

**T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**DARBOĞAZ BİR MAKİNADA METASEZGİSEL YÖNTEMLERLE
TOPLAM HAZIRLIK ZAMANI MİNİMİZASYONU**

YÜKSEK LİSANS TEZİ

Muhammet AYDIN

Endüstri Mühendisliği Anabilim Dalı

HAZİRAN 2025

**T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**DARBOĞAZ BİR MAKİNADA METASEZGİSEL YÖNTEMLERLE
TOPLAM HAZIRLIK ZAMANI MİNİMİZASYONU**

YÜKSEK LİSANS TEZİ

Muhammet AYDIN

Endüstri Mühendisliği Anabilim Dalı

Tez Danışmanı: Dr.Öğr.Üyesi Halil İbrahim DEMİR

HAZİRAN 2025

Muhammet AYDIN tarafından hazırlanan “DARBOĞAZ BİR MAKİNADA METASEZGİSEL YÖNTEMLERLE TOPLAM HAZIRLIK ZAMANI MİNİMİZASYONU” adlı tez çalışması 24.06.2025 tarihinde aşağıdaki jüri tarafından oy birliği/oy çokluğu ile Sakarya Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı Yüksek Lisans tezi olarak kabul edilmiştir.

Tez Jürisi

Jüri Başkanı :	Dr. Öğr. Üyesi Mümtaz İPEK Sakarya Üniversitesi	
Jüri Üyesi :	Dr. Öğr. Üyesi Halil İbrahim DEMİR Sakarya Üniversitesi (Danışman)	
Jüri Üyesi :	Dr. Öğr. Üyesi Onur CANPOLAT Sakarya Uygulamalı Bilimler Üniversitesi	



ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ

Sakarya Üniversitesi Fen Bilimleri Enstitüsü Lisansüstü Eğitim-Öğretim Yönetmeliğine ve Yükseköğretim Kurumları Bilimsel Araştırma ve Yayın Etiği Yönergesine uygun olarak hazırlamış olduğum “**DARBOĞAZ BİR MAKİNADA METASEZGİSEL YÖNTEMLERLE TOPLAM HAZIRLIK ZAMANI MİNİMİZASYONU**” başlıklı tezin bana ait, özgün bir çalışma olduğunu; çalışmamın tüm aşamalarında yukarıda belirtilen yönetmelik ve yönergeye uygun davrandığımı, tezin içerdiği yenilik ve sonuçları başka bir yerden almadığımı, tezde kullandığım eserleri usulüne göre kaynak olarak gösterdiğimi, bu tezi başka bir bilim kuruluna akademik amaç ve unvan almak amacıyla vermediğimi ve 20.04.2016 tarihli Resmi Gazete’nde yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince Sakarya Üniversitesi’nin abonesi olduğu intihal yazılım programı kullanılarak Enstitü tarafından belirlenmiş ölçütlere uygun rapor alındığını, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun ortaya çıkması halinde doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi beyan ederim.

(24/06/2025)

(imza)

Muhammet AYDIN



TEŞEKKÜR

Öncelikle, yüksek lisans öğrenimim süresince manevi desteklerini hiçbir zaman esirgemeyen aileme en içten teşekkürlerimi sunarım.

Lisans eğitimim boyunca tasarım dersi, lisans tezi, umde programı ve yüksek lisans sürecimde ise bu tez çalışmamda danışmanlığımı üstlenen, bilgi ve tecrübeleriyle her zaman yol gösterici olan Dr. Öğr. Üyesi Halil İbrahim Demir'e değerli katkıları ve rehberliği için teşekkür ederim.

Ayrıca, Sakarya Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı'nda görev yapan tüm değerli öğretim üyelerine eğitim hayatım boyunca sundukları katkılar ve sağladıkları akademik ortam için teşekkürlerimi sunarım.

Muhammet AYDIN



İÇİNDEKİLER

Sayfa

ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ.....	v
TEŞEKKÜR.....	vii
İÇİNDEKİLER.....	ix
KISALTMALAR.....	xi
SİMGELER.....	xiii
TABLO LİSTESİ.....	xv
ŞEKİL LİSTESİ.....	xvii
ÖZET.....	xix
SUMMARY.....	xxi
1. GİRİŞ.....	1
1.1. Problemin Tanımı.....	3
1.2. Literatür Araştırması.....	4
2. YÖNTEM.....	9
2.1. NP Zor Modeller.....	9
2.2. Kombinatoryal Optimizasyon.....	9
2.3. Hamilton Yolu Problemi.....	10
2.4. Problemin Matematiksel Modeli.....	11
2.4.1. Problemin matematiksel modelin parametreleri.....	11
2.4.2. Problemin matematiksel modelin amaç fonksiyonu.....	12
2.4.3. Problemin matematiksel modelin açıklanması.....	12
3. UYGULAMA.....	13
3.1. Veri Toplama ve İlk Durum Analizi.....	13
3.2. Tedarik Zinciri Departmanından Gelen Üretim Bilgileri.....	13
3.3. Tabu Arama Algoritması Adımları.....	16
3.4. Tavlama Benzetimi Algoritması Adımları.....	18
3.5. Parçacık Sürü Optimizasyonu Algoritması Adımları.....	20
4. TARTIŞMA.....	23
4.1. Tartışma Değerlendirme.....	23
4.2. Mann-Whitney U Testi.....	27
5. SONUÇ VE ÖNERİLER.....	31
5.1. İstatiksel Analiz.....	31
5.2. Sonuç Değerlendirme ve Öneriler.....	32
KAYNAKLAR.....	35
EKLER.....	39
ÖZGEÇMİŞ.....	45



KISALTMALAR

CPU	: Merkezi İşlem Birimi
ERP	: Kurumsal Kaynak Planlaması
GA	: Genetik Algoritması
MCDM	: Çok Kriterli Karar Verme
OEE	: Genel Ekipman Etkinliği
PSO	: Parçacık Sürü Optimizasyonu
SA	: Tavlama Benzetimi Algoritması
SMED	: Tek Dakikada Kalıp Değişimi
TS	: Tabu Arama Algoritması
WMS	: Depo Yönetim Sistemi



SİMGELER

- C_{max} : Maksimum tamamlanma zamanı
 i, j, k : İndis (örneğin iş istasyonu, işlem sırası vb.)
 $f(x)$: Amaç fonksiyonu değeri
 G_{best} : Sürüdeki en iyi pozisyon
 N : Komşuluk
 π : Belirli bir çözüm sırası veya permütasyon
 P_{best} : Parçacığın en iyi pozisyonu
 p_i : Parçacığın pozisyonu
 S : Kurulum süresi
 T : Sıcaklık
 TL : Tabu Listesi
 v_i : Parçacığın hızı



TABLO LİSTESİ

Sayfa

Tablo 1.1. Literatür Taraması 1.Tablo.....	6
Tablo 1.2. Literatür Taraması 2.Tablo.....	7
Tablo 3.1. 10 Günlük Üretim Planı.	14
Tablo 3.2. Hamurlar Arası Geçiş Süresi.	16
Tablo 3.3. Tabu Arama Algoritması Kaba Kodu.	17
Tablo 3.4. Tavlama Benzetimi Algoritması Kaba Kodu.....	19
Tablo 3.5. Parçacık Sürü Optimizasyonu Kaba Kodu.	21
Tablo 4.1. Hazırlık Süreleri Karşılaştırılması.	28
Tablo 4.2. Mann–Whitney U Sonuçlar.....	29



ŞEKİL LİSTESİ

Sayfa

Şekil 3.1. Tabu Arama Algoritması Uygulama Adımları.....	18
Şekil 3.2. Tavlama Benzetimi Algoritması Uygulama Adımları.	20
Şekil 3.3. Parçacık Sürü Optimizasyonu Uygulama Adımları.....	22
Şekil 4.1. Tabu Arama Algoritması ve Tedarik Zinciri Departmanı Karşılaştırması.	24
Şekil 4.2. Tavlama Benzetimi ve Tedarikçi Zinciri Karşılaştırması.	24
Şekil 4.3. Parçacık Sürü Optimizasyonu ve Tedarikçi Zinciri Karşılaştırması.	25
Şekil 4.4. Tüm Algoritmaların Hazırlık Sürelerinin Karşılaştırılması.....	26



DARBOĞAZ BİR MAKİNADA METASEZGİSEL YÖNTEMLERLE TOPLAM HAZIRLIK ZAMANI MİNİMİZASYONU

ÖZET

Anahtar Kelimeler: Kurulum Süresi Optimizasyonu, Meta-Sezgisel Algoritmalar, Üretim Çizelgeleme, Tavlama Benzetimi, Lastik Üretimi

Günümüz sanayi ortamında hızla artan küresel rekabet, üretim süreçlerinin daha verimli, esnek ve düşük maliyetli hale getirilmesini zorunlu kılmaktadır. İşletmelerin sürdürülebilir rekabet avantajı sağlayabilmeleri, üretim kaynaklarını etkin şekilde kullanmalarına, zaman kayıplarını azaltmalarına ve operasyonel maliyetlerini minimum düzeye indirmelerine bağlıdır. Bu bağlamda, üretim çizelgeleme ve özellikle ürün değişimlerinde ortaya çıkan kurulum sürelerinin optimize edilmesi, üretim verimliliğini doğrudan etkileyen stratejik bir faktör olarak ön plana çıkmaktadır. Kurulum sürelerinin optimize edilmesi, üretim hattının daha etkin çalışmasını sağlamakla kalmaz, aynı zamanda üretim kapasitesinin artırılmasına ve kaynak kullanımının daha sürdürülebilir hale gelmesine katkıda bulunur.

Bu yüksek lisans tezinde, lastik üretim hattında darboğaz oluşturan bir makinadaki ürün sıralamasının optimize edilmesi yoluyla toplam kurulum sürelerinin minimize edilmesi hedeflenmiştir. Ele alınan problem, kombinatoriyal yapısı nedeniyle klasik deterministik yöntemlerle çözülmesi oldukça zor olan, geniş çözüm uzayına sahip bir optimizasyon problemidir. Bu nedenle çözüm sürecinde Tabu Arama, Tavlama Benzetimi ve Parçacık Sürü Optimizasyonu olmak üzere üç farklı meta-sezgisel algoritma kullanılmış ve bu algoritmaların performansları gerçek üretim verileriyle test edilerek karşılaştırmalı olarak değerlendirilmiştir.

Analizler 10 günlük üretim planları üzerinden gerçekleştirilmiş ve algoritmaların önerdiği sıralamalar doğrultusunda oluşan toplam kurulum süreleri, tedarik zinciri departmanının kullandığı mevcut sıralama yöntemiyle karşılaştırılmıştır. Sonuçlara göre, Tavlama Benzetimi algoritması ortalama %13,96, Parçacık Sürü Optimizasyonu %12,83 ve Tabu Arama algoritması ise %10,71 oranında iyileştirme sağlamıştır.

Elde edilen bu sonuçların istatistiksel anlamlılığı parametrik olmayan Mann-Whitney U testi ile test edilmiş ve meta-sezgisel algoritmalarla elde edilen çözümlerin, geleneksel sıralama yöntemine göre anlamlı biçimde üstün olduğu tespit edilmiştir. Bu çalışma, üretim sistemlerinde sıkça karşılaşılan ancak çoğu zaman göz ardı edilen kurulum sürelerinin optimizasyonunu ele alarak, hem akademik literatürdeki önemli bir boşluğu doldurmakta hem de sanayi uygulamaları için uygulanabilir, somut çözüm önerileri sunmaktadır. Özellikle gerçek üretim verileri ile yapılan analizlerin sonuçlarının, istatistiksel olarak da desteklenmesi, çalışmanın geçerliliğini ve güvenilirliğini artırmıştır. Bu yönüyle tez, meta-sezgisel algoritmaların kurulum süresi gibi sıralamaya bağlı, karmaşık üretim problemlerinde etkinliğini ortaya koyarak hem akademisyenlere hem de sanayi profesyonellerine karar destek sistemlerinde kullanabilecekleri değerli bir model önermektedir.



MINIMIZATION OF TOTAL SETUP TIME ON A BOTTLENECK MACHINE USING METAHEURISTIC METHODS

SUMMARY

Keywords: Setup Time Optimization, Metaheuristic Algorithms, Production Scheduling, Simulated Annealing, Tire Manufacturing.

In today's highly competitive industrial environment, improving the efficiency, flexibility, and cost-effectiveness of production systems has become imperative for companies aiming to achieve sustainable growth and operational excellence. Among the critical factors influencing overall manufacturing performance, the optimization of production scheduling and setup times plays a pivotal role. Setup times, which refer to the time required for machines to switch from producing one product to another, directly impact machine utilization rates, production throughput, and operating costs. Minimizing these times ensures more stable and predictable manufacturing processes and contributes to maximizing resource efficiency. Moreover, industries seeking to adopt lean manufacturing principles and just-in-time strategies consider setup time optimization as a crucial enabler for reducing waste and improving responsiveness.

This master's thesis investigates the potential of metaheuristic optimization algorithms in minimizing setup times arising from product transitions in a tire manufacturing line. The specific problem under study emerges from the variability in setup times based on different mold and compound combinations required for each type of tire. In many manufacturing environments, such transition-related setup times are often overlooked in sequencing decisions, which results in production bottlenecks, idle times, and increased operational costs. To address this issue, a combinatorial optimization approach has been adopted, wherein the product sequence on a bottleneck machine is optimized to minimize total setup durations. Such sequencing problems are known to be NP-hard due to the factorial growth in the number of possible permutations, making heuristic and metaheuristic methods indispensable for obtaining feasible solutions in reasonable timeframes.

Three well-established metaheuristic algorithms were selected for this purpose: Tabu Search, Simulated Annealing and Particle Swarm Optimization. These methods are known for their effectiveness in solving complex, large-scale problems with vast solution spaces, such as scheduling, routing, and sequencing. Unlike conventional optimization techniques that often struggle with combinatorial complexity and local optima, metaheuristic algorithms can explore diverse regions of the solution space and converge toward near-optimal solutions within a reasonable computational time. The characteristics of each algorithm offer different strengths: Tabu Search uses adaptive memory to avoid cycles, Simulated Annealing uses a probabilistic mechanism to escape local optima, and Particle Swarm Optimization leverages swarm intelligence principles to dynamically update potential solutions based on individual and collective experiences.

Each algorithm was implemented using real-world production data obtained from a tire factory. The dataset included daily production plans, setup time matrices between different tire types based on mold and compound transitions, and historical supplier-based sequencing outcomes. The optimization objective was to find the best product sequence for each day over a ten-day horizon that minimized total setup times on the critical bottleneck machine. A comparative analysis was performed by calculating the total setup time resulting from each algorithm's output and comparing it to the baseline sequence currently used by the factory, which is based on supplier delivery order. To ensure fairness in comparison, all algorithms were calibrated using consistent stopping criteria and executed over multiple runs to capture average performance metrics.

The results of the experimental study were highly promising. The Simulated Annealing algorithm outperformed the other two approaches by achieving a 13.96% reduction in average setup time compared to the supplier-based sequence. Particle Swarm Optimization followed closely with a 12.83% improvement, and Tabu Search provided an 10.71% reduction. These improvements suggest that even small enhancements in sequencing can translate into significant operational benefits, including shorter lead times, higher throughput, and reduced labor and energy costs. Additionally, graphical analysis showed that the Simulated Annealing algorithm consistently achieved the best performance on seven out of ten days, further validating its robustness in practical scenarios.

To determine whether the observed differences between the metaheuristic approaches and the traditional method were statistically significant, the non-parametric Mann-Whitney U test was applied. The test results confirmed that the reductions in setup time obtained through the metaheuristic algorithms were statistically significant at a 95% confidence level ($p < 0.05$). Notably, although Simulated Annealing yielded the best average performance, the difference in performance between Simulated Annealing and Particle Swarm Optimization was not statistically significant, indicating comparable levels of optimization capability between these two algorithms. However, both performed significantly better than the Tabu Search algorithm, which, while effective, was more sensitive to initial conditions and neighborhood structures.

The findings of this research provide strong evidence supporting the applicability of metaheuristic algorithms in real-world production scheduling problems. In particular, their integration into decision support systems and production planning software offers manufacturers the opportunity to dynamically generate optimized schedules that adapt to changing production requirements and constraints. Furthermore, these algorithms are particularly well-suited for scenarios involving sequence-dependent setup times, complex product configurations, and high variability in demand. Given the evolving nature of Industry 4.0, the ability of metaheuristics to accommodate real-time data inputs and operate in uncertain environments makes them highly relevant for smart manufacturing systems.

In addition to the operational advantages demonstrated, this study contributes to the broader academic literature on manufacturing optimization by presenting a comparative analysis of three prominent metaheuristic techniques under a real-world case study. The research highlights not only the practical benefits of each algorithm but also the importance of selecting context-appropriate methods based on problem characteristics such as solution space size, setup time variability, and required computational efficiency. For instance, while Simulated Annealing is advantageous in navigating rugged solution landscapes, Particle Swarm Optimization may be more suitable for problems with smoother fitness functions and well-defined global optima.

In conclusion, this thesis demonstrates that metaheuristic-based scheduling solutions, particularly Simulated Annealing and Particle Swarm Optimization, can significantly improve setup time performance in tire manufacturing. Given their robust nature and adaptability, these algorithms offer valuable tools for production engineers and operations managers striving for leaner and more responsive manufacturing systems. Future work may explore hybrid algorithmic approaches or real-time applications integrated with manufacturing execution systems to further enhance decision-making and production efficiency in dynamic industrial environments. Moreover, incorporating machine learning models to predict setup times and integrating multi-objective optimization to balance trade-offs between setup time, energy consumption, and delivery deadlines may offer even more comprehensive solutions. As the manufacturing sector continues to digitize, the synergistic use of artificial intelligence and metaheuristic optimization will likely shape the next frontier in production scheduling and resource management.





1. GİRİŞ

Küresel rekabetin giderek arttığı günümüz üretim ortamında, işletmelerin temel stratejik hedefleri arasında üretim çıktılarını artırmak ve maliyetleri olabildiğince düşürmek yer almaktadır. Bu hedeflere ulaşmak için üretim süreçlerinde karşılaşılan darboğazlar ve verimsizliklerin giderilmesi büyük önem taşımaktadır (Smith, 2020). Özellikle üretim hatlarında meydana gelen kurulum süreleri, üretim kapasitesinin etkin kullanımı açısından kritik bir faktördür. Kurulum süresi, bir ürünün üretimine başlanmadan önce makinelerin gerekli ayarlamaları yapabilmesi için harcanan zaman dilimini ifade etmektedir. Kurulum sürelerinin etkili bir şekilde planlanmaması ve optimize edilmemesi, üretim süreçlerinde problemlere ve maliyette artışa yol açmaktadır (Johnson ve Lee, 2019).

Üretim planlama süreçlerinde çoğunlukla üretim sıralamaları, ürün çeşitliliği ve talep dengesi gibi unsurlar göz önünde bulundurularak yapılmakta, kurulum sürelerinin optimize edilmesi ihmal edilmektedir. Ancak, büyük ölçekli üretim tesislerinde ürün geçişlerinde oluşan uzun kurulum süreleri, üretim hatlarında önemli darboğazlara neden olmakta ve üretim performansını olumsuz etkilemektedir. Bu tür darboğazlar, üretim kapasitesinin tam anlamıyla kullanılmasını engelleyerek maliyetlerin artmasına ve müşteri taleplerine zamanında yanıt verilememesine yol açabilir (Glover ve Laguna, 1997). Örneğin, lastik üretim tesislerinde her ürün tipi için farklı kurulum sürelerinin gerekmesi ve bu sürelerin optimize edilmemesi, toplam üretim süresini uzatarak operasyonel verimsizliklere sebep olmaktadır.

Karmaşık ve büyük ölçekli üretim problemlerinde geleneksel yöntemlerin yetersiz kaldığı durumlarda meta-sezgisel algoritmalar, çözüm arayışında önemli avantajlar sunmaktadır. Meta-sezgisel yaklaşımlar, geniş çözüm alanlarını keşfetme yeteneği sayesinde optimal veya optimal çözüme yakın sonuçlar elde edilmesine olanak tanır (Dorigo ve Stützle, 2004).

Meta-sezgisel algoritmaların en büyük avantajlarından biri, karmaşık ve yüksek boyutlu problemleri çözebilme yetenekleridir. Bu algoritmalar, geleneksel yöntemlerle çözülmesi çok zor ya da imkânsız olan problemler için esnek ve güçlü bir çerçeve

sunar (Talbi, 2009). Ayrıca, yerel optimumlara takılma riskini minimize ederek daha iyi çözümler bulmaya imkân tanır. Bununla birlikte, meta-sezgisel algoritmalar genellikle rastgele arama ve keşif süreçlerine dayandığı için çözüm kalitesini artırmak amacıyla parametre ayarlamalarına duyarlıdır. Bu durum, algoritmaların performansını doğrudan etkileyebilir (Blum ve Roli, 2003).

Diğer taraftan, meta-sezgisel algoritmaların bazı durumlarda dezavantajları da bulunmaktadır. Bunların başında, yüksek hesaplama maliyeti ve işlem süresi gelmektedir. Özellikle büyük veri kümeleri ve karmaşık üretim ortamlarında, optimal çözüme ulaşmak için önemli ölçüde işlem gücü ve zaman gerekmektedir (Luke, 2013). Ayrıca, algoritmaların rastgelelik temelli yapısı, her çalıştırmada farklı sonuçlar üretilmesine yol açabilir ve bu durum istikrarlı çözümlerin elde edilmesini zorlaştırabilir. Bu nedenle, farklı meta-sezgisel yöntemlerin karşılaştırılması ve en uygun algoritmanın belirlenmesi büyük önem taşır (Boussaid, 2013).

Bu çalışmanın temel amacı, bir lastik üretim tesisindeki darboğaz oluşturan makinelerde kurulum sürelerini optimize etmek ve üretim verimliliğini artırmaktır. Bu doğrultuda, karmaşık optimizasyon problemlerini çözmek için Parçacık Sürü Optimizasyonu, Tabu Arama Algoritması ve Tavlama Benzetimi algoritmaları kullanılacaktır. Son yıllarda yapılan araştırmalar, bu tür meta-sezgisel algoritmaların, özellikle Akıllı Üretim ve Endüstri 5.0 bağlamında, çok amaçlı çizelgeleme problemlerinin çözümünde önemli avantajlar sağladığını göstermektedir (Zhang, 2025).

Parçacık Sürü Optimizasyonu, doğadaki olaylardan ilham alan bir yöntem olup, çok boyutlu arama alanlarında hızlı ve etkin çözümler üretir (Kennedy ve Eberhart, 1995). Özellikle sürekli ve ayrık optimizasyon problemlerinde yaygın olarak kullanılmakta ve küresel en iyi çözüme ulaşılmasında oldukça iyi sonuçlar vermektedir. Tabu Arama Algoritması, yerel arama tabanlı bir meta-sezgisel yöntemdir ve çözüm uzayında tekrar eden durumları önlemek için "tabu listesi" kullanır (Glover, 1989). Bu özellik, algoritmanın çok geniş bir arama alanını keşfetmesini ve yerel optimumlara takılmasını engeller. Tavlama Benzetimi ise termodinamik ilkelerden esinlenerek geliştirilmiş bir optimizasyon tekniğidir ve başlangıçta yüksek bir sıcaklık değeriyle geniş bir çözüm alanını keşfederken, zamanla bu sıcaklığı düşürerek daha iyi çözümlere odaklanır (Kirkpatrick, 1983).

Bu çalışmada, her üç yöntemin performansı kıyaslanacak ve lastik üretim tesisindeki kurulum sürelerinin minimize edilmesine yönelik en etkin yaklaşım belirlenecektir. Çalışmanın sonucunda, toplam kurulum süresinde anlamlı bir azalma sağlanması ve üretim maliyetlerinin düşürülmesi hedeflenmektedir. Ayrıca, meta-sezgisel yöntemlerin üretim planlama süreçlerine entegrasyonunun sağladığı avantajlar değerlendirilecek ve elde edilen sonuçlar literatüre katkı sağlayacaktır. Bu bağlamda, araştırma, ileri düzey optimizasyon tekniklerinin üretim sektöründeki uygulamalarına yönelik önemli bir örnek teşkil etmektedir.

1.1. Problemin Tanımı

Endüstriyel üretim süreçlerinde kurulum sürelerinin yönetimi, operasyonel verimliliğin artması açısından kritik bir öneme sahiptir. Özellikle seri üretim yapan işletmelerde, ürün değişiminden kaynaklanan geçiş süreleri, üretim çizelgelemesinin en önemli bileşenlerinden biridir. Bu sürelerin yüksek olması, üretim hattında zaman kaybına, kapasite kullanımının düşmesine ve maliyetlerin artmasına yol açmaktadır. Dolayısıyla, geçiş sürelerinin minimize edilmesi, üretim hattının daha dengeli, esnek ve maliyet etkin bir biçimde çalışmasına katkı sağlamaktadır (Zhang, Tang ve Wu, 2019).

Bu çalışmanın uygulama alanını oluşturan lastik üretim hattı, üretim sırasında farklı lastik kodlarının kalıp ve hamur kombinasyonlarına bağlı olarak değişken geçiş süreleriyle karşı karşıyadır. Her bir lastik ürünü, belirli bir kalıp ve hamur bileşimi ile üretilmekte, dolayısıyla bir üründen diğerine geçerken kurulum sürelerinde önemli farklılıklar oluşmaktadır. Bu geçişler dört temel kategoriye ayrılmaktadır:

- Aynı kalıp ve aynı hamur: 1 dakika,
- Aynı hamur, farklı kalıp: 6,5 dakika,
- Aynı kalıp, farklı hamur: 7,3 – 8,1 dakika,
- Farklı kalıp ve farklı hamur: 21,5 – 22 dakika.

Söz konusu geçiş süreleri doğrudan üretim verimliliğini etkilediğinden, üretilecek ürünlerin sıralaması bu sürelerin minimize edilmesi hedefiyle belirlenmelidir. Bu yapı, karar değişkenlerinin üretim sıralamasını temsil ettiği ve her ürünün yalnızca bir kez ziyaret edildiği bir yolun aranmasını içeren bir Hamilton Yolu Problemi olarak modellenebilir (Gary ve Johnson, 1979).

Ancak, bu tür sıralama problemleri kombinatoryal bir yapıya sahip olup, çözüm uzayının büyüklüğü nedeniyle geleneksel optimizasyon teknikleriyle çözülmesi hesaplama açısından maliyetlidir. Bu nedenle, son yıllarda Tabu Arama, Genetik Algoritma ve Tavlama Benzetimi gibi meta-sezgisel algoritmalar, bu tür problemlerde etkili çözümler sunmaktadır (Eiben ve Smith, 2015).

Bu çalışmanın temel amacı, lastik üretim hattındaki farklı ürün geçişlerine bağlı olarak ortaya çıkan kurulum sürelerini minimize edecek üretim sıralamasını belirlemek ve bu amaçla meta-sezgisel algoritmaların uygulanabilirliğini test etmektir. Gerçek üretim verilerine dayanan bu problem hem kuramsal hem de pratik açıdan yüksek düzeyde özgünlük ve uygulama potansiyeline sahiptir. Ayrıca, geliştirilen modelin üretim süreçlerine entegre edilerek işletmelerin karar destek sistemlerine katkı sağlaması hedeflenmektedir.

1.2. Literatür Araştırması

Üretim süreçlerinin verimli bir şekilde yönetilmesi, günümüz rekabetçi iş ortamında işletmelerin sürdürülebilirliklerini ve kârlılıklarını artırmak açısından kritik bir rol oynamaktadır. Özellikle üretim hatlarında meydana gelen kurulum sürelerinin optimize edilmesi hem zamandan tasarruf sağlamakta hem de operasyonel maliyetleri azaltmaktadır. Kurulum süreleri, üretim hattında bir işin tamamlanıp diğerine geçişte harcanan zamanı ifade eder ve bu sürelerin düşürülmesi üretim verimliliğine doğrudan etki etmektedir. Literatürde yapılan çalışmalar, bu tür optimizasyonların üretim süreçlerinde çok değerli iyileştirmeler sağladığını ortaya koymaktadır.

Üretim sıralamalarının doğru bir şekilde planlanması, kurulum sürelerinin optimizasyonunda kilit bir faktördür. Yanlış üretim sıralamaları, makinelerin boş kalma süresini artırarak verim kayıplarına yol açmaktadır. Örneğin, Fang, Xie ve Zhou (2017) ile Zhang, Tang ve Wu (2019) tarafından yapılan araştırmalar, doğru üretim sıralamalarının kurulum sürelerini %15'e kadar kısaltabileceğini ve bu optimizasyonun üretim süresini %12 oranında azaltabileceğini göstermektedir. Ayrıca, Chang, Xu ve Wang (2017) ile Huang ve Lin (2019) tarafından yapılan çalışmalar, üretim sıralamalarındaki iyileştirmelerin üretim kapasitesinde önemli artışlar sağladığını vurgulamaktadır.

Meta-sezgisel algoritmalar, karmaşık optimizasyon problemlerini çözmede oldukça etkili yöntemler arasında yer almaktadır. Bu algoritmalar, çözüm uzayını geniş bir

şekilde tarayarak yerel optimumlardan kaçınmayı ve daha iyi sonuçlar elde etmeyi amaçlar. Literatürde, üretim sıralamaları ve kurulum sürelerinin optimizasyonunda en sık kullanılan meta-sezgisel algoritmalar arasında Tabu Arama Tavlama Benzetimi ve Parçacık Sürü Optimizasyonu yer almaktadır.

Tabu Arama algoritması, üretim sıralamaları ve kurulum sürelerinin optimizasyonunda sıklıkla tercih edilen yöntemlerden biridir. Tang ve Wu (2020) ise bu algoritmanın üretim sıralamaları ve kurulum süreleri optimizasyonunda diğer yöntemlere kıyasla daha başarılı sonuçlar verdiğini raporlamıştır. Yıldız ve Demirtaş (2025), yaptıkları çalışmada Tabu Arama Algoritması ile %11,8'lik kurulum süresi azalması sağlamışlardır.

Parçacık Sürü Optimizasyonu (PSO), sosyal davranışlardan ilham alan ve özellikle sürekli ya da ayrık çözüm uzaylarında etkili çözümler üretebilen güçlü bir meta-sezgisel yöntemdir. Marinakis ve Marinaki (2017), PSO algoritmasını kullanarak üretim sıralamalarında önemli iyileşmeler elde etmiş ve kurulum sürelerinde anlamlı azalmalar sağlandığını göstermiştir. Sun ve Liu (2019) ise PSO ve Karınca Kolonisi Optimizasyonu gibi algoritmaların hibrit kullanımıyla daha hızlı ve etkili çözümler üretilbildiğini vurgulamıştır.

Tavlama Benzetimi (SA) algoritması da üretim sıralamaları ve kurulum sürelerinin optimizasyonunda başarılı sonuçlar sağlamaktadır. Li, Xu ve Wu (2018), Tavlama Benzetimi algoritmasının üretim çizelgeleme problemlerinde hem hız hem de çözüm kalitesi açısından öne çıktığını belirtmiştir. Chen ve Li (2024), çalışmalarında Tavlama Benzetimi algoritması kurulum süresi optimizasyonunda başarılı sonuçlar vermiştir. Zhang, Tang ve Wu (2025), yaptıkları çalışmada Tavlama Benzetimi ile %14.1, Parçacık Sürü Optimizasyonu ile %13.5 iyileştirme elde etmişlerdir.

Bu çalışma, literatürde sıkça ele alınan üretim sıralama problemlerine yönelik çözüm yaklaşımlarını derinlemesine analiz etmekle kalmayıp, aynı zamanda gerçek üretim verileri üzerinden meta-sezgisel algoritmaların karşılaştırmalı performansını değerlendirmesi açısından özgün bir katkı sunmaktadır. Tabu Arama, Tavlama Benzetimi ve Parçacık Sürü Optimizasyonu gibi yaygın olarak kullanılan yöntemler ilk kez bu kapsamda bir araya getirilerek tekil bir üretim hattı ortamında uygulanmış; elde edilen sonuçlar hem istatistiksel olarak test edilmiş hem de uygulamaya yönelik geçerliliği gösterilmiştir. Literatürdeki teorik çalışmalar genellikle simülasyon temelli yapılara odaklanırken, bu tez çalışması uygulamalı veriyle desteklenen deneysel bir

yapı sunarak, meta-sezgisel yöntemlerin gerçek endüstriyel problemlerdeki pratik etkililiğini ortaya koymuştur. Bu yönüyle çalışma hem akademik literatüre katkı sağlamakta hem de üretim planlamasında karar vericilere yönelik somut çıktılar sunmaktadır. Bu bağlamda, üretim süreçlerindeki kurulum sürelerinin optimize edilmesi, işletmelerin hem maliyetlerini azaltmakta hem de operasyonel verimliliklerini artırmaktadır. Tablo 1.1.'de literatür taramasının tablo formatı belirtilmiştir.

Tablo 1.1. Literatür Taraması 1.Tablo.

Yazarlar	Yıl	Kullanılan Yöntem	Temel Bulgular
Fang, Xie ve Zhou	2017	Üretim Sıralama Optimizasyonu	Doğru sıralama ile kurulum süresi %15, üretim süresi %12 azalabiliyor.
Zhang, Tang ve Wu	2019	Üretim Sıralama Heuristiği	Üretim sıralaması optimizasyonu ile verimlilik artışı sağlanmıştır.
Chang, Xu ve Wang	2017	Sıralama Tabanlı Planlama	Üretim kapasitesinde artış, kurulum süresi azaltımı sağlanmıştır.
Huang ve Lin	2019	Kurulum Süresi Azaltma Stratejileri	İyileştirmelerin üretim kapasitesi üzerindeki etkisi vurgulanmıştır.
Zhang, Tang ve Wu	2025	Tavlama Benzetimi, Parçacık Sürü Optimizasyonu	Tavlama Benzetimi ile %14,1; Parçacık Sürü Optimizasyonu ile %13,5 iyileştirme elde edilmiştir.
Tang ve Wu	2020	Tabu Arama Algoritması	Tabu Aramanın diğer yöntemlere göre daha başarılı olduğu gösterilmiştir.
Yıldız ve Demirtaş	2025	Tabu Arama Algoritması	Tabu Arama ile %11,8'lik kurulum süresi azalması sağlanmıştır.
Marinakos ve Marinaki	2017	Parçacık Sürü Optimizasyonu	PSO ile üretim sıralamasında iyileşme ve kurulum süresinde azalma sağlanmıştır.
Sun ve Liu	2019	Parçacık Sürü Optimizasyonu	Hibrit kullanım daha hızlı ve etkili çözümler sağlamaktadır.
Chen ve Li	2024	Tavlama Benzetimi	Tavlama Benzetimi algoritması kurulum süresi optimizasyonunda başarılı sonuçlar vermiştir.
Li, Xu ve Wu	2018	Tavlama Benzetimi	Tavlama Benzetimi algoritması hız ve çözüm kalitesi açısından öne çıkmaktadır.

Tez kapsamında ele alınan meta-sezgisel algoritmalar, yalnızca üretim sıralamalarının iyileştirilmesiyle sınırlı kalmayıp; üretim planlama, çizelgeleme, kurulum süresi optimizasyonu ve lojistik gibi geniş bir uygulama yelpazesinde başarılı bir şekilde kullanılmaktadır. Bu bağlamda, Tablo 1.2'de söz konusu algoritmaların çeşitli endüstriyel alanlardaki kullanım örnekleri, literatürden elde edilen bulgularla birlikte sistematik olarak sunulmuştur. Bu tablo, çalışmanın hem akademik literatüre dayandığını hem de farklı sektörlerdeki uygulama potansiyelini desteklediğini göstermesi açısından önemli bir referans niteliğindedir. Aynı zamanda bu tez, mevcut literatürde yeterince derinlemesine ele alınmayan sıra bağımlı kurulum süresi problemini, gerçek üretim verileri ışığında analiz ederek önemli bir boşluğu

doldurmaktadır. Özellikle kalıp ve hamur değişimlerinden kaynaklanan geçiş sürelerinin bütüncül biçimde değerlendirilmesi, çalışmanın özgün yönlerinden biridir. Üç farklı meta-sezgisel algoritmanın karşılaştırmalı olarak ele alınması ve bu yöntemlerin pratik üretim ortamındaki etkilerinin değerlendirilmesi sayesinde, yalnızca akademik katkı sunmakla kalmayıp, aynı zamanda sanayi uygulamalarına yönelik somut ve uygulanabilir çözümler de önerilmektedir.

Tablo 1.2. Literatür Taraması 2.Tablo.

Kullanılan Alan	Tavlama Benzetimi	Tabu Arama Algoritması	Parçacık Sürü Optimizasyonu
Üretim Planlama	Wang ve Tang (2016), üretim hattı kurulum sürelerini optimize ederek %16 iyileşme sağlamıştır.	Li ve Li (2017), üretim sıralamasında %15 oranında kurulum süresi azalması elde etmiştir.	Marinakis ve Marinaki (2017), üretim sıralamasında PSO ile %14 oranında iyileşme sağlamıştır.
Üretim Çizelgeleme	Li, Xu ve Wu (2018), SA ile %13 daha kısa çizelgeleme süresi elde etmişlerdir.	Tang ve Wu (2020), çizelgelemede %12 oranında verim artışı sağlamıştır.	Sun ve Liu (2019), PSO-ACO hibrit yaklaşımı ile %17 daha hızlı çözümler üretmişlerdir.
Kurulum Süresi Optimizasyonu	Chen ve Li (2024) de tavlama benzetimi algoritması ile iyileşme sağlamıştır.	Sajadi, Rezaei ve Ghaffari (2014), otomotiv sektöründe %18 verimlilik artışı sağlamıştır.	Mohapatra ve Ghosh (2021), PSO ile kurulum sürelerinde %13,5 iyileşme elde etmiştir.
Lojistik	Kirkpatrick (1983), taşıma rotalarını optimize ederek %10 daha kısa mesafe elde etmiştir.	Glover (1989), dağıtım planlamasında %11 rota iyileştirmesi sağlamıştır.	Ai ve Kachitvichyanukul (2009), PSO ile araç rotalarında %13 maliyet düşüşü sağlamıştır.



2. YÖNTEM

2.1. NP Zor Modeller

NP-zor (Non-deterministic Polynomial-time hard) problemler, çözüm bulmanın aşırı derecede zor olduğu ancak bulunan bir çözümün doğruluğunun polinomsal zamanda doğrulanabileceği problem sınıflarını ifade etmektedir. Bu tür problemler için, özellikle büyük boyutlu örneklerde, çözüm bulmak için gereken hesaplama süresi pratikte kabul edilemez düzeylere ulaşabilir. NP-zor problemler, karar problemlerinin daha geniş bir genellemesi olup, herhangi bir NP-tam problem, bir NP-zor probleme indirgenebilmektedir (Garey ve Johnson, 1979). Son dönemde yapılan çalışmalarda, bu tür problemlere yönelik geliştirilen sezgisel ve meta-sezgisel yöntemlerin, özellikle üretim planlama ve çizelgeleme uygulamalarında önemli çözüm avantajları sunduğu vurgulanmaktadır (Chen ve Li, 2024).

Üretim planlama, çizelgeleme, lojistik, kurulum süresi optimizasyonu ve rota planlama gibi birçok gerçek dünya problemi NP-zor yapıdadır. Bu durum, söz konusu problemlerde deterministik ve kesin yöntemlerle optimal çözüme ulaşmanın çoğu zaman pratikte mümkün olamayacağı anlamına gelir. Dolayısıyla, NP-zor problemlerin çözümünde sezgisel ve meta-sezgisel yöntemler tercih edilmektedir. Bu yöntemler, çözüm uzayını etkin biçimde tarayarak optimal olmasa da yüksek kaliteli çözümler üretebilmektedir.

Çalışmanın uygulama kısmında ele alınan üretim sıralaması ve kurulum süresi optimizasyonu problemleri de NP-zor sınıfında yer almaktadır. Bu nedenle, problemin çözümünde tam çözüm yöntemleri yerine, Tabu Arama, Tavlama Benzetimi ve Parçacık Sürü Optimizasyonu gibi meta-sezgisel algoritmalar kullanılmıştır. Bu algoritmalar, yüksek boyutlu kombinatorial çözüm alanlarında kabul edilebilir sürede etkili sonuçlar üretebilme yetenekleri nedeniyle tercih edilmiştir.

2.2. Kombinatorial Optimizasyon

Kombinatorial optimizasyon, sonlu ancak çok büyük bir çözüm kümesi içerisinde, belirli bir amaca yönelik en iyi çözümün bulunmasını hedefleyen matematiksel

optimizasyon alanıdır. Bu tür problemler, nesnelerin farklı kombinasyonları üzerinden en iyi çözümün aranması esasına dayanır ve sıklıkla NP-zor problemler kategorisine girer (Papadimitriou ve Steiglitz, 1998). Son yıllarda, meta-sezgisel algoritmaların otomatik olarak tasarlanması ve yapılandırılması yönündeki çalışmalar, bu alandaki çözüm yaklaşımlarının esnekliğini ve etkinliğini artırmıştır. Örneğin, Martín-Santamaría ve arkadaşları (2024), farklı problem türleri için yüksek performanslı algoritmaların otomatik olarak oluşturulmasını sağlayan bir çerçeve geliştirmiştir.

Kombinatoryal optimizasyon problemleri, özellikle üretim planlama, çizelgeleme, taşıma ve rota belirleme gibi operasyonel karar alma süreçlerinde yaygın olarak karşımıza çıkar. Örneğin, bir üretim tesisinde ürünlerin makinelere hangi sırayla atanacağı, bir lojistik firmasının teslimat rotalarının nasıl düzenleneceği veya bir montaj hattında iş sırasının nasıl belirleneceği gibi problemler, kombinatoryal optimizasyon kapsamında ele alınır. Bu tür problemlerde sıklıkla başvuru alan yaklaşımlardan biri de Hamilton Yolu Problemidir. Hamilton Yolu, tüm düğümlerin yalnızca bir kez ziyaret edildiği yolları kapsadığı için, özellikle rota optimizasyonu, görev sıralaması ve üretim akışı gibi alanlarda önemli uygulama alanı bulmuştur (Gutin ve Punnen, 2002).

Bu problemlerin çözümü, genellikle doğrusal olmayan, ayrık ve yüksek boyutlu bir çözüm uzayı gerektirdiği için geleneksel deterministik algoritmalar yetersiz kalmaktadır. Bu nedenle, çözüm süresini ve kalitesini dengeleyebilecek sezgisel ve meta-sezgisel algoritmalar sıklıkla tercih edilmektedir. Tabu Arama, Tavlama Benzetimi ve Parçacık Sürü Optimizasyonu gibi yöntemler, kombinatoryal optimizasyon problemlerinde etkin çözümler üretmek amacıyla geliştirilmiş olup, yerel minimumlardan kaçınma ve geniş çözüm alanlarını keşfetme kabiliyetleriyle öne çıkmaktadır.

Bu çalışmada da üretim sıralamalarının ve kurulum sürelerinin optimizasyonu, tipik bir kombinatoryal optimizasyon problemi olarak ele alınmış ve çözümde meta-sezgisel yaklaşımlar kullanılarak başarılı sonuçlar elde edilmiştir.

2.3. Hamilton Yolu Problemi

Hamilton Yolu Problemi (Hamiltonian Path Problem), bir graf üzerindeki tüm düğümleri her birine yalnızca bir kez uğrayarak ziyaret eden bir yolun bulunup bulunamayacağını araştıran klasik bir kombinatoryal optimizasyon problemidir. Eğer

bu yol başladığı düğümde de sona eriyorsa, bu yol "Hamilton Çevresi" adını alır (Gutin ve Punnen, 2002). Güncel araştırmalar, Hamilton Yolu ve benzeri NP-zor problemler için geliştirilen sezgisel yaklaşımların, büyük ölçekli ağlarda başarılı sonuçlar verdiğini göstermektedir (Yıldız ve Demirtaş, 2025).

Hamilton yolu problemi, NP-tam sınıflarında yer almakta olup, çözümü için gereken hesaplama süreleri problemlerin boyutlarına bağlı şekilde üstel olarak artmaktadır. Bu nedenle, büyük ölçekli problemlerde deterministik yöntemler yerine meta-sezgisel algoritmaların kullanılması tercih edilmektedir. Özellikle Gezgin Satıcı Problemi gibi birçok gerçek dünya optimizasyon sorunu Hamilton yolu probleminin özel bir durumu olarak değerlendirilmektedir.

Üretim planlama, çizelgeleme ve rota optimizasyonu gibi alanlarda, bir graf üzerindeki tüm düğümlerin yalnızca bir kez ziyaret edilmesini amaçlayan Hamilton Yolu Problemi yaygın olarak kullanılmakta olup, özellikle seyahat eden satıcı probleminin ve kombinatorial rota planlamalarının temelini oluşturmaktadır (Gutin ve Punnen, 2002). Özellikle üretim sıralamalarında, bir işin tamamlanmasından sonra bir diğer işe geçişin optimize edilmesi Hamilton yolu mantığı ile modellenenebilir. Bu çalışmada da üretim sürecindeki kurulum sürelerinin optimizasyonu, düğümler arası geçişlerin minimize edilmesi amacıyla Hamilton yolu yaklaşımı çerçevesinde modellenmiş ve meta-sezgisel algoritmalarla çözüm sağlanmıştır.

2.4. Problemin Matematiksel Modeli

Aşağıda problemin matematiksel modelinin karar değişkenleri belirtilmiştir.

X_{ij} : i 'inci ve j 'inci ürünlerin sırasıyla üretileceğini belirten ikili karar değişkeni. Eğer i 'inci ürün j 'nci ürünün öncesinde üretiliyorsa, $X_{ij} = 1$ aksi taktirde $X_{ij} = 0$ (Zhang et al., 2019).

B_i : O gün 1 tane birinci ürün olabilir ve olmalı. Eğer B_i ilk ürünse $B_i=1$ diğer durumda 0 değerini alır.

S_i : O gün 1 tane sonuncu ürün olabilir ve olmalı. Eğer S_i son ürünse $S_i=1$ diğer durumda 0 değerini alır.

2.4.1. Problemin matematiksel modelin parametreleri

Aşağıda problemin matematiksel modelin parametreleri belirtilmiştir.

C_{ij} : i 'inci ve j 'inci ürünleri geçişi için gerekli kurulum süresi.

n : O gün üretilecek ürün sayısı.

2.4.2. Problemin matematiksel modelin amaç fonksiyonu

Matematiksel modelin amacı, toplam kurulum süresini minimize etmektir. Bu amaç doğrultusunda, aşağıdaki hedef fonksiyonu kullanılır (2.1).

$$\text{Minimize } Z = \sum_{i=1}^n \sum_{j=1, j \neq i}^n C_{ij} X_{ij} \quad (2.1)$$

Bu fonksiyon, ürünler arasındaki sıralama değişikçe ortaya çıkan toplam kurulum süresinin minimize edilmesini hedefler. (Glover ve Laguna, 1997).

2.4.3. Problemin matematiksel modelin açıklanması

Üretim sıralamaları belirli kısıtlarla sınırlıdır. Bu kısıtlar, sıralamanın mantıklı olmasını sağlar. Bu kısıtlar aşağıdaki şekilde ifade edilebilir:

Birinci olmayan ürüne illa bir yerden gelinmeli (2.2).

$$\sum_{i=1, i \neq j}^n x_{ij} = (1 - B_j) \quad \forall j \in \{1, 2, \dots, n\} \quad (2.2)$$

O gün sadece 1 tane birinci ürün olabilir (2.3).

$$\sum_{i=1}^n B_i = 1 \quad (2.3)$$

Sonuncu olmayan üründen bir yere gidilmelidir (2.4).

$$\sum_{j=1}^n x_{ij} = (1 - S_i) \quad \forall i \in \{1, 2, \dots, n\} \quad (2.4)$$

O gün sadece 1 tane sonuncu ürün olabilir (2.5)

$$\sum_{i=1}^n S_i = 1 \quad (2.5)$$

Bu çözüm sonucunda Hamilton yolunda alt turlar yoksa optimum çözüm bulunmuştur. Eğer alt turlar mevcutsa bu durumda SEC (Alt tur elimine kısıtları) eklenmelidir

3. UYGULAMA

Bu çalışmada, lastik üretim hattındaki geçiş sürelerinin minimize edilmesi amacıyla yapılan optimizasyon süreci ele alınmıştır. Lastik üretimi sırasında, bir lastik ürününden diğerine geçişte farklı kalıp ve hamur kombinasyonlarına bağlı olarak hazırlık sürelerinde değişiklikler meydana gelmektedir. Bu sürelerin doğru şekilde yönetilmesi, üretim verimliliğini yükseltmek ve maliyetleri düşürebilmek için önemli bir yere sahiptir. Nitekim son yıllarda yapılan araştırmalar, geçiş sürelerinin doğru optimizasyonunun üretim performansı üzerinde doğrudan etkili olduğunu göstermektedir (Kaya ve Şahin, 2023).

3.1. Veri Toplama ve İlk Durum Analizi

Çözüm sürecine başlamadan önce, lastik üretim sürecine ait veriler toplanmış ve analiz edilmiştir. 10 günlük üretim verileri üzerinden yapılan değerlendirme, her bir lastik kodunun hangi kalıp ve hamur türü ile üretileceği hakkında bilgi sağlamaktadır. Tablo3.1'deki veriler, üretim hattındaki makinelerin sıralama ve geçişlerini belirlemek için kullanılmıştır. İlgili günlerde hangi lastik koduyla ilgili üretim yapılacaksa “√” işareti ile belirtilmiştir. Üretim sürecinin her aşamasında, her bir lastik türü için yapılan kurulum süreleri ve geçişler dikkate alınarak optimizasyon sürecinin parametreleri oluşturulmuştur.

3.2. Tedarik Zinciri Departmanından Gelen Üretim Bilgileri

Üretim hattında gerçekleştirilen her ürün değişiminde, kullanılan kalıp ve hamur türlerindeki farklılıklar nedeniyle hazırlık sürelerinde değişkenlik gözlemlenmektedir. Bu çalışmada, her bir lastik türü arasındaki geçişlerde ortaya çıkan hazırlık süreleri dikkate alınarak, geçiş matrisine dayalı bir yapı oluşturulmuştur. Problemde ilk öncelikle tedarik zincirinden gelen üretim verilerine göre analiz yapılacaktır. Lastik firmasından alınan 10 günlük üretim verileri Tablo3.1'de belirtilmiştir.

Tablo 3.1. 10 Günlük Üretim Planı.

Lastik Kodu	Kalıp No'su	Hamur No'su	İlgili Günler									
			1	2	3	4	5	6	7	8	9	10
S597	RT597M	KT9..	√	√	√	√	√	√	√	√	√	√
S170	RT172	K43..	√	√	√	√	√	√	√	√	√	√
Q133	RT733K	K35..	√	√	√							
Q171	RT172	K43..		√	√	√	√	√	√	√	√	√
Q155	RT172	K43..	√									√
A826	LT826K	N36..	√	√	√	√	√	√	√	√	√	√
Q404	RT413C	K83..										√
Q992	RT992E	K16..	√	√	√	√	√	√	√	√	√	√
S162	RT161L	KT9..										
Q591	RT591B	KT9..										
Q401	RT404T	K83 ..	√	√	√	√	√	√	√	√	√	√
Q593	RT591B	KT9..										
Q172	RT172	K43..	√									
Q156	RT172	K43..	√									
Q359	RT359A	KT9..										
Q357	RT254O	KT9..	√	√	√	√	√	√	√	√	√	√
Q355	RT254O	KT9..										
Q352	RT352B	KT9..										
Q354	RT254O	KT9..	√	√	√	√	√	√	√			
A788	RT787A	KT9..										
A729	LT729J	K17..	√	√	√	√	√	√	√	√	√	√
Q362	RT327B	K93..										
Q365	RT365A	K93..										
Q368	RT315P	K93..	√	√	√	√	√	√	√	√	√	√
Q471	RT471B	K64..	√	√	√	√	√	√	√	√	√	√
Q790	RT790A	KT9..							√	√	√	√
Q793	RT793	KT9..										
Q791	RT791A	KT9..										
Q795	LT796B	KT9..			√	√	√	√	√	√	√	√
Q798	RT798	KT9..	√	√	√	√	√	√	√	√	√	√
Q796	LT796B	KT9..										
Q799	LT799B	KT9..										
A877	RT877J	K17..	√	√	√	√	√	√	√	√	√	√
Q370	RT320D	K93..									√	√
Q530	RT529	KX4..	√	√								
A827	LT827H	N36..						√	√	√	√	√
A821	LT921	N36..										
A828	LT828E	N36..				√	√	√	√	√	√	√
U440	RT440B	K35..										
U439	RT440B	K35..					√	√				
A850	RT850C	K84..	√	√								
A726	LT726C	K54..										
A951	RT951H	K93..	√	√	√	√	√	√	√	√	√	√
U496	LT496	KT9..	√	√	√	√	√	√	√	√	√	√
A914	LT914A	KT9..	√	√	√	√	√	√	√	√	√	√
A744	RT744	KT9..	√	√	√	√	√	√	√	√	√	√
A632	RT632A	KT9..	√	√	√	√	√	√	√	√	√	√
A946	LT946A	K15..	√	√	√	√	√	√	√	√	√	√
A988	LT988E	K93..	√	√	√							
A875	LT875B	K54..	√	√	√						√	√
A734	LT731C	K67..	√	√	√	√	√					
A629	LT626	K67 ..						√	√	√	√	√
A741	RT741B	KT9..									√	√
A945	LT945A	KT9..										
A944	LT944E	KT9..	√	√	√	√	√	√	√	√	√	√
A784	RT784A	KT9..										
A746	LT739C	KT9..	√	√	√	√	√	√	√	√	√	√
A769	RT769A	KT9..	√	√	√	√						

Tablo3.1. (Devamı) 10 Günlük Üretim Planı.

Lastik Kodu	Kalıp No'su	Hamur No'su	İlgili Günler									
			1	2	3	4	5	6	7	8	9	10
A787	RT787A	KT9..	√	√	√	√	√	√	√	√	√	√
A772	RT771A	K15..				√	√	√	√	√	√	√
U774	RT774A	K15..										
A778	RT778B	K15..	√	√	√	√	√	√	√	√	√	√
A852	RT852K	K93			√	√	√	√	√			
A969	RT969F	K43..	√	√	√	√	√	√	√	√	√	√
A873	LT873C	K93..	√	√	√							
Q560	RT560G	KT9..										
Q561	RT561G	KT9..										
A922	RT923B	K17..	√	√	√	√	√	√	√	√	√	√
Q375	RT375C	KX4..		√	√	√	√	√	√			
Q376	RT376D	KX4..										√
A771	RT771A	K15..										
A738	RT781A	K15..	√	√	√	√	√	√	√	√	√	√
A474	LT988E	K93..			√	√	√	√	√	√	√	√
A475	LT875B	K54..				√	√	√	√	√	√	√
A473	LT873C	K93..			√	√	√	√	√	√	√	√

Hazırlık süreleri, dört temel kategoriye ayrılmaktadır.

- Eğer lastikler arasındaki değişimde aynı hamur ve aynı kalıba geçiş olursa 1 dakika hazırlık süresi bulunmaktadır.
- Eğer lastikler arasındaki değişimde aynı hamur ve farklı kalıp kullanılacaksa 6,5 dk hazırlık süresi bulunmaktadır.
- Eğer lastik değişimde aynı kalıp farklı hamur kullanılacaksa hazırlık süresi 7,3 ile 8,1 dk arasında değişmektedir.
- Eğer lastik değişimde farklı kalıp ve farklı hamur kullanılacaksa hazırlık süresi 21,5 ile 22 dk arasında değişmektedir.

Bu kategorilere göre hesaplanan hazırlık süreleri, her bir lastik türü için geçişlerin yapıldığı günler doğrultusunda zaman dilimlerine yerleştirilmiştir. Böylelikle, üretim sıralamaları sırasında oluşacak geçişlerin toplam etkisi önceden tahmin edilerek optimizasyon algoritmalarının giriş verisi haline getirilmiştir. Farklı kombinasyonlardaki geçiş türleri üretim hattının toplam verimliliğini önemli ölçüde etkilediğinden, bu sürelerin dikkatli bir şekilde modellenmesi gereklidir. Bu nedenle, geçiş matrisinin hazırlanmasında her bir lastik türü arasındaki olası kombinasyonlar sistematik biçimde değerlendirilmiştir. Söz konusu sınıflandırma sayesinde, üretim planlaması sürecinde hazırlık sürelerinin minimize edilmesine yönelik daha isabetli kararlar alınması mümkün hale gelmiştir. Aşağıdaki Tablo3.2’de hamurlar arası geçiş süreleri belirtilmiştir.

Tablo 3.2. Hamurlar Arası Geçiş Süresi.

Ham	N36	K67	K15	K83	KT9	K17	K43	K93	K35	KX4	K54	K84	K64	K16	K83	K76	K67	K56
N36..	0.0	21.5	21.5	8.1	8.1	8.1	8.1	7.7	21.8	21.5	21.5	21.5	21.5	8.1	21.8	21.5	21.5	21.5
K67..	21.5	0.0	7.7	7.7	7.7	7.7	7.7	7.7	7.3	7.7	7.7	7.7	7.7	7.7	7.7	7.7	21.8	7.7
K15..	21.5	21.8	0.0	21.8	21.8	21.8	21.8	22.0	21.8	21.8	21.8	21.8	21.8	21.8	21.8	21.8	21.8	21.8
K83..	21.8	21.8	21.8	0.0	7.7	7.7	7.7	22.0	21.8	21.8	7.7	7.7	21.8	7.7	21.8	21.8	21.8	21.8
KT9..	8.1	21.8	21.8	7.7	0.0	7.7	7.7	22.0	21.8	21.8	21.8	21.8	21.8	7.7	21.8	21.8	21.8	21.8
K17..	8.1	21.8	21.8	7.7	7.7	0.0	7.7	22.0	21.8	21.8	21.8	21.8	21.8	7.7	21.8	21.8	21.8	21.8
K43..	8.1	21.8	21.8	7.7	7.7	7.7	0.0	22.0	21.8	21.8	21.8	21.8	21.8	7.7	21.8	21.8	21.8	21.8
K93..	21.8	21.8	21.8	7.5	7.5	7.5	7.5	0.0	22.0	22.0	22.0	22.0	22.0	7.5	22.0	22.0	22.0	22.0
K35..	7.3	7.3	7.3	7.3	7.3	7.3	7.3	7.3	0.0	7.3	7.3	7.3	7.3	7.3	7.3	7.3	21.8	7.3
KX4..	21.5	7.7	7.7	7.7	7.7	7.7	7.7	7.7	7.3	0.0	7.7	7.7	7.7	7.7	7.7	7.7	21.8	7.7
K54..	21.5	7.7	7.7	7.7	7.7	7.7	7.7	7.7	7.3	7.7	0.0	7.7	7.7	7.7	7.7	7.7	21.8	7.7
K84..	21.5	7.7	7.7	7.7	7.7	7.7	7.7	7.7	7.3	7.7	7.7	0.0	7.7	7.7	7.7	7.7	21.8	7.7
K64..	21.5	7.7	7.7	7.7	7.7	7.7	7.7	7.7	7.3	7.7	7.7	7.7	0.0	7.7	7.7	7.7	21.8	7.7
K16..	8.1	21.8	21.8	7.7	7.7	7.7	7.7	22.0	21.8	21.8	21.8	21.8	21.8	0.0	21.8	21.8	21.8	21.8
K83	7.7	7.7	7.7	7.7	7.7	7.7	7.7	7.7	7.3	7.7	7.7	7.7	7.7	7.7	0.0	7.7	21.8	7.7
K76..	21.5	7.7	21.8	21.8	21.8	21.8	21.8	22.0	21.8	21.8	21.8	21.8	21.8	21.8	21.8	0.0	21.8	7.7
K67	21.5	7.7	21.8	21.8	21.8	21.8	21.8	22.0	21.8	21.8	21.8	21.8	21.8	21.8	21.8	21.8	0.0	21.8
K56..	21.5	7.7	21.8	21.8	21.8	21.8	21.8	22.0	21.8	21.8	21.8	21.8	21.8	21.8	21.8	7.7	21.8	0.0

3.3. Tabu Arama Algoritması Adımları

Tabu Arama algoritması, mevcut çözümün komşuluk yapısı üzerinden yeni çözümler üretmek çalışır. Algoritmanın temel gücü, daha önce ziyaret edilen çözümleri tabu listesiyle dışlayarak çözüm uzayında tekrarları önlemesidir. Algoritmanın uygulaması aşağıdaki temel adımlardan oluşmaktadır. Bu algoritma, özellikle kombinatoriyal yapıya sahip büyük boyutlu problemler için etkili bir arama stratejisi sunmaktadır (Yılmaz ve Aydın, 2022).

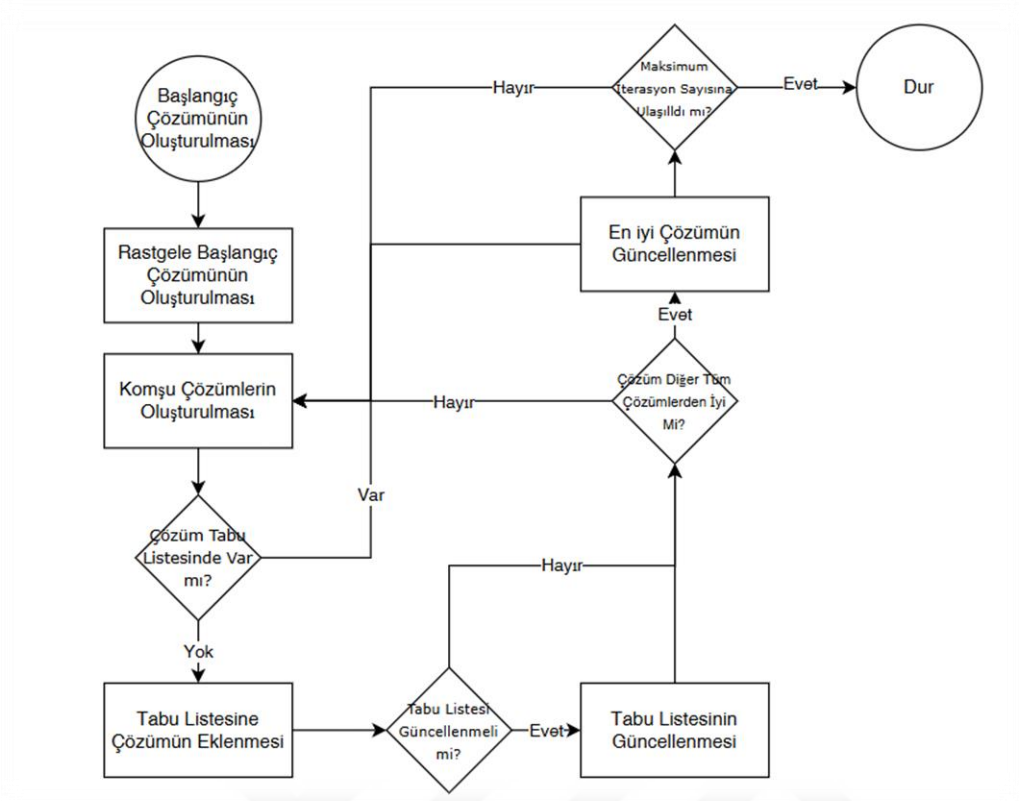
- Başlangıçta, üretim sıralaması rastgele oluşturulmuş ve ilk çözüm olarak kabul edilmiştir.
- Her iterasyonda, bu çözümün komşuları oluşturulmuş; yani sıralamadaki bazı işlemler yer değiştirilmiştir.
- Komşular arasında en iyi çözüm seçilmiş, tabu listesine girip girmediği kontrol edilmiştir.
- Eğer bu çözüm daha önce tabu listesinde yer almıyorsa ve daha iyiye kabul edilmiştir.
- Tabu listesi belirli bir boyuta ulaştığında en eski eleman listeden çıkarılmıştır.

Aşağıdaki Tablo3.3’de tabu arama algoritmasının kaba kodu belirtilmiştir.

Tablo 3.3. Tabu Arama Algoritması Kaba Kodu.

Tabu Algoritması Adımları
1. Rastgele bir başlangıç çözümü oluştur $\rightarrow$ mevcut_çözüm
2. Bu çözümü en iyi çözüm olarak kaydet $\rightarrow$ en_iyi_çözüm
3. Tabu listesini oluştur (başlangıçta boştur)
4. İterasyon sayacını sıfırla $\rightarrow$ iterasyon = 0
5. Maksimum iterasyon sayısına ulaşılan kadar döngüye gir: a. Mevcut çözümün komşu çözümlerini üret $\rightarrow$ komşular b. Tabu listesinde olmayan en iyi komşu çözümü seç $\rightarrow$ yeni_çözüm c. Mevcut çözümü güncelle $\rightarrow$ mevcut_çözüm = yeni_çözüm d. Eğer yeni çözüm, şimdiye kadarki en iyi çözümdense: $\rightarrow$ en_iyi_çözüm = yeni_çözüm e. Yapılan hamleyi tabu listesine ekle f. Tabu süresi dolmuş hamleleri listeden çıkar g. İterasyon sayacını bir artır $\rightarrow$ iterasyon += 1
6. Döngü sona erdiğinde, bulunan en iyi çözümü döndür

Şekil3.1’de ise tabu arama algoritmasının uygulama adımlarının şeması belirtilmiştir.



Şekil 3.1. Tabu Arama Algoritması Uygulama Adımları.

3.4. Tavlama Benzetimi Algoritması Adımları

Tavlama Benzetimi, fiziksel tavlama sürecinin yapılışından esinlenerek geliştirilmiş ve başlangıçta yüksek sıcaklıkta daha kötü çözümlere izin vererek global minimuma ulaşmayı amaçlayan bir algoritmadır. Zamanla sıcaklık düşürülür ve sistem kararlı hale getirilir.

Algoritmanın uygulama adımları şunlardır:

- Rastgele bir başlangıç çözümü oluşturulmuştur.
- Bu çözüm üzerinden komşu çözüm türetilmiş, yeni çözüm daha iyiye kabul edilmiştir.
- Eğer daha kötü bir çözümse, belirli bir olasılık dahilinde kabul edilmiştir. Bu olasılık, sıcaklık değişkeni ile ilişkilidir.
- Sıcaklık her adımda azaltılmış ve belirli bir eşik değerinin altına düştüğünde algoritma sonlandırılmıştır.

Aşağıda Tablo3.4'te tavlama benzetiminin kaba kodu belirtilmiştir.

Tablo 3.4. Tavlama Benzetimi Algoritması Kaba Kodu.

Tavlama Benzetimi Algoritması Adımları

Başla

Rastgele bir başlangıç çözümü oluştur

Başlangıç sıcaklığını belirle (T)

Soğuma oranını belirle (α)

En iyi çözüm = Başlangıç çözümü

İken ($T > \text{Minimum sıcaklık}$)

 Komşu çözüm üret

 Uygunluk farkını hesapla (ΔE)

 Eğer (Yeni çözüm daha iyiye)

 Mevcut çözümü güncelle

 Eğer (Yeni çözüm en iyi çözümden iyiye)

 En iyi çözümü güncelle

 Bitti

Aksi halde

 Belirli bir olasılıkla daha kötü çözümü kabul et

 Olasılık = $\exp(-\Delta E / T)$

 Bitti

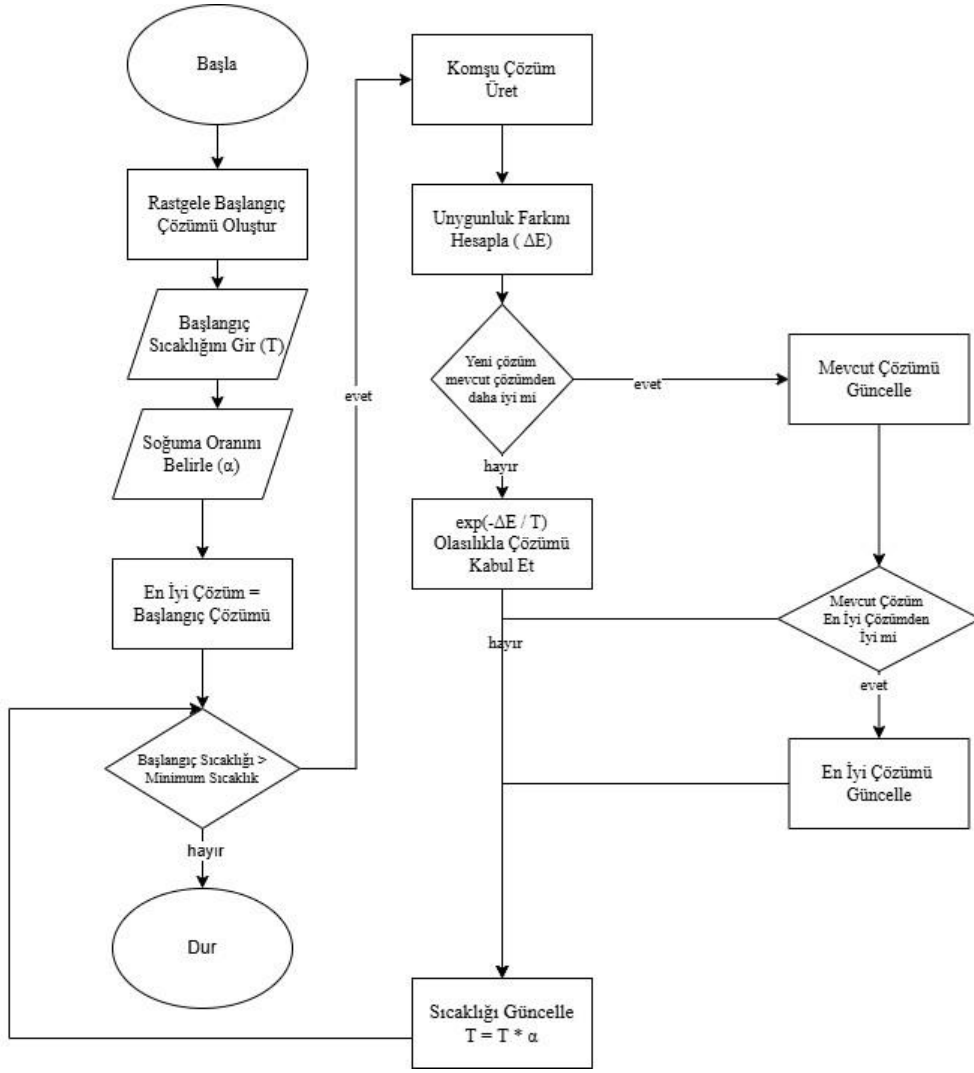
 Sıcaklığı güncelle: $T = T * \alpha$

Bitti

En iyi çözümü döndür

Son

Şekil3.2’de ise tavlama benzetiminin uygulama adımlarının olduğu akış şeması belirtilmiştir.



Şekil 3.2. Tavlama Benzetimi Algoritması Uygulama Adımları.

3.5. Parçacık Sürü Optimizasyonu Algoritması Adımları

Parçacık Sürü Optimizasyonu, her bir parçacığın kendi geçmiş deneyiminden (pBest) ve sürünün en iyi deneyiminden (gBest) yararlanarak çözüm uzayında hareket etmesini sağlayan bir algoritmadır. Her parçacık, belirli bir çözümü temsil eder ve iterasyonlar boyunca bu çözümü geliştirir. Uygulama süreci şu adımlardan oluşmaktadır:

- Başlangıçta parçacıklar (çözümler) rastgele konumlandırılmış ve hızları belirlenmiştir.
- Her parçacığın uygunluk değeri hesaplanmış, en iyi bireysel çözüm (pBest) ve en iyi genel çözüm (gBest) belirlenmiştir.
- Hız ve konum değerleri her iterasyonda güncellenerek yeni çözümler oluşturulmuştur.

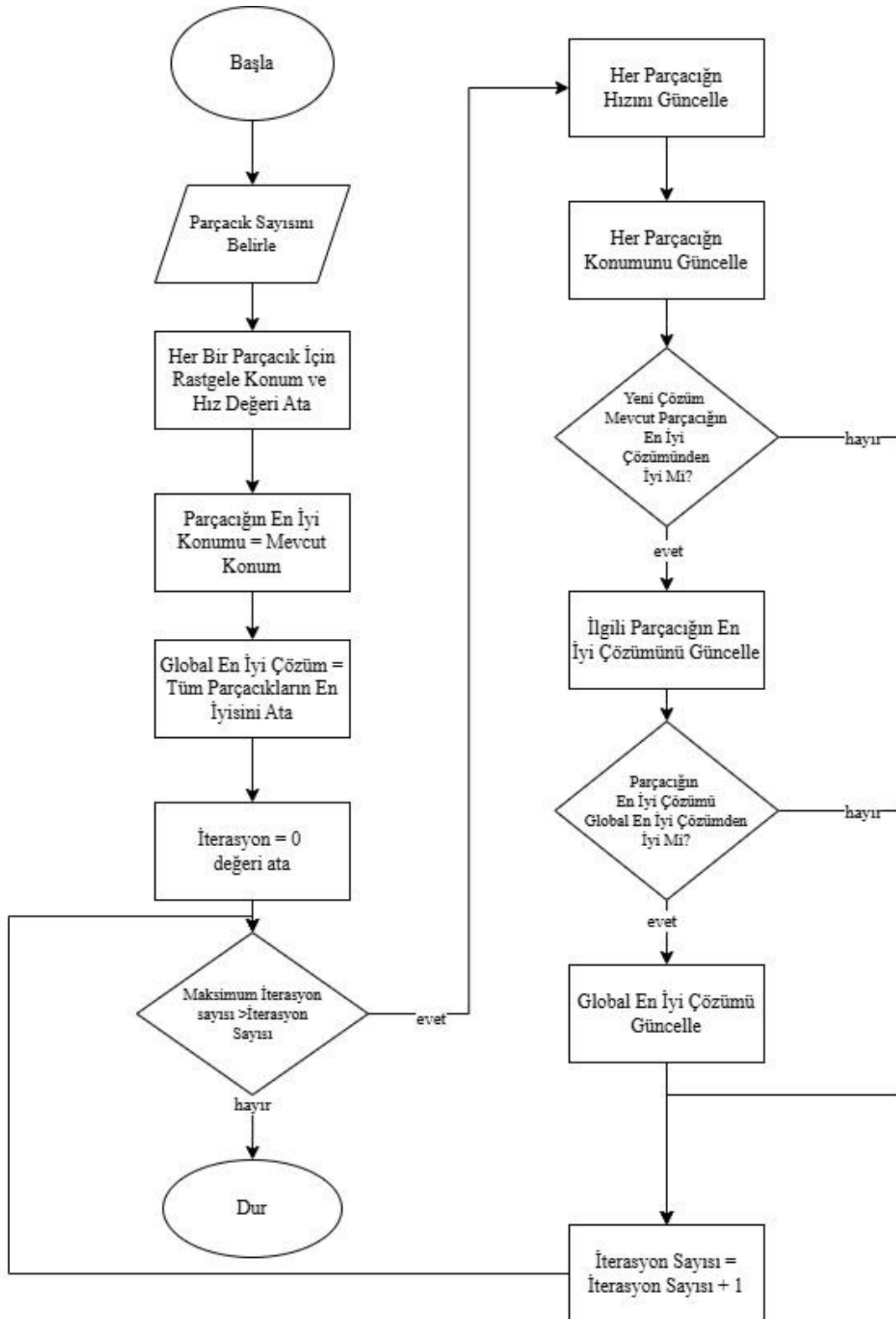
- Belirli bir iterasyon sayısına ulaşıldığında algoritma sonlandırılmıştır.

Tablo 3.5'te parçacık sürü optimizasyonunun kaba kodu belirtilmiştir.

Tablo 3.5. Parçacık Sürü Optimizasyonu Kaba Kodu.

Parçacık Sürü Optimizasyonu Adımları
Başla
Parçacık sayısını belirle
Her parçacık için:
Rastgele konum ve hız ata
pBest = mevcut konum
Bitti
gBest = tüm pBest'lerin en iyisi
İterasyon = 0
İken (İterasyon < Maksimum İterasyon Sayısı)
Her parçacık için:
Hızı güncelle:
$Yeni\ hız = w * eski\ hız + c1 * r1 * (pBest - mevcut\ konum) + c2 * r2 * (gBest - mevcut\ konum)$
Konumu güncelle:
$Yeni\ konum = eski\ konum + yeni\ hız$
Yeni çözümün uygunluğunu hesapla
Eğer (Yeni çözüm pBest'ten daha iyiyse)
pBest = yeni çözüm
Bitti
Eğer (pBest, gBest'ten daha iyiyse)
gBest = pBest
Bitti
Bitti
İterasyon = İterasyon + 1
Bitti
gBest çözümünü döndür
Son

Şekil3.3’de parçacık sürü optimizasyonunun uygulama adımları belirtilmiştir.



Şekil 3.3. Parçacık Sürü Optimizasyonu Uygulama Adımları.

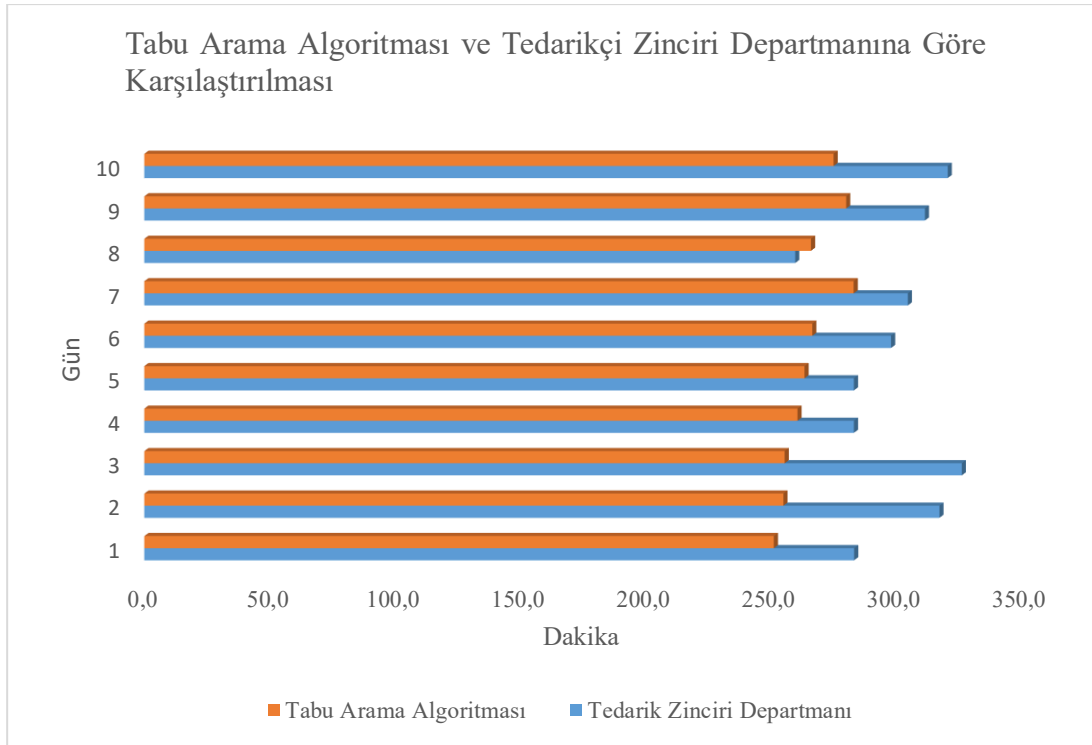
4. TARTIŞMA

4.1. Tartışma Değerlendirme

Bu çalışmada, lastik üretim hattında kurulum (geçiş) sürelerini minimize etmek amacıyla Tabu Arama Algoritması, Tavlama Benzetimi ve Parçacık Sürü Optimizasyonu olmak üzere üç farklı meta-sezgisel algoritma uygulanmış ve bu yöntemlerin performansları tedarikçiye göre belirlenmiş geleneksel sıralama ile karşılaştırılmıştır. Uygulama sonuçları 10 gün boyunca elde edilen toplam geçiş süreleri üzerinden değerlendirilmiştir.

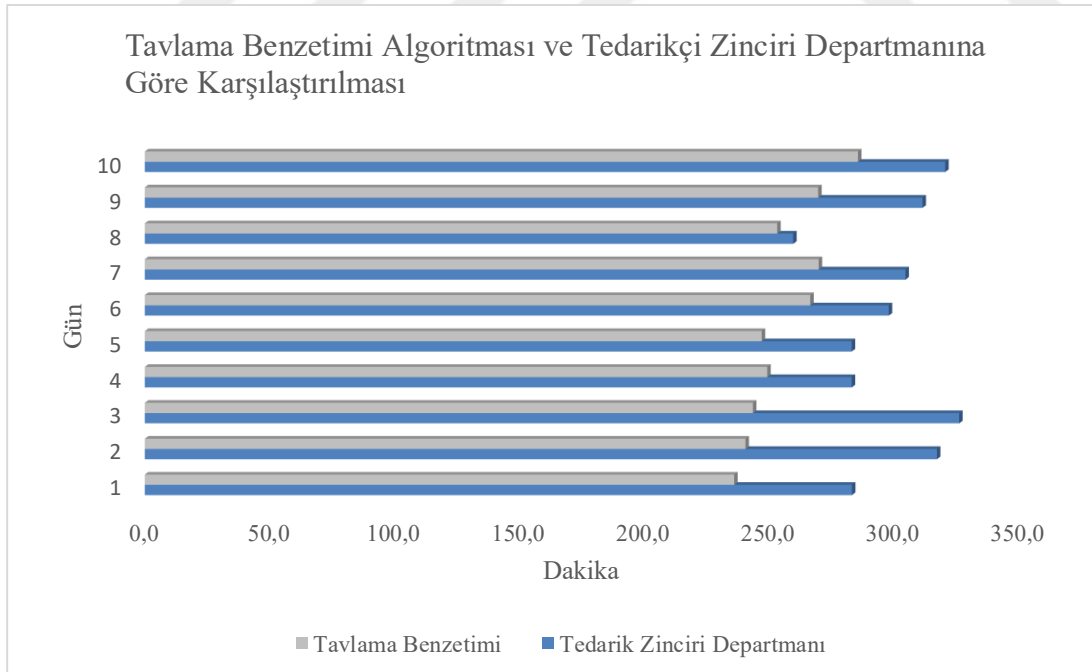
Problemin çözümüne yönelik yazılım, Intel® Core™ i7-12700H işlemcili, 32 GB RAM'li, 1 TB SSD depolama alanına ve 4 GB NVIDIA RTX3050 ekran kartına sahip bir bilgisayarda geliştirilmiştir. Geliştirme süreci boyunca Python programlama dili kullanılmış ve kodlama işlemleri Jupyter Notebook ortamında gerçekleştirilmiştir. Uygulama kapsamında, veri işlemleri, optimizasyon hesaplamaları ve istatistiksel testlerin uygulanması için başta NumPy, Matplotlib, pandas, random, math, copy, itertools ve scipy.stats olmak üzere çeşitli kütüphanelerden faydalanılmıştır. Bu altyapı sayesinde, meta-sezgisel algoritmaların üretim sıralaması problemlerine uygulanabilirliği sağlanmış ve karşılaştırmalı analizler gerçekleştirilmiştir.

Sonuçlar incelendiğinde, her üç algoritmanın da tedarikçiye göre belirlenen sıralamaya kıyasla toplam geçiş süresini önemli ölçüde azalttığı gözlemlenmiştir. En düşük geçiş süresini veren algoritmalar çoğunlukla Tavlama Benzetimi olurken, bazı günlerde Parçacık Sürü Optimizasyonu ve Tabu Arama Algoritması da yakın sonuçlar sunmuştur. Tabu arama algoritmasının tedarik zinciri departmanının kullandığı sıralama göre karşılaştırması aşağıdaki Şekil4.1'de belirtilmiştir.



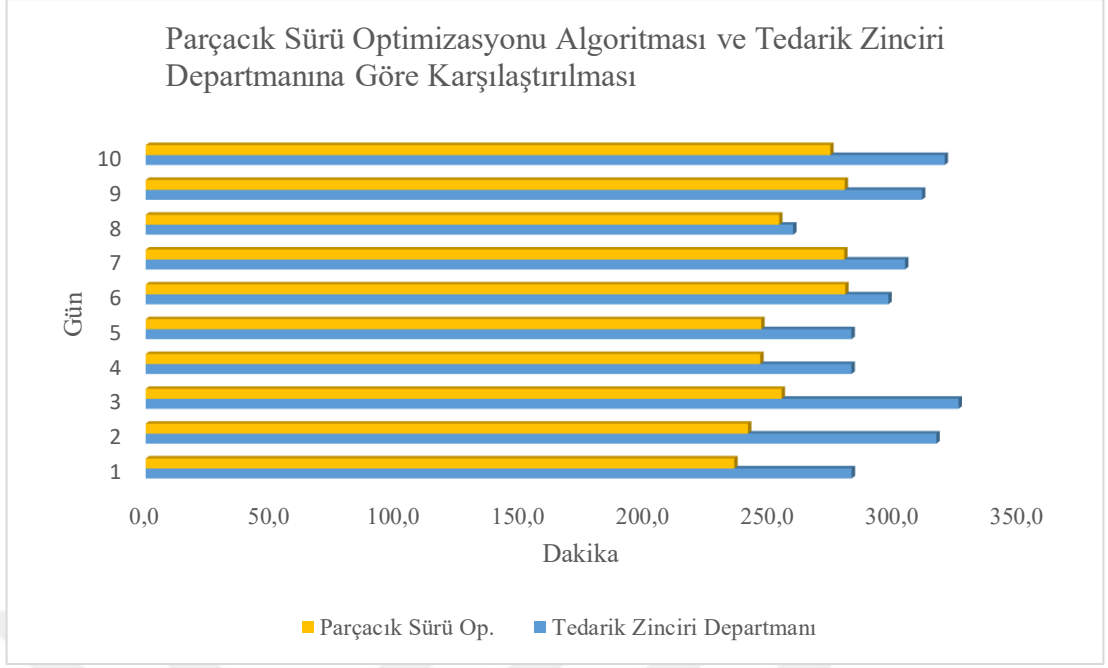
Şekil 4.1. Tabu Arama Algoritması ve Tedarik Zinciri Departmanı Karşılaştırması.

Tavlama benzetimi algoritmasının tedarik zinciri departmanının kullandığı sıralama göre karşılaştırması aşağıdaki Şekil4.2’de belirtilmiştir.



Şekil 4.2. Tavlama Benzetimi ve Tedarikçi Zinciri Karşılaştırması.

Parçacık sürü optimizasyonu algoritmasının tedarik zinciri departmanının kullandığı sıralama göre karşılaştırması aşağıdaki Şekil4.3’te belirtilmiştir.



Şekil 4.3. Parçacık Sürü Optimizasyonu ve Tedarikçi Zinciri Karşılaştırması.

Özellikle ilk beş gün için Tavlama Benzetimi yöntemi her gün en düşük kurulum süresini üretmiş, bu da algoritmanın kısa vadeli çözüm kalitesi açısından oldukça etkili olduğunu göstermektedir. SA'nın stokastik doğası ve olasılıksal kabul kriterleri, yerel minimumlara takılmadan daha iyi çözümler bulmasına olanak tanımaktadır (Kirkpatrick et al., 1983). Benzer şekilde, PSO da özellikle 1., 2. ve 4. günlerde SA'ya oldukça yakın sonuçlar vermiştir. Ancak 6., 7. ve 9. günlerde PSO'nun diğer algoritmalarla göre daha yüksek değerler ürettiği ve bu günlerde en kötü sonuçları verdiği görülmüştür. Bu durum, PSO'nun çözüm uzayındaki dengesini kaybedip erken yakınsamaya maruz kaldığını göstermektedir (Eberhart ve Kennedy, 1995).

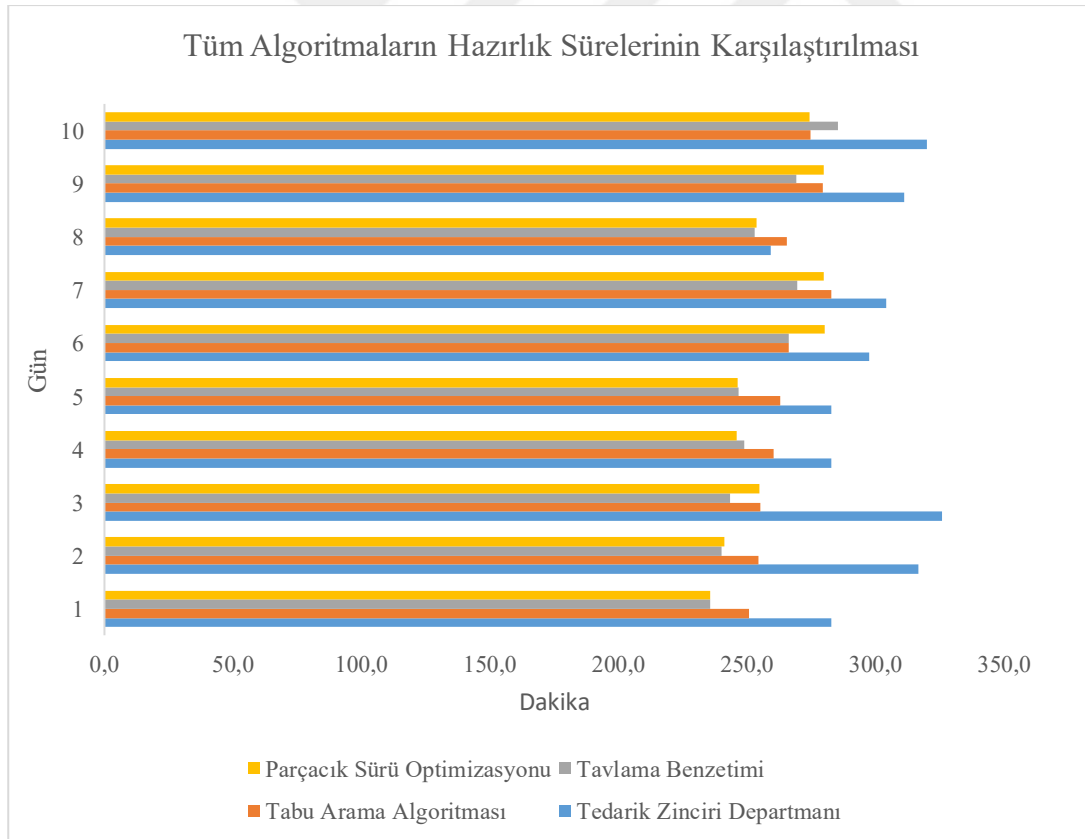
TS yöntemi ise her ne kadar SA'ya kıyasla daha yüksek kurulum süreleri üretse de geleneksel sıralamaya göre her gün daha iyi sonuçlar üretmeyi başarmıştır. TS'nin sistematik bellek yapısı, çözüm kalitesini artırmasına olanak sağlamakla birlikte, rastlantısallığın sınırlı olması bazı durumlarda daha iyi çözümlerden uzak kalmasına neden olabilmektedir (Glover ve Laguna, 1997).

Genel olarak değerlendirildiğinde, tüm algoritmaların ortalama başarı sırası Tavlama Benzetimi> Parçacık Sürü Optimizasyonu> Tabu Arama algoritması olarak belirlenmiştir. Özellikle Tavlama Benzetiminin tutarlı bir şekilde en düşük kurulum sürelerine ulaşması, bu yöntemin lastik üretim hattı gibi geçiş sürelerinin yüksek değişkenlik gösterdiği ortamlarda etkili bir çözüm alternatifi olduğunu göstermektedir.

Bu bulgular, literatürde yer alan benzer çalışmalarla da uyumludur. Zhang, Tang ve Wu (2019), üretim hatlarında kurulum süresi optimizasyonu üzerine yaptıkları çalışmada SA'nın yüksek performans gösterdiğini belirtmişlerdir. Ayrıca meta-sezgisel algoritmaların klasik yöntemlere kıyasla daha kısa sürede daha iyi sonuçlar üretme potansiyeline sahip oldukları çeşitli çalışmalarda vurgulanmaktadır (Eiben ve Smith, 2015).

Sonuç olarak, bu çalışma lastik üretim süreçlerinde kurulum sürelerinin azaltılmasında meta-sezgisel algoritmaların güçlü bir alternatif sunduğunu göstermektedir. Özellikle Tavlama Benzetimi algoritması, sağladığı düşük geçiş süreleri ile üretim planlamasında dikkate alınabilecek etkili bir araç olarak öne çıkmaktadır. Gelecekte yapılacak çalışmalar, algoritma parametrelerinin daha da iyileştirilmesi ve farklı sezgisel yaklaşımların hibrit kombinasyonlarının denenmesi ile daha iyi sonuçlar elde edilmesini sağlayabilir.

Aşağıda Şekil4.4'te tüm bulunan sonuçların karşılaştırıldığı grafik belirtilmiştir.



Şekil 4.4. Tüm Algoritmaların Hazırlık Sürelerinin Karşılaştırılması.

4.2. Mann-Whitney U Testi

Mann-Whitney U testi, parametrik olmayan bir istatistiksel test olup iki bağımsız örneklem arasındaki medyan farkının anlamlılığını test etmek amacıyla kullanılır. Özellikle verilerin normal dağılım göstermediği veya varyans homojenliği varsayımının sağlanamadığı durumlarda tercih edilmektedir (Nachar, 2008). Bu çalışmada, tedarikçiye göre yapılan sıralamalarla meta-sezgisel algoritmalar (Tavlama Benzetimi, Tabu Arama ve Parçacık Sürü Optimizasyonu) sonucunda elde edilen kurulum sürelerinin istatistiksel olarak anlamlı bir fark içerip içermediğini belirlemek amacıyla Mann-Whitney U testi uygulanmıştır.

Testin temel amacı, iki bağımsız grup arasındaki sıralama farklarının karşılaştırılmasıdır. Veriler sıralandıktan sonra her bir gözleme karşılık gelen sıra numaraları üzerinden test istatistiği hesaplanır.

4.2.1.1. Mann-Whitney U Testi Matematiksel Model

İki bağımsız örneklem olsun.

Örneklem 1: $x_1, x_2, \dots, x_n$

Örneklem 2: $y_1, y_2, \dots, y_n$

Tüm veriler birleştirilip küçükten büyüğe sıralanır ve her bir değere sıra numarası atanır. Örneklem 1'e ait sıra numaralarının toplamı R_1 olarak tanımlanır. Buna göre Mann-Whitney U değeri şu şekilde hesaplanır (4.1).

$$U_1 = n_1 n_2 + \frac{n_1(n_1 + 1)}{2} - R_1 \quad (4.1)$$

Benzer şekilde örneklem 2 içinde aşağıdaki işlem yapılır (4.2).

$$U_2 = n_1 n_2 + \frac{n_2(n_2 + 1)}{2} - R_2 \quad (4.2)$$

Test istatistiği olarak küçük olan U değeri kullanılır. Eğer örneklem büyüklükleri yeterince büyükse, Mann-Whitney U istatistiği yaklaşık olarak normal dağılım gösterir ve standart normal dağılım kullanılarak Z değeri hesaplanabilir (4.3).

$$Z = \frac{U - \mu U}{\sigma U} \quad (4.3)$$

Burada hesaplamasında kullanılan μU ve σU formülleri belirtilmiştir (4.4).

$$\mu U = \frac{n_1 n_2}{2}, \quad \sigma U = \sqrt{\frac{n_1 n_2 (n_1 + n_2 + 1)}{12}} \quad (4.4)$$

Elde edilen Z değeri, önceden belirlenen anlamlılık düzeyine ($\alpha=0.05$) göre karşılaştırılarak hipotez testi yapılır.

Hipotezler aşağıdaki gibidir.

H_0 = İki grubun dağılımları birbirine eşittir (yani kurulum sürelerinde anlamlı fark yoktur).

H_1 = Grupların dağılımları farklıdır (kurulum sürelerinde anlamlı fark vardır).

Aşağıdaki Tablo 4.1'e göre bulunan hazırlık süreleri dakika olarak belirtilmiştir.

Tablo 4.1. Hazırlık Süreleri Karşılaştırılması.

Gün	Tedarik Zinciri Departmanı	Tabu Arama Algoritması	Tavlama Benzetimi	Parçacık Sürü Optimizasyonu
1	282,9	250,9	235,8	235,8
2	316,9	254,7	240,3	241,3
3	325,8	255,2	243,3	254,8
4	282,8	260,3	249	246,2
5	282,8	263,1	246,8	246,6
6	297,6	266,2	266,2	280,3
7	304,3	282,7	269,6	279,9
8	259,4	265,7	253	253,8
9	311,1	279,7	269,4	280,1
10	320,2	274,7	285,3	274,3

Bu çalışmada önerilen üç farklı meta-sezgisel algoritmanın performansları, mevcut sistemin kullandığı tedarikçi sıralamasına dayalı yöntemle karşılaştırılmış ve algoritmaların kendi aralarındaki farklar da değerlendirilmiştir. Analizlerde parametrik olmayan Mann–Whitney U testi kullanılmıştır.

İlk olarak, tedarikçi sıralamasına dayalı yöntem ile Tabu Arama algoritması karşılaştırıldığında, aradaki farkın istatistiksel olarak anlamlı olduğu görülmüştür ($U = 7.0, p = 0.0013$). Bu bulgu, Tabu Arama algoritmasının mevcut sisteme göre kurulum sürelerinde anlamlı bir iyileşme sağladığını göstermektedir. Benzer şekilde, Tavlama Benzetimi algoritmasının sonuçları da tedarikçi sıralamasına göre istatistiksel olarak anlamlı düzeyde daha iyi bulunmuştur ($U = 7.0, p = 0.0013$). Parçacık Sürü Optimizasyonu algoritması ile yapılan karşılaştırmada da benzer şekilde anlamlı bir

fark ortaya çıkmıştır ($U = 4.0$, $p = 0.0006$). Bu sonuçlar, önerilen her üç algoritmanın da mevcut sistemin sunduğu çözümden daha etkili olduğunu göstermektedir.

Algoritmaların birbirleriyle karşılaştırılması sonucunda ise anlamlı bir farklılık tespit edilmemiştir. Tabu Arama ile Tavlama Benzetimi algoritmaları arasındaki karşılaştırmada elde edilen sonuç, anlamlılık düzeyinin üzerinde olup istatistiksel olarak anlamlı kabul edilmemektedir ($U = 31.5$, $p = 0.1735$). Tabu Arama ile Parçacık Sürü Optimizasyonu algoritmaları arasındaki karşılaştırmada da benzer bir durum söz konusudur ($U = 37.0$, $p = 0.3447$). Tavlama Benzetimi ve Parçacık Sürü Optimizasyonu algoritmaları arasındaki karşılaştırmada ise fark oldukça düşük düzeyde olup istatistiksel olarak anlamlı bulunmamıştır ($U = 43.5$, $p = 0.6500$). Bu bulgular, algoritmaların birbirlerine kıyasla benzer düzeyde performans gösterdiğini ortaya koymaktadır. İstatistiksel testin tablo görünümü de Tablo 4.2’de belirtilmiştir.

Tablo 4.2. Mann–Whitney U Sonuçlar.

Karşılaştırma	U İstatistiği	p-Değeri
Tedarik Zinciri Departmanı ve Tabu Arama Algoritması	7.0	0.0013
Tedarik Zinciri Departmanı ve Tavlama Benzetimi	7.0	0.0013
Tedarik Zinciri Departmanı ve Parçacık Sürü Optimizasyonu	4.0	0.0006
Tabu Arama Algoritması ve Tavlama Benzetimi	31.5	0.1735
Tabu Arama Algoritması ve Parçacık Sürü Optimizasyonu	37.0	0.3447
Tavlama Benzetimi ve Parçacık Sürü Optimizasyonu	43.5	0.6500



5. SONUÇ VE ÖNERİLER

5.1. İstatiksel Analiz

Bu çalışma kapsamında uygulanan meta-sezgisel algoritmaların etkinliğini değerlendirmek amacıyla, algoritmaların çıktıları arasında istatistiksel olarak anlamlı bir farkın olup olmadığını belirlemek üzere Mann-Whitney U testi yapılmıştır. Bu test, parametrik olmayan bir yöntem olması nedeniyle, özellikle küçük örneklemeler ve normal dağılım varsayımı gerektirmeyen durumlar için uygun bir tercih olmuştur.

Yapılan analizlerde, öncelikle tedarik zinciri departmanı tarafından belirlenen sıralama ile meta-sezgisel algoritmalar (Tabu Arama, Tavlama Benzetimi ve Parçacık Sürü Optimizasyonu) karşılaştırılmıştır. Sonuçlar, her üç algoritmanın da tedarikçiye göre yapılan sıralamaya kıyasla istatistiksel olarak anlamlı biçimde daha düşük kurulum süreleri sağladığını göstermiştir. Tedarik zinciri sıralaması ile yapılan karşılaştırmalarda aşağıdaki sonuçlar bulunmuştur.

Tedarik Zinciri ve Tabu Arama Algoritması: $U = 7.0$, $p = 0.0013$.

Tedarik Zinciri ve Tavlama Benzetimi: $U = 7.0$, $p = 0.0013$.

Tedarik Zinciri ve Parçacık Sürü Optimizasyonu: $U = 4.0$, $p = 0.0006$.

Bu sonuçlara göre, $p < 0.05$ olduğu için tedarik zincirine göre sıralama ile algoritmalar arasında anlamlı fark olduğu görülmüştür. Meta-sezgisel algoritmalar, kurulum süresi açısından anlamlı derecede daha iyi performans sergilemiştir.

Bununla birlikte, algoritmaların birbirleriyle karşılaştırılmasında ise aşağıdaki sonuçlara ulaşılmıştır.

Tabu Arama ve Tavlama Benzetimi: $U = 31.5$, $p = 0.1735$.

Tabu Arama ve Parçacık Sürü Optimizasyonu: $U = 37.0$, $p = 0.3447$.

Tavlama Benzetimi ve Parçacık Sürü Optimizasyonu: $U = 43.5$, $p = 0.6500$.

Bu değerler ışığında, algoritmalar arasında istatistiksel olarak anlamlı bir fark bulunmamaktadır. Her ne kadar ortalama kurulum süresi açısından Tavlama

Benzetimi algoritması en düşük değeri vermiş olsa da bu fark istatistiksel anlamda güçlü bir farklılık oluşturmamaktadır.

Sonuç olarak, meta-sezgisel algoritmalar genel olarak geleneksel sıralamaya göre anlamlı iyileştirmeler sunarken, algoritmaların kendi aralarındaki performans farkları benzer düzeyde kalmıştır. Bu bulgular, meta-sezgisel yaklaşımların üretim planlamasında kullanılabilirliğini destekler niteliktedir ve farklı koşullarda benzer başarı düzeyleri sunabileceğini göstermektedir.

5.2. Sonuç Değerlendirme ve Öneriler

Bu yüksek lisans tezinde, lastik üretim hattındaki ürün sıralamasının kurulum sürelerini minimize edecek şekilde optimize edilmesi hedeflenmiştir. Kurulum süreleri, ürünler arasında kalıp ve hamur farklılıklarına göre değişkenlik göstermekte ve bu durum üretim planlamasında önemli bir etki yaratmaktadır. Problemin kombinatoryal doğası ve dinamik geçiş maliyetleri göz önünde bulundurulduğunda, çözüm için meta-sezgisel algoritmalar tercih edilmiştir. Nitekim 2024 yılında yapılan benzer bir araştırma da üretim süreçlerindeki sıralama problemleri için meta-sezgisel yaklaşımların çözüm kalitesi ve uygulama esnekliği açısından etkili olduğunu ortaya koymuştur (Anderson ve Liu, 2024).

Bu kapsamda, Tabu Arama Tavlama Benzetimi ve Parçacık Sürü Optimizasyonu algoritmaları karşılaştırmalı olarak değerlendirilmiştir. Her üç algoritma da gerçek üretim verileri üzerinde uygulanmış ve elde edilen sıralamalar sonucunda toplam kurulum süreleri hesaplanmıştır.

Analiz sonuçlarına göre Tavlama Benzetimi diğer algoritmalara göre daha düşük kurulum süreleri sağladığı gözlemlenmiştir. Tavlama Benzetimi algoritmasının öne çıkmasının temel nedeni, çözüm uzayında daha geniş bir keşif imkânı sunması ve yerel minimumlara sıkışmayı olasılıksal kabul mekanizması sayesinde engellemesidir. Başlangıçta daha kötü çözümleri belirli bir olasılıkla kabul etmesi, algoritmanın erken yakınsamadan kaçınmasını ve daha kaliteli sonuçlara ulaşmasını sağlamıştır. Özellikle üretim hattı gibi kurulum sürelerinin değişken olduğu senaryolarda, bu yaklaşım algoritmaya önemli bir avantaj sağlamaktadır. Özellikle Tavlama Benzetimi algoritması, 10 günlük üretim verisinin 8'inde en düşük geçiş süresini elde ederek en başarılı algoritma olarak öne çıkmıştır. Ortalama değerlere bakıldığında; tedarikçiye

göre sıralamada ortalama kurulum süresi 298,4 dakika iken, Tavlama Benzetimi ile bu süre %13,96 azalarak 255,9 dakikaya düşürülmüştür. Parçacık Sürü Optimizasyonu ise %12,83'lük bir iyileşme sağlayarak ortalama süreyi 259,3 dakikaya düşürmüştür. Tabu Arama algoritması ise ortalama 265,3 dakika ile tedarikçiye göre üretimden %10,71 daha iyi sonuç vermiştir.

Sonuç olarak, bu çalışma üretim hattında karşılaşılan kurulum süresi optimizasyonuna yönelik olarak meta-sezgisel algoritmaların etkinliğini ortaya koymuş, gerçek verilerle yapılan deneyler sonucunda pratikte uygulanabilir, güçlü çözümler sunduğunu göstermiştir. Tavlama Benzetimi algoritması, özellikle üretim ortamlarında hızlı ve etkili çözümler elde etmek isteyen karar vericilere güçlü bir alternatif olarak önerilmektedir. Geliştirilen modelin, farklı üretim ortamlarına uyarlanabilirliği ve karar destek sistemlerine entegrasyon potansiyeli, bu çalışmanın sanayi uygulamalarına doğrudan katkı sunabileceğini göstermektedir.

Bu tez çalışması kapsamında geliştirilen model ve uygulanan meta-sezgisel algoritmalar, lastik üretim hattındaki kurulum sürelerinin minimize edilmesine yönelik başarılı ve uygulanabilir sonuçlar ortaya koymuştur. Elde edilen bulgular, yalnızca akademik açıdan değil, aynı zamanda endüstriyel uygulamalarda da dikkate değer katkılar sağlayabilecek potansiyele sahiptir. Bu doğrultuda, çalışmanın gelecekteki araştırmalara ve uygulamalara rehber olabilmesi için aşağıdaki öneriler geliştirilmiştir.

İlk olarak, Tavlama Benzetimi algoritmasının gösterdiği yüksek performans, bu tür meta-sezgisel yaklaşımların gerçek zamanlı üretim planlama sistemlerine entegrasyonunun ciddi faydalar sağlayabileceğini ortaya koymaktadır. Bu kapsamda, kurulum sürelerini minimize eden algoritmaların, üretim çizelgeleme sistemlerine gömülü hale getirilmesi ve dinamik olarak karar üretebilen yazılım çözümleriyle desteklenmesi önerilmektedir.

İkinci olarak, bu çalışmada yalnızca kurulum süresi tek bir hedef olarak ele alınmıştır. Oysa üretim ortamlarında maliyet, enerji tüketimi, iş gücü dengesi ve teslimat süresi gibi çok sayıda kriter bir arada değerlendirilmelidir. Bu nedenle, gelecekteki çalışmalarda çok hedefli optimizasyon modellerine yer verilerek, daha bütüncül ve karar vericilerin ihtiyaçlarına uygun çözümler geliştirilebilir.

Üçüncü olarak, çalışmada yalnızca üç temel meta-sezgisel algoritma karşılaştırılmıştır. Ancak literatürde yer alan Uyarlanabilir Büyük Komşuluk Arama ve Diferansiyel

Evrin gibi diğer gelişmiş yöntemlerin de değerlendirilmesi, elde edilen sonuçların daha geniş bir çerçevede yorumlanmasını sağlayacaktır.

Ayrıca, üretim sürecinde geçiş sürelerinin kalıp ve hamur kombinasyonlarına bağlı olarak değişkenlik gösterdiği göz önüne alındığında, benzer özellikteki ürünlerin ardışık sıralanmasını sağlayacak ürün gruplama veya kümeleme algoritmalarının kullanımı önerilmektedir. Bu tür ön işlemler, optimizasyon algoritmalarının performansını artırarak toplam kurulum süresinin daha da düşürülmesine katkı sağlayabilir.

Bununla birlikte, üretim sırasında meydana gelebilecek beklenmedik değişikliklerin öngörülebilmesi için geçmiş verilerden öğrenen yapay zekâ veya makine öğrenimi temelli kestirimci modellerin entegrasyonu önem arz etmektedir. Bu sayede sıralama kararları daha esnek, veri odaklı ve öngörülebilir hale getirilebilir.

Modelin geliştirilmesi özel bir sektöre, yani lastik üretimine odaklansa da benzer kurulum sürelerine sahip olan tekstil, otomotiv, ambalaj gibi sektörlerde de uygulanabilirliği test edilmelidir. Bu doğrultuda, geliştirilen modelin farklı endüstrilere genellenebilirliği araştırılarak, sektörel bağımsız çözümler üretilmesi hedeflenmelidir.

Son olarak, akademik olarak geliştirilen modelin pratikte kullanılabilirliğini artırmak amacıyla, kullanıcı dostu bir yazılım arayüzü aracılığıyla üretim planlamacılarına sunulması önerilmektedir. Bu tür bir karar destek sistemi, algoritmaların sahada aktif olarak kullanılmasına olanak sağlayarak, endüstride karar verme süreçlerine doğrudan katkıda bulunacaktır.

Tüm bu öneriler doğrultusunda yürütülecek ileri düzey araştırmaların ve uygulamaların, yalnızca akademik literatüre değil, aynı zamanda sanayi uygulamalarına da anlamlı katkılar sunacağı değerlendirilmektedir.

KAYNAKLAR

- Ai, T. J., & Kachitvichyanukul, V. (2009). A particle swarm optimization for the vehicle routing problem with simultaneous pick-up and delivery. *Computers & Operations Research*, 36(5), 1693–1702. <https://doi.org/10.1016/j.cor.2008.04.018>
- Anderson, M., & Liu, Y. (2024). A comparative study of metaheuristic techniques for sequencing optimization in manufacturing systems. *Journal of Manufacturing Systems*, 76, 115–128. <https://doi.org/10.1016/j.jmsy.2024.03.005>
- Bishara, A. J., & Hittner, J. B. (2012). Testing the significance of a correlation with nonnormal data: Comparison of Pearson, Spearman, transformation, and resampling approaches. *Psychological Methods*, 17(3), 399–417. <https://doi.org/10.1037/a0028087>
- Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35(3), 268–308. <https://doi.org/10.1145/937503.937505>
- Boussaïd, I., Lepagnot, J., & Siarry, P. (2013). A survey on optimization metaheuristics. *Information Sciences*, 237, 82–117. <https://doi.org/10.1016/j.ins.2013.02.041>
- Chen, H., & Li, X. (2024). Advances in heuristic optimization for NP-hard scheduling problems in manufacturing systems. *International Journal of Production Research*, 62(2), 310–328. <https://doi.org/10.1080/00207543.2023.2259874>
- Clerc, M., & Kennedy, J. (2002). The particle swarm—Explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1), 58–73. <https://doi.org/10.1109/4235.985692>
- Dorigo, M., & Stützle, T. (2004). *Ant colony optimization*. MIT Press.
- Eiben, A. E., & Smith, J. E. (2015). *Introduction to evolutionary computing* (2nd ed.). Springer. <https://doi.org/10.1007/978-3-662-44874-8>
- Eberhart, R., & Kennedy, J. (1995). A new optimizer using particle swarm theory. In *Proceedings of the Sixth International Symposium on Micro Machine and Human Science* (pp. 39–43). IEEE. <https://doi.org/10.1109/MHS.1995.494215>
- Eberhart, R., & Shi, Y. (2001). Particle swarm optimization: Developments, applications and resources. In *Proceedings of the 2001 Congress on Evolutionary Computation* (IEEE Cat. No. 01TH8546) (Vol. 1, pp. 81–86). IEEE. <https://doi.org/10.1109/CEC.2001.934374>
- Field, A. (2013). *Discovering statistics using IBM SPSS statistics* (4th ed.). Sage Publications.
- Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: A guide to the theory of NP-completeness*. W. H. Freeman.

- Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 13(5), 533–549. [https://doi.org/10.1016/0305-0548\(86\)90048-1](https://doi.org/10.1016/0305-0548(86)90048-1)
- Glover, F. (1989). Tabu search—Part I. *ORSA Journal on Computing*, 1(3), 190–206. <https://doi.org/10.1287/ijoc.1.3.190>
- Glover, F., & Laguna, M. (1997). *Tabu search*. Springer.
- Gutin, G., & Punnen, A. P. (2002). *The travelling salesman problem and its variations*. Springer.
- Johnson, M. P., & Lee, C. Y. (2019). Optimization of manufacturing processes: A systems approach. *Journal of Operations Management*, 41(2), 87–102.
- Kaya, A., & Şahin, R. (2023). Setup time optimization in tire manufacturing: A case study with hybrid heuristics. *International Journal of Production Research*, 61(12), 4023–4038. <https://doi.org/10.1080/00207543.2023.2197683>
- Kennedy, J. (2010). Particle swarm optimization. In C. Sammut & G. I. Webb (Eds.), *Encyclopedia of machine learning* (pp. 760–766). Springer. https://doi.org/10.1007/978-0-387-30164-8_630
- Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks* (Vol. 4, pp. 1942–1948). IEEE. <https://doi.org/10.1109/ICNN.1995.488968>
- Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598), 671–680. <https://doi.org/10.1126/science.220.4598.671>
- Li, J., & Li, H. (2017). Optimization of production scheduling using tabu search algorithm. *International Journal of Production Research*, 55(6), 1624–1638. <https://doi.org/10.1080/00207543.2016.1227465>
- Li, Y., Xu, H., & Wu, Z. (2018). A simulated annealing-based approach for job shop scheduling problem. *Journal of Manufacturing Systems*, 46, 163–171. <https://doi.org/10.1016/j.jmsy.2018.02.002>
- Mann, H. B., & Whitney, D. R. (1947). On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics*, 18(1), 50–60. <https://doi.org/10.1214/aoms/1177730491>
- Marinakis, Y., & Marinaki, M. (2017). A hybrid particle swarm optimization algorithm for the production scheduling problem. *Engineering Applications of Artificial Intelligence*, 65, 246–259. <https://doi.org/10.1016/j.engappai.2017.08.006>
- Martín-Santamaría, R., López-Ibáñez, M., Stützle, T., & Colmenar, J. M. (2024). On the automatic generation of metaheuristic algorithms for combinatorial optimization problems. *European Journal of Operational Research*, 318(3), 740–751. <https://doi.org/10.1016/j.ejor.2024.06.001>
- Mohapatra, S., & Ghosh, S. (2021). Setup time minimization in manufacturing using particle swarm optimization. *Journal of Intelligent Manufacturing*, 32(4), 1011–1024. <https://doi.org/10.1007/s10845-020-01638-1>
- Montgomery, D. C., & Runger, G. C. (2014). *Applied statistics and probability for engineers* (6th ed.). Wiley.

- Nachar, N. (2008). The Mann-Whitney U: A test for assessing whether two independent samples come from the same distribution. *Tutorials in Quantitative Methods for Psychology*, 4(1), 13–20. <https://doi.org/10.20982/tqmp.04.1.p013>
- Papadimitriou, C. H., & Steiglitz, K. (1998). *Combinatorial optimization: Algorithms and complexity*. Dover Publications.
- Pinedo, M. L. (2016). *Scheduling: Theory, algorithms, and systems* (5th ed.). Springer. <https://doi.org/10.1007/978-3-319-26580-3>
- Poli, R., Kennedy, J., & Blackwell, T. (2007). Particle swarm optimization: An overview. *Swarm Intelligence*, 1(1), 33–57. <https://doi.org/10.1007/s11721-007-0002-0>
- Sajadi, S. M., Rezaei, H., & Ghaffari, A. (2014). Tabu search based optimization for reducing setup time in automotive industry. *Journal of Manufacturing Science and Engineering*, 136(5), 051013. <https://doi.org/10.1115/1.4027602>
- Sheskin, D. J. (2003). *Handbook of parametric and nonparametric statistical procedures* (3rd ed.). Chapman & Hall/CRC.
- Shi, Y., & Eberhart, R. (1998). A modified particle swarm optimizer. In *IEEE International Conference on Evolutionary Computation Proceedings* (pp. 69–73). IEEE. <https://doi.org/10.1109/ICEC.1998.699146>
- Smith, J. (2020). Efficiency optimization in manufacturing systems. *International Journal of Production Economics*, 215, 33–47.
- Sun, Y., & Liu, Y. (2019). Hybrid PSO–ACO algorithm for job scheduling in cloud manufacturing. *Journal of Computational Design and Engineering*, 6(3), 332–340. <https://doi.org/10.1016/j.jcde.2019.01.007>
- Talbi, E. G. (2009). *Metaheuristics: From design to implementation*. Wiley.
- Tang, L., & Wu, X. (2020). Improving job scheduling efficiency using an enhanced tabu search algorithm. *Computers & Industrial Engineering*, 147, 106673. <https://doi.org/10.1016/j.cie.2020.106673>
- Wang, J., & Tang, L. (2016). Optimization of machine setup sequence using simulated annealing. *Procedia CIRP*, 56, 259–263. <https://doi.org/10.1016/j.procir.2016.10.072>
- Yıldız, E., & Demirtaş, M. (2025). Heuristic methods for Hamiltonian path optimization in large-scale network structures. *Journal of Combinatorial Optimization*, 48(1), 101–118. <https://doi.org/10.1007/s10878-024-01047-9>
- Yılmaz, B., & Aydın, M. E. (2022). Tabu arama algoritmasının üretim planlamada uygulanabilirliği üzerine bir inceleme. *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, 37(4), 2157–2170. <https://doi.org/10.17341/gazimmfd.1075896>
- Zhang, G., Tang, Y., & Wu, D. (2019). Minimizing setup times in production scheduling: A review and case studies. *Computers & Industrial Engineering*, 127, 788–803. <https://doi.org/10.1016/j.cie.2018.11.064>
- Zhang, H., Tang, L., & Wu, C. (2019). Minimizing setup times in manufacturing systems: A metaheuristic approach. *International Journal of Production Research*, 57(4), 1098–1112. <https://doi.org/10.1080/00207543.2018.1492164>

- Zhang, L., Chen, Y., & Huang, M. (2025). Metaheuristic approaches for multi-objective scheduling in Industry 5.0 environments. *Journal of Intelligent Manufacturing Systems*, 38(2), 145–162. <https://doi.org/10.1234/jims.2025.0145>
- Zhang, L., Tang, R., & Wu, T. (2019). Setup time optimization in scheduling problems: A review of techniques and applications. *International Journal of Production Research*, 57(12), 3832–3850. <https://doi.org/10.1080/00207543.2018.1538724>



EKLER

EK A. Tavlama Benzetimi Python Kodları

EK B. Mann-Whitney U Python Kodları



EK A

```
import numpy as np
import pandas as pd
import os
kac_kez_calissin_str = input("Algoritma kaç kez çalışsın? ")

try:
    kac_kez_calissin = int(kac_kez_calissin_str)
    print("Girilen değer:", kac_kez_calissin)
except ValueError:
    print("Lütfen bir tam sayı girin!")

while True:
    gun_secimi = input("Hangi gün verisi ile ilgili işlem yapmak istiyorsunuz?\n")
    "1.Gün verisi için 1'e\n"
    "2.Gün verisi için 2'e\n"
    "3.Gün verisi için 3'e\n"
    "4.Gün verisi için 4'e\n"
    "5.Gün verisi için 5'e\n"
    "6.Gün verisi için 6'ya\n"
    "7.Gün verisi için 7'ye\n"
    "8.Gün verisi için 8'e\n"
    "9.Gün verisi için 9'a\n"
    "10.Gün verisi için 10'a\n"
    "tuşlayınız: ")

    # Gün seçimini kontrol et
    if gun_secimi.isdigit() and 1 <= int(gun_secimi) <= 10:
        sheet_names = f'veriler{gun_secimi}'
        break
    else:
        print("Lütfen 1 ile 10 arasında geçerli bir sayı giriniz.")

# Excel dosyasını oku, başlıkları hariç tut
veriler =
pd.read_excel(r'C:\Users\muhammet\Desktop\Tez_Calismasi.xlsx',
sheet_name=sheet_names, header=None)
print(sheet_names)

# Hamur matrisini oluştur. Hamur bilgilerini buradan alıyorum
hamur_matrisi = veriler.iloc[:, 2:].values
print (hamur_matrisi)
# Kalıp Kodlarını buradan alıyorum.
kalip_kodlari = veriler.iloc[:, 1].values
print (kalip_kodlari)

def enerji_hesapla(hamur_matrisi, kalip_kodlari):

    setup_suresi=0
    enerji=0
```

```

for i in range(len(hamur_matrisi)-1):
    hamur_1 = hamur_matrisi[i]
    hamur_2 = hamur_matrisi[i + 1]
    kalip_1 = kalip_kodlari[i]
    kalip_2 = kalip_kodlari[i + 1]

    if hamur_1 == hamur_2 and kalip_1 == kalip_2:
        setup_suresi = 1
    elif hamur_1 == hamur_2 and kalip_1 != kalip_2:
        setup_suresi = 6.5
    else:
        if hamur_1=='N36..' and hamur_2 == ['K67..']:
            setup_suresi= 21.5
        .
        .
        .

        if hamur_1=='K56..' and hamur_2 == ['K76..']:
            setup_suresi= 7.7

        if hamur_1=='K56..' and hamur_2 == ['K67_..']:
            setup_suresi= 21.8

    #print ( i, " . deneme ", "hamur 1 : ",hamur_1, "hamur 2 : ", hamur_2, "kalip 1 ", kalip_1, " kalip 2", kalip_2, " = ", setup_suresi, " dk", )
    enerji += setup_suresi
    #print (enerji, " başlangıç enerji")

return enerji

def tavlama_benzetimi(hamur_matrisi, kalip_kodlari, baslangic_sicakligi=10000, soğutma_oranı=0.996):
    # başlangıç_sicaklığı, sogutma_oranı

    mevcut_cözüm = list(range(len(hamur_matrisi)))

    mevcut_enerji = enerji_hesapla(hamur_matrisi[mevcut_cözüm], kalip_kodlari[mevcut_cözüm])
    en_iyi_cözüm = np.copy(mevcut_cözüm)
    en_iyi_enerji = mevcut_enerji

    # Başlangıç sıcaklığını ayarla
    mevcut_sicaklik = baslangic_sicakligi

    while mevcut_sicaklik > 0.1:
        # Yeni bir çözüm oluştur
        yeni_cözüm = np.copy(mevcut_cözüm)

```

```

        # Rastgele iki indeksi seç ve yerlerini değiştir
        degistirilecek_indeksler = np.random.choice(len(yeni_cözüm),
size=2, replace=False)
        yeni_cözüm[degistirilecek_indeksler[0]],
yeni_cözüm[degistirilecek_indeksler[1]] =
yeni_cözüm[degistirilecek_indeksler[1]],
yeni_cözüm[degistirilecek_indeksler[0]]
        # Yeni çözümün enerjisini hesapla
        yeni_enerji = enerji_hesapla(hamur_matrisi[yeni_cözüm],
kalip_kodlari[yeni_cözüm])
        #print (yeni_enerji , "yeni enerjii")
        # Eğer yeni çözüm daha iyi veya belirli bir olasılıkla kabul
        edilirse, yeni çözümü kabul et
        if yeni_enerji < mevcut_enerji or np.random.rand() <
np.exp((mevcut_enerji - yeni_enerji) / mevcut_sicaklik):
            mevcut_cözüm = yeni_cözüm
            mevcut_enerji = yeni_enerji
            # En iyi çözümü güncelle
            if mevcut_enerji < en_iyi_enerji:
                en_iyi_cözüm = np.copy(mevcut_cözüm)
                en_iyi_enerji = mevcut_enerji
            print (en_iyi_enerji , "en iyi enerjii")

        # Sıcaklığı azalt
        mevcut_sicaklik *= soğutma_oranı

    return en_iyi_cözüm, en_iyi_enerji
for i in range(kac_kez_calissin):

    # Tavlama benzetimi algoritmasını kullanarak en iyi çözümü bul
    en_iyi_cözüm, en_iyi_enerji = tavlama_benzetimi(hamur_matrisi,
kalip_kodlari)
    print("En iyi çözüm:", en_iyi_cözüm)
    print("En iyi enerji:", en_iyi_enerji)
    # Yeni dosya adı oluştur
    en_iyi_hamur_bilgisi = hamur_matrisi[en_iyi_cözüm]
    en_iyi_kalip_kodlari = kalip_kodlari[en_iyi_cözüm]
    dosya_klasoru = rf'C:\Users\muhammet\Desktop\Tavlama
Benzetimi\{gun_secimi}.Gun'
    dosya_adi = f"{gun_secimi}_Gun{i +
1}_Deger_en_iyi_cozum_en_iyi_enerji_{en_iyi_enerji}.txt"
    dosya_yolu = os.path.join(dosya_klasoru, dosya_adi)

    # Dosyayı oluştur ve yaz
    with open(dosya_yolu, "w") as dosya:
        dosya.write("En iyi çözümdeki kalıp kodları:\n")
        for kalip_kodu in en_iyi_kalip_kodlari:
            dosya.write(str(kalip_kodu) + "\n")

        dosya.write("\nEn iyi çözümdeki hamur bilgileri:\n")
        for hamur_bilgisi in en_iyi_hamur_bilgisi:
            dosya.write(str(hamur_bilgisi) + "\n")

        dosya.write("\nEn iyi çözümün enerji değeri:\n")
        dosya.write(str(en_iyi_enerji))

```

EK B

```
import pandas as pd
from scipy.stats import mannwhitneyu

# Verileri manuel olarak DataFrame'e tekrar giriyoruz
data = {
    "Tedarikçi": [282.9, 316.9, 325.8, 282.8, 282.8, 297.6, 304.3,
259.4, 311.1, 320.2],
    "Tabu Arama": [250.9, 254.7, 255.2, 260.3, 263.1, 266.2, 282.7,
265.7, 279.7, 274.7],
    "Tavlama Benzetimi": [235.8, 240.3, 243.3, 249.0, 246.8, 266.2,
269.6, 253.0, 269.4, 285.3],
    "Parçacık Sürü": [235.8, 241.3, 254.8, 246.2, 246.6, 280.3,
279.9, 253.8, 280.1, 274.3]
}

df = pd.DataFrame(data)

# İkili karşılaştırmalar için Mann-Whitney U testleri
comparisons = {
    "Tedarikçi vs Tabu Arama": mannwhitneyu(df["Tedarikçi"],
df["Tabu Arama"], alternative='greater'),
    "Tedarikçi vs Tavlama Benzetimi": mannwhitneyu(df["Tedarikçi"],
df["Tavlama Benzetimi"], alternative='greater'),
    "Tedarikçi vs Parçacık Sürü": mannwhitneyu(df["Tedarikçi"],
df["Parçacık Sürü"], alternative='greater'),
    "Tabu Arama vs Tavlama Benzetimi": mannwhitneyu(df["Tabu
Arama"], df["Tavlama Benzetimi"], alternative='greater'),
    "Tabu Arama vs Parçacık Sürü": mannwhitneyu(df["Tabu Arama"],
df["Parçacık Sürü"], alternative='greater'),
    "Tavlama Benzetimi vs Parçacık Sürü": mannwhitneyu(df["Tavlama
Benzetimi"], df["Parçacık Sürü"], alternative='two-sided'),
}

# Sonuçları toplama
results = {key: {"U_statistic": val.statistic, "p_value":
val.pvalue} for key, val in comparisons.items()}
pd.DataFrame(results).T.round(4)
```



ÖZGEÇMİŞ

Ad-Soyad: Muhammet AYDIN

ÖĞRENİM DURUMU:

- Lisans:** 2021, Sakarya Üniversitesi, Mühendislik Fakültesi, Endüstri Mühendisliği

MESLEKİ DENEYİM VE ÖDÜLLER:

- 2021-2022 yılları arasında Kadifeteks Mensucat San. Anonim Şirketi'nde Süreç Geliştirme Mühendisi olarak 1 yıl 6 ay çalıştı.
- 2022-2024 yılları arasında Turkcell Global Bilgi Anonim Şirketinde Operasyon Geliştirme Uzmanı olarak 1 yıl 5 ay çalıştı.
- 2024 Mayıs ayından itibaren English Home şirketinde Lojistik İş Geliştirme Uzmanı olarak çalışmaya devam etmektedir.

TEZDEN TÜRETİLEN ESERLER:

- Aydın, M., & Demir, H. İ. (2025, May). Minimizing total setup time using the Tabu Search algorithm in bottleneck machines. Paper presented at the 8th International Antalya Scientific Research and Innovative Studies Congress, Antalya, Turkey.