte dijitalleşme ve veri odaklı çalışmamızdır. İşimizi geliştirmek için yapay zekayı ve veri bilimini odağımıza alıyoruz. Bunu yapabilmek adına da bilimin ana merkezi olan üniversiteler ile iş birliği içinde çalışmanın çok kritik olduğunun bilincindeyiz.

Bu proje kapsamında Unilever olarak Algida ürünlerinin doğru perakende satış noktalarına doğru çeşitlilikte önerilmesi için veri odaklı bir çözüm geliştirilmesini amaçladık. Değerli akademisyenlerimiz ve öğrencilerimiz bu amaca yönelik olarak getirdikleri inovatif çözümler ile hali hazırdaki karar verme modellerimize yeni bir bakış açısı getirdiler. İş stratejimiz de hazırlanan modellere entegre edilerek, doğrudan sahada uygulanmaya hazır bir analitik çözüm geliştirildi.

Projeye olanak sağlayan saygıdeğer Üniversite-Sanayi İşbirliği Koordinatörleri'ne ve süreç boyunca değerli bilgileri ile her zaman destek olan Bilkent Üniversitesi'nin Akademisyenleri'ne teşekkürlerimizi sunuyoruz.

Birlikte gerçekleştirdiğimiz bu katma değerli projenin Bilkent Üniversitesi'ndeki mühendis adaylarımıza çok faydalı olduğunu ve ileride yapacakları çalışmalarda bir pusula niteliğinde olmasını umuyor, kendilerine çok teşekkür ediyoruz. Tüm öğrencilerimize bundan sonraki iş ve akademik hayatlarında başarılar diler, geleceğin mühendis adaylarını bir gün aramızda görmekten gurur duyarız.

Ömür GÖÇER
Unilever TUI, Data & Analytics Director

All the best of success...

and all the best of luck!

PROJELER
PROJECTS

Algida Önerilen Ürün Portföyü Optimizasyonu | 1

Unilever

Proje Ekibi
Eda Coşkun, Kerem Fidan, İpek Han, Ali Egemen Özdek,
Pelin Nur Özgür, Nilsu Gül Sözer, Okan Yıldız

Şirket Danışmanı
Sumru Ergök Eker
Veri Bilimi Yöneticisi

Akademik Danışman
Dr. Öğr. Üyesi Taghi Khaniyev
Endüstri Mühendisliği Bölümü

Özet

Unilever, Algida ürünlerinin iş ortakları aracılığıyla perakende satış noktalarına önerilmesinde veri odaklı bir yaklaşım amaçlamaktadır. Bu projeyle, satış noktalarındaki aylık toplam satış hacmini ençoklayacak ürün portföyünü belirleyen bir karar destek sistemi geliştirilmiştir. İlk aşamada, geçmiş veri kaynakları kullanılarak satış noktaları için önerilen bir ürün portföyünün sağlayacağı satış hacmini yaklaşık %60 doğrulukla tahmin eden bir derin sinir ağı modeli eğitilmiştir. İkinci aşamada, bu tahminler baz alınarak toplam satışı ençoklayacak önerilen portföyü belirleyen matematiksel optimizasyon modeli geliştirilmiştir. Analitik model, aktif ürün kataloğundaki değişiklikleri yansıtmak ve satış ekiplerinden gelen istekleri entegre etmek için şirket ile istişare sonucu son halini almıştır.

Anahtar Sözcükler: Ürün Portföyü, Optimizasyon, Satış, Karar Destek Sistemi, Tahminleme, Brüt Satış Değeri (BSD).

Algida Suggested Assortment Optimization

Abstract

Unilever, the global leader in the FMCG industry, aims to implement a data-driven approach for their assortment suggestions of Algida products in their points of sales (PoS) by collaborating with their business partners. The purpose of this project is to develop a decision support solution that identifies the suggested optimal product assortment which maximizes the total predicted monthly gross sales value (GSV) for each PoS. In the first stage of the project, a multi-modal database containing historical data is used to train a deep neural network which can predict the total expected GSV for a suggested assortment in a PoS with an accuracy of approximately %60. In the second stage, a mathematical optimization model is formulated to identify the suggested optimum product assortment that would maximize the predicted GSV based on the predictions of the neural network trained in the previous stage. This end-to-end analytics solution pipeline took its final form after several rounds of iterative updates to reflect the changes in the active product portfolio and to integrate the new business requirements requested by the sales teams. Suggested assortments proposed by the model, which on average offers a 13.5% predicted increase in total GSV, were carefully analyzed and approved by the business teams. To validate the efficacy of the proposed solution, a pilot implementation (still ongoing) has been initiated in ten PoSs.

Keywords: Suggested assortment, optimization, sales, decision-support system, forecasting, gross sales value (GSV).

1.1 Algida and Problem Identification

1.1.1 Company description and system analysis

Unilever is one of the world's leading suppliers of beauty & wellbeing, personal care, home care, nutrition, and ice cream products, with sales in over 190 countries and products used by 3.4 billion people every day. They have generated sales of EUR 60.1 billion in 2022. Their vision is to be the global leader in sustainable business and to demonstrate how their purpose-led, future-fit business model drives superior performance. They have a long tradition of being a progressive, responsible business. The Unilever Compass, their sustainable business strategy, is set out to help them deliver superior performance and drive sustainable and responsible growth, while: (i) improving the health of the planet, (ii) improving people's health, confidence, and wellbeing, (iii) and contributing to a fairer and more socially inclusive world.

Algida has the biggest share in the Turkey ice cream market among other brands. There are four main ice cream categories that Algida sells its products, which are Category 1, Category 2, Category 3 and Category 4 in different ice cream cabinets with various capacities, such as Cabinet Type 1 and Cabinet Type 2. Currently, ice cream cabinets are mainly operated and replenished by the Algida Sales Representatives (ASRs). Each ASR is responsible for specific point of sales (PoS) and visits them with a particular frequency based on replenishment needs. ASRs refill the ice cream cabinet with specific suggested assortments of products according to the segment of that PoS. These segments are “Segment 1”, “Segment 2”, “Segment 3”, “Segment 4”, and “Segment 5”. Additionally, ASRs are the primary contact persons with the owners of PoSs in the whole sales processes. This means ASRs are significant players in the decision-making processes.

1.1.2 Problem and deliverables

Algida’s PoSs are distributed among various districts which bear different demographic characteristics, signalling the need for differentiated assortments. Currently, the suggested assortments are affected by the judgment of ASRs while the company’s wish is to automate this process with a data-driven approach and increase sales with well-predicted suggested assortments. In our problem focus, we only considered Category 1 type products and Cabinet Type 1 for pilot implementation.

Therefore, the main deliverable we provided the company with is a decision support tool that determines a suggested product assortment by a static suggested assortment optimization model. This main deliverable also contributes to achieving an increase in the number of sales in each PoS, enhancing the distribution approach of Algida brands and enabling Algida to analyze customer behavior better to perform their sales operations.

To observe the improvements and success of the deliverables of our model, we set some performance measures during pilot implementation. The main performance measure is the percentage of Gross Sales Value (GSV) increase for each PoS. We expect the GSV of each PoS to increase with the new suggested assortments provided by our data-driven suggested assortment optimization model. Additionally, another performance measure is balancing variety in ice cream categories (Category 1.1, Category 1.2, Category 1.3, Category 1.4) for each PoS achieved by the recommended “Price Range” constraints enabling our model to include various products from each recommended price range to ensure inclusiveness.

1.2 Model and Proposed System

1.2.1 Conceptual model

We received customer segments, product table, sales, store attributes, and cabinet data from the company, which are the inputs of our solution. Three input streams are created: Static (X) vector, Y matrix and Z vector. X vectors contain non-time series data about the PoSs' static features (e.g., demographic of the neighborhood, closeness to school etc.), Y matrix is a historical sales matrix indicating the past 12 months' sales records for each sub-brand in terms of GSV value for each PoS and each observation month, and Z vectors contain binary values indicating the existence of each product in the current suggested assortment of PoSs. Then, we ran our machine learning model, analyzed the characteristics of each PoS and obtained weights of each sub-brand accordingly which are necessary for the utilization of our suggested assortment optimization model. As the output, our model maximizes the predicted total GSV over the next month for each PoS. The conceptual model to the Algida Suggested Assortment Optimization problem can be seen in Figure 1.1.

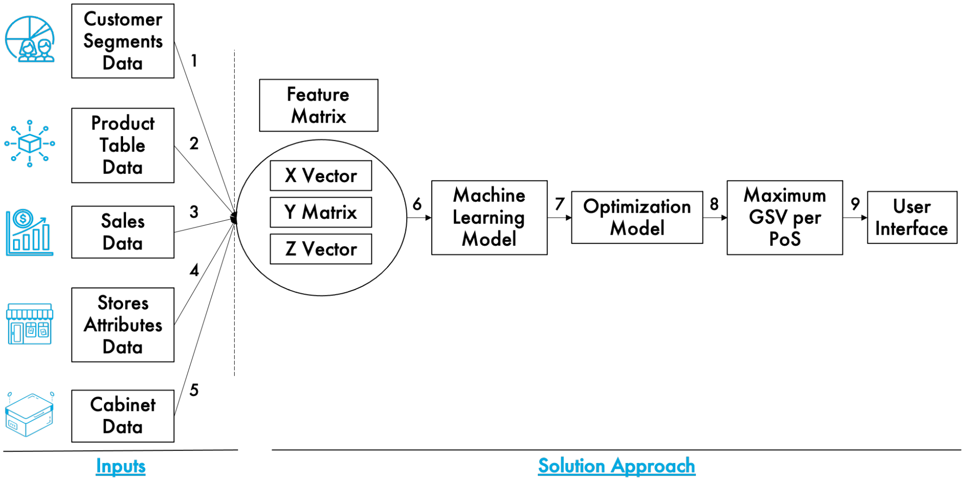


Figure 1.1: Conceptual model

1.2.2 Training a machine learning model

Due to the time series nature of the past sales data, it is needed to use a recurrent neural network. LSTM is a form of recurrent neural network that is introduced in order to eliminate the vanishing gradient issue of simple RNN architectures (Hochreiter and Schmidhuber, 1997). Since we worked with labeled data and had a response variable (S) which is the GSV value, we decided to use supervised training and utilize LSTM in our model to be able to obtain better results. We decided to work with 1,000 PoS chosen in line with the frequency of their districts' recordings in sales data and 15 observation months, only 15,000 observations of our matrices were included in the model training. Secondly, since neural networks were utilized in our ML algorithm and they do not work well when the differences between the scales are high, we normalized the GSV values in the Y matrix into the interval of zero and one. In other words, the maximum value of GSV takes value of one, and the minimum value of GSV takes value of zero, and the rest was linearly distributed. After completing the preparation of the data, we splitted the matrices into three parts which are the training data, the validation data, and the test data. We chose the initial 12 observation months for each PoS and assigned randomly chosen 9 of them (it takes 60% of the total rows) for the training, while we assigned 3 of them (it takes 20% of the total rows) for the validation. The last 3 months (it takes 20% of the total rows) are chosen for the test data.

In our model, we used a moderate dropout rate of 0.5 to prevent our model from overfitting and underfitting. Moreover, we decided to use more than one layer in our deep learning model to provide better learning. However, to get an efficient running time, we ended up using two layers. Dense, which is the number of nodes in a layer, is another parameter that we decided on. By taking the running times into account, we set the two layers' dense as 400 and 40 respectively in our model. Furthermore, the number of epochs demonstrates how many iterations are done. It is expected that the performance of the model on the training improves as epoch number increases so we set the epochs to 100, beyond which we did not observe any significant improvement in the training performance. The goal was to capture the model with the best test performance according to the R-squared value which was the main indicator to be used in evaluating the performance of our ML model (Fox and Monette, 1992). Therefore, just before over-fitting the training data, we tried to catch the value before the validation performance deteriorates. We recorded the epoch giving the least error and its weights to utilize them in the optimization part. Overall, the output of the ML model is an aggregated prediction of the corresponding total GSV value for the given suggested assortment.

1.2.3 Suggested assortment optimization model

To determine the optimal suggested assortment for each PoS, we developed a mathematical model whose inputs will be the feature matrices and weights obtained from the trained ML model that we have constructed. The main objective of the model is to find the suggested assortment maximizing the GSV over the next month of each PoS. The model works according to the below constraints in general:

- First and second layer linearization constraints in which there are 400 and 40 nodes, respectively (Fischetti and Jo, 2018).
- Recommended "Price Range" constraints. The aim is to maximize the number of products placed in the suggested assortment from each recommended price range, which enables customers from different demographics to have equal chances to reach their preferred products.
- A constraint that ensures the number of products in the current suggested assortment and the number of products in the new suggested assortment are the same for each PoS.
- Deviation constraints limit the number of different products between the current and new suggested assortments to a maximum of 4, enhancing consistency with the current system.
- Delist products elimination constraint to ensure that delist products in 2023 selection are not added to the optimal suggested assortment.

Parameters, decision variables and the mathematical model can be seen in the Appendix.

1.2.4 Utilized platforms and software

Our solution approach for determining the suggested optimal ice cream assortments in PoSs consists of three steps: machine learning, optimization, and user interface.

In machine learning, we used three inputs: past sales, static features, and whether a product is in the current suggested assortment or not as a binary variable. These inputs are used to train our model, which will be run once a year in January with updated input data. We use Python for our machine learning, which Algida already has access to.

In optimization, we used Gurobi to optimize the suggested assortments based on sales data and static features. For user interface, we have written our UI in React Native, which can be easily integrated into Algida's existing app Pera as a plugin. This will allow ASRs to see our suggested assortments and make recommendations to the sales points.

1.3 Validation of Our Approach

Throughout the modeling stage, we conducted several discussions with ASRs. These interactions revealed that ASRs prefer a flexible solution that minimizes deviation from the current suggested assortment, allowing them

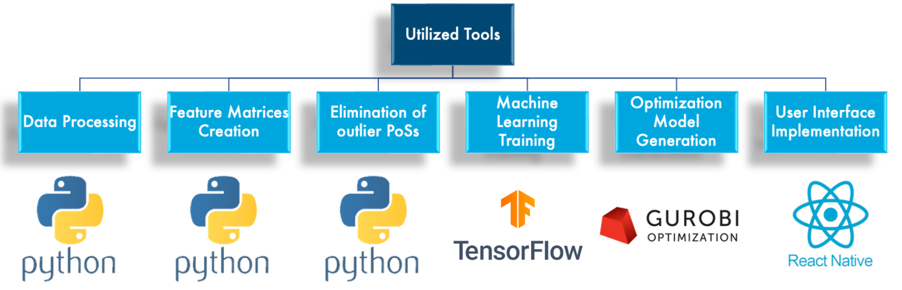


Figure 1.2: Utilized platforms and softwares

to control the number of products that differ from the existing suggested assortment. Taking this into consideration, we finalized the deviation constraint to ensure practicality and alignment with ASRs' preferences.

In our model validation process, we compared our machine learning predictions with the Multiple Linear Regression (MLR) Model to determine the extent to which GSV can be attributed to linear relationships versus nonlinear patterns. Through examination of the difference in test R-squared values, we concluded that our ML model outperformed the MLR model by capturing some of the nonlinearities in the data.

To further validate our optimization model, we compared the different assortments suggested by our model under various requirements. Specifically, we examined whether the suggested assortments included the products with high coefficients in the MLR model when we want to increase the number of products, and excluded the products with low coefficients when we wanted to decrease. Our conceptual representation revealed a positive correlation between coefficients and being included in an suggested assortment.

Overall, our approach can be flexibly adjusted to include more or fewer products based on the current suggested assortment, while considering recommended "Price Range" constraints, provides a meaningful solution to ASRs' daily operations. This is particularly relevant as Unilever's current assortment suggestion system relies heavily on human judgment and only considers sales forecasts of products in the existing suggested assortments.

1.4 Integration and Implementation

Integration of our solution approach for determining the optimal suggested ice cream assortments in PoSs involves a pilot study with ten intervention and ten control PoSs, which are selected from a predetermined selected district. These ten PoSs are from Segment 4 and Segment 5. The reason behind this situation is the choice of the company. They preferred all selected PoSs to be served by the same distributor and accordingly to be in the

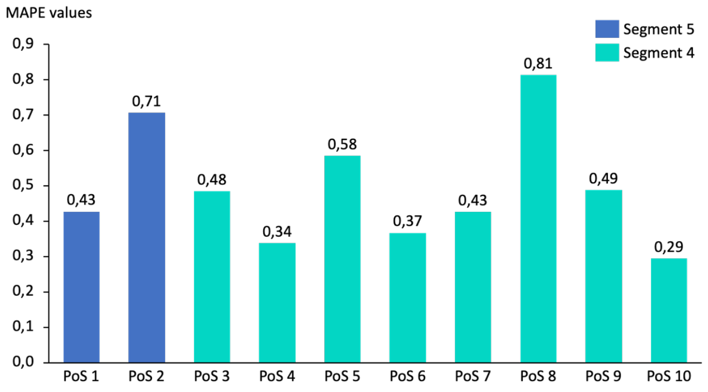


Figure 1.3: MAPE values for the selected ten PoSs

same or similar segments for the ease of the pilot implementation. Hence, the chosen distributor is from the predetermined district and PoSs are from Segment 4 and Segment 5. The control group was chosen to have very similar static characteristics and MAPE values to our intervention group; Figure 1.3 shows the MAPE values of the intervention group.

Our target metric for evaluating performance is the predicted total GSV Value in the next month, which is compared between the intervention and control groups under the results of approximately one month period.

We are conducting a pilot study to evaluate the effectiveness of our suggested assortments and make enhancements in our solution approach as necessary. The insights that are going to be gained from this study will enable us to optimize suggested the ice cream assortments in PoSs and maximize overall sales. Tables 1.1 and 1.2 show suggested assortments for these two segments.

At the end of the pilot study, as we are going to observe the actual change in the GSV of each PoS, we will look at the "difference of differences" between the control group and the intervention group to measure the actual net change in the GSV that is occurred only through the new assortment suggestion strategy in the intervention group. Since the net GSV change may be affected from the other factors that are beyond our control such as the change in the economic status of the country, change in the sales and production approach of the Algida, we are comparing the net GSV increase in the control group based on the previous years' realized GSVs with the actual net change in the GSV of intervention group.

It must be stressed that Algida and Algida distributor(s) only share "recommended resale prices" with PoSs. According to competition law rules, each PoS is absolutely free to set its own resale prices (which may differ from recommended prices). Therefore, the realized GSVs will depend

Segment 4			
PoS	Suggested Assortment		
3	Category	Sub-brand Code	Sub-brand Name
4	Category 1.4	1	COMBO KAKAO VANILYA
5	Category 1.1	3	S.CORN. CHOCO DISC VANILYA
6	Category 1.1	4	CORNETTO ALL CHOCOLATE
7	Category 1.2	5	MAGNUM DOUBLE/KARADUT-BOGURTLEN
8	Category 1.4	8	FRIGOLA
9	Category 1.4	9	COMBO CILEK VANILYA
10	Category 1.1	10	S.CORN MILKA
	Category 1.4	11	ALGIDA MARAŞ CUP
	Category 1.2	12	MAGNUM COOKIE
	Category 1.2	13	MAGNUM VANILLA/CHOC
	Category 1.2	14	MAGNUM WHITE
	Category 1.4	15	CLASSICS ANTEPFISTIKLI
	Category 1.4	16	NOGGER SANDWICH
	Category 1.1	17	S.CORN. PISTACHIO DISC
	Category 1.1	18	S.CORN OREO DISC
	Category 1.3	19	TWISTER ORMAN MEYVELERI
	Category 1.1	20	CORNETTO KAYMAK
	Category 1.4	22	ALGIDA BOOM BOOM
	Category 1.1	23	ALG.MRS.CONES
	Category 1.2	24	MAGNUM DOUBLE/CHOCOLATE
	Category 1.2	25	MAGNUM DOUBLE SUNLOVER
	Category 1.2	26	MAGNUM DOUBLE STARCHASER
	Category 1.4	27	ALGIDA BOOM BOOM GOLD
	Category 1.1	29	CORNETTO ASK YILDIZI
	Category 1.3	30	ALGIDA MOO SHAKE

Table 1.1: The suggested assortment for Segment 4

on the resale prices set by PoSs.

After the completion of the pilot study, implementation is quite straightforward. As Unilever does not want to deviate from their current suggested assortments by more than four products, our model will propose four new products in the suggested assortment as long as it increases the GSV in that PoS by eliminating the four products that are least contributing to the GSV. ASRs will be able to see the newly proposed products in the suggested assortments via their tablets.

1.5 Benefits to the Company

1.5.1 Gross sales value increase

Through rigorous analysis and advanced data modeling techniques, we conducted a comprehensive estimation of the GSV increase for each selected PoS. Our forecasts revealed a wide range of potential growth, spanning from 12.82% to 15.56% increase in GSV, as seen in Figure 1.4. Overall, the average GSV change across all selected ten PoSs was projected to be a substantial 13.5%; see Table 1.3.

Segment 5			
PoS	Suggested Assortment		
1	Category	Sub-brand Code	Sub-brand Name
2	Category 1.4	1	COMBO KAKAO VANILYA
	Category 1.2	2	MAGNUMVAN/ALM.CHOC
	Category 1.1	3	S.CORN. CHOCO DISC VANILYA
	Category 1.1	4	CORNETTO ALL CHOCOLATE
	Category 1.2	5	MAGNUM DOUBLE/KARADUT-BOGURTLEN
	Category 1.3	6	BUZ PARMAK LEMON -30
	Category 1.3	7	MAX MINIMILK KARMA
	Category 1.4	8	FRIGOLA
	Category 1.4	9	COMBO CILEK VANILYA
	Category 1.1	10	S.CORN MILKA
	Category 1.2	12	MAGNUM COOKIE
	Category 1.2	13	MAGNUM VANILLA/CHOC
	Category 1.2	14	MAGNUM WHITE
	Category 1.4	16	NOGGER SANDWICH
	Category 1.1	17	S.CORN. PISTACHIO DISC
	Category 1.1	18	S.CORN OREO DISC
	Category 1.1	21	CORNETTO KAYMAK
	Category 1.4	22	ALGIDA BOOM BOOM
	Category 1.1	23	ALG.MRS.CONES
	Category 1.2	24	MAGNUM DOUBLE/CHOCOLATE
	Category 1.2	25	MAGNUM DOUBLE SUNLOVER
	Category 1.2	26	MAGNUM DOUBLE STARCHASER
	Category 1.4	27	ALGIDA BOOM BOOM GOLD
	Category 1.1	28	CORNETTO DOGRULUK CESARET
	Category 1.1	29	CORNETTO ASK YILDIZI

Table 1.2: The suggested assortment for Segment 5

The results of our analysis unveiled growth possibilities, with all PoSs expected to experience sensible increases in GSV value. Furthermore, as it is observed, all of the PoSs have similar GSV increases but not the same. This situation underscores that even if the PoSs are from the same or similar segments, our model achieves to obtain different GSV increases due to its capability of considering different static features of them.

1.5.2 Balanced ice cream category variety

Balancing variety in ice cream categories (Category 1.1, Category 1.2, Category 1.3, Category 1.4) for each PoS is our second performance criteria. This balance is achieved by the recommended "Price Range" constraints enabling our model to include various products from each recommended

Segment	**GSV increase**
Segment 4	13.07%
Segment 5	14.94%

Table 1.3: Two segments and respective GSV increases

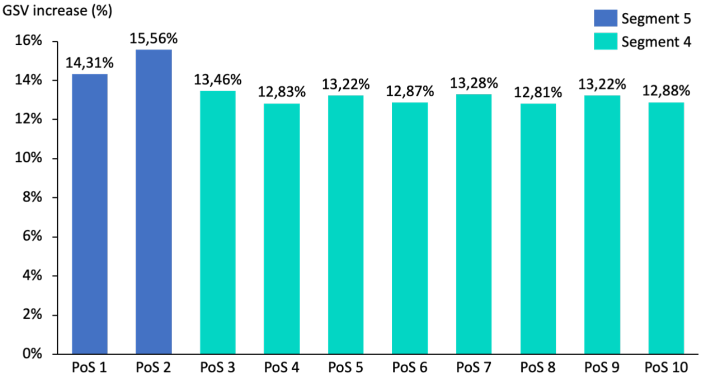


Figure 1.4: GSV increases for the selected ten PoSs

price range. Since each recommended price range approximately matches with a category, our model yields a more balanced ice cream category division. The overall summary of comparison between current and optimized suggested assortment can be seen in Table 1.4.

It is important to note that these results are theoretical and obtained by comparing the results of optimal suggested assortments with current suggested assortments. Estimations are based on comprehensive data analysis, but market dynamics are subject to change, and various external factors can influence actual outcomes. Therefore, continuous monitoring and adaptation is going to be embraced during the pilot study. Hence, practical results for both performance measures are going to be gathered with the pilot.

1.6 Conclusion

This project aims to develop a data-driven approach for determining the optimal suggested assortments of Algida products in each PoS by eliminating the human-bias in the current system. This data-driven approach is constructed by utilizing machine learning algorithms and with the help of optimization tools. As the company's expectation from us was to capture a GSV increase while also considering balanced diversification among

	Optimized Suggested Assortment	**Deviation**
Category 1.1	32.80%	-4.00%
Category 1.2	28.80%	-3.20%
Category 1.3	8.00%	0.80%
Category 1.4	30.40%	6.40%

Table 1.4: Overall category breakdown of optimized suggested assortments and deviations from currently suggested assortments

ice cream categories with the newly proposed suggested assortment in each PoS, we are able to observe GSV increase for each PoS while also considering balanced diversification among ice cream categories.

As this project is planned to be finalized in June 2023, there is a possibility that the model will be updated with gathered data from the pilot study results. Observing tangible benefits to the company with the pilot implementation, we foresee that our model can be fully integrated into the company's operating system as a future suggestion.

Bibliography

Fischetti, M. and J. Jo (2018, 07). Deep neural networks and mixed integer linear optimization. *Constraints 23*.

Fox, J. and G. Monette (1992). An examination of r-squared and goodness-of-fit in regression models. *Journal of the American statistical Association 87*(419), 113–119.

Hochreiter, S. and J. Schmidhuber (1997). Long short-term memory. *Neural Computation 9*(8), 1735–1780.

Appendix: The Generalized Model

The Parameters

- $\text{feature\_}z_i = \begin{cases} 1, & \text{if sub-brand } i \text{ is in the current assortment,} \\ & \forall i \in \{1, \ldots, 112\} \\ 0, & \text{otherwise} \end{cases}$

- $dense_1 : 400 =$ Number of nodes in hidden layer 1.

- $dense_2 : 40 =$ Number of nodes in hidden layer 2.

- w^1_{bj} = Weight value for the bias that comes from first layer from node j, $j \in \{1, ..., 400\}$.

- w^2_{bj} = Weight value for the bias that comes from first layer from node j, $j \in \{1, ..., 40\}$.

- w^3_{bj} = Weight value for the bias of the output node.

- w^1_{xkj} = Weight value that comes from first layer from node j for X vector's kth index, $j \in \{1, ..., 400\}$, $k \in \{1, ..., 100\}$.

- w^1_{ykj} = Weight value that comes from first layer from node j for Y matrix's kth index, $j \in \{1, ..., 400\}$, $k \in \{1, ..., 100\}$.

- w^1_{zkj} = Weight value that comes from first layer from node j for Z vector's kth index, $j \in \{1, ..., 400\}$, $k \in \{1, ..., 112\}$.
- w^2_{kj} = Weight value comes that from first layer's k-th node and goes to j-th node of second layer, $k \in \{1, ..., 400\}$, $j \in \{1, ..., 40\}$.
- w^3_k = weight value that comes from the 2nd layer's k-th node and goes to the output node, $k \in \{1, ..., 40\}$.
- $currentproductamount_i$ = Total product amount in the current assortment of ith observation, $i \in \{1, ..., 3000\}$.
- DPI = List of delist products' indexes among $[1, 112]$.
- G_1 = Index list of sub-brands having $\frac{\text{GSV}}{\text{Liters}} < 30$.
- G_2 = Index list of sub-brands having $30 \leq \frac{\text{GSV}}{\text{Liters}} < 50$.
- G_3 = Index list of sub-brands having $50 \leq \frac{\text{GSV}}{\text{Liters}} < 80$.
- G_4 = Index list of sub-brands having $80 \leq \frac{\text{GSV}}{\text{Liters}}$.
- Seg_i = Segment of the ith PoS,
  $i \in$ {Sub-Economy, Economy, ..., Premium+}.

The Decision Variables

- $z_i = \begin{cases} 1, & \text{if sub-brand i is in the assortment,} \\ & \forall i \in \{1, \ldots, 112\} \\ 0, & \text{otherwise} \end{cases}$
- x_i = Static features provided by the company, $i \in \{1, \ldots, 100\}$.
- y_i = Past sales data, $i \in \{1, \ldots, 100\}$.
- h_{ij} = Intermediate output of hidden layer i's jth node , $i \in \{1, 2\}$, $i \in \{1, \ldots, 400\}$.
- $u_{ij} \in \{0, 1\}$ = Binary variable that is identified for the linearization of the ith hidden layer's jth node, $i \in \{1, 2\}$, $i \in \{1, \ldots, 400\}$.
- S = Output of deep neural network's algorithm representing the total GSV.
- δ_i = Decision variable showing the difference between current assortment and the assortment given by the model for product i, $i \in \{1, ..., 112\}$.
- abs_δ_i = Decision variable showing the absolute value of the δ_i, $i \in \{1, ..., 112\}$.

The Generalized Model

$$\max S$$

s.t.

- $h_{0j} \geq w^1_{bj} + \sum_{k=1}^{100} w^1_{xkj} \times x_k + \sum_{k=1}^{100} w^1_{ykj} \times y_k + \sum_{k=1}^{112} w^1_{zkj} \times z_k, \forall j \in \{1, ..., 400\}$
- $h_{0j} \geq 0, \forall j \in \{1, ..., 400\}$
- $h_{0j} \leq w^1_{bj} + \sum_{k=1}^{100} w^1_{xkj} \times x_k + \sum_{k=1}^{100} w^1_{ykj} \times y_k + \sum_{k=1}^{112} w^1_{zkj} \times z_k + M \times (1 - u_{0j}), \forall j \in \{1, ..., 400\}$
- $h_{0j} \leq M \times u_{0j}, \forall j \in \{1, ..., 400\}$
- $h_{1j} \geq w^2_{bj} + \sum_{k=1}^{400} w^2_{kj} \times h_{0k}, \forall j \in \{1, ..., 40\}$
- $h_{1j} \geq 0, \forall j \in \{1, ..., 40\}$
- $h_{1j} \leq w^2_{bj} + \sum_{k=1}^{400} w^2_{kj} \times h_{0k} + M \times (1 - u_{1j}), \forall j \in \{1, ..., 40\}$
- $h_{1j} \leq M \times u_{1j}, \forall j \in \{1, ..., 40\}$
- $S = w^3_{bj} + \sum_{k=1}^{40} w^3_k \times h_{1k}$
- If $Seg_i =$ "Premium", then;
  - $\sum_{k \in G_1} z_k \geq 2$
  - $\sum_{k \in G_2} z_k \geq 1$
  - $\sum_{k \in G_3} z_k \geq 1$
  - $\sum_{k \in G_4} z_k \leq 5$
- If $Seg_i =$ "Premium+", then;
  - $\sum_{k \in G_1} z_k \geq 4$
  - $\sum_{k \in G_2} z_k \geq 3$
  - $\sum_{k \in G_3} z_k \geq 1$
  - $\sum_{k \in G_4} z_k \leq 6$
- $\sum_{k=1}^{112} z_k = currentproductamount_i$
- $\delta_i = |z_i - feature\_z_i|$
- $abs\_\delta_i = |\delta_i|$

- $\sum_{i=1}^{112} abs\_\delta_i \leq 4$
- $\sum_{k \in DPI} z_k = 0$
- $x_i, y_i \geq 0, \forall i \in \{1, ..., 100\}$
- $z_i \geq 0, \forall i \in \{1, ..., 112\}$
- $u_{0j} \in \{0, 1\}, \forall j \in \{1, ..., 400\}$
- $h_{1j} \in \{0, 1\}, \forall j \in \{1, ..., 40\}$
- $u_{1j} \in \{0, 1\}, \forall j \in \{1, ..., 40\}$
- $abs\_\delta_i \in \{0, 1\}, \forall i \in \{1, ..., 112\}$
- $\delta_i \in \{-1, 0, 1\}, \forall i \in \{1, ..., 112\}$

2 Arçelik Bulaşık Makinesi Yan Montaj Hattı Gruplaması

Arçelik Bulaşık Makinesi İşletmesi

Proje Ekibi
Nurefşan Taş, Neslihan Selin Gökdemir, Bilge Su Arık, Aslı Ayça Zararsız
Baha Bora Şenel, Mahmut Bilici, Defne Altınalan

Şirket Danışmanı
Muhammet Erol
Endüstri Mühendisi

Akademik Danışman
Doç. Dr. Ayşe Selin Kocaman
Endüstri Mühendisliği Bölümü

Özet

Arçelik Bulaşık Makinesi Fabrikası'nın elektronik kart gruplama ünitesinde üretkenlik ve verimlilik ile ilgili sorun yaşanmaktadır. Ayrıca kartların montaj hatlarından bağımsız bir yerde gruplandırma işlemlerine tabi tutulmuş olması, daha fazla işçilik ve fabrika içi ulaşım ihtiyacı yaratmaktadır. Bu proje sayesinde kart gruplama ünitelerinin montaj hatlarına entegre edilmesi sağlanarak fabrikada senkron üretime ulaşılması hedeflenmektedir. Kart gruplama ünitesindeki üretkenlik ve etkinlik düzeylerini artırmanın yanı sıra işçilik maliyetlerinin ve nakliye süresinin azaltılmasına da katkı sağlayabilir.

Anahtar Sözcükler: kart gruplama, etkinlik, üretkenlik, eşzamanlı üretim

Arçelik Dishwasher Side Assembly Line Grouping

Abstract

In the current system of Arcelik Dishwasher Factory, there is an efficiency issue in the electronic card grouping unit. Also, the cards have undergone grouping processes at an independent place from the assembly lines, which causes the requirement of more labor and intra-factory transportation. This project aims to reach synchronous production in the factory by integrating card grouping units into assembly lines. It may contribute to reducing labor costs and transportation time, besides increasing productivity and efficiency levels in the card grouping unit.

Keywords: card grouping, efficiency, productivity, synchronous production

2.1 Company Information

Arçelik was founded in 1995 by Vehbi Koç and Lütfi Doruk. Arçelik rapidly expanded product range by adopting the principle of "One product-one factory" in 1970 and 1980. Arçelik operates globally with more than 40,000 employees and 12 brands (Arçelik, Beko, Grundig, Blomberg, ElektraBregenz, Arctic, Leisure, Flavel, Defy, Altus, Dawlance, Voltas Beko), 28 production facilities in 9 countries. Arçelik's factory in Ankara started mass production in 1993 and produces dishwashers.

2.2 System Analysis

2.2.1 System Description

The current system consists of assembly lines, metal molding areas, plastic injection units, and an electronic card grouping unit. 4 lines of assembly work simultaneously to assemble the machines, the lines output a dishwasher every 16 seconds. The demand/production level for the card grouping units is derived from the demands of the assembly lines and recorded in an excel file. Engineers manually review the data and make decisions based on them. There are two rooms for the card grouping. Testing the card, grouping and labeling are done in one of these rooms. After these processes are done, the WIPs are transferred to the other room. In this room, WIPs go through processes such as installing models into cards, grouping cables, labeling them, and putting them in a box for transfer to assembly lines. Workers produce 13 units at a particular time. The parts that come from the subcontractor also need to undergo these processes for which the second room is responsible.

2.2.2 Problem Definition

In the current system, there is a problem with efficiency and effectiveness, which causes inequality in the workload allocation among workers, especially in the first part of the card grouping unit. The decision-making system is made manually during operation, which prevents some side assembly lines from seeing the necessary information, such as the occupancy rate of side assembly lines, so it is difficult to know what the operator, machine and arrival in the system will need. The change in demand in the card group room causes efficiency and effectiveness problems. Since the electronic card grouping unit is separate from the main assembly line, there occurs waste workforce for storing and transporting cards.This problem can be solved effectively when the company adopts a layout that can allow card grouping unit operations to be performed simultaneously with the corresponding assembly lines. This project focuses on minimizing the number of operators needed and maximizing the efficiency of work and the utilization of space by integrating the current card grouping unit's cycle time into the existing assembly line. In the simple assembly line balancing with predefined cycle-time problem, our objective function is to minimize workstations as Kriengkorakot and Pianthong (2007) stated.

2.3 Model and Proposed Solution

2.3.1 Critical Assumptions

- Cycle time is examined as predetermined and deterministic
- There are locations for intermediate stocks on the assembly line.
- Task times for each task are known deterministically.
- Precedence relationships between tasks are known.
- The accepted maximum number of operators in each workstation is determined.
- The travel times of operators are ignored.

2.3.2 Major Constraints

- Demand information comes in every two weeks.
- Some specific jobs within the precedence relation rules will not start before the previous jobs are finished.

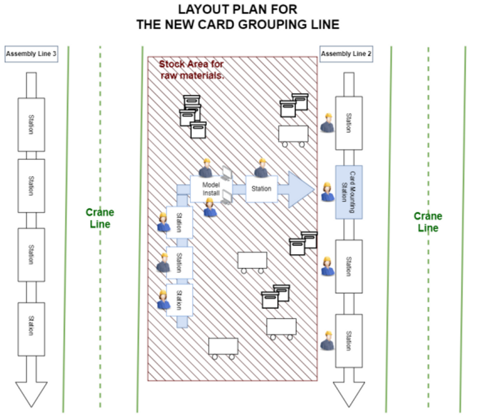


Figure 2.1: Proposed layout plan

- The completion time of each task will not exceed the cycle time.
- Stations will not be able to exceed certain workload limits.
- Each task can be assigned to exactly one station.

2.3.3 Solution Approach

Solution approach aimed to address major inconveniences in Card Grouping process by relocating and integrating the unit into the assembly line, to avoid carrying semi-finished products twice. A time study was conducted to identify possible spaces where new stations could be inserted, and three of the assembly lines have enough room for new stations. A mathematical model was developed to balance Card Grouping process using the SALBP-1 method, along with heuristic methods such as the Largest Candidate Rule and the Ranked Positional Weighted method. Furthermore, a conceptual layout was proposed for the new stations; see Figure 2.1. To analyze the data and make decisions regarding the number of shifts and workers required to meet the demand and production rate, an interface using Python and/or Excel Macro VBA was developed to import data from SQL. The model was designed to run in different scenarios to ensure that the Card Grouping process does not become a bottleneck for the assembly.

2.3.4 Mathematical Model

Let $s = 1, \ldots, n$ be stations, $t = 1 \ldots m$ be tasks, T_t be time per task t, and C be the cycle time of the assembly line. The model has two decision variables: $X(t, s)$ decides which task should be assigned to which station and $Y(s)$ create new stations when the previous station is full; namely,

$$X_{t,s} = \begin{cases} 1 & \text{if task } t \text{ is assigned to station } s \\ 0 & \text{otherwise} \end{cases}$$

$$Y_s = \begin{cases} 1 & \text{if station } s \text{ is created} \\ 0 & \text{otherwise} \end{cases}$$

The objective function is to minimize the total number of the stations

$$\sum_{s=1}^{n} Y_s. \tag{2.1}$$

The model has seven constraints:

$$\sum_{t=1}^{m} T_t \cdot X_{t,s} \leq C \cdot Y(s) \quad \forall s : 1 \ldots n \tag{2.2}$$

$$\sum_{s=1}^{n} X_{t,s} = 1 \quad \forall t : 1 \ldots m \tag{2.3}$$

$$Y_{s+1} \leq Y_s \quad \forall s : 1 \ldots n-1 \tag{2.4}$$

$$\sum_{s=1}^{n} s \cdot X_{u,s} \leq \sum_{s=1}^{n} s \cdot X_{t,s} \quad \forall u < t \tag{2.5}$$

$$X_{t,s} \in 0,1 \tag{2.6}$$

$$Y_s \in 0,1 \tag{2.7}$$

Constraint (2) ensures that no established station exceeds the cycle time. (3) makes sure that every task is assigned to a station. The increment of the index of the stations to be opened is stabilized by (4). Constraint (5) checks the task precedence relationships. (6) and (7) are binary constraints.

2.4 Heuristics

2.4.1 Largest Candidate Rule

In this method, the tasks are assigned to workstations based on the size of elements time, T_e (work elements time) values which is the time it takes to complete the task. The cycle time is T_c.

The task times for the elements are sorted in descending order then the largest feasible task is assigned to a station. A feasible task is one that meets the precedence criteria while also preventing the total of the T_e values at the station from exceeding the cycle time T_c. When no more tasks are feasible that is either no task is left or the addition of a new task makes the total T_e larger than the T_c a new station is created. Then the same procedure is applied to the new station until no tasks are left.

2.4.2 Ranked Weighted Position Method

For each task, a ranked positional weight value (abbreviated RPW) is computed. The RPW considers both the task's Te value and its location in the precedence diagram. The tasks are then assigned to workstations in the order of their RPW values.

Tasks are ranked according to their importance to the completion of all tasks that depend on them. RPW for task j is defined as

$$RPW_j = T_{e_j} + \sum_{k \in F_j} T_{e_k},$$

where F_j is the set of all elements which follow e_j, k is an element of F_j if and only if k is reachable from j by a directed path that begins at j.

2.4.3 User Interface

We developed a user interface for our solution methods. We used tkinter and PySimpleGUI Python libraries to build the interface. After running the code, the user is greeted with a language selection menu. After selecting the language, the user is directed to a screen shown in Figure 2.2. They can choose one of the three available solution methods: Mathematical Model, Largest Candidate Rule, and Ranked Positional Weighted Method. This screen also includes a user manual button and a language selection button that enables the user to return to the language selection screen.

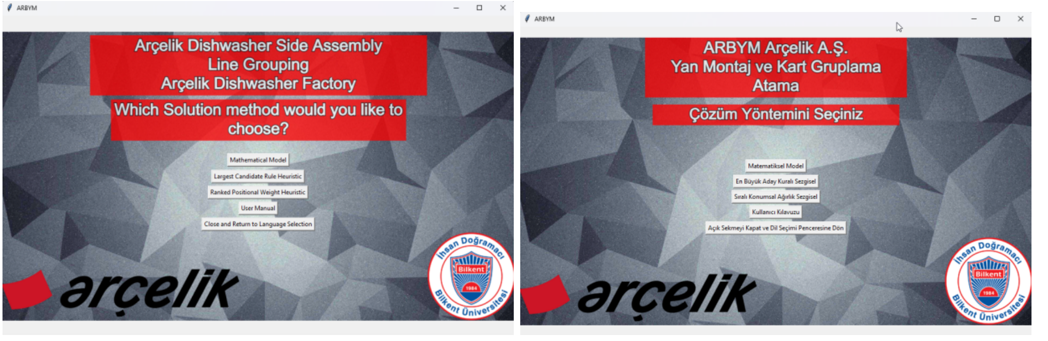


Figure 2.2: User interface

2.5 Validation

The system will be validated through conceptual validity, face validity and a pilot study with metrics, considering constraints, inputs, and outputs.

2.5.1 Conceptual Validity

The project aims to integrate card grouping units and assembly lines within limited space constraints. The number of workstations is minimized to achieve this while assuming that task times are deterministic and workers' errors are negligible. One task exceeds the cycle time, which is resolved by using parallel stations and manipulating task time values. The model provides a strategic solution for the facility's layout, which can be adjusted in response to changes in demand or cycle time.

2.5.2 Face Validity

Alongside the Industrial advisor Muhammet Erol, we identified suitable areas in assembly lines for card grouping stations. The plan includes removing unnecessary items and integrating L-shaped lines into open areas. Task times for each card are different, which may cause accumulation. However, Muhammet Erol believes this will not be a problem for the line.

2.5.3 Operational Validity

The team has planned to conduct a pilot study in the factory to validate the feasibility of the proposed solution. We will create one of the L-shaped lines and compare the pilot study's outcome with the planned outcome. The pilot study will focus on some critical metrics, including the number of stations, area constraints, and task times of stations. The team will ensure that the number of stations outputted by the model is sufficient to keep the assembly line going without interruption. In addition, we will validate that the task times of stations are not larger than the cycle time for the assembly line to operate smoothly. If there are any cards with a longer model installation that may interrupt the system, the team will develop a buffer policy to compensate for the interruption. The team will also maximize the number of material stocks during the pilot study to minimize transportation.

2.5.4 Mathematical Model Validation

The team conducted a validation process to check if the model's output was reasonable and accurate by comparing it with actual time study data. The focus was on whether the number of stations is appropriate according to the cycle time, and the task times of different card models were considered separately. The mathematical model was run using Python-Gurobi, and the results were validated by discussing them with the decision-maker.

2.5.5 Heuristic Validation

The team made some trials with the codes for the heuristics to compare two heuristics (RPW and LCR) for card models E5 and E10. The task

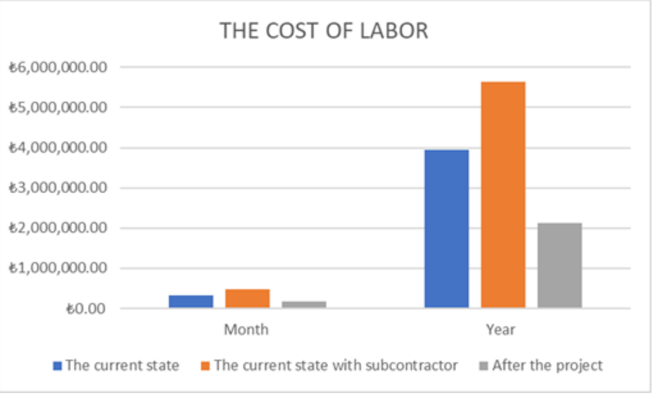


Figure 2.3: Labor cost benefits

times were measured and it was observed that both heuristics give the same results within the card model tasks' times.

2.6 Results and Benefits For The Company

There are 28 people in the card grouping unit, so 14 workers are responsible in each room. After implementing this project, each assembly line will require five people for card grouping processes. In other words, 15 people will be enough. Assuming the minimum wage cost to employers is approximately 11,760 Turkish Liras (based on the KPMG report), there could be about a 46 percent reduction in labor costs. Based on the industrial advisor's opinion, without the need for subcontractors in grouping processes could end up with less need for workers, which is about 15 people. The labor cost could decrease by about 62 percent; see Figure 2.3.

Card grouping currently takes place in two rooms. Additional L-shaped stations in the project will be helpful to get rid of these two rooms shown in

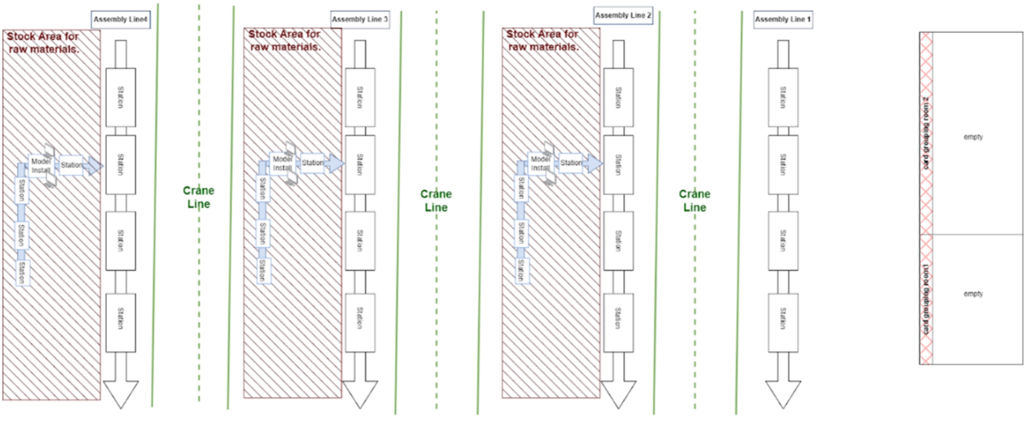


Figure 2.4: Proposed layout plan in detail

Figure 2.4. Stock area for card grouping behind assembly lines is sufficient to satisfy the space need for these workstations when considering what we examined during the factory visit. The saving in space could be equal to 60 square meters in total.

After comparing the current system with the project's outcome through simulation according to the model outlined in Figure 2.5, productivity, and resource utilization will be better, and in-plant transfer times will shrink.

2.7 Conclusion

The aim is to create a plan that enables switching to synchronous manufacturing by adding additional card grouping stations to the assembly lines and providing a schedule of the number of workers needed for each workstation. A user-friendly interface is developed by utilizing tkinter and PySimpleGUI libraries for Python code, allowing users to select one of three solution methods (mathematical model, largest candidate rule heuristic, ranked positional method heuristic) and enter their desired inputs. The interface for our optimization model allows employees to easily enter inputs and receive output

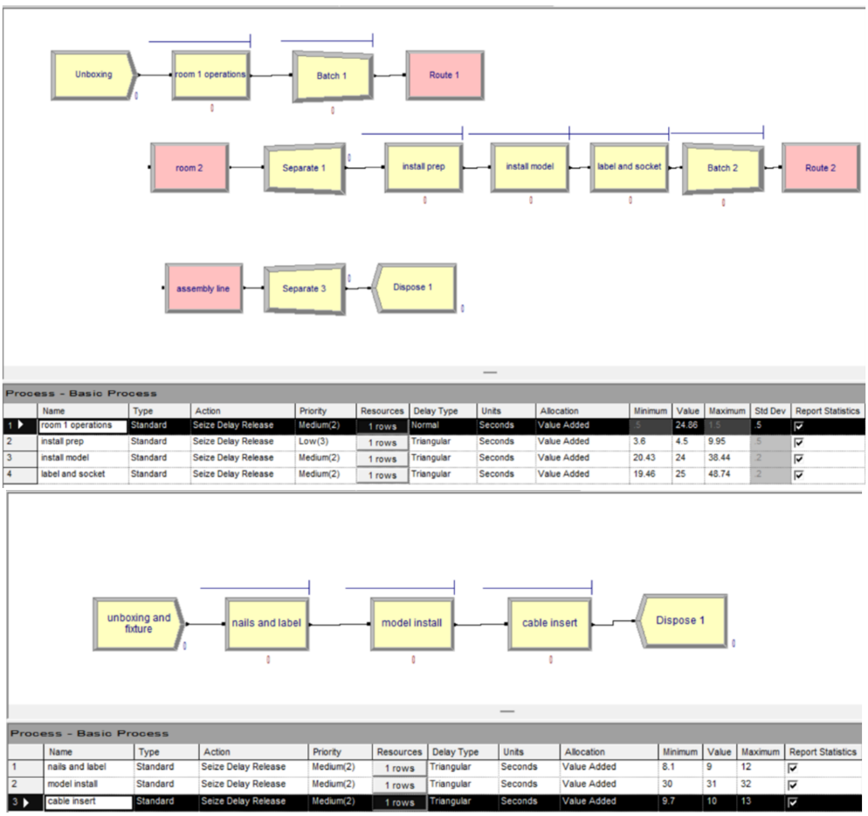


Figure 2.5: Simulation model

without technical knowledge, providing valuable insights for the business.

During the project, meetings are held with advisors to find a suitable place for the card grouping station in the assembly line. We have created a recommended integration plan. In this pilot plan, the environment of the appropriate place will be arranged and the unused items will be removed. Tables for the workers and computers will be rearranged into L shape. Necessary training will be provided to workers. The system will be integrated and final checks will be done to ensure that the system works correctly. Trained workers will be assigned to the stations and the results will be observed. In the future, the L structure should be completely integrated to the current assembly line in order synchronize manufacturing. If any mistakes or errors occur during the integration, modifications based given feedbacks will prevent future mistakes.

Bibliography

Kriengkorakot, N. and N. Pianthong (2007). The assembly line balancing problem: review articles. *Engineering and Applied Science Research 34*(2), 133–140.

3 | Batarya Üretim ve Araç Montaj Hattı Arasındaki Çeşitlilik Yönetimi

Oyak Renault

Proje Ekibi
Zeynep Birsen, Fatih Musab Bozkurt, Arda Doğan,
Gözde Gür, Berkay Kablan, Barçın Ozan, Bora Yıldırım

Şirket Danışmanı
Eyüphan Altunbay
Endüstriyel Tesisler Taahhüt Mühendisi

Akademik Danışman
Dr. Emre Uzun
Endüstri Mühendisliği Bölümü

Özet

Oyak Renault Otomobil Fabrikaları, Clio E-tech araçlarında kullanılan bataryaların üretim planlamasında karma siparişler, hatalı bataryalar ve talep belirsizliği gibi problemler sebebiyle üretimde aksaklık yaşamaktadır. Bu aksaklıkları çözmek için, üretim planlama sürecini optimize eden bir matematiksel model önerilmiştir. Envanter tutma için "tristok" alanı ve bu alandaki envanteri enazlayan bir üretim planı tasarlanmıştır. Üüretim planı, Excel çözücü kullanılarak uygulanmış, farklı senaryolarda doğrulanmıştır. Sonuçlar, modelin batarya üretim planlamasını iyileştirdiğini ve üretim hattındaki kesintileri önlediğini göstermektedir. Çalışmamız Oyak Renault'da batarya üretimi ile araç montajı arasındaki uyumu artırarak fabrikanın genel işletme başarımını geliştirmeye katkıda bulunmaktadır.

Anahtar Sözcükler: Matematiksel Model, Senkronizasyon, Üretim Çizelgeleme, Güvenli Envanter Miktarı

Diversity Management Installation between Battery and Vehicle Assembly Lines

Abstract

Oyak Renault faces challenges in production planning for batteries used in Clio E-tech vehicles due to mixed orders, defective batteries, and demand uncertainty. To address these issues, a mathematical model is proposed to optimize the production planning process. The solution involves implementing a stock area named as tristock area for inventory holding and utilizing a mathematical model to create a synchronized production schedule that minimizes inventory at the tristock area. The solution is implemented using Excel Solver and validated under different scenarios. The results show that the model effectively optimizes the production planning of batteries, reducing the risk of stock-outs and disruptions in the production line. This project contributes to enhancing production efficiency and alignment between battery production and vehicle assembly at Oyak Renault, improving the overall operational performance of the factory.

Keywords: Mathematical model, Synchronization, Production Scheduling, Automotive Industry, Safety Stock

3.1 Company Information

Oyak Renault is an automobile factory established in Turkey in 1969 that produces Renault vehicles, engines, and mechanical parts. The factory has a capacity of 390,000 automobiles and 920,000 engines. The facility manufactures three models: Clio, Clio E-Tech (hybrid), and Megane Sedan. In 2021, Oyak Renault manufactured 248,000 passenger vehicles, accounting for 31.7% of total automobile manufacturing in Turkey and exporting 188,000 of them, making it the leader in automobile exports with a 33.2% share of Turkey's automobile exports (Oyak, 2021).

3.2 System Analysis

The battery production system for Clio E-tech vehicles comprises two main parts: the base battery assembly line and the U-conveyor assembly line. Batteries are produced and inspected in the base battery assembly line, which functions as a single-line system. The batteries are processed in lots of 18 batteries each, and their order is maintained until they reach the inspection and battery charging stage. According to future expectations of the company, these lots will contain mixed types of batteries which will be referred as mixed-lot production. Subsequently, batteries that pass inspection are transferred to the U-conveyor assembly line. At the end of the

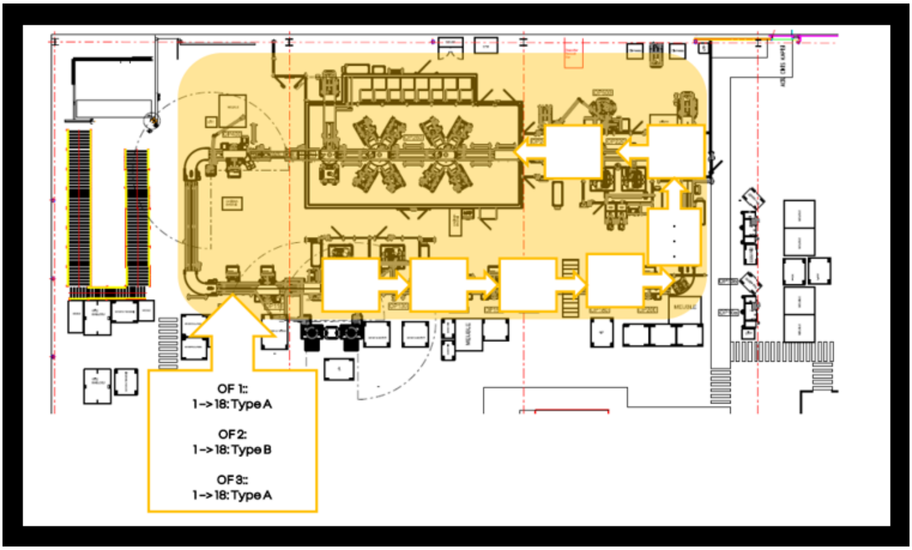


Figure 3.1: Flowchart of the Current Base Battery Assembly Line

U-conveyor, the finished batteries are loaded onto a trolley, which transports them to the vehicle production line. The flowchart of the current base battery assembly line is shown in Figure 3.1

Batteries enter the system as a lot and go through production and inspection processes in a single-line system. After inspection, batteries in good condition move to the U-conveyor assembly line for further processing. However, there are issues with mixed order problems among different battery types, defective batteries, and demand uncertainty, leading to delays in production. Additionally, there are deviations between the planned number of vehicles to be produced and the actual number of vehicles produced, causing further delays. The production plan for vehicles is finalized 3.5 hours before production, but it takes approximately 5 hours for a battery to reach the vehicle production line. To address these problems, a tristock area will be implemented to hold inventory of batteries and prevent delays. The timeline of the process is shown in Figure 3.2. Batteries in the tristock area will be stocked and separated according to their types to ensure alignment with the vehicle production line.

3.3 Problem Definition

The current production system for batteries used in Clio E-tech vehicles is facing delays, mixed order problems, and disruptions due to different battery types, defective batteries, and demand uncertainty. These issues arise from varying charging and inspection durations for each battery, potential errors in batteries making them unusable, and deviations between planned and

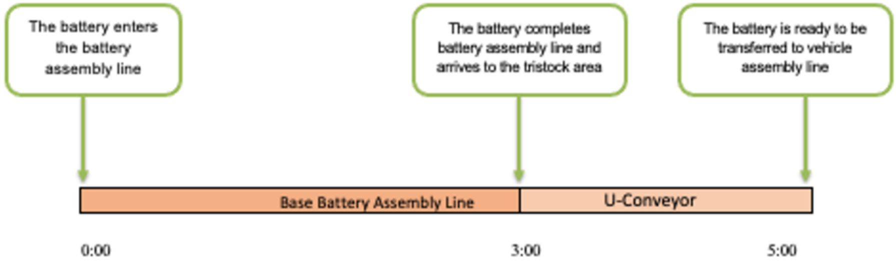


Figure 3.2: Timeline of the Process

actual vehicle production numbers. These issues necessitate improvements to enhance production efficiency and alignment between battery production and vehicle assembly.

3.4 Model

The proposed solution for the battery production planning problem at Oyak Renault is based on mathematical modeling. The mathematical model is designed to create a synchronized production schedule that minimizes the inventory allocated at the tristock area in each period while considering the demand from the vehicle production line and the production capacity of the battery production line. To develop the mathematical model, several assumptions are made based on the given information. These assumptions include the use of aggregate production planning for lot-production, disaggregated production planning for mixed-lot production to avoid mixing different battery types, uniform distribution of daily demand over 24 hours of production, maximum capacity of 320 batteries in the tristock area, no human errors in assembly and transportation processes, and the arrival of a battery to the vehicle production line taking five hours after initialization of its production, among others.

The model integrates the principle of safety stock and follows a continuous review policy. This approach is motivated by the need to prevent stock-outs and interruptions in the production line due to demand and supply uncertainty, as discussed by Gonçalves et al. (2020). The safety stock level is calculated by considering the deviation between planned and actual production amounts, as well as the probability of defective batteries entering the production line. The deviation is fitted to an empirical distribution and a service level of 95% is set. The safety stock is calculated as 0.29 per period, accounting for the probability of 2% defective batteries leaving the system at the charging phase (0.52 per period) and the likelihood of consecutive defects. When the batteries are sent in mixed-lot production the safety stock quantity is determined as five per period (hourly periods), and a portion of the battery production rate is allocated to immediately refill the safety stock. For lot production, the safety stock is recalculated assuming a period of 45 minutes since it takes 45 minutes for a lot of batteries until the end of the assembly line, resulting in a safety stock of four per period.

The main objective of the project is to define a safety stock level and production plan for each type of battery to prevent stock-outs that may interrupt the production line. The mathematical model is designed to minimize the inventory level for each battery type in each period, taking into account the company's expectations and production constraints.

The model is implemented using Excel Solver and tested on three days with maximum demand from the vehicle assembly line. Inputs to the model include daily demand, vehicle demand weight parameter, initial inventory amount, production amounts for prior periods, and compatibility matrix for battery and vehicle types. The approach is motivated by findings of Eppen and Martin (1988) and Takeda-Berger et al. (2022), who discuss safety stock determination and scheduling methods for manufacturing systems. The mathematical model for mixed-lot production and lot-production are shown in Appendices 3.A and 3.B, respectively.

3.5 Validation

The model's validity was evaluated by inputting the vehicle production demand, which was provided by the company, into an Excel solver. The data set for the last year (2022) was provided by the company, and the maximum demand amount in consecutive three days (August 22 - August 24) was extracted from this data set. In the current situation, the company can stock up to 44 batteries, therefore it is assumed that the tristock capacity is 44 in the validation. Assuming a maximum vehicle production of 402 vehicles per day and a uniform distribution of demand over a day at a rate of 17 vehicles per hour, it was determined that 397 batteries were produced per day, which was sufficient to meet the demand.

In this scenario, the maximum level of inventory in the tristock area reached 18. It was observed that with the provided schedule of vehicle production, the tristock area was able to provide enough capacity, including

	Satisfied Demand Rate	Does the Production Stops?
3-Day Data	100%	No
Maximum Demand from Only One Type	100%	No
Equal Units of Demand from Each Type	100%	No

Figure 3.3: Validation Scenarios

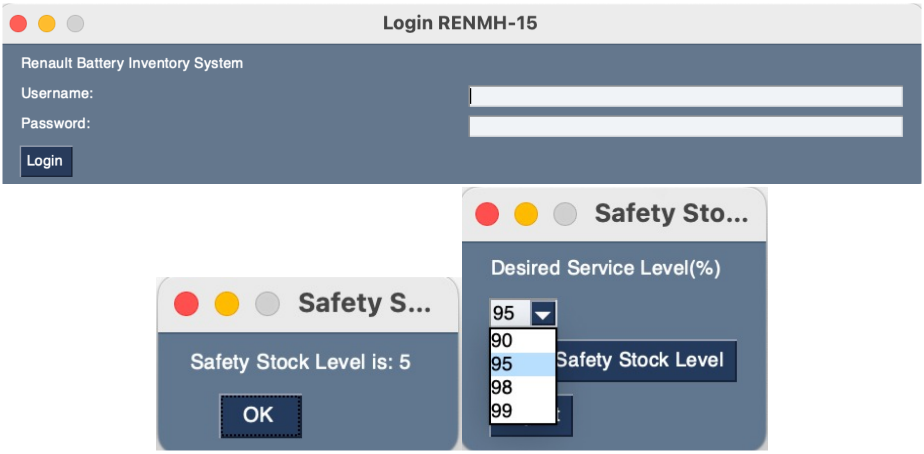


Figure 3.4: User interface: service level option (bottom left), corresponding inventory level

the safety stock level. The validation scenarios can be seen in Figure 3.3.

3.6 User Interface

The outcomes of the project include the development of an inventory policy that incorporates safety stock to meet various demand patterns and prevent stockouts. Decision support tools for stock level regulation and dynamic stock tracking are also implemented. A user interface using Jupiter is created to display the results of the battery production planning model and facilitate production planning in sync with the vehicle assembly line.

Users access the user interface through a login process using their username/email and password, and then choose from three options: battery production plan, safety stock level, and change parameters. In the battery production plan option, users input initial inventory and production quantities for each battery type for the first periods, and then upload an Excel file containing total demand data. The output, which is the battery production plan for each period, is displayed, providing users with a comprehensive overview of their production strategy.

In the safety stock level option, users can select the desired service level and input the defect rate of batteries in the production line. The system then calculates the appropriate safety stock level for each battery type, taking into consideration the desired service level and defect rate. This helps users ensure that they have sufficient inventory buffers to meet demand fluctuations and maintain customer service levels. The service level option and corresponding safety stock level is shown in Figure 5 and Figure 6, respectively.

The change parameters option allows users to make adjustments to bat-

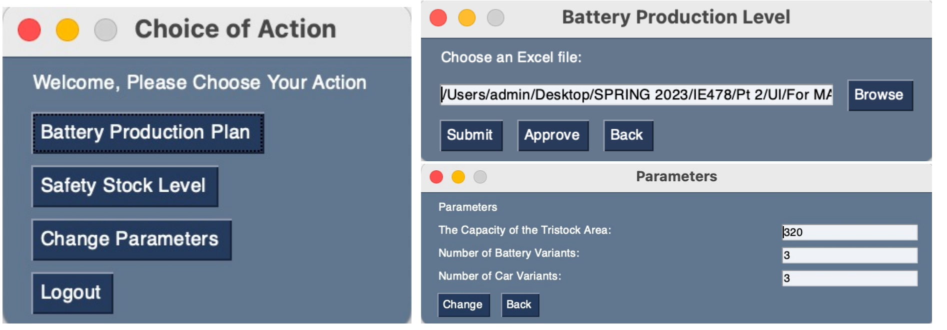


Figure 3.5: Choice of action, data input, parameter control page

tery or vehicle quantities, or capacity based on their specific requirements. This feature aligns with the mathematical model of mixed lot production, enabling users to fine-tune their production plans according to changing business needs, production constraints, or other factors. Figure 3.5 shows the related pages of User-Interface. Regular reviews and updates of the model and user interface may be necessary to ensure accuracy and effectiveness in supporting production planning decisions.

Additionally, training and support will be provided to ensure proper utilization of the user interface and effective decision-making based on the outputs generated by the mathematical model.

3.7 Pilot Study

In order to integrate our approach into the company's existing system, a pilot study is being conducted at a time that is convenient for the company. Prior to the pilot study, the user interface is set up, and an explanatory booklet is provided to assist with implementation and ensure ease of use. The company is asked to decide on the starting inventory level, production amount, and vehicle production amount. The user interface takes the information from the formatted Excel files provided by the company to ensure that the production plan is in sync with the data. Upon completion of the data entry process, the setup process is finalized, and the product is prepared for the pilot study.

This pilot study aims to test two main points. Firstly, the user interface is observed to ensure that it is working as planned and able to properly utilize the model with the given data. The company ensures that the data is inputted correctly to observe this. Secondly, the efficiency of the production plan provided is checked through observation by the company. Even though the current battery production cannot be synchronized with the vehicle assembly line, with the given tristock inventory level, synchronization of the vehicle assembly line and the battery production line is expected to be reached. The company is asked to conduct the pilot study for a full day to sufficiently observe the implementation.

During the pilot study, regular reviews and updates of the model and user interface are conducted to ensure accuracy and effectiveness in supporting production planning decisions.

3.8 Benefits to the Company

The proposed model aims to minimize stock levels and synchronize the battery and vehicle production lines, with a preference for the stock level as the ideal key performance indicator (KPI). The construction of the tristock area has increased the available stock area by eight times, enabling uninterrupted production and improved production line synchronization.

Implementation of the production plan utilizing planned production data has resulted in a 4.1% improvement in the battery production line's capacity to satisfy demand, from 552 vehicles per day to 576 vehicles per day. The user interface provided with the model allows for better visibility and control over the production process, leading to improved decision-making and production planning.

In addition to immediate benefits, the proposed model has the potential to support the company's future plans for managing variety in production, as it can facilitate coordination and planning for multiple battery types. The potential benefits in terms of improved stock level management, production synchronization, and production flexibility can result in significant cost savings and increased operational efficiency.

3.9 Conclusion

In conclusion, the proposed solution for battery production planning at Oyak Renault, based on mathematical modeling, is designed to address the challenges of managing stock levels, demand uncertainty, and production synchronization in the battery production line. The model aims to minimize inventory levels, prevent stock-outs, and ensure uninterrupted production in the tristock area. By considering factors such as demand, production capacity, and safety stock requirements, the model provides a synchronized production schedule that optimizes production efficiency. The implementation of the production plan based on the proposed model has already yielded positive outcomes, including a 4.1% improvement in the battery production line's capacity to satisfy demand. The user interface provided with the model has also improved visibility and control over the production process, leading to better decision-making and production planning. Furthermore, the proposed model has the potential to support future plans for managing variety in production. By improving stock level management, production synchronization, and production flexibility, the model has increased operational efficiency. Overall, the proposed solution based on mathematical modeling offers a promising approach to tackle the battery production planning problem at Oyak Renault, providing tangible benefits in terms of improved production efficiency, stock level management, and production line synchronization.

Bibliography

Eppen, G. D. and R. K. Martin (1988). Determining safety stock in the presence of stochastic lead time and demand. *Management Science 34*(11), 1380–1390.

Gonçalves, J. N., M. Sameiro Carvalho, and P. Cortez (2020). Operations research models and methods for safety stock determination: A review. *Operations Research Perspectives 7*, 100164.

Oyak (2021). Oyak - 2021 faaliyet raporu. https://www.oyak.com.tr/sites/1/upload/files/__OYAK_FR21TR_OK_WEB-685.pdf. Accessed: 2022-11-16.

Takeda-Berger, S. L., Ícaro Romolo Sousa Agostino, M. R. F. da Silva, and E. M. Frazzon (2022). Reactive production scheduling approach based on inventory availability. *IFAC-PapersOnLine 55*(10), 418–423. 10th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2022.

Appendices

3.A Mixed-lot Production Model

The sets are I = Battery types, J = Vehicle types, T = Periods. The parameters are

$$
\begin{aligned}
Acap_i &= \text{Tristock capacity for battery type } i \text{ less than or equal to 320} \\
Bcap &= \text{Battery production capacity} \\
y_{ij} &= \begin{cases} 1, & \text{if battery } i \text{ is compatible with car } j \\ 0, & \text{otherwise} \end{cases} \\
I_i &= \text{Initial inventory level for battery type } i \\
D_{jt} &= \text{Demand of vehicle type } j \text{ in time } t.
\end{aligned}
$$

Decision Variables are

$$P_{it} = \text{Produced amount of battery type } i \text{ in time } t$$

$$I_{it} = \text{Inventory level of battery type } i \text{ in time } t$$
$$G_{ijt} = \text{Number of battery type } i \text{ produced for vehicle type } j \text{ in time } t$$

Finally, the model is to

$$\text{minimize} \qquad \sum_{i \in I} \sum_{t \in T} I_{it}$$

subject to

$$
\begin{aligned}
& I_{it} \leq Acap_i && \forall i \in I, t \in T \\
& \sum_{i \in I} P_{it} \leq B_{cap} && \forall t \in T \\
& I_{it+3} = I_{it+2} - \sum_{j \in J} G_{ijt+5} + P_{it} && \forall i \in I, j \in J, t \in T \\
& \sum_{i \in I} \sum_{j \in J} y_{ij} Z_{it} \leq 1 && \forall t \in T \\
& \sum_{i \in I} y_{ij} G_{ijt} \leq D_{jt} && \forall j \in J, t \in T \\
& I_{it} \geq 0 && \forall i \in I, t \in T \\
& P_{it} \geq 0 && \forall i \in I, t \in T \\
& G_{ijt} \geq 0 && \forall i \in I, j \in J, t \in T
\end{aligned}
$$

3.B Lot Extension

The sets are $I =$ Battery types, $J =$ Vehicle types, $T =$ Periods. The parameters are

$$Acap_i = \text{Tristock capacity for battery type } i \text{ less than or equal to } 320$$
$$Bcap = \text{Battery production capacity}$$
$$y_{ij} = \begin{cases} 1, & \text{if battery } i \text{ is compatible with car } j \\ 0, & \text{otherwise} \end{cases}$$
$$I_i = \text{Initial inventory level for battery type } i$$
$$D_{jt} = \text{Demand of vehicle type } j \text{ in time } t.$$

Decision Variables are

$$P_{it} = \text{Produced amount of battery type } i \text{ in time } t$$
$$I_{it} = \text{Inventory level of battery type } i \text{ in time } t$$
$$G_{ijt} = \text{Number of battery type } i \text{ produced for vehicle type } j \text{ in time } t$$
$$Z_{it} = \begin{cases} 1, & \text{if battery } i \text{ is produced in time } t \\ 0, & \text{otherwise.} \end{cases}$$

Finally, the model is to

$$\text{minimize} \qquad \sum_{i \in I} \sum_{t \in T} I_{it}$$

subject to

$$
\begin{aligned}
& I_{it} \leq Acap_i && \forall i \in I, t \in T \\
& \sum_{i \in I} P_{it} \leq B_{cap} && \forall t \in T \\
& I_{it+4} = I_{it+3} - \sum_{j \in J} G_{ijt+6} + P_{it} && \forall i \in I, j \in J, t \in T \\
& \sum_{i \in I} \sum_{j \in J} y_{ij} Z_{it} \leq 1 && \forall t \in T \\
& \sum_{i \in I} y_{ij} G_{ijt} \leq D_{jt} && \forall j \in J, t \in T \\
& P_{it} \leq 18 Z_{it} && \forall i \in I, t \in T \\
& I_{it} \geq 0 && \forall i \in I, t \in T \\
& P_{it} \geq 0 && \forall i \in I, t \in T \\
& G_{ijt} \geq 0 && \forall i \in I, j \in J, t \in T \\
& Z_{it} \in \{0, 1\} && \forall i \in I, t \in T
\end{aligned}
$$

Buzdolabı Taleplerinin Üretim Tesislerine Dağılımı | 4

Arçelik Buzdolabı İşletmesi

Proje Ekibi
Ali Bingöl, Sena Aslı Bozkurt, Kaan Çakıroğlu, Ece Efeoğlu,
Mustafa Günal, Damla Kırdım Nazife İlkin Uslu,

Şirket Danışmanı
Yaşar Semih Tolukan
Metot Mühendisi

Akademik Danışman
Dr. Emre Uzun
Endüstri Mühendisliği Bölümü

Özet

Arçelik A.Ş. tüm talepleri mümkün olan en düşük lojistik maliyetlerle karşılamak için farklı buzdolabı modellerinin üretimini üretim tesislerinden dünya çapındaki sevkiyat noktalarına dağıtmayı amaçlamaktadır. Şirketin mevcut uygulaması, ağırlıklı olarak uzman görüşlerine dayandığından optimum çözümü garanti etmemektedir. Bu projede, talep dağılımı problemini etkin şekilde çözecek matematiksel modele sahip, kullanıcı dostu bir karar destek sistemi geliştirilmiştir. Bu karar destek sistemi, bir kullanıcı arayüzü içinde şirketin optimum talep dağılımını en az zaman ve insan kaynağı ile elde etmesini sağlamaktadır. Aynı girdi verileri ve koşullar sağlandığında Arçelik'in talep dağılımı maliyetinde %10,62'lik bir iyileşme olduğunu göstermektedir.

Anahtar Sözcükler: Talep Dağılımı, üretim planlama, lojistik maliyet, lojistik eniyileme, kapasite kullanım oranı, karar destek sistemi.

Allocation of Forecasted Refrigerator Demand to Production Facilities

Abstract

Arçelik A.Ş. seeks to allocate the production of different refrigerator models from production facilities to shipment destinations across the globe to satisfy all demands with the lowest possible logistics costs. The company's current application does not ensure optimal solution since it relies heavily on expert opinions. In this project, a decision support system with a mathematical model to solve the demand allocation problem in an efficient and user-friendly manner is developed. This decision support system provides an optimal demand allocation to the company within a user interface and shows a 10.62% improvement in Arçelik's demand allocation cost with the same input data and conditions.

Keywords: Demand allocation, production planning, decision support system, minimum logistics cost, assignment, networks.

4.1 Company Description

Arçelik A.Ş. is a white goods company established in 1955 in Turkey. With a turnover of 68.2 billion TL in 2021 and more than 40,000 employees worldwide, the company is the most prominent white goods company in Turkey and the second largest in Europe in its sector (Arçelik, 2022).

4.2 Current System and Problem Definition

Arçelik sells different types of refrigerators to more than 100 countries, with production facilities in nine countries globally. To be satisfied, the demands of each product type from different countries are allocated to production facilities where the demanded products are produced. This allocation is called demand allocation. To obtain a five-year demand allocation plan, they analyze the current sales data and develop a demand forecast for the next five years. Afterwards, they allocate the demands among different production facilities for each product type for the following five years. The current decision methodology to allocate the demands among production facilities is based on expert opinion. The total demand allocation cost and utilization rates of the production facilities are the crucial key performance indicators (KPI) that experts consider while making decisions. Furthermore, it is undesirable for the utilization rate of production facilities to fall below a specific limit. The company seeks to reach the minimum possible cost while satisfying the demand and keeping capacity utilization over a certain threshold. In addition, each production facility has constraints such as capacity, limited flexibility, and minimum utilization rate to produce different product types.

The problem is based on allocating refrigerator demands to the production facilities and the company's capacity planning. The project aims to allocate five-year product demands to production facilities with the minimum possible cost by considering the constraints.

Hence, the problem can mainly be defined as the lack of a system that follows production capacity, provides minimum demand allocation costs, and suggests capacity increases in production facilities in case of an insufficient production capacity to satisfy all demands. In this project, a decision support system that guarantees the feasibility and optimality of the demand allocations is developed to minimize the logistics, production, and capacity increase costs while keeping the utilization rates of the production facilities above a desired threshold and preventing unsatisfied demand by considering possible capacity increase decisions.

4.3 Proposed Solution Strategy

4.3.1 Critical Assumptions

Producing different product types in different production facilities has different unit production costs. Products are shipped to various countries from different production facility. Shipping has a logistics cost per unit for each product type composed of transportation and customs duty. However, this logistics cost is assumed to be zero if a product is shipped to a place in the same country as it was produced.

Moreover, these logistics costs per unit are obtained from the snapshot of the logistics cost data of a particular time of the year, annually for each product type, and they are constant. They are calculated for one specific transportation type and route only, and those cannot be changed. Therefore, from production facilities to demand points the products are shipped, the logistics cost per unit for each product type is assumed to be constant regardless of the quantity.

There are multiple types of products that Arçelik produces. However, each type requires different configurations in the facilities. The available configurations to produce different product types are assumed to be fixed and cannot be changed.

All production facilities have capacities that denote the maximum number of products they can produce annually for each product type. These capacities are assumed to be fixed to the figures that are provided by the company and can not be changed unless a capacity increase decision is made.

It is assumed that the decision support system provided will be used annually for the next five years. The demand allocation obtained for a given year will change the input data for the next years. In other words, updated input data will be inherited from previous years for each year.

4.3.2 Major Constraints

Arçelik operates multiple production facilities that can only ship products to specific countries, based on predefined shipment routes. Only these routes must be used and each production facility is capable of producing only a specific set of product types. Additionally, the number of products to be produced for each product type is subject to a capacity constraint in each production facility and the utilization level of each production facilitiy should not be under its specified threshold. Furthermore, Arçelik must meet the demands for different product types in all demand points.

As a result, it is necessary to ensure that all demand is satisfied, while taking into account the production limitations and utilization level requirements of each production facility and the specific shipment routes available.

4.3.3 Objectives

The project's main aim is to minimize total costs consisting of logistics, production, and capacity increase costs by allocating the demand of different product types among production facilities to be satisfied.

Another objective of our project is to hold the capacity utilization of the facilities above a certain threshold. Although it is not possible to change the capacities without a capacity increase, it is possible to adjust their utilization by demand allocation and satisfy the utilization requirements of the company.

4.3.4 Solution Approach and Method

To solve the problem of minimizing cost while ensuring demand satisfaction and meeting utilization limitations, a mathematical model is used. SciPy is selected as the optimization framework in Python to solve the model. The system is parametric and can be used in different scenarios as long as the input data is in the required format, allowing the company to utilize it in case of any future change in production flexibility or capacity of production facilities.

4.3.5 Conceptual Model

The system requires nine main inputs, including forecasted demands, logistics costs, production costs, capacity increase costs, production capacities, production availability matrix, shipping availability matrix, capacity increase upper bounds, and utilization level. Furthermore, optional inputs include capacity increase investment horizon and expected interest rate to calculate the annual cost of capacity increases without dominating the other costs. Additionally, capacity changes from the previous year are inherited for the upcoming years.

The system determines and returns the number of a product type produced in each production facility and transported to a demand point, as well as whether there is a capacity increase decision for a product type in a production facility or not, and the additional number of a product type that can be produced in a production facility after a capacity increase decision. The representation of the entire system can be seen below.

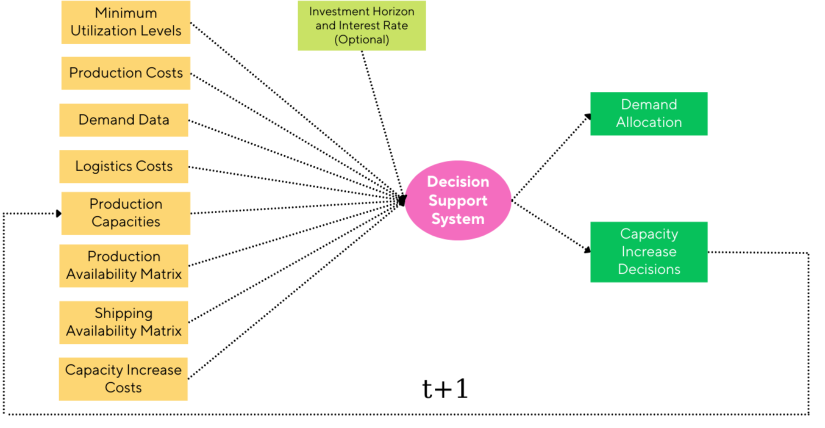


Figure 4.1: Representation of the System

4.3.6 Mathematical Model

Parameters:

F : Set of production facilities

T : Set of product types

D : Set of demand points (countries)

U_i : Minimum required utilization rate of production facility $i \in F$

c_{ij}^p: Unit cost of transporting product type p from production facility i to demand point j, $i \in F, j \in D, p \in T$

t_i^p : Production capacity of facility i for product type p, $i \in F, p \in T$

$TC_i = \sum_p t_i^p$: Total production capacity of facility i, $i \in F$

d_j^p : Demand of demand point j for product type p, $j \in D, p \in T$

$$e_{ij}^p: \begin{cases} 1, & \text{if there is a route to demand point } j \text{ from production facility } i \\ & \text{for product type } p,\ i \in F, j \in D, p \in T \\ 0, & \text{otherwise} \end{cases}$$

$c_i'^p$: Capacity increase fixed cost of increasing the production capacity for product type p in production facility i, $i \in F$, $p \in T$

v_i^p: Capacity increase variable cost of increasing the production capacity for product type p in production facility i, $i \in F$, $p \in T$

S_i^p: Upper bound of capacity increase for product type p in production facility i, $i \in F$, $p \in T$

pc_i^p : Production cost of production facility i for product type p, $p \in T$, $i \in F$

f: Interest rate

N: Investment horizon in years

$CRF = \frac{[i(1+i)^N]}{[(1+i)^N-1]}$: The capital-recovery factor that annualizes capacity increase fixed and variable costs

Decision Variables:

x_{ij}^p: Number of product type p to be produced and transported from production facility i to demand point j, $i \in F, j \in D, p \in T$

$$Z_i^p: \begin{cases} 1 & \text{if there is an capacity increase decision in the production} \\ & \text{facility } i \text{ for product type } p,\ i \in F, \\ 0 & \text{otherwise,} \end{cases}$$

$t_i'^p$: Additional number of product type p that can be produced in production facility i after a capacity increase decision. $i \in F, p \in T$

Model:

$$\text{Minimize} \sum_{i,j,p} (c_{ij}^p + pc_i^p)x_{ij}^p + (c_i'^p z_i^p + v_i^p t_i'^p)CRF$$

$$\text{subject to} \sum_j \sum_p \frac{x_{ij}^p}{TC_i} \geq U_i \qquad \forall i \in F \qquad (4.1)$$

$$\sum_j x_{ij}^p \leq t_i^p + t_i'^p \qquad \forall i \in F, \forall p \in T \quad (4.2)$$

$$\sum_i x_{ij}^p \geq d_j^p \qquad \forall j \in D, \forall p \in T \quad (4.3)$$

$$x_{ij}^p \leq (t_i^p + t_i'^p)e_{ij}^p \qquad \forall i \in F, \forall j \in D, \forall p \in T \quad (4.4)$$

$$t_i'^p \leq S_i^p Z_i^p \qquad \forall i \in F, \forall p \in T \quad (4.5)$$

$$x_{ij}^p \in \mathbb{Z}^+ \qquad \forall i \in F, \forall j \in D, \forall p \in T \quad (4.6)$$

$$t_i'^p \in \mathbb{Z}^+ \qquad \forall i \in F, \forall p \in T \quad (4.7)$$

$$Z_i^p \in \{0, 1\} \qquad \forall i \in F, \forall p \in T \quad (4.8)$$

As mentioned before, the objective function minimizes the total demand allocation cost that consits of logistics, production, and capacity increase costs. Constraint (4.1) ensures that all production facility utilization rates are higher than the minimum required utilization rate. Constraint (4.2) ensures that the production of any product type does not exceed the initial capacity plus the additional capacity obtained by any capacity increase decision made. Constraint (4.3) ensures that all demands are satisfied. Constraint (4.4) ensures that no products can be shipped if it is not possible to send product type p from country i to country j. Constraint (4.5) ensures that additional capacity variables $t_i'^p$ must be zero if no capacity increase decision is made in production facility i for product type p. This constraint also ensures that the capacity increase does not exceed the capacity increase upper bound if any capacity increase decision is made. Remaining three constraints (4.6,4.7,4.8) ensures that x_{ij}^p and $t_i'^p$ are nonnegative integers, while Z_i^p is binary.

4.4 Validation

The model needs to yield applicable results with real data to show that our solution is acceptable for Arçelik and our main assumptions are reasonable. To observe the model's credibility and applicability in the company, 2022's demand allocation provided by the company is compared with our model's results using the same data and parameters. Along with our industrial advisor, we confirmed that with our solution methodology the production wouldn't get interrupted and hence our solution was valid. In Figure 4.2,

Facility	H4 Type	Country Shipped	Arçelik 2022	Decision Support System
C114	T3	C84	2163	2163
C131	T2	C10	4862.909857	0
C28	T2	C10	2438.834256	7302

Figure 4.2: Allocation Comparison

example allocations of the company and our decision support system is demonstrated. As it can be seen, in the first row our allocation matches with the company's allocation. On the other hand, there is a difference in the demand allocation of the company and our system. In our allocation, for product type T2, all 7,302 products are sent from production facility C28 whereas in the company's allocation 4,863 of them were sent from C131 and 2,439 of them were sent from C28 since it is cheaper to send product type T2 16 from C28 to C10 than sending them from C131 to C10. The behavior shown in this example exists in general comparison of allocation decisions between Arçelik and our decision support system. Hence, our model yields applicable solutions in line with Arçelik's expectations and past decisions while improving the demand allocation. In addition, when we examine the 2022 allocations of the company, the capacities were not exceeded, and the demands were met, indicating that the company's allocations for 2022 are also feasible for our model.

4.5 Implementation and Pilot Study

For the pilot study, the decision support system is sent to Arçelik as an executable file. After the security checks conducted by the company, the decision support system is tested using a sample data and results are observed. Based on this observation Arçelik requested a revision in the user interface output. The user interface is edited regarding the feedbacks from the company and the decision support system is sent to Arçelik as new executable file with a modified outputs.

After the pilot study, the decision support system is ready to be used by Arçelik with their forecast demand data as input to obtain a demand allocation. How to use the decision support system, how to input the data, and how to interpret the results are clearly explained in the user manual. Briefly, to initiate the system, the GBTD.exe file, provided by the group, must be launched. To utilize the system, a compatible input file in .xlsx format must be chosen. Once the input file is chosen, users are to click on the "Allocate" button located at the bottom of the screen, to select optional inputs for investment return period, interest rate, and years for demand allocation. The demand allocation outputs and capacity increase recommendations are available in an Excel file in the same folder as the decision support system. The system offers an executive summary page that includes essential data such as costs, production per production facility and utilization levels.

4.6 Benchmarking and Benefits

We compared the demand allocations of 2022 decided by Arçelik with the results of our model using the same inputs, regarding the KPI values are set by the company as total demand allocation cost and utilization rates of the production facilities. With the decision support system we provide, the total cost of demand allocation for 2022 can be made 10.62% less costly without any capacity increase decision or utilization level constraint. A trade-off exists between the total allocation cost and the minimum production facility utilization rate. We conducted a demand allocation cost-minimum utilization level trade-off analysis. Since the lowest utilization level is 50.1% in our

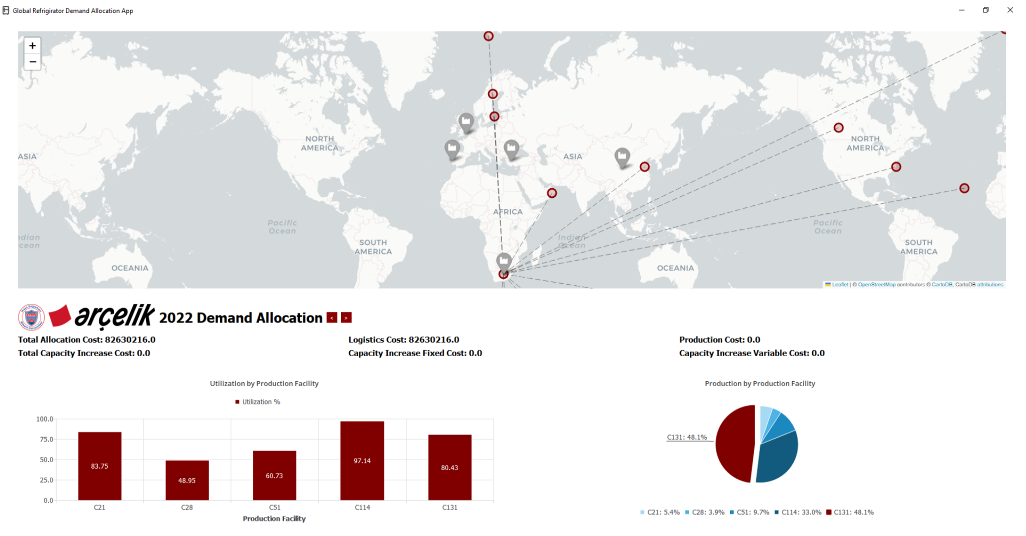


Figure 4.3: Result Summary Page of the User Interface

solution with a minimum utilization level of zero, we analyzed the trade-off at different utilization levels of 60%, 70%, and 80%. As we observed, demand allocation cost increases as the utilization rate increases since the demand allocation minimization problem gets more restricted. In addition to improving the demand allocation cost by 10.62% with the same input data, our model obtains a demand allocation with a significantly lower cost than Arçelik, even with an 80% minimum utilization level constraint. The trade-off between the total demand allocation cost and the utilization requirement can be seen in Figure 4.4.

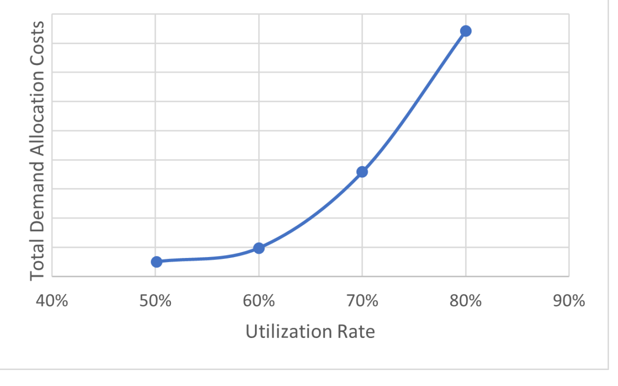


Figure 4.4: Pareto Frontier of Cost vs Utilization

As for the other benefits to the company, the decision support system provides a user-friendly interface and optimization time of approximately three minutes per year allocated. The system allows for easy modification of the inputs. Therefore, the company will be able to handle changes rapidly. Additionally, the system enables the company to conduct allocation cost-utilization level trade-off analysis by altering various parameters. The user interface and short optimization time ensure that this process is not labor and time intensive.

4.7 Conclusion

To conclude, our decision support system considers demand data, logistics costs, production and capacity increase costs, capacities, production availability matrix, shipping availability matrix, and minimum required utilization rate; and decides the demand allocation with minimum cost and balanced utilization between the production facilities. The project's main aim, an optimal demand allocation plan in a shorter time, is satisfied with credible results with the real-life data provided by Arçelik.

Bibliography

Arçelik (2022, March). Arçelik, 2021 annual report. https://www.arcelikglobal.com/media/6798/arcelik21_faaliyet_raporu.pdf.

Ciro ve Pazar Payı Odaklı Saha Satış Rotası Planlama Karar Destek Sistemi | 5

Solvoyo

Proje Ekibi

Deniz Dalgıç, Buse Erdoğan, Zeyneb Gülhan, Ahmet Talha Hamarat, Doğukan Kalaycıoğlu, Abdusselam Laçin, Pınar Yıldırım

Şirket Danışmanı

İshak Memigüven
Otomasyon ve Bilgi Sistemleri Yöneticisi

Akademik Danışmanlar

Prof. Dr. Ülkü Gürler
Endüstri Mühendisliği Bölümü
Doç. Dr. Emre Berk
İşletme Bölümü

Özet

Solvoyo'nun bir müşterisinin geleneksel satış kanallarındaki taleplerinin karşılanmasını sağlayan bir karar destek sistemi tasarlanmıştır. Matematiksel model ile satış aracına yüklenmesi gereken ürün miktarları, uğranılacak satış noktalarının rotası ve gidilen satış noktasında bırakılacak ürün miktarı kararları verilerek elde edilen ciroyu ve belirli ürünlerin raftaki görünürlüğünü eniyilemek hedeflenmiştir. Model, takım yön bulma probleminin gerekirci uyarlaması olarak yazılmıştır. Algoritmanın kullanımı için bir arayüz geliştirilmiştir. Sonuçlar güncel kullanılan strateji ile kıyaslandığında, ciroda %5,66'lık bir artış gözlemlenmiştir.

Anahtar Sözcükler: Sıcak satış, ciro eniyileme, rotalama, ürün görünürlüğü, karar destek sistemi.

Sales Route Planning Decision Support System Based on Revenue and Market Share

Abstract

A decision support system has been created for a client of Solvoyo that enables the demands of traditional sales channels to be met. The mathematical model prepared, it is aimed to optimize the revenue and the visibility of certain products in the market by making decisions about the number of products that should be loaded into the sales vehicle, the route of the sales points to be visited and how many products should be left at the sales point. The model is written as a deterministic version of the Team Orienteering Problem (TOP). A user interface has been designed for the use of the algorithm. The proposed solution increased the revenues by 5.66%.

Keywords: Traditional sales, revenue optimization, routing, product visibility, decision support system.

5.1 Company Information

Solvoyo is a software development company that offers an end-to-end supply chain planning and analytics platform with machine learning technology. Solvoyo's main concerns are analyzing problems experienced by its clients and seeking automated solutions to these problems. The scope of the services provided can be listed as follows: retail, demand, supply, inventory, transportation, sales and operations planning, supply chain analytics and visibility, strategic network design, and SaaS. The client base of Solvoyo consists of internationally renowned companies such as Unilever, Düzey, A101, Defacto, Penti, P&G, Vestel, and Mopaş. They provide services for apparel, grocery, e-commerce, q-commerce, and consumer packaged goods industries.

5.2 System Analysis and Problem

5.2.1 Current Sales Network

The current setting of a retail sales network area can be defined as replenishing the inventories of sales locations on a given network from a single supplier warehouse. In the current status of the traditional sales channels, since these sales points do not have an adapted inventory control management system, it is mostly left to the initiative of the salesperson to determine the number of the SKUs that will be dropped to these specific sales points. Here, the personal opinions and decisions of the salesperson are the biggest factors that negatively affect the process. This results in sub-optimal allocations of the SKUs. The primary goal of Solvoyo is to lessen the influence of individual preferences when deciding how to distribute the SKUs.

5.2.2 Details About the System

Within the scope of this project, it was requested to create a decision support system suitable for client Düzey. In Düzey, product distribution is made with one type of truck. The total truck capacity is 800 kilograms, so 480 kilograms is filled with cold-frozen products and 320 kilograms with dry products. Düzey stocks its inventory in the main warehouse, where it is used as the first filling point of the trucks and the termination point of the trucks' routes. The total number of customers connected to this main warehouse is 89. In practice, fulfilling the entire available shelf space of a sales point that is allocated to an SKU would make sense as a diminished shelf space suggests high demand for that specific SKU for this specific sales point. This is where constraints such as truck capacity, the financial standing of this sales point, and the demand distribution/current shelf status of other sales points came into play since these are all binding factors that could suggest a sub-optimality for doing this specified event. This allocation could result in the salesperson being unable to give a specific SKU quantity to other points in their route, which would prohibit a stockout until the next shipment. In addition to inventory control management, the routing of trucks is also an important aspect of the current system. These trucks exit the supplier warehouse, each assigned a specific route consisting of sales points, distribute the respective allocated SKU quantities to each sales point, and then come back to the supplier warehouse at the end of their shift.

5.2.3 Problem Definition

The problem scope can be divided into two main parts such as deciding the SKU portfolio for each customer and determining the optimal route for any delivery day. Deciding SKU portfolio is required in terms of providing information for Solvoyo to know how many of each SKU should be loaded while considering the truck's capacity. SKU portfolio is shaped according to the customer's demand. There are different decision mechanisms for customers in terms of the quantity of stock and the amount of capacity allocated to that SKU. Determining the optimal route for delivery scheduling is another scope of our problem that prioritize Solvoyo's current objectives which are maximizing revenue and increasing some of the product's visibility in the market. The problem also requires to be considered through some restrictions, which are limited working hours per day, limited loading capacity of trucks, and selling the product to the appropriate sales locations while considering penetration, which is the minimum quantity satisfying product's visibility on the shelf, and customers' financial status. All decisions are taken by the salesperson, which reduces the efficiency of the process.

5.3 Mathematical Model

For the solution of the current problem, a routing model that maximizes the revenue, starts the route from the warehouse and finishes in the warehouse and can visit the same market more than once is needed to adhere to the constraints. When a literature review is conducted, Team Orienteering Problem (TOP) has been found to be the most suitable solution method. The model is written as the Team Orienteering Problem with position-dependent rewards (TOP-PDR) (Panadero et al., 2022).

5.3.1 Critical Assumptions

There is never stock-out in the warehouse. The duration between two points is considered deterministic based on the distance. The loading process in the warehouse will be done at most once a day and takes an average of one hour. Logistic expenses are not considered. Which day of the week the products are delivered does not matter for the sales points. One truck is used for each day.

5.3.2 Parameters and Decision Variables

Parameters

$$
\begin{aligned}
P_{iu} &= \text{Price of SKU } u \text{ in store } i, i \in I, u \in U \\
W_u &= \text{Weight of SKU } u, u \in U \\
D_{ij} &= \text{Distance between customer } i \text{ and customer } j, i \in I, j \in I \\
Pen_{iu} &= \text{Penetration demand of SKU } u \text{ in store } i, i \in I, u \in U \\
Dem_{iu} &= \text{Total demand of SKU } u \text{ in store } i, i \in I, u \in U \\
X &= \text{Capacity of dry products of the vehicle} \\
Y &= \text{Capacity of refrigerated products of the vehicle} \\
T &= \text{Total available time for transportation in one day} \\
F_i &= \text{Financial standing/budget of store } i, i \in I \\
N &= \text{Total number of market} \\
MR &= \text{Minimum revenue for one truck}
\end{aligned}
$$

The demands of the sales points (Dem_{iu}) are determined on a weekly basis, but the routes are created daily. Therefore, on the first day of the week, the model is solved with total weekly demands. The products delivered on the first day are subtracted from the total demand and the demands of the next day are found. In this way, the demand parameter is updated every day by calculating the remaining demand.

Decision Variables

$$A_{ijk} = \begin{cases} 1, & \text{if vehicle } k \in K \text{ goes to store } j \in I \text{ from store } i, \\ 0, & \text{otherwise} \end{cases}$$

$$z_i = \begin{cases} 1, & \text{if } \sum_{j=1}^{I} \sum_{k=1}^{K} A_{ijk} \geq 1 \\ 0, & \text{otherwise} \end{cases}$$

$$B_{iuk} = \text{Amount of SKU } u \in U \text{ dropped to } i \in I, \text{ by } k \in K, i \in I$$

$$u_i = \text{Rank of each node } i \in I \text{ in order of visits}$$

The model decides how many of each product to put in the truck at the beginning of the day, the route the truck will follow, and how many of each product to leave at each sales point.

5.3.3 Objective Function

$$\max \sum_{i=1}^{I} \sum_{u=1}^{U} \sum_{k=1}^{K} P_{iu} B_{iuk}$$

The main aim of Solvoyo is maximizing revenue. However, Solvoyo also wants to increase the visibility of products in the market. In that matter, while the objective function of the model only considers revenue, penetration constraint has been added to the constraints to increase the visibility of the products.

5.3.4 Critical Constraints

The mathematical model of the problem is given in the appendix. The standard TOP constraints are modified with additional constraints to fulfill the project's aim. Some of them are as follows:

Constraint (15) ensures that sales points cannot exceed their determined budget. Constraint (11) and (12) satisfy the penetration of each product should be left at the sales point that is visited along the route in order to ensure certain products are visible in the market. Constraint (17) enables to avoid the use of low-utilization trucks, trucks with less than the minimum truck revenue (MR) should not be used.

Different tools were used to solve the model. While Excel VBA was used to bring the data shared by the company into a format that can be entered as input to the model, CPLEX was used as a solver. The model is solved by calling CPLEX to Python software language, so that the parameters can be changed quickly, and outputs can be observed easily.

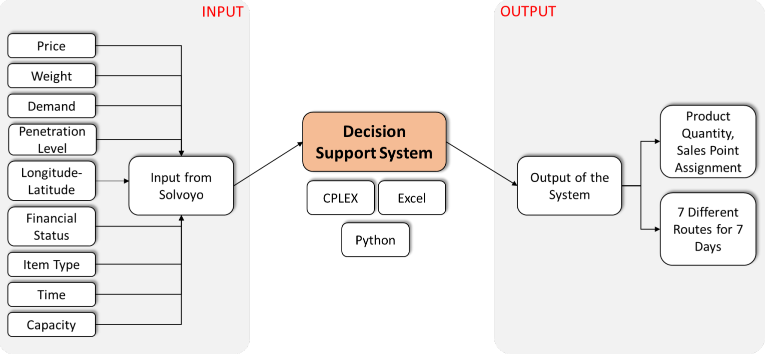


Figure 5.1: Flow chart of conceptual solution model

5.4 Verification and Validation

To verify the model, a smaller dataset was utilized. After the data implementation, we evaluated the model's performance with suitable input data. By applying this approach, we sought to ensure the accuracy and reliability of the model's outcome and to confirm the constraints of financial status, time, capacity, and penetration.

When trying to run the model with the real data given by the company, the comprehensive definition of the model revealed that the high number of decision variables resulted in significant computational time. To mitigate this issue, the model was iterated to generate seven routes for the weekly operation plan. The model is executed for the first day, utilizing the seven-day demand data as input. Then, the model is executed again for the second day, with the data that previous day's demand subtracted from the overall demand. This process is repeated for seven days, resulting in the generation of routes and the determination of their corresponding objective values. This approach provides a more robust and practical representation of the potential earnings that a truck could generate over the course of a week, taking into account the fluctuations in demand that may occur. The detailed table that shows the number of decision variables, and consequently, the decrease in the computational time, is in Table 5.1. Several sensitivity analyses with different time and capacity parameters are done successfully. After implementing the changes, an examination was carried out to ascertain the precision of the objective function values associated with the routes and the number of products left. In order to measure the credibility of the model against real results, we compared the results with

Before Applying Loop Procedure		
Decision Variable Name	Variable Type	Variable Count
A_{ijk}	Binary	90 * 90 * 7 = 56700
B_{iuk}	Integer	148 * 90 * 7 = 93240
U_i	Integer	90
Z_i	Binary	90 * 7 = 630
After Applying Loop Procedure		
Decision Variable Name	**Variable Type**	**Variable Count**
A_{ijk}	Binary	90 * 90 * 1 = 8100
B_{iuk}	Integer	148 * 90 * 1 = 13320
U_i	Integer	90
Z_i	Binary	90 * 1 = 90

Table 5.1: Number of variables

three different bi-week past operations. The period is chosen by considering truck capacity utilization, which is the main constraint for the model. The income, number of SKUs, and number of sales points visited were compared in three different past scenarios.

Among the scenarios, Solvoyo's current system shows that the company's methodology has weak consistency changing periodically. Even though the past income is sometimes higher than our model, they did not reach the total number of SKU as much as the model has.

Solvoyo's current system is focused more on profit than penetration, however, as discussed with the company, while earning more, they would also prefer to put more products on the shelves since they are in a competitive market. All in all, the decision support system we have prepared provides both an increase in revenue and penetration levels. As a result, the model provides the salesperson with optimal SKU capacity usage and an increased penetration level for each sales point on a fixed route.

5.5 Integration and Implementation

The software tool consists of a user interface created using Excel VBA and a mathematical model code written in Python utilizing the CPLEX library. The interface consists of running the model by linking Python and Excel Macros, adding new data to the system by an UserForm, shaping the inputs to be used as parameters for the model and finally demonstrating

	The Model	Scenario #1	Scenario #2	Scenario #3
Income	23,098	23,301	20,055	16,397
Capacity Usage	799.94	773.46	702.428	798.73
# of SKU	103	59	64	63
# of Sales Point	51	58	44	63

Figure 5.2: Scenario analysis

the outputs in an Excel file. As a result, the system is designed for long-term use, as it can provide an optimal solution for different variations of customers and products.

5.6 Benefits to The Company

New data was requested from the company and a benchmark was made with this new data. At this stage, when the data was handled, it was observed that the company missed many potential sales. Accordingly, it was observed that the company's truck was filled according to the forecast values entered. However, since the mistake made in the forecast would affect the entire sales, after consulting our instructors, it was decided to put back up stock to use the vehicle capacity more efficiently.

Approximately 3.5 months of the four-month data given to us was trained to obtain the forecast error amount for each product. To improve the forecast, the back-up stock is obtained by examining the forecast error amount considering the product's past actual demand amounts. The time interval for the training is n days, t = 1...n. To test the algorithm, the forecast day is n+1 th day, 106 th day in our case. After testing different approaches, an algorithm was created using the logic of the uniform distribution function, which would not overload the truck and allow us to catch underestimated sales. The empirical distribution values were obtained by using the difference between the forecast and actual data in the historical data. Depending on these values, the formula is;

$$\frac{\text{Forecast Value} - \min(\text{Actual Demand})}{\max(\text{Actual Demand}) - \min(\text{Actual Demand})}$$

By using this equation, back up values were added to the forecast.

In the sample sales district provided by the company, using the actual forecasts and sales values on the date 16/03/2023, when the model is run with the trained forecast data, the model yielded 5.66% more revenue compared to the past results.

Revenue According to Past Sales Distribution	Revenue According to Model Outputs	Improvement %
8.976,30 ₺	9.484,70 ₺	5,66%

Figure 5.3: Comparison results

According to these numbers, our solution approach of utilizing trained forecast data and using it as a parameter for the mathematical model to give the allocation of truck and sales routes that maximize the revenue improved the current system. The improvement resulted from allocating more products than the realized sales values compared to the prior allocated numbers according to the initial forecast, now being able to sell these extra units. Another aspect that contributed to this improvement is the optimal route/visiting sequence of sales points which enabled the truck to visit additional sales nodes within the time constraint, which posses potentially high revenues.

5.7 Conclusion

In conclusion, the decision-making process in the current traditional sales system relies on the salesperson's experience, relationships with local stores, and daily predictions. To optimize revenue and the visibility of certain products on shelves, this hot sales process is supported by a mathematical model. The output provides the amount of product in the vehicle, the route including sales points, and the amount of product to be left at each sales point. Based on these outputs, improvements on revenue have been achieved by 5.66%.

Bibliography

Panadero, J., E. Barrena, A. A. Juan, and D. Canca (2022). The stochastic team orienteering problem with position-dependent rewards. *Mathematics 10*(16).

Appendix: Mathematical Model

$$\max \sum_{i=1}^{I} \sum_{u=1}^{U} \sum_{k=1}^{K} P_{iu} B_{iuk} \tag{5.1}$$

s.t

$$\sum_{i=1}^{I} \sum_{u=1}^{19} B_{iuk} W_u \leq Y, \qquad \forall k \in K \tag{5.2}$$

$$\sum_{i=1}^{I} \sum_{u=20}^{148} B_{iuk} W_u \leq X, \qquad \forall k \in K \tag{5.3}$$

$$\sum_{i=1}^{I} \sum_{j=1}^{I} A_{ijk} D_{ij} \leq T, \qquad \forall k \in K \tag{5.4}$$

$$B_{iuk} \leq \sum_{j=1}^{I} A_{ijk} Dem_{iu}, \qquad \forall k \in K, u \in U, i \in I \tag{5.5}$$

$$\sum_{j=1}^{I} A_{1jk} = 1, \qquad \forall k \in K \tag{5.6}$$

$$\sum_{i=1}^{I} A_{i1k} = 1, \qquad \forall k \in K \tag{5.7}$$

$$\sum_{j=1}^{I} A_{jik} - \sum_{j=1}^{I} A_{ijk} = 0, \qquad \forall k \in K, i \in I \tag{5.8}$$

$$\sum_{j=1}^{I} A_{jik} \leq 1, \qquad \forall k \in K, i \in I \tag{5.9}$$

$$\sum_{k=1}^{K} B_{iuk} \leq Dem_{iu}, \qquad \forall u \in U, i \in I \tag{5.10}$$

$$\sum_{k=1}^{K} B_{iuk} \geq Pen_{iu} z_i, \qquad \forall u \in U, i \in I \tag{5.11}$$

$$M z_i \geq \sum_{k=1}^{K} \sum_{j=1}^{I} A_{ijk}, \qquad \forall i \in I \tag{5.12}$$

$$\sum_{k=1}^{K} \sum_{j=1}^{I} A_{ijk} \geq z_i, \qquad \forall i \in I \tag{5.13}$$

$$\sum_{k=1}^{K} A_{jik} = 0, \qquad \forall i, j \in I, i = j \tag{5.14}$$

$$\sum_{u=1}^{U} \sum_{k=1}^{K} P_{iu} B_{iuk} \leq F_i, \qquad \forall i \in I \tag{5.15}$$

$$u_i - u_j + N A_{ijk} \leq N - 1, \qquad \forall i \in I, j \in I, k \in K, i \neq j, j \neq 1 \tag{5.16}$$

$$\sum_{i=1}^{I} \sum_{u=1}^{U} P_{iu} B_{iuk} \geq MR, \qquad \forall k \in K \tag{5.17}$$

$$B_{iuk} \geq 0, \qquad \forall i \in I, u \in U, k \in K \tag{5.18}$$

$$A_{ijk} \in \{0, 1\} \qquad \forall i, j \in I, k \in K \tag{5.19}$$

$$z_i \in \{0, 1\} \tag{5.20}$$

Çevrim Süresi İyileştirmeleri ve Anomalilerin Tespiti | 6

Arçelik Elektronik İşletmesi

Proje Ekibi
Ezgi Amasyalı, Elif Atılsın, Ahmet Ege Çakır,
Necati Furkan Çolak, Nur Eraslan, Müge Evran, Serkan Uygun

Şirket Danışmanları
Orhan Sancaklı, Otomasyon Yöneticisi
Hüseyin Artun, Yazılım Testleri Otomasyon Mühendisi
Tüğçe Pınar Öztürk, Proje Yöneticisi

Akademik Danışman
Prof. Dr. Savaş Dayanık
Endüstri Mühendisliği Bölümü

Özet

Arçelik Elektronik İşletmesi'nde, üretilen televizyonlar paketlenmeden önce yazılım ve donanım kabul testlerinden geçmektedir. Testlerde yaşanan aksaklıklar hattın durmasına veya anomalilerin oluşmasına yol açmaktadır. Oluşturulan gerçek zamanlı kontrol çizitleri sayesinde anomalilerin tespiti ve hata analizi olanağı sağlanmaktadır. Ek olarak, statik bir test sırasının olması aynı hat üzerinde üretilen farklı model televizyonların çevrim sürelerini kısıtlamaktadır. Geçmiş test bilgileri otomatik analiz edilerek, dinamik bir hat dengeleme sistemi oluşturulmuştur. Böylelikle üretimi yapılacak modele özel hazırlanan test sırası, çevrim süresinin azaltılmasına yardımcı olacaktır.

Anahtar Sözcükler: Kalite kontrol, hat dengeleme, veri analizi.

Improvement of Cycle Times and Detection of Anomalies

Abstract

At Arçelik Electronics Facility, the televisions produced go through software and hardware acceptance tests before packaging. Any malfunctions during these tests can stop the production line or cause anomalies. Real-time control diagrams are created to detect anomalies and facilitate error analysis. Additionally, having a static test sequence limits the cycle times of different model televisions produced on the same line. A dynamic line balancing system is created using past test data. As a test sequence specific to the model will be produced, this will help reduce cycle times during production.

Keywords: Quality control, line balancing, data analysis.

6.1 Company and Problem Definition

Arçelik is a worldwide company established in 1955 that offers services on household appliances and technology in Turkey. There are about 42.000 workers and 28 plants. Our main focus in this project was on Arçelik Electronics Plant, which comprises the Technology and Research Development Department.

The main problem is the cycle time anomalies occurring due to time fluctuations during the quality process. There are several terminals, and each can apply a different number of tests because each production line manufactures different products.

The aim of the project is to design a methodological approach and interface to detect the cycle time anomalies, which is our monitoring system, and line balancing system, which improves company's performance. Since in a business with high demands and tight deadlines, any delay is a possible loss of profit; it is vital to provide a solid and implementable approach applicable to any data for the future use.

6.2 Line Balancing and Monitoring System

The "Line Balancing and Monitoring System" is a Python-based application that is at the heart of our project. The application can be reached via the user interface and includes the following functions:

- Running the developed line balancing model dynamically to suit the current production batch.
- Monitoring the control charts of each station in real-time.

- Chart inspection tool that can help analyze the anomalies in cycle times.

The application is an integrated system that houses several modules that function according to the above-mentioned points. When the application is opened, the main window will welcome the user. It is designed with ease of use and easy navigation in mind.

6.3 Developing the Line Balancing Model

The line balancing model was developed by the project team to tackle the static test sequence used by Arçelik. The model is solved in R with the lpSolve optimization package. The lpSolve package allows us to code the developed Mixed-Integer Linear Programming model. The input to the model is the past data gathered from the operations in the production line. The past data are advised to contain the past two weeks' testing information to achieve more general and smoothed testing times. The input is an Excel file that the Arçelik team can easily retrieve from the company database.

The mathematical model can be seen in the appendix. The model Dou et al. (2017) has helped us to develop the precedence constraints of our model. However, the model Gao et al. (2009) has heavily influenced the development of our model as it was an assignment problem of task-to-station, which is very similar to our problem. It provided a feasible sequence of tasks with assignments to the terminal with the objective function to minimize the difference between the terminals' maximum and minimum cycle times. With the help of these two influencing models, our model aims to provide a solution to the static test sequence in the production line.

6.3.1 Critical Assumptions

Our critical assumptions are the following:

- No errors emerge during the transportation between testing stations.
- Transportation times between stations are constant. The transfer time from the preceding terminal is added to the processing time of the current terminal.
- Processing times are taken as the 80th percentile of the processing time distributions, under controls, of tasks.
- Immediate reading of testing specifications is assumed for effective online quality control.
- There are no read/write errors occurring in the remote server connected to the terminal computers.

6.3.2 Major Constraints

For the line balancing problem, some major constraints are as follows:

- The television units must pass prerequisite tests to move forward.
- Some tests can be assigned to only certain terminals due to hardware limitations.
- Some tests have to be conducted twice.
- Different chassis take different sequences of tests.
- Testing results are stored on the Arçelik server.
- Additional tests are done to calibrate a test for new products.

6.3.3 Model Verification

After the model was finalized, verification of the model was conducted. We have conducted six main tests to verify the model.

- Reduced Models: We have analyzed the reduced models by deleting certain constraints. The resulting problem was solved via other means of optimization tools. We have made sure the same solution is obtained from our model.
- Precedence Relations: By changing up the precedence relations and identifying new ones we have analyzed the difference in the model's output.
- Duplicate Test: Certain tests have to be duplicated. By changing the parameters, we have verified that the tests we want to be duplicated appear twice in the sequence.
- Empty Terminals: Certain terminals have to be left empty. By changing the desired empty terminal we have verified the chosen terminal is not assigned any tests.
- Stay Constraints: There are tests that have to stay in a certain terminal defined by the Arçelik team. By changing this input we have verified that the task is assigned to the terminal that we identify and not any other terminal.
- Objective Function: We have manually changed some of the input cycle times to verify the objective function is changing to our expected values. We have achieved the expected outcomes to verify with the degeneracy test.

We changed the needed parameters for each test and successfully reached the target output for the test cases. As all the tests were a success, the model was deemed verified.

6.3.4 Model Validation

The model must be validated to be used in a real-life scenario. The important step for our model to be deemed valid is the feedback and the expert opinion we got from the Arçelik team. As they are experts on software testing automation, their recognition of the model as a viable tool for decision-making is important. Another important step for the validation of the model is the operation validity as the model will be used in a manufacturing setting. Conducting a pilot study is crucial for operational validity. Therefore, we have acquired a new set of data to obtain a test sequence for a specific television model. Then, we provided this test sequence to try it in the production line without affecting any other television model production. The results of this pilot study will determine the final verdict of the validity of our model.

6.3.5 Pilot Study and Comparison

In Figure 6.1, we can see the original terminals that a particular chassis, in this case "AND1", goes through. Figure 6.2 shows the cycle time that a sample television spends in those terminals. The cycle time of that chassis can be depicted as 31.78 seconds, with the T01 creating a bottleneck for the cycle time. With the help of our decision support system, Arçelik will be able to configure the testing sequence dynamically and get a specific test sequence for that chassis.

In the pilot study, the AND1 chassis is chosen as the parameter. When the past two weeks' data is inputted into the program, the Arçelik team receives the testing sequence matrix seen in Figure 6.3. The '1' means that the test present in Arçelik's system is allocated to the corresponding terminal, and if the matrix cell is a dot, that test is not present in that terminal. When the test sequence is tested on the production line, the theoretical cycle times can be seen in Figure 6.4. The theoretical improvement in the cycle times is affected by the T05A/T05B terminal with a 19.46 second cycle time for the system. The cycle time has theoretically improved by 39%. The Arçelik team expects to report about 10% to 15% improvement in the cycle times in that particular chassis. We can see that the implementation in the actual production line resulted in much lower than the theoretical calculations. This is to be expected as there are much more variables in play in real-life scenario that can affect the cycle time of a production run.

	T1	T2	T3	T4	T5A	T5B	T6	T7	T18	T9	T10	T11
TEST15	1	0	0	0	0	0	0	0	0	0	0	0
TEST18	1	0	0	0	0	0	0	0	0	0	0	0
TEST2	1	0	0	0	0	0	0	0	0	0	0	0
TEST3	1	0	0	0	0	0	0	0	0	0	0	0
TEST4	1	0	0	0	0	0	0	0	0	0	0	0
TEST7	1	0	0	0	0	0	0	0	0	0	0	0
TEST8	1	0	0	0	0	0	0	0	0	0	0	0
TEST23	0	0	0	0	0	0	0	0	0	0	1	0
TEST34	0	0	0	0	0	0	0	0	0	0	1	0
TEST35	0	0	0	0	0	0	0	0	0	0	1	0
TEST36	0	0	0	0	0	0	0	0	0	0	1	0
TEST38	0	0	0	0	0	0	0	0	0	0	0	1
TEST39	0	0	0	0	0	0	0	0	0	0	0	1
TEST8	0	0	0	0	0	0	0	0	0	0	0	1
TEST19	0	1	0	0	0	0	0	0	0	0	0	0
TEST20	0	0	0	1	0	0	0	0	0	0	0	0
TEST21	0	0	0	1	0	0	0	0	0	0	0	0
TEST22	0	0	0	1	0	0	0	0	0	0	0	0
TEST23	0	0	0	0	1	0	0	0	0	0	0	0
TEST24	0	0	0	0	1	0	0	0	0	0	0	0
TEST25	0	0	0	0	1	0	0	0	0	0	0	0
TEST26	0	0	0	0	1	0	0	0	0	0	0	0
TEST23	0	0	0	0	0	1	0	0	0	0	0	0
TEST24	0	0	0	0	0	1	0	0	0	0	0	0
TEST25	0	0	0	0	0	1	0	0	0	0	0	0
TEST26	0	0	0	0	0	1	0	0	0	0	0	0
TEST27	0	0	0	0	0	0	1	0	0	0	0	0
TEST28	0	0	0	0	0	0	1	0	0	0	0	0
TEST29	0	0	0	0	0	0	0	1	0	0	0	0
TEST30	0	0	0	0	0	0	0	1	0	0	0	0

Figure 6.1: Current Test Sequence

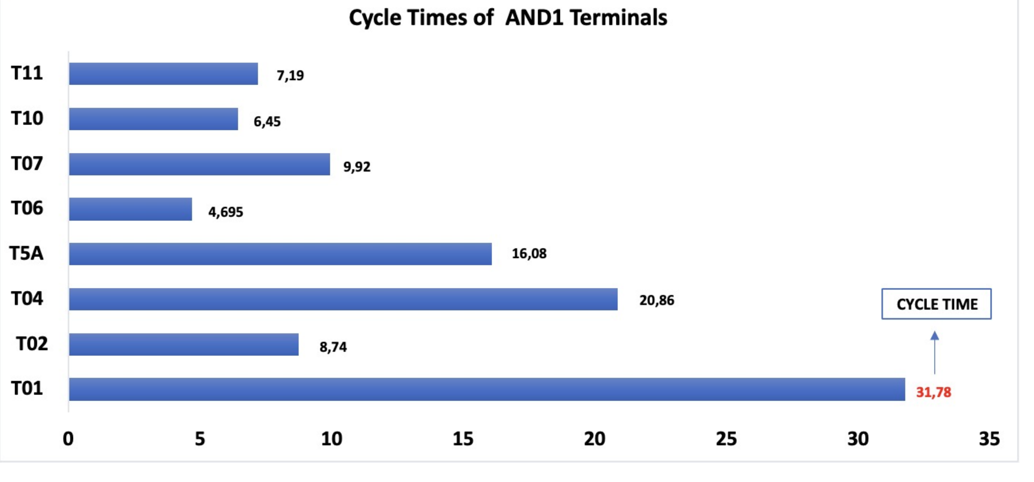


Figure 6.2: Current Cycle Times of Arçelik's System

	PID	T0	T1	T2	T3	T4	T5A	T5B	T6	T7	T8	T9	T10	T11	T12	T13
TEST1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TEST14	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TEST15	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TEST2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TEST3	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TEST4	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TEST8	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TEST18	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
TEST2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
TEST3	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
TEST4	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
TEST6	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
TEST7	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
TEST8	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
TEST23	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
TEST33	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
TEST34	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
TEST35	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
TEST36	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
TEST37	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
TEST38	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
TEST8	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
TEST27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
TEST35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
TEST38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
TEST39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
TEST40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
TEST8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
TEST19	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
TEST20	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
TEST21	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
TEST22	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
TEST23	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
TEST24	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
TEST25	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
TEST26	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
TEST23	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
TEST24	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
TEST25	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
TEST26	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
TEST27	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
TEST28	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
TEST29	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
TEST30	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Figure 6.3: Improved Test Sequence

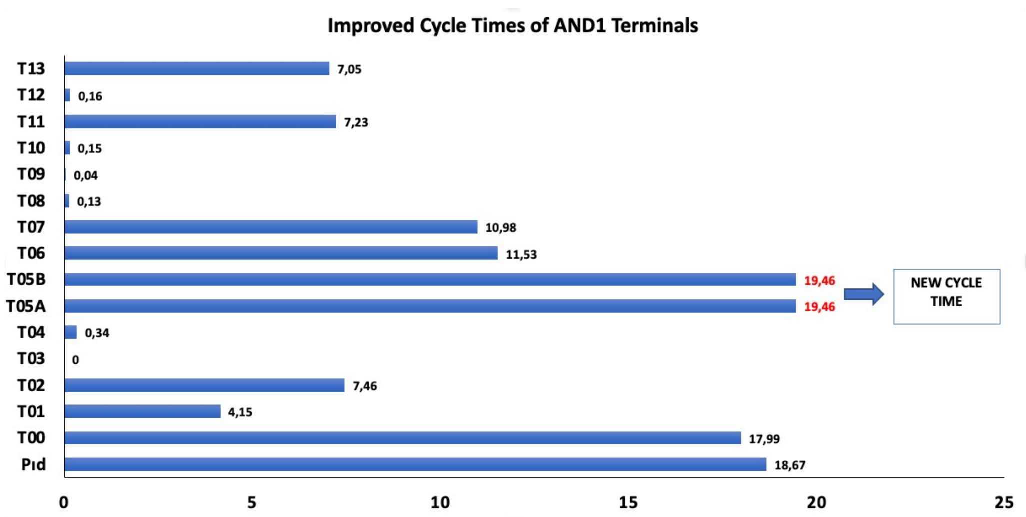


Figure 6.4: Improved Cycle Times of Arçelik's System

6.4 Integration and Implementation

The proposed solution is the application we have created. Arçelik is able to use this interface and integrate it into their current system successfully. This interface has two key elements inside:

- A section in which Arçelik is able to see the data patterns through quality control charts and detect the anomalies that occurred with the help of these charts. The system also allows Arçelik to see the data patterns daily. It is possible that Arçelik analyzes the quality control charts from past data. If there is a rule violation in the data or the defective items is above the threshold determined by Arçelik, the system will print a warning. The user interface will group the items according to chassis code, terminal and product types of the items. This process takes place as follows:
  - The project in focus for Arçelik is fundamentally a classification system of their production lines, utilizing a color-coded scheme based on error percentages. The error percentage is computed by the proportion of defective products against the total output. To distinguish the performance of each line, three colors are employed: red, yellow, and green.
  - Arçelik has determined the targeted values based on historical data and in accordance with its current systems as follows: The color red is assigned to a production line where the error rate is above 3%, signifying a high defect rate. Yellow is used for error rates between 3% and 1%, indicating a moderate level of errors. Green, on the other hand, denotes a commendable performance with an error rate of less than 1%. However, input from the user is included in order to integrate it into the company system and to provide a system that can be used for many years.
  - The design of this project is an interface that allows users to adjust the error thresholds for each color. This feature ensures the system can be tailored to suit various scenarios and requirements, enhancing its utility and adaptability.
  - The project offers an additional feature of recommended line visit numbers corresponding to the selected error rate. For example, an error rate of 5% suggests five line visits, whereas an error rate of 3% suggests nine visits. These visit numbers are derived from the probabilities of false alarms, providing a data-driven approach to maintain quality control.

  - Arçelik's main objective is to maintain a stable error rate around 1.4% is taken into account within this system. However, it should be noted that achieving this error rate implies a need for approximately 15 line visits, as per the calculated false alarm probabilities. This illustrates the relationship between the desire for a lower error rate and the consequent increase in required line visits.

- A section in which the line balancing mathematical model provides a better layout for the company. The outputs of this model are cycle times for the chosen terminal. The model outputs can be seen in our interface as well.

6.5 Benefits to Arçelik

The line balancing and monitoring system establishes itself as a decision-making support to enhance the company's daily operations. Currently a single test sequence is used for every different model manufactured. This static test sequence reduces production efficiency as the sequence is not optimal for every model. By using our application, the Arçelik team will be able to receive testing sequences for each production batch and change the production line's testing sequences according to the output. Furthermore, the ability to create quality control charts will help the Arçelik team to conduct a detailed analysis of the current test sequence. Arçelik will be able to detect possible errors and take immediate actions to remove them by using our system. In summary, the benefits are:

- application with a user-friendly design that requires a minimal amount of manual user input,

- dynamic test sequence to minimize the cycle times according to the produced television batch,

- higher throughput rate with optimized test sequences,

- decrease in idle time and downtime duration due to an optimized test sequence, and

- visual monitoring module to assist users on identifying and predicting quality control errors,

6.6 Conclusion

The implementation process includes adapting our designed user interface into Arçelik's current system. To conclude, our solution provides a systematic detection approach that aims to minimize cycle time anomalies. Arçelik can see the related quality control charts as well as their interpretations. They can analyze the data patterns and the charts are user-friendly. Our line balancing model can be seen also in our interface. Our whole solution increases the productivity of the company both in terms of time efficiency and quality of the finished products.

Bibliography

Dou, J., J. Li, and X. Zhao (2017). A novel discrete particle swarm algorithm for assembly line balancing problems. *Assembly Automation 37*(4), 452–463.

Gao, J., L. Sun, L. Wang, and M. Gen (2009). An efficient approach for type ii robotic assembly line balancing problems. *Computers & Industrial Engineering 56*(3), 1065–1080.

Appendix: Line Balancing Model

Declarations:

$T = \{1, ..., 40\}$: List of tests

$T_r = \{1, ..., 16\}$: List of terminals.

$C = \{1, ..., 8\}$: Range of chassis present in the system.

$S = \{1, ..., 5\}$: Range of television sizes present in the system.

$D = \{8, 15, 38, 39, 40\}$: Set of test that has to be done twice.

$O = \{5\}$: Terminals that have to be left empty due to Arçelik exceptions.

Parameters:

$$t_{ikl} = \text{80th percentile test time of test } i \in T, \text{ chassis } k \in C, \text{ size } l \in S$$

$$\text{stay}_{ij} = \begin{cases} 1, & \text{If test } i \text{ stays in terminal } j \\ 0, & \text{otherwise} \end{cases} \qquad i \in T, j \in T_r$$

$$t_i = \max(\text{time}(i, k, l)), \quad i \in T$$

$$\text{pred}_i = \text{list of tests preceding test } i; e.g., \ \text{pred}_{20} = \{18, 19\}, i \in T$$

Decision Variables:

$$x_{ij} = \begin{cases} 1, & \text{if test } i \text{ is assigned to terminal } j \\ 0, & \text{otherwise} \end{cases} \qquad i \in Tj \in TR$$

z = maximum time imbalance across stations

Model:

$$\text{minimize } z \tag{6.1}$$

subject to:

$$x_{i,j} = 0, \qquad \forall j \in O, \qquad i \in T \tag{6.2}$$

$$x_{8,14} = 1 \tag{6.3}$$

$$\sum_{j \in Tr\ :j<14} x_{8,j} = 1 \tag{6.4}$$

$$\sum_{j \in Tr} x_{i,j} = 1, \qquad \forall i \in T\ D \tag{6.5}$$

$$\sum_{j \in Tr} x_{i,j} = 2, \qquad \forall i \in D \tag{6.6}$$

$$\sum_{i \in T} x_{i,j} \geq 1, \qquad \forall j \in T_r\ O \tag{6.7}$$

$$\sum_{j \in Tr} j x_{h,j} \leq \sum_{j \in T_r} j x_{i,j} \qquad \forall i \in T \qquad \forall h \in pred(i) \tag{6.8}$$

$$x_{i,j} \geq stay_{i,j} \qquad \forall i \in T \qquad \forall j \in T_r \tag{6.9}$$

$$z \geq \max \sum_{j \in Tr} t_i x_{i,j} - \min \sum_{j \in Tr} t_i x_{i,j} \qquad \forall i \in T \tag{6.10}$$

Linearization of (12):

$$M \geq \sum_{i \in T} t_i x_{i,j} \qquad \forall j \in T_r \tag{6.11}$$

$$m \leq \sum_{i \in T} t_i x_{i,j} \qquad \forall j \in T_r \tag{6.12}$$

$$z \geq M - m \tag{6.13}$$

$$x_{ij} \in \{0,1\}, \qquad i \in T, j \in T_r \tag{6.14}$$

Explanation of Constraints:

(3) Objective tries to minimize the maximum difference between any two terminals.

(4) There must not be any test assigned to the terminals in the set: O.

(5) TEST 8 has to be present in T14.

(6) TEST 8 has to be present once before T14.

(7) Ensures that all the tests are assigned to any terminals. There cannot be any unassigned tests.

(8) Tests declared in set D must be present in the test sequence two times.

(9) Ensures that at least one test is assigned to the terminals; a terminal cannot stay empty unless the terminal is in set O.

(10) Ensures that the existing precedence constraints are satisfied. If test j precedes i, test j has to be in the same terminal or any preceding terminal.

(11) Ensures that the hardware limitations of tests are satisfied.

(12-15) Objective tries to minimize the difference between the maximum cycle time and minimum cycle time of any two terminals.

(16) Binary constraint for the decision variable.

Damperli Kamyon Kestirimci Bakım Tahmini | 7

Demir Export

Proje Ekibi
Ayşe Çağlar, Bora Ege Duygun, Özgün Barış Kır,
Ahmet Alperen Tartıcı, Hatice Ünal, Senih Yılmaz, Berçinay Yücel

Şirket Danışmanları
Haydar Çınar, Lider Veri Bilimcisi
Pınar Tekin, Proje Portföyü ve
Dijital Çözümler Sorumlusu
Özge Göksu Başer, Veri Analisti

Akademik Danışmanlar
Prof. Dr. Ülkü Gürler
Endüstri Mühendisliği Bölümü
Doç. Dr. Emre Berk
İşletme Bölümü

Özet

Demir Export Kangal Kömür Madeninde faaliyet gösteren kamyonlara arızi ve periyodik olmak üzere iki tür bakım yapılmaktadır. Periyodik bakım ile değiştirilmesine gerek olmayan parçalar değiştirilmekte ve buna bağlı olarak operasyonda duruşlar gerçekleşmektedir. Arızi bakımda ise kamyonun arıza vermesiyle bakım gerçekleştirilmekte ve bu durum arızanın bağlı olduğu birçok operasyonu etkilediğinden durum şirket için maliyetli olmaktadır. Şirket dijitalleşme faaliyetleri kapsamında 2017'den bu yana kamyonlardan sensörler ile veri toplamaktadır. Bu bağlamda proje, toplanan verileri kullanarak arızaların gerçekleşeceği zamanı tahmin edebilecek bir karar mekanizması geliştirmeyi amaçlar.

Anahtar Sözcükler: Kestirimci Bakım, Keşifsel Veri Analizi, Zaman Serileri

Dump Truck Predictive Maintenance Estimation

Abstract

Two types of maintenance are carried out on the trucks operating in Demir Export Kangal Coal Mine, breakdown and periodic. Periodic maintenance involves replacing parts that do not need to be changed, which results in operational downtimes. In case of breakdown maintenance, maintenance is carried out when the truck fails, and this situation is costly for the company as it affects many operations to which the fault depends. The company has been using sensor-equipped trucks to gather data as part of its digitalization initiatives since 2017. In this context, the project aims to develop a decision mechanism that can predict when failures will occur using the collected data.

Keywords: Predictive Maintenance, Exploratory Data Analysis, Time Series

7.1 About the Company

Demir Export Inc. is a mining company established in 1957 under Koç Holding. It employs over 2200 people. The business, which began by producing iron ore, expanded its business in 1970 to include lead and zinc mining. Today, in addition to such ores, it also produces gold, silver, and copper concentrates. With a production of 11.5 million tons of coal annually, Demir Export is Turkey's largest coal producer. It has 4 active operations which are Bakırtepe Gold Mine, Divriği Iron Mine, Eynez Underground Coal Mine and Kangal Coal Mine.

7.2 System Analysis and Problem Definition

Different maintenance techniques are used in the Kangal Coal Mine to repair truck problems. For example, when a truck part spontaneously breaks down, breakdown maintenance is performed. In this procedure, the company either replaces the broken part or finds an alternative corrective action that addresses the problem. Another, the company adopts periodic maintenance, which is replacing specific parts in a periodic manner to avoid malfunctions. However, the parts that do not need to be changed are nonetheless routinely replaced with periodic maintenance. As a result, downtimes happen even when no maintenance is required. Still, there is a third choice: predictive maintenance. Predictive maintenance refers to the ability to identify trucks that are likely to break down in advance and perform the appropriate corrective action beforehand. To adopt this method, the company placed sensors to measure various truck data to determine the underlying issues.

In our project, therefore, these measurements were used to construct a decision support tool that determines the likelihood of malfunctions for certain faults so that malfunction-prone vehicles can be identified before they crash (Hashemian, 2010).

7.3 Solution Approach

In order to predict the error codes in the future, it is assumed that the particular errors are generated based on a probability distribution. The initial step taken to recover the underlying distribution of the errors is preprocessing the raw data. After discarding the durations in which the trucks were idle and not used, one can measure the actual working times of the trucks between consecutive errors, and these samples can be used as interarrival times (see Section 7.5 for the details of preprocessing). The second step is, by using these interarrival times, one can approximate the original probability distribution in terms of either well known distributions such as Weibull, Gaussian, exponential or rather less known distributions such as Nakagami, Maxwell, Fisk by using the SciPy library in Python. However, merely recovering the probability distribution of the errors is not enough for making predictions. To overcome this issue, one can calculate the conditional probability distribution of the original distribution via

$$p = P(X < t + T \mid X > T) = 1 - \frac{1 - F(t + T)}{1 - F(T)}.$$

Here, the p represents the probability of encountering an error within a duration of t, given that an error has not been observed for a duration of T. For example, if t is 5 days and T is 10 days, then p is the probability of observing an error within the next 5 days given that an error has not been observed for the past 10 days. An advantage of this method is that the cumulative distribution function (represented in the formula as F) does not need to be written analytically, since the required values can be calculated numerically by using the SciPy library in Python. As a third step, different t values were displayed in the user interface to observe different probabilities in a graphical manner. The details of the first two steps (preprocessing and model fitting) were discussed at length in Section 7.5 whereas the third step (graphical user interface) was displayed step-by-step in Section 7.6.

7.4 Verification and Validation

7.4.1 Verification

The verification of the mathematical model has been accomplished in two different setups. In the first setup, the underlying distribution of the error codes has been assumed to be uniform[0,100] (in days). Since it is analytically known that the conditional distribution of uniform is also uniform with different parameters, we have checked if the result of our algorithm coincides with theoretical expectations. Assuming that an error has not observed for the last 0 days (meaning “T=0” in the formula above), the output is as follows: Figure 7.1 shows that the underlying distribution is

Distribution: uniform
Parameters: (0.0056474625463209804, 99.9914839688625)
P-value of K-S Test: 0.9044
Samples are likely from 'uniform' distribution with 90.44% confidence.

Figure 7.1: K-S test result for uniform distribution

uniform[0,99.99] with 90.44% confidence. In Figure 7.2, the conditional cumulative probability distribution displays that the next error is expected to be observed uniformly within the following next 100 days. Now, for the same underlying distribution, if we assume that an error has not been observed for the last 50 days instead of 0 days, the outputs become as in Figure 7.3 Again, we have found the underlying distribution as uniform[0.2, 99.57] with 82.56% confidence based on the Kolmogorov-Smirnov test, just like the previous version. In Figure 7.3, it is observed that the next error is expected to be uniformly distributed for the following 50 days, in other

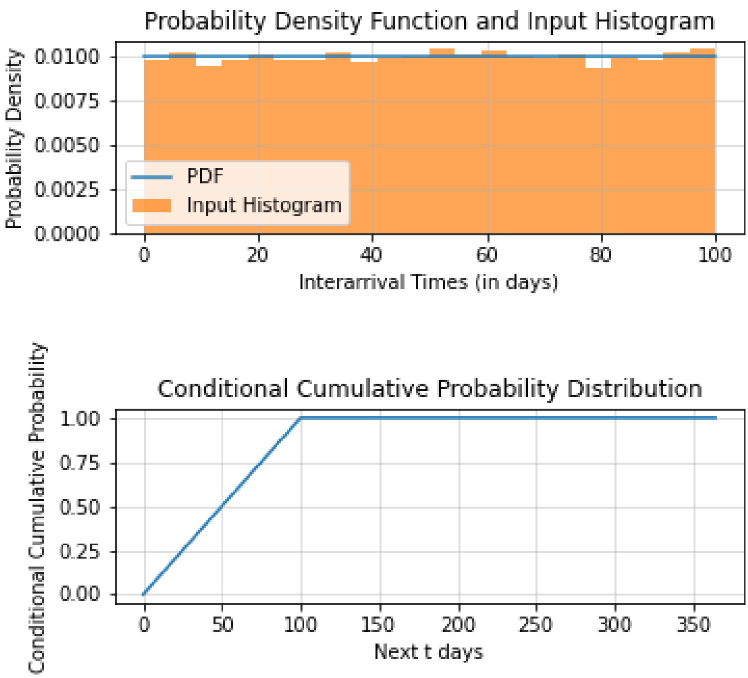


Figure 7.2: Pdf and cdf of fitted distributions

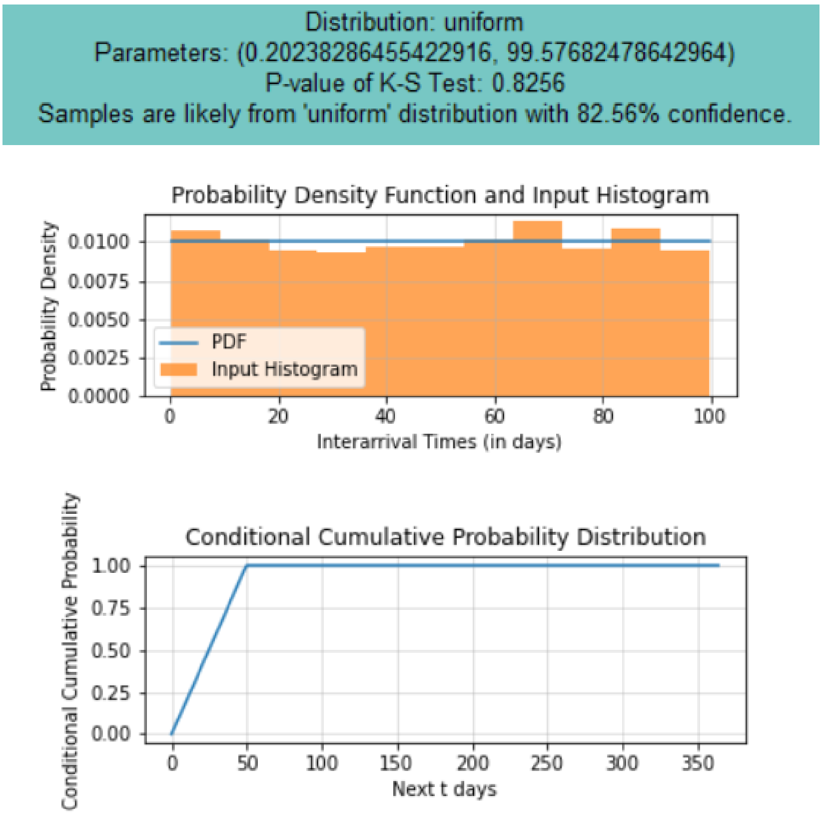


Figure 7.3: K-S test resultand fitted pdf function

words, the conditional cdf is uniform[0,50]. Thus, since this finding also coincides with theoretical expectations, the model is verified to be working properly in this setup.

In the second setup, it is assumed that the underlying distribution of the errors is exponential with mean 100, instead of uniform[0,100]. Theoretically, regardless of the number of days passed without the emergence of an error, the conditional cumulative probability distribution should be the same because of the memoryless property of the exponential distribution. Indeed, generating exponential random variables as interarrival times and forecasting them for different parameters (T=5 and T=50, meaning that an error was not observed for the last 5 and 50 days, respectively) yields Figure 7.4 Here, the conditional distributions yield the identical result for different parameters. That is, regardless of how many days pass without the emergence of an error, the future expectations always remain the same, as expected by the memoryless property of the exponential distribution.

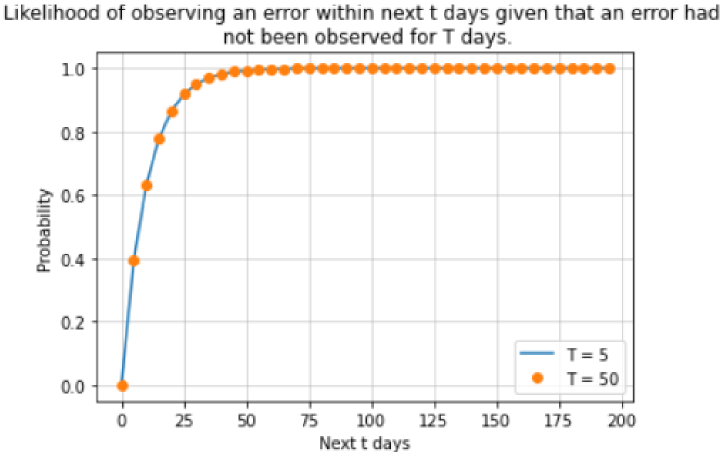


Figure 7.4: Exponential distribution

7.4.2 Validation

For the validation of the mathematical model, the interarrival times were randomly partitioned into two sets, the training set containing 80% of the total and the validation set containing the remaining 20%. Then, an underlying distribution was fitted by only using the samples in the training set. Finally, the samples in the validation set were checked if they belong to the fitted distribution by Kolmogorov-Smirnov test. Doing such for a sample truck yielded the results in Figure 7.5 The underlying distribution

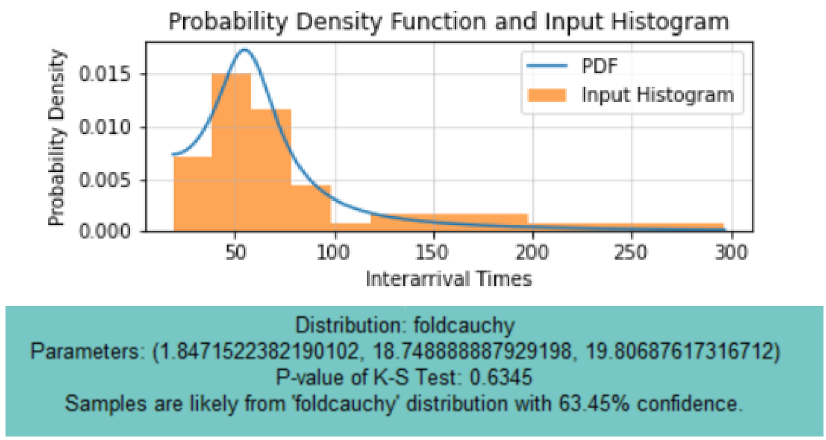


Figure 7.5: Validation

was found to be "foldcauchy" distribution (i.e., normed Cauchy r.v.) with 63.45% confidence for the samples in the training set.The interarrival times in the validation set belong to the fitted distribution with 30.28% confidence for this particular sampling. While being lower than the original fit, this result is still sufficient to assert that with a significance level of 5%, it is not possible to reject the null hypothesis which states that the samples belong to the “foldcauchy” distribution.

7.5 Integration and Implementation

Our implementation of the “Forecasting by Probability Distribution” method started with eliminating the idle times in which the trucks were not working. First of all, since there was a huge time gap before February 2021 due to the Covid-19 Pandemic, the origin of the data set is taken as February 2021. Figure 7.6 shows that the particular truck was not used for a long

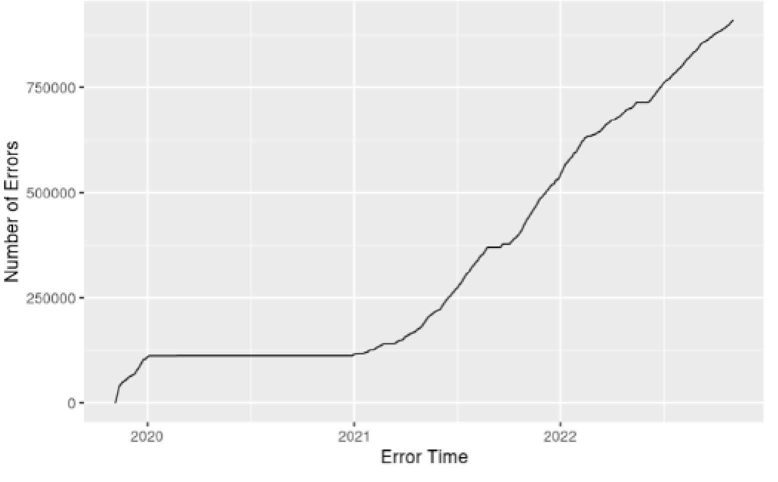


Figure 7.6: Cumulative number of errors versus time

time between 2020 and 2021. Therefore, using the interarrival times that occurred between 2020 and 2021 was quite likely to result in the recovery of a flawed underlying distribution. In fact, it is easier to see in Figure 7.7 that this interarrival time is indeed an outlier, and should be precluded during the modeling phase (the tall vertical line at the end of the 2020).

Moreover, the actual working time of the truck was calculated based on the “EngineSpeed” feature, which displays the RPM of the engine at the time it is measured. That is, the intervals in which the engine speed was observed to be “0” were precluded during the modeling phase. Nevertheless, since the collection of the error data was made in a periodic manner, specifically every 10 seconds, there was a need to cluster the error outputs that resulted from the same event. To solve this issue,a threshold parameter was defined in order to distinguish whether consecutively generated error codes are independent or not. For example, setting a threshold of one week means that any consecutive errors that have emerged in the same week are considered as a single error, and they are considered two different errors if

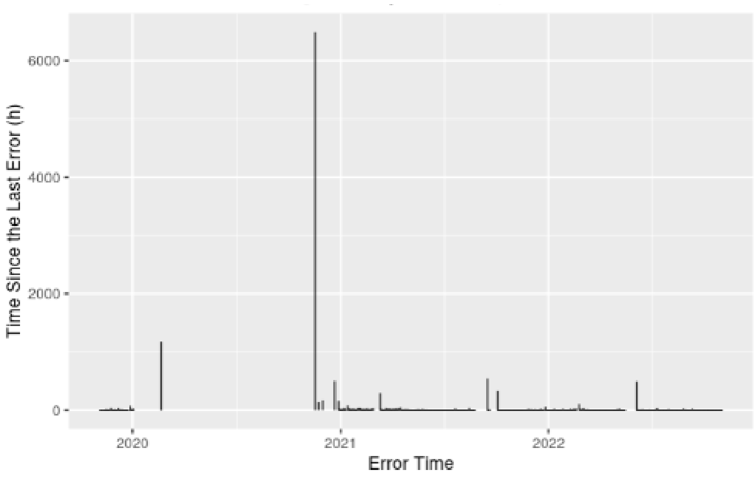


Figure 7.7: Event times and times passed since the previous event time

the duration between them exceeds this particular threshold. By applying that method, at the very least, the dependency between subsequent errors were aimed to be minimized. The problem here was to determine what should be the specified value of the threshold. Since the determination of such a threshold requires domain knowledge about trucks in general, for example how much working time of a truck would likely to pass to generate independent occurrences, the best option was found to be to keep this parameter as a variable that is going to be decided by the experts in the company. Then, the SciPy library in Python was used to fit these interarrival times into several known distributions in which the parameters of each distribution was calculated with the "Maximum Likelihood Estimator" method whereas the success of each prediction was determined with the Kolmogorov-Smirnov test. Finally, the findings which show the likelihood of encountering an error in the future were displayed graphically with a graphical user interface.

7.6 Graphical User Interface

The graphical user interface (GUI) retrieves interarrival times from an Excel file and a user entered parameter that shows the number of days passed without observing an error; see Figure 7.8 first two screenshots. Therefore, selecting the input Excel file and typing the number of days without the corresponding error code, our model runs in the background to find the best fitting three distributions to the preprocessed interarrival values. The resulting screen prompts three buttons to plot the clicked button's Probability Density and Conditional Cumulative Probability distributions. For example, running our code with the interarrival times computed in the pre-

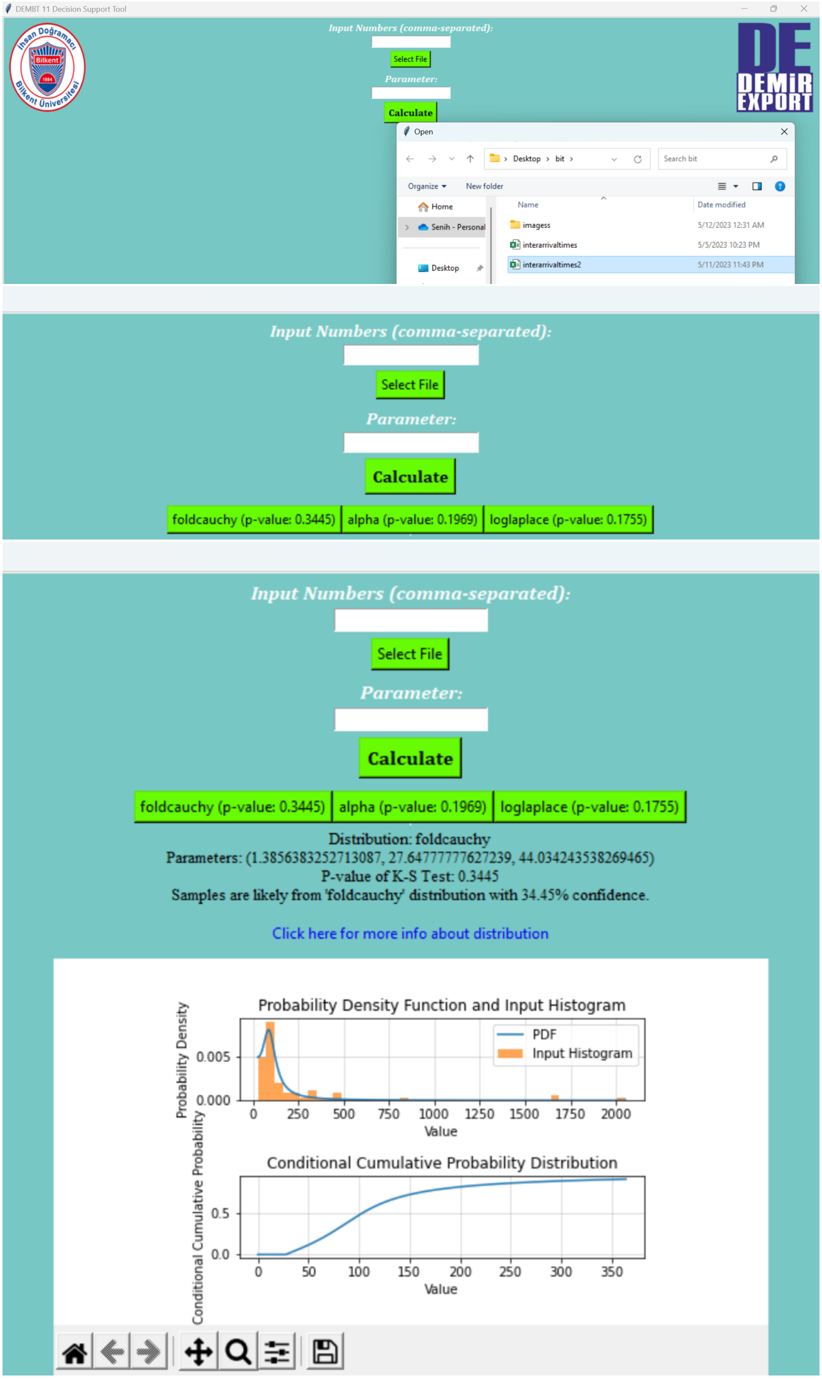


Figure 7.8: User interface

processing step, we obtain last screenshot in Figure 7.8. The desired probability distributions can be selected to plot their corresponding probability density and conditional cumulative probability.

7.7 Benefits to the Company

At the moment, the company does not use any form of previous data to predict truck problems. So they either perform periodic maintenance or they stop the operations to perform breakdown maintenance, both of which lead to inefficiencies in the operations. However, with the help of the algorithm as a decision support tool, the company will be able to predict when a truck will break down, allowing it to schedule maintenance of the trucks before any breakdowns happen. By doing this, it will be possible to reduce the inefficiencies caused by unanticipated system failures.

Bibliography

Hashemian, H. M. (2010). State-of-the-art predictive maintenance techniques. *IEEE Transactions on Instrumentation and measurement 60*(1), 226–236.

Depo Yerleşim Tasarımı ve Adresleme Karar Destek Sistemi | 8

Tepe Betopan

Proje Ekibi
Ezgi Akdeniz, Mertcan Süleyman Doğan, Yiğit Durgun,
Beyza Naz Elçiçek, Utku Kuzey Ergül, Can Baran Kocatürk, Deniz Yayla

Şirket Danışmanı
Çağatay Çaparlı
Üretim Planlama ve Lojistik Müdürü

Akademik Danışman
Prof. Dr. Bahar Yetiş Kara
Endüstri Mühendisliği Bölümü

Özet

Çimento-bazlı parçacık üretimi yapan Tepe Betopan A.Ş.'nin yarı mamul deposunda forklift hareketlerini enazlamak amaçlanmıştır. İşletme maliyetlerini düşürürken verimliliği artırmak ve forklift hareketlerini azaltmak için karışık tamsayılı doğrusal programlama ve sezgisel yaklaşımlar geliştirilmiştir. Karar destek sisteminin rahat kullanımı için grafik kullanıcı arayüzü oluşturulmuştur. Forklift hareketlerinin genel olarak azalması beklenmektedir. Projenin fabrika içi operasyonel verimlilik açısından önemli sonuçları olacaktır.

Anahtar Sözcükler: Depo yerleşimi, adresleme algoritması, ürün elleçleme, forklift hareketleri

Warehouse Layout Design and Decision Support System for Addressing

Abstract

Forklift movements inside the semi-finished warehouse of Tepe Betopan A.Ş., a well-known producer in the cement-bonded particle field have been optimized. To decrease forklift movement and boost efficiency while lowering operating expenses, the study combines mixed integer linear programming and heuristic approaches. An optimization model is created through investigating the current warehouse organization and forklift movement patterns, and it is then solved using mixed integer linear programming methods. The solution approach is subsequently modified using heuristic methods. A graphical user interface is built via Microsoft Excel Visual Basic for Applications. Forklift movements are predicted to decline, which will have significant implications for the company's operational efficiency.

Keywords: Warehouse layout , addressing algorithm, material handling, forklift movement

8.1 About the Company

Tepe Betopan, which initiated their production of its registered brand "Betopan" in 1984, is a cement-bonded particle board manufacturing company whose sole ownership is held by Bilkent Holding. The company develops products to keep up with the advancing technology. The product portfolio, consisting of 21 end-products, ramifies into three categories: cement-bonded particle board, fiber cement board, and self-colored fiber cement board. At the present, two plants cater to the customers' demands: Bilkent and Ankara ASO factories. The former plant focuses on cement-bonded boards, whereas the latter concentrates on fiber cement products. As of 2015, annual production volume of Tepe Betopan, including both of its plants, raised up to 117,000 m3 (Tepe Betopan, 2019).

8.2 Problem Definition

Focusing on the stock keeping units (SKU) at the Bilkent factory, two main SKUs are manufactured on the existing production line: Betopan and Betopanplus. Though their chemical compositions are similar, Betopanplus is a non-patterned plated with two surfaces armored with natural minerals. The chemical similarity of the stock keeping units brings forth production convenience for initial phases by eliminating the setup time of the machines and processes. The Air Curing operation within the production system is critical to enhancing Betopan and Betopanplus, which takes place in semi-good inventory. However, the current policy suggests that the first semi-good departing from the pressing operation is located at the very back of the inventory. As manufacturing continues, new batches arriving at the inventory get moved ahead of the previously settled products. The current situation is represented in Figure 8.1. This operational decision brings forth problems during the removal of the products outside of the warehouse – to the next operation. In the ideal production schedule, the products are expected to spend 10 days inside the semi-finished warehouse. However, during the removal, the oldest products inside the warehouse are located at the end of the warehouse. As a result, the forklifts have to conduct repositioning between the palettes so that they can reach the older ones. The operators 5 could spend up to 2 hours on the repositioning, depending on the amount of pallets they need to relocate. Due to the operational inconvenience, the forklifts perpetually remove the products that are close to the entrance that did not fulfill the necessary amount of time inside the warehouse for hardening (showcased in Figure 8.1). Moreover, relocated palettes could interrupt the removal process for palettes deployed at other rows, because repositioning inside the warehouse is randomly operated. This increases the defective rate of the products due to two major reasons:

1. The products located at the back of the warehouse are ignored such that they spend more than 10 days and fracture due to major amount of loss in their dewiness.

2. The products located at the front of the warehouse do not spend solid 10 days such that as they are transferred to the next phases of the production, the defective rate increases due to minor amount of loss in their dewiness.

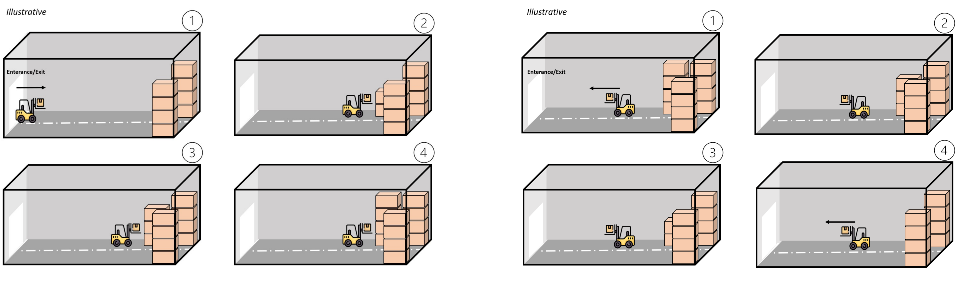


Figure 8.1: How loading and unloading are done

8.3 Proposed Solution Strategy

In order to develop a rigid solution plan, we established our presumptions about the system and the limitations we might encounter. Based on this, we charted out an approach for resolving the issue that we have utilized throughout the project.

8.3.1 Operational Dynamics

The semi-finished goods warehouse will only be used to store the pallets that contain outputs of the production line of Tepe Betopan. The number of days the semi-finished goods wait inside the warehouse affects the capacity of the warehouse because of the insufficiency explained beforehand. To handle each material inside the warehouse, the number of days for the goods to wait in the warehouse vary in between 8 and 12 days. In a two-hour period, the number of pallets brought into the warehouse by forklifts is four pallets. Every 15 minutes, a pallet is removed from the warehouse by forklifts.

8.3.2 Critical Assumptions

Additionally, assumptions are needed to reflect the real-life problem in operations research. This project holds the following assumptions: Forklifts and the production line operate non-stop. Unexpected conditions such as forklift breakdowns or line maintenances are neglected. The daily production inflow and outflow rates are 55 pallets. The addressing model will not be considering the cases where excess demand or special situations take place. Also, the production plan is the input for the warehouse inflow and it is known 15 days in advance. In the proposed solution there will be no limitation to aisle types. The forklifts should be moving in a rectilinear manner.

8.3.3 Capacity Maximization Model

For capacity maximization, we developed a mixed-integer linear programming model. The parameters of the model are solely measured from the warehouse by the project group. The model was solved on CPLEX. The output is the capacity of the warehouse with 475 pallets and three distinct bays. Overall, with the current inflow and outflow rates, 8.63 days of pallets could be stored via the output. The model is presented as follows:

- **Parameters**

  $I = \{1, 2\}$:Set of orientations that the pallets can be located

  (1: Short edge looking to the entrance)

  (2: Long edge looking to the entrance).

  $J = \{1, 2, 3\}$: Set of bays that will be opened in the warehouse.

  $P_l = 35$: Length of a pallet (+2×22.5cm tolerances included).

  $P_w = 16$: Width of a pallet (+2×12.5cm tolerances included).

$W = 58$: Recommended aisle width (580cm originally).

$G_L = 476$: Total vertical length of the grid (4760cm originally).

M: A large enough number (e.g. 1,000,000).

- **Decision Variables**

  y : y-axis coordinate (outer edge) of the cut-off point for the horizontal isle.

  b_{ji} : Number of pallets that could be stored in bay $j \in J$ of the warehouse using orientation $i \in I$.

  $$O_{ji} = \begin{cases} 1, & \text{If orientation } i \in I \text{ is utilized in bay } j \in J. \\ 0, & \text{Otherwise.} \end{cases}$$

- **Model**

$$\begin{aligned} \textbf{max} \quad & \left\lfloor \frac{48}{P_w} \right\rfloor b_{11} + \left\lfloor \frac{48}{P_l} \right\rfloor b_{12} + \left\lfloor \frac{G_W - (48 + W)}{P_w} \right\rfloor b_{21} \\ & + \left\lfloor \frac{G_W - (48 + W)}{P_l} \right\rfloor b_{22} + \left\lfloor \frac{G_W - W}{P_w} \right\rfloor b_{31} + \left\lfloor \frac{G_W - W}{P_l} \right\rfloor b_{32} \end{aligned}$$

s.t.

$$\begin{aligned} &(1) \quad \sum_{i \in I} O_{ji} \leq 1 \qquad \forall j \in J \\ &(2) \quad b_{j1} \leq \left(\frac{y - W}{P_l} \right) + M(1 - O_{j1}) \forall j \in J \\ &(3) \quad b_{j2} \leq \left(\frac{y - W}{P_w} \right) + M(1 - O_{j2}) \forall j \in J \\ &(4) \quad b_{ji} \leq M O_{ji} \qquad \forall j \in J, \forall i \in I \\ &(5) \quad W \leq y \leq G_L \text{ and } \in \mathbb{Z} \\ &(6) \quad b_{ji} \geq 0 \text{ and } \in \mathbb{Z} \qquad \forall j \in J, \forall i \in I \\ &(7) \quad O_{ji} \in \{0, 1\} \qquad \forall j \in J, \forall i \in I \end{aligned}$$

Explanations for Capacity Maximization

Objective function is to maximize the total number of pallets by multiplying the floor (function) of each bay's width over the pallet length (or width, depending on the orientation) with the number of pallets that could be stored in the bay. The explanation for the constraints is as follows: 1. There must only be one orientation used in each bay. 2.-7. Determines the number of rows for each bay, depending on the orientation. 8. No pallet is located in an unselected orientation. 9. Limits for decision variable y, which varies between 580cm and 4760cm. 10. b_{ji} being nonnegative variables. 11. O_{ji} being binary variables

8.3.4 Relocation Minimization Model

For the forklift movements minimization model, we were inspired by a mixed linear integer model of Akyüz and Lee (2014). Their container relocation model is concerned with minimizing the container's movements while reaching ones at lower positions. The problem tackles two-dimensional positioning. In our case, we changed the set of positions to three-dimensional. Our setting is that, for each row, the pallet with the hardest (takes the most time) to reach for, is 1 and goes up to the most salient pallet. In other words, the model does not interpret stacks, but rather, a 1-dimensional array of furthermost to closest ranks. The model was solved in Python, DOCplex library which incorporated two-dimensional sets. See the appendix for the model.

8.3.5 Methodological Approach

To confront the computational time inefficiency problem, a heuristic methodology was developed via Microsoft Excel - Visual Basic for Applications (VBA), which dictates an algorithm to handle the addressing of the pallets. The one and only consideration is to find an appropriate place for the pallets to be located at a time instance t. A time instance t corresponds to either an arrival of a pallet or a departure of a pallet. In other words, the set of time instances consists of the arrival times and departure times of the pallets, like the forklift movements minimization model that is explained above. If there is a departure at time instance t, the algorithm checks if there is a pallet that must be relocated. If a relocation occurs, the algorithm assigns a new position for the relocated pallet. If there is no relocation needed, we can remove the SKU outside of the semi-finished warehouse without a relocation. If the observation at time instance t is either a departure resulting in a relocation or an arrival of a new pallet, then the pallets will be located according to flowchart below.

8.4 Verification and Validation

It has been stated that the problem is NP-hard Akyüz and Lee (2014). In this regard, the model was solved with 32 slots (including 2 bays, the first bay of 2 columns with 12 slots, and the second bay of 1 column with 8 slots). Therefore, the model was run for this small instance of data to get verification. Hence, the model was verified with a small number of parameters, and the process took about 33 minutes. When the heuristic

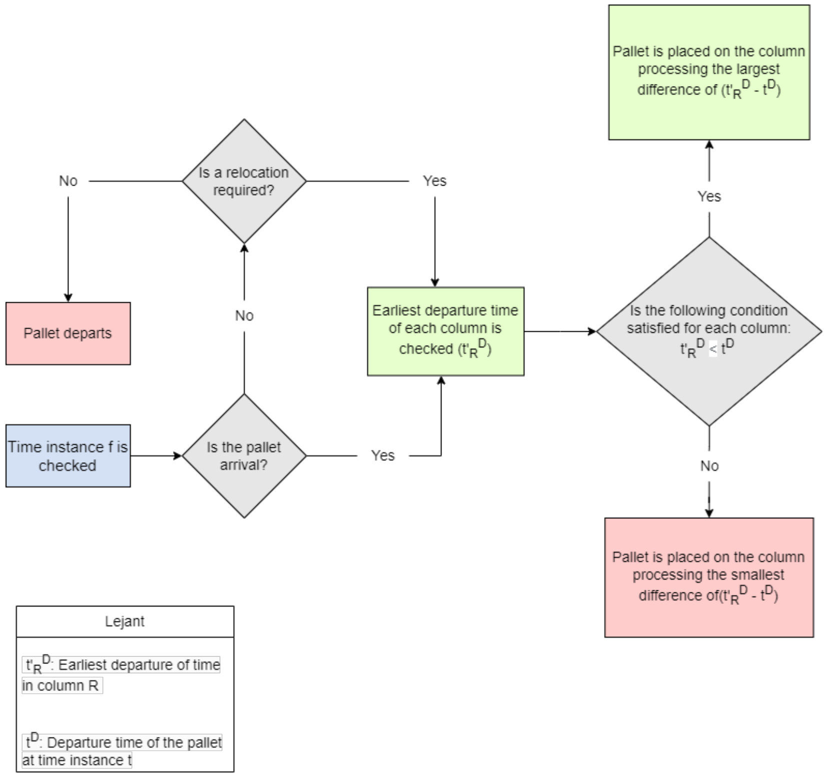


Figure 8.2: Methodological Approach Flow Chart

algorithm was run with the same data provided above, the yielded output was the same as each other, with 2 relocations.

8.5 Integration and Implementation

To create ease for the users (Industrial Advisor and company personnel) a graphical user interface for the algorithm was established (see Figure 8.3). During the pilot study of launching the algorithm, feedback is received from IA and company personnel on the regular basis.

For initialization, the inputs are taken from the user. The inputs are how many days will the pallets be produced to stay in the warehouse and the activities of the pallets.

While the algorithm is running, each activity of the pallets is checked. If the pallet departs with no need for relocation, it is removed immediately. Otherwise, each row's earliest pallet departure time is checked, and if in progress, the pallet's departure time is greater than the minimum time, it is located at the closest time. Otherwise, a feasible row with minimum deviation is chosen and an in-progress pallet is located there. The algorithm delivers two outputs: the total number of relocations during the time hori-

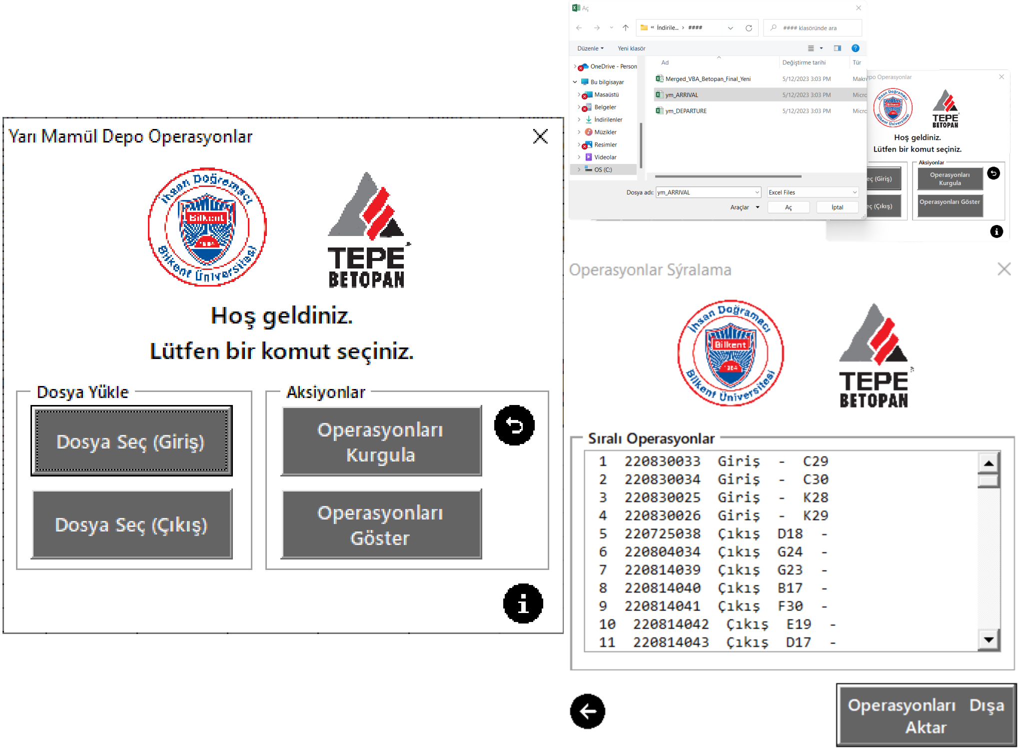


Figure 8.3: Graphical user interface main screen (left), arrival sheet selection (top right), view operations (bottom right)

zon and the logged activities of the pallets. The total number of relocations provides how well the algorithm performs compared to the current practice and logged activities dictate how to perform the arrival and departures of the pallets. The decision support system has three different spreadsheets in Excel: Arrival-Departure, Layout and Log.

8.5.1 Arrival-Departure

The Arrival-Departure spreadsheet is the one where user input of activities and corresponding time horizons are received. Once the user enters all activities of the pallets for a certain time period, the algorithm will start running. Under the assumption of 110 pallet activities per day, with 55 arrivals and 55 departures, each operation must be ordered chronologically. This button separates each day's activities and returns a precedence relationship. For example, assume the user inputs data for 10 days, starting from March 13. Then, March 13 pallet data are understood, and unique time instance values of March 13 are determined by our algorithm. This precedence relationship is returned from March 13 to March 23 for each day consecutively. This update provides ease on the user interface because it is now easier to keep track of activity relations.

8.5.2 Layout

The Layout spreadsheet shows the final layout of the warehouse at the end of the day according to the user input for visualization purposes. In the top left corner of this spreadsheet, within the green box, the user can see the total number of relocations done during that period. Stack-depth tells the user how many stacks are situated in a row. By the company's safety policy, at most four pallets could be stacked per depth. Hence, a row with a stack depth of 7 can at most have 28 pallets.

8.5.3 Log

The Log spreadsheet dictates in which precedence should the pallets move, specifying which action class (arrival, relocation, departure) each operation belongs to with their previous and current locations at the warehouse. All of the operations from the previous spreadsheet are handled with the minimum number of relocations.

Giriş-Çıkış Sayfası (Arrival-Departure Sheet)				Operasyon Çizelgesi (Log Sheet)				
Parti No.	SKU	Giriş Tarihi	Çıkış Tarihi	Palet No.	Action	Previous Location	Current Location	Operation No.
220725038	BETOPAN **16x1250x2800mm**	5/10/2022	5/18/2022	*220830033*	Arrival	-	C29	1
220804034	BETOPAN **16x1250x2800mm**	5/10/2022	5/18/2022	*220830034*	Arrival	-	C30	2
220814039	BETOPAN **16x1250x2800mm**	5/10/2022	5/18/2022	*220830025*	Arrival	-	K28	3
220814040	BETOPAN **16x1250x2800mm**	5/10/2022	5/18/2022	*220830026*	Arrival	-	K29	4
220814041	BETOPAN **16x1250x2800mm**	5/10/2022	5/18/2022	*220725038*	Departure	D18	-	5
220814042	BETOPAN **16x1250x2800mm**	5/10/2022	5/18/2022	*220804034*	Departure	G24	-	6
220814043	BETOPAN **16x1250x2800mm**	5/10/2022	5/18/2022	*220814039*	Departure	G23	-	7
220815003	BETOPAN **16x1250x2900mm**	5/10/2022	5/18/2022	*220814040*	Departure	B17	-	8
220815004	BETOPAN **18x1250x2500mm**	5/10/2022	5/18/2022	*220814041*	Departure	F30	-	9
220815005	BETOPAN **18x1250x2500mm**	5/10/2022	5/18/2022	*220814042*	Departure	E19	-	10
220815006	BETOPAN **18x1250x2500mm**	5/10/2022	5/18/2022	*220814043*	Departure	D17	-	11
220817042	BETOPAN **18x1250x2500mm**	5/10/2022	5/18/2022	*220815003*	Departure	H28	-	12
220817043	BETOPAN **18x1250x2500mm**	5/10/2022	5/18/2022	*220830027*	Arrival	-	E19	13
220817044	BETOPAN **18x1250x2500mm**	5/10/2022	5/18/2022	*220830028*	Arrival	-	E20	14
220818049	BETOPAN **18x1250x2500mm**	5/10/2022	5/18/2022	*220830029*	Arrival	-	E21	15
220819001	BETOPAN **18x1250x2500mm**	5/10/2022	5/18/2022	*220830030*	Arrival	-	E22	16
220819002	BETOPAN **18x1250x2500mm**	5/10/2022	5/18/2022	*220815004*	Departure	F29	-	17
220819003	BETOPAN **18x1250x2500mm**	5/10/2022	5/18/2022	*220815005*	Departure	G22	-	18
220820021	BETOPAN **18x1250x2500mm**	5/10/2022	5/18/2022	*220815006*	Departure	G21	-	19

Dizilim Sayfası (Layout Sheet)

174	Row 1	Row 2	Row 3	Row 4	Row 5	Row 6	Row 7	Row 8
Stack-Depth	7	7	7	7	7	7	7	7
Stack-1	100140	100268	100381	100175	100248	100165	100330	100221
Stack-2	100088	100211	100370	100357	100376	100128	100281	100397
Stack-3	100270	100363	100299	100203	100220	100137	100226	100296
Stack-4	100119	100242	100264	100368	100071	100384	100239	100276
Stack-5	100109	100153	100328	100393	100056	100127	100267	100257
Stack-6	100108	100308	100085	100096	100365	100201	100290	100286
Stack-7	100318	100241	100224	100364	100385	100190	100081	100112
Stack-8	100254	100151	100142	100091	100095	100225	100344	100317
Stack-9	100212	100115	100100	100329	100236	100104	100300	100059
Stack-10	100292	100347	100078	100389	100098	100064	100256	100144
Stack-11	100072	100097	100259	100336	100325	100288	100352	100306
Stack-12	100249	100246		100354	100185	100230	100077	100274
Stack-13	220830050	100372		100111	100158	100110	100265	100125
Stack-14	220831001	100174			100062	100076	100087	100157
Stack-15	220830040	100237			100213	100260	100134	100209

Figure 8.4: GUI Sheets Example

8.6 Benefits to the Company

With the decision support system on VBA, the main benefit that is aimed to be achieved is a decrease in the number of relocated pallets inside the warehouse for a specific time period.

The current relocation number of the pallets inside the warehouse changes daily due to the random assignment of the pallets. Since a proper observation could not be obtained, the expert opinion was consulted for the number of relocations for one day. The warehouse chief stated that on average, 60 relocations per day take place. The heuristic algorithm, on the other hand, results in 32 relocations per day, which translates into an improvement in the number of relocations, possible improvement in the forklift utilization, and hence, company's profit. Another benefit the company is going to receive is the location logs of the pallets. Previously, the company was not recording the pallets' locations but only providing the pallets with a label stuck to them. This policy was not supporting an integrated system of data management. With the decision support system, the users can keep track of which pallet is located in the warehouse and which pallets were relocated from which prior position to the subsequent.

8.7 Pilot Study

During the pilot study, it was our aim to minimize the number of relocations inside the warehouse with real semi-finished products data. Since the products are cumbersome to deploy, we did not initialize the locations of pallets and started with the random layout that was already installed in the factory. We prepared the algorithm with 9th of May data: arrival and departure. On May 9th, there were 355 pallets residing, 60 pallets arriving and 35 were departing. The scope of the project only regarded the pallets inside the warehouse, so other arrival/departure pallets operating but locating outside this area were not considered. In conclusion, our decision support system provided the company with 0 relocations during one day, which could be considered a decent improvement compared to the randomly fluctuating number of relocations.

8.8 Conclusion

Air curing is the most critical process in the production system. The process ensures that semi-finished goods (pallets) mature within the specified dates in the semi-finished goods warehouse. Examining the current situation, the objective is to evacuate the semi-goods which are thought to be completed. However, this policy is not efficient because the locations of the pallets are not known and the goods may stay in the warehouse at varying times. To tackle this situation, the pallets which are thought to be done are taken out of the warehouse, and during this process, other pallets are randomly distributed in the warehouse. The random policy results in the relocation of the pallets. The project intends to resolve the relocation problem. With the capacity maximization model and the heuristic algorithm which was proposed in the report, we managed to capture a decrease in the relocation number per day. Furthermore, with the proposed decision support system, company personnel could keep tracking the locations and logs of the pallets.

Bibliography

Akyüz, M. and C. Lee (2014). A mathematical formulation and efficient heuristics for the dynamic container relocation problem. Volume 61, pp. 101–118. Wiley Online Library.

Appendix: Relocation Minimization Model

- **Parameters**

J: Set of pallets.

K: Set of positions.

T_j: Discrete times elapsed between each pallet ($j \in J$)'s arrival and departure.

T: Set of the unions of T_j ($j \in J$).

W: Set of columns ($\forall k \in K$).

O: Set of pallets that do not require any other pallet's relocation during removal.

$$\delta_{kk'} = \begin{cases} 1, & \text{If } k \text{ is located under } k' \ (\forall k, k' \in K). \\ 0, & \text{Otherwise.} \end{cases}$$

M: Set of stacks.

- **Decision Variables**

$$x_{jkt} = \begin{cases} 1, & \text{If pallet } j \in J \text{ is located in slot } k \in K \text{ at time } t. \\ 0, & \text{Otherwise.} \end{cases}$$

$$y_{jt} = \begin{cases} 1, & \text{If pallet } j \in J \text{ is relocated in time } t. \\ 0, & \text{Otherwise.} \end{cases}$$

- **Model**

$$\min \sum_{t \in \{T_j - T_j^d\}} \sum_{j \in J} y_{jt}$$

s.t.

$$(1) \quad \sum_{j \in J: t \in T_j} x_{jkt} \leq 1 \qquad \forall k \in K, \forall t \in T$$

$$(2) \quad \sum_{k \in K} x_{jkt} = 1 \qquad \forall t \in T_j, \forall j \in J$$

$$(3) \quad \delta_{kk+1} \left(\sum_{j \in J: t \in T_j} x_{jk+1t} - \sum_{j \in J: t \in T_j} x_{jkt} \right) \leq 0, \forall k \in K \setminus O, \forall t \in T$$

$$(4) \delta_{kk+1}(x_{jk+1t} + x_{jkt}) - y_{jt} \leq 1 \qquad \forall k \in K \setminus O, t = T_j^d, \forall j \neq j' \in J$$

$$(5) \quad x_{jkt} - \sum_{\substack{j' \in J \setminus \{j\}: \\ t = T_j^d}} \sum_{k' \in K} \delta_{k'k} x_{j'k't} \leq 1 - y_{jt}, \forall k \in K, \forall t \in T_j, \forall j \in J$$

$$(6) \quad \left. \begin{array}{l} x_{jkt} - x_{jkt+1} \leq y_{jt} \\ x_{jkt+1} - x_{jkt} \leq y_{jt} \end{array} \right\} \qquad \forall k \in K, \forall t \in \{T_j - T_j^d\}, \forall j \in J$$

$$(7) \quad x_{jkt} + x_{jkt+1} + \sum_{k' \in K} \delta_{k'k} x_{jk't+1} + \sum_{k' \in K} \delta_{kk'} x_{jk't+1} \leq 2 - y_{jt}$$

$$\forall k \in K, \forall t \in \{T_j - T_j^d\}, \forall j \in J$$

$$(8) \quad x_{jkt} + \sum_{k' \in K} \delta_{kk''} x_{j'k''t+1} + x_{jk't+1} + \sum_{k'' \in K} \delta_{k'k''} x_{j'k''t+1} + y_{jt} + y_{j't} \leq 5$$

$$\forall k, k' \in \{K - O\}, W(k) \neq W(k'), t \in \{T_j - T_j^d\} \cap \{T_{j'} - T_{j'}^d\},$$

$$\forall j \neq j' \in J$$

$$(9) \quad x_{jkt} \in \{0, 1\} \qquad \forall k \in K, \forall t \in T_j, \forall j \in J$$

$$(10) \quad y_{jt} \in \{0, 1\} \qquad \forall t \in \{T_j - T_j^d\}, \forall j \in J$$

Objective function is to Minimize the total number of pallet relocations other than the departing pallet itself over all of the time periods. Constraint 1. At most one pallet could be located in a single slot. 2. Each pallet must be placed to a slot in its time period. 3. Slot k + 1 is full if and only if slot k is full. This constraint is valid for the pallets located in the same column. 4.-5. Relocation constraints: 4 stands for if a pallet j' is located under another pallet j and the departure time of j' has come, then j must be relocated. 5 stands for there must be no relocations done for a pallet j', which is located under j while j is leaving. 6. If a pallet j is not leaving while discrete time elapses (i.e., some other pallets are leaving and j is not being relocated), then it remains in the same location (slot). 7. If a pallet j is being relocated but not being departed from the warehouse, it must not be placed in the same column at the next time period. 8. If there are two pallets being relocated because of another pallet's departure under them, then the two pallets' positions must change. This is to minimize any potential relocations. Note that once the pallets are relocated, their position is fixed for that time interval. 9-10. x_{jkt} and y_{jt} being binary variables.

Depo Yönetim Sistemi ve Araç Yükleme Optimizasyonu | 9

Bakioğlu Holding-Polibak

Proje Ekibi

Elanur Çöplü, Ayça Erdem, Öykü İman, Roshaan Abbas Jaffery, Sehar İrfan Mangi, Zeynep Nalbant, Umut Ceyda Serhat

Şirket Danışmanı
Beste Yıldız
Proje Geliştirme Mühendisi

Akademik Danışman
Prof. Dr. Oya Ekin Karaşan
Endüstri Mühendisliği Bölümü

Özet

Bakioğlu Holding bünyesinde olan Polibak Plastik Film Sanayi ve Ticaret A.Ş. için yapılan depo yönetim sistemi ve sevkiyat planlama araç kapasitelerinin maksimize edilmesi projeleri için öne sürülen çözümler ve süreçleri incelenmektedir. Projelerimizin amaçları eleçleme sürecinin hızlandırılması ve sevkiyat planlama sürecinin kısaltılmasıdır. Matematiksel modellerimiz Python CBC ve CP-SAT programları kullanılarak çözülmüştür.

Anahtar Sözcükler: Depo Yerleşim, İlişki Madenciliği, Sevkiyat Planlama, Araç içi optimizasyonu, Karma Tamsayı Programlama

Warehouse Management System and Vehicle Loading Optimization

Abstract

This document provides an overview of the development of a warehouse management system and a vehicle loading optimization system for Polibak Plastik Film Sanayi ve Ticaret A.S., a subsidiary of Bakioglu Holding. The proposed solutions have been designed to improve the efficiency and effectiveness of the company's logistics operations, with a focus on enhancing warehouse management and maximizing the utilization of vehicle capacities while reducing costs. The mathematical models developed to support these solutions have been solved in the Python programming language, utilizing the open-source CBC and CP-SAT solvers.

Keywords: Storage Location Assignment Problem, Association Mining, Vehicle Loading Optimization, Mixed Integer Programming

9.1 Company Description

Polibak, as part of Bakioglu Holding, is one of Turkey's leading plastic film manufacturers, producing BOPP (Biaxially Oriented Polypropylene) and Cast Polypropylene (CPP) films since 1994. It has a total annual production capacity of 180,000 tonnes of BOPP and 36,000 tonnes of metalized BOPP film. Polibak currently exports to multiple countries around the world (Polibak, 2022a).

9.2 System Analysis

9.2.1 Warehouse Management System

Polibak currently has three storage units in their factory with capacities of 300 tonnes, 800 tonnes, and 1600 tonnes, respectively. There is an average of 300 tonnes (500 pallets) of daily product entry into the warehouses and an output of 350 tonnes (550-600 pallets) of products. Four different categories of materials flow through Polibak's warehouses: raw materials, finished products, semi-finished products, and packaging (Polibak, 2022b).

The warehouses have no shelving system and no more than two pallets stacked upon each other, resulting in only 45% (including vertical space) of space being utilized. The products are stored in bulk based on customer orders and the forklift operators handle product allocations and picking, with no formal algorithmic system in place. Polibak intends to build a new warehouse, a 220-meter by 45-meter space with an AS/RS double-deep shelving system with 13 shelf levels and 5 corridors, each with its own crane,

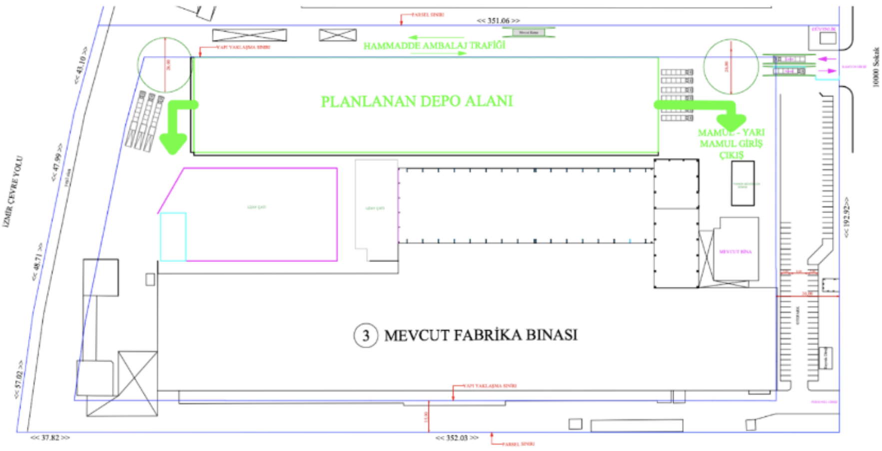


Figure 9.1: Planned ASRS Layout

to store all four categories of products. The layout can be seen in Figure 9.1. The main focus of the project is to create a decision support system that will allocate each incoming product of raw materials and end-product categories in the warehouse such that order-picking times are minimized and warehouse space utilization is maximized (Polibak, 2022a).

9.2.2 Vehicle Loading Optimization

A vehicle is generally loaded with a single customer's orders, but in some local cases, a vehicle may accommodate more than one customer's orders. The data required for loading items to be placed in the vehicle is given to the Shipping Department, and a placement plan, for the items in each order, is made by hand. The vehicle sizes vary according to the shipment size, location, and vehicle availability at shipment time. The three main types of vehicles are trucks, lorries, and containers. The lorries are small trucks and containers that can be separated and placed on a ship to be sent overseas. Depending on the type of vehicle and shipment destination, certain rules may need to be followed, for example, different weight limits and whether or not the first axle area of the vehicle has a weight limit. The objective is to create an efficient algorithm of positioning orders onto vehicles, such that their space utilization is maximized with specified constraints.

9.3 Proposed System

9.3.1 Warehouse Management System

The project is based on the article "The adaptive approach for storage assignment by mining data of warehouse management systems for distribution centers" (Chiang et al., 2011) which has been adapted to our project needs. This paper proposes a heuristic approach using an association index called the AIX, which is used to formulate a storage location assignment model by applying association rule mining. The AIX evaluates the fitness (higher value corresponds to better fitness) between incoming products and unassigned storage locations by placing frequently ordered products nearer to the exits and, by doing so, enhancing the efficiency of order picking in terms of time taken or distance covered by the order picker.

Critical Assumptions and Constraints

We let pallet sizes determine the product type since both can represent each other in the model. In the warehouse, there are two designated gates on each end. The right side of the warehouse needs to house finished products while raw materials must be placed on the left side. Moreover, products with lengths larger than 145 cm must be placed in the last corridor. Products with heights larger than 210 cm should be placed on the top shelf while products with heights less than 140 cm should be placed on the first shelf.

Development of The Association Index (AIX)

Polibak's warehouse data spanning 21 months has been used to generate the association index (AIX) that requires the weighted support count (found using the apriori algorithm), the turnover rates of products, and the storage location distance from the beginning of the first cell of the corresponding aisle near the exit found using Euclidean distance. This is designed for an AS/RS system where the crane simultaneously moves in the x and y axes at the same speed along each axis. The double-deep shelving system has also been integrated into the development of the AIX, where the inner depth is prioritized over the outer depth.

Running the Mathematical Model with AIX

The generated AIX matrix is integrated into the objective function of our storage location assignment binary integer programming model assigning products to locations such that the AIX value is maximized since a higher AIX value between a certain location and a product indicates more suitability compared to other competing products for the same location. The model is coded and run on Python using CBC as an open-source mixed-integer program (MIP) mathematical solver (Coin-OR, 2023). The results are in the form of 4D coordinates (corridor, column, shelf level, double-deep level), which indicate where the incoming item is well placed.

Mathematical Model of Warehouse Management System

Indices

- $R = i, j, l, d : r$ *is a set that indicates the location of a cell by defining an aisle (i), column (j), shelf level (l) and depth (d)*

- $K = 1, .., k : K$ *is the set of product types*

- $A = 5, j, l, d : A$ *is the set of locations where product types with lengths equal or higher than 145 cm need to be placed*

- $B = i, j, 1, d : B$ *is the set of locations where product types with heights equal or less than 140 cm can be placed*

- $C = i, j, 13, d : C$ *is the set of locations where product types with heights equal or higher than 210 cm can be placed*

Parameters

- l_k : *The length of product type k in cm*

- p_k : *The number of items with product type k*

- AIX_{rk} : *Estimator of the relationship between location r and product type k*

- $AIX_{rk} : \frac{(\sum_{k'} wsupc_{kk'}) \cdot T_k}{D_r}$

- $wsupc_{kk'}$: *The weighted support count between product type k and allocated product type k'*

- T_k : *The turnover rate of product type k*

- D_r : *The distance between location r and the outbound exit*

Decision Variables

$$x_{rk} = \begin{cases} 1, & \text{If the product type k is put in location r} \\ 0, & \text{otherwise} \end{cases}$$

Objective Function and Constraints

$$\text{maximize} \sum_r^R \sum_k^K x_{rk} AIX_{rk}$$

subject to,

$$\sum_{r \in R} x_{rk} = p_k \qquad \forall k \in K \tag{9.1}$$

$$\sum_{k \in K} x_{rk} \leq 3 \qquad \forall r \in R \tag{9.2}$$

$$\sum_{k \in K} l_k * x_{rk} \leq 290 \qquad \forall r \in R \tag{9.3}$$

$$\sum_{r \in A} x_{rk} = p_k \qquad \forall k \mid l_k \geq 145 \tag{9.4}$$

$$\sum_{r \in B} x_{rk} = p_k \qquad \forall k \mid h_k \leq 140 \tag{9.5}$$

$$\sum_{r \in C} x_{rk} = p_k \qquad \forall k \mid h_k \geq 210 \tag{9.6}$$

$$x_{rk} \in \{0, 1\} \qquad \forall k \in K, \forall r \in R \tag{9.7}$$

9.3.2 Vehicle Loading Optimization

Critical Assumptions and Constraints

Aside from the single bin-packing MIP assumptions, the model is also based on the truck being a rectangular container with fixed dimensions and the pallets being rectangular boxes with their specified dimensions (x, y, and z values) given in the Cartesian coordinate system. Pallets can be rotated in the x and z directions but not in the y direction since they are stacked during transit, and rotation may cause damage to the products.

For heavier vehicles, we must consider the limitation that the front axle may carry only 4.5 tonnes without surpassing 3.5 meters in length in the y-dimension. Aside from the pallets placed on the truck's front axle, each pallet may have only one pallet piled on top. The pallet piled on top must not be more than 20 cm smaller than the pallet it is stacked on otherwise there can be an imbalance between the pallets.

Development of Vehicle Loading Model

The model is based on the paper 'Container packing problem with balance constraints'Moon and Nguyen (2013), adapted to Polibak's set constraints. In Excel, the sizes of the objects to be loaded are entered as a 3D matrix based on their item numbers, dimensions, and rotations. Their volume and weight data are also taken from the Company Order Excel file. The vehicle's far left corner is the coordinate system's origin (0,0,0). The model was initially solved using the Xpress Solver and also adapted to Python and solved using Gurobi. To achieve UI integration, obtain graphic results, and avoid solver costs and licenses based on company requirements, the MIP model programmed in Python was solved using an open-source solver library called CP-SAT solver OR-Tools by Google-Developers (2023), and the 3D graph output of the model shows which items are selected to be loaded onto the truck and the coordinates of their positions in the truck. The items are labeled with their radius values as the company uses this

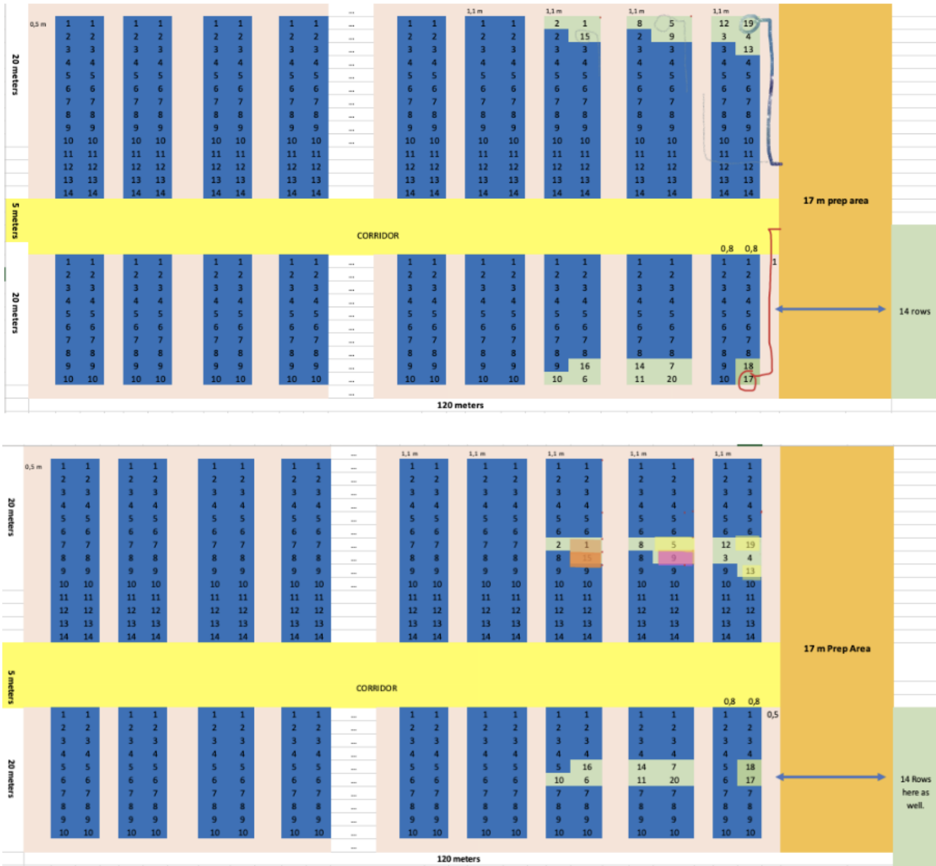


Figure 9.2: Current layout used for benchmarking: empty (top) and half-full

data to identify them in transportation.

Mathematical Model of Vehicle Optimization

The objective function is to maximize the total volume utilized while loading the vehicle, i.e., maximize the number of packed items or

$$\text{maximize } \sum_{i=1}^{n} V_i X_i.$$

The model works as a MIP 3D Knapsack problem and has been given in the appendix. For the model, there are five main constraint groups: (1) preventing overlap of items and vehicle boundaries and making sure that a direction according to an item is not assigned if another item is not present, (2) weight limits of the vehicle, (3) the 20 cm rule which assigns only one item on top of another item and ensures that the of the pallets on top are not smaller than 20 cm on each side. (4) the axle constraints which prevent overload on the front axle of the vehicles and is necessary only for certain shipments, (5) the rotation constraint ensures that the items on top of each

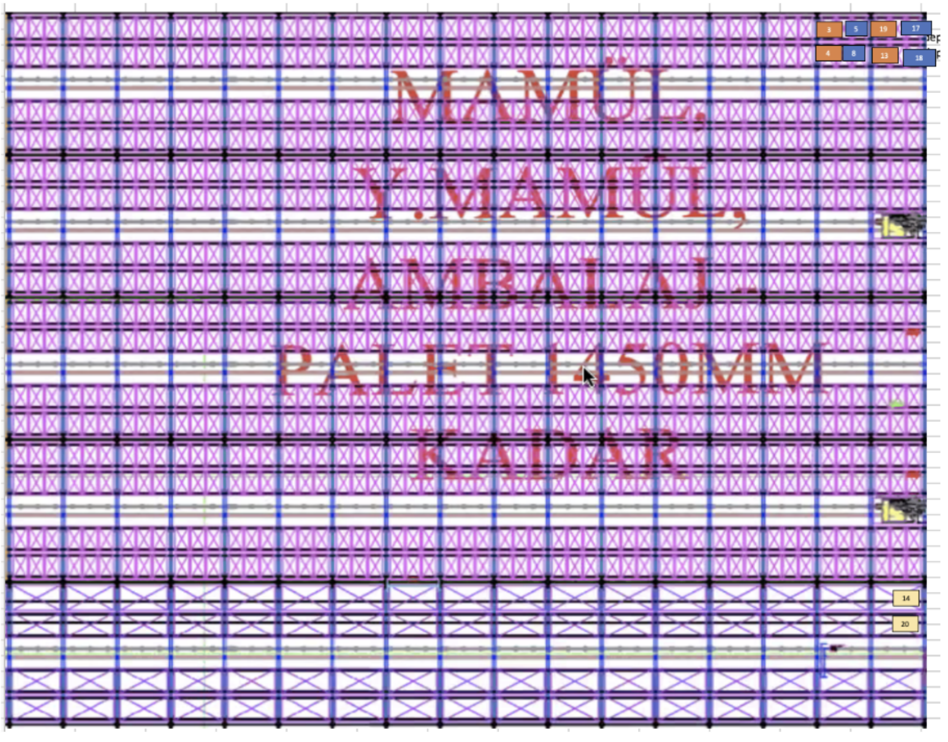


Figure 9.3: Planned layout for end product locations

other have the same rotation. This constraint is devised for the purpose of balance between the stacked pallets.

9.3.3 Validation and Benchmarking

Warehouse Management System

Following the company's test for validation and the dataset of 20 products provided accordingly, validation was carried out in three different layouts. The first layout was the current system based on customer batches and it assumes the whole depot is empty while the second layout consists of the current depot being half-full and then stocked with products. Both layouts are shown in Figure 9.2.

The third layout used was the new depot - the AS/RS system and is given in Figure 9.3. Since the dimensions of these layouts differ, the distances traveled were adjusted by taking a ratio with respect to the whole width or length of that space. The products being allocated and the subsequent results can be seen in Figure 9.4.

We observed the percentage and time improvements in the two different systems (for three different scenarios) for the same product. For example, in the empty depot compared to the AS/RS system, for SKU 15 we can observe

Empty Depot							
Old Layout	Distance	adjusted		New Layout	Distance	adjusted	Distance Improvement
sku 19	21,65 m	0,180		sku 19	1,45 m	0,028	84%
sku 5	23,75 m	0,198		sku 5	4,35 m	0,084	58%
sku 17	21,65 m	0,180		sku 17	1,45 m	0,028	84%
sku 15	24,5 m	0,201		sku 15	5,30 m	0,102	49%
Old Layout	Time			New Layout	Time		Time Improvement
sku 19	0,18619	min		sku 19	0,009715	min	95%
sku 5	0,20425	min		sku 5	0,029145	min	86%
sku 17	0,18619	min		sku 17	0,009715	min	95%
sku 15	0,2107	min		sku 15	0,03551	min	83%

Half Full Depot							
Old Layout	Distance	adjusted		New Layout	Distance	adjusted	Distance Improvement
sku 15	16,4	0,137		sku 15	5,30 m	0,102	26%
sku 9	14,3	0,119		sku 9	3,2 m	0,062	48%
sku 13	10,85	0,090		sku 13	1,45 m	0,028	69%
sku 19	13,55	0,113		sku 19	1,45 m	0,028	75%
sku 5	15,4	0,128		sku 5	4,35 m	0,084	35%
Old Layout	Time			New Layout	Time		Time Improvement
sku 15	0,14104	min		sku 15	0,03551	min	75%
sku 9	0,20898	min		sku 9	0,02144	min	90%
sku 13	0,09331	min		sku 13	0,009715	min	90%
sku 19	0,11653	min		sku 19	0,009715	min	92%
sku 5	0,13244	dk		sku 5	0,029145	dk	78%

Figure 9.4: Benchmarking Results for Empty and Half full Depot

a distance improvement of 45%. In the half-empty depot compared to the AS/RS system, SKU 5 has a 35% distance improvement. It was concluded that not only does the developed warehouse management system satisfy the company's requirements and constraints but also greatly improves product allocation efficiency, taking roughly a couple of minutes to allocate products to an empty layout by running the code.

It must also be noted that the quantified improvements are very high due to the layouts being compared (the current and new layouts in Figures 9.2 and 9.4, respectively) having different physical environments.

Vehicle Loading Optimization

The model works within all specified constraints and may be used to place 100 products that are in line with the real system. The performance indicators include allocation time and vehicle capacity utilization. According

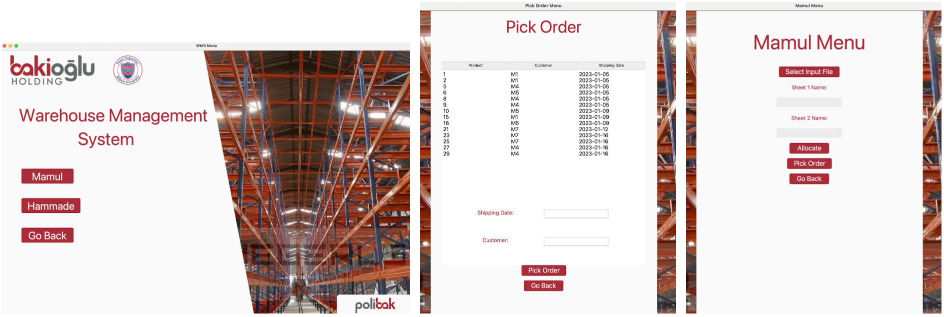


Figure 9.5: Decision Support System for WMS

to the Shipping Department's Planning Executive, making the handwritten loading plan can take up to 20 minutes, but the vehicle loading optimization system output is obtained in less than 5 minutes. The same number of objects are distributed in a similar manner. Moreover, for the model coded in Fico Xpress, results are obtained in under 5 minutes for larger data sets (over 70 items) and between 1-4 minutes for medium-sized data sets. Overall, with the model, we observed roughly a 75% reduction in the time required to compute product placement.

9.3.4 Pilot Study and Implementation

Warehouse Management System

Both the implementation and pilot study stages of this new system are expected to be done in the future once the new warehouse is built. The decision support system is built in Python using the Tkinter library (see Appendix F). The main objective of putting items ordered together closer to each other or putting products with high turnover rates nearer to the door is the same regardless of the warehouse's physical environment. This will be done using a stand-alone desktop application in the new depot. As mentioned, the pilot study and implementation cannot be done at this stage for this project. We will provide a user manual to the company in order for them to understand the system and the code which they can use when the new warehouse is built.

Vehicle Loading Optimization

The model is expected to be incorporated by the company into a stand-alone desktop program for vehicle loading, with 3D display written in Python Mayavi package (Python-Software-Foundation, 2023). The pilot study and implementation stages were conducted in the Polibak headquarters in Izmir on 9 May 2023. The model was examined by Polibak project team followed by a meeting with IT department regarding the restrictions of the firewall

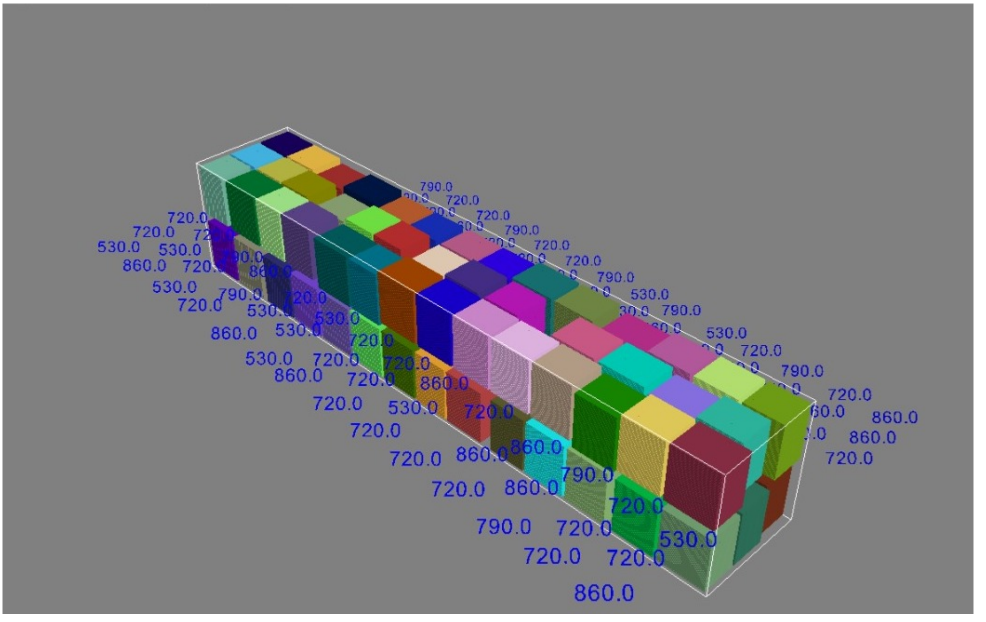


Figure 9.6: 3D Output of Vehicle Loading System

in the company. With the help of the IT department, VS Code was downloaded into the Transportation Department computer. Necessary packages were imported as well as different required images and extra programs were also executed for the decision support system. Initially, a past data set was plugged into the model program and the 3D output was analyzed. After confirming with the company, a new data set for the 9th of May transportation was run through the code. It took approximately 5 minutes for the code to run and then render a 3D model; see Figure 9.6. A handmade plan was also written, which took approximately 20 minutes. After comparing the results and receiving confirmation, the plan was taken to the transportation area, and the new vehicle was successfully loaded according to the code output plan. The user interface can be seen in Figure 9.7.

9.3.5 Benefits to the Company

Warehouse Management System

We minimize human error in product allocation. We reduce order-picking time and warehouse management costs; e.g, labor. AIX is based on a machine learning algorithm; dynamic and improves with the accumulation of more data, giving better AIX values for product allocation and resulting in more effective placement

Figure 9.7: Decision Support System for Vehicle Loading Optimization

Vehicle Loading Optimization

Time taken to get order allocations in vehicles is reduced by 75%. Human error from manual computations will be reduced. Excessive loading and unloading will be avoided in case of incorrect allocations. Maximum number of items in truck reduces number of vehicles required, lowers logistical costs.

9.3.6 Conclusion and Future Work

Warehouse Management System

Our proposed model has additional constraints specified by Polibak in addition to the constraints laid out in the research paper, which solves the model as a simple storage assignment problem with binary integers. Moreover, the development of the AIX in the project also considers a 3D space accessed by cranes. This project provides a dynamic heuristic approach for not only a warehouse with an AS/RS (Automated Storage and Retrieval System) but is also adaptable to different warehouse layouts. Companies can extract useful information from the warehouse data increasing the efficiency of the warehouse operations and cutting down on logistics costs. This decision support system is designed to be accessible and easy to understand while keeping the records of system transactions. The AIX model can take on further parameters that can be adapted to different purposes as well by the user companies to keep track of small and large warehouses with different requirements.

Vehicle Loading Optimization

This project aims to maximize vehicle space utilization by placing pallets in the most efficient way possible. It can also be adapted to various vehicle dimensions, weight limits, and pallet sizes. Instead of creating a plan by hand, the company will be able to use an orderly method of loading pallets. This model will reduce the procedure's planning time, as the output of the model specifies where and how many pallets should be placed onto the vehicle and helps avoid difficulties with pallet positioning. Since the model maximizes the number of items that can be packed onto a single vehicle, the total number of trucks used by the firm will be reduced, resulting in saving logistics costs. In the future, the algorithm can be further built by making the model consider different loading places and adding family groups reflecting different customer orders.

Bibliography

Chiang, D. M.-H., C.-P. Lin, and M.-C. Chen (2011). The adaptive approach for storage assignment by mining data of warehouse management system for distribution centres. *Enterprise Information Systems 5*(2), 219–234.

Coin-OR (2023). Cbc: Open-source mixed integer programming solver. https://github.com/coin-or/Cbc.

Google-Developers (2023). Google Optimization Google OR-Tools Constraint Programming Solver. Accessed on April 20, 2023.

Moon, I. and T. V. Nguyen (2013). Container packing problem with balance constraints. *OR Spectrum 36*(4), 837–878.

Polibak (2022a). Polibak sirket profili. http://www.polibak.com.tr/sirket-profili.aspx.

Polibak, M. (2022b). Polibak pazarlar. http://www.polibak.com.tr/pazarlar.aspx.

Python-Software-Foundation (2023). Mayavi. https://pypi.org/project/mayavi/.

Appendix: Vehicle Optimization Model

Mathematical Model(s)

Indices

- $i = 1,..,n$: *Total Number of Pallets in That Order*
- $j = 1,..,n$: *Total Number of Pallets in That Order*
- $d = 1,2,3$: *Dimensions of x, y, z*
- $r = 1,2$: *Rotation of position*

Parameters

- w_i : *The weight of the item i*
- V_i : *Volume of the i^{th} item*
- S_{ird} : *Size of item i rotated with rotation r along dimension d*
- S_d : *Size of container along dimension d*
- $V_{container}$: *Volume of the container*
- W_{lim} : *Weight limit of truck*

Decision Variables

$$X_i : \begin{cases} 1, & \text{If item i is packed into the container} \\ 0, & \text{otherwise} \end{cases}$$

$$R_{ir} : \begin{cases} 1, & \text{If item i is rotated with rotation r} \\ 0, & \text{otherwise} \end{cases}$$

$$Y^+_{ijd} : \begin{cases} 1, & \text{If item i is packed and item j is also packed along dimension d according to i} \\ 0, & \text{otherwise} \end{cases}$$

$$Y^-_{ijd} : \begin{cases} 1, & \text{If item j is packed and item i is also packed along dimension d according to j} \\ 0, & \text{otherwise} \end{cases}$$

$$A_i : \begin{cases} 1, & \text{If the item i put in the vehicle is also on the first axle area of the vehicle} \\ 0, & \text{otherwise} \end{cases}$$

$$B_i : \begin{cases} 1, & \text{If } A_i \text{ and } X_i \text{ is both 1} \\ 0, & \text{otherwise} \end{cases}$$

$$U_{ij} : \begin{cases} 1, & \text{If either i is on top of j or j is on top of i.} \\ 0, & \text{otherwise} \end{cases}$$

$$um_{ij} : \begin{cases} 1, & \text{If the decision variable } Y^-_{ijz} \text{ is 1 and all the other Y's are 0} \\ 0, & \text{otherwise} \end{cases}$$

$$up_{ij} : \begin{cases} 1, & \text{If the decision variable } Y^+_{ijz} \text{ is 1 and all the other Y's are 0} \\ 0, & \text{otherwise} \end{cases}$$

C_{id} : Coordinate of item i along dimension d when placed in the vehicle

Objective Function and Constraints

$$\text{maximize } \sum_{i=1}^{n} V_i X_i$$

subject to,

$$\sum_{i=1}^{n} V_i X_i \leq V_{container} \quad (1)$$

$$C_{id} \leq S_d X_i \quad \forall i \in 1,..,n \forall d \in x,y,z \quad (2)$$

$$C_{id} \geq 0 \quad \forall i \in 1,..,n \forall d \in x,y,z \quad (3)$$

$$\sum_{r=1}^{2} R_{ir} = X_i \quad \forall i \in 1,..,n \quad (4)$$

$$C_{id} + \sum_{r=1}^{2} S_{ird} R_{ir} \leq S_d \quad \forall i \in 1,..,n \forall d \in x,y,z \quad (5)$$

$$C_{id} + \sum_{r=1}^{2} S_{ird} R_{ir} - S_d(1 - Y^+_{ijd}) \leq C_{jd} \quad \forall i,j \in 1,..,n \forall d \in x,y,z \quad \forall i < j \quad (6)$$

$$C_{jd} + \sum_{r=1}^{2} S_{jrd} R_{jr} - S_d(1 - Y^-_{ijd}) \leq C_{id} \quad \forall i,j \in 1,..,n \forall d \in x,y,z \quad \forall i < j \quad (7)$$

$$X_i + X_j - 1 \leq \sum_{d=1}^{3} (Y^+_{ijd} + Y^-_{ijd}) \quad \forall i,j \in 1,..,n \forall d \in x,y,z \quad \forall i < j \quad (8)$$

$$Y^+_{ijd} \leq X_j \quad \forall i,j \in 1,..,n \forall d \in x,y,z \quad \forall i < j \quad (9)$$

$$Y^-_{ijd} \leq X_j \quad \forall i,j \in 1,..,n \forall d \in x,y,z \quad \forall i < j \quad (10)$$

$$\sum_{i=1}^{n} w_i X_i \leq W_{lim} \quad \forall i,j \in 1,..,n \forall d \in x,y,z \quad \forall i < j \quad (11)$$

$$up_{ij} \leq Y^+_{ijz} + (1 - Y^-_{ijz}) + (1 - Y^-_{ijy}) + (1 - Y^-_{ijx}) + (1 - Y^+_{ijx}) + (1 - Y^+_{ijy}) - 5 \quad \forall i,j \in 1,..,n \quad (12)$$

$$um_{ij} \leq Y^-_{ijz} + (1 - Y^+_{ijz}) + (1 - Y^-_{ijy}) + (1 - Y^-_{ijx}) + (1 - Y^+_{ijx}) + (1 - Y^+_{ijy}) - 5 \quad \forall i,j \in 1,..,n \quad (13)$$

$$up_{ij} \leq Y^+_{ijz} \quad \forall i,j \in 1,..,n \quad (14)$$

$$up_{ij} \leq 1 - Y^-_{ijz} \quad \forall i,j \in 1,..,n \quad (15)$$

$$up_{ij} \leq 1 - Y^-_{ijy} \quad \forall i,j \in 1,..,n \quad (16)$$

$$up_{ij} \leq 1 - Y^-_{ijx} \quad \forall i,j \in 1,..,n \quad (17)$$

$$up_{ij} \leq 1 - Y^+_{ijx} \quad \forall i,j \in 1,..,n \quad (18)$$

$$up_{ij} \leq 1 - Y^+_{ijy} \quad \forall i,j \in 1,..,n \quad (19)$$

$$um_{ij} \leq Y^-_{ijz} \quad \forall i,j \in 1,..,n \quad (20)$$

$$um_{ij} \leq 1 - Y^+_{ijz} \quad \forall i,j \in 1,..,n \quad (21)$$

$$um_{ij} \leq 1 - Y^-_{ijy} \quad \forall i,j \in 1,..,n \quad (22)$$

$$um_{ij} \leq 1 - Y^-_{ijx} \quad \forall i,j \in 1,..,n \quad (23)$$

$$um_{ij} \leq 1 - Y^+_{ijx} \quad \forall i,j \in 1,..,n \quad (24)$$

$$um_{ij} \leq 1 - Y^+_{ijy} \quad \forall i,j \in 1,..,n \quad (25)$$

$$C_{ix} + 20 \geq C_{jx} - S_x(1 - U_{ij}) \quad \forall i,j \in 1,..,n \forall i < j \quad (26)$$

$$C_{ix} + \sum_{r=1}^{2} R_{ir} S_{irx} - 20 \geq C_{jx} + \sum_{r=1}^{2} R_{jr} S_{jrx} - S_x(1 - U_{ij}) \quad \forall i,j \in 1,..,n \forall i < j \quad (27)$$

$$R_{i1} - R_{j1} \leq 1 - u_{ij} \quad \forall i,j \in 1,..,n \forall i < j \quad (28)$$

$$R_{j1} - R_{i1} \leq 1 - u_{ij} \quad \forall i,j \in 1,..,n \forall i < j \quad (29)$$

$$R_{i2} - R_{j2} \leq 1 - u_{ij} \quad \forall i,j \in 1,..,n \forall i < j \quad (30)$$

$$R_{j2} - R_{i2} \leq 1 - u_{ij} \quad \forall i,j \in 1,..,n \forall i < j \quad (31)$$

$$C_{iy} \geq 3.5 - S_y A_i \quad \forall i,j \in 1,..,n \quad (32)$$

$$B_i \leq A_i \quad \forall i \in 1,..,n \quad (33)$$

$$B_i \leq X_i \quad \forall i \in 1,..,n \quad (34)$$

$$B_i \leq A_i + X_i - 1 \quad \forall i \in 1,..,n \quad (35)$$

$$\sum_{i=1}^{n} B_i w_i \leq 4500 \quad \forall i \in 1,..,n \quad (36)$$

$$\begin{array}{ll} X_i, A_i, B_i, up_i, um_i \in \{0,1\} & \forall i \in 1,..,n \\ R_{ir} \in \{0,1\} & \forall i \in 1,..,n \forall r \in 1,2 \\ Y^+_{ijd}, Y^-_{ijd} \in \{0,1\} & \forall i,j \in 1,..,n \forall d \in x,y,z \\ U_{ij} \in \{0,1\} & \forall i,j \in 1,..,n \end{array}$$

İhale Sürecindeki Projeler İçin Mühendislik Süresi Tahminleme

10

FNSS Savunma Sistemleri

Proje Ekibi
İbrahim Onurcan Aksoy, Ahmet Arman Atalay, Damla Karagözoğlu, Muhittin Can Korkut, Erman Kurter, Gamze Yıldırım, İrem Yılmaz

Şirket Danışmanları
Gülçin Özden ve Başak Demirkol
Mühendislik Planlama ve Koordinasyon Departmanı

Akademik Danışman
Prof. Dr. Savaş Dayanık
Endüstri Mühendisliği Bölümü

Özet

FNSS, projelerin mühendislik sürelerini sürdürülebilir olmayan, yetkililerin fikirlerine ve tecrübelerine bağlı bir sistemle hesaplamaktadır. Projemizin amacı daha sürdürülebilir bir tahminleme sistemi sağlayan kullanıcı dostu bir arayüz oluşturmaktır. Bu doğrultuda, öncelikle şirketin projelerle ilgili verilerine kolaylıkla erişebileceği bir Excel dosyası oluşturulmuştur. Ayrıca geçmiş projeleri inceleyen ve istatistiksel olarak proje süresi tahminlemesi yapan bir parametrik model R yazılımı ile kurulmuştur. Yüzde 50 seviyesinin üzerinde olduğu kabul edilen şirketin mevcut tahminleme metodunun sapma miktarı, modelimizle birlikte iyileştirilmiştir. Bununla birlikte, tahminleme metodu veri analizine dayalı, sistematik bir hale dönüştürülmüştür.

Anahtar Sözcükler: Iş gücü tahminlemesi, regresyon, karar ağacı

Engineering Hours Estimation for the Projects in Bid Processes

Abstract

FNSS calculates the engineering times of the projects with an unsustainable system and depends on the opinions and experiences of the experts. The aim of our project is to create a decision support system that provides a sustainable and objective forecasting system. In this direction, an Excel file was created, where the company can easily access the data about the projects. Besides, a statistical model that examines past projects and estimates project durations for new projects has been built with R software. The deviation amount of the company's current estimation method, which is considered to be above the 50 percent level, has been improved with our model. Moreover, the estimation method has been transformed into a systematic one based on data analysis.

Keywords: workforce estimation, regression, decision tree

10.1 FNSS Defence Systems

FNSS Defense Systems was a land defence systems company established in 1988, focused on engineering combat vehicles and turrets. It designs and manufactures tracked and wheeled armored combat vehicles and has shipped hundreds worldwide, serving users in numerous nations. Jointly owned by Nurol Holding and BAE Systems, FNSS is Turkey's largest private defense industry company. It has a global presence with offices, facilities, or joint projects in UAE, Oman, Saudi Arabia, Malaysia, Indonesia, and the Philippines (Özsüt, 2022).

The annual reports of Nurol Holding are examined and the financial statements of FNSS for 2021 are taken into account. According to the tables in FNSS's financial statement, short-term trade receivables are 353,848 thousand Turkish Liras, short-term trade payables are 985,502 thousand Turkish Liras, cash is 566,826 thousand Turkish Liras, stocks are 286,831 thousand Turkish Liras, short-term prepaid expenses are 287.381 thousand Turkish Liras and long-term advances are 809,573 thousand Turkish Liras (Özsüt, 2022).

Within the scope of the Weapon Carrier Vehicles (STA) Project signed with the Undersecretariat of Defense Industries on 27 June 2016 for the armored anti-tank vehicle needs of the Turkish Armed Forces, a total of 260 units had produced until 2021, 184 units of KAPLAN STA and 76 units of PARS 4X4 STA. Moreover, since the new agreements with the Presidency of Defence Industries are in process, the company is not only one of the best in its sector, but also has been growing each day (Özsüt, 2022).

Throughout the project, we worked with the Engineering R&D Planning and Coordination unit consisting of three engineers and one manager. Their main responsibilities are not only limited to planning and coordination units such as task management, workload planning or process studies, but only includes the knowledge management and follow up of the old projects in partnership with technology and innovation processes function.

10.2 Engineering Hours Estimation

We will describe the current system of FNSS engineering hour estimation and performance measure of the project.

10.2.1 Current System Analysis

Recently, in FNSS, each engineer records his/her working time for each part of the project s/he works on to the company's ERP system. For example, an engineer may have worked four hours on PARS and four hours on KAPLAN projects on a day. At the end of the day, this engineer processes the data on the ERP system. S/he does it as follows: Firstly, s/he chooses the PARS project and then chooses its sub-modules. For example, s/he may have worked on the body design or s/he may have made an optimization study to make the track systems lighter, after selecting them, s/he processes the number of hours s/he spent for that part. For this reason, every spending of the engineering hour in the subsection of each operation (design, improvement, logistics planning, etc.) and sub-modules of each subsection (shell or palet design etc.) to be processed in the RD department and the system is recorded in the ERP system.

During the bidding, the department managers of subdivisions make their own estimates of the number of hours needed for their directorates, based on the number of hours spent on previous projects stored in the ERP. These predictions are made as follows. Details of projects are stored in different Excel files. Some of these include estimates of that project before it started. One of them contains data including the changes made during the process of the project, while another contains the final template of the completed project.

They have working hours, departments, and past projects' data that is thought to be required to decide estimated project hours that are collected in different Excel files. According to these data, each department's manager makes predictions for new projects subtasks' hours. After the estimates from the manager of each department are collected, the head of the proposal committee pre- pares the final bidding proposal according to an average estimate spending hours taking into consideration some of his/her own criteria.

10.2.2 Problem Definition

In light of the information above, we can easily say that the general problem is that while estimating the engineering workforce, an estimation is made based on the experiences of the people working in the institution over the years. Given that forecasting cannot be made when these people are retired, and that a human forecast cannot work better than a model, a statistical forecasting model for engineering's workforce hours will give more accurate and systematic engineering time estimates.

10.2.3 Performance Measures of the Solution

According to the problem definition, the solution should decrease the human effort for estimation process and give closer estimated hours to real project hours. To apply that, we increased the count of project related inputs and designed a parametric model for generating estimations based on statistical approaches.

10.3 Solution Strategy

10.3.1 Critical Assumptions

The project scope is limited to engineering hours needed for the conceptual and detailed design processes. Customers' requests may change. If there is such a case, In line with these requests, the duration is re-estimated for the remaining part of that customer's project. In addition, new technological developments have a direct impact on the workload of the project, so estimations are based on the technological accumulation in today's conditions. Factors affecting the complexity of achieving engineering hours should not be increased too much so that FNSS can give a competitive budget for the projects. We focused on the projects that the company has spent at least 30,000 actual engineering hours on and completed at least 97%.

10.3.2 Major Constraints

As a defense industry manufacturer, there may be some tangible/intangible parameters to maintain the desired production capacity, but it cannot contain imaginary parameters, and also there may be lower and upper limits that can be called challenges. There are some internal and external challenges. Internal challenges can be exemplified as the size of the workforce, workforce diversity, number of working hours, engineers' experiences, time spent on previous projects specifically, employee profiles (past projects, previous jobs, education), and capital resources. According to given parameters related to the complexity of work and worker qualification, these can be enumerated (e.g. rating the complexity of work or a worker's qualification on a scale) via analysis of the effects of these parameters in FNSS's past project durations. External challenges could be national and international defense industry project standards and lead times of the imported parts (Bashir and Thomson, 2001).

10.3.3 Objectives

The aim of the project is to create a decision support system that gives an estimate of the duration of the work and the need for engineering effort in potential projects based on the effects of the data of previous projects on the project durations. Furthermore, the goal of every company is to increase its profitability. Reducing the amount of deviation for the projects; in other words, making more accurate estimations will also increase the total profit that the company will get from the projects.

10.3.4 Solution Approach

Conceptual Model

Our decision support system collects the input data from the related departments, and puts the data into the parametric model that we created by regres- sion analysis of parameters and historical data. At first, the model finds hours of subsystems of a project. Then it sums these hours to reach the total project hour. The important features of our systems as follows:

- While calculating the experience coefficient, the model generates an output created by weighted averages based on the time spent in FNSS by the engineers, each of whom worked on a specific task,

- Total subsystems of the relevant project,

- Project similarity (a related department rated each task from 1 to 5) (1 = No engineering time will be wasted as the subsystem design matches the previous projects exactly, 5 = subsystem being designed for the first time),

- Project complexity (if there is a subsystem detail that has an impact on the time, concerning the department rated it 1 otherwise 2) (1 = The subsystem in the project contains an unusual and time-consuming detail, 2= There are no unusual details required for the subsystem),

- If the related projects are from domestic or foreign customers (where 1 is for domestic projects and 2 is for foreign projects),

- The variant order of the project. So, q symbolizes the variant order. Since, the additional requirements to generate the subtask the first time, it has a time difference with other units (The number q indicates what number of variants of the project),

- Whether the project is a modernization project or not,

- Whether the project is a “turret product” or not.

The flowchart of the estimation system model can be seen as;

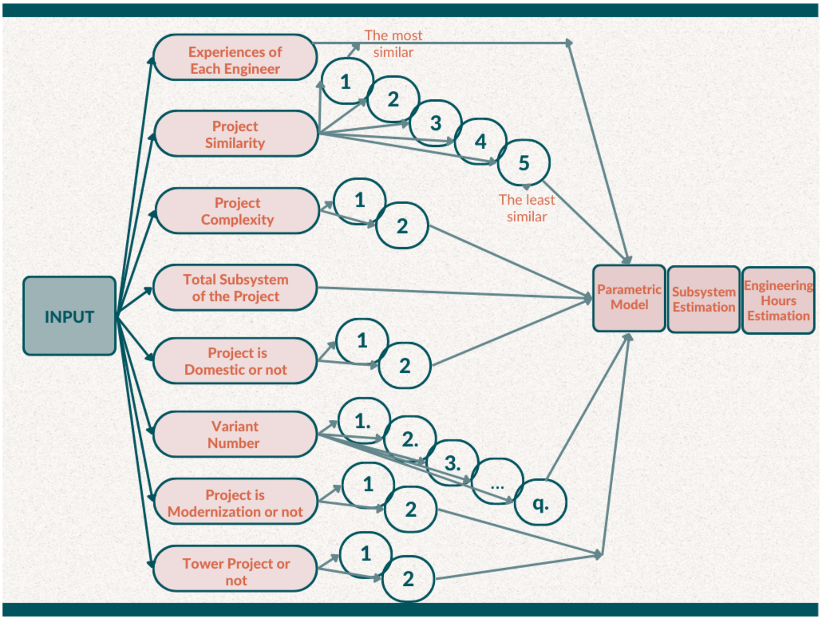


Figure 10.1: Flowchart of the Estimation System

Mathematical Model

As it is mentioned before in order to obtain the estimations of the engineering hours for the projects in the design phases it is offered to use a parametric model. The method is justified by several research papers concerning similar estimation problems. The formula that we use for estimating the time spent on each subsystem of the projects is (Salam et al., 2007);

$$\widehat{Y} = a\, x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n},$$

where

- $\hat{Y}$ is estimated hour for the related subsystem,

- X's are the features that affect product complexity(i.e. similarity, total subsystems, etc.),

- α's are the powers (coefficients) that show the overall effect of the relevant feature which is automatically generated according to historical data via R software,

- a is a general coefficient specific to the formula.

This parametric model is converted into a multivariable linear by taking logarithms of each side as in

$$\log\left(\widehat{Y}\right) = a + \alpha_1 \log(x_1) + \ldots + \alpha_n \log(x_n),$$

and this is solved with linear regression methods (Salam et al., 2007).

Solution Method

The R language is used for fitting and obtaining the a and α values for equation of each subsystem. Also, R is used to predict the hours' values for the new data. In order to validate fitted values, statistical measures are used to evaluate the significance of a and α values. There are also methods to determine the outliers in the data that could be removed so that it would yield a better fit. In addition to features that mentioned as x's in mathematical formula, squares of these x's and multiplications of these features are investigated as a new feature to be added. Trying to find good fitting features among all features mentioned before(x's and features that are derived from x's), it is not possible to try all combinations of these features since the number of subsets of the features is 2^N, where N is the number of features. So a heuristic such as forward addition or backward elimination is used in feature selection by using the F statistic and AIC(Akaike Information Criterion) in each step. If a subtask has enough data, the data could be divided into training data and test data. The model is fitted on training data and tested on the test data to calculate different measures to calculate the accuracy of the model.

10.3.5 Verification

A random data is generated from the model and a random noise added to it to simulate a real fitting situation since the data is still to be collected on some of the features. For our randomly generated data for a subtask is $(a, \alpha_1, \alpha_2, ..., \alpha_6) = (100, 0.5, 0.4, 0.3, 0.2, 0.1, 0)$ and $\log \varepsilon$ has normal(0, 0.1)) distribution. As it could be seen, the 6th feature's coefficient is 0. So In the analysis, we should be able to get rid of it with our statistical methods.

A random dataset of size 50 is generated with the attributes above, and fitted in R Language. As a result, estimated parameters got close to real values. For the intercept we took exponential of estimation and observed that it is very close to real value. Drop1 function is used to execute one iteration of backward elimination. The model is generated according to the model Hours = Experience + Similarity + Total Subsystem + Complexity + Domestic or Foreign + Variant Order.

For that specific example, Variant Order input doesn't feed the model well because both F statistics and AIC tells us that the difference between the original model and the model without the "Variant Order" feature is not significant. Therefore, it could be dropped. This is expected since it do not have any correlation with Hours value.

After dropping the mentioned feature, a new iteration is tried and according to F statistics, all features are significant and any of them should not be dropped. The last model becomes the optimal model.

10.3.6 Validation

We curreny have eight features for each subsystem where past projects are our data. These are the number of subtasks in the project, experience of the team in years, project similarity and complexity, customer information related to its country, variant order of the project, whether the project is modernization and whether the project is tower product. We worked on the subtask samples that have enough historical data to generate a statistical model. Also these subsystems are most common tasks in all projects of FNSS. To get the reliable results, we use the past projects that completed at least 90%. To analyze the results, we used Leave-One-Out-Cross-Validation methodology. As a result of this method, we got MAPE (mean absolute percentage error) values between 47% and 150%.

10.4 Integration and Implementation

This model will be added as an interface created via Shiny to the reporting system used by FNSS in their R&D projects depicted in Figure 10.2. Thanks to this interface, the time estimates of the projects that will be started or planned to be started can be calculated. Time estimation can be made by entering certain attributes in the user-friendly interface containing the model we created. In this way, when RD employees start a new project, they will be able to make a time estimation automatically. Moreover, the use of interface will add the project generated by it to the previous data and upgrade the model with every new project entries. Therefore, even if the project has unexpected deviation results for some subtasks, it will be more truthful and usable in the long-run. "Results of the fitted models" page shows the detailed historical data and analysis of the selected subtask to help to user for estimating engineering hours.

Project Manager

12-05-2023

Data to Make the Prediction

Project Name	Subsystem Name	Task Count	Experience	Similarity	Unexpected Complexity	Variant	Country
Yeni1	ARMOUR	50.00	5.00	3.00	2.00	1.00	3.00
Yeni1	CONDESPHAS	50.00	6.00	2.00	1.00	2.00	1.00
Yeni1	EE CNTRL	50.00	7.00	2.00	1.00	3.00	2.00
Yeni1	HULL	50.00	8.00	3.00	2.00	4.00	3.00
Yeni1	HYDRAULIC	50.00	9.00	5.00	2.00	4.00	1.00
Yeni1	SYSDETAILD_DETAIL DESIGN	50.00	10.00	2.00	1.00	2.00	2.00
Yeni2	ARMOUR	50.00	6.00	3.00	2.00	1.00	3.00
Yeni4	HULL	40.00	12.00	3.00	1.00	2.00	1.00

Prediction and the Model | LOOCV Results of the Fitted Models

Subsystems:

ARMOUR

ProjectName:

Yeni1

Prediction : 2043.46 Hours

Mean Crossvalidation Percentage 68.02 %

first page

Prediction and the Model | LOOCV Results of the Fitted Models

Show 25 entries

Search:

Project	Task_Count	Variant	Country	Experience	Hours	Predicted	LOOCV Percentage deviation
OKK_6X6KOM	36	2	1	11.60	300.50	280.0192	0.06815571
PHILSABER6	43	1	2	6.97	1207.50	899.6733	0.25492891
MED_TANK	66	1	2	8.16	1294.75	923.2260	0.28694649
OMTTZA_KOM	81	2	1	9.82	1085.00	735.1463	0.32244577

second page

Figure 10.2: User Interface

10.5 Benefits to the Company

As a result of our observations, our hypothetical estimation model, which we realized on a completed project, whose data we did not include in our analysis, with the leave one out cross-validation methodology, reduced the margin of error to 50% on average and provided a significant amount of improvement. In addition, the interface is integrated into the company during the estimation periods of the projects, now the unit managers will be able to make the time estimation they need to prepare for their own units in a shorter time since they have concrete data in their hands, and more data-based with little margin of error and for a short time.

10.6 Conclusion

In conlusion, FNSS Defence Systems needed more automated and usable engineering hour estimator to upgrade their bidding process. By recording the historical project data, collecting related inputs for them and applying statistical approaches to them, the company would get more accurate estimations more quickly. Moreover, for every new project and newly discovered inputs, the system will upgrade itself. In the end, the company would decrease the estimation deviation and make their bidding system more sustainable by our product.

Bibliography

Bashir, H. and V. Thomson (2001, 01). Estimating effort and time for design projects. Unpublished manuscript. Accessed https://www.researchgate.net/publication/228902168_Estimating_effort_and_time_for_design_projects on 17 May 2023.

Salam, A., N. F. Bhuiyan, G. J. Gouw, and S. A. Raza (2007). Estimating design effort in product development: A case study at Pratt & Whitney Canada. In *2007 IEEE International Conference on Industrial Engineering and Engineering Management*, pp. 989–993.

Özsüt, O. T. (2022). Nurol Holding Yönetim Kurulunun Yıllık Faaliyet Raporuna İlişkin Bağımsız Denetçi Raporu.

Kaotik Depolu Sipariş Karşılama Merkezlerinde Ürün Toplama Sürelerinin Enazlanması OPLOG

11

Proje Ekibi

Ege Başaran, Gizem Bumin, Berk Ender, Sung Min Kim, Ahmet Burak Okur, Berfin Ulus, Arda Uzunoğlu

Şirket Danışmanı
Yaren Yılmaz
Araştırma ve Geliştirme Mühendisi

Akademik Danışman
Doç. Dr. Özlem Karsu
Endüstri Mühendisliği Bölümü

Özet

Bu projede OPLOG'un depolama sistemi analiz edilmiştir. OPLOG, mevcut depolama süreçleri için kaotik bir depolama sistemi kullanır. Ürün ve depo nitelik bilgilerini içeren veriler analiz edilerek, toplama sürelerinin en aza indirilmesi amaçlanarak, yerleştirme yönergeleri oluşturulur. Bu raporun sonucu, toplama sürecinin performansını arttıran ürün yerleştirme stratejileri için en uygun rehber olacaktır. Kaotik bir sistemin dışında, ürün ilişkileri analiz edilerek ürün yerleştirme yöntemi yapılacaktır. Ürünlerin yerleştirilmesi birlikteliklere göre düzenlenecek, böylece toplama süreleri azaltılabilecektir.

Anahtar Sözcükler: Kaotik Depo, Yerleştirme, Toplama, Depo yönetim Sistemi (WMS), Veri Analizi, Apriori Algoritması, Birliktelik Kuralları

Minimizing Item Picking Times In Fulfillment Centers With Chaotic Storage

Abstract

In this project, the storage system of OPLOG is analyzed, and solution methods are proposed with an industrial engineering perspective to improve the picking process. OPLOG uses a chaotic storage system for its current warehousing processes. Analyzing the data containing item and bin attribute information, Put-away guidelines are formed to minimize pick-up times. The outcome of this report is an optimal guide for product placement strategies that enhance the performance of the picking process. Apart from a chaotic system, the product placement method is done by analyzing product associations. Placement of the products is organized according to associations so that the pickup times can be decreased.

Keywords: Chaotic Warehouse, Put-away, Picking, Warehouse Management System(WMS), Data Analysis, Apriori Algorithm, Association Rules

11.1 System Analysis and Problem

OPLOG is a fourth-party logistics (4PL) service provider founded in 2013. It offers logistics services such as warehousing, order fulfillment, and delivery process management to the world's largest corporations, with over 300 employees. OPLOG has two fulfillment centers hosting over 3.5 million products and uses chaotic storage. The fulfillment centers operate 24/7, and the average delivery duration is 19 hours, with same-day delivery available for Istanbul. OPLOG's partner companies send their products to fulfillment centers, and the rest of the process is handled by OPLOG.

OPLOG uses a special warehouse management system (WMS) and offers end-to-end visibility and tracking to eliminate operational challenges and unpredictable costs. It also provides value-added services such as customized packaging and preferred shipping techniques requested by customer companies. OPLOG deals with all operational activities on a single online platform integrated with the most popular sales platforms used by partners for e-commerce. One of OPLOG's important services is optimal shipping, where it works with multiple shipping companies to decide the best courier for a package based on the package size using smart algorithms simultaneously. OPLOG also handles quality control and returned products, where the products undergo quality control processes to be added to stocks, and returns are reported to the product owner companies. OPLOG provides reports to its partner customers about their product sales rates and offers solutions for unsold products.

In the Dilovası warehouse (B2B), our main aim is to decrease picking time by determining associated products and arranging their placements. In other words, the distance the picker travels between different products in the same order list needs to be decreased by clustering the products that are selling together. Note that in this warehouse, order lists contain only one client's products since the orders came from business partners, not from individual customers. Since it is the case, warehouses already store the same branded products close to each other. Also, in some cases, clients notify OPLOG that they usually order product pairs. If clients warn them, OPLOG tries to place those product pairs that are frequently ordered together close to each other to minimize the distance between those products. Unfortunately, this is not the case in general. OPLOG wants to implement these product associations to decrease pickup times by placing associated products close to each other.

11.2 Solution Approach and Model

OPLOG provided us with sales order data containing the last month's orders and products included in those orders. For the implementation of Association Rules, three algorithms which are AIS, STEM, and Apriori, are found for a candidate solution approach. Although it is observed that they can detect strong correlations in an itemset, considering the fact that AIS and STEM can generate and count many small candidate itemsets, the Apriori algorithm seemed a more applicable method for our problem since the Dilovası warehouse contains a large number of itemsets. Apriori Algorithm/Association Rule Mining was a very important preliminary step for the product placement guidance process, it provided results about which products should be stored close to each other. With the availability of antecedent and consequent relationships, product placement guidance is provided via the Python program so that order-picking time can be minimized (Edureka, 2020). For the Apriori Algorithm, the Association is measured using three types of matrices.

We have Support, Confidence, and a Lift. So, support is defined as the frequency of item A or the combination of items A and B. Support is the frequency of the products are ordered, and the combination of the frequency of the items are ordered. With the Support, the less frequent items can be filtered out. The Confidence, Lift, and Leverage calculation can be seen in the equations below (Prawira et al., 2020). Confidence indicates how often items A and B occur together, given the number of times A occurs in the data, which is also expressed as A $\rightarrow$ B. Lift is mainly the strength of any rule. A lift value can be between 0 and ∞, and greater than 1 means that item B is likely to be chosen if item A is chosen, while a value less than 1 means that item B is unlikely to be chosen if item A is chosen.

The initial step of the problem is acquiring the required sales-order data, which includes Operation ID, Orderlist ID, and product IDs. It is important to identify order lists that contain particular products in the dataset in order not the Apriori algorithm to detect the association of the products within different orders. After the preparation of the data into a binary order/product form, for each Operation ID, we used mlxtend for apriori and association rules to spot possible correlated and associated products in different orders. Since we investigated that the number of total orders for each Operation ID varies from 8 to 77458, the Python code had to be developed to try different min-support values in the apriori function to spot possible associated products and rules that are derived from them. Python code uses a for loop to try different values of min support starting from 0.5 and decreasing by 0.001. As the min support value decreases, the number of association rules gradually or rapidly increases, which is an important aspect for determining the accuracy of the rules. The loop stops when the difference between the number of association rules between two apriori results equals one-tenth of the "total number of distinct ProductIds" by using sequent min support as an input. Since the increase in the number of association rules with different min support values is not linear, this method is successful when acquiring the "most reasonable" number of rules that include the paired products at a balanced rate, also by choosing the most accurate min support value for the apriori function. For chosen Operation ID "2916A049-9B6C-4F3E-A029-29A1DB17B765", Apriori Algorithm results, and part of the found Association Results which include Antecedent Product ID, Consequent Product ID, Antecedent Support, Consequent Support, Support,

```
OperationId: 2916A049-9B6C-4F3E-A029-29A1DB17B765    min_support chosen as: 0.014
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(2BC46BB3-5C63-4296-8CCA-1EA19300BCD2)	(CCF612E1-78F7-4687-82A2-85139B477A6A)	0.049232	0.147974	0.022680	0.460674	3.113210	0.015395	1.579798
1	(CCF612E1-78F7-4687-82A2-85139B477A6A)	(2BC46BB3-5C63-4296-8CCA-1EA19300BCD2)	0.147974	0.049232	0.022680	0.153271	3.113210	0.015395	1.122871
2	(403611D9-EA51-4E8A-8478-541A98C6C438)	(2FEDCAA4-BDCD-4A4A-AE2A-03178AB608F0)	0.024340	0.061955	0.016872	0.693182	11.188388	0.015364	3.057330
3	(2FEDCAA4-BDCD-4A4A-AE2A-03178AB608F0)	(403611D9-EA51-4E8A-8478-541A98C6C438)	0.061955	0.024340	0.016872	0.272321	11.188388	0.015364	1.340785
4	(5597DEE4-BF87-4BD7-953F-D9AB8566585B)	(689CE6DB-6A7C-4ABD-B790-006D04D15D68)	0.060987	0.125294	0.019361	0.317460	2.533726	0.011720	1.281546
5	(689CE6DB-6A7C-4ABD-B790-006D04D15D68)	(5597DEE4-BF87-4BD7-953F-D9AB8566585B)	0.125294	0.060987	0.019361	0.154525	2.533726	0.011720	1.110634
6	(CCF612E1-78F7-4687-82A2-85139B477A6A)	(5597DEE4-BF87-4BD7-953F-D9AB8566585B)	0.147974	0.060987	0.016595	0.112150	1.838896	0.007571	1.057625
7	(5597DEE4-BF87-4BD7-953F-D9AB8566585B)	(CCF612E1-78F7-4687-82A2-85139B477A6A)	0.060987	0.147974	0.016595	0.272109	1.838896	0.007571	1.170540
8	(8AD743C3-2F54-4792-95F9-BC1BCE83FD60)	(689CE6DB-6A7C-4ABD-B790-006D04D15D68)	0.078827	0.125294	0.018670	0.236842	1.890293	0.008793	1.146167
9	(689CE6DB-6A7C-4ABD-B790-006D04D15D68)	(8AD743C3-2F54-4792-95F9-BC1BCE83FD60)	0.125294	0.078827	0.018670	0.149007	1.890293	0.008793	1.082468

Figure 11.1: Association Results from Apriori Algorithm

Confidence, Lift, Leverage, and Conviction are shared in Figure 11.1.

After association rules with certain min support values were acquired from the apriori algorithm, Python code was further developed for the product placement guidance stage related to putting associated/clustered products together. Since the associations between products were already obtained from the previous part, for this part, we acquired the product-location data in order to run the algorithm for clustered product placement. The algorithm we developed in Python can work individually for any chosen Product ID, which is prepared to use when a product is at its put-away stage. Once an ID of a Product is entered into the system, previously analyzed association rules, products that are associated with the entered Product, and their current locations in the warehouse are provided by the program. Then for each associated product, a different weight value is calculated from the min support values obtained from their particular rules. In other words, each product is associated with a Product at a different level, and weight is used to represent this level aspect. After weights are calculated, they are used with associated products' locations that contain the highest stock in order to provide guidance and a range for putting the chosen product in such a way that it is "centered" among its associ-

```
Enter the product id(one of the above): CCF612E1-78F7-4687-82A2-85139B477A6A

Result Summary:
For the chosen ProductId CCF612E1-78F7-4687-82A2-85139B477A6A Associated Products with their association weight
are:

2BC46BB3-5C63-4296-8CCA-1EA19300BCD2    0.163
5597DEE4-BF87-4BD7-953F-D9AB8566585B    0.119
689CE6DB-6A7C-4ABD-B790-006D04D15D68    0.273
8648DFA4-41B5-4EB1-9862-84986C070019    0.122
F2D9AF7D-6714-4AA6-8A0A-C0414793B314    0.322

All of the current locations of those productId's are:

AY-Y1-K1-G054
AY-Y1-K1-G061
AY-Y1-K1-G063
AY-Y1-K1-G064
AY-Y1-K2-G042
AY-Y1-K3-G052
AY-Y1-K4-G008
AY-Y1-K4-G009
AY-Y1-K4-G017
AY-Y1-K4-G018
AY-Y1-K5-G003
AY-Y1-K5-G017

Locations with maximum stock of those ProductId's are:

AY-Y1-K1-G061     ProductId: 2BC46BB3-5C63-4296-8CCA-1EA19300BCD2
AY-Y1-K1-G063     ProductId: 5597DEE4-BF87-4BD7-953F-D9AB8566585B
AY-Y1-K1-G054     ProductId: 689CE6DB-6A7C-4ABD-B790-006D04D15D68
AY-Y1-K3-G052     ProductId: 8648DFA4-41B5-4EB1-9862-84986C070019
AY-Y1-K1-G064     ProductId: F2D9AF7D-6714-4AA6-8A0A-C0414793B314

Center of the locations with respect to weights has found as Bin 59 in corridor AY.

Product with ProductId CCF612E1-78F7-4687-82A2-85139B477A6A can be placed in corridor AY between bins 54-64.

Validation:

Average distance before the algorithm: 5.2(cells) Average distance after centering algorithm: 4.6(cells).
```

Figure 11.2: Python Output for Specific ProductID

ated products. The example for entered ProductId "CCF612E1-78F7-4687-82A2-85139B477A6A" output of the Python code that indicates the Results Summary is shared in Figure 11.2 so that the clustered product placement stage and the algorithm can be better understood.

11.3 Validation

Simulation for layout assignment via Excel VBA is used to observe and validate the relationship between the "number of unique products in order" and total picking distance cost with accurate placement of correlated items, utilizing the apriori algorithm and association rules from the Solution Approach. In order to correctly determine our approach for simulation, we had to be informed about the warehouse's current layout and routing system, which is provided by OPLOG during a meeting. The warehouse consists of horizontal and vertical aisles with vertical/horizontal movements only, and the S-Shape routing strategy is used during the order-picking process. Certain products are placed in specific areas designated for different aisles, with some brands having their own aisle. Our simulation focused on a single aisle, "AY," where products belonging to OperationID "2916A049-9B6C-4F3E-A029-29A1DB17B765" were distributed.

We started the simulation by choosing a specific product pair "7d50a2cf-98ca-4fd3-90b6-f36153849541"-"8286c85a-9f06-4532-9e50-86c93f3a3645" and obtained 75 sample orders that include that product pair from SalesOrder data. Different order lists with different content alongside the chosen product pair were simulated, and the total distance cost was calculated 300 times for each order list that includes a different number of unique products. The program randomly filled cells for the first case without layout adjustment, and for the second case, the program replaced one correlated product close to its pair to decrease the total travel/distance cost. It is concluded that if the number of unique products in an order list increases, the average improvement is expected to decrease since the significant change effect is also

Unique Product Range	Average Improvement	Min	Max
2	39.08%	6.90%	55.74%
3+	23.03%	23.03%	23.03%
5-10	7.45%	3.64%	14.16%
11-15	2.42%	1.80%	3.43%
16-20	1.07%	0.64%	1.85%
21-30	0.35%	0.11%	2.10%
31+	0.009%	0.02%	0.19%
General Result	*9.03%*	*0.02%*	*55.74%*

Table 11.1: Simulation Improvements

New Product ID	The Location Before Algorithm	Bins of Associated Products	Suggested Location by Algorithm	Average of Differences Before Algorithm	Average of Differences After Algorithm
2BC46BB3-5C63-4296-8CCA-1EA19300BCD2	AY-Y1-K1-G061	G064	G064	3	0
CCF612E1-78F7-4687-82A2-85139B477A6A	AY-Y1-K1-G064	G061; G063; G054; G052; G064	G059	5.2	4.6
2FEDCAA4-BDCD-4A4A-AE2A-03178AB608F0	AY-Y1-K1-G056	G030	G030	26	0
689CE6DB-6A7C-4ABD-B790-006D04D15D68	AY-Y1-K1-G054	G063; G043; G064; G064	G060	10	7
5597DEE4-BF87-4BD7-953F-D9AB8566585B	AY-Y1-K1-G063	G054; G064	G059	5	0
8AD743C3-2F54-4792-95F9-BC1BCE83FD60	AY-Y1-K3-G043	G054	G054	11	0
F2D9AF7D-6714-4AA6-8A0A-C0414793B314	AY-Y1-K1-G064	G054; G064	G060	5	5
7D50A2CF-98CA-4FD3-90B6-F36153849541	AY-Y1-K1-G048	G053	G053	5	0
8286C85A-9F06-4532-9E50-86C93F3A3645	AY-Y1-K1-G053	G048	G048	5	0
8648DFA4-41B5-4EB1-9862-84986C070019	AY-Y1-K3-G052	G064	G057	6.5	6.5
D9D4F543-8FC6-45ED-8F2B-89AB1C4C7C16	AY-Y1-K3-G051	G052	G052	1	0

Figure 11.3: Validation with Python Code Output

weakened; see Table 11.1.

Besides validation with simulation, we also wanted to validate our approach by using our Python code. For the same OperationID "2916A049-9B6C-4F3E- A029-29A1DB17B765", the algorithm/code generated 20 association rules, and for each unique ProductId that is included in association rules, we tested the "clustered product placement" approach. For each ProductId, we calculated the average distance between that particular product and its associated products. Then, by using the centering algorithm and running our code, we calculated the same value again. Results that indicate changes and improvements in terms of bin distances are shared in Figure 11.3.

11.4 Integration and Implementation

For the implementation of the project, we used Python code to suggest a suitable location range for the put-away process of products. This location range was based on the location information of pairs of products. We used past weighted data of their pairs to assign the location range to enter products. This way, the operator could place the product within the suggested location range provided by our model. This approach ensured efficient and effective product placement during the put-away process. To integrate our Python code into OPLOG's WMS system, we acted in accordance with the instructions given by the company officials during the implementation process. The project's implementation process will begin by forming an appropriate team size and deciding on the architectural structure of the software. Data processing pipelines will be developed for retrieving data from warehouse databases, and the cloud infrastructure will be set up. The product clustering service will be developed to calculate weekly clusters, and the system for the simultaneous clustering of products will be evaluated. A service will also be developed to determine recommended location ranges for products in clusters, and a backend will be created for external system queries. Finally, the mobile development team will be briefed on the necessary screen and process changes, and the UX/UI team will share their designs for user interactions with the mobile team.

11.5 Benefits to the company

The updated project scope of the company aims to detect and minimize the picking time for products that are not currently sold as associates in the existing system by utilizing appropriate placement and arrangement. The project's primary objective is to achieve faster picking, better customer service, improved employee productivity, balanced workload, and employee well-being.

To achieve the aforementioned objectives, the team has developed a simulation model utilizing single-aisle testing. The model was used to compare the existing system with five scenarios based on the Apriori output. The scenarios involved relocating paired and single products to enhance clustering and distance. The table below indicates that the algorithm's output resulted in improvements in product distances. The first column of the table shows the Product IDs and their associated product IDs, while their weights which are used to determine the center point of the associated products, are listed in the second column.

After computing the center point, the average distance metrics were recorded for the products before and after applying the algorithm. The results show that the distance improvements ranged up to 100%. For a product that only has a single associated product, 100% improvement is observed since the distance between two products is lowered to zero by the clustering algorithm. For observing an average improvement, an example of a case is provided in Figure 11.4. Including 12 unique products and their 22 associated products for randomly picked bins that are feasible, the general average is calculated as 72 %. This procedure is repeated for the same ProductId associations with differently chosen bins 200 times, and the overall improvement average is calculated as 72 %. This suggests that we expect a 72 % improvement in terms of distance cost as a result of using the clustered product placement algorithm.

ProductID (Red = Chosen, Black = Associated)	Bin Number(s)	Weights	Random Bin Chooser	Associated Location(s)	Center	Average Distance Before	Average Distance After	Difference	Improvement
CCF612E1-78F7-4687-82A2-85139B477A6A	64				43	16.6	14.865	2	10%
2BC46BB3-5C63-4296-8CCA-1EA19300BCD2	61-17	0.163	1	61					
F2D9AF7D-6714-4AA6-8A0A-C0414793B314	64	0.119	1	64					
8648DFA4-41B5-4EB1-9862-84986C070019	52-09	0.273	1	52					
689CE6DB-6A7C-4ABD-B790-006D04D15D68	54-42-03	0.122	2	42					
5597DEE4-BF87-4BD7-953F-D9AB8566585B	63-08-18	0.322	3	18					
2FEDCAA4-BDCD-4A4A-AE2A-03178AB608F0	56				1	55	0	55	100%
403611D9-EA51-4E8A-8478-541A98C6C438	01-30	1	1	1					
689CE6DB-6A7C-4ABD-B790-006D04D15D68	54				24	32.5	17.691	15	46%
5597DEE4-BF87-4BD7-953F-D9AB8566585B	63-08-18	0.193	3	18					
8AD743C3-2F54-4792-95F9-BC1BCE83FD60	17-43-20	0.186	3	20					
CCF612E1-78F7-4687-82A2-85139B477A6A	64-19-04-17-16	0.377	3	4					
F2D9AF7D-6714-4AA6-8A0A-C0414793B314	64	0.245	1	64					
5597DEE4-BF87-4BD7-953F-D9AB8566585B	63				38	26.5	17.5	9	34%
689CE6DB-6A7C-4ABD-B790-006D04D15D68	54-42-03	0.538	1	54					
CCF612E1-78F7-4687-82A2-85139B477A6A	64-19-04-17-16	0.462	2	19					
8AD743C3-2F54-4792-95F9-BC1BCE83FD60	43				3	40	0	40	100%
689CE6DB-6A7C-4ABD-B790-006D04D15D68	54-42-03	1	3	3					
F2D9AF7D-6714-4AA6-8A0A-C0414793B314	64				27	33.5	11.5	22	66%
689CE6DB-6A7C-4ABD-B790-006D04D15D68	54-42-03	0.355	2	42					
CCF612E1-78F7-4687-82A2-85139B477A6A	64-19-04-17-16	0.645	2	19					
8286C85A-9F06-4532-9E50-86C93F3A3645	53				16	37	0	37	100%
7D50A2CF-98CA-4FD3-90B6-F36153849541	48-16	1	2	16					
7D50A2CF-98CA-4FD3-90B6-F36153849541	48				18	30	0	30	100%
8286C85A-9F06-4532-9E50-86C93F3A3645	04-53-18	1	3	18					
8648DFA4-41B5-4EB1-9862-84986C070019	52				28	24.5	23.5	1	4%
CCF612E1-78F7-4687-82A2-85139B477A6A	64-19-04-17-16	0.49	3	4					
D9D4F543-8FC6-45ED-8F2B-89AB1C4C7C16	27-51	0.51	2	51					
D9D4F543-8FC6-45ED-8F2B-89AB1C4C7C16	51				52	1	0	1	100%
8648DFA4-41B5-4EB1-9862-84986C070019	52-09	1	1	52					
403611D9-EA51-4E8A-8478-541A98C6C438	30				56	26	0	26	100%
2FEDCAA4-BDCD-4A4A-AE2A-03178AB608F0	56	1	1	56					
2BC46BB3-5C63-4296-8CCA-1EA19300BCD2	61				64	3	0	3	100%
CCF612E1-78F7-4687-82A2-85139B477A6A	64-19-04-17-16	1	1	64					
GENERAL AVERAGE									72%

Figure 11.4: Improvement Table

Bibliography

Edureka, M. (2020). Apriori algorithm explained association rule mining finding frequent itemset. https://www.youtube.com/watch?v=guVvtZ7ZClw. Accessed on 16 May 2023.

Prawira, T. Y., S. Sunardi, and A. Fadlil (2020). Market basket analysis to identify stock handling patterns & item arrangement patterns using apriori algorithms. *Khazanah Informatika: Jurnal Ilmu Komputer dan Informatika 6*(1), 33–41.

12 | Lojistik Süreçlerinde Depo Düzenine Dayalı İşgücü Optimizasyonu

Eti Gıda

Proje Ekibi
Ayşe Beyza Çanakçı, Birgi Anıl Hacıoğlu, Doruk Karakaş,
Zeynep Betül Malkoç, Esin Julia Rathert, Zeynep Şahin, Efecan Şentürk

Şirket Danışmanı
Zafer Ünlüer
Süreç Geliştirme Proje Yöneticisi

Akademik Danışman
Dr. Öğr. Üyesi Firdevs Ulus
Endüstri Mühendisliği Bölümü

Özet

Önerilen proje, ETI Gida'nın Ankara İvedik Deposu'ndaki işgücünün verimliliğinin artırılması için depo düzenine yönelik iyileştirme önerileri sunmaktadır. Projede, işçilerin ve forklift operatörlerinin verimsiz hareket etmesine sebep olan neden, ürünler ve raf yerleşimleri arasında bir ilişkinin olmaması olarak tespit edilmiştir. Bu iki kısımlı problem üzerine iki ardışık modelden oluşan bir sistem önerisi yapıldı. Önerilen çözümün ilk kısmı için Python paketlerini kullanarak Apriori algoritmasıyla sık seçilen ürünleri tanımlamak için bir model geliştirildi. İlk modelin çıktısı, yeni ürünlerin raf hücrelerine yerleştirilmesine odaklanan ikinci modelde girdi olarak kullanıldı. Sunulan modellerin güvenilirliğini ve etkilerini tespit etmek için önerilen çözüm farklı depo düzeni verileriyle test edildi. Son olarak depo çalışanları için kullanıcı dostu bir arayüz geliştirildi.

Anahtar Sözcükler: depo, işgücü, atama, mesafe, zaman, verimlilik, kullanım

Workforce Optimization Based on Warehouse Layout in Logistics Processes

Abstract

The proposed project offers solutions for the problems faced by ETI Gida in its Ankara warehouse regarding layout and workforce optimization. The main issue was the excessive movements of workers and forklift operators due to a lack of relationship between products and rack shelves. A model utilizing the Apriori algorithm, and Python packages was developed to identify frequently picked products, which was used as input for the second model focusing on allocating new products to rack cells using Python. The offered models were tested with different warehouse layout settings to ensure their effectiveness, and a user-friendly interface was developed for warehouse employees.

Keywords: warehouse, workforce, allocation, distance, time, efficiency, utilization

12.1 Company Description

ETİ Gıda A.Ş. founded by Firuz Kanatlı commenced its operations in 1961 in Eskişehir. ETİ changed its name to "ETİ Gıda Sanayi ve Ticaret A.Ş." in 1972. ETİ Gıda, which produces biscuits, cookies, cakes, pies, chocolate, wafers, baby food, and ready-made food products, has seven different factories, mostly located in Eskişehir. These factories are ETİ Biscuit Factory, Cake Factory, Craker Factory, Chocolate Factory, Bozüyük Factory, Integrated Grain Processing Factory, and Dairy Products Factory. Having 45 brands and nearly 300 products, ETİ exports to approximately 40 countries on five continents.

12.2 System and Problem Analysis

12.2.1 Current Situation and Processes

ETİ İvedik Warehouse has three cold storage areas which are called Sütburger, chocolate, and ambient areas with different temperature conditions. The forklifts inside the warehouse unload the products coming from Eskişehir Logistics Center by truck and put them in the storage cells on the upper floors or unload the products in those cells to the shopping areas on the ground floor. All cells in the warehouse have their own unique barcode system, which enables communication between the shopping areas and rack cells. Every morning more than 40 route vehicles are loaded with the products to be distributed. Therefore, work orders are prepared beforehand on the previous day. The demanded products are collected from the shopping areas, and these areas are fed from the (storage) rack cells.

12.2.2 Analysis of Current System

Currently, there are different methods applied to increase workforce utilization of the warehouse. As mentioned earlier, a coding system similar to Kanban is already applied, and the products' shopping area locations are planned based on the sales rate of the goods while considering the necessary requirements. Workers collect products from shopping areas to prepare for the demands of different customers, and shopping areas in ETİ İvedik Warehouse are replenished with an (S,s) policy. Once a shopping area was emptied, the forklift to carry the necessary pallet was observed to be traveling to different corridors. The relationship between the pallet placement to upper rack cells and the same product's shopping area relation is decided by the warehouse workers. Forklift operators empty the truck of new arriving products by placing random pallets in rack cells that are closest to the truck unloading area. This means that the pallet placement for rack cells does not consider the shopping area (SA) places. In this project, the placement of these products takes SA locations into account as well.

12.3 Proposed Solution Strategy

We designed 1) storage location assignment in shopping area cells and 2) new arriving product pallet storage location assignment in rack cells. The first model assigns each SKU to the shopping area cells and the second model uses this information as its input. Placement in shopping areas will affect the storage location assignment in rack cells. The flow chart in Figure 12.1 represents the stages and their relations.

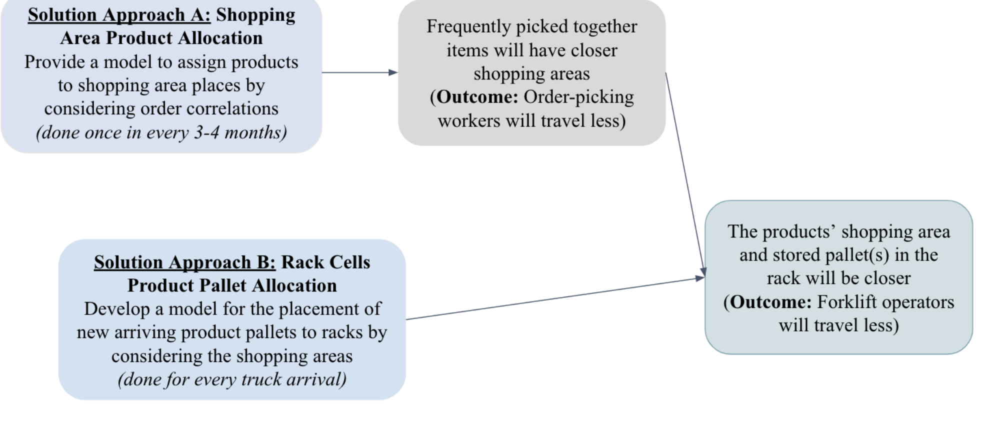


Figure 12.1: Flow Chart of The Planned Progress

In our project, the Apriori Algorithm was utilized via Python. SKUs that are ordered together along with the rules between them are found using the package's association rules function. This function results in conditional probability values known as confidence. It should also be noted that we have only considered the pairwise relationship between SKUs at this stage of our progress. However, there are several studies focusing not only on pairs of SKUs to establish item families (Li et al., 2021).

12.3.1 Shopping Area Allocation Grouping Heuristic

The developed heuristic algorithm finds the groups of SKUs and assigns them to the groups of shopping areas that mostly include 6 shopping areas. This approach significantly reduces the model's computational load and run time.

1. Find individual support values of each item by using the Apriori function and by setting the minimum threshold to an arbitrarily small value not to lose the support of an SKU. Additionally, find the throughput values of SKUs. Then, construct a metric that is constituted of the sum of these parameters.

2. Sort SKUs according to their support and throughput values in descending order.

3. Pick the first 3 SKUs from the list and match them with highly correlated SKUs by using confidence values which are the conditional probabilities (match them with their complements).
   As an example, assume that SKU_a, SKU_b, and SKU_c are the SKUs with the highest 3 support values.

   3.1. Algorithm searches for values $\mathrm{P}(SKU_i \mid SKU_a)$, $\forall i \in \{1, ..., m\} \setminus \{b, c\}$

   3.2. It picks the maximum value among these conditional probabilities. Thus, the SKU that SKU_a is highly correlated is established. This pair is stored.

   3.3. Assume i took value k as the result of the previous step. Now the algorithm searches for $\mathrm{P}(SKU_i \mid SKU_b)$, $\forall i \in \{1, ..., m\} \setminus \{a, c, k\}$.

   3.4. It picks the maximum value among these conditional probabilities. Thus, the SKU that SKU_b is highly correlated is established. This pair is stored.

   3.5. Assume i took value l as the result of the previous step. Now the algorithm searches for $\mathrm{P}(SKU_i \mid SKU_c)$, $\forall i \in \{1, ..., m\} \setminus \{a, b, k, l\}$.

3.6. It picks the maximum value among these conditional probabilities. Thus, the SKU that SKU_c is highly correlated is established. This pair is stored.

3.7. Assume i took value n as the result of the previous step. The group of 6 SKUs is formed as $\{SKU_a$,SKU_b,SKU_k, SKU_l, SKU_c, $SKU_n\}$.

3.8. These steps are repeated iteratively until all SKUs are placed in a group.

4. Sort groups of shopping cells according to their distance to the pool area.

5. Assign SKU groups with cell groups accordingly, e.g., the group of 6 having the most frequent 3 SKUs must be assigned to the cell group closest to the shopping area.

6. Run the assignment model for every group of SKUs and shopping areas in order to reallocate SKUs within groups.

12.3.2 Shopping Area Product Allocation Model

Sets: This model is planned to run a couple of times in a year, to capture the seasonality factor.

SKUs: $M = \{1, ..., m\}$

Shopping Areas: $N = \{1, ..., n\}$

There is a one-to-one correspondence between shopping areas and SKUs. Hence, $|M| = |N|$. Also, a shopping area can be empty but must be assigned to an SKU. In parallel, a new SKU must be assigned to a shopping area. It was observed that when an SKU, which has not been assigned to a shopping area, arrives then a new shopping area is created.

Parameters:

d_{jl}: distance between shopping areas j and l, $\forall j, l \in \{1, ..., n\}$,

c_{ik}: the confidence value of SKUs $\{i, k\}$, $\forall i, k \in \{1, ..., m\}$,

m: total number of SKUs to be allocated

n: total number of existing shopping areas

Decision Variables:

$$x_{ij} = \begin{cases} 1 & \text{if SKU } i \text{ is placed in shopping area } j \\ 0 & \text{otherwise} \qquad i \in \{1, ..., m\} \text{ and } j \in \{1, ..., n\} \end{cases}$$

$$y_{ijkl} = x_{ij}x_{kl} \qquad i, k \in \{1, ..., m\} \text{ and } j, l \in \{1, ..., n\}$$

Mathematical Model:

$$\textbf{min} \quad \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{m}\sum_{l=1}^{n} c_{ik}d_{jl}y_{ijkl}$$

$$\sum_{i=1}^{m} x_{ij} = 1 \qquad \forall j \in \{1, ..., n\} \tag{12.1}$$

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad \forall i \in \{1, ..., m\} \tag{12.2}$$

$$y_{ijkl} \geq x_{ij} + x_{kl} - 1 \quad \forall i, k \in \{1, ..., m\} \quad \forall j, l \in \{1, ..., n\} \tag{12.3}$$

$$y_{ijkl} \leq \frac{x_{ij} + x_{kl}}{2} \quad \forall i, k \in \{1, ..., m\} \quad \forall j, l \in \{1, ..., n\} \tag{12.4}$$

Objective Function and Constraints: The objective function minimizes the distance and confidence-based penalty by placing the products with high confidence values to closer shopping areas. Constraints (1) and (2) ensure that every product must be placed in one of the shopping cells, and a cell must be assigned with a specific product. Constraints (3) and (4) stand for linearization of the multiplication of x_{ij} and x_{kl}.

Rack Cells Product Pallet Allocation Model

Sets:

1. M: SKU of Eti products (h)
2. N: New arriving product pallets (i)
3. J: Rack cells (j)
4. K: Cell groups (k)

Parameters:

1. $$g_{jk} = \begin{cases} 1 & \text{if rack cell } j \text{ belongs to group } k \quad \forall j \in J, \forall k \in K \\ 0 & \text{otherwise} \end{cases}$$

2.

$$M_{ih} = \begin{cases} 1 & \text{if arriving pallet } i \text{ carries SKU } h \quad \forall i \in N, \forall h \in M \\ 0 & \text{otherwise} \end{cases}$$

3. d_{kh}: distance between cell group k and shopping area of SKU h $\quad \forall k \in K, \forall h \in M$

4. c_h: demand coefficient of SKU h $\qquad \forall h \in M$

Decision Variables:

$$x_{ij} = \begin{cases} 1 & \text{if received item } i \text{ is placed in cell } j \quad \forall i \in N, \forall j \in J \\ 0 & \text{otherwise} \end{cases}$$

Model:

$$\textbf{min} \quad \sum_{k \in K} \sum_{h \in M} \sum_{i \in N} \sum_{j \in J} \sum_{i \in N} x_{ij} g_{jk} d_{kh} c_h M_{ih} \tag{12.5}$$

$$\sum_{j \in J} x_{ij} = 1 \qquad \forall i \in N \tag{12.6}$$

$$\sum_{i \in N} x_{ij} \leq E_j \qquad \forall j \in J \tag{12.7}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i \in N \tag{12.8}$$

Objective Function and Constraints: The objective function minimizes the distance and demand amount-based penalty by assigning the product pallets closer to the respective SKU's shopping area. The first constraint, enumerated by (6) is driven by the assignment problem method. This constraint ensures that each pallet must be placed in a cell. The second constraint, enumerated by (7), assures that the model does not consider placing a pallet in a cell that is already occupied and no more than one pallet can be stored in a rack cell. Finally, the last constraint is the boundary constraint to introduce the variable to the model as a binary variable.

12.4 Verification and Validation

12.4.1 Degeneracy Testing

Degeneracy testing is essential in ensuring the reliability of algorithms and models. It involves subjecting them to various scenarios to determine their consistency and robustness. In this study, a subgroup of SKUs was reallocated within shopping areas to test the algorithm's sensitivity to demand changes in different regions. The highly frequent SKU, 486100, was assigned to the shopping area closest to the pool area, while the low-demand

SKU, 2895000, was assigned to the furthest shopping area. To test the model's sensitivity to demand changes, a hypothetical scenario was created, by manipulating the data, where SKU 2895000 was frequently ordered. After re-running the model, the SKU was assigned to a shopping area group closer to the pool area, indicating the algorithm's sensitivity to demand changes in different regions. That is, it is assigned to a shopping area group SH0017-SH0026 with its distance to the pool area smaller than its previous group (54.69 meters away from the pool area), by 46.32 meters. This degeneracy test ensures that the algorithm and model perform consistently and reliably in different scenarios.

0	1	2	3	4	5	6	7
SH0189-SH0195	4861100	1361200	1361300	4637300	1637400	2140100	3260400
SH0001-SH0006	1615300	2810900	4964700	3810100	3813100	2895000	0

Figure 12.2: Results for Two Groups of Shopping Areas and SKUs

12.4.2 Choice of the Confidence Parameter

To be consistent with the approach in the heuristic algorithm, the confidence parameter (c_{ik}) in the objective function of the model was also calculated by having the SKU with higher support as the antecedent, e.g., in the given part of the conditional probability. Another approach would be choosing $\max\{\mathrm{P}(i \mid k), \mathrm{P}(k \mid i)\}$. Thus, these approaches would result in a single value, that is c_{ik} would be equal to c_{ki}, $\forall\ k$, and i. A benchmarking process was also applied by utilizing these two approaches so as to analyze the percentage improvement or deterioration for the proposed configuration of the shopping area placement model.

12.4.3 Operational Validity on Data

The heuristic algorithm expects to have SKUs assigned to the nearest group of shopping areas (SH0189-SH0195) as being frequent/with high support values. In the left Figure 12.3, the output of the Python code calculating individual support values can be observed. As expected, half of the SKU group assigned to this shopping area group is constituted of the SKUs with the highest support values calculated in Python. In addition to this, the heuristic pairs these SKUs with their highly correlated SKUs, and this can be viewed in the right Figure 12.3.

12.5 Implementation and Integration

The user interface is designed using Python's PyQt package (Willman, 2020). It includes 6 different options which correspond to various applications of the warehouse product allocation. Figure 12.4 displays the screen-

Index	support	itemsets
22	0.246143179	1361300
188	0.241462992	4637300
38	0.238689548	1637400
92	0.229155833	2140100
154	0.219448778	3260400
21	0.211821806	1361200
64	0.208528341	1812300
195	0.200034668	4861100
173	0.199167967	3931100

	anteceden	consequen	confidence	
11307	**4637300**	**1637400**	**0.5735821966977745**	**max**
21891	4637300	2140100	0.4910265613783202	
30307	4637300	3260400	0.4587221823402728	
17174	4637300	1812300	0.44723618090452255	

	anteceden	consequen	confidence	
6455	**1361300**	**1361200**	0.6288732394366197	**max**
6815	1361300	1637400	0.40845070422535207	
7103	1361300	5837000	0.39859154929577467	
6998	1361300	3031700	0.39788732394366194	

	antecedents	conseq	confidence	
11306	1637400	~~4637300~~	0.5802469135802468	**Exception List**
11144	**1637400**	**2140100**	0.4923747276688453	**2nd max**
11251	1637400	3260400	0.4887436456063907	
11320	1637400	4861100	0.45461147421931736	

Figure 12.3: Highly Frequent SKUs Placed in the First Group of Shopping Areas and Maximum Confidence Value Pairs for the Highly Frequent SKUs

shot of the user interface. In certain sections of the user interface, an Excel file is needed to input historical demand data from ETİ, along with the latest SKU list and shopping area list. The reason for not performing the integration of the user interface with the ETİ database was to maintain confidentiality since ETİ managers want to keep their database inside the company. Therefore, the user interface was designed to prompt users to input the necessary information in Excel format instead.

At the beginning of May 2023, the user interface and user manual were given to ETİ for the pilot study. The study involved offering new product

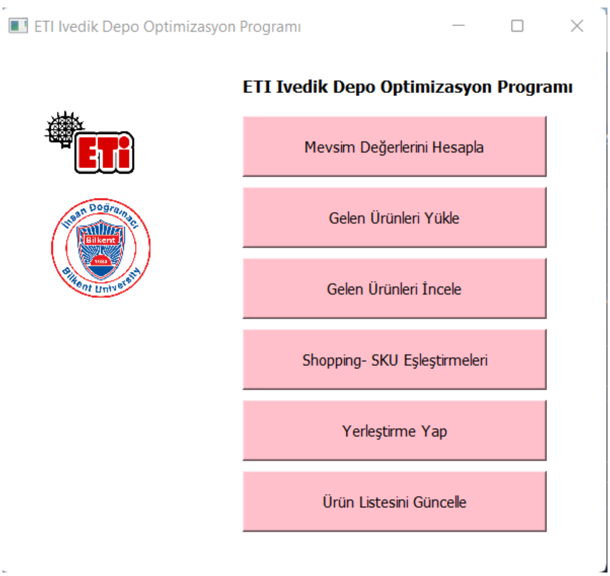


Figure 12.4: User Interface Main Page Screenshot

allocation of rack shelves and shopping areas using the developed system.

12.6 Benefits to the Company

Our product placement optimization in the rack shelves eliminates the workforce waste caused by the unnecessary roaming around the warehouse. After the implementation of our system, forklift operators will no longer have to travel long eliminable distances. Relocating products in the shopping area by observing the data, which also targets less roaming time in the warehouse. The project intends to increase the amount of work handled in a shift without changing the number of personnel or tools at all. The proposed systems were designed to be dynamic, which indicates that they will apply not just to only one future scenario but to all of the possible future demands, seasonalities, and other special warehouse layout conditions.

12.7 Conclusion and Future Advancements

The goal of the project was to reduce the distance between shopping areas and rack cell placement. In the current warehouse configuration, the distance traveled by an order picker is 763,311 meters, but in our proposed configuration, it is only 597,192 meters, on a yearly basis. Our configuration results in a 22% improvement in distance traveled annually for handling processes. For rack cells product placement, the unloading process of 23 trucks was compared between the model and the simulated real-life system. The model's objective function, which corresponds to the total SA distances and pool area distances, was 10% lower than the current system used by ETİ, indicating a shorter distance penalty of 47,142 meters compared to 52,465 meters. Considering the current ongoing method, it is not possible to provide the model with the pallets of SKUs' expiry dates before the truck is unloaded. After discussing with the project's Industrial Advisor, expiry date integration could be offered as an extension of our mathematical model if the ETI's current system allows it.

Bibliography

Li, Y., F. Méndez-Mediavilla, C. Temponi, J. W. Kim, and J. Jimenez (2021, 02). A heuristic storage location assignment based on frequent itemset classes to improve order picking operations. *Applied Sciences 11*, 1839.

Willman, J. (2020). *Beginning PyQt: A Hands-on Approach to GUI Programming*. Apress.

13 | Milk Run Operasyonları ve Malzeme Taşıma Süreçleri Eniyilemesi

Arçelik Buzdolabı İşletmesi

Proje Ekibi
Sena Aydın, Deniz Görgün, Aslı Kalaycı, Zareepa Pahore, Irmak Samur, Melih Safa Şimşek, Batuhan Yazıcıoğlu

Şirket Danışmanı
Cansu Demir
Üretim Planlama Mühendisi

Akademik Danışman
Prof. Dr. Oya Ekin Karaşan
Endüstri Mühendisliği Bölümü

Özet

Bu projede Arçelik Eskişehir fabrikasındaki milk run süreçlerinin geliştirilmesi hedeflenmiştir. Otomatik yönlendirmeli araç ve çekici malzeme taşıma araçlarına malzeme türleri atanarak ve en iyi milk run rotaları tasarlanarak verimli bir malzeme taşıma sistemi önerildi. Atama modeli ve zaman ekranlarıyla yetkilendirilmiş araç rotalama problemi sunuldu. Zaman ekranlarıyla yetkilendirilmiş araç rotalama problemini çözmek için Genetik Algoritma uygulandı. Şirkete kapsamlı bir karar destek sistemi sunmak için PySimpleGUI kullanarak kullanıcı arayüzü oluşturuldu.

Anahtar Sözcükler: milk run, malzeme taşıma, kuruluş içi lojistik, genetik algoritma

Milkrun Operations and Material Handling Processes Optimization

Abstract

The aim of this project is to improve milk run processes for Arçelik Eskişehir's Extension Factory 6x2. An efficient material handling system has been proposed by assigning material handling devices such as AGVs and tugger trains to material types and designing milk run routes for tugger trains. An assignment model and a Capacitated VRP with Time Windows model are presented. To solve the CVRPTW, Genetic Algorithm was implemented in Python and a user interface was developed in PySimpleGUI for stakeholders' ease of use. As a result of the proposed system, there was a 7% improvement in AGV utilization and a 54% decrease in distances travelled by the tugger trains.

Keywords: milk run, material handling, in-plant logistics, genetic algorithm

13.1 Company Information

Arçelik was established in 1955 in Sütlüce, İstanbul where the headquarters are located today. In the late 1950s, Arçelik produced Turkey's first washing machine and refrigerator. Today, it owns 12 brands internationally manufacturing high-end products for its customers. These brands operate in various fields and provide services to different countries. For example, Beko is one of the leading brands providing household appliances to other countries. Arçelik has 9 production facilities together with 15 R&D centers in Turkey. Each facility is responsible for producing a specific household appliance such as compressors in Eskişehir or dishwasher plant in Ankara.

13.2 System Description

In Arçelik's Eskişehir Refrigerator Factory, there are 112 workstations producing 225 refrigerator models using 8885 material types. 160 trucks supply 3400 different types of raw materials to the factory each day. These materials need to be delivered from the supermarket to workstations using two material handling devices: AGVs and Tugger Trains with a threshold time of 45 minutes or 90 minutes depending on the buffer capacity of the workstation. While AGVs are automated vehicles, tugger trains require operators. Operators in the supermarket can view the work orders on a screen sorted according to their criticalness as ascertained by the threshold values. When a material is required at a workstation, operators carry the boxes to an AGV which then delivers it automatically or transport the materials using tugger trains. There are currently 6 AGVs, each having a capacity of six boxes and each box having one type of material. They cannot deliver materials to multiple workstations along a route and can only visit one point in the factory and return. This is because AGV containers are loaded and unloaded by workers manually and stopping at multiple point would increase their workload. Additionally, AGVs have a high investment cost of 45000 Euros. Since the routes of the AGVs are predetermined, the main issue is increasing their utilization to ensure returns on the investments made and because they are more efficient than tugger trains.

However, certain material types are required in large amounts or have large dimensions and can only be carried by tugger trains. There are currently 7 tugger trains which can stop and deliver materials to multiple workstations along a route. For the tugger trains, milk run routes need to be designed and planned to improve efficiency as no milk run routes currently exist. There are 5 pre-determined unidirectional paths which tugger trains can travel along on the factory floor and the milk run routes must be designed taking these movement restrictions into account.

13.3 Problem Definition

Arçelik has initiated a project called Factory 6x2, an extension of the current Factory 6. With the extension factory, production capacity will be doubled and there may be delays in material handling due to work order accumulations in the system. Current AGV and tugger train numbers may no longer be sufficient and traffic across the factory floor may lead to blockages. Hence, the company intends to optimize the material handling processes to ensure an efficient production environment. The existing material distribution system will be integrated into the new assembly line built with this extension. As AGVs are expensive requiring an investment of 45000 Euros, only 3 AGVs will be purchased for the new assembly line in the extension factory. The remaining material distribution will be done by tugger trains. The aim of this project is to decide the material types which will be carried by AGVs and tugger trains, respectively while increasing the utilization of AGVs. Furthermore, milk run routes of tugger trains must be designed to prevent blockages and ensure efficient delivery of materials to the workstations.

13.4 Solution Approach

13.4.1 Mathematical Models

An assignment model was formulated to decide which material types would be assigned to which material handling devices i.e. AGVs or tugger trains.

This is a one-time strategic decision and Akturk and Yilmaz (1996) was utilized to develop the model. The assignment model initially aimed to increase utilization of AGVs to 80% as required by the company however, this constraint introduced bottlenecks in the system. As a result, to ensure feasibility, AGV utilization was increased to 68%. Furthermore, the model ensured that only those materials who have 80% of their demand met by AGVs are carried by them and the rest are assigned to tugger trains.

For the milk run routes, a Capacitated VRP with Time Windows model was developed. The time window in which materials will be delivered, their demand and workstation locations were considered. As the demand is highly stochastic and cannot be predicted, we take an hourly snapshot of the system to plan the milk run routes. For example, work order data with demands occurring between 9am-10am are considered and then the model is run on this data. Hence, milk run routes are decided on an hourly basis. However, for large data sizes e.g. 100 workstations, the VRP model does not run in a reasonable amount of time making it hard to implement. Therefore, to obtain a solution of the VRP for large data size in short amounts of time, we proceeded with the Genetic Algorithm. It will provide company with the sequence of drop-off points that tugger trains should follow in their milk run routes. You may refer to the appendices for details of the assignment model and CVRPTW.

13.4.2 Genetic Algorithm

We coded the genetic algorithm using Python to solve the CVRP with Time Windows. The genetic algorithm is based on the process of natural selection where a base population mutates and generates new offspring and iteratively improves in this manner until the fittest individuals are found. It consists of five phases: Generating Initial Population, Calculating Fitness, Selection, Crossover and Mutation (Okur et al., 2020).Once the genetic algorithm finished execution, it displays the best individuals in the population, their fitness values, and the final routes of the tugger trains to satisfy all the demand on time.

The parameters of the genetic algorithm such as crossover probability, mutation probability, population size and number of generations control the solution quality obtained. Number of generations and population size should be larger than the number of nodes (workstations) as this allows searching more of the solution space. The higher the crossover probability, the greater the solution space searched by the algorithm. However, increasing the crossover probability to 100% is not efficient as beyond a certain point, increasing the solution space searched does not generate any improvement in the objective function. Next, mutation occurs to introduce diversity in the population and to essentially, try to find if a better solution exists elsewhere in the solution space. The mutation probability is fixed to be less than 5% since higher values of this will lead to a totally random search and the algorithm will not produce meaningful results.

13.5 Validation

13.5.1 Assignment Model Validation

The assignment model is solved in Gurobi solver. The mathematical model considers the time elapsed between the supermarket and the drop-off points, demand of the material segments in the drop-off points within a shift, the total number of tours that an AGV can handle in a shift, the total operation time of an AGV, and the capacity of an AGV. For data selection, a systematic methodology was followed.

Considering 3 months' work-order data and annual production plan, we obtained material segment demand within a shift for the drop-off points. The current factory will operate with full capacity and the models that will be produced only in the extension factory only amount to a fraction of the models produced in the current factory. Hence, we generated demand considering that the extension factory operates with full capacity. By measuring the distance between workstations and drop-off points, the workstations are assigned to the closest drop-off points. Drop-off points' distance and time taken to travel from the supermarket is measured using the AutoCAD layout. Currently there are 6 AGVs covering 90% of the material segments' demand in the current factory however, the extension factory will operate with 3 AGVs. In order to validate our model, the model is first run with 6 AGVs and we observe a 7% optimality gap in Gurobi, when comparing the model with the current factory. Furthermore, if the drop-off point is close to the supermarket and its demand is low, the model ensures the demand to be fully covered by the AGVs. With the assignments provided by this model, 68% of the demand is covered by 3 AGVs in the extension factory and 780 out of 1148 boxes transported with AGVs in one shift.

13.5.2 Genetic Algorithm Validation

The purpose of the Genetic Algorithm is to find the milk run routes which minimize the total distance traveled. It is important to note that although the milk run routes are not fixed, there are five predetermined unidirectional paths along which tugger trains are allowed to travel. This restriction in tugger train movement has to be accounted for when merging workstations in a single route. Hence, workstations whose demand is delivered in the same route must belong to the same path along the factor floor. For validation, according to company work order data we created input data as shown in Figure 13.1. As previously mentioned, we consider an hourly snapshot as demand is uncertain and dynamic. Hence, in this figure the data represents the work orders between 8am-9am and the routes taken by tugger trains in the factory to satisfy the demand across the workstations.

Station	Routes	Demand	For Vlookup	X	Y	Earliest Time	Latest Time	Dummy Node Number
1	0-7-0	1	7	[illegible]	[illegible]	[illegible]	[illegible]	
2	0-22-0	6	22	[illegible]	[illegible]	[illegible]	[illegible]	
3	0-15-0	1	15	[illegible]	[illegible]	[illegible]	[illegible]	
4	0-55-0	4	55	[illegible]	[illegible]	[illegible]	[illegible]	
5	0-51-0	1	51	[illegible]	[illegible]	[illegible]	[illegible]	
6	0-28-0	1	28	[illegible]	[illegible]	[illegible]	[illegible]	
7	0-9-0	1	9	[illegible]	[illegible]	[illegible]	[illegible]	
8	0-9-0	1	9	[illegible]	[illegible]	[illegible]	[illegible]	75
9	0-56-0	1	56	[illegible]	[illegible]	[illegible]	[illegible]	
10	0-51-0	1	51	[illegible]	[illegible]	[illegible]	[illegible]	76
11	0-47-0	1	47	[illegible]	[illegible]	[illegible]	[illegible]	

Figure 13.1: Input Work Order Data

We used AutoCAD to calculate the distance matrix for GA and to find the distance traveled using the routes in the factory provided by the IA. Next, we ran the Genetic Algorithm for this demand data and obtained new milk run routes and compared the distance obtained by GA with the factory. We carried out these steps for the most interesting demand scenarios to estimate the percentage improvement. The routes obtained by GA for each unidirectional path can be seen in Figure 13.2. The arrows show the x coordinate which uniquely defines each path and next to it, the routes to

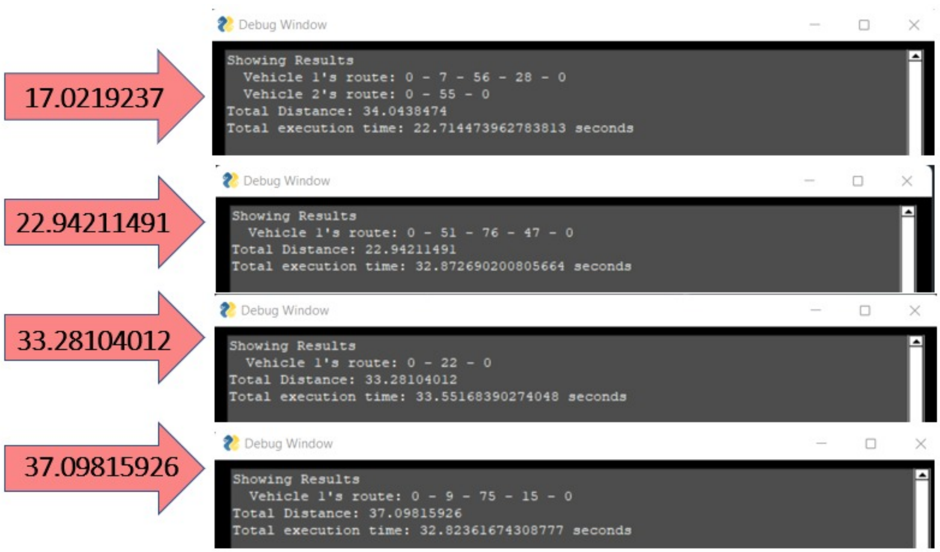


Figure 13.2: GA Output of Milk Run Routes for Each Path

deliver demand of the workstations which lie along each path can be seen. Figure 13.1 shows that in the factory operators intuitively decide when to deliver materials to workstations and only visit one workstation in each route and come back to the supermarket. However, through a systematic approach in the GA, in a single route the workstations which lie along the same path are merged and have their demands delivered together resulting in a high improvement rate of 54.75%.

Lastly, we have calibrated the parameters of the Genetic Algorithm to find the most appropriate values for our system by considering the improvement in the objective function which they generate. The best values of the parameters are shown in Figure 13.3. Increasing crossover probability, population size and number of generations beyond these values does not generate any further improvement in the objective function.

Population Size	**400**
Crossover Probability	**0.85**
Mutation Probability	**0.02**
Number of Generations	**300**

Figure 13.3: Calibration of Parameters for GA

13.6 Integration and Implementation

To integrate the decision support system provided by our Genetic Algorithm and Assignment Model, a user interface was created using PySimpleGUI which can be seen in Figure 13.4. For the company officials' ease of use, a user manual has been designed. As the Genetic Algorithm is run in Python, which is an open-source software, there are no additional investments required. The company has work orders data including workstation coordinates, demand and time windows in Excel which is then converted to .txt format for the Python code to ensure compatibility. This decision support system can run data not only for the extension factory but can be utilized for any future factories in other locations as well. Lastly, the maximum execution time of the Genetic Algorithm is 47.5 seconds making it easily implementable on an hourly basis.

13.7 Benefits to the Company

Comparing the AGVs' demand coverage rate in the current system, a 17% improvement was achieved with the assignment model. However, we imposed an additional constraint that if less than 80% of a material segment's

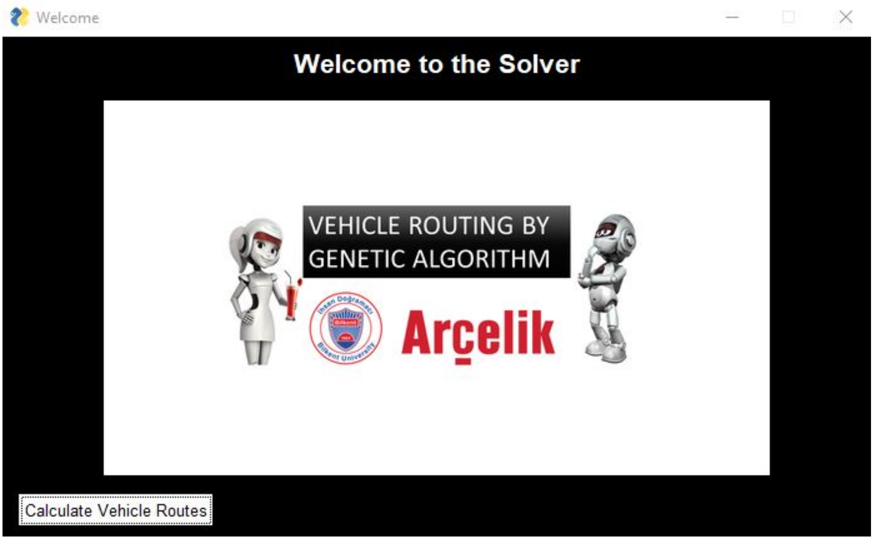


Figure 13.4: User Interface

demand is satisfied by an AGV, this material segment is assigned to tugger trains. Although, the final improvement rate then is 7% it prevents potential bottlenecks in the system and ensure efficient utilization of AGVs. Demand coverage rate for three difference scenarios can be seen in the figure below.

When comparing the milk run routes obtained by the Genetic Algorithm and by our CVRPTW in Cplex for a small dataset, the optimality gap is found to be approximately, 14%. By comparing the autocad layout and current routes followed in the factory with the GA's milk run routes we observe a 54.75% improvement in distances travelled. This large improvement is due to the fact that operators deliver demand to only a single workstation and come back to the supermarket whereas the GA merges multiple workstations along a single route. Hence, the distance traveled by GA to deliver all the workstation demand per hour is substantially less compared to the current system.

13.8 Conclusion

With doubling of the production capacity as a result of the new extension Factory 6x2, Arçelik was facing problems due to blockages along factory floor and inefficient material handling. By using the assignment model, we determined the material handling vehicles and material type assignments which would result in improved AGV vehicle utilization. The CVRPTW solved by the Genetic Algorithm provides milk run routes of tugger train based on hourly work orders data and decreases the distances traveled by

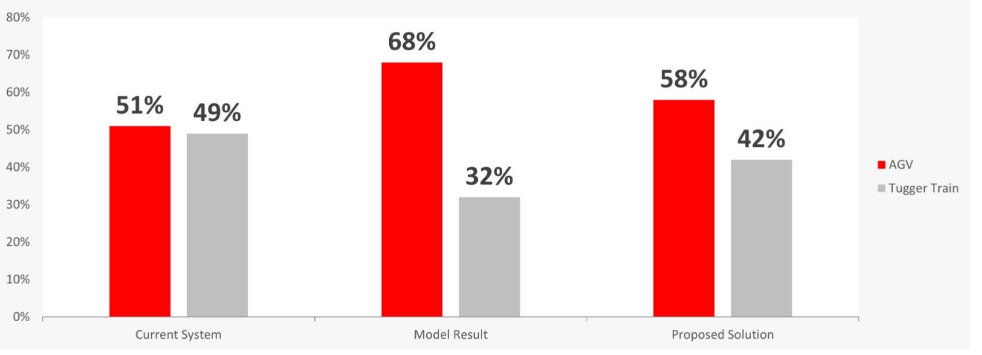


Figure 13.5: Demand Coverage Rate per Vehicle Type

tugger trains. The user interface is designed to make the decision support compatible with the company's current system and ensures efficient material handling processes in the extension factory. In the future, the company can use this system for new factories in other locations as well.

Bibliography

Akturk, M. S. and H. Yilmaz (1996). Scheduling of automated guided vehicles in a decision making hierarchy. *International Journal of Production Research 34*(2), 577–591.

Okur, E., M. Atlas, et al. (2020). Araç rotalama probleminin genetik algoritma ile çözümü.

Appendices

13.A Assignment Model

Sets
A = {1,2,3}: Set of AGVs
M = {1,..,M}: Set of material types
D = {1,..,D}: Set of drop-off points
K = {1,..,K}: Set of AGV tours

Parameters
l_d = Time taken to reach drop-off point $d \in D$
dem_{md} = Demand of material $m \in M$ at drop-off point $d \in D$
T_a = Total time AGV $a \in A$ can operate in a day
Q_a = Capacity of AGV $a \in A$

Decision Variables

$$x_{amdk} = \begin{cases} 1, & \text{if } m \in M \text{ is delivered by } a \in A \text{ along } k \in K \text{ to } d \in D \\ 0, & \text{otherwise} \end{cases}$$

$$y_{amdk} = \text{amount of } m \in M \text{ delivered by } a \in A \text{ along } k \in K \text{ to } d \in D$$

$$z_{adk} = \begin{cases} 1, & \text{if } D \in D \text{ iss visited along } k \in K \text{ of } a \in \text{A} \\ 0, & \text{otherwise} \end{cases}$$

Model

$$\max \sum_a \sum_m \sum_d \sum_k y_{amdk}$$

s.t.

$$y_{amdk} \leq dem_{md} x_{amdk} \qquad \forall a, m, d, k \qquad (1)$$

$$\sum_a x_{amdk} \leq 1 \qquad \forall m, d, k \qquad (2)$$

$$y_{amdk} \leq dem_{md} z_{adk} \qquad \forall a, m, d, k \qquad (3)$$

$$\sum_m \sum_d y_{amdk} \leq Q_a \qquad \forall a, k \qquad (4)$$

$$\sum_m \sum_d \sum_k x_{amdk} l_d \leq T_a \qquad \forall a \qquad (5)$$

$$\sum_m \sum_d \sum_k x_{amdk} l_d \geq 0.8 T_a \qquad \forall a \qquad (6)$$

$$\sum_d z_{adk} \leq 1 \qquad \forall a, k \qquad (7)$$

$$z_{adk} \leq \sum_m x_{amdk} \qquad \forall a, d, k \qquad (8)$$

$$\sum_m x_{amdk} \leq M z_{adk} \qquad \forall a, d, k \qquad (9)$$

$$x_{amdk} \leq y_{amdk} \qquad \forall a, m, d, k \qquad (10)$$

$$\sum_a \sum_d x_{amdk} \leq 1 \qquad \forall m, k \qquad (11)$$

$$\sum_a \sum_k y_{amdk} \leq dem_{md} \qquad \forall m, d \qquad (12)$$

$$x_{amdk} \in \{0, 1\} \qquad \forall a, m, d, k \qquad (13)$$

$$z_{adk} \in \{0, 1\} \qquad \forall a, d, k \qquad (14)$$

$$y_{amdk} \geq 0 \qquad \forall a, m, d, k \qquad (15)$$

13.B CVRPTW Model

Sets

I = {1,..,i}: Set of Nodes
J = {1,..,j}: Set of Nodes
Node 1 is the depot

Parameters

Q = Capacity of tugger train
D_j = Demand of drop-off point j, $j \in J$
s_i = Service time of point i, $i \in I$
t_{ij} = Travel time from point $i \in I$ to point $j \in J$
m = Number of available tugger trains taken from the user
e_i = The earliest time service can start at point i, $i \in I$
l_i = The latest time service can end at drop-off point i, $i \in I$

Decision Variables

$$x_{ij} = \begin{cases} 1, & \text{if tugger train travels from point } i \in I \text{ to point } j \in J \\ 0, & \text{otherwise} \end{cases}$$

$$z_j = \begin{cases} 1, & \text{If drop-off point } j \text{ is visited, } j \in J \\ 0, & \text{otherwise} \end{cases}$$

t_i = Time right after leaving point i, $i \in I$

u_i = Load of vehicle right after leaving point i, $i \in I$

Model

$$\min \sum_i \sum_j t_{ij} x_{ij}$$

s.t.

$$\sum_i x_{ij} = z_j \qquad \forall j, j \neq 1, \tag{1}$$

$$\sum_i x_{ij} = \sum_i x_{ji} \qquad \forall j, \tag{2}$$

$$\sum_j x_{1j} \leq m \tag{3}$$

$$M z_j >= D_j \qquad \forall j, j \neq 1 \tag{4}$$

$$z_j \leq D_j \qquad \forall j, j \neq 1 \tag{5}$$

$$u_j \geq u_i - D_j - M(1 - x_{ij}) \qquad \forall i, j, j \neq 1 \tag{6}$$

$$u_i \leq Q \qquad \forall i \tag{7}$$

$$t_i - t_j + Mx_{ij} \le M - (t_{ij} + s_i)x_{ij} \qquad \forall i, j, j \neq 1 \tag{8}$$

$$s_i + e_i \le t_i \le l_i \qquad \forall i \tag{9}$$

$$\sum_{j, j \neq i} (x_{ij} - x_{ji}) = 0 \qquad \forall i \tag{10}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i, j \tag{11}$$

$$z_j \in \{0, 1\} \qquad \forall j \tag{12}$$

$$t_i, u_i \ge 0 \qquad \forall i \tag{13}$$

14 Mobil Uygulamalar İçin En İyi Reklam Taban Fiyatları

ReklamUp

Proje Ekibi
Selim Enes Altuntaş, Elif Arslan, Deniz Kemal Konar,
Batuhan Oğuz, Mustafa Temtek, Can Uryan, Turgut Can Zeybek

Şirket Danışmanı
Ayşegül Yıldırım
ReklamUp, İcra Kurulu Başkanı

Akademik Danışman
Prof. Dr. Mustafa Ç. Pınar
Endüstri Mühendisliği Bölümü

Özet

Reklam aracılığı, yayıncıların envanterlerini doldurmak için birden fazla reklam ağı kullanarak reklam gelirlerini artırmalarına olanak tanıyan bir süreçtir. Reklam aracılık platformları bu süreci otomatikleştirerek yayıncıların reklamdan para kazanma stratejilerini enyileştirmelerine ve gelir potansiyellerini en üst düzeye çıkarmalarına yardımcı olur. Bu çalışmada, mevcut ReklamUp sistemi ile ilgili konuları, örneğin taban fiyatlarının gelişigüzel belirlenmesini, inceledik. Daha sonra en uygun taban fiyatlarını sistematik olarak belirleyebileceğimiz ve müşterilerin gelirini artırabileceğimiz bir model ortaya koyduk ve algoritmamızı bu soruna bir çözüm olarak önermekteyiz.

Anahtar Sözcükler: Gerçek-zamanlı artırma-eksiltme, gelir yönetimi, taban fiyat, karşılama oranı

Best Online Advertising Floor Prices for Mobile Applications

Abstract

Ad mediation is a process that allows publishers to increase their ad revenue by using multiple ad networks to fill their inventory. platforms automate this process, helping publishers optimize their ad monetization strategies and maximize their revenue potential. ReklamUp is an ad mediation company that struggles with the systematic determination of floor prices in this novel sector without any precedented best practices. In this study, we state the issues regarding the current system of ReklamUp -the arbitrary determination of floor prices. We then present a model where we can systematically determine the optimal floor prices and increase the clients' revenue and propose our algorithm as a solution to this problem.

Keywords: Ad Mediation, Real-time bidding, revenue management, floor pricing, ad request, fill rate.

14.1 System and Problem Definition

14.1.1 Analysis of the Current System

App developers try to display ads with the right strategy to increase their earnings, while advertisers create inventories (ads) to maximize their exposure. ReklamUp is a mediator firm that maximizes its clients' (the app developers') revenue.

In real-time bidding, demand-side platforms (DSP) such as Google Ad Manager attempt to find the best-priced ad for their customers' ad spots. For every real-time bidding, a is set to commence bidding. Applications take part in waterfalls in order to utilize the vast pool of data that consists of multiple platforms. In the waterfall, the predetermined floor prices are sorted in descending order. After that, the query starts from the highest floor price and if the floor price responds to the query, then it means that there, in fact, is an advertiser willing to pay your designated fee for your vacant spot.

After a positive response is generated for the specific query, ReklamUp buys the vacant ad spot for the designated floor price. After receiving the floor price bids, the system generates an, generally higher than the determined floor price by minimal margins. ReklamUp's revenue is the amount surpassing the determined floor price.

Since the bidding takes place almost instantaneously, the process isn't necessarily real-time. The advertisers set a boundary for their upper limit where the system won't consider them after surpassing their particular limitation for the vacant ad spot.

Bids are affected by various factors, such as the country where the ad is displayed (tier of the country), the device, and the type of advertisement (banner, interstitial, and rewarded video). The country's tier depends on the prices they are willing to pay. The higher prices a country is willing to pay, the higher its tier becomes; hence we, as the mediator, know that they most likely are willing to pay more for our clients' vacant spots. Different floor prices should be determined depending on those factors. Hence, ReklamUp determines floor prices for each ad type for three different tiers of countries and various device types.

14.1.2 The Problem and Its Scope

The main issue with the current system is the arbitrary determination of floor prices, which are based on intuition and heavily rely upon the unreliable human factor. The lack of an algorithm that can be monitored and optimized causes the clients and, in turn, ReklamUp to leave money on the table. Another issue is that when the number of decision variables increases (tier of the country, device type, etc.), the unreliable trial-and-error method makes the process that much more arduous for everyone involved.

Though technically, the bidding is a dynamic problem, as stated by Afshar et al. (2020), we have decided to treat it as a stochastic one since it happens in mere moments. We analyzed the historical data stored within the bidding system -that is otherwise not stochastically utilized by ReklamUp. Then, we fitted the floor price vs. relevant metrics distributions using stochastic methods and developed a combinatorial algorithm that yields the specific combination of five revenue-maximizing floor prices. We have heavily utilized the pricing and revenue optimization methods in the modeling part of this strategy.

14.2 Solution Approach

Conceptual Model is described by Figure 14.1. We give the mathematical model below.

Sets:

i = (Set of Floor Prices)
j = (Set of Floor Prices)

Parameters:

Ar_i = total ad request number of floor price i given in the dataset
Fr_i = fill rate percentage of floor price i given in the dataset
R_i = Total revenue estimated from floor price i
n = Number of Floor Prices

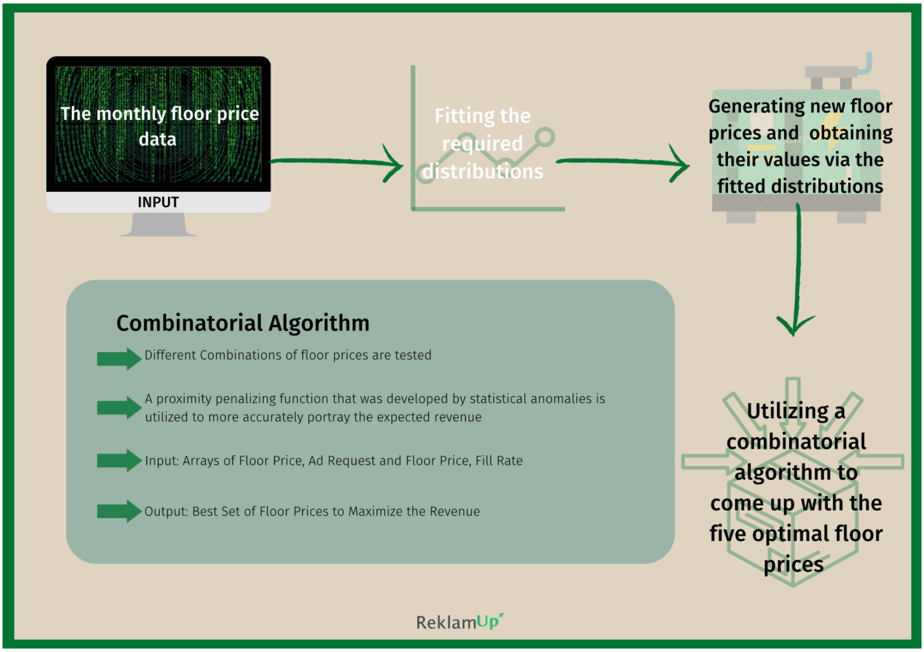


Figure 14.1: Conceptual Model of Our Solution

$$M = \frac{\sum_{Min(i)}^{Max(i)} i}{n}$$

Decision Variables:

$$X_i = \begin{cases} 1, & \text{if floor price i is chosen} \\ 0, & \text{otherwise} \end{cases}$$

$$X_j = \begin{cases} 1, & \text{if floor price j is chosen} \\ 0, & \text{otherwise} \end{cases}$$

$$Y_{i,j} = \begin{cases} 1, & \text{if } X_i \text{ and } X_j \text{ equals to 1} \\ 0, & \text{otherwise} \end{cases}$$

$$P1_{i,j} = \begin{cases} 1, & \text{if } |\,(X_i * i) - (X_j * j)\,| \leq \frac{M}{5} \\ 0, & \text{otherwise} \end{cases}$$

$$P2_{i,j} = \begin{cases} 1, & \text{if } \frac{M}{5} < |\,(X_i * i) - (X_j * j)\,| \leq \frac{M}{2} \\ 0, & \text{otherwise} \end{cases}$$

$$P3_{i,j} = \begin{cases} 1, & \text{if } \frac{M}{2} < |\,(X_i * i) - (X_j * j)\,| \leq M \\ 0, & \text{otherwise} \end{cases}$$

P = Penalty Cost

Objective Function: Max $\sum_{Min(i)}^{Max(i)} (R_i * X_i) - P$

Constraints:

- $R_i = \frac{Ar_i * Fr_i * i}{1000}$ ∀ i
- $\sum_{Min(i)}^{Max(i)} X_i = 5$
- $Y_{i,j} \geq X_i + X_j - 1$ ∀ i,j where i ≠ j
- $Y_{i,j} \leq X_i$ ∀ i,j where i ≠ j
- $Y_{i,j} \geq X_j$ ∀ i,j where i ≠ j
- $(i - j) * Y_{i,j} \geq (\frac{M}{n} - \mid i - j \mid)$
- $P = \sum_i \sum_j ((P3_{i,j} * 0.1 * R_i) + (P2_{i,j} * 0.2 * R_i) + (P1_{i,j} * 0.3 * R_i))$ where j > i
- $Y_{i,j} \geq P1_{i,j} + P2_{i,j} + P3_{i,j}$ ∀ i,j where i ≠ j
- $X_i, X_j, Y_{i,j}, P1_{i,j}, P2_{i,j}, P3_{i,j} \epsilon$ {0,1}

The objective is to maximize the expected revenue generated by the auction, subject to the following constraints:

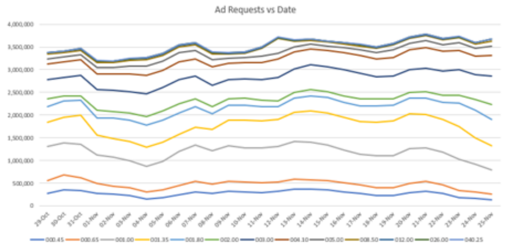
- $P_{min} \leq P_i \leq P_{max} \ \forall \ I$
- $P_i \neq P_j \ where \ i \neq j$
- $|P_i - P_j| \geq \sigma \ \forall \ i, j \ where \ i \neq j$
- $\sum_{i \epsilon S} P_i * F_i * A_i \geq R_{mine}$
- $N_{min} \leq \sum_{i \epsilon S} A_i \leq N_{max}$
- $N_{min} \leq \sum_{i \epsilon S} (P(i) = P_i) * A_i \leq N_{max} \ \forall \ I$
- $\sum_{i \epsilon S} (P(i) = P_i) * A_i - \sum_{i \epsilon S} (P(i) = P_{i+1}) * A_i \leq M \ \forall \ 1 \leq i \leq 4$

14.2.1 Solution Method

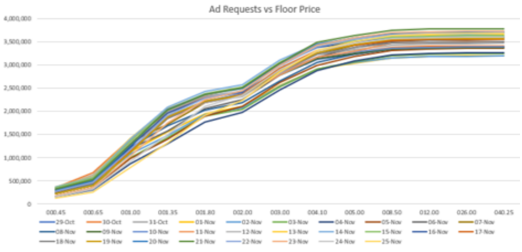
Our solution method starts by receiving input from the company. This input contains the monthly restricted waterfall data for a specific application, which includes ad request, average eCPM, and revenue values for each floor price. The data must be from a single platform for a specific ad type to achieve neater results without statistical anomalies. The algorithm separates the raw data into days and calculates the additional metrics for our solution to the original data provided by the company (Fill Rate and App Revenue). An example will best describe the reasoning behind calculating these daily average values. Suppose we have the floor price of two dollars throughout the thirty days of data we have received as input. We sum the statistics of \$2 over thirty days and divide them by thirty to obtain the average value of the desired statistics. This is done to achieve precise results. Our algorithm then receives the data from the Excel spreadsheet in the same format the company can download from the Google Ad Manager Portal. It utilizes the average data it computed to determine the best-fit distribution for the Fill Rate. For ad request, we use for intrapolation. Because of the waterfall logic of the auctions we know that all the ad request graphs are going to have increasingly less additional ad requests as the floor price increases. Due to this phenomena we decided to use piecewise linear functions to intrapolate the ad requests values. After fitting, the algorithm starts generating new floor prices from the monthly averages of the data. The algorithm generates 30 equally distanced new floor prices between the minimum and the maximum floor prices the company decided to use for the previous month. By trial and error method, we have determined that dividing the interval into 30 equally distanced floor prices is a good compromise between expanding the prospective pool of floor prices and minimizing the manipulation of the end results. The algorithm then plugs all the generated and pre-existing values into the aforementioned fits to estimate the and. This (estimating the pre-existing values' and the generated values' ad requests and fill rates) is done to preserve the waterfall logic. In the waterfall, the ad requests steadily drop with the price; however, this might not be the case if we only estimate the generated floor prices' ad requests. Hence we fit comprehensively.

Also, we graphed the data with the pre-existing and added metrics for us to visualize the relationship between the statistics and understand the dispositions of the waterfalls for the respective days and prices as in Figure 14.2. By determining the best fill rate distribution and utilizing the piecewise intrapolation technique for the ad request, we can estimate the expected revenue of the new floor prices we have generated. After obtaining these values by the appropriate fits, we calculate the expected app revenue by dividing the impressions (Impressions = Fill Rate x Ad Request) by a thousand and multiplying this value with the floor prices. This part aims to develop an understanding of the behavior of the various floor prices and expand the pool of prospective ones. Eventually, we estimated the impression, ad request, and app revenue for the newly generated floor price values, giving us a profound insight into the bidding characteristics of the specific parameters we are observing.

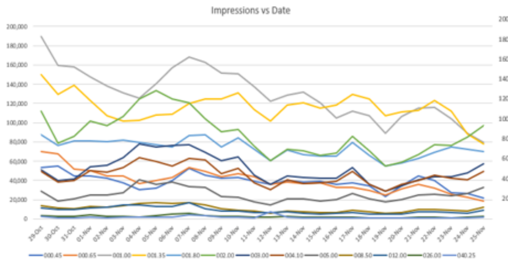
Combinatorial Algorithm In general, combinatorial algorithms involve systematically generating all possible configurations of a given set of objects and then selecting the one(s) that satisfy certain conditions or con-



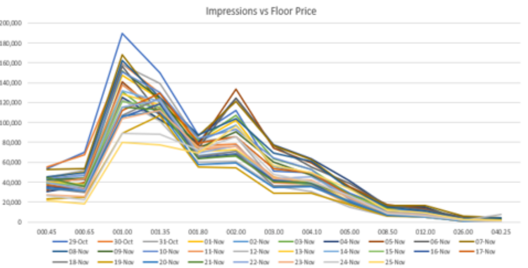
Graph 1: Ad Requests per Day for Each Floor Price



Graph 2: Ad Requests per Floor Price for Each Day



Graph 3: Impressions per Day for Each Floor Price



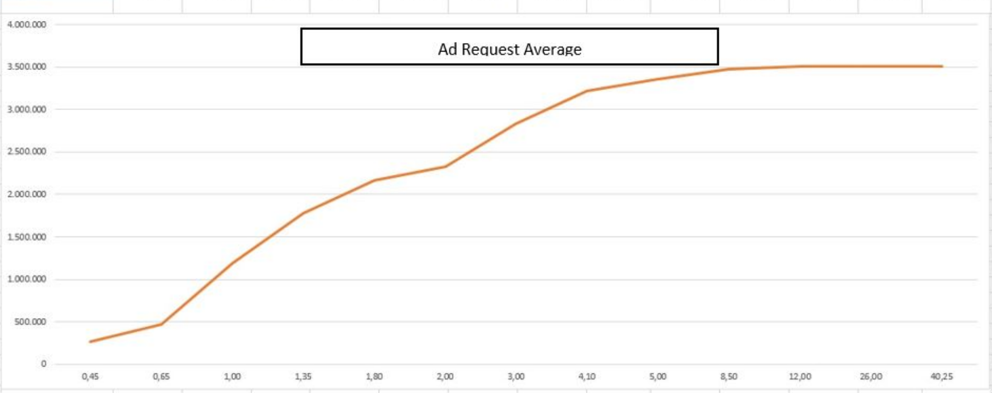
Graph 4: Impressions per Floor Price for Each Day

Figure 14.2: Daily graphs

straints. Some examples of combinatorial algorithms include the brute-force approach, which involves testing every possible combination until a valid solution is found, and the dynamic programming approach, which uses a recursive process to break down a problem into smaller subproblems that can be solved more efficiently. Since the novel computers are advanced, and the expected data size is mostly manageable, we decided to use the brute-force approach for our specific optimization problem. The crux of our algorithm lies with the proximity penalizing function that we have developed by analyzing the raw data that was presented to us. In the waterfall, the fill rates are supposed to increase as the floor price decreases; see Figure 14.3. However, in certain close proximity cases, we have observed a steady shift, if not a slight decrease, in fill rate as the floor price gets lower. Thanks to these anomalies, we were able to quickly devise a simple yet efficient proximity penalizing function to make sure that our results reflected that of actual scenarios.

We need to use a proximity penalizing function because of the adverse effects of the close floor prices on each other. Although we want to cover our bases by ensuring that we don't lose the potential revenue by overspreading our five-floor prices, the opposite is just as -if not more detrimental to our income. For example, by choosing floor prices of \$1 and \$1.5, we are reducing the effectiveness by decreasing the interval of additional bids we attempt to receive. By bidding every \$0.5, we are trying to get all the bids

Floor Price	0,45	0,65	1,00	1,35	1,80	2,00	3,00	4,10	5,00	8,50	12,00	26,00	40,25
Average	269.672	472.737	1.185.099	1.774.077	2.165.313	2.325.489	2.827.770	3.215.486	3.361.429	3.476.634	3.504.243	3.510.304	3.514.595



Floor price	0,45	0,65	1,00	1,35	1,80	2,00	3,00	4,10	5,00	8,50	12,00	26,00	40,25
Average	0,1472	0,08502	0,11045	0,06584	0,03441	0,03839	0,01852	0,01425	0,00755	0,0031	0,00259	0,00072	0,00056

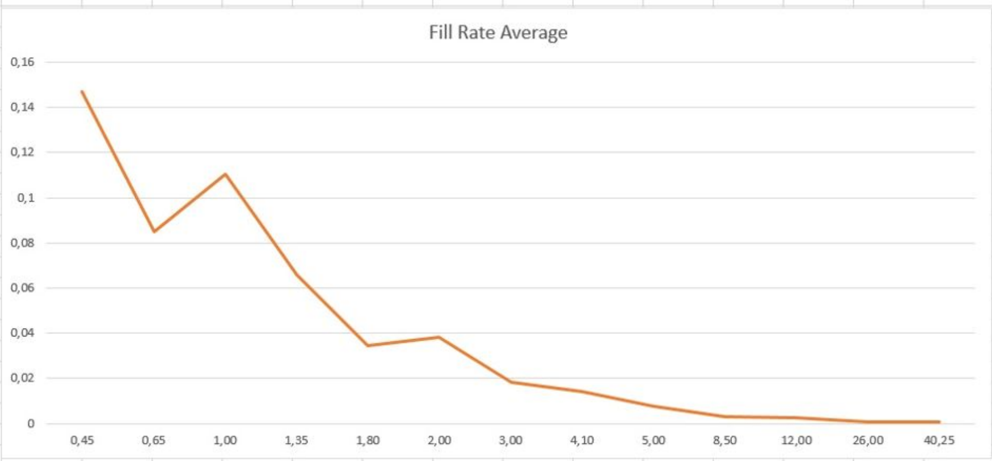


Figure 14.3: Ad Request and Fill Rate vs Floor Price

within that tiny interval, whereas a better approach would have been to spread them further apart. However spreading the floor prices too much may result in one of our competitors stepping in between our prices and winning the auction for themselves. So, “How further?” is the question that our algorithm solves for the company and all the end users.

14.3 Validation

For the validation, we utilized the current system's historic data and compared to the results of proposed mathematical model and algorithm. We compared the output of our proposed solution, our five floor price proposal, with the output of historical data.

We first, acquired two-month historic data from the company and ran the model in the first months and checked if it was similar to the optimal floor price points in the second month. Later we used hypothesis testing and checked whether those two months had similar values.

We fitted the distribution to estimate competitor floor price bids. To measure the validity of this distribution, we assigned a hypothesis test to the prices we have found and the prices we have estimated. Another possible way to check this is to test how well it fits with actuality by extracting one of our data and estimating it.

Later, the company provided us with the most recent 30-day data. We recommended them with five floor prices that will potentially yield the highest revenue margin. The data we provided currently undergoes an process in one of the company's apps. For the testing, 5% of the ads on the waterfall will be given the prices we recommend. After a week, the data results are compared to those acquired with the company's old practices. Here, we are trying to validate, if our percentage were 100% instead of 5%, whether the algorithm would yield higher results. To control this, we created a statistical infrastructure which consists of the.

14.4 Benefits to the Company

The success of the customers of ReklamUp is essential for its survival. With the decision support system developed in this project, there will be an increase in customers'—therefore, ReklamUp's—income. The program we deliver at the end of this project will allow ReklamUp to increase capital for its future endeavors. To observe the improvement, we conduct a benchmark analysis by using our proposed solutions in an existing app. We are still waiting on the results from the company to better yield a percentage increase in the generated revenue.

ReklamUp's customers can work with more than one Google Ad Manager consultant; hence, delivering better results than the competition is crucial for ReklamUp's profitability. Therefore it is not only essential for ReklamUp to provide satisfactory results to their clients, but also to win the inherent competition between the Google Ad Manager companies as well.

Another benefit of introducing our decision support system is that the process runs with minimum issues for a long time, and the company's earnings will be near optimal. The novel system introduced by us will not depend on user experience, meaning that a fresh employee with almost no experience can get the hang of it in a matter of minutes thanks to the user manual we deliver to the company, allowing for more workforce to work more effectively for a better, brighter future.

14.5 Integration and Conclusions

The pre-existing method of bidding is the trial and error method. In which the highly experienced employees use their expertise and prior knowledge to develop educated guesses to predict the bidding characteristics of the specific auction. They utilize Excel to generate Google's data sheets regarding their previous auction statistics and determine the best floor prices thereafter. Since they were already proficient in Microsoft Excel -seeing as they were running their day-to-day operations through it- it would be most natural for the company if we also based our code on Microsoft Excel. As our algorithm consists of two parts, Excel and Python, and Python is free to download and easy to use, the integration process will be seamless. A basic schematic for implementation can be seen in Figure 14.4. The raw data is in the form that Google generates, so the company doesn't have to format its raw data manually; the algorithm takes care of that. After uploading the data to our algorithm, the optimal floor prices based on the previous auction performances are a button click away.

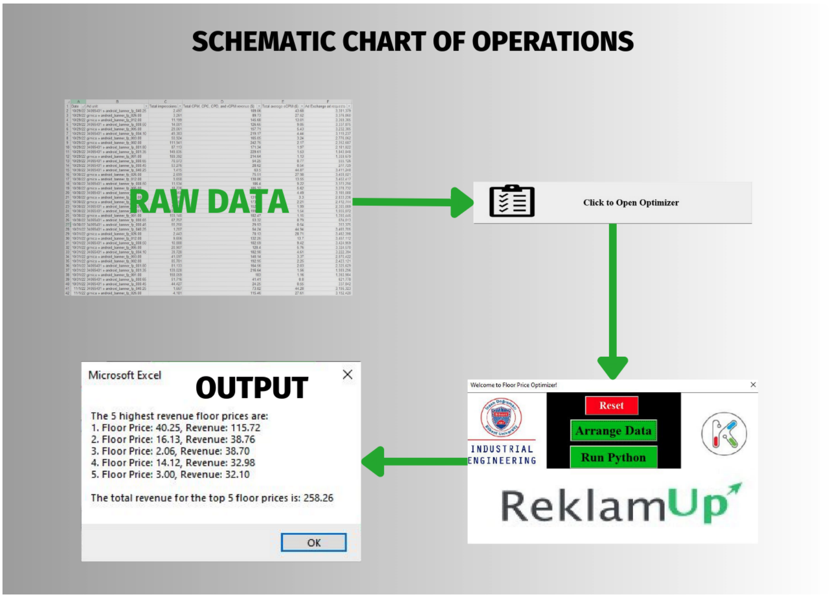


Figure 14.4: Schematic Chart of Operations

To conclude, our algorithm provides an efficient and easy-to-implement method for the floor price determination problem of ReklamUp. By developing the code in an environment that they are already familiar in and keeping the unnecessary complexities at a minimum, we have enabled the company to achieve better results and provided them with a tool that they can understand in no time.

The determination of floor prices depends on 3 seperate factors; ad type, device type, and user location. The pilot datasets that we have received from the company whilst developing a solution approach for this problem was for Banner ad type, and Android users, and in United States of America. So our pilot study has been conducted for a specific version of this determination problem. However as the arbitrary determination of floor prices are a major setback that reduces the effectiveness and the revenue we strongly encourage ReklamUp to implement a modified version of our algorithm for the other variations of the determination problem (For apple users, other ad types, and different countries).

Bibliography

Afshar, R. R., Y. Zhang, M. Firat, U. Kaymak, A. I. Metin, G. S. Tarakcioglu, and C. Bas (2020). Reserve price optimization with header bidding and ad exchange. In *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE.

Sipariş Karşılama Merkezlerine Envanter Paylaştıran Proaktif Karar Destek Sistemi | 15

OPLOG

Proje Ekibi
Zeynep Altıner, Ali Fuat Barış, Simge Çınar,
Sena Ceren Göktaş, Hüseyin Şahan, Defne Tan, Yağız Yaman

Şirket Danışmanı
Can Ayhan
CEO Ofis

Akademik Danışman
Doç. Dr. Ayşe Selin Kocaman
Endüstri Mühendisliği Bölümü

Özet

OPLOG e-ticaret firmalarına bütüncül bir sipariş karşılama hizmeti sunan bir şirkettir. Firmalar her hafta depolara yeni tedarik yollamaktadır. Gelen tedarik depolara yanlış bir oranda dağıtıldığında teslimat süreleri uzamakta ve müşteri memnuniyeti azalmaktadır. Bu proje, OPLOG'a sipariş teslimat süresini enazlayacak şekilde müşterilerinden gelen tedarik miktarlarını beş deposu arasında paylaştırma kararını vermesini sağlayan proaktif bir sistem sunulmasını amaçlamaktadır. Sistem ham satış verisini işleyerek şehir ve ürünün satış dağılımına göre çeşitli metotlarla talep tahmini yapar ve tam sayı programlama modeli ile envanteri depolara dağıtma kararını verir. Farklı senaryolar detaylıca düşünülerek kullanıcı dostu bir arayüz teslim edilmiştir. Tasarlanan sistem, şirketin kendi sistemine göre toplam teslimat süresinde %12,55'lik bir iyileşme sağlamıştır.

Anahtar Sözcükler: Envanter yönetimi, tahminleme, proaktif aktarmalı ulaştırma, depo ağı, envanter paylaştırma.

Pro-active Decision Support System for Inventory Allocation to Fulfillment Centers

Abstract

OPLOG is a company that provides a comprehensive order fulfillment service to e-commerce companies. Every week, companies send new supplies to warehouses. When the incoming supply is distributed incorrectly, delivery times are prolonged and customer satisfaction decreases. This project aims to provide OPLOG with a proactive system that minimizes the delivery time by deciding how to distribute the supply amounts from their customers among five warehouses. The system processes raw sales data and uses various methods to forecast demand based on the sales distribution by city and product. It then makes inventory distribution decisions among warehouses using an integer programming model. A user-friendly interface was delivered, taking into account various possible scenarios. The designed system resulted in a 12.55% improvement in total delivery time when compared to the company's current system.

Keywords: Inventory management, forecast, proactive transsshipment, fulfillment network, inventory allocation.

15.1 OPLOG

OPLOG provides order fulfillment (fourth-party logistics service) to its customers. It works integrated with the most popular sales platforms, and its customers are e-commerce companies that operate their business on the sales platforms, such as Hepsiburada and Trendyol, in addition to the ones that sell their products from their own platforms such as Mugo and Sony.

15.1.1 E-commerce Fulfillment System Analysis

Currently, OPLOG has more than 10 million orders registered into their system, and among them, 310.000 Stock Keeping Units (SKU) are identified. However, only around 65.000 of these SKUs are actively observed in the system with full information. OPLOG keeps customers' products in warehouses to prepare orders and manage the delivery when needed by providing end-to-end fulfillment services. The order statuses are observed, and the orders are received, stored, arranged, and forwarded to the relevant courier company respectively after this integration. Subsequently, the products are delivered to the end-customers by courier firms. End-customers, who are defined as the company or person who receives the delivery from the courier, can be local stores or individual customers. There are three fulfillment centers planned to be opened in Istanbul, Ankara, and Izmir in addition to the three warehouses that are currently active and located in Kocaeli. OPLOG aims to reduce delivery times with new fulfillment centers that are closer to end customers.

15.1.2 Problem Definition and Performance Measures

OPLOG has allocated a storage area for each customer to store their brand's products. Delivery time depends on where orders are given and from which fulfillment center they are met, in other words, how supplies have been allocated at the beginning of the week. Therefore, stock distribution to fulfillment centers should be planned efficiently. OPLOG does not have a proper decision system to handle the complexity that new fulfillment centers will bring since all of their fulfillment centers were located in the same city. Therefore, a proper decision support system which includes data manipulation, clustering algorithm for demand points, segmentation and forecasting of SKUs and an integer programming model for allocation have been developed to minimize the last-mile delivery times, which is the main performance measure.

15.2 Model Development

15.2.1 Assumptions and Major Constraints

Regarding the scope of the main problem, the assumptions, constraints, objectives and solution approach are developed. OPLOG is planning to open three new warehouses at different locations during 2023 and there will be six fulfillment centers in total. However, our solution is developed for five fulfillment centers because the additional one (of the six) is already allocated to a specific customer. It is assumed that the availability of transportation is valid for all time periods. However, trucks will be sent only at full capacity for inter-transfers between fulfillment centers. In terms of the capacity of fulfillment centers and products, it is measured by their relative volume units. The distance data between the fulfillment centers and the centroid points of end-customers' demand is reached by cities' respective locations. The approximate delivery time is taken from cargo companies.

15.2.2 Conceptual Model

The conceptual model consists of several components.

- To improve the quality of analysis performed on datasets, a comprehensive data cleaning is performed, and the data is deeply analyzed to reach meaningful insights.

- Analysis of the sales by cities indicated that a clustering approach was needed to estimate SKU demands with sufficient data points. Hence, the clustering algorithm is performed to aggregate sales.

- Data manipulation and wrangling steps are performed to eliminate outliers, transform data to the required format and create new features.

- The demand of each Zone-SKU pair is predicted by tailored forecasting algorithms.

- The IP model for allocation is developed to distribute the SKUs to fulfillment centers. The output of the IP model is the weekly allocation amount of SKUs to fulfillment centers and the inter-transfer amount within fulfillment centers.

15.2.3 Data Cleaning

The raw data contains missing values, inconsistencies, and duplicates such as inconsistent and non-standardized city names. To ensure that the data is accurate, complete, and consistent, a detailed data cleaning is performed on R using data.table and lubridate libraries.

15.2.4 Clustering Algorithm

Insufficient sales data for City-SKU pairs hinders accurate demand forecasting, highlighting the need for increased sales in many cities to improve the accuracy of SKU demand forecasts. This leads to clustering cities so that each zone has sufficient sales that enable proper forecasting.

In the clustering, the Weighted K-means algorithm has been used to group cities of Turkey into a specified number of zones, 5. The number of zones is chosen by the elbow method and pareto analysis. The algorithm is fed by the 81x81 distance matrix, and the data frame consists of cities and their total sales for two years. The distances are converged to scaled dimensional space, which are latitude and longitude. Then with three features, the method is performed. The weight of a city is selected as the city's total sales in the last two years. Thus, the zones become more sensitive to regional demand patterns as their centroids are located closer to the high-demand regions, in addition to including cities that are geographically close to each other. Clustering is performed on Python using scikit learn and pandas libraries. In the end, expert opinion is taken into consideration for final adjustments.

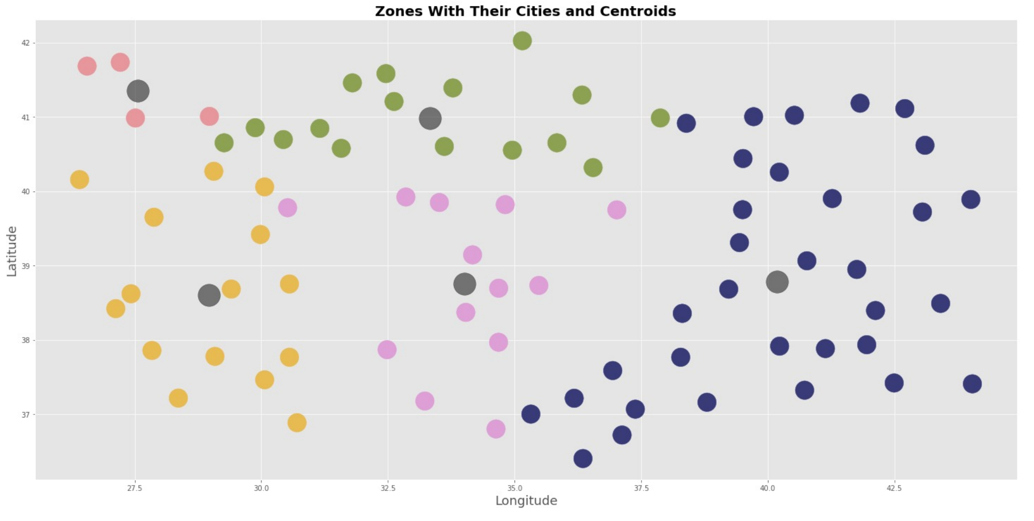


Figure 15.1: Zones with their cities and centroids.

15.2.5 Data Manipulation and Wrangling

In the data wrangling and manipulation step, the outliers are removed based on the Tukey Fence method. Subsequently, sales data is aggregated by week. Then, patterns of Zone-SKU pairs are analyzed to differentiate the sales patterns of products. It is aimed to group products according to their sales patterns since products with different sales patterns and complexities require different tailored forecasting algorithms. A new feature called Sales Frequency Ratio (SFR) is calculated for each Zone-SKU pair, which is defined as the number of weeks a Zone-SKU was sold over a given time period divided by the total number of weeks that the Zone-SKU pair was available for sale during that period. Finally, Zone-SKU pairs are categorized based on their SFR and the number of weeks they were available for sale. This is performed on R using data.table and lubridate packages.

The resulting categorization of the SKUs and respective forecasting methods can be viewed in Figure 15.2.

15.2.6 Forecasting Models

Forecasting has been done weekly in accordance with the replenishment frequency of the fulfillment centers. The focus was on predicting the following week's sales for each Zone-SKU pair. The time series methods' potential to yield accurate predictions in short-term forecasts (Ye and Eskenazi, 2013) benefited since the forecasting period is one week. The general procedure includes the division of sales data into training and validation data sets, the calculation of parameters of forecasting models using the training data, and the calculation of error margins of forecasting models using the validation

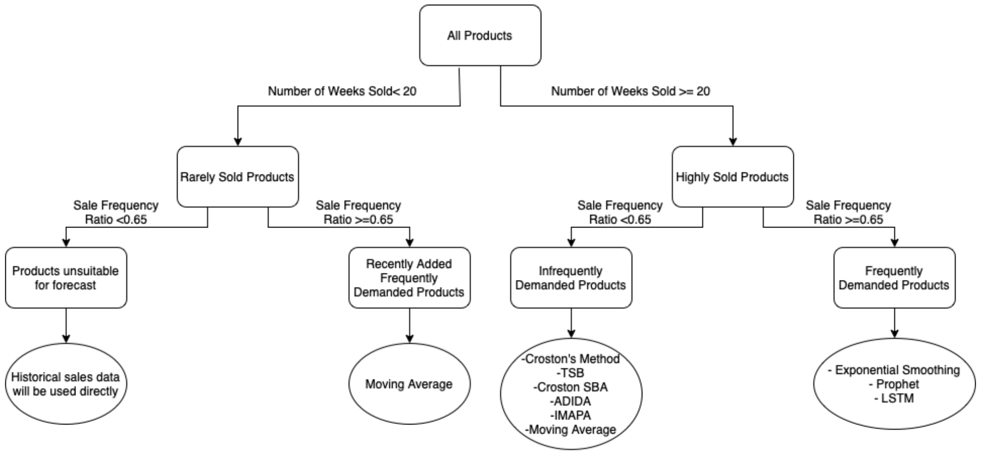


Figure 15.2: Products categories and forecasting methodologies

data. This procedure is followed for each forecasting method determined in the categorization; see Figure 15.2, and the method yielding the lowest MAE value has been chosen automatically. The methods of ES, Prophet, and LSTM are used to forecast the demand of frequently demanded products and MAE values are calculated consecutively for each pair then the prediction which has the smallest MAE have been chosen. An example can be viewed in Table 15.1. Forecasting algorithms have been executed using Python. Libraries of Python corresponding to the methods of different product segments are used in the forecasting algorithms. For infrequently demanded products statsforecast library is used while sklearn, keras libraries are used for frequently demanded products and recently added frequently demanded products. The resulting forecast values are utilized as input in the Integer Programming Model. This model is employed to determine the optimal allocation of replenishment amounts among the five fulfillment centers.

ProductID	Zone	Week	Sales	ES (MAE)	Prophet (MAE)	LSTM (MAE)	Prediction
0012FC**-74..	1	101	9	3.1209	4.5294	3.9578	12
135Y7D**-75 ..	1	101	4	1.8759	1.6792	1.5829	4
B51323**-43...	1	101	7	5.6920	3.4021	4.4892	10
02A2R**-43...	1	101	0	2.6800	2.5870	3.1950	3
004323**-12...	1	101	2	5.7903	4.9572	4.6920	7
...	...	...	...	...	...	...	...
008C98**-C7	1	101	5	1.5892	0.5983	1.2582	6

Table 15.1: frequently demanded products forecasting performances

15.2.7 Integer Programming Model for Allocation

The motivation is to provide a model that OPLOG could run at the start of each week, decide on the allocation amounts, and re-run after the forecasts are updated in a rolling horizon manner. Some decisions, such as consolidation of inter-transfers and replenishment from the inbound node, depend on the initiative of OPLOG. Moreover, some out-of-scope aspects, such as truck capacity planning, are neglected by sending each SKU order separately in the model. Model has been coded and solved using Gurobi Optimization Software. The corresponding model and the definitions of parameters/decision variables can be viewed in Appendix 15.6. The sets are defined as; I:={1,..,5} denotes the set of fulfillment centers, J:={1,..5} denotes the set of zones and K:={1,..,k} denotes the set of SKUs. Model minimizes the last-mile delivery times of the orders with the corresponding constraints as follows;

Constraint (15.2) guarantees that the number of SKUs sent from fulfillment centers to the zones is enough to cover the demand of SKUs. In case it is not, the decision variable $B_{j,k}$ covers it to prevent infeasibility. Its amount is limited by adding a great penalty to the objective function. This amount will be compensated by the customer itself. Therefore, there is no issue related to OPLOG. Constraint (15.3) ensures that the allowed delivery from a fulfillment center can not exceed the total available SKUs which consist of allocated amounts, incoming and outgoing inter-transfers and initial inventory. Constraint (15.4) is the capacity constraint in terms of volume for the fulfillment centers with the balance equations provided in previous constraints. Constraints (15.5) ensure that the inter-transfer amount from a fulfillment center cannot exceed the initial inventory and incoming inter-transfers. Constraint (15.6) ensures that the given supply by a customer is reduced by E_k in a case where the overall total available capacity of the system is less than the provided supply. Constraint (15.7) ensures that provided supplies are accepted into the fulfillment centers unless the capacity restricts it. In case the capacity is exceeded, the model forces excess inventory to be non-zero. Constraint (15.8) determines the number of trucks used in inter-transfer from fulfillment center i1 to i2 $N_{i1,i2}$ and restricts the model to only use trucks at their full capacity. Constraint (15.9) disables the inter-transfers within a fulfillment center itself. The remaining constraints (15.10), (15.11), (15.12), (15.13), (15.14) and (15.15) are integer constraints.

15.3 Validation

Validation of any project remains crucial to observe the proposed solution's impact on the current setup. The main elements of the proposed solution, forecasting, and mathematical integer programming model, have been assessed. Expert opinion has been obtained while conducting the operational validity and real system measurements. Before the validation step began, mutual consulting is conducted between the team and company advisors. Consequently, the way to evaluate the model output with the existing process has been decided as follows and the flow of the algorithm can be viewed in Figure 15.3.

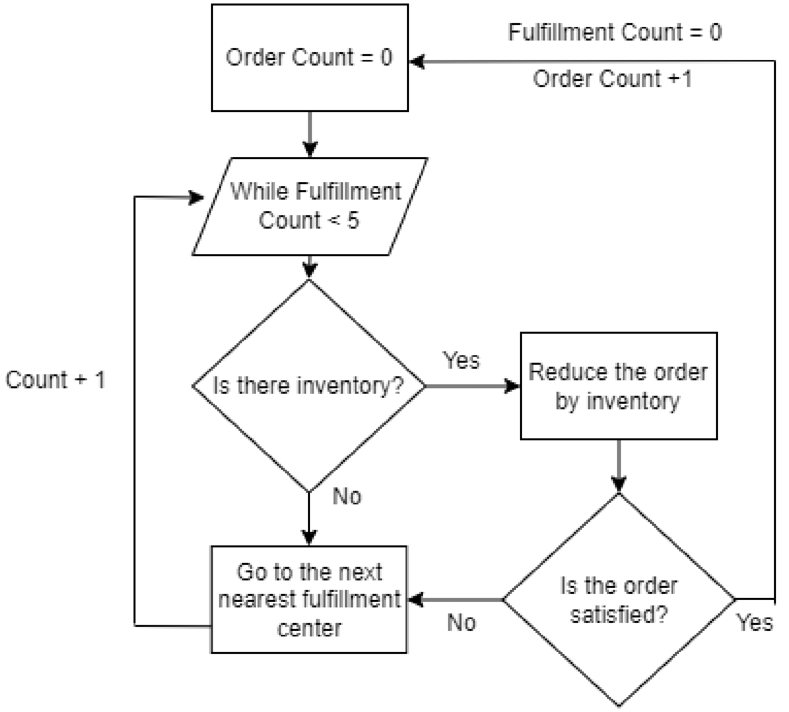


Figure 15.3: Flow chart of validation algorithm.

Distances of each fulfillment center are taken from cargo companies. With those data and model output, validation is simulated through December 2022 using the data from the past 100 weeks. Validation starts with ordering fulfillment centers to the sales point respective to distances. Then it scans the closest fulfillment center to determine whether the model allocated the required amount of that product or not. Then if demand is not fully satisfied, the second nearest fulfillment center would be checked. This process continues until there is no fulfillment center to be checked. If the demand cannot be met, it becomes backorder because of the lack of supply.

The supply of each product is determined based on the number of sales that have occurred to prevent its misleading effects. The respective Python code can be seen in Listing 15.1. Lastly, the runtime for forecasting is approximately one hour, while it is fifteen minutes for the integer programming model. Thus, as its runtime suggests, the overall system has been decided to be operationally valid.

Listing 15.1: Code of Validation Algorithm

```python
cols = ['Closest1', 'Closest2', 'Closest3', 'Closest4']
for i, row in merged_data.iterrows():
    count = 0
    last_index_flag = i== len(merged_data)-1
    last_product_flag = row['ProductId'] == merged_data.loc[i+1- int(last_index_flag), 'ProductId']
    while row['Sales'] > 0:
        if count < 4:
            sales_backup= row['Sales']
            row['Sales'] = max(0,row['Sales'] - merged_data.loc[i, row[cols[count]]])
            merged_data.loc[i+int(last_product_flag)-int(last_index_flag), row[cols[count]]] = max(0,merged_data.loc[i, row[cols[count]]] - sales_backup)
            merged_data.loc[i, 'Sales'] = row['Sales']
            count +=1
        else:
            count = 0
            row['Sales'] = - row['Sales']
            merged_data.loc[i, 'Sales'] = row['Sales']
        continue
```

15.4 Integration and Implementation

One of the standard GUI libraries of Python, Tkinter, has been used for the user-friendly front-end design and coding of the pro-active inventory allocation decision support system (PIADSS) which can be used once a week to decide on the optimal allocation of SKUs to the fulfillment centers

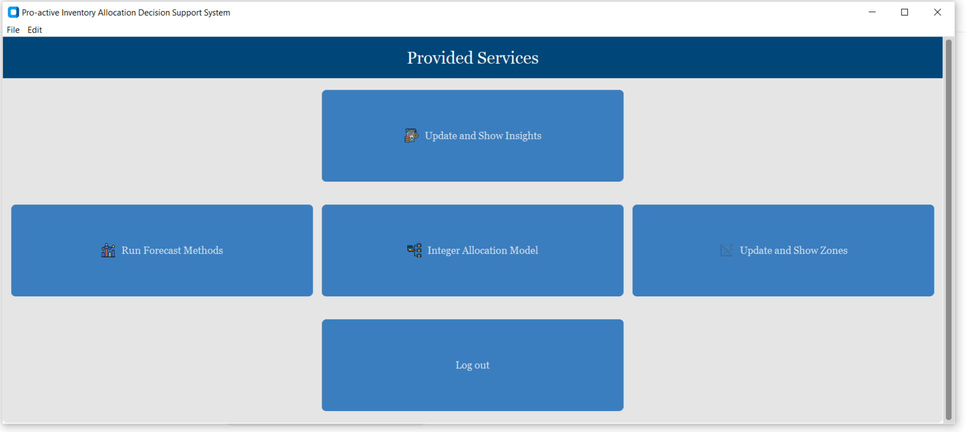


Figure 15.4: Homepage of the PIADSS which displays provided services.

to minimize the last-mile delivery times. PIADSS offers different functionalities such as creating clusters, data insights, forecasting algorithms, and integer mathematical programming that the user can choose from the homepage window. The corresponding windows and the outputs of the PIADSS can be viewed Figures 15.4, 15.5, and 15.6. This system as a whole was developed to enable the user to be more responsive to the customers' replenishment requests, delivery times of orders and to analyze their current occupancy better as a whole system.

The inputs of sales data are taken as Excel files, which is supposed to be done through connected databases when integrated into the current system of OPLOG, outliers are eliminated through specified methods, clustering of cities is performed, different forecasting methods are applied and output

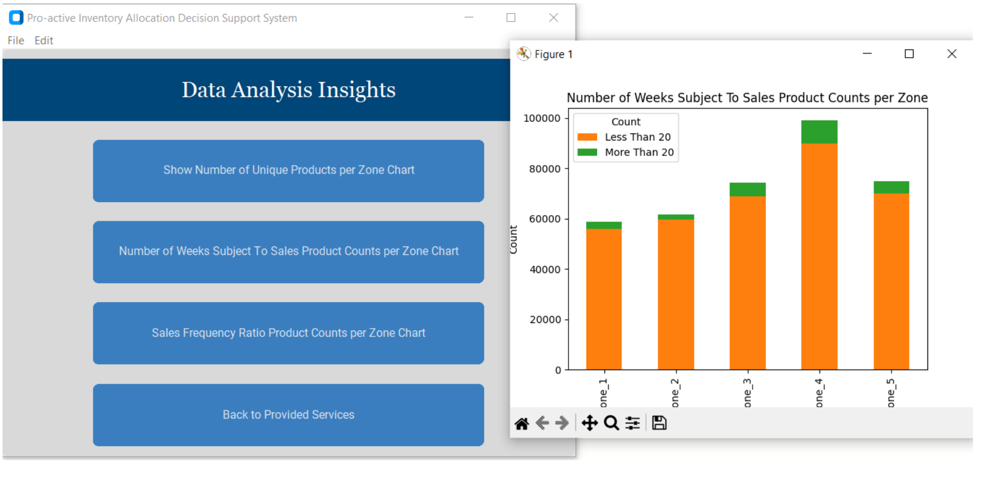


Figure 15.5: Data analysis window of the PIADSS which plots histograms of SFR and number of unique products per zone etc.

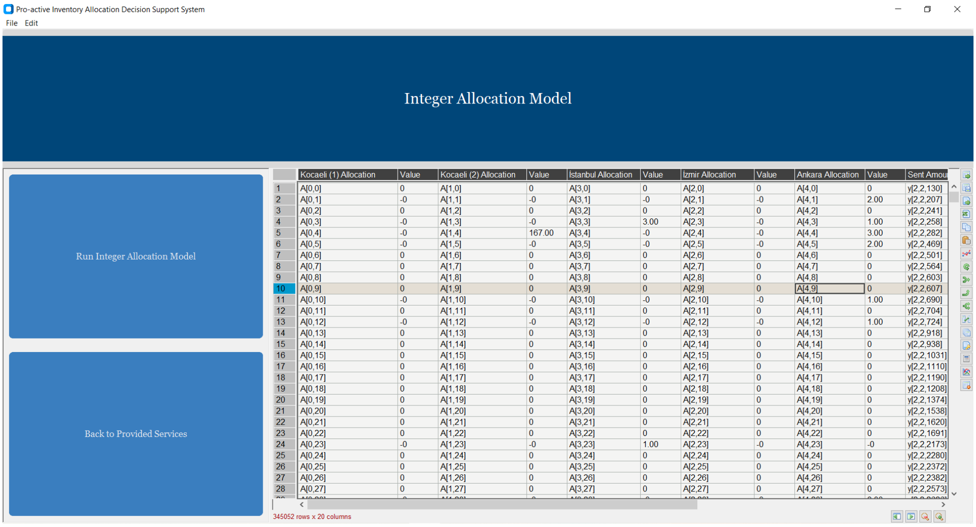


Figure 15.6: Integer programming model for allocation window of the PI-ADSS which runs the Gurobi code and displays the decision variables as a table which can be plotted/downloaded.

forecast values are used in the integer programming model for allocation window respectively. The outputs of each functionality can also be downloaded as Excel files which eases information sharing with other departments as well. Additionally, each module is separately provided to OPLOG to increase the flexibility of integrating the modules into their current software.

15.5 Benchmarking and Benefits to OPLOG

This project reduces the delivery time to end customers and increases customer satisfaction. The created system calculates the sales predictions from the historical data and optimizes the allocation network as a whole by deciding the required SKU placement to the fulfillment centers. In the validation step, past sales data is compared with the proposed system's output. According to these real system measurements, total delivery time is reduced by 12.55% which forms the main KPI of this project. Secondly, the distribution of errors in forecasting is right skewed which implies most of the forecast has either correct or deviate one; see Figure 15.7.

15.6 Conclusions

Our proposed system has successfully reduced total delivery times, allowing OPLOG to meet its customers' orders more quickly. We are proud to report that the system also has a positive impact on the environment by reducing CO_2 emissions. By optimizing the allocation of the replenishments, we were

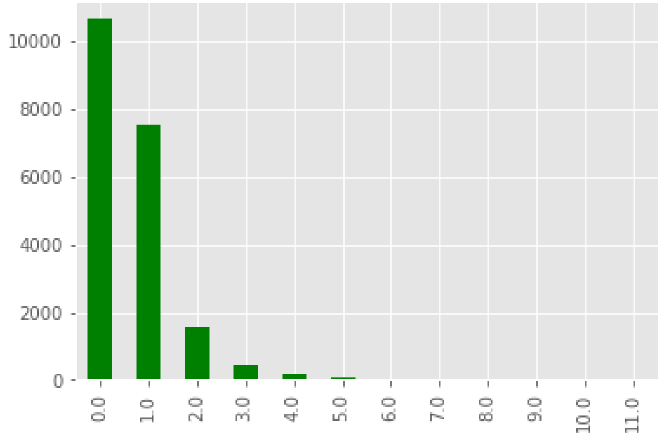


Figure 15.7: MAE distribution

able to reduce the number of trips needed to complete deliveries, which resulted in lower fuel consumption and reduced greenhouse gas emissions. We estimate that this has led to a potential reduction of 397,644 tons of CO_2 emissions per year, contributing to OPLOG's sustainability goals and helping to mitigate the impact of transportation on the environment. By implementing the system, we have demonstrated our commitment to both efficiency and sustainability (IEA-International Energy Agency, 2022).

Bibliography

IEA-International Energy Agency (2022). World Energy Outlook 2022. Retrieved from https://www.iea.org/reports/world-energy-outlook-2022.

Ye, F. and J. Eskenazi (2013). Sales forecast using a hybrid learning method based on stable seasonal pattern and support vector regression. In W. E. Wong and T. Ma (Eds.), *Emerging Technologies for Information Systems, Computing, and Management*, New York, NY, pp. 1251–1259. Springer New York.

Appendix: Allocation Model

Parameters:

$T_{i,j}$**:** The time required for transshipment from warehouse i to zone j, $\forall i \in I, \forall j \in J$.

$D_{j,k}$**:** The forecasted demand amount for SKU k in zone j, $\forall j \in J, \forall k \in K$.

$I_{i,k}^0$**:** The initial inventory amount of SKU k in warehouse i at the beginning of the week, $\forall i \in I, \forall k \in K$.

S_k**:** The amount of SKUs given by customers to OPLOG to distribute among fulfillment centers, $\forall k \in K$.

V_k**:** The volume of the SKU k, $\forall k \in K$

C_i**:** Capacity of the warehouse i, $\forall i \in I$.

F**:** A big number. It is the corresponding penalty for not accepting supply into the network when capacity exists. Therefore, the full network capacity is utilized.

P**:** A big number. It is the corresponding penalty for having unsatisfied demand. It prevents the model from not satisfying the demand in the case of sufficient SKU amount to cover it.

H**:** The volume capacity of trucks that are used in inter-transfer.

Decision Variables:

$y_{i,j,k}$**:** The number of SKU k sent to zone j from warehouse i, $\forall i \in I, \forall j \in J, \forall k \in K$.

E_k**:** The excess supply of SKU k that cannot be accepted due to overall capacity, $\forall k \in K$. It equals to zero if there is enough capacity in fulfillment centers overall Turkey.

$B_{j,k}$**:** The unsatisfied demand in zone j for SKU k, $\forall j \in J, \forall k \in K$. If the total amount of SKU k in the system is enough to cover the demand, it equals to zero.

$A_{i,k}$**:** The allocated SKU k to fulfillment center i, $\forall i \in I, \forall k \in K$. If the sum of the initial inventory and inter-transfer flows is enough to cover the demand, it equals to zero.

$Z_{i_1,i_2,k}$**:** The transferred amount of SKU k from fulfillment center i_1 to fulfillment center i_2 , $\forall i_1, i_2 \in I, \forall k \in K$.

N_{i_1,i_2}**:** The number of trucks used for transfer from fulfillment center i_1 to fulfillment center i_2, $\forall i_1, i_2 \in I, \forall k \in K$.

Model:

$$\min \quad \sum_{i\in I}\sum_{j\in J} T_{i,j}\sum_{k\in K} y_{i,j,k} + P\sum_{j\in J}\sum_{k\in K} B_{j,k} + F\sum_{k\in K} E_k \tag{15.1}$$

s.t.

$$\sum_{i\in I} y_{i,j,k} + B_{j,k} \geq D_{j,k} \quad \forall j\in J, \forall k\in K \tag{15.2}$$

$$\sum_{j\in J} y_{i,j,k} \leq A_{i,k} + \sum_{i_1\in I} Z_{i_1,i,k} - \sum_{i_2\in I} Z_{i,i_2,k} + I^0_{i,k} \quad \forall i\in I, \forall k\in K \tag{15.3}$$

$$\sum_{k\in K} V_k(I^0_{i,k} + A_{i,k} + \sum_{i_1\in I} Z_{i_1,i,k} - \sum_{i_2\in I} Z_{i,i_2,,k}) \leq C_i \quad \forall i\in \ I \tag{15.4}$$

$$I^0_{i,k} + \sum_{i_1\in I} Z_{i_1,i,k} \geq \sum_{i_2\in I} Z_{i,i_2,k} \quad \forall k\in K, \forall i\in I \tag{15.5}$$

$$\sum_{i\in I} A_{i,k} = S_k - E_k \quad \forall k\in K \tag{15.6}$$

$$\sum_{k\in K} V_k E_k \geq \sum_{k\in K} V_k(I^0_{i,k} + S_k) - \sum_{i\in I} C_i \quad \forall i\in I \tag{15.7}$$

$$\sum_{k\in K} V_k Z_{i_1,i_2,k} = H\cdot N_{i_1,i_2} \quad \forall i_1,i_2\in \ I \tag{15.8}$$

$$Z_{i_1,i_2,k} = 0 \quad \forall i_1,i_2\in I, \forall k\in K \tag{15.9}$$

$$y_{i,j,k}\in \mathbb{Z}^{0+} \quad \forall i\in I, \forall j\in J, \forall k\in K \tag{15.10}$$

$$A_{i,k}\in \mathbb{Z}^{0+} \quad \forall i\in I\ k\in K \tag{15.11}$$

$$E_k\in \mathbb{Z}^{0+} \quad \forall k\in K \tag{15.12}$$

$$B_{j,k}\in \mathbb{Z}^{0+} \quad \forall j\in J,\ \forall k\in K \tag{15.13}$$

$$Z_{i_1,i_2,k}\in \mathbb{Z}^{0+} \quad \forall i_1,i_2\in I,\ \forall k\in K \tag{15.14}$$

$$N_{i_1,i_2}\in \mathbb{Z}^{0+} \quad \forall i_1,i_2\in I \tag{15.15}$$

Sürdürülebilir Kalkınma Amaçları için Stratejik Bütçe Tahsisleri | 16

UNDP IICPSD SDG AI Lab

Proje Ekibi
Yağmur Ece Akın, Pelin Barlas, Arda Baş, Güneş Çakmakcı, Umur Arda Çalışkan, Beyza Ulubek, Alper Can Yılmaz

Şirket Danışmanı
Gökhan Dikmener
Teknik Uzman

Akademik Danışman
Doç. Dr. Özlem Karsu
Endüstri Mühendisliği Bölümü

Özet

Hükümetlerin bütçe tahsis kararlarının Sürdürülebilir Kalkınma Amaçları'nı (SKA) nasıl etkilediğini değerlendirmek için bir metodoloji önerilmiştir. SKA sistemi hakkında kısa bir açıklama sağlanmış, mevcut sistemi ve sorun analiz edilmiş, varsayımlar ve kısıtlamalarla birlikte bir çözüm stratejisi geliştirilmiştir. Çalışmamız, bütçe tahsis kararlarının SKA'lar üzerindeki etkisini değerlendirmek için bir rehber olacaktır.

Anahtar Sözcükler: SKA, Bütçe Tahsisi, Etki Analizi, Bütçe Etiketlenmesi.

Strategic Budget Allocations for Sustainable Development Goals

Abstract

A methodology to assess how governments' budget allocation decisions impact Sustainable Development Goals (SDGs) has been proposed. The SDG system is described, the current system and associated challenges are examined, and a solution strategy, taking into consideration assumptions and constraints, is proposed. This work serves as a guide for assessing the impact of budget allocation decisions on SDGs.

Keywords: SDG, Budget Allocation, Impact Analysis, Budget Tagging.

16.1 Description of the System

16.1.1 The UNDP

United Nations Development Programme, UNDP, is a United Nations led organization whose primary goal is to eradicate poverty and reduce inequality on an international scale. The United Nations Special Fund and the United Nations Expanded Programme of Technical Assistance, established in 1949 and 1958 respectively, were combined to form UNDP by the General Assembly in 1966. They are the governors of the Sustainable Development Goals (SDGs). In 2015, the United Nations developed these goals, aiming to accomplish them by 2030 in order to create a more sustainable and improved world for all. To ensure social, economic, and environmental sustainability, actions taken towards these goals must be balanced and not prioritize one over another. Presently, the UNDP has outlined 17 SDGs (UNDP, 2022). The UNDP Nature, Climate, and Energy Cluster, UNDP Finance Sector Hub, and UNDP IICPSD collaborated to establish the SDG AI Lab. The SDG AI Lab provides research and advisory support for optimizing the application of artificial intelligence (AI) and machine learning (ML) in sustainable development (SDG AI Lab, 2022).

16.2 System Analysis and Problem

16.2.1 Current System Analysis

The United Nations adopted the Sustainable Development Goals (SDGs) as a universal call to action, with the primary objective of eradicating poverty, protecting the planet, and ensuring that by 2030, all people enjoy peace and prosperity UNDP (2022). These 17 interrelated global goals emphasize the synergistic environmental, social, and economic dimensions of sustainable development(Schleicher et al., 2018).

In order to quantify the progress of the goals, several targets are defined under each goal. A target can be either of type "outcome" or can be of type "means of implementation" Bartram et al. (2018). The summary of the Goal system can be seen below at Figure 16.1.

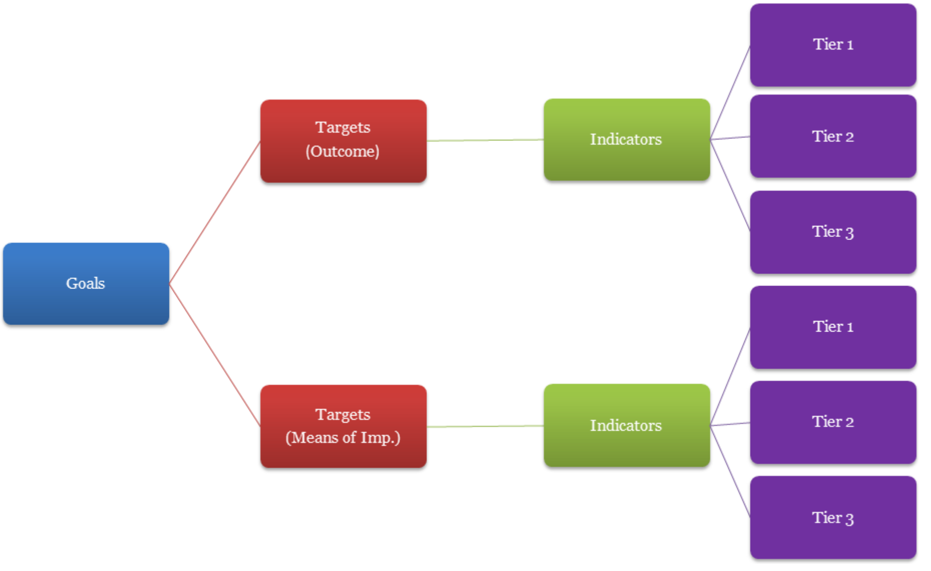


Figure 16.1: Summary of the Goal System

Since the SDGs are considered universal goals, they comprise the governments as well as the private sector and non-governmental organizations. While all of the mentioned parties have a role in the process of achieving the goals, their transactions are generally parallel with governmental activities. Therefore, analyzing government activities is the most important component and a good leader for observing the progress on SDGs.

In order to efficiently achieve their objectives, including those related to SDGs, governments depend on budget allocation. To optimize impact and resource allocation, budget tagging has emerged as a method for tracking the collective efforts of multiple agencies working towards common goals (Okitasari and Kandpal, 2022).

16.2.2 Problem Definition

The prioritization of Sustainable Development Goals (SDGs), their respective targets, and indicators are subject to variation depending on the specific requirements of individual countries. The primary challenge to address involves devising the most efficient solution in terms of cost and time while identifying the necessary resources for implementation. The significant part of the overall analysis of sources of financing for the SDGs is the public budget and the problem for the governments and stakeholders in this situation is to decide where to invest their limited resources (UNDP, 2021). The problem under examination concentrates on public resources and expenditures.

The Scope of the Problem

Within the context of problem definition, the main goal that forms the scope of the problem is the following: While understanding the causal mechanisms between SDG's and budget allocations, analyzing the outcomes of the fiscal years according to budget expenditure of government to prioritized SDG's with respect to country's needs. Additionally, this analysis seeks to determine whether an optimal trajectory can be achieved in subsequent years.

16.3 Solution Strategy

16.3.1 Critical Assumptions

The present study concentrates on Mexico, primarily due to the availability of SDG data related to government investments. It is essential to note that the influence of private and non-governmental entities is not considered in this analysis; the focus remains solely on government expenditure.

16.3.2 Objective

UNDP seeks to link resource allocation decisions with SDG performance by optimizing budget allocations to maximize SDG scores. The goal is to distribute the budget among activities to both spend a meaningful amount on SDGs and deliver effective results measured through scores.

16.3.3 Solution Approach

The objectives specified in part 3.2 are accomplished through the use of a mathematical model. A model featuring a single objective function that maximizes the total score of the SDGs is developed. MIPmodule in Pythonis utilized to implement the solution.

16.3.4 Analysis

Diminishing Returns of SDGs

As investments toward the SDGs increase, there is a decline in the rate of return in terms of scores, unless these investments are strategically targeted or efficiently allocated. The factors contributing to a decrease in utility can be classified under four main categories: limited resources, overlapping objectives, external factors, and delayed effects.

Piecewise Linear Form

For certain goals, it may be relatively easier to improve the scores at lower levels; however, once a saturation point is reached, substantial financial investments may not significantly impact the score. Conversely, for other goals, the behavior can be the opposite. Consequently, we hypothesize that the relationship between expenditure and the impact on SDG scores follows a piecewise linear model. To identify the breakpoints, eight intervals are defined within the score range of 0 to 100. The parameter s_{jk} is employed to utilize the piecewise model, with k representing the score intervals. Due to data limitations, each goal is examined individually, drawing from various articles and reports. An extensive literature review is conducted to determine the concavity level of the corresponding function representing the relationship between expenditure and impact. The distribution of the expenditure growth rates (EGR) of the SDGs is provided in Figure 16.2.

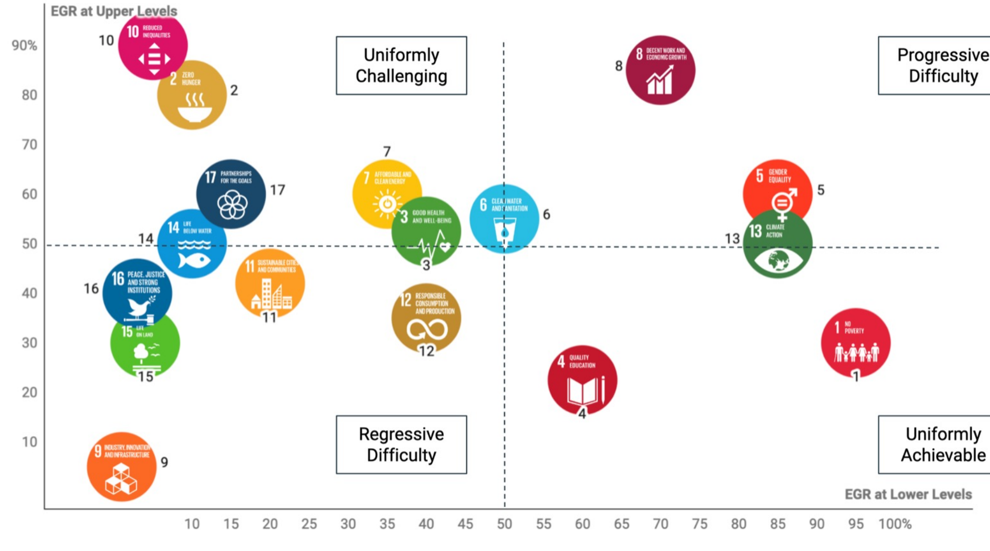


Figure 16.2: Expenditure Growth Rate Distribution of SDGs

Based on the findings, the behavior of each goal at higher and lower scores is ascertained. Consequently, two distinct growth rates are employed for each goal to establish the piecewise linear model. Table 16.1 reports the breakpoint values of SDGs. Figure 16.3 shows the piecewise plot of SDG 1 as an example.

Determination of Weights

A specific proportion of an expenditure allocated to a particular activity will influence one target, while a different proportion will affect another. The document provided by the Ministry of Finance of Mexico indicates the

	0	20	40	60	80	85	90	95	100
	A0	A1	A2	A3	A4	A5	A6	A7	A8
1	0.00	20.09	46.22	80.18	122.62	135.89	152.47	173.20	1000.00
2	0.00	24.75	51.98	81.92	231.67	418.84	1354.72	6034.14	29431.19
3	0.00	55.28	143.74	285.26	639.07	860.21	1413.04	2795.12	6250.32
4	0.00	85.48	205.14	372.67	596.05	670.51	769.79	902.16	1078.65
5	0.00	14.63	31.46	50.81	115.31	169.05	348.22	945.43	2936.14
6	0.00	8.51	21.27	40.42	72.33	85.63	107.79	144.72	206.27
7	0.00	81.05	214.79	435.45	1171.00	1783.95	3827.13	10637.73	33339.73
8	0.00	110.35	253.80	440.30	1683.57	3755.69	17569.86	109664.27	723627.00
9	0.00	50.38	115.87	201.01	307.43	340.68	382.26	434.22	1000.00
10	0.00	17.21	36.99	59.75	287.29	856.16	6544.81	63431.29	632296.10
11	0.00	5.50	15.40	33.22	62.93	75.31	95.93	130.31	1000.00
12	0.00	2.91	6.25	10.10	15.23	16.94	19.22	22.26	1000.00
13	0.00	2.65	5.70	9.21	17.97	23.45	37.15	71.39	156.99
14	0.00	8.84	18.56	29.26	42.63	46.80	52.03	58.55	1000.00
15	0.00	2.42	4.95	7.62	11.42	12.78	14.72	17.50	1000.00
16	0.00	118.92	255.67	412.93	675.04	784.26	966.28	1269.64	1775.25
17	0.00	41.26	86.64	136.56	192.02	207.43	224.55	243.57	1000.00

Table 16.1: Breakpoints

SDGs and targets associated with activity spending. We examined four years of available data independently and computed weights for each target and activity based on their frequency of occurrence. Four Excel spreadsheets were created to display weights for each year, spanning from 2018 to 2021. These weights were then multiplied with budget expenditures to allocate them to specific targets.

Connection between Expenditure and Score

Once the indicator scores for an SDG are computed, their average is calculated, and this value is considered the overall score for that particular SDG. The assumption made by decision-maker is that all indicators have an equal impact on an SDG's score. We associated expenditures with scores by pre-

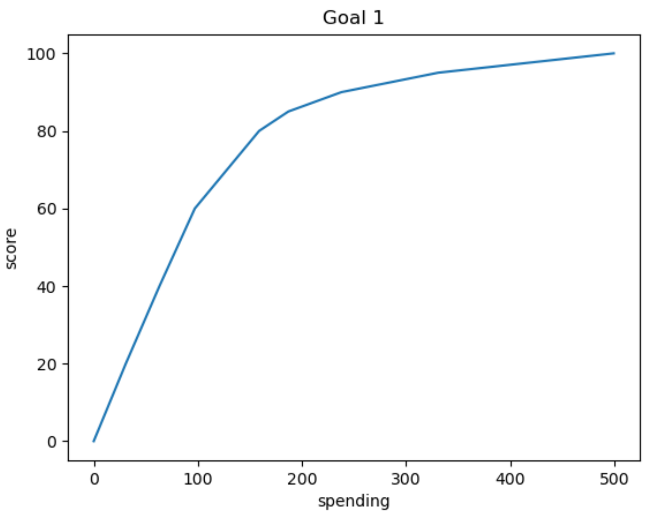


Figure 16.3: SDG 1 Piecewise Expenditure - Score Function

suming that expenditures generate scores without negative or zero values. It was determined that it would be more appropriate to change the level of the parameter from target level to goal level, as this allowed for a more meaningful output with aggregate data.

16.3.5 Model

Parameters

- A : Set of activities
- G : Set of goals
- K : Number of breakpoints in the piecewise linear function of scores versus expenditure
- A_{jk} : Expenditure breakpoints
- s_{jk} : Score created by unit expenditure in goal j in interval k, $\quad \forall j \in G$ $\forall k \in \{1, ..., K\}$
- b : Total budget
- c_i : Cost of activity i, $\quad \forall i \in A$
- w_{ij} Weight of activity i for goal $j \quad \forall i \in A, \quad \forall j \in G, \quad (\sum_{j \in G} w_{ij} = 1, \quad \forall i \in A)$

Decision Variables

- x_j : Amount of expenditure in goal $j \quad \forall j \in G$
- y_i : Percentage of cost met for activity $i \quad \forall i \in A$
- u_j : Score of goal $j \quad \forall j \in G$
- z_{jk} : Weights for break point k to determine the amount spend for each goal $j \quad \forall j \in G, \quad \forall k \in \{1, ..., K\}$
- t_{jk} : 1 if amount spend for goal j is in interval k, 0 else $\quad \forall j \in G,$ $\forall k \in \{1, ..., K-1\}$

Objective Function

$$\max \sum_{j \in G} u_j$$

Constraints

$$\sum_{i \in A} y_i c_i \leq b \tag{16.1}$$

$$x_j = \sum_{i \in A} c_i y_i w_{ij} \qquad \forall j \in G \tag{16.2}$$

$$u_j = \sum_{k \in \{1,...,K\}} A_{jk} s_{jk} z_{jk} \qquad \forall j \in G \tag{16.3}$$

$$x_j = \sum_{k \in \{1,...,K\}} A_{jk} z_{jk} \qquad \forall j \in G \tag{16.4}$$

$$\sum_{k \in \{1,...,K\}} z_{jk} = 1 \qquad \forall j \in G \tag{16.5}$$

$$\sum_{k \in \{1,...,K-1\}} t_{jk} = 1 \qquad \forall j \in G \tag{16.6}$$

$$z_{jk} \leq t_{j,k-1} + t_{j,k} \qquad \forall j \in G \quad \forall k \in \{2, ..., K-1\} \tag{16.7}$$

$$z_{j1} \leq t_{j1} \qquad \forall j \in G \tag{16.8}$$

$$z_{j,K} \leq t_{j,K-1} \qquad \forall j \in G \tag{16.9}$$

$$z_{jk} \geq 0 \qquad \forall j \in G \quad \forall k \in \{1, ..., K\} \tag{16.10}$$

$$t_{jk} \in \{0, 1\} \qquad \forall j \in G \quad \forall k \in \{1, ..., K-1\} \tag{16.11}$$

$$0 \leq y_i \leq 1 \qquad \forall i \in A \tag{16.12}$$

$$0 \leq u_j \leq 100 \qquad \forall j \in G \tag{16.13}$$

Model Explanation

Constraints 1 through 4 represent the following: Constraint 1 ensures that the total money spent on activities does not exceed the overall budget, Constraint 2 computes the total expenditure dedicated to goal j using weights, Constraint 3 calculates the score of a goal by multiplying the score creation coefficient s_{jk} and the expenditure allocated to that goal, and Constraint 4 identifies the interval in which the expenditure for goal j is located. Constraints 5 through 9 state that the amount spent on a goal can only be in one interval. Constraints 10 to 13 indicate the following: Constraint 10 asserts that weights for breakpoints must be greater than or equal to 0, Constraint 11 specifies that the association of an expenditure for a goal with an interval is a binary variable, Constraint 12 dictates that the percentage of cost met for an activity cannot be greater than the cost of that activity or cannot be less than 0, and Constraint 13 states that the score of SDGs cannot surpass 100 or fall below 0.

16.3.6 Verification

Various tests were conducted to assess the accuracy of the constructed model. The first method used was the continuity test, which involves setting the budget equal to the total activity cost to examine the outcome when sufficient funding is available for all activities. As anticipated, this adjustment led to an increase in the overall average SDG score. The budget was then returned to its initial level, and the weight of all activities associated with the 15th SDG was altered to 0 to ensure the proper functioning of the model. After running the model again, both the expenditure and score of the 15th SDG were observed to be 0. The continuity test was further utilized using s_{jk} values. By modifying the piecewise linear function, the score and amount spent on the 7th SDG were increased, leading to an overall average SDG score increase from 86.5 to 88.65, without altering the total amount spent on the SDGs, as the budget remained unchanged. The simplified model was tested by eliminating the budget constraint, resulting in an average overall SDG score of 90.01. The verification of the model was corroborated by the observed changes between input and output variables and the analysis of the simplified model. However, the consistency test and degeneracy test could not be implemented due to limitations in variables. The model's flow was traced, and arbitrary data with extreme values were analyzed to determine the response.

	Score (Real vs Model) with exact spending on each goal (alpha = 1,beta = 1)							
	2018		2019		2020		2021	
	Real	Model	Real	Model	Real	Model	Real	Model
Goal 1	92.5	86.7	92.3	94.1	89.1	88.7	90.3	92.5
Goal 2	61.3	62.5	60.5	61.2	60.5	60.2	60.5	61.2
Goal 3	80.5	73.1	78.7	74.7	78.3	77.9	78.3	80.4
Goal 4	96.4	91.5	96.1	92.7	95.9	95.0	95.9	96.6
Goal 5	77.0	74.9	78.1	75.9	77.9	76.6	77.9	78.6
Goal 6	77.0	72.4	77.3	74.9	77.5	75.0	77.5	81.1
Goal 7	67.9	67.4	68.3	71.5	68.3	64.3	68.3	74.8
Goal 8	65.1	62.1	64.3	63.5	63.4	65.6	63.4	65.0
Goal 9	43.8	52.3	47.1	37.1	49.1	42.8	49.0	67.2
Goal 10	39.2	41.3	39.2	36.8	39.2	39.7	39.2	53.6
Goal 11	82.4	85.7	83.6	64.0	82.4	76.3	81.2	88.1
Goal 12	86.1	82.6	86.0	68.5	86.0	87.7	86.0	87.7
Goal 13	83.5	87.7	83.8	70.2	84.9	89.4	84.9	77.7
Goal 14	66.3	61.4	66.4	62.1	66.4	67.0	66.4	69.5
Goal 15	56.0	68.6	55.9	54.5	55.1	47.4	55.0	61.9
Goal 16	53.6	47.4	53.7	51.0	53.9	59.3	54.4	62.9
Goal 17	64.8	61.4	64.5	63.9	64.5	65.9	64.5	66.0
Average	**70.2**	**69.3**	**70.3**	**65.7**	**70.1**	**69.3**	**70.2**	**74.4**

Table 16.2: Validation results

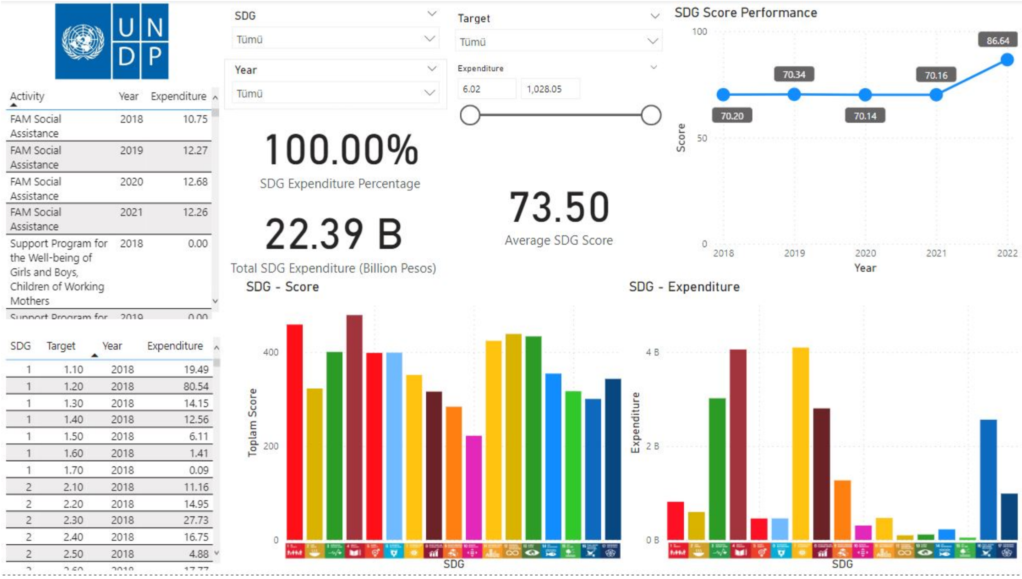


Figure 16.4: Power BI Report Screen

16.3.7 Validation

To validate the model numerically, the initial step involved adjusting the total budget and activity expenditures to correspond with the 2021 figures. This approach allowed for the examination of any significant deviations from reality. Given that the model utilizes four years of historical data to determine an average improvement rate, some discrepancies between the model's findings and the actual 2021 results were expected. The actual average score for 2018-2021 was 70.2, while the model's outcome was 69.675, resulting in a minimal difference of only 0.74%. Table 16.2 summarizes the validation results. Since our system comprises governments and their budget allocations, it is not feasible to establish a pilot study where the proposed methodology can be implemented.

16.4 Outcome and Deliverables

16.4.1 Outcome and Deliverables

The objective of this project is to optimize a country's SDG budget allocation by utilizing a mathematical model to ascertain the capital expenditure required for each related activity to achieve maximum improvement within the available budget. Upon completion of the project, the institution can employ the mathematical model for subsequent years, necessitating only input changes (such as total budget or minimum improvement level for selected SDGs) without needing assistance to modify the model.

	Budget allocation (Real vs model) with exact budget each year (alpha = 2)							
	2018		2019		2020		2021	
	Real	Model	Real	Model	Real	Model	Real	Model
Goal 1	134.4	186.3	179.6	186.3	144.5	186.3	168.3	186.3
Goal 2	100.4	231.7	91.0	231.7	83.1	231.7	90.7	231.7
Goal 3	509.4	314.8	534.5	314.8	582.0	314.8	624.2	612.8
Goal 4	749.1	592.8	776.4	584.9	828.8	553.9	877.0	702.3
Goal 5	71.4	114.1	73.9	114.1	75.6	114.1	80.4	114.1
Goal 6	66.7	159.1	72.1	159.1	72.3	159.1	88.0	159.1
Goal 7	710.1	284.5	823.5	505.2	625.0	505.2	914.3	505.2
Goal 8	488.8	360.7	575.4	360.7	706.0	360.7	672.6	360.7
Goal 9	220.2	417.1	155.7	417.1	179.8	417.1	296.7	417.1
Goal 10	43.8	65.1	38.8	65.1	42.0	124.9	57.8	124.9
Goal 11	92.4	151.2	51.2	151.2	70.1	151.2	103.0	151.2
Goal 12	15.6	25.6	11.1	25.6	21.6	25.6	24.3	25.6
Goal 13	20.7	48.1	10.0	48.1	23.1	48.1	12.6	48.1
Goal 14	30.8	78.9	31.5	78.9	36.7	78.9	47.0	78.9
Goal 15	9.3	17.6	7.0	17.6	6.0	17.6	8.0	17.6
Goal 16	359.1	458.2	387.5	458.2	452.8	572.3	534.4	720.3
Goal 17	139.5	255.9	155.4	255.9	167.7	255.9	190.5	333.9
Total	**3,761.7**	**3,761.7**	**3,974.5**	**3,974.5**	**4,117.3**	**4,117.3**	**4,789.7**	**4,789.7**

Table 16.3: Benchmarking Expenditures

A Power BI report displays budget tagging and spending contributions to overall welfare. The report is dynamic, facilitating detailed and interactive viewing. The template and initial data are supplied, allowing the institution to input new data into Power BI for future years with varying data as they execute the mathematical model. Figure 16.4 displays the Power BI report.

The code is executed in Python for the decision support system. Subsequently, users are prompted to enter the activity cost multiplier and budget multiplier via an input screen. Based on the provided inputs, the corresponding Power BI screen is displayed, offering users tailored data visualizations that cater to their particular requirements.

16.4.2 Benefits to the Institution

The successful completion of this project will benefit UNDP SDG AI LAB and contribute to advancing the global goals of eradicating poverty, protecting the environment, and ensuring peace and prosperity for all by 2030. Even minor progress in the project's humanitarian objectives will enhance the institutions' reputation.

UNDP SDG AI LAB employs AI and ML techniques for problem-solving; however, external factors can limit their effectiveness. Operations research and industrial engineering principles are applied to offer a new perspective and address the interlinked indicators that necessitate flexible budget allocation, which is currently lacking in AI and ML approaches.

	Scores (Real vs Model) with exact budget each year (alpha = 2)							
	2018		2019		2020		2021	
	Real	Model	Real	Model	Real	Model	Real	Model
Goal 1	92.5	95.0	92.3	95.0	89.1	95.0	90.3	95.0
Goal 2	61.3	80.0	60.5	80.0	60.5	80.0	60.5	80.0
Goal 3	80.5	60.0	78.7	60.0	78.3	60.0	78.3	80.0
Goal 4	96.4	82.8	96.1	82.2	95.9	80.0	95.9	89.4
Goal 5	77.0	85.0	78.1	85.0	77.9	85.0	77.9	85.0
Goal 6	77.0	90.0	77.3	90.0	77.5	90.0	77.5	90.0
Goal 7	67.9	40.0	68.3	60.0	68.3	60.0	68.3	60.0
Goal 8	65.1	60.0	64.3	60.0	63.4	60.0	63.4	60.0
Goal 9	43.8	95.0	47.1	95.0	49.1	95.0	49.0	95.0
Goal 10	39.2	60.0	39.2	60.0	39.2	65.3	39.2	65.3
Goal 11	82.4	95.0	83.6	95.0	82.4	95.0	81.2	95.0
Goal 12	86.1	95.0	86.0	95.0	86.0	95.0	86.0	95.0
Goal 13	83.5	96.8	83.8	96.8	84.9	96.8	84.9	96.8
Goal 14	66.3	89.1	66.4	89.1	66.4	89.1	66.4	89.1
Goal 15	56.0	95.0	55.9	95.0	55.1	95.0	55.0	95.0
Goal 16	53.6	60.0	53.7	60.0	53.9	68.7	54.4	80.0
Goal 17	64.8	80.0	64.5	80.0	64.5	80.0	64.5	85.0
Average	**70.2**	**79.9**	**70.3**	**81.1**	**70.1**	**81.8**	**70.2**	**84.4**

Table 16.4: Benchmarking Scores

16.4.3 Benchmarking

To assess the benefits of our model compared to the current system, we utilized the exact 2018-2021 budget and allowed the model to determine expenditures for activities. The actual average score of 2018-2021 was 70.2. Using the same budget, the model yielded a score of 81.8, resulting in an improvement of 16.5%. The expenditure and score values, along with the cost and budget data for 2018-2021 are presented Tables 16.3 and 16.4.

Bibliography

Bartram, J., C. Brocklehurst, and D. Bradley (2018). Policy review of the means of implementation targets and indicators for the sustainable development goal for water and sanitation. *NPJ Clean Water 1*(3), 43–47.

Okitasari, M. and R. Kandpal (2022). Budgeting for the SDGs: Lessons from the 2021 Voluntary National Reviews. *United Nations University Institute for the Advanced Study of Sustainability 32*, 1–4.

Schleicher, J., M. Schaafsma, and B. Vira (2018). Will the Sustainable Development Goals address the links between poverty and the natural environment? *Current Opinion in Environmental Sustainability 34*(1), 43–47.

SDG AI Lab (2022). SDG AI Lab. https://sdgailab.org/ [Online; accessed October 22, 2022].

UNDP (2021). SDG Alignment and Budget Tagging: Towards an SDG Taxonomy - Analysis for Columbia.

UNDP (2022). Sustainable development goals: United Nations Development Programme. https://www.undp.org/sustainable-development-goal [Online; accessed October 22, 2022].

17 Üretim Planlaması, Çizelgeleme ve Süreç Eniyileme

Meteksan Savunma

Proje Ekibi
M. Uygar Altınok, Elif Cengiz, Emrecan Dağdelen,
Ahu Erdoğan, Umut Tepe, Ali Türe, H. Ege Yücel

Şirket Danışmanı
Okan Akay
Üretim Planlama Müdürü

Akademik Danışman
Doç. Dr. Özlem Çavuş İyigün
Endüstri Mühendisliği Bölümü

Özet

Meteksan Savunma, atölye tipi ve süreç bazlı üretim yapan bir savunma sanayii şirketidir. Şirketin ürünlerinin birçok alt bileşenden oluşması, üretim sırasında ürünlerin arasında öncelik ilişkisi olması, ve üretimin ortak kullanılan makinelerde gerçekleştirilmesi üretim planlanmasını oldukça zorlaştırıyor. Bizler, Meteksan Savunma'nın üretim planlarını optimize etmek için gelişmiş matematiksel algoritmalar kullanan bir üretim çizelgeleme karar destek sistemi geliştirdik. Maksimum makine kullanımı ve siparişlerde gecikmeme prensibini uygulayan bu yazılım, şirketin yıllık, aylık ve günlük üretim planları geliştirmesini, makine ve insan gücü planlaması yapmasını ve siparişler hakkında kararların alınmasını sağlıyor.

Anahtar Sözcükler: Çizelgeleme, süreç analizi, üretim planlama, karar destek sistemi

Production Planning, Scheduling, and Process Improvement

Abstract

Meteksan Defense is a defense industry company making workshop-type and process-based production. Since the company's products consist of many sub-components, there is a priority relationship between the products, and the production is carried out on shared machines makes production planning difficult. We have developed a production scheduling decision support system using advanced mathematical algorithms to optimize Meteksan Defense's production plans. This software, which applies the principle of maximum machine utilization and no delay in orders, enables the company to develop annual, monthly, and daily production plans, make informed machine and manpower planning for production, and make quick decisions about orders.

Keywords: Scheduling, process analysis, production planning, decision support system

17.1 Company and Problem Definition

17.1.1 Company Description

Meteksan Defense is a leading defense industry company based in Ankara, Turkey, with a reputation for excellence in advanced technology systems and solutions. Established in 2006, the company has been providing innovative defense solutions for more than three decades and has become a key player in the global defense industry.

Meteksan Defense specializes in the development, design, production, and integration of state-of-the-art systems, including radar and electronic warfare systems, communication systems, and command and control systems. The company's core capabilities also include electro-optical systems, avionics, and simulators.

17.1.2 Problem Definition

Meteksan Defence is faced with a complex challenge in its production processes, as its products are comprised of multiple subcomponents that require prioritized production sequences and are produced using shared machinery. The relationship between the subcomponents creates a hierarchy of priorities that must be taken into account during the production process. The use of shared machinery further complicates the planning process. The availability of machinery must be considered when scheduling the production of each subcomponent, as it can affect the timing of subsequent production processes.

Under these circumstances, Meteksan Defense faces significant challenges in production planning. To plan scheduling, Meteksan obtains MRP from SAP, and then prepares schedules manually based on the MRP given by SAP. Since the scheduling is done manually despite the support of MRP, it is done based on experience rather than a mathematical model. This results in manual planning processes done by production planning engineers, which are prone to error and require significant time and effort. Our analysis indicates that the absence of detailed scheduling software leads to a reduction of capacity utilization rates between 5% and 20%.

Furthermore, the absence of accurate production capacity estimations creates uncertainties in order management, which poses challenges in decision-making regarding order acceptance and rejection. Delaying delivery deadlines can result in significant financial costs in the defense industry. Meteksan Defense's lack of production analytics, particularly with worker and equipment management further complicates the challenges faced by the company. The absence of data-driven decision-making processes hinders the company's ability to optimize production and make informed investment decisions, leading to suboptimal outcomes.

Throughout the project, we focused on key performance metrics that would help address the challenges faced by Meteksan Defense in production planning. Our primary goal was to increase capacity utilization rates, improve tracking of worker and equipment usage, enable detailed scheduling, and identify production capacity across different production scenarios. We developed a platform that allowed the company to monitor these metrics and gain insight into the production process. This platform enabled more informed decision-making related to operational and strategic matters. By utilizing the detailed scheduling program, we were able to address the challenges of the complex production process, enabling the company to optimize production planning and better allocate resources. The platform also provided accurate estimations of production capacity, which improved order management and enabled more informed order acceptance or rejection decisions. Finally, the production analytics provided by the platform enabled data-driven decision-making processes, leading to improved investment decisions and more efficient production.

17.2 Solution Approach

During the mathematical development of our model, we conducted extensive research to identify existing solutions for similar problems in the literature (Bitran and Tirupati, 1993; Erfanian and Pirayesh, 2016; Kaya and Fığlalı, 2018; Sung and Kim, 2008). However, due to the high complexity of the problem and the unique challenges faced by Meteksan Defense, we were unable to find a pre-existing model that could be adapted to address their specific needs. Consequently, we developed a model from scratch, without relying on any external sources for inspiration. Several assumptions were made in building the model for Meteksan Defense's production process. i) All required supplies for production will be supplied and ready at the scheduled time. ii) New workers will not be late in completing work within the required labor time. iii) Setup times will have a fixed duration and will not change based on the previous product. iv) Defectives are assumed to be low and ignored. Since the run time of the solution concept is short, in case of any defective parts one can simply re-run the tool to obtain a new schedule. We have come up with a deterministic mixed integer program model to achieve these objectives according to assumptions.

We focused on three critical aspects. Firstly, our solution concept must be capable of producing output with daily, weekly, and monthly plans that display the production line's current status, including which products are being manufactured, the machines in use, and the corresponding time intervals. This output must be easily observable and will assist Meteksan's production planners in shaping the production plan. Secondly, we ensured that the solution concept produces outputs within an acceptable time frame, making them immediately usable by the Meteksan team. Finally, we designed the solution concept to allow for scenario-based planning by enabling inputs to be adjusted according to the factory's current state. Since we realized that the company needed a decision support system that could accommodate variable production plans and adapt to changing conditions.

Given the complexity of Meteksan's production system and the wide range of machinery and products involved, we recognized that a highly powerful computing platform was required. Therefore, we opted to use Gurobi as the optimization platform for our model. While our initial mathematical model was successful in solving the problem conceptually for small data sets, its implementation on real-world data sets posed a challenge due to very high run times. This hindered our second goal of providing results within an acceptable timeframe. To overcome this issue, we developed a heuristic approach that enabled our model to break down the problem into smaller components and solve it faster. Specifically, our initial model evaluated the production across all departments simultaneously, whereas our heuristic approach now prioritizes each department individually. It first generates an optimal schedule for the first department using its data, and then uses this schedule as input for the second department, continuing this process for all five departments in the system. As a result, our run times have significantly decreased to acceptable levels, making our model practical for use by the company. In addition, we divided the production data into two main parts: the main products and their components and sub-components. The main products are made up of several components and components are made up of several sub-components, we have released the data on how much of the main products are wanted to be produced in which month, to be entered as input. We formulated our solution concept to accurately identify the components and sub-components of each main product, and determine their priority relationship. Our solution methodology was designed as a two-layer structure, where the first layer determines the sub-components that need to be produced first based on the desired quantity of the main products, as components cannot be produced without the relevant sub-components. The second layer then schedules the components according to the precedence relations, building on top of the sub-component scheduling. This effectively resolved the complexity arising from precedence relations, which makes manual planning challenging.

In consideration of Meteksan's priority of scenario-based planning as the third critical aspect, we focused on developing a solution method that provides flexibility. Specifically, our approach aimed to demonstrate the different scheduling options that arise when workers work overtime versus during regular working hours, while also allowing for optional planning. By incorporating these elements, our solution method enables Meteksan to explore different scheduling scenarios and make informed decisions based on their unique needs and preferences. The mathematical model of the heuristic are given in the appendix.

17.3 Validation

Thus, we have instead employed a few different approaches for validation following Landry et al. (1983) closely.

17.3.1 Conceptual Validation

We went back and forth with the IA to ensure that the heuristic approach currently utilized was not missing a crucial aspect of the real life system and the needs of the IA. All of the assumptions have been made together with the IA and so going over them to ensure they were not a vital part of the system has been relatively smooth: We first create a final schedule for the sub-components and then take that output as input for another schedule, i.e. another iteration of the solution method, where the components are scheduled. Thus, the relationship between each component and its subcomponent, if it has one, is protected so that a component can only be produced after all of its sub-components have been produced. It had also been assumed previously that changes to capacity can be made in a

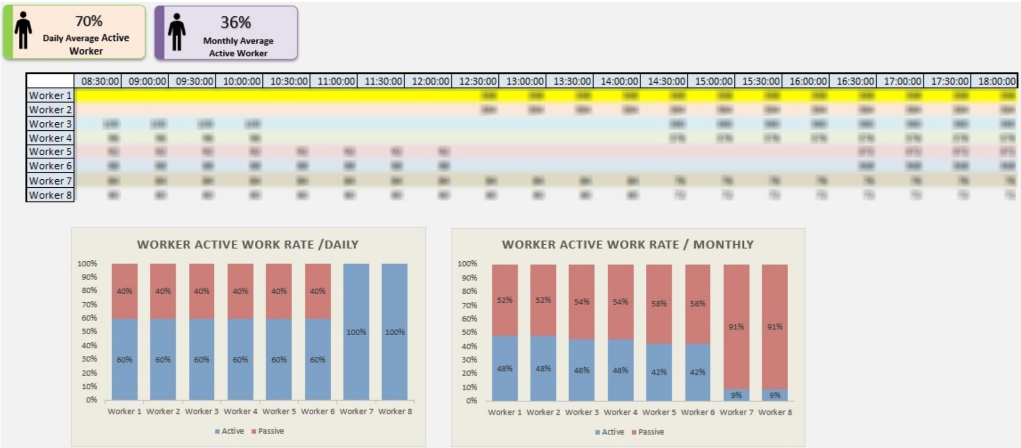


Figure 17.1: Gantt Chart

day to be closer to optimality. However, while subcontracting employees is possible, they can only be subcontracted in intervals of a month. Thus, the solution approach has been changed so that capacity changes can only be made monthly.

17.3.2 Operational Validation

A benefit of the tool we have developed is that it can be utilized as many times as necessary. Utilizing the model on its own meant a run-time of 10 to 12 hours for a small data set which was not viable. On the other hand, the heuristic developed has a run-time of on average 15 to 20 minutes to output a schedule per month. Thus, the tool can be utilized whenever there are new orders incoming to offer deadlines more accurately to customers or see if the current schedule can accommodate any new orders.

The inputs needed by the solution approach are the process times for each component and sub-component to be produced in each department. These have had already been obtained by Meteksan Savunma for at least 5 different times each component has been produced. Thus, it is reasonable to assume the input data is correct and is available. Another input is capacity which Meteksan Savunma wanted control over and can be put in to accurately reflect real life.

17.3.3 Historical Results

We also compared the schedule outputted by the solution concept with the schedule the production planners at Meteksan Savunma had created for the month of December, 2022. A snippet of the optimal schedule for one of the departments and its utilization rates is displayed in Figure 17.1 along with the schedule Figure 17.2 Meteksan Savunma had planned for the same

Proje	Birim	Gün	Adet	Notlar
A	[illegible]	8	10	Cumartesi Dahil
	[illegible]	14	10	Cumartesi Dahil
	[illegible]	14	10	Cumartesi Dahil
	[illegible]	14	10	Cumartesi Dahil
	[illegible]	3	10	Haftaiçi
	[illegible]	3	10	Haftaiçi
	[illegible]	3	10	Haftaiçi
C	[illegible]	27	100	Cumartesi Dahil
	[illegible]	22	120	Cumartesi Dahil
	[illegible]	27	100	Cumartesi Dahil
	[illegible]	10	142	Cumartesi Dahil
B	[illegible]	15	39	Cumartesi Dahil
	[illegible]	15	39	Cumartesi Dahil
	[illegible]	27	70	Cumartesi Dahil
	[illegible]	10	78	Cumartesi Dahil

Figure 17.2: Production Schedule Planned by Meteksan Savunma

products. The optimal schedule takes 200 hours utilizing the same overtime information that had been planned for the month of December which the planners had anticipated to take 245.5 hours. The tool also considers how many components had some subcomponents already in the inventory or not which the planners had also kept in mind. The optimal schedule has been face-validated by the IA.

17.3.4 Pilot Study and Comparison

A pilot study has been conducted which made it possible compare results of the solution approach with the results observed in the system. This will be further explained in the following section.

17.4 Integration and Implementation

As part of a pilot study to test the system in real life conditions, Defense has provided the main products to be produced for a certain week. This input was then scheduled using the decision support tool by also considering the capacities available for that week. The optimal schedule works on a basis of one piece flow logic; Meteksan Defense also wanted to observe the system output for different scenarios meaning different batch sizes since overtime information was already known for that particular week. Thus, the team outputted the optimal schedule along with different batching strategies for the first two departments in production of the products to be produced that week. This is given in Figure 17.3. For this particular week, Meteksan Savunma chose to produce as in Scenario 5 given in Figure 17.3.

This resulted in 32 hours of real-time production for the first department and our solution tool scheduled the first department's production for 32 hours of active production as well. The optimal schedule for the first

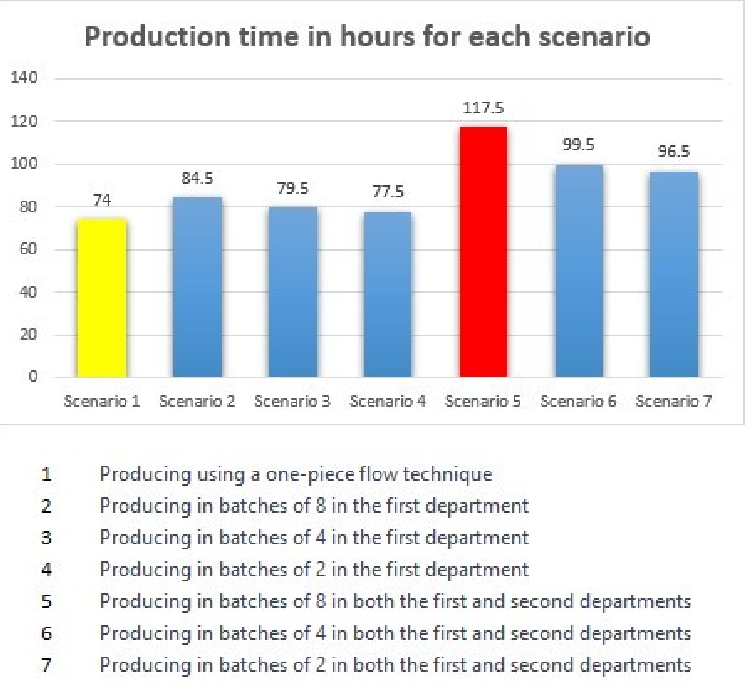


Figure 17.3: The pilot study preparations

department is 29.5 hours in comparison. These results can be also seen in Figure 17.4. Hence, we've confirmed the validity of the solution concept but also proven that manufacturing without batches is more efficient. Meteksan Defense plans to utilize the decision support system to plan their more standardized products per month and make use of the system to plan for possible capacity expansions in the future by inputting their future orders.

Figure 17.4: The comparison of chosen scenario and the optimal.

17.5 Benchmarking and Benefits

It was our intention to gauge the manufacturing plan to Meteksan Defense's own experience. The major objective was to provide a schedule that was more efficient than the current one in terms of capacity utilization rate and quantity of products produced. Increasing the efficiency of the departments and capacity utilizations have been one of the most important points in the scheduling process. In this context, while the scheduling made for the same projects is done in 27 days, that is, within 245.5 working hours in the company, the same scheduling is done within 200 working hours thanks to the software developed by the team. It can be said that approximately 20 percent of progress has been made in terms of time. Moreover, depending on the complexity of the problem, the decision-support tool created by the team takes 15-30 minutes. Compared to the two work days they use to make such scheduling, the tool created by the team takes just 2.7% as long as the time required by the planning department to prepare for the following month using the current method and prepares a monthly schedule.

17.6 Deliverables and Conclusion

In conclusion, the challenges faced by Meteksan Defense in production planning were significant, with multiple subcomponents that required prioritized production sequences and shared machinery. The absence of detailed scheduling software led to reduced capacity utilization rates, uncertainties in order management, and suboptimal decision-making related to operational and strategic matters.

The solution we provided in the form of a platform allowed the company to monitor key performance metrics, enable detailed scheduling, improve tracking of worker and equipment usage, and identify production capacity across different production scenarios. By utilizing this platform, Meteksan Defense was able to optimize production planning, better allocate resources, make more informed order acceptance or rejection decisions.

Overall, the platform provided data-driven decision-making processes, leading to increased efficiency and improved outcomes. In addition, an user interface was made to show that the model works better. The user interface is depicted by Figure 17.5.

Bibliography

Bitran, G. R. and D. Tirupati (1993). Chapter 10 Hierarchical production planning. In *Logistics of Production and Inventory*, pp. 523–568. Elsevier.

Erfanian, M. and M. Pirayesh (2016). Integration aggregate production

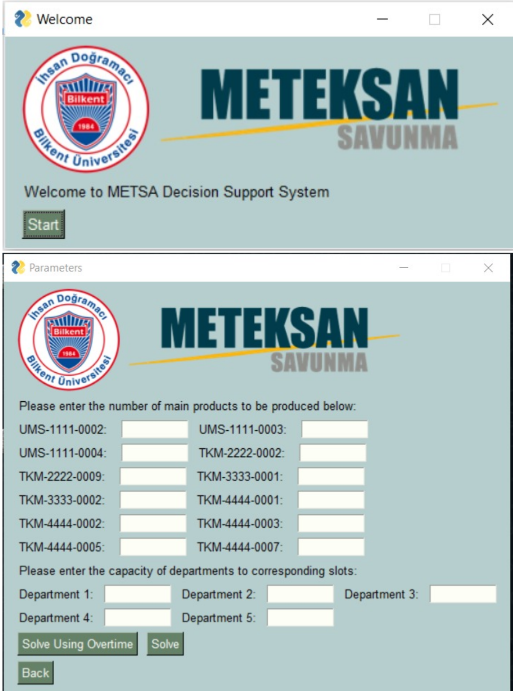


Figure 17.5: User Interface

planning and maintenance using mixed integer linear programming. In *2016 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. IEEE.

Kaya, S. and N. Fığlalı (2018). Çok Amaçlı Esnek Atölye Tipi Çizelgeleme Problemlerinin Çözümünde Meta Sezgisel Yöntemlerin Kullanımı. *Harran University Journal of Engineering 3*(3), 222–233.

Landry, M., J.-L. Malouin, and M. Oral (1983). Model validation in operations research. *European Journal of Operational Research 14*(3), 207–220. Methodology, Risk and Personnel.

Sung, C. S. and H. A. Kim (2008). A two-stage multiple-machine assembly scheduling problem for minimizing sum of completion times. *Int. J. Prod. Econ. 113*(2), 1038–1048.

Appendix: Model

Index Sets

J = Set of components

K = A singular department

T = Set of regular time periods given in 30 minute increments

$T^{'}$ = Set of overtime periods given in 30 minute increments

N = Set of natural numbers

Parameters

P_{jk} = Process time of component j at department k, $j \in J, k \in K$

G_k = Minimum number of workers required at department $k \in K$

L_k = Maximum number of workers allowed at department $k \in K$

D_j = Deadline of component $j \in J$

$cost_{kt}$ = Cost of operation at department k during period t, $k \in K, t \in T \cup T^{'}$

$Ocost_{kt}$ = Overtime cost in period t in department $k \in K$, $t \in T^{'}$

Decision Variables

C_{kt} = The number of regular workers at department k during period t,$k \in K, t \in T$

X_{jkt} = Binary decision variable which is 1 if component j starts to be processed at department k at period t, $j \in J, k \in K, t \in T$

Z_{jkt} = Binary decision variable which is 1 if component j is being processed at department k at period t, $j \in J, k \in K, t \in T$

$O_{kt'}$ = Number of workers making overtime at department k for period $t^{'}$, $k \in K, t^{'} \in T^{'}$

T_{jk} = The period component j starts being processed at department k , $j \in J, k \in K$

Objective

$\min \sum_{k \in K} \sum_{tinT \cup T'} cost_{kt}(C_{kt} + O_{kt})$

Constraints

$T_{jk} + P_{jk} \leq T_{j(k+1)}, j \in J, k \in K$

$T_{jk_{last}} + P_{jk_{last}} \leq D_j, j \in J$

$T_{jk} = \sum_{t \in T} tX_{jkt}, j \in J, k \in K$

$\sum_{tinT} X_{jkt} = 1, j \in J, k \in K$

$X_{jkt} \leq Z_{jkt_p}, j \in J, k \in K, t \in \{1, \ldots, t_{last} + 1 - P_{jk}\}, t_p \in \{t, \ldots, t + 1 - P_{jk}\}$

$\sum_{j \in J} Z_{jkt} \leq O_{kt}, k \in K, t \in T^{'}$

$C_{kt} = C_{k(t+1)}, k \in K, t \in \{20(n-1)+1, \ldots, 20(n-1)+15\}, n \in N$

$C_{kt} \geq G_k, k \in K, t \in T$

$C_{kt} \leq L_k, k \in K, t \in T$

$C_{kt} \in N, k \in K, t \in T$

$T_{jk} \in N, j \in J, k \in K$

$X_{jkt}, Z_{jkt} \in \{0, 1\}, j \in J, k \in K, t \in T$

$\sum_{t \in T \cup T'} Z_{jkt} = P_{jk}, j \in J, k \in K$

$O_t \geq O_{t+1}, t \in \{20n + 17, \ldots, 20n + 19\}, n \in N$

$O_t \leq C_{k(t-1)}, k \in K, t = 20n + 17, n \in N$

$\sum_{j \in J} Z_{jkt} \leq C_{kt}, k \in K, t \in T$

$\sum_{k \in K} Z_{jkt} \leq 1, j \in J, t \in T$

$O_{kt} \leq L_k, k \in K, t \in T^{'}$

18 | Üretim Tesisleri Arası Tedarik Zinciri Tasarımı

Norm Fasteners

Proje Ekibi
Hazal Albayrak, Beyza Alkış, Berfin Aş, Hilal Aydın,
İpek Bayram, Mustafa Mert Eskicioğlu, Umut Gubari

Şirket Danışmanı
Fatma Akdemir
Tedarik Zinciri Metot Yöneticisi

Akademik Danışman
Doç. Dr. Özlem Çavuş İyigün
Endüstri Mühendisliği Bölümü

Özet

Bu proje sayesinde makine kapasite ve kullanılabilirliğine dair veriler kullanılarak Norm Fasteners'ın İzmir ve Salihli'deki tesisleri arasındaki ara taşımaları enazlayacak ürün rotalarının bulunmasına yönelik bir yaklaşım sunulması hedeflenmektedir. Şirketin aylık ürün rotalarını belirlerken kullanması için değişen araç sayısına adapte olabilen bir arayüz tasarlanmıştır. Aylık ürün rotaları, kullanılan araç tipi ve sayısı obur algoritma kullanarak belirlenmiştir ve bu sayede %29,9 iyileştirme sağlanmıştır.

Anahtar Sözcükler: Lojistik, süreç analizi, performans iyileştirme, nakliye optimizasyonu, güzergah belirleme, obur algoritma.

Supply Chain Design Between Production Facilities

Abstract

The aim of this project is to present an approach to find product routes that will minimize intermediate transportation between Norm Fasteners' facilities in İzmir and Salihli by using machine capacity and alternative machines data. An interface that can adapt to the changing number of vehicles has been designed for the company to use when determining monthly product routes. Monthly product routes, vehicle types, and the number of vehicles used were determined by using a greedy algorithm, resulting in 29,9% improvement.

Keywords: Logistics,process analysis, performance improvement, transportation optimization, routing, greedy algorithm.

18.1 Company Information

Norm Fasteners has factories located in Izmir and Salihli. The factory started with bolt production in 1973, then continued with the incorporation of many companies over time. As of today, Norm Group continues its activities in 12 different production facilities, 9 logistics centers, 4 R&D design centers, and with more than 3,500 employees and 20 companies in total, 4 of which are abroad. In 2 of these facilities, which have a total annual production capacity of 180,000 tons, more than 16,000 different products are produced for more than 300 customers. All of the company's production facilities are in Turkey, with logistics centers and resident engineers located in different parts of the world. Also, Norm Fasteners operates in national and international markets, mainly in the automotive, electronics, technology, furniture, construction and machinery sectors. The company received approximately 82% of its annual turnover in 2021 from Norm Fasteners which has a 42% market share in the automotive sector.

18.2 System and Problem Description

18.2.1 Current System Analysis

Fastener production occurs in two different locations, Izmir and Salihli which include 7 production facilities in total. The production process, which starts with engineering and R&D, continues with surface treatment and annealing, cold forming, heat treatment, secondary processes, coating, locking, and control steps, and ends with the packaging step.

In the current system, the company does not use a supply chain design for routing between facilities. The company uses a monthly production planning schedule for the manufacturing of the products. Currently, there are 7 different integrated production facilities and more than 16,000 different products. Every facility has a production schedule but not every machine can process every product. Not every facility has the same machines, meaning that facilities are not identical. Also, machines in the production plants have a limited capacity. The machine capacities cause instability in production planning. Since products require different types of machines for different processes, such products require transportation routes between facilities. There are three types of vehicles which are trailers, small trucks, and big trucks. Trailers are used for transportation between facilities in Salihli, small trucks are used for transportation between facilities in Izmir, and big trucks are used for intercity transportation.

18.2.2 Problem Definition and Its Scope

Transportation is a necessity due to machines required for finishing a product can be located in different facilities. This causes the starting and ending facility of a product to be different, which increases the cost of transportation. According to the data, 21% of products are transported between facilities and this leads to the company's problem of excess transportation costs. After the data analysis, it was realized that while some products can be completed in the same facility or region where the production started, there may be some products unnecessarily transported to different facilities or regions due to machine capacities and suboptimal assignments.

The scope of the project is to analyze the route of products between machines according to the processes of these products while optimizing the monthly transportation routes based on the monthly demand.

18.3 Solution Approach

As the first step of our solution strategy, we characterized the problem by making some assumptions about the system due to the complexity and uniqueness of the project. The following key assumptions are made.

- The model is evaluated on a monthly basis. Therefore, monthly demand is considered.
- Production capacities of machines are considered on a monthly basis.
- Kilogram unit is used for products and machine capacities.
- A process does not extend to the next month, it starts and finishes in the same month.

- The fuel consumption per km of all vehicles is assumed to be equal.
- The place of machines cannot be changed.
- The loading and unloading times of the products are ignored.
- A big truck performs a maximum of 4 trips per day between Izmir and Salihli.

The parameters utilized in the mathematical model such as the required processing time for each machine and machine capabilities were generated from the data provided by the company. We also directly used the demand, distance, and vehicle capacity data.

While developing the mathematical model, we considered some major constraints. These were:

- Each machine has production capacities for each product and process.
- Not every product can be produced on every machine.

To solve the mathematical model, we coded it via Python Gurobi Solver. However, due to the large number of product groups produced in the company's facilities and parameters with big data, we were not able to obtain the optimal solution for the real-sized data. Therefore, to prevent the model from becoming unsolvable, research has been carried out for heuristic approaches. As it was mentioned before, there are products that can be produced in a single facility or region, meaning that all the processes that this product goes through can be performed in the same facility or region. Therefore, there is unnecessary transportation that can be eliminated through the optimization of the system. For this reason and some other benefits that will be discussed in the related section, a greedy algorithm was applied to minimize transportation. We implemented the algorithm in Python and it was observed that computation time was reduced drastically.

18.3.1 Mathematical Model

To optimize all the transportation between 7 facilities, mathematical model in Figure 18.1 has been proposed. All the processes of each product in the optimum routes starting from the first production facility to the last facility where they are packaged are carried out. The objective is to minimize the total transportation distance. By achieving this, it is aimed to reduce transportation costs.

Sets:

A = The set of all vehicles

A_1 = The set of all vehicles in Izmir

A_2 = The set of all vehicles in Salihli

A_3 = The set of all vehicles transporting between Izmir and Salihli

I = The set of all facilities

I_1 = The set of facilities in Izmir region

I_2 = The set of facilities in Salihli region

K = The set of products

M = The set of machines

P = The set of processes

S_k = The set of process sequences of products

V = The set of vehicle types

Parameters:

d_{ij} = Distance between facility i and j, $\quad i,j \in I$

dem_k = The demand of product k on that month, $\quad k \in K$

$$mac_{imp} = \begin{cases} 1 & \text{If the process } p \text{ can be done in machine } m \text{ at facility } i \\ 0 & \text{otherwise} \end{cases}$$

$i \in I, \ m \in M, \ p \in P$

r_{ikmp} = The amount of process time in minutes for product k for process p in machine m, in facility i, $\ i \in I, \ k \in K, \ m \in M, \ p \in P$

$vCap_v$ = The capacity of in kilograms for vehicle in type v, $\quad v \in V$

Decision Variables:

X_{ijav} = The number of travels by vehicle a in type v from facility i to j in a month, $\ i,j \in I, \ a \in A, \ v \in V$

Y_{ijkmp} = The amount of product k produced for process p in machine m at facility i to be transported to facility j, $\ i,j \in I, \ k \in K, \ m \in M, \ p \in P$

Model:

$$\min \sum_{i=1}^{I}\sum_{j=1}^{I}\sum_{a=1}^{A}\sum_{v=1}^{V}(X_{ijav} * d_{ij}) \tag{1}$$

subject to

$$\sum_{k=1}^{K}\sum_{j=1}^{I\setminus\{8,9\}}(Y_{ijkmp} * r_{ikmp}/100000) \leq 43200 * mac_{imp} \qquad \forall i \in I \setminus \{8,9\}, \ \forall m \in M, \forall p \in S_k \tag{2}$$

$$\sum_{k=1}^{K}\sum_{j=1}^{I\setminus\{8,9\}}\sum_{p=1}^{S_k}(Y_{ijkmp} * r_{ikmp}/100000) \leq 43200 \qquad \forall i \in I \setminus \{8,9\}, \ m \in M \tag{3}$$

$$\sum_{i=1}^{I\setminus\{8,9\}}\sum_{j=1}^{I\setminus\{8,9\}}\sum_{m=1}^{M} Y_{ijkmp} = dem_k \qquad \forall p \in S_k, \ \forall k \in K \tag{4}$$

$$\sum_{i=1}^{I\setminus\{8,9\}}\sum_{m=1}^{M} Y_{ijkmp} = \sum_{l=1}^{I\setminus\{8,9\}}\sum_{m'=1}^{M} Y_{jlkm'(p+1)} \qquad \forall j \in I \setminus \{8,9\}, \ \forall k \in K, \ \forall p \in S_k \setminus \{last\ element\ of\ S_k\} \tag{5}$$

$$\sum_{k=1}^{K}\sum_{m=1}^{M}\sum_{p=1}^{P} Y_{ijkmp} \leq \sum_{a=1}^{A_1} X_{ija1} * vCap_1 \qquad \forall i,j \in I_1, \ i \neq j \tag{6}$$

$$\sum_{k=1}^{K}\sum_{m=1}^{M}\sum_{p=1}^{P} Y_{ijkmp} \leq \sum_{a=1}^{A_2} X_{ija2} * vCap_2 \qquad \forall i,j \in I_2, \ i \neq j \tag{7}$$

$$\sum_{a=1}^{A_3} X_{89a3} * vCap_3 \geq \sum_{k=1}^{K}\sum_{m=1}^{M}\sum_{p=1}^{P} Y_{89kmp} \tag{8}$$

$$\sum_{a=1}^{A_3} X_{98a3} * vCap_3 \geq \sum_{k=1}^{K}\sum_{m=1}^{M}\sum_{p=1}^{P} Y_{98kmp} \tag{9}$$

$$X_{89a3} + X_{98a3} \leq 120 \qquad \forall a \in A_3 \tag{10}$$

$$\sum_{i=1}^{I_1}\sum_{j=5}^{I_2} Y_{ijkmp} = Y_{89kmp} \qquad \forall k \in K, \forall m \in M, \forall p \in P \tag{11}$$

$$\sum_{i=5}^{I_2}\sum_{j=1}^{I_1} Y_{ijkmp} = Y_{98kmp} \qquad \forall k \in K, \forall m \in M, \forall p \in P \tag{12}$$

$$X_{ijav} \geq 0 \qquad \forall i,j \in I, \ \forall a \in A, \forall v \in V \tag{13}$$

$$Y_{ijkmp} \geq 0 \qquad \forall i,j \in I, \ \forall k \in K, \ \forall m \in M, \ \forall p \in P \tag{14}$$

Figure 18.1: Mathematical model for the problem

18.3.2 Greedy Algorithm

A heuristic approach for quicker run time is needed since the project is evaluated on a monthly basis but may need changes for daily output performance. The mathematical model itself was not capable of providing the outputs needed for the amount of product produced in a month. To speed up the solution process and produce satisfactory results, we needed to develop heuristic methods. Among all available heuristics in literature, a greedy algorithm by Qu and Bard (2013) was the most suitable.

The greedy algorithm firstly assigns the first process of each product to the nearest available machines. Then, for the subsequent processes, it sends the products to machines capable of the next process for each product in the same facilities. If no machines are found, then it sends the product to the nearest facility which includes a machine that can perform that process.

18.4 Verification and Validation

18.4.1 Verification

The verification of the mathematical model was checked by the optimal solutions that were obtained from Python Gurobi Solver. Different combinations of parameters were tested to observe the change in the optimal solution. The results of the model show that the optimal value and decision variables are relevant to the entered parameters. Extreme situations were considered to check if the model is working correctly. If the demand for a product is greater than the capacity of the assigned vehicle, the model distributes the amount of the products into separate vehicles. Additionally, four possible cases for checking the transportation of products were observed in the model according to the sequences of the products. These cases were verified from the output of the relevant decision variables. The model travels 627.5 km whereas the algorithm travels 671.5 km for 35 products. The model gives a better result by 6.55 percent.

18.4.2 Validation

While doing the modeling, the operational frequency of the model and whether the model provides the outputs daily are considered. We make sure that the model works with the data we created and the partial data that we obtained from the company. For more comprehensive validation, different data sets are created by our team such as distance parameters, process times, vehicle capacities, and a binary parameter in which whether or not machines can execute a certain process is checked. The Python code of the model is created by using Gurobi Solver and the code was run by increasing the number of products. Due to the size of the data, the transportation routes for at most 35 products can be determined using the mathematical model. Therefore, to compare the results of the mathematical model and the algorithm, the algorithm code was run by using the same set of data of these 35 products.

In the validation phase, the monthly distance traveled by the company's vehicles for October 2022 was analyzed. It has been calculated that a total of 2551.1 km has been covered for three different vehicle types. Considering the distance covered by the algorithm with this approach, the algorithm has calculated that 2178.6 km must be traveled in the same month for the same amount of demand. When these calculated values and the result of the algorithm are examined, it is seen that there is an improvement of 14.6 percent on the basis of the total distance traveled in October 2022.

18.5 Deliverables

The user interface was developed using the Python programming language, where PyQt5 library is used, with the greedy algorithm integrated into it. The modules and classes being imported are all part of the PyQt5 library, which is a set of Python bindings for the Qt application framework. The reason for choosing Python to develop the user interface was mainly due to the fact that it is an open-source programming language, which means that the company will not have any problems while running the code (Fitzpatrick, 2020).

We provided a user-friendly tool for the company to decide on monthly production routes. This interface is to serve as a decision support tool for the company to compare the results of the algorithm and their current system and then decide on the route for each product, the number of vehicles used in the run period, and how much of which product is produced on which machine. The user interface consists of three important consecutive steps; accepting inputs from Excel which are the parameters of the algorithm provided by the company, executing the heuristic by Python, and displaying the outputs to the user as in Figure 18.2. The route and transported amount of every product can be seen in Figure 18.3.

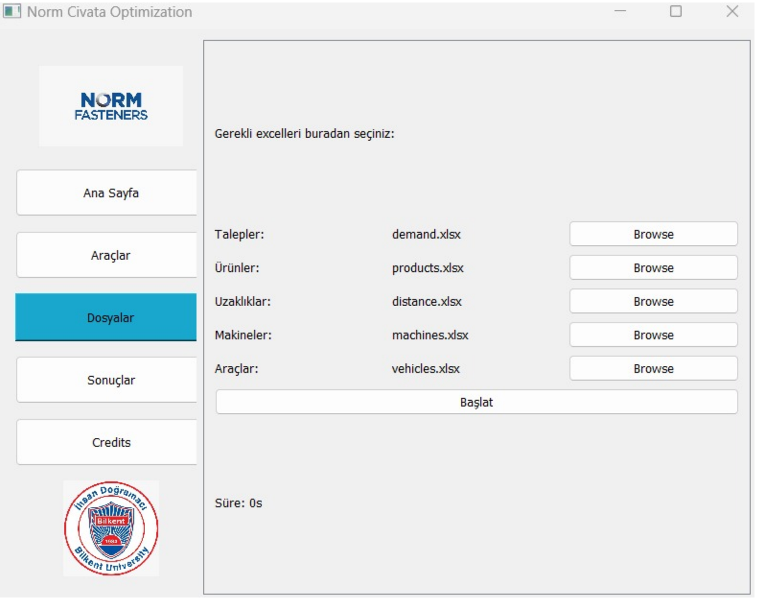


Figure 18.2: User interface imported files

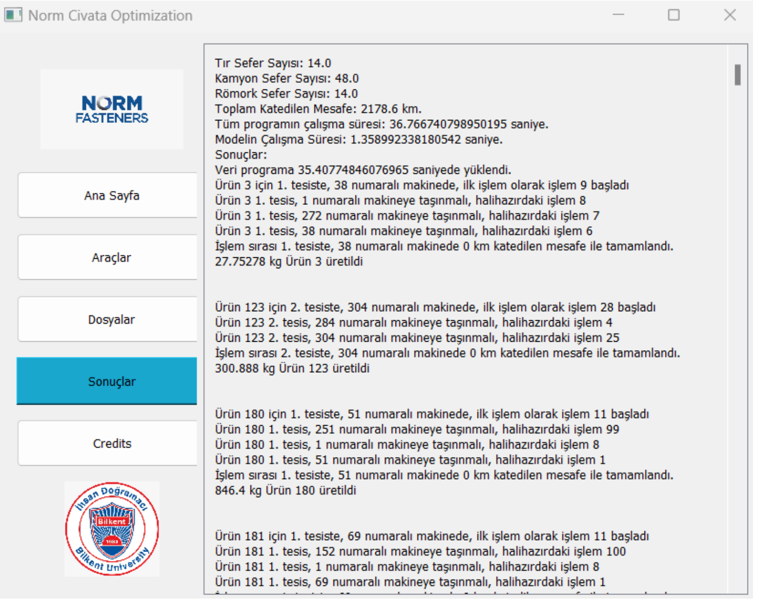


Figure 18.3: User interface results

18.6 Benchmarking and Benefits

Expected benefits to the company are decreasing the amount of transportation between facilities, reducing the company's expenditures related to transportation, preventing the company from time loss, and determination of the vehicle routes according to the plan. Furthermore, the production location and transportation plan deviate monthly. Therefore, our solution approach provides a dynamic model that can adapt to changing scenarios.

In this context, when the 12-month time period between October 2022 and September 2023 is examined, it is calculated that an improvement of 29.9% has been achieved on a monthly average in the total distance traveled by vehicles. A comparison of distance traveled in kilometers for each month can be seen in Figure 18.4 and improvement rates in Figure 18.5.

18.7 Conclusions

In the current system of Norm Fasteners, a monthly production planning schedule is used. The company requires monthly transportation routes according to the demand of the products. The proposed algorithm decides in which facilities the products in the company's supply chain will be produced. Thus, it minimizes the distance traveled by the vehicles used to transport the products between the facilities. Using this algorithm, the company can save on transportation costs by creating a more efficient production route for its products. This algorithm can be strategically used in the product

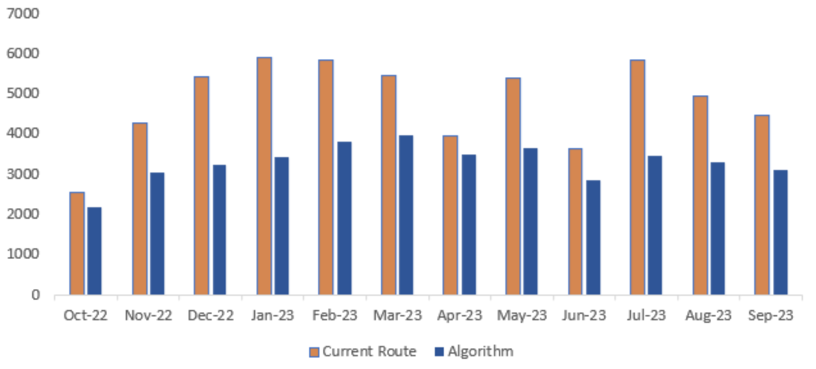


Figure 18.4: Monthly Distance Comparison

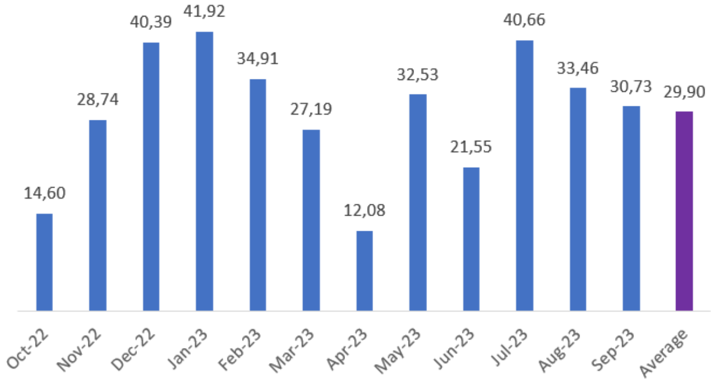


Figure 18.5: Monthly Improvement Rate Comparison

routing mechanism of the company's multi-product planning.

Bibliography

Fitzpatrick, M. (2020). *Create GUI Applications with Python & Qt5: The hands-on guide to making apps with Python*. Martin Fitzpatrick.

Qu, Y. and J. F. Bard (2013). The heterogeneous pickup and delivery problem with configurable vehicle capacity. *Transportation Research Part C: Emerging Technologies 32*, 1–20.

Veri İzleme ve Analizi ile Kaliteyle İlgili Sorunların Erken Tespiti | 19

Arçelik Elektronik İşletmesi

Proje Ekibi

Cemil Ateş, Efe Umut Ateş, Gökçe Cenik, Çağrı Delibaş,
Doruk Barkın Durak, Zeynep Şenkal, Zekiye Tütüncü

Şirket Danışmanı
Burak Beşer
Üretim Bölümü Mühendisi

Akademik Danışman
Dr. Öğr. Üyesi Emre Nadar
Endüstri Mühendisliği Bölümü

Özet

Arçelik, üretimde ciddi kayıplara yol açmadan önce ürünlerdeki kusurları ve türlerini tespit etmek için toplanan test verilerini kullanan bir online karar destek sistemine ihtiyaç duymaktadır. Bu sistem aynı zamanda Arçelik'in tespit edilen ürün kusurların kök nedenlerini teşhis etmesine de yardımcı olacaktır. Hataların erken tespiti ve önlenmesi, hem kalite hem de üretimde etkinliği artıracaktır.

Anahtar Sözcükler: kalite kontrol ve geliştirme, uç değer, kalite testi

Early Detection of Quality-Related Issues with Data Monitoring and Analysis

Abstract

Arçelik needs an online decision support system that uses the test data collected in order to detect the defects and their types before they cause any serious loss in production. This system should also help Arçelik diagnose the root causes of the defects detected. Early detection and prevention of the defects will improve effectiveness in both quality and production.

Keywords: quality control and improvement, outlier, quality testing

19.1 General Information

Founded in 1955, Arçelik became a leader in Turkey's white goods sector. Even though the company is famous in the white goods industry, it also offers products ranging from air conditioning, TV, cash registers, electronic boards, B2B products, and cell phones to medical equipment. With 28 plants and 42.000 employees, the company provides products and services to 146 countries. Arçelik has 12 brands and had a $6.95 billion consolidated net sales in 2021. Some of the interests and technologies of the company include big data and image processing, IoT, machine learning, gaming, and Android TVs.

19.1.1 Arçelik's Existing System

The main product of Arçelik's Çerkezköy plant is television. Existing system works with the Chain Rule shown in Figure 19.1.

19.1.2 Problem Definition

Each defect or problem has several possible root causes. For instance, a shade problem may be due to cable attachment or main board. By trial and error, the repairman tries to find the exact root cause of the problem. However, the set of root causes of a problem is not static. The fact that technology develops every day causes the production process of television to change. New and unprecedented root causes may emerge up the next day as a consequence. The repairman records these defects in the system. The manual recording of defects forms the data set. However, there is a large amount of data; therefore, it is difficult for Arçelik to prioritize one data over the other. Since the data is too large, some problems go unnoticed and cause them to pile up to a point where it causes a decrease in production output. That is why Arçelik must prioritize the correct data to make an early detection of these problems.

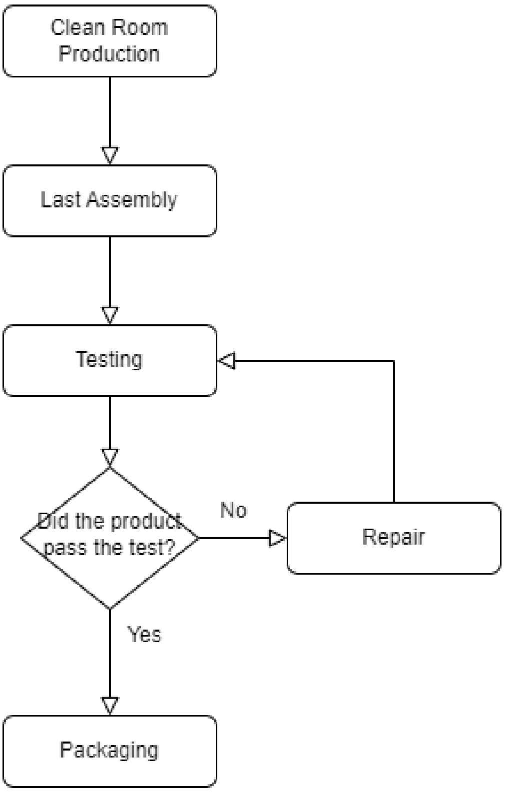


Figure 19.1: Flow chart of the existing system

Arçelik's current warning system reports defects and undesired situations by e-mail. The system notifies Arçelik when five consecutive defects are observed or 16 defective products are detected within an hour. The report gives a defect percentage which has an upper limit for defect rate but it is not integrated into the system. The system delivers reports momentarily. However, Arçelik would prefer to have daily, three month-, and six month reports to make comparisons. It also would have been better if the system could receive data from repair and production and connect them.

19.2 Proposed Solution Strategy

19.2.1 Critical Assumptions

The possibility of a product having more than one defect has been disregarded. This implies equality of defective number and defect number. Significantly small weight of production line 5 is disregarded in the confusion matrix.(The confusion matrix gives a weight for each production line where weights represent defective fraction). Data set consists of combined data (real and generated). Company and project group agreed upon that proposed system is applicable with the given data set.

19.2.2 Major Constraints

The system is built upon 2021 data set. The company collects data on a daily basis therefore, the sample size is chosen daily as the data values are the most consistent in that time interval. There were seven defect types: Panel, Mainboard, Electrical Parts, Moduls, Lips PSU Board, Process and Mechanical. However, after validation (with 2022 data), two more defect types are recognized: Assembly Process and Safety Incidence.

19.2.3 Objectives

The objective is to present two alternative prediction methods to Arçelik: prediction of the fractions of the defective items coming from each production line and the defect type fraction over the total number of defects detected. Implementation of both alternatives will select the better-performing one in terms of accuracy. This system also presents the utilization of statistical control charts which will address the shortcomings of the company's current system regarding data monitoring and reporting. Since control charts contain visual and numerical data representations, they will improve the company's reporting system and also, the productivity will increase through early detection.

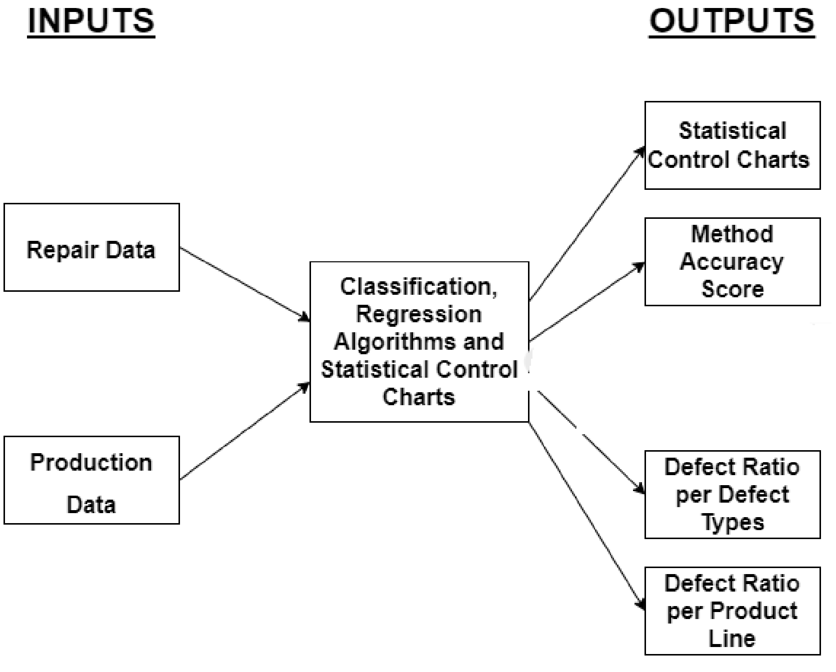


Figure 19.2: Proposed System Illustration

19.2.4 Solution Approach: Statistical Control Charts

Figure 19.2 shows that production and repair data the company provided are inputs for creating statistical control charts: p-charts and u-charts. The p-chart shows the total number of defectives over sample size. The u-charts visualizes the investigation the defect ratios per defect type.

P-chart. The p-chart is created for the total defects per total sample (Montgomery, 1997). The generated p chart for the last 30 days of the data can be seen in Figure 19.3. According to the chart 18th and 25th days have outliers. In other words, the defect-to-production ratio exceeds the upper control limit (Gardiner and Montgomery, 1987). Note that the UCL and LCL of the charts are not fixed due to the variations in the sample size. The zone rules are as follows:

- **Rule 1**: A value being outside the 3σ
- **Rule 2**: 8 or more values on either part of the centerline not coinciding
- **Rule 3**: 4 out of 5 values taking place in 2-sigma zone.
- **Rule 4**: At least 6 values consecutively rising or declining
- **Rule 5**: 2 out of 3 values within 3-sigma zone
- **Rule 6**: 14 consecutive data points should not zigzag

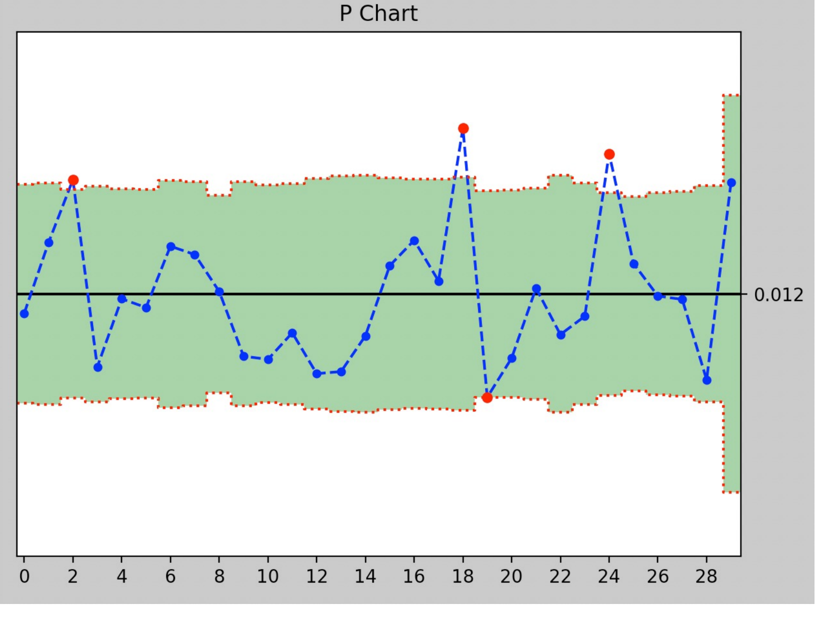


Figure 19.3: p chart

- **Rule 7**: Existence of observable trend or cycle

U-chart. The most common defect type that is panel defect's u chart is presented in Figure 19.4: On the 2nd and 24th days, there are two outliers,

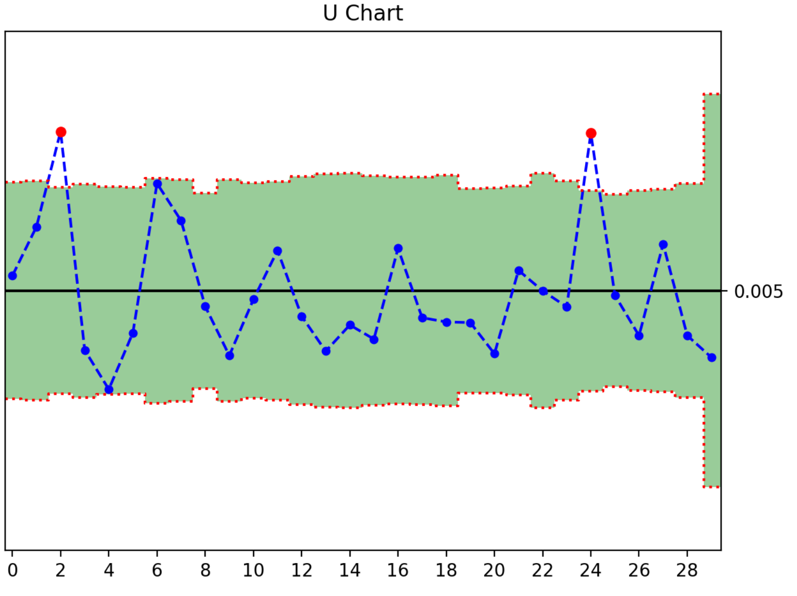


Figure 19.4: u-chart of panel defects

indicating some events have occurred on these days to make the system out of control. The U-charts of 9 defect types are created with 2022 data.

The programming language used was Python. Statistical control charts, u-chart and p-chart, were used. The calculations of p and u-charts are performed with Python and cross-checked via Excel.

Savic (2006) states that the p-chart ensures outlier detection and monitoring of the fraction of non-confirming units in sample size. U chart tracks the number of defects per unit.

19.2.5 Validation

The system is built upon the 2021 data set. The proposed system is used to create a user interface(UI) and UI uses 2022 data for validation.

The current system of the company will give alert when 16 repairs are recorded. The validation will be done by comparing the times the current system of the company gave alarms and the times the presented system gave alarms. The comparisons will be made for P-charts of 15, 30 and 60 work days (standardized). And type 1 error (where the system should have given an alarm but did not) and type 2 error (where did system should not

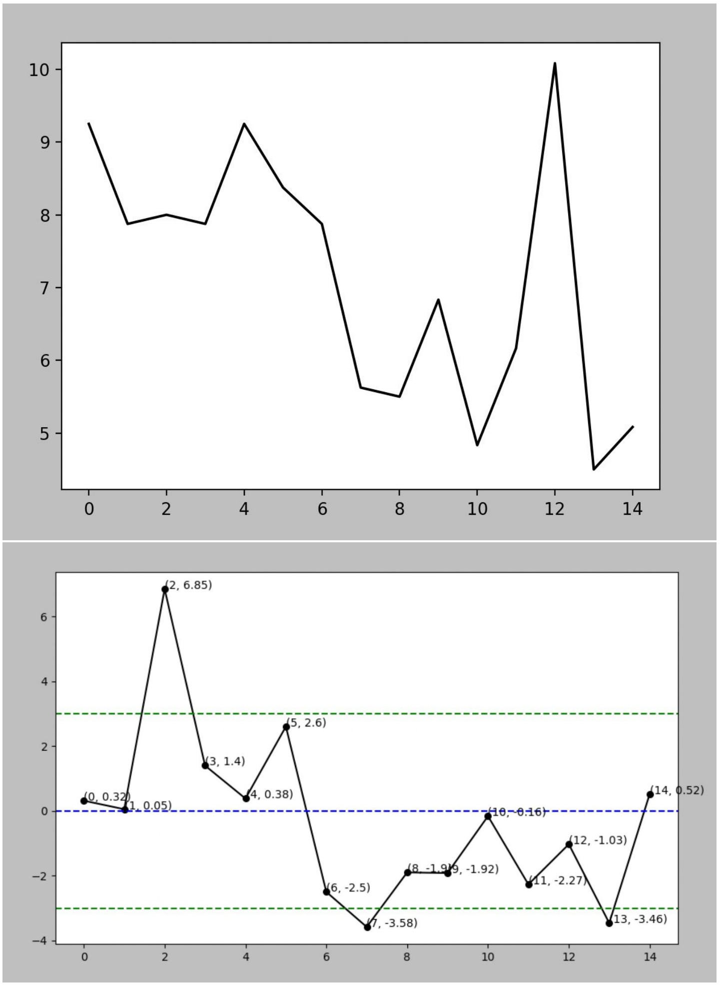


Figure 19.5: Company Alarm Chart (top) and Standardized P-chart (15 days)

have give an alarm but did) will be calculated through the results obtained from this comparison.

Figure 19.5 shows 15-day Company Alarm Chart and the presented system The charts indicate that the presented system gives an alarm on the second day, exceeding the upper control limit while the data on the 7th and 13th days stand below the lower control limit whereas the company's own system does not give any alarm as there is no day exceeding the threshold level (16).

Figure 19.6 shows 30-day company alarm chart and the presented system. Alongside the problematic days detected on the 15-day P-chart, the 30-day P-chart of the presented system detects problems on days 16 to 20, exceeding upper control limit, and on the 25th day, standing below the lower

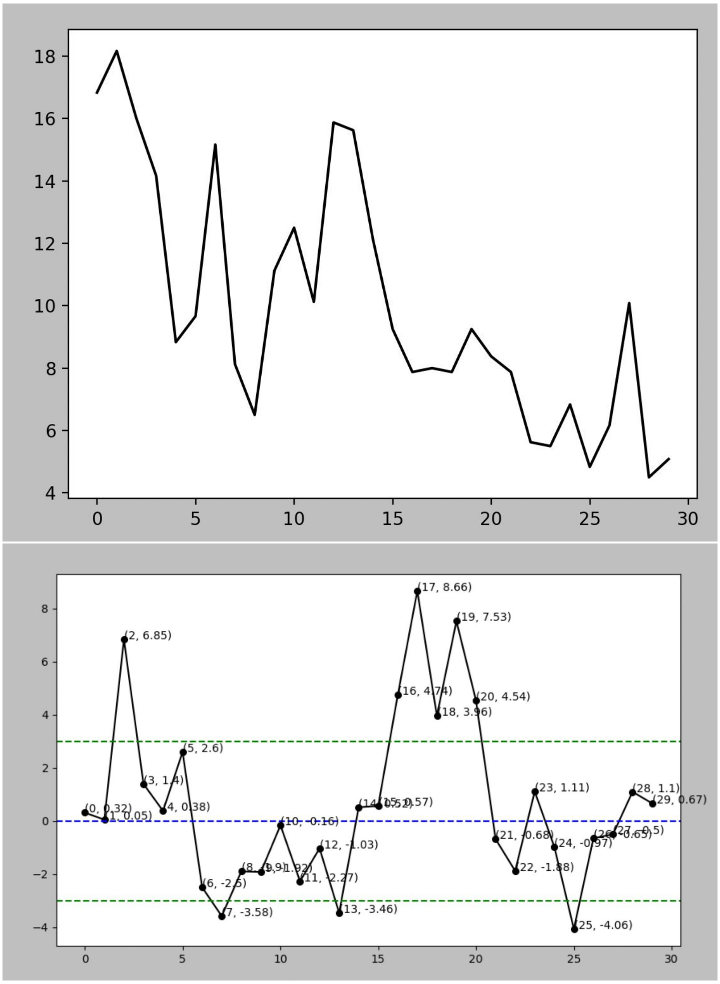


Figure 19.6: Company Alarm Chart (top) and standardized P-chart (30 days)

control limit. And the company's system gives alarm on the first 3 days, exceeding threshold level and on the 7th and 25th days, compatible with the presented system, repair data is indeed low.

The 60-day company alarm chart and the presented system are in Figure 19.7 Alongside the problematic days detected on the 30-day P-chart, on the 60-day P-chart of the presented system days 40, 48 and 54 exceed the upper control limit while the days 42, 51, 56 and 59 stand below the lower control limit while the company system gives alarm on days 2, 8, 11 to 13, 29 and 31 to 33 as the days exceeding the threshold level and on the 56th and 59th days, compatible with the presented system, repair data is indeed low.

Type 1 error percentage is 0% for 15-days, 6.67% for 30-days, and 11.67% 60-days. Type 2 error percentage is 20% is for 15-days, 16.67% for 30-days,

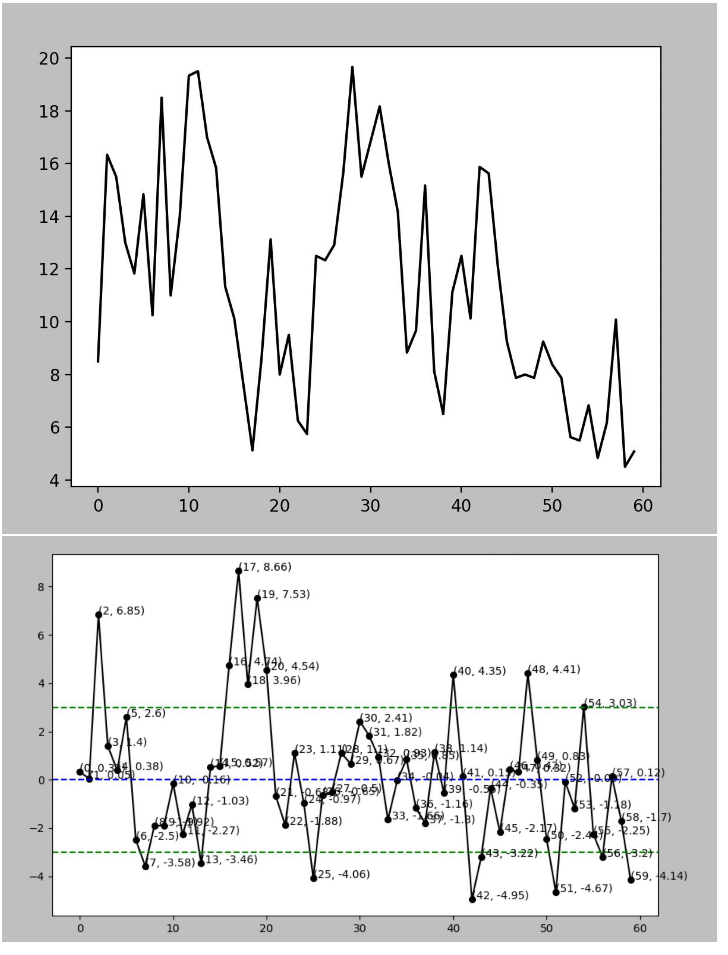


Figure 19.7: Company alarm chart (top) and standardized P-chart 60 days

and 18.33% for 60-days.

19.3 Outcome and Deliverables

19.3.1 Outcome

The correlation between the production line and defect types are examined and the defects' root causes are identified before they pile up to cause production loss. Statistical control charts (p-charts, u-charts) analyzes the data and present a visual report.

19.3.2 Deliverables

The company is supplied with a user interface and a manual consisting of models, algorithms used, and instructions on using the designed system (with practical implementation).

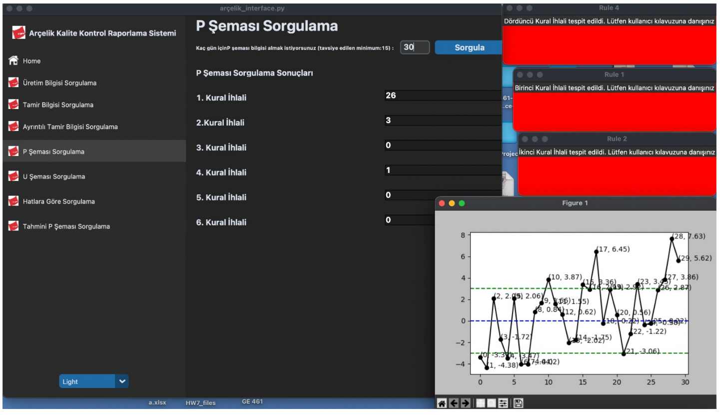


Figure 19.8: UI Screenshot

The User Interface in Figure 19.8 has the following tools: Production interrogation, repair interrogation, detailed repair interrogation, P-chart interrogation, U-chart interrogation and Production line interrogation. The execution of all these tools require the number of work days as an input.

Production and repair interrogation tools show product and repair number of inputted work days while the detailed repair interrogation returns a sub-defect refraction for a chosen defect type. U-chart interrogation creates the U-chart of the inputted defect type and as for the P-chart interrogation tool, it both creates a (standardized) P-chart and notifies the users in case of a zone rule violation (with a pop-up), redirecting them to the user manual. Line-based examination will return defect fraction of defect types in each production line.

19.3.3 Benchmarking and Benefits to the Company

Arçelik's current defect monitoring system is under human inspection so, it is subjective, slow and cost-inefficient. The proposed system will decrease human dependency on the last assembly, deliver an easy-to-implement user interface to the company and record the defects early. With early detection, production loss expectations will diminish significantly. The benchmarking comes after the validation step.

19.3.4 Implementation Plan and Pilot Study

The User Interface code has been sent to the company along with an informative presentation on its implementation. The IT department of the company will integrate it to their system and then, they will send back the results they obtained to confirm whether this new presented system is working as expected.

Bibliography

Gardiner, J. S. and D. C. Montgomery (1987). Using statistical control charts for software quality control. *Quality and Reliability Engineering International 3*(1), 15–20.

Montgomery, D. C. (1997). *Introduction to Statistical Quality Control* (3. ed ed.). Wiley.

Savic, M. (2006, Jan). p-charts in the quality control of the grading process in the high education. *Panoeconomicus 53*(3), 335–347.

20 | Yapay Zeka Tabanlı Çok Değişkenli İş Emri Planlama Sistemi

Supply Chain Wizard, LLC

Proje Ekibi
Mustafa Emre Aydoğdu, Cansu Efe, Ertuğrul Gültepe,
Ziad-Abu Irhayem, Yılmaz Kendirli, Ahmet Tarık Özbek, Ege Özgülen

Şirket Danışmanı
Yunus Emre Yurdagül
Uzman Analist

Akademik Danışman
Mis. Doç. Dr. Arnab Basu
Endüstri Mühendisliği Bölümü

Özet

Önerilen bu rapor, ilaç üreticilerinin karşılaştıkları sorunları tanımlamayı amaçlamaktadır ve SCW, olası çözüm yöntemleri konusunda danışmanlık vermekte ve bazı fikirler sunmaktadır. SCW, IoT uygulamalarının ve OEE yazılımının kullanımıyla tedarik zinciri danışmanlığı sunar. Bu projenin odak noktası, yüksek ve öngörülemeyen değişim süreleri ve makine arızaları nedeniyle ilaç üreticileri için çizelgeleme komplikasyonları ile ilgilidir. Bu komplikasyonlar, yapılan çalışmalar doğrultusunda iki aşamalı bir şekilde çözülmüştür. Bu çözüm yönteminde buluşsal yöntemler ve modelleme sistemlerinden faydalanılmıştır. Aynı zamanda makine öğrenimi yöntemi ile de değişim süreleri tahmin edilmiştir.

Anahtar Sözcükler: IoT, OEE, çizelgeleme, kesinti, yapay zeka, Makine Öğrenimi

AI-based Multi-Variable Dynamic Work Order Planning System

Abstract

This proposed report aims to describe the problems faced by pharmaceutical manufacturers, and SCW advises and offers some ideas on possible solutions. SCW offers supply chain consulting through the use of IoT applications and OEE software. The focus of this project is on scheduling complications for pharmaceutical manufacturers due to high and unpredictable changeover times and machine failures. These complications were resolved in two stages. In this solution method, heuristic methods and modeling systems were used. At the same time, appropriate production planning systems were presented for the company with the machine learning method.

Keywords: IoT, OEE, scheduling, downtime, AI, Machine Learning

20.1 Company Information

Founded in 2014 by Evren Özkaya, LLC is the one of the fastest growing companies in the field of Supply Chain Consulting and Technologies. The main three fields of Supply Chain Wizard are Supply Chain Wizard Academy, Supply Chain Wizard Consulting, and Supply Chain Wizard Solution. The field of Supply Chain Wizard Consulting consists of two main branches, serialization consulting and supply chain consulting. The supply chain consulting branch is concerned with advanced analytics, IoT, big data, blockchain, SC security, supply chain strategy design, E2E process optimization, and automation.

20.2 System Analysis

In the pharmaceutical industry, work order completion speed is essential since this industry directly affects human lives (Cundell, 2022). Particularly in crisis circumstances, the industry needs to be prepared to produce faster to ensure quick and reliable supply for their customers. However, in this industry, idle times and downtimes affect the makespan of the production system negatively which is shown in Figure 20.1.

Idle time is non-productive time not spent on working with appropriate equipment. Idle time is caused by a lack of demand, a lack of manning or unforeseen production interruptions. Downtime is non-productive time caused by failure and maintenance of equipment, activities like cleaning, setup and adjustments, and employee break times. There are two different types of downtimes, planned downtimes and unplanned downtimes. Planned downtimes stand for pre-scheduled intervals of non-productive times. An

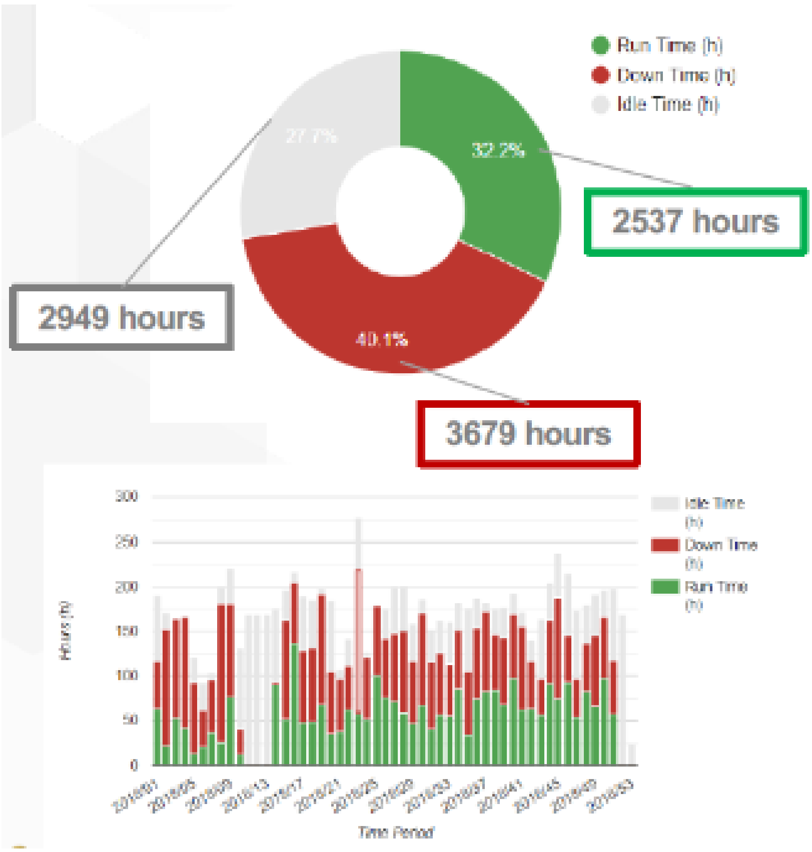


Figure 20.1: The distributions of downtime, run time, and idle time

important example of planned downtime activities is change-over activities like setup and cleaning. Between the production of different products, the production line needs to be cleaned to get rid of any residues from the first item, and a setup must be done before starting to produce the next item. These change-over times are planned downtimes since these activities are expected, and should be considered in the preparation of the schedule. In the current system, 38.6% of the total downtime is due to cleaning. Setup and adjustment times for changeovers account for another 34.1% of total downtime. Hence, 72.7% of all downtime is caused by changeover activities. Another planned downtime to consider during scheduling is maintenance. All machines in the production line must go through a scheduled maintenance process to avoid causing unnecessary costs and disrupting production. Unplanned downtime stands for any downtime on the production line caused by an unexpected problem and usually comes without prior warning. These downtimes can be caused by machine, process, hardware and human failures. These failures account for 23.5% of the total downtime.

20.3 Objective of the Project

The fundamental aim of the project is to create an interface that can be utilized to come up with an efficient schedule under certain constraints. In the preparation of the schedule, the project must account for changeover times between different jobs, which are uncertain and expected to be sequence dependent. These must be predicted via careful analysis of the past data and use of machine learning methods. There are also several other constraints such as due dates for orders, compatibilities of production lines, and products and capacities. Briefly, the objective of the project is to develop an automated decision support system with a dynamic interface to come up with efficient schedules by taking variable runtimes and changeover times together with aforementioned constraints into account. In addition to all of the requirements above, the system must allow humans to make changes to the schedule before coming up with the final plan. As for the critical objective functions, the scheduling heuristic or algorithmic approach will aim to minimize average tardiness in production. These critical objectives are suggested to be achieved by decreasing primarily the changeover times and possibly handling and idle times.

20.4 Performance Measures

By using production resources more efficiently, the customers aim to decrease manufacturing costs, and average makespan while increasing the throughput rates of their production lines. Within the scope of the project, time and cost-efficient schedules under a variety of conditions are aimed to be obtained by considering constraints such as due dates of orders, and product-line compatibilities. In addition to these constraints, parameters like the manufacturing speed of different production lines, and line cleanup and setup times will also be taken into account.

20.5 Model Development and Improvement

The scheduling phase of our project is separated into two stages. In the first stage, work orders are assigned to compatible production lines to minimize total run and changeover times. In the second stage, assigned work orders are scheduled for each production line. Our general model development scheme can be seen in Figure 20.2.

20.5.1 Assignment Phase

Before assigning work orders to the lines, we need to find compatibility scores which consist of information about production time, changeover time,

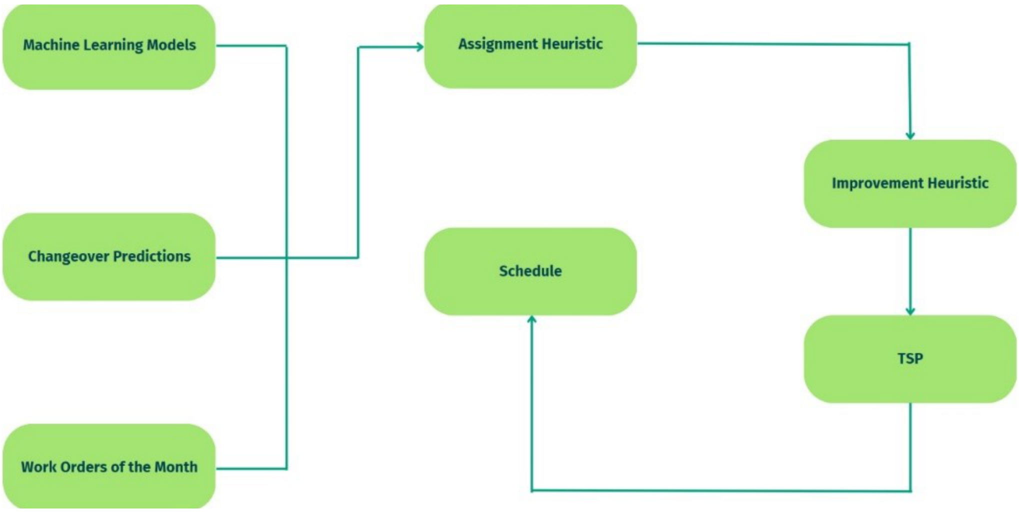


Figure 20.2: Model Development

and downtime. That's why we assigned the work orders to the lines according to the compatibility score calculated as in (20.1). If

$$\begin{aligned} T_{\mathrm{wo}_i,l} &= \text{Estimated runtime needed for work order } \mathrm{wo}_i \text{ in line } l \\ CO_{\mathrm{wo}_i,\mathrm{wo}_j} &= \text{Esimated change-over time between work orders } \mathrm{wo}_i \text{ and } \mathrm{wo}_j \\ l_{\mathrm{wo}} &= \text{List of work orders assigned to line } l, \end{aligned}$$

then we define

$$\text{Change-over Score}_{\mathrm{wo}_i,l} = \frac{1}{|l_{\mathrm{wo}}|} \sum_{\mathrm{wo}_j \in l_{\mathrm{wo}}} CO_{\mathrm{wo}_i,\mathrm{wo}_j}$$

$$\text{Score}_{\mathrm{wo}_i,l} = T_{\mathrm{wo}_i,l} + \text{Change-over Score}_{\mathrm{wo}_i,l}. \tag{20.1}$$

Finding a feasible solution with acceptable performance measures is good, but the aim is to find a near-optimal solution for the problem at hand. To provide near-optimality, an improvement heuristic is utilized after the assignment stage (first stage) is complete to come up with improved work order – production line assignments. The improvement heuristic is an adaptation of the 2-opt heuristic in the context of the scheduling system on hand.

20.5.2 TSP Phase

In this phase, we applied Travelling Salesperson Problem (TSP) formulation to scheduled assigned work orders, and by doing that we minimized the total makespan while accounting for due dates and time window constraints. The TSP formulation uses nodes to represent unique work orders and arcs to denote changeover times. Changeover times are considered between product families rather than between each unique product to simplify the dataset and data-cleaning process, and provide more meaningful insights. Time windows are widened to account for late order filling when strict time windows are unfeasible.

As for the second stage, the TSP formulation used through Google-OR tools to schedule work orders on their assigned production lines already guarantees near-optimality since it uses sophisticated algorithms such as Cristophides Algorithm which provides the best polynomial time approximation for the TSP (Beasley and Christofides, 1989). Thus, any improvement heuristic for the second stage will statistically fail to improve the second-stage solution significantly.

20.6 Validation

We have used 736 work orders and 98 production lines during the validation step and all these values are from May 2022 orders. Our validation models results are below:

- Our model's makespan for all production lines is 617.290 minutes compared to the makespan of the scedule that is actually used by the company, which is 670.114 minutes. As a result, our results provided a 7.75% decrease in total makespan across all lines. This decrease was primarily caused by two different factors: run times and cleanup times.

- Our model's run time for all production lines is 416.974 minutes whereas the company's actual run time is 439.166 minutes.This corresponds to a 5% decrease of the run time.

- Our model's cleanup time for all production lines is 93.630 minutes whereas the company's actual cleanup time is 125.275 minutes. This corresponds to 25% decreases in the cleanup time. This result is to be expected since our models and methods focus primarily on reducing time allocated to the changeover activities.

To ensure the consistency of our results, we also performed an extended validation on the yearly data we have, with a scheduling span of 2 weeks, which allowed us to have 53 trials. Over these trials, our average makespan improvement was 11.06%, and average cleanup time improvement was 7.22%, with reasonable standard deviations. Detailed results from our extended validation is reported in Table 20.1 and in Figure 20.3.

In the validation step, there are some given assumptions that we need to consider to make an appropriate production plan. These assumptions are the following:

	name	min	Q1	median	Q3	max	mean	sd	n
1	Cleanup	0.26	0.33	0.36	0.41	0.53	0.37	0.06	53.00
2	Makespan	0.01	0.07	0.09	0.13	0.47	0.11	0.09	53.00
3	Runtime	-0.01	0.01	0.04	0.08	0.54	0.07	0.11	53.00
4	Setup	-0.36	-0.14	-0.10	-0.05	0.12	-0.10	0.09	53.00

Table 20.1: Statistics of cross-validated improvements

- Planned downtimes for cleanup, maintenance, and setup purposes are encouraged and inevitable since the products dealt with are pharmaceutical chemicals which have low tolerances for any hygiene or maintenance-related faults. They are still expected to make up a significant portion of the makespan, even though there is some space for improvement.

- Feasibility constraints between production lines and products is expected as each production line is capable of producing only a set of defined products and not necessarily all products.

20.7 User Interface

As this project is planned to be a part of Supply Chain Wizard's service module, SCW engineers wanted to use their own User Interface. Therefore this project only consists of a simple UI for the demonstration and clarity purposes. While creating the interface, we were inspired by the actual interface that SCW uses. The interface aims to summarize and display our outputs on the production line. In the dashboard, the employees can view the situation of the production line at any point in time. The interface a glimpse of which shown in Figure 20.4 consists of the following modules:

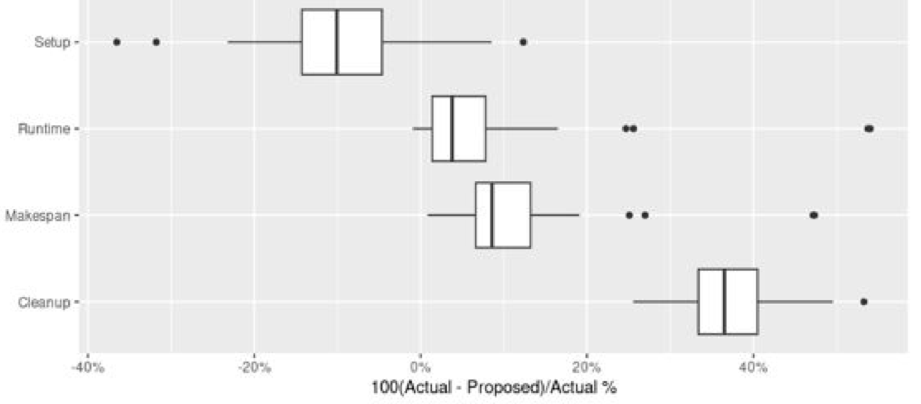


Figure 20.3: Boxplots of cross-validated improvements

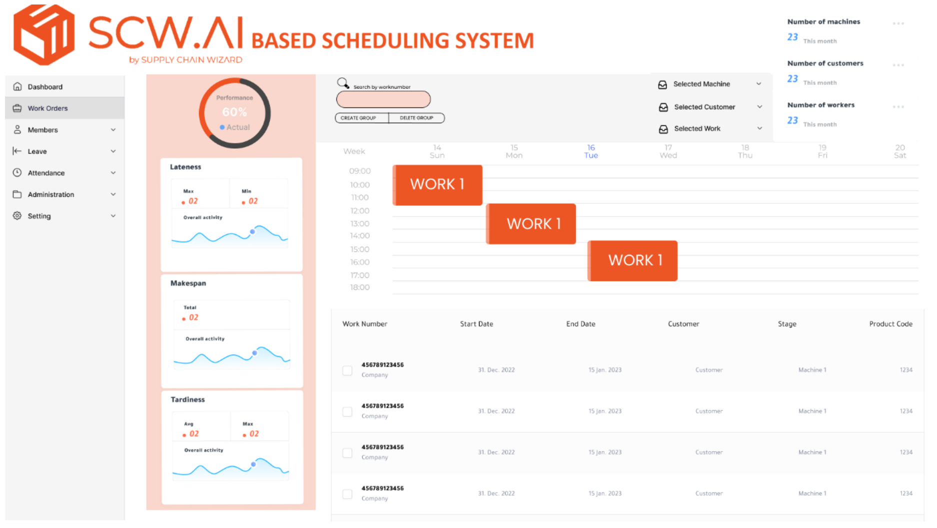


Figure 20.4: The user interface

- Search button for work orders that employees want to oversee.
- Lateness, make span and tardiness tables where employees can see what these values are.
- Schedule of the work orders in a day. Moreover, the work orders which will be done on a weekly basis can be observed.
- How many machines, workers and customers for that work order can be seen.
- Finally at the bottom, there is information that the start and end date of the work order and the company belongs to that work order.

20.8 Project Contributions

Our model will contribute to SCW's future planning system by decreasing total makespan by 7.75% which was approved as a significant contribution by company officials. The application will be used by the Supply Chain Wizard to solve its customers' problems, by allowing the company to increase customer satisfaction by decreasing production times with higher production capacities and increasing operational reliability and efficiency.

Supply Chain Wizard is expected to provide benefits to its clients who will use the system in their daily operations in several areas, including:

- Reduced makespans

- Decreased operational risks
- Increased throughput, production capacity
- Increased revenue due to reduced costs
- Ability to adapt to changing production conditions
- Ability to meet market needs on time

20.9 Project Implementation

After the validation is done, we showed our finalized models and results to the company. They also approved our results, and from our discussion with our industry advisor, we came up with an implementation plan. The first step of our implementation plan will be creating a single application for our project that is easy to use and understand, with a user-friendly interface. This application will include all of our predictive models, heuristic models, and linear models in Python. This application is already in the work, and will be functioning in a short time. After developing the application, we will offer it to our industrial advisor. Upon receiving the application, our advisors will gather work order data from their clients. This data will be different from the data used to train, verify and validate our models. After that, he will use the application we developed on this data, to come up with a schedule. Then, he will compare this schedule with the actual schedule realized by their client, and get benchmarks. He will repeat this process several times to validate the findings. If there are any problems, we will investigate them and provide solutions. Finally, if there is a need to improve the application or its UI, we will reshape it according to the feedback.

Bibliography

Beasley, J. E. and N. Christofides (1989). An algorithm for the resource constrained shortest path problem. *Networks 19*(4), 379–394.

Cundell, T. (2022, September). Gmp challenges with time limitations in pharma manufacturing. https://www.pda.org/pda-letter-portal/home/full-article/gmp-challenges-with-time-limitations-in-pharma-manufacturing [Accessed 19-Apr-2023].

Yurt İçi Sevkiyat Lojistiği Optimizasyonu | 21

Doğadan

Proje Ekibi
Beyza Bakır, Aysel Başak Bula, Bora Çetin, Sude Dübek,
Ömer Durukan Kılıç, Zeynep Kayra Özçakar, Merve Öztürk

Şirket Danışmanı
Yüce Cankur
Kıdemli Lojistik Uzmanı

Akademik Danışman
Prof. Dr. Bahar Yetiş
Endüstri Mühendisliği Bölümü

Özet

Bu raporda, Tam Kamyon Taşıma (TKT) ve Parsiyel Taşıma (PT) olmak üzere iki sevkiyat yapısı arasında paylaştırma yapan bir lojistik optimizasyon problemi ele alınmıştır. Toplam sevkiyat lojistiği maliyetlerini azaltmak için Türkiye'nin önde gelen çay üreticisi Doğadan'a Sevkiyat Lojistiği operasyonları için bir karar destek sistemi sağlamak üzere talep analizi ile birlikte rota bazlı doğrusal karma tamsayılı bir programlama modeli, ve kolay uygulanabilir bir algoritma oluşturulmuştur. Bu algoritmanın sonucunda lojistik maliyetlerinde %3,7'lik bir azalma gözlemlenmiştir. Yapılan pilot çalışmasına göre, kurulan sistem sayesinde lojistik operasyonlarında kolaylık sağlandığı gözlemlenmiştir.

Anahtar Sözcükler: Sevkiyat Lojistiği, Parsiyel Taşımacılık, Matematiksel Modelleme

Domestic Outbound Logistics Optimization

Abstract

This report presents a logistics optimization problem aimed at achieving the optimal allocation of orders between two shipment options: Full Truck Load (FTL) and Less-than Truck Load (LTL). The objective is to reduce the overall outbound logistics costs for Doğadan, a prominent Turkish tea producer. To achieve this, a path-based linear mixed integer programming model and an easy-to-apply algorithm are developed. The solutions derived from this algorithm lead to a notable 3.7% reduction in Outbound Logistics operations costs. A Decision Support System (DSS) for their Outbound Logistics operations is provided during the pilot study, which demonstrated improved ease and effectiveness in conducting logistics operations.

Keywords: Outbound Logistics, Partial Transportation, Mathematical Modelling

21.1 Introduction

Logistics operations have been affected by recent global challenges such as COVID-19 and the Russian invasion of Ukraine. Furthermore, in countries with high inflation such as Turkey, logistics expenditures have drastically increased by 80%. In Turkey, Discounter markets have emerged as a cheaper alternative to national gross markets and retailers, offering better prices through local brands, fewer store expenditures, and less inventory. In a market where consumers prefer discounter options, efficient logistics is essential for better pricing, management of operations, and improved shelf availability. Producers may face a choice between less market availability and greater transportation costs due to frequent orders by markets. This project aims to offer Doğadan solutions to their rising logistics costs due to the increase in fuel prices in Turkey.

21.2 Company and System Analysis

21.2.1 Company Description

Inviting consumers to a “good” life with 48 years of expertise, Doğadan is the founder and market leader of the infusion category in Turkey. More than 1.5 billion tea bags are produced annually in one of Turkey’s largest and most modern tea bag production facilities in Akyurt, Ankara.

Doğadan, affiliate of The Coca Cola Company since 2007, continues to offer healthy and innovative products to contribute to the quality of life of consumers in Turkey and more than 35 countries abroad with the same enthusiasm as on the first day for 48 years.

21.2.2 Logistics System Description of Doğadan

Doğadan has two main logistics systems: inbound and outbound. Inbound logistics cover raw materials and packing materials, contractor transportation, and shipments, while outbound logistics cover domestic and international deliveries. Doğadan works with FCA for outbound logistics planning and with 3PL firms for their logistics operations. They have different operational dynamics for different customers, including Discounters, NKA, and LKA. Orders are viewed and controlled directly through the company's planning system or received indirectly through emails and manually incorporated. Doğadan frequently uses partial transportation due to different customers and dynamics.

21.2.3 Detailed Outbound Logistics Analysis

Doğadan is facing an issue with rising outbound logistics costs due to increasing inflation rates and fuel prices. To cut down expenses, the company aims to regulate its outbound logistics operations.

NKA accounts for 37.5% of Doğadan's total orders, and they monitor order arrivals closely. Late orders are canceled immediately, incurring additional fees. Precision in delivery timing is crucial for Doğadan. LKA has more flexible schedules, so arrangements and adjustments can be made more easily when delivering products to them.

Doğadan's discounter customers include ŞOK, BİM, and A101. Unlike NKA, each discounter warehouse places separate orders at irregular periods depending on their inventory levels. Thus, FTL shipments to discounters are significantly lower than those to NKA or LKA. FTL shipments are mainly made to non-discounters (91%) while LTL shipments are mainly made to discounters (59% in terms of desi).

To handle the irregularity in order frequencies and amounts, Doğadan waits for local discounter warehouses for two extra days to accumulate demand and reduce extra LTL costs. This approach avoids order cancellations and sales losses since BİM and A101 do not cancel orders unless the backlog becomes unreasonably long.

21.3 Problem Definition

The main issue for the company is the rising outbound logistics costs for Discounter customers due to irregular order periods and smaller order amounts. This results in more shipments using LTL, leading to higher costs. Therefore, the company is focusing on improving local Discounter deliveries to address this problem.

The project aims to reduce outbound logistics costs by optimizing vehicle utilization and creating a shipment strategy for the company using a decision support system (DSS). Inputs include order information, truck capacities, cost information, and customer groups. The system helps select between LTL/FTL and possible routes to improve decision-making.

21.4 Literature Review

Previous academic work on deciding between different modes of transportation is generally based on problem specific approaches. (Meixell and Norbis, 2008) The literature uses methods such as Genetic Algotithms (Caputo et al., 2006), Multicriteria Decision Analysis (Vega et al., 2021) and MILP Models. For larger sized instances, some examples from the literary work can be routing and path based approach, or heuristic applications such as Chu (2005) that decides between FTL and LTL modes.

While there are various studies available on optimizing shipment modes and determining the appropriate balance between FTL and LTL transportation, none of the existing resources specifically match the requirements and constraints of the current problem. Therefore, it is necessary to develop a comprehensive solution approach that is tailored to the company's specific needs, constraints, and resources.

However, many of the mathematical modeling formulations and heuristic approaches discussed in Outbound Logistics Planning could be adapted to develop a system that determines the optimal balance between FTL and LTL shipments, as well as routing and scheduling strategies. Additionally, clustering methods could be utilized to break down the problem into more manageable pieces and then assemble a complete picture.

21.5 Detailed Solution Methodology

The solution strategy incorporates dynamics, assumptions, constraints, and various aspects of the problem, and is shown in Figure 21.1. After realiz-

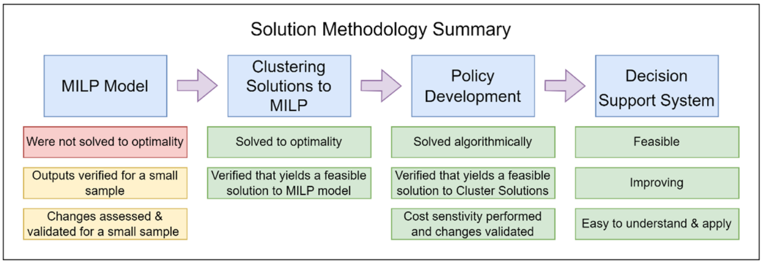


Figure 21.1: Rough-cut Solution Methodology

ing that the original mixed-integer programming model was too complex to yield a solution due to computational limitations, clustering methods to divide the problem into more manageable parts is utilized. A policy is developed to determine the optimal shipment for each subproblem and compared it with the newly developed shipment policy for Doğadan.

The results showed that the policy improved the 2022 data by 3.72%, albeit 2% less than the model proposed. The policy algorithm was implemented with an interface to manage routine tasks and clustering management, making it suitable for practical operational applications in Doğadan's facility.

The main decision for Doğadan is to minimize logistics costs by determining the optimal mode of transportation, and the problem can be reduced to three questions: which mode of transportation to use, whether to merge shipments, and whether to wait for an opportunity to decrease costs. Doğadan's conventional shipment strategy lacks customer-specific forecasting and optimization, but the proposed solution methodology consists of preparing a deterministic optimization model, dividing customers into clusters, solving deterministic MILP for each isolated cluster, determining the timing of the shipment for each cluster by conducting a frequency analysis, and preparing a shipment policy for Doğadan based on the cluster-day assignments. This methodology can effectively overcome the issue of uncertainty that arises with the conventional method.

21.5.1 First Phase: Deterministic Optimization

The path-based formulation is used to optimize the logistics operations of Doğadan assuming demand certainty. The approach involves creating a mathematical model that determines the optimal shipment strategy based on shipment times, locations, and amounts. To run this model, every possible path for shipments is generated, requiring significant preprocessing effort. Parameter preprocessing is conducted by a python script that takes the shipment data and converts company information to parameter matrices that is ready to be used in the MILP Model.

Mixed Integer Programming Model

The aim is to minimize outbound logistics costs using routes generated from a previous algorithm. A MILP model is proposed in the appendix to solve the problem optimally as a whole. The demand must be satisfied within a certain number of days for every order. Due to the model's size and complexity, it is divided into sub-models for efficient computation.

Path Generation

The paths between Doğadan's production facility and Discounter's warehouses are listed in a matrix, which is generated by considering all possible combinations of districts. The paths that have a total distance of more than 20% km of the final destination are eliminated, while the others are considered possible shipment routes. The selection of paths is shown in Figure 21.2. The generated paths are filtered based on the constraints given by the industrial advisor (IA).

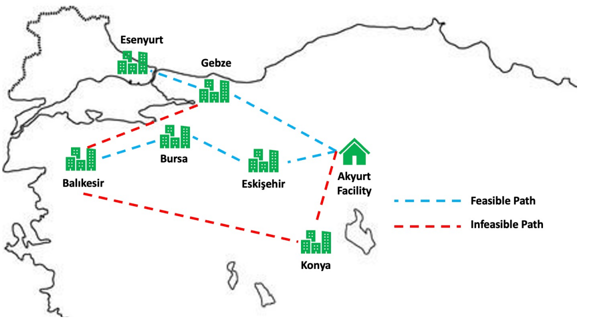


Figure 21.2: Route Selection Example

FTL – LTL Cost Generation

The cost parameters of the model are estimated based on Doğadan's shipment data. Separate regression models are created for each Discounter warehouse to estimate FTL costs and a Multivariate Random Forest model is fitted to predict FTL costs while taking distances into account. An estimated FTL cost matrix is generated for each day and depot for box trucks and trailer trucks. To produce the LTL cost matrix, a linear regression model is utilized. The estimated cost values simulate the 2022 cost by overestimating it by only 2.02%.

MILP Solutions

Due to the size of the problem, the model could not be solved as a whole, so two scenarios were considered, and runtime statistics were collected. The first scenario was based on node counts, which showed a significant increase in runtimes as the node count increased. Therefore, the approach was shifted to a "from part to whole" approach.

21.5.2 Second Phase: Submodel Optimization

The potential solutions in the relevant literature were researched and a constrained K-means algorithm was ultimately adopted. The approach involved dividing the problem into more manageable sections for analysis, and the clusters were formed by setting the minimum and maximum cluster sizes to 6 and 12, respectively. The nodes were clustered into 17 distinct groups, each containing between 6 and 12 nodes.

After the clusters were created, the model was repeatedly solved using the CPLEX Python API. The model terminates after 12 hours and by optimally solving all 17 clusters. The solutions are shown in Table 21.1, which forms the basis for testing the effectiveness of the policy algorithm development phase. We compared the policy we developed with the results above for different parameter values.

Cluster	Impr. (%)	Cluster	Impr. (%)
SE Anatolia	0.00%	SE Marmara	4.32 %
Istanbul I	8.99%	Ankara II	11.30 %
Medit. West	1.49%	NE Anatolia	0.00%
Medit. East	4.28%	SE Anatolia	0.00%
Izmir North	2.44%	Aegean	0.41%
Istanbul II	2.38%	Istanbul III	0.29&
Ankara I	17.01%	Central Anatolia	0.47%
Black Sea	12.78%	Izmir South	2.65%
East Marmara	9.38%		

Table 21.1: Clustering Solutions

21.5.3 Third Phase: Policy Development

The next step is to use a heuristic approach to solve the MILP model and compare the results to create an easy-to-apply policy.

Frequency Analysis for Delivery Day Selection

Initially, the days are assigned to clusters based on demand characteristics. A scoring method is developed to choose pairs of two delivery days for each cluster based on the total pallet size ordered on those days. The method gives weighted scores to each pair and chooses the one with the highest score. The weights prioritize depots with larger orders, and different weights are calculated based on the total pallet size of the orders. The Monday-Thursday, Tuesday-Friday, and Wednesday-Saturday pairs are applied to the system, and the demand structure is analyzed by summing consecutive three weights. The pair that is more evenly distributed is chosen based on a simple multiplication calculation. The final step of the scoring system chooses the pair with the highest result after the multiplication. This way, delivery days are selected, and the frequency analysis section is finalized.

Policy Algorithm Development and Results

The shipment algorithm helps logistics managers decide which orders can be combined and which delivery type to use to minimize costs for their company is presented. The daily order list is sorted from maximum to minimum pallet size, and different delivery types are chosen based on the total demand amount and breakeven points. The effectiveness of the model is tested through scenario analysis, and the output cost is compared to the Clustering MILP costs. Every day the orders are sent according to the procedure in Listing 21.1.

Listing 21.1: Shipment Mode Selection Algorithm

```
input: bvn_16, bvn_30, orders
output: tuple(ftl_30, ftl_16, ltl)
pack = choose max 3 elements from orders
while len(orders) > 0:
  if sum(pack) < bvn_16:
    ltl.append(all elements in orders)
    orders = orders - orders
  elseif sum(pack) in [bvn_16, 16]:
    ftl_16.append(pack)
    orders = orders - pack
  elseif sum(pack) in (16, bvn_30):
    divide = pack - 16
    ftl_16.append(pack - divide)
    orders = orders - pack
    orders.append(divide)
  elseif sum(pack) in [bvn_30, 30]:
    ftl_30.append(pack)
    orders = orders - pack
  elseif sum(pack) > 30:
    divide = pack - 30
    ftl_30.append(pack - divide)
    orders = orders - pack
    orders.append(divide)
  endif
return (ftl_30, ftl_16, ltl)
end
```

The Policy Algorithm results show a slight underperformance of 1.88% compared to the clustering model but still improve the actual solution by about 3.7%. Some disruptions were caused by some clusters that were assigned to Thursdays but could only satisfy the orders of Friday on the following Monday, causing small cost deviations of up to 1%.

21.6 Model and Policy Validation

The model and policy algorithm rely on demand and cost data as inputs, with emphasis on cost changes due to potential disruptions in real life. LTL and FTL costs were estimated and calibrated, with the MILP model playing an important role.

Two different cost analyses on LTL costs (and implicitly, breakeven points) were performed and their costs were compared in this study:

- LTL values were estimated and multiplied with 100 equidistant coefficients ranging from 1 to 2, and the costs between the policy results and 2022 actual costs calculated with the new LTL estimations were compared.

- LTL costs were created by dividing the FTL16 costs into 100 values between 7 and 12, hence fixed breakeven points were used.

21.6.1 LTL Cost Calibration

LTL cost calibration refers to multiplying the LTL costs with constants to test different breakeven values. The actual estimated 2022 costs are linear in LTL costs because FTL costs are the same. Decreasing the average breakeven point of the problem by increasing LTL costs is expected to result in a decreasing increase in the cost trend when the policy is executed for increasing multiplier values.

The calibration analysis in Figure 21.3 shows that the policy performed as expected and yielded better improvement values if the breakeven points decreased in real life. The policy results remain valid and improve under different breakeven point characteristics, and the improvement increases

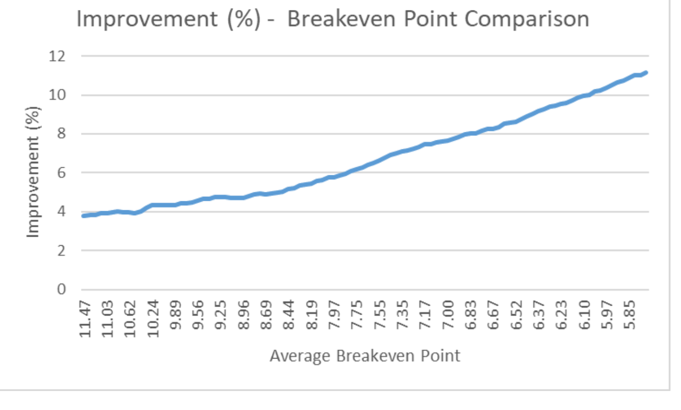


Figure 21.3: Improvement Analysis Under Multiplied LTL costs

as the policy objective function is concave with respect to the decreasing breakeven points.

21.6.2 Fixed Breakeven Point Analysis

In this validation analysis, LTL costs were estimated by dividing all matrix entries to 100 fixed breakeven values between 7 and 12 using approved FTL16 costs. It is expected that the costs shall decrease since the FTL costs are the same and the LTL costs are decreasing. Figure 21.4 show the cost comparison and improvement ratios.

It can be observed that the improvement rate is in a decreasing trend with respect to the breakeven points, which was also expected. Under fixed breakeven point assumptions, the policy still improves, with costs getting closer to the actual costs of 2022 as the breakeven point increases.

The policy algorithm is valid as it improves under different cost inputs and characteristics. And the improvement percentages are subject to comparison as the pilot study and operational statistics are observed.

21.7 Implementation and Pilot Study

The implementation plan has two phases: a manual phase followed by a phase that utilizes a decision support system (DSS). The pilot study commenced on 24 April 2023, with the first two weeks being manual and the following two weeks utilizing the DSS. During the initial two-week manual phase, the company's decision-makers, primarily the IA, followed the determined shipment policy for the provided clusters. Shipments were made to specific customers on predetermined days, which aligns with the company's methodology, clusters, and day selections, as opposed to what the IA had been doing previously.

The manual execution of the methodology produced results that were

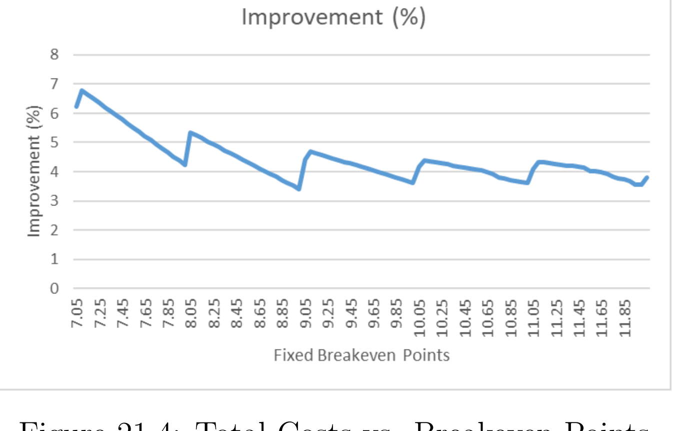


Figure 21.4: Total Costs vs. Breakeven Points

completely aligned with expectations and the model's optimal, ideal scenario. The manual approach did not impede the smooth, efficient, and effective implementation of the methodology.

The subsequent two weeks of the pilot study involved the utilization of the DSS, which was provided to the IA as an executable file that did not compromise privacy. The DSS required modifications to integrate with the company's IT systems. Although pallet information was initially kept in a separate interface, these issues were resolved, and the DSS was seamlessly integrated and has been operating effectively. Consequently, the company can now make decisions in mere seconds utilizing the provided DSS, resulting in significant time savings when making shipment decisions.

The pilot study is set to conclude on 19 May, thereby concluding the four-week implementation period.

21.8 Benchmarking and Benefits to Doğadan

The project brought several benefits to Doğadan, primarily decreasing logistics costs and increasing truck utilization. The policy showed better improvements for clusters around Istanbul and Gebze, with up to 12% improvement achieved as the breakeven rate decreased to 6%. The DSS was 3.7% better than the 2022 cost data, resulting in a saving of 153,654.87 TL, and provided improvement in all possible cost scenarios. The elimination of uncertainty in shipment decisions was another significant benefit, as the DSS allowed for certain shipping to specific customers at certain times using appropriate routes that yielded the least cost possible. Furthermore, there was an expected significant decrease in labor time for warehouse workers and truck drivers due to the easier handling of packages.

21.9 Conclusion and Remarks

Yüce Cankur, the IA, has consistently expressed his satisfaction with the solution strategy, the handling of operations, and the simplification of shipment decisions. The company had been facing challenges due to rising shipment, transportation, and fuel costs, underscoring the significance of a cost-effective shipment strategy. The company executives emphasized that the improvements in outbound logistics costs were achieved without compromising customer satisfaction, a key performance metric for them.

Looking forward, we recommend that the company utilize the DSS regularly and commit to this solution strategy to sustain the solution system. Additionally, we suggest that their IT department support the integration of our DSS into their databases and systems more efficiently. As their systems may change in the future, updating or integrating our DSS effectively will be necessary, and they will need to make the necessary amendments carefully.

Bibliography

Caputo, A., L. Fratocchi, and P. Pelagagge (2006). A genetic approach for freight transportation planning. *Industrial Management and Data Systems 106*(5), 719–738.

Chu, C.-W. (2005). A heuristic algorithm for the truckload and less-than-truckload problem. *European Journal of Operational Research 165*(3), 657–667.

Meixell, M. and M. Norbis (2008, 08). A review of the transportation mode choice and carrier selection literature. *International Journal of Logistics Management, The 19*, 183–211.

Vega, D., D. Lemos, P. Silva, and J. Vieira (2021). Criteria analysis for deciding the LTL and FTL modes of transport. *Gestão en Produção 28*(2), 31–40.

Appendix: MILP Model

1. **Important Sets**

$$
\begin{aligned}
I &= \{1, 2, 3, ..., \mathcal{I}\} && \text{Warehouse/Destination Indices} \\
T &= \{1, 2, 3, ..., \mathcal{T}\} && \text{Time Index (days)} \\
J &= \{1, 2, 3, ..., \mathcal{J}\} && \text{Route Index} \\
O &= \{1, 2, 3, ..., \omega\} && \text{Order Index} \\
N = T - \{\mathcal{T}, ..\mathcal{T} &- p + 1\} && \text{First } \mathcal{T}\text{-p time indices} \\
L = \{\mathcal{T}, ..\mathcal{T} &- p + 1\} && \text{Last p time indices}
\end{aligned}
$$

2. **Parameters**

- p: Delay tolerance parameter (in days)
- D_{ot}: Demand of order o at time t
- U_{ot}: 1 if there is a demand on o at time t, 0 otherwise.
- fc^{30}_{jt}: Cost for 30 pallet FTL shipment for each route j.
- fc^{16}_{jt}: Cost for 16 pallet FTL shipment for each route j.
- lc_{it}: unit cost for per pallet for LTL shipment to node i.
- Q_{oi}: 1 if order o is sending to node i, 0 otherwise.

- r_{ij}: 1 if route j includes node i, 0 otherwise.
- Q^{16}: Capacity of 16 pallet truck
- Q^{30}: Capacity of 30 pallet truck
- M: A large number $M \in \mathbb{R}^{++}$

3. **Decision Variables**

- K^{16}_{jt}: number of 16-pallet trucks following route j in day t
- K^{30}_{jt}: number of 30-pallet trucks following route j in day t
- A^t_{oij}: number of pallets delivered in day t to node i with FTL 30 in route j
- B^t_{oij}: number of pallets delivered in day t to node i with FTL 16 in route j
- C^t_{oi}: number of pallets delivered in day t to node i with LTL
- a^t_{oj}: number of pallets delivered in day t for order o with FTL 30 in route j
- b^t_{oj}: number of pallets delivered in day t for order o with FTL 16 in route j

4. **Objective Function**

$$\text{minimize} \sum_{o=1}^{\omega}\sum_{t=1}^{\mathcal{T}}\sum_{i=1}^{\mathcal{I}} lc_{it}C^t_{oi} + \sum_{t=1}^{\mathcal{T}}\sum_{j=1}^{\mathcal{J}} fc^{16}_{jt}K^{16}_{jt} + fc^{30}_{jt}K^{30}_{jt}$$

5. **Constraints**

$$A^t_{oij} + B^t_{oij} + C^t_{oi} \leq MQ_{oi} \quad \forall t \in T, \forall o \in O, \forall j \in J, \forall i \in I \qquad (21.1)$$

$$A^t_{oij} + B^t_{oij} \leq Mr_{ij} \quad \forall t \in T, \forall o \in O, \forall j \in J, \forall i \in I \qquad (21.2)$$

$$D_{os} = U_{os}(\sum_{t=s}^{s+p}\sum_{j=1}^{\mathcal{J}}\sum_{i=1}^{\mathcal{I}}(A^t_{oij} + B^t_{oij}) + \sum_{t=s}^{s+p}\sum_{i=1}^{I} C^t_{io}) \quad \forall s \in N, \forall o \in O \qquad (21.3)$$

$$D_{os} = U_{os}(\sum_{t=s}^{\mathcal{T}}\sum_{j=1}^{\mathcal{J}}\sum_{i=1}^{\mathcal{I}}(A^t_{oij} + B^t_{oij}) + \sum_{t=s}^{\mathcal{T}}\sum_{i=1}^{I} C^t_{io}) \quad \forall s \in L, \forall o \in O \qquad (21.4)$$

$$a^t_{oj} \geq \sum_{i=1}^{\mathcal{I}} A^t_{oij} \quad \forall t \in T, \forall o \in O, \forall j \in J \qquad (21.5)$$

$$b_{oj}^t \geq \sum_{i=1}^{\mathcal{I}} B_{oij}^t \quad \forall t \in T, \forall o \in O, \forall j \in J \tag{21.6}$$

$$\sum_{o=1}^{\omega} \frac{a_{oj}^t}{Q^{30}} \leq K_{jt}^{30} \quad \forall t \in T, \forall j \in J \tag{21.7}$$

$$\sum_{o=1}^{\omega} \frac{b_{oj}^t}{Q^{16}} \leq K_{jt}^{16} \quad \forall t \in T, \forall j \in J \tag{21.8}$$

$$a_{oj}^t, b_{oj}^t, A_{oij}^t, B_{oij}^t, C_{oi}^t \geq 0 \quad \forall t \in T, \forall o \in O, \forall j \in J, \forall i \in I \tag{21.9}$$

$$a_{oj}^t, b_{oj}^t, K_{jt}^{30}, K_{jt}^{16} \in \mathbb{Z} \quad \forall t \in T, \forall o \in O, \forall j \in J, \forall i \in I \tag{21.10}$$

Dizin

Index