

TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**DEĞİŞİK TİP PARÇA ÜRETEN ROBOTLU AKIŞ HÜCRELERİNDE
DÖNGÜSEL ÇİZELGELEME**

YÜKSEK LİSANS TEZİ

Betül ÇOBAN

Endüstri Mühendisliği Anabilim Dalı

Tez Danışmanı: Doç. Dr. Hakan GÜLTEKİN

Haziran 2016

Fen Bilimleri Enstitüsü Onayı

.....
Prof. Dr. Osman EROĞUL
Müdür

Bu tezin Yüksek Lisans derecesinin tüm gereksinimlerini sağladığını onaylarım.

.....
Prof. Dr. Tahir HANALIOĞLU
Anabilimdalı Başkanı

TOBB ETÜ, Fen Bilimleri Enstitüsü'nün **141311012** numaralı Yüksek Lisans Öğrencisi **Betül ÇOBAN**'ın ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı "**DEĞİŞİK TİP PARÇA ÜRETEN ROBOTLU AKIŞ HÜCRELERİNDE DÖNGÜSEL ÇİZELGELEME**" başlıklı tezi **30.06.2016** tarihinde aşağıda imzaları olan jüri tarafından kabul edilmiştir.

Tez Danışmanı : **Doç. Dr. Hakan GÜLTEKİN**
TOBB Ekonomive Teknoloji Üniversitesi

Jüri Üyeleri : **Doç. Dr. Sinan GÜREL (Başkan)**
Orta Doğu Teknik Üniversitesi

Yrd. Doç. Dr. Ayşegül ALTIN KAYHAN
TOBB Ekonomive Teknoloji Üniversitesi

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, alıntı yapılan kaynaklara eksiksiz atıf yapıldığını, referansların tam olarak belirtildiğini ve ayrıca bu tezin TOBB ETÜ Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırlandığını bildiririm.

Betül ÇOBAN

ÖZET

Yüksek Lisans Tezi

DEĞİŞİK TİP PARÇA ÜRETEN ROBOTLU AKIŞ HÜCRELERİNDE DÖNGÜSEL ÇİZELGELEME

Betül ÇOBAN

TOBB Ekonomi ve Teknoloji Üniversitesi
Fen Bilimleri Enstitüsü
Endüstri Mühendisliği Anabilim Dalı

Tez Danışmanı: Doç.Dr. Hakan GÜLTEKİN

Tarih: Haziran 2016

Bu çalışmada ele alınan problem, tek tutuculu robotun kullanıldığı ve değişik tip parça üreten robotik hücrelerin tasarımı problemidir. Tek tutuculu robotlar aynı anda üzerlerinde tek bir parça bulundurabilir. Sistemde parçalar makinelerde işlenirken, makineler arası parça transferi ile makinelerin yüklenmesi ve boşaltılması ise sistemdeki tek bir robot tarafından yapılmaktadır.

Üretilcek her tip parçanın sistemdeki makinelerde işlenmesi gerekmekte, parça ve makineye özgü işlem zamanları bulunmaktadır. Bu tür sistemlerde hem robot belirli bir bilgisayar kodunu takip ederek aktivitelerini yaptığı için hem de uygulaması ve denetlemesi kolay olduğu için döngüsel üretim tercih edilmektedir. n adet parça üretilmesi ile oluşan " n -birim döngüsü" adı verilen döngü çeşidi ele alınmıştır.

Ele alınan problemde çıktı miktarını en büyükleyen yani döngü zamanını en küçükleyen robot hareket sıralaması ve parça üretim sıralamasını eş zamanlı belirleyen bir çözüm yöntemi belirlenmesi amaçlanmaktadır.

Literatürde m makineli bir sistemde değişik tip parça üretimi ele alınan ve üretim stratejisi olarak döngüsel üretimin tercih edildiği ayrıca n -birim döngülerini inceleyen robot hareket ve parça üretim sıralamalarını eş zamanlı belirleyen bir çalışma yoktur. Kesin çözüm için matematiksel modelleme tabanlı yöntemler geliştirilmiştir. Fakat, NP-Zor olan

bu problem için, özellikle problem boyutu büyüdüğünde matematiksel modellerle çözüm elde edilemeyeceğinden sezgisel algoritmalar geliştirilmiştir. Geliştirilen matematiksel model ve sezgiselin performansını ölçmek amacı ile birçok deneysel çalışma yapılmıştır.

Anahtar Kelimeler: Robotik hücreler, Değişik tip parça, Döngüsel üretim.



ABSTRACT

Master of Science

Cyclic scheduling in robotic cells producing multiple part types

Betül ÇOBAN

TOBB University of Economics and Technology
Institute of Natural and Applied Sciences
Department of Industrial Engineering

Supervisor: Doç.Dr. Hakan GÜLTEKİN

Date: June 2016

In this study, the considered problem is design of robotic cell producing multiple parts and consists of machines served by a single gripper robot. A single gripper robot can hold only one part at a time. In the system, while the parts are processing in the machines, transfers between machines and loading/unloading of machines is performed by a single robot.

Each parts have to be proceed in all machines, there are specific process time according to parts and machines. In this system, cyclic production is preferred because both the robot performs activities by following computer codes and it is easy to implement and control. " n -unit cycle" as a sequence of robot activities in which the system returns to the same state after producing n parts is handled.

In this problem, the aim is generated a solution method to simultaneously determine the optimal part sequence and robot move sequence that maximize throughput or equivalently minimize the long-run average cycle time.

In the literature, there is no study that is determining n -unit cycle that minimizes the cycle time for multiple part-type production in m machines to simultaneously determine robot move sequences and part sequence. The mathematical model is developed as an exact solution method. Therefore, for this problem includes scheduling of robot moves and sequencing of parts simultaneously which is shown to be NP-hard, heuristic methods

are developed because no available integer solutions are obtained particularly for large-scaled problems. Experimental studies is performed to evaluate the performance of the proposed algorithms.

Keywords: Robotic cell, Multiple part type, Cyclic manufacturing.



TEŞEKKÜR

Bu yolda pes etmeyi ne zaman düşünsem üzerimde emeklerini düşünüp vazgeçtiğim aileme, merhameti ile desteğini her daim yanımda hissettiğim anneme, azim ve çalışkanlıkta örnek aldığım bu hayattaki en büyük dayanağım olan babama, sevgileri ile beni desteksiz bırakmayan kardeşlerim Atıf ve Beyza'ya teşekkürlerimi iletiyorum. Yüksek lisans dönemi boyunca çözülmeyen tüm soru ve sorunlarımı çözmek için emeğini ve zamanını harcamaktan hiçbir zaman kaçınmayan danışmanım Doç. Dr. Hakan Gültekin'e, ve kıymetli tecrübelerinden faydalandığım TOBB Ekonomi ve Teknoloji Üniversitesi Endüstri Mühendisliği Bölümü öğretim üyelerine bu sayede teşekkürlerimi iletiyorum. Onlara ayırdığım zaman azalsa da fedakarlıkları ile benimle hala arkadaş olmaya devam eden tüm dostlarıma, ayrıca manevi ve akademik desteği ve bana beden daha çok inanması nedeni ile Cansu'ya teşekkür ediyorum. Uzaklardan aynı dertler ile dertlendiğimiz bir nevi meslektaşım olan Elif'e, robot hikayelerini dinleyip heyecanlanan Törenur'a ve daha sayamadığım tüm arkadaşlarıma da teşekkürü borç bilirim. TÜBİTAK'a 213M435 numaralı proje kapsamında verdiği destek için teşekkürlerimi iletiyorum. Son olarak TOBB Ekonomi ve Teknoloji Üniversitesi'ne burs sağladığı için teşekkürü borç bilirim.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	iv
ABSTRACT	vi
TEŞEKKÜR	viii
İÇİNDEKİLER	ix
ŞEKİL LİSTESİ	x
ÇİZELGE LİSTESİ	xi
KISALTMALAR	xii
SEMBOL LİSTESİ	xiii
1. GİRİŞ	1
1.1 Tezin Amacı	3
1.2 Literatür Araştırması	3
1.3 Problem Tanımı	8
2. MATEMATİKSEL MODEL	13
2.1 Altsınır Hesaplanması	19
3. SEZGİSEL YÖNTEM	25
3.1 n -birim Döngüler için Geliştirilen Algoritma	25
3.1.1 Tabu arama algoritması	26
3.1.2 Genetik algoritma	32
3.2 1-birim Döngüler için Geliştirilen Algoritma	35
4. GELİŞTİRİLEN ÇÖZÜM YÖNTEMLERİNİN KARŞILAŞTIRILMASI ..	37
4.1 Parametre Kalibrasyonu	38
4.2 Test Sonuçları	45
5. SONUÇ VE ÖNERİLER	49
KAYNAKLAR	52
EKLER	54
ÖZGEÇMİŞ	61

ŞEKİL LİSTESİ

	<u>Sayfa</u>
Şekil 1.1: Robotik Hücre.	2
Şekil 1.2: Aktivite (A_j) tanımı.	8
Şekil 1.3: Gant şeması.	12
Şekil 3.1: Tabu arama algoritmasının akış şeması.	26
Şekil 3.2: Genetik algoritma akış şeması.	33



ÇİZELGE LİSTESİ

	<u>Sayfa</u>
Çizelge 1.1: Değişik tip parça üretim problemi için literatür özeti.	7
Çizelge 1.2: İşlem zamanları.	11
Çizelge 1.3: Robot hareket sıralaması ve aktivite başlangıç zamanları.	11
Çizelge 2.1: Matematiksel modelden elde edilen sonuçlar.	18
Çizelge 2.2: Altsınır hesaplama örneği verileri.	21
Çizelge 2.3: Altsınır hesaplama örneği verilerinin sıralanmış hali.	21
Çizelge 2.4: Altsınır eklendikten sonra elde edilen sonuçlar.	22
Çizelge 3.1: İşlem zamanları.	29
Çizelge 4.1: Paralel ve normal kodlanan algoritmanın karşılaştırılması.	38
Çizelge 4.2: n -birim algoritmasında kullanılan parametreler ve denenen seviyeler.	39
Çizelge 4.3: n -birim algoritmasının parametre kalibrasyon koşullarının sonuçları- 1.	40
Çizelge 4.4: n -birim algoritmasının parametre kalibrasyon koşullarının sonuçları- 2.	41
Çizelge 4.5: 1-birim algoritmasında kullanılan parametreler ve denenen seviyeler.	42
Çizelge 4.6: 1-birim algoritmasının parametre kalibrasyon koşullarının sonuçları- 1.	43
Çizelge 4.7: 1-birim algoritmasının parametre kalibrasyon koşullarının sonuçları- 2.	44
Çizelge 4.8: Matematiksel model ve sezgisel algoritma sonuçları-1.	46
Çizelge 4.9: Matematiksel model ve sezgisel algoritma sonuçları-2.	47
Çizelge 4.10: Başlangıç çözümü verilen matematiksel model ve sezgisel algo- ritma sonuçları-1.	48
Çizelge 4.11: Başlangıç çözümü verilen matematiksel model ve sezgisel algo- ritma sonuçları-2.	48

KISALTMALAR

RHS	:Robot Hareket Sıralaması
PS	:Parça Sıralaması
TAA	:Tabu Arama Algoritması
GA	:Genetik Algoritma
ZL	:Zaman Limiti
TL	:Tabu Listesi



SEMBOL LİSTESİ

Bu çalışmada kullanılmış olan simgeler açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler Açıklama

A_j	Robotun j makinesini boşaltıp parçayı $j + 1$ makinesine taşıması
δ	Ardışık iki makine arasındaki transfer zamanı
ε	Makine yükleme/boşaltma zamanı
P_m^n	$n \in \mathcal{N}$ parçasının makine $m \in \mathcal{N}$ 'deki işlem süresi
T_h	Pozisyon $h \in \mathcal{P}$ 'ye atanan aktivitenin başlangıç zamanı
CT	Döngü zamanı
γ_g	Genetik Algoritma için kullanılan durma kriteri
γ_t	Tabu Arama Algoritması için kullanılan durma kriteri
ρ	Mutasyon olasılığı
σ	Elit çözüm sayısı belirlemek için kullanılan katsayı
λ	Tabu listesi belirlemek için kullanılan uzunluk katsayı

1. GİRİŞ

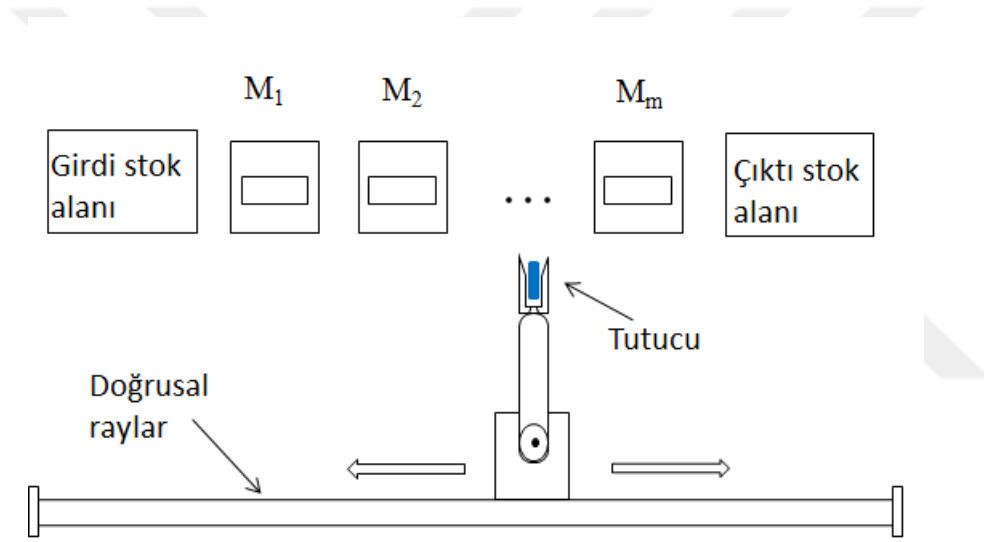
Üretim sektöründeki rekabet ve teknolojinin sürekli devam eden gelişmeleri dikkate alındığında firmaların üretim süreçlerini iyileştirmelerinin bir gereklilik olduğu görülmektedir. Müşterilerin beklentilerinin zamanında ve daha az maliyet ile karşılanması asıl amaç haline gelmiştir. Üretim hızı, hem zamanında teslimat hem de birim ürün maliyetini doğrudan etkilemektedir. Dolayısıyla, günümüzün üretim firmalarında üretim hızını arttırmak amacı ile bir çok çalışma yapılmaktadır.

Son yıllarda firmalar, üretim sistemlerinin tamamında veya belirli bir kısmında, üretkenliği, güvenliği, kaliteyi ve esnekliği arttırmak amacı ile endüstriyel robot kullanmayı tercih etmektedir. Robot kullanımı aynı zamanda birim ürün üretim zamanını azaltmaya yani çıktı miktarını arttırmaya ve tutarlılığı sağlamaya yardımcı olmaktadır. Yukarıda da öneminden bahsedilen üretim hızını etkileyen en önemli faktörlerden bazıları, üretim sürecinde gerekli olan parça yükleme, boşaltma ve transfer operasyonlarıdır. Üretim hatlarında, birçok makinede işlem görmesi gereken parçalar olmaktadır ve bu parçaların makinelere yüklenmesi, makinelerden boşaltılması ve makineler arası transferi gibi süreçler gerçekleştirilmelidir. Firmalar, bahsedilen süreçleri gerçekleştirebilmek amacıyla yapılan malzeme elleçleme operasyonlarında kullanılmak üzere robotları tercih etmektedir. Endüstriyel robotların üretim hattında kullanımının firmalarda oluşturduğu kanı, gün geçtikçe artan satış rakamları ile görülmektedir. Robot Endüstrisi Kuruluşu'nun (Robotic Industries Association) 2015 yılının ilk yarısı için yayınladığı raporda \$840 milyon değerinde 14.232 robot sipariş edildiği bilgisi yer almaktadır. 2014 yılının aynı periyoduna göre, robot kullanım miktarı %1 ve elde edilen gelir ise %7 artmıştır. Endüstride kullanılan robotların büyük çoğunluğu (% 40) malzeme elleçleme için kullanılmaktadır. Açıkça görülüyor ki, bir çok önemli endüstri dalında kullanılan robotlu ürünlerin piyasasında elleçleme robotları günden güne artan en büyük talep payına sahip kategoridir.

Üretim hattında kullanılan robotlar ile çözülmesi gereken yeni problemler ortaya çıkmaktadır. Robot kullanımında, robotun hangi hareket dizisini takip ederek çalışması gerektiği bilgisi robotların temin edildiği firmalar tarafından sağlanmamaktadır. Bu durumda, ro-

botların temin edildiği firmalar tarafından sağlanmamaktadır. Bu durumda, robot hareketlerini komuta edecek ve bunu firmanın istekleri doğrultusunda amaca hizmet edecek şekilde yapacak bir yöntem geliştirilmelidir. Robotların hareketlerinin çizelgesini belirledikten sonra bu çizelge robotlara yüklenen yazılım kodlarından faydalanılarak kullanılmaktadır.

Endüstriyel robotların kullanıldığı sistemlere, bir girdi stoğu, bir çıktı stoğu ve bir adet malzeme elleçleme robotu kullanılan üretim atölyelerine, literatürde robotik hücre denilmektedir. Sistemde kullanılacak robotlar, aynı anda tutabildikleri parça sayısına göre tek tutuculu (single gripper) veya çift tutuculu (dual gripper) olarak sınıflandırılmaktadır. m adet makineden oluşan tek tutuculu bir akış tipi robotik hücre Şekil 1.1’de gösterilmiştir.



Şekil 1.1: Robotik hücre.

Robotlar bahsedildiği üzere bir yazılım kodu ile takip edilmekte ayrıca uygulaması ve kontrolü daha kolay olduğu için literatürde birçok çalışmada döngüsel üretim ele alınmaktadır. Robotun aynı aktiviteler dizisini sürekli tekrar ederek parçaları transfer ettiği üretim şekli döngüsel üretimdir. Döngünün bir tekrarında sistemde üretilen parça sayısı döngünün niteliğini belirler. Eğer bir döngüde sistemde n adet parça üretiliyor ise bu döngüye n -birim döngüsü adı verilmektedir. Tek tip parçalar için hem basit olduğu hem de $m \leq 3$ durumunda optimal olduğu, $m \geq 4$ durumunda da iyi sonuçlar verdiği için genellikle literatürde 1-birim döngüler ele alınmıştır. Fakat değişik tip parça üretiminde $m = 2$ için bile optimal olmadıkları ispatlanmıştır [1]. Bu sebeple bu çalışmada daha genel bir tanımlı olan n -birim döngüler ele alınmıştır.

Özetle, bu tezde ele alınacak problem; tek tutuculu bir robotun kullanıldığı ve değişik tip parçaların üretildiği akış tipi bir üretim hattında döngüsel üretim stratejisi benimsenerek n -birim döngülerde çevrim zamanını en küçükleyecek şekilde robot hareket sıralaması ve parça üretim sıralamasını belirlemektir. Probleme değişik tip parça üretimi ele alınsa da literatürde daha geniş içeriğe sahip olan tek tip parça üretimi ele alınan çalışmalar da incelemiştir.

1.1 Tezin Amacı

Robotik malzeme taşıma sistemleri kullanan veya kullanmayı düşünen üretim firmalarının ve bu firmalara robotik hücre kurulumu yapan firmaların bu sistemden en yüksek verimi (en yüksek bir çıktı oranını) elde etmelerini sağlamak amacı ile bazı çizelgeleme problemlerinin çözülmesi gerekmektedir. Çizelgeleme problemlerini çözenin amacı, robotun parça yükleme ve boşaltma aktivitelerini hangi sıra ile yapması gerektiği, üretilen parçaların hangi sıra ile üretileceği gibi sorulara cevap verebilmektir. Çalışmanın sonunda, robotlara entegre olabilecek şekilde uygulaması ve denetlemesi kolay olduğu için geliştirilecek olan yöntemlerin bilgisayar kodu haline getirilmesi planlanmaktadır. Robot hareketleri ve parça çizelgelemesini bir bütün halinde düşünülüp problemin çözülmesi böylece birim ürün üretim zamanını azaltılması da amaçlanmaktadır.

1.2 Literatür Araştırması

Modern üretim sistemlerinin kullanımının artması ile robotik hücrelerin etkin kullanımı amacıyla 1990'larda başlayan çalışmalar gün geçtikçe artmaktadır. Robotik Hücre Çizelgeleme literatüründe yapılan çalışmalarda amaçlanan, birim ürün üretme zamanını en küçüklemek dolayısıyla çıktı miktarını en büyüklemek üzere robot hareketlerinin ve üretilen parçaların çizelgelenmesidir. Literatürdeki birçok çalışmada tek tip parçanın üretildiği sistemleri ele alınmıştır. Bu tür sistemlerde, parça çizelgeleme problemi ortadan kalkmakta, tek problem olarak robot hareketlerinin çizelgelenmesi kalmaktadır. Literatürdeki neredeyse bütün çalışmalarda döngüsel çizelgeleme üzerine yoğunlaşmıştır. Döngüsel çizelgelemede amaç tekrar edilebilir hareketler dizisi olarak tanımlanan en iyi robot hareket döngüsünü bulmaktır.

Literatürdeki çalışmalar ilk olarak robotun tek tutucusunun bulunduğu ve bu tutucuda tek bir parça taşıyabildiği (tek tutuculu) robotik hücreler üstünde yoğunlaşmıştır. Bu tip robotik hücrelerde tek tip parça üretimi ele alınsa da gelişen teknoloji ve müşteri istekleri doğrultusunda farklı türde parçaların üretildiği, robotun ve üretim hücresinin değişik yapılarında ve teknolojik özelliklere sahip olduğu çeşitli problemler de çalışılmıştır.

Robotik hücre çizelgeleme problemleri birçok faktöre göre değişiklik göstermektedir. Bu faktörlerden başlıcaları; robotik hücrenin tesis içerisindeki yerleşimi, makineler arası ara stok alanının olup olmadığı, sistemdeki robot sayısı ve özellikleri, üretilecek parça tipi gibi sıralanabilir. Literatürde kategorizasyon yapılırken bu maddeler farklı içeriklere sahip olabilmektedir. Robot üretim tesisi içerisinde farklı konumlarda olabilir; doğrusal bir ray üstünde veya dairesel bir atölyenin merkezinde yer alabilir. Tek tip ya da farklı tip parçaların üretilmesi de bu grupta ifade edilir. Amaç fonksiyonu, bir çok çalışmada birim çevrim süresini minimize etmektir. Birim çevrim süresi 1-birim döngülerle limitlenebileceği gibi daha genel döngü tanımı olan n -birim döngüleri de kullanılabilir. Yayılma zamanı (makespan) en küçükleme amacını ele alan az sayıda çalışma da bulunmaktadır. Robotik hücre çizelgelemesi alanında yapılan literatür taraması makalelerinde bu fakörler ve daha fazlasını ele alarak yapılan kategorizasyonlar verilmiştir.

Robotik hücre çizelgelemesi Sethi vd. tarafından yapılan çalışma ile literatürde yerini almıştır [2]. Bu çalışmada, m makineli sistem için $m!$ adet 1-birim döngüsü bulunduğu gösterilmiştir. Ayrıca iki makineli robotik hücreler için 1-birimlik döngülerden birinin optimal olacağı ispatlanmış ve 3 makineli sistem için de optimal olabileceği gösterilmiştir.

Crama ve van de Klundert [3] tarafından yapılan çalışmada, 3 makine için 1-birim döngülerin optimal olduğu ispatlanmış ve m -makineli robotik hücre için en iyi 1-birim döngüyü veren bir dinamik programlama algoritması geliştirilmiştir. Brauner ve Finke [4], [5], $m \geq 4$ hücreler için n -birim döngülerin 1-birim döngülerden daha iyi sonuçlar verebileceğini göstermişlerdir. Dawande vd. [6] herhangi iki makine arasındaki transfer zamanının sabit varsayıldığı robotik hücre çizelgeleme problemini ele almıştır. Bu varsayım altında optimal 1-birim döngüsünü bulan bir algoritma geliştirilmiştir.

Sethi vd. [2] robotik hücre çizelgelemesinde birçok konuyu ilk ele alan çalışma olmuştur. Değişik tip parça üretimi de ilk defa bu çalışmada yer almıştır. 2 ve 3 makineli sistemleri

ele alan bu çalışmada problemin karmaşıklığı üzerinde çalışmalar yapılmıştır. Robot hareket çevrimi sabitlenip ve her bir parça için talep değerleri verildiğinde, En Küçük Parça Kümesi (MPS:Minimal Part Set)'ndeki optimal parça sıralamasını belirleyen polinom zamanlı bir algoritma sunulmuştur. Üretilen parçalar literatürde en küçük parça kümesi adı verilen yaklaşımla belirlenirler. Örneğin, talep miktarları parça A için 40, parça B için 35, parça C için 25 olarak verilmiş ise, buna karşılık gelen en küçük parça kümesi A'dan 8, B'den 7 ve C'den de 5 adet olmak üzere toplam 20 parçadan oluşmaktadır. Dolayısıyla bu örnek için $n = 20$ 'dir ve bu 20 parça döngüsel olarak tekrar tekrar üretilecektir.

Bilge ve Ulusoy [7] m -makinelikli esnek bir üretim sistemi için otomatik güdümlü araçların çizelgelemesi hakkında çalışmışlardır. Sriskandarajah vd. [8] m -makinelikli bir robotik hücrede verilen robot hareket sıralaması için parça sıralaması problemini karmaşıklık sınıflarına ayırmışlardır. Bu sınıflandırmanın sonucunda olası $m!$ tane 1-birim döngü için parça sıralaması probleminin, $(2m - 2)$ tanesinin polinom zamanlı çözülebildiği ve geriye kalan problemlerin NP-tam oldukları belirtilmiştir. Aneja ve Kamoun [9], 2 makinelikli sistemler için robot hareket ve parça üretim sıralamasını Gezgin Satıcı Problemi'nin (Travelling Salesman Problem:TSP) özel bir hali olarak bulan $O(n \log n)$ karmaşıklığına sahip bir algoritma geliştirmişlerdir.

3 makinelikli bir sistem için problemin karmaşıklığını inceleyen Hall vd. [10] robot hareket sıralaması ve parça üretim sıralamasını belirleyen ve döngü zamanını en küçükleyen bir yöntem geliştirmişlerdir. Hurink ve Knust [11] bir adet robotun kullanıldığı akış tipi üretim sistemleri için çizelgeleme problemlerinin NP-zor olduğunu ispatlamış ve tabu arama algoritması önermiştir.

Soukhal ve Martineau [12] değişik tip parçaların üretildiği sistemler için tam sayılı bir model ve parça sıralamasını belirlemek amacıyla genetik algoritma geliştirmiştir. Carlier vd. [13] aynı problem için makinelerdeki bekleme zamanını (blocked time) da dikkate alarak yayılma zamanını en küçükleyen bir ayrışma algoritması geliştirmişlerdir. İlk olarak parça sıralamasını belirleyip, daha sonra bu parça sıralamasını kullanarak robot hareket sıralamasını belirleyen bir yöntem geliştirmişlerdir. Ayrıca bu çalışmada beş farklı altı altsınır geliştirilmiş ve algoritma performansı bu altsınır değerleri kullanılarak karşılaştırmıştır. Zahrouni ve Kamoun [14] değişik tip parçaların üretildiği 3 makinelikli bir akış tipi üretim sistemi için NEH (Nawaz, Ensore ve Ham [15]) algoritmasından ilham alarak bir çözüm kurucu algoritma geliştirmişlerdir.

Batur vd. [16] tarafından yapılan çalışmada, 3 farklı ürünün üretildiği, 2 makinenin olduğu bir sistemde üretim esnekliğinin etkisi gösterilmiştir. Gezgin Satıcı Probleminin bir varyasyonu olan bir formülasyon geliştirilmiştir. Esnekliğin avantajı olarak tüm makineler, tüm işlemleri yapabiliyor bu yüzden toplam işlem zamanları makinalara bölüştürülebilmektedir. İlk durumda her parça işlem süresi bir makinede tamamlamak zorunda fakat ikinci durumda birinci parça için iki ayrı makineye de işlem süresi dağıtılmıştır. İkinci durumda, robot fazladan hareket etmek zorunda olsa da makinelerdeki bekleme zamanını azalttığı için makineler daha etkin kullanılmıştır. Sonuçta ikinci durumda ilk duruma göre döngü zamanı %17 azalmıştır. Bu sonuç ile beraber üretimde esnekliğin döngü zamanına etkisi gösterilmiştir.

Son yıllarda yapılan çalışmalarda ise, Fazel Zarandi vd. [17] makine ayar zamanları ve boşaltma/yükleme zamanlarını dikkate alarak 2 makineli bir sistem ele almışlardır. Verilen parça üretim sıralamasını kullanarak optimal robot hareket sıralamasını bulan Karma Tam Sayılı doğrusal bir model geliştirmişlerdir. Ek olarak, büyük çaplı problemlerin çözümü için dal sınır algoritması ve tavlama benzetimi algoritması önermişlerdir. Son olarak, Atabak vd. [18] üretim sistemini etapların oluşturduğu ve her bir etapta m adet makine bulunan bir problem ele almışlardır. Yine transferden bir robotun sorumlu olduğu ve değişik tip parçaların üretildiği bir sistem için Karma Tam Sayılı doğrusal bir model geliştirmişlerdir. Büyük çaplı problemlerin çözümü için tavlama benzetimi algoritması önermişlerdir.

Literatürde bulunan çalışmalardan ele alınan problem ile daha yakından alakalı olanların detayları Çizelge 1.1'de görülebilir. Bu çizelgede mevcut çalışmanın özellikleri de belirtilmiş ve literatürden farkı irdelenmiştir. Yapılan tüm literatür araştırmalarının bir sonucu olarak oraya konulmuş olan tablo halinde başlıca çalışmaların bir nevi özeti verilmiştir. Çizelge 1.1'de de görüldüğü üzere bu tezde, literatürde çalışılmayan bir problem ele alınmıştır. Literatürde, değişik tip parça üreten m makineli bir sistem için döngü zamanını en küçükleyen bir problem ile karşılaşılmamıştır.

Çizelge 1.1: Değişik tip parça üretim problemi için literatür özeti.

*Çalışma	*MS	*Döngü*AF	*RHS	*PS	*ÇY	*Temel Katkı	
Sethi vd. [2]	2,3	1- birim	Döngü zamanı	Verilmiş ✓	-	Gilmore ve Gomory algoritması uygulanmış	
Hall vd. [10]	3	1- birim	Döngü zamanı	Verilmiş ✓		RHS bulma probleminin NP-tam olduğu ispatlanmış	
Aneja ve Kamoun [9]	2	n - birim	Döngü zamanı	✓	✓	Ayrı $O(n \log n)$ karmaşıklığına sahip bir algoritma geliştirilmiştir	
Soukhal vd. [12]	M	-	Yayıma zamanı	✓	✓	Eş zamanlı	Problemin özel durumu için karmaşıklık analizi yapılmış ve NP-zor yaklaşımı ispatlanmış
Carlier vd. [13]	3,4,5	-	Yayıma zamanı	✓	✓	Eş zamanlı	RHS için dal sınırı ve genetik algoritma geliştirilmiştir
Fazel Zarandi vd. [17]	2	n - birim	Döngü zamanı	✓	✓	Eş zamanlı	Farklı çalışma hızlarına sahip 2 makinenin bulunduğu bir sistem için matematiksel model ve tavlama benzetim algoritması geliştirilmiştir
Atabak [18]	M	-	Yayıma zamanı	✓	✓	Eş zamanlı	m tane makinenin bulunduğu bir sistem için matematiksel model ve tavlama benzetim algoritması geliştirilmiştir
Mevcut Çalışma	M	n - birim	Döngü zamanı	✓	✓	Eş zamanlı	Tam sayılı bir matematiksel model ve Hibrit (Genetik + Tabu Arama) meta-sezgisel algoritma geliştirilmiştir

* MS: Makine sayısı, RHS: Robot Hareket Sıralaması, PS: Parça Sıralaması, AF: Amaç Fonksiyonu, ÇY: Çözüm Yaklaşımı

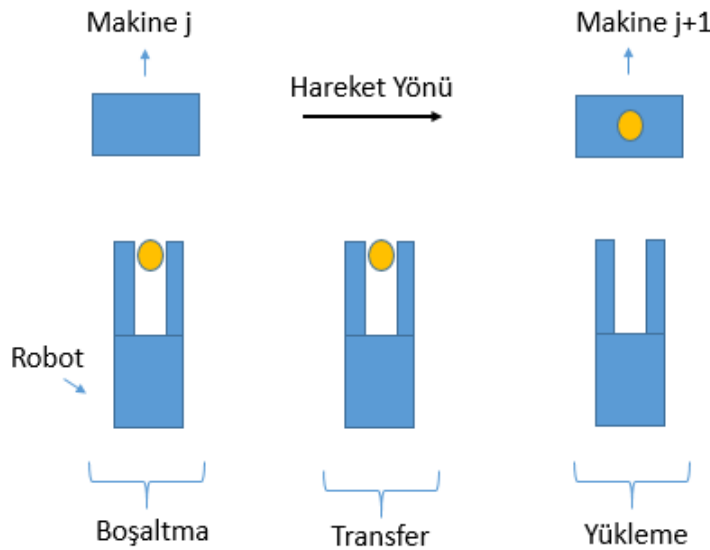
1.3 Problem Tanımı

Ele alınan sistemde m adet makine olduğu ve bu makinelerde akış atölyesi şeklinde üretilmesi gereken n adet parça olduğu varsayılmaktadır. Bu n adet parça tekrarlı bir şekilde üretilecektir. Yani bir n adetlik parti tamamlandığında yeni bir n adetlik parti üretilecektir. Bu durum Tam Zamanında Üretim (JIT: Just In Time) yapan firmaların üretim şeklidir. Ara stokları düşürmek amacıyla üretilecek parçalar küçük partiler halinde üretilirler. Üretilen parçalar daha önce de bahsedilen en küçük parça kümesi adı verilen yaklaşımla belirlenirler.

Ele alınan problemde verilmek istenen kararlar; üretilecek parçaların üretim sırasının ve robot hareket sırasının belirlenmesidir. Bu iki karar da problemin amaç fonksiyonu olan çevrim zamanının en küçüklenmesini etkilemektedir. Dolayısıyla her iki kararın beraberce ele alınması gerekmektedir. Problemin çözüm yöntemine geçmeden, literatürde döngünün ifade edilmesi için kullanılan aşağıdaki notasyonun tanımlanması gerekmektedir [3]:

A_j : Robotun j makinesini boşaltıp parçayı $j + 1$ makinesine taşıması ve bu makineyi yüklemesinden oluşan robot hareket kümesini ifade etmektedir ($j = (0, 1, 2, \dots, m)$. 0: girdi stoğu, $m + 1$: çıktı stoğu)

Aktivite tanımını oluşturan işlemler Şekil 1.2’de gösterilmiştir.



Şekil 1.2: Aktivite (A_j) tanımı.

Makineler arasında herhangi bir stok alanı olmadığı ve sistem bir akış atölyesi olduğu için robot bir makineyi boşalttığı anda yapabileceği tek hareket üzerindeki parçayı bir sonraki makineye yüklemektir. Bu sebeple verilen aktivite tanımı olası bütün robot hareket döngülerini ifade etmek için yeterlidir. Örneğin $A_0A_2A_3A_1$ 3 makineli bir sistemde olabilecek 1-birim döngülerinden bir tanesini göstermektedir.

Geliştirilen matematiksel modeldeki temel mantık bu aktivitelerin sıralanmasıdır (veya pozisyonlara atanmasıdır). Bu yapılırken aynı zamanda her aktiviteye bir başlangıç zamanı tayin edilecektir. Üretilmesi gereken n adet parçanın tüm makinelerde işlem gördükten sonra çıkış stoğuna getirilmesi ile üretimi tamamlanmaktadır. Bu sebeple, m makineli bir sistemde bütün A_j $j = (0, 1, 2, \dots, m)$ aktiviteleri n defa tekrarlanmalıdır. Verilen aktivite tanımında da görüldüğü üzere $(m + 1)$ adet aktivite yapılması gerekmektedir. Dolayısıyla, bu aktivitelerin $n(m + 1)$ adet pozisyona yerleştirilmesi gerekmektedir. Döngüsel üretim yapıldığı için, sıralama belirlendikten sonra aynı aktivite dizisi tekrar edilecektir. Bu sebeple, ilk aktivitenin A_0 olması literatürde genel kabul gören ve genelliği bozmayan bir varsayımdır. Bu çalışmada da bütün döngülerin A_0 aktivitesi ile başladığı varsayılmaktadır. Aktivite sıralamasının yanında bu aktivitelerin başlangıç zamanlarının da belirlenmesi gerekmektedir.

Diğer taraftan, aktiviteler pozisyonlara rastgele bir şekilde yerleştirilemezler. Örneğin, eğer $j + 1$ makinesi dolu ise robot j makinesini boşaltamaz. Yani A_j aktivitesini yapamaz. Benzer şekilde j makinesi boşsa robot A_j aktivitesini yapamaz. Yani zaten boş olan bir makineyi tekrar boşaltamaz. 1-birim döngülerin ele alındığı çalışmalarda robot aktivitelerinin herhangi bir permütasyonu olurlu bir döngü verirken bu çalışmada n -birim döngüler ele alındığı için bu ifade doğru değildir. Probleme özgü literatürde verilen olurluluk şartları aşağıda verilmiştir [8] ve bu şartlar modelde kısıtlar sayesinde sağlanmaktadır.

OLURLULUK ŞARTLARI:

- A_j , $j = (0, 1, 2, \dots, m)$ aktiviteleri n defa tekrarlanmalıdır.
- Herhangi iki A_j , $j = (1, 2, \dots, m - 1)$ aktiviteleri arasında mutlaka bir A_{j+1} ve bir A_{j-1} aktiviteleri yapılmalıdır.
- Herhangi iki A_0 arasında mutlaka bir A_1 aktivitesi yapılmalıdır.
- Herhangi iki A_m arasında mutlaka bir A_{m-1} aktivitesi yapılmalıdır.

Örnek 1: 3-makinelı bir robotik hücrede dört farklı parçanın üretiminin gerçekleştiği varsayalım. Yükleme/boşaltma ve transfer zamanları $\varepsilon = 1$ ve $\delta = 1$ olsun. İşlem zamanları ise Çizelge 1.2’de verilmiştir. Verilen problemin optimal robot hareket sıralaması (σ_{RMS}) ve parça sıralaması $\sigma_{PS} = \{4, 1, 2, 3\}$ bir sonraki bölümde anlatılacak olan matematiksel modelden elde edilmiştir. Her bir aktiviteye ait başlangıç zamanları ve σ_{RMS} Çizelge 1.3’te verilmiştir.

Çizelge 1.2: İşlem zamanları.

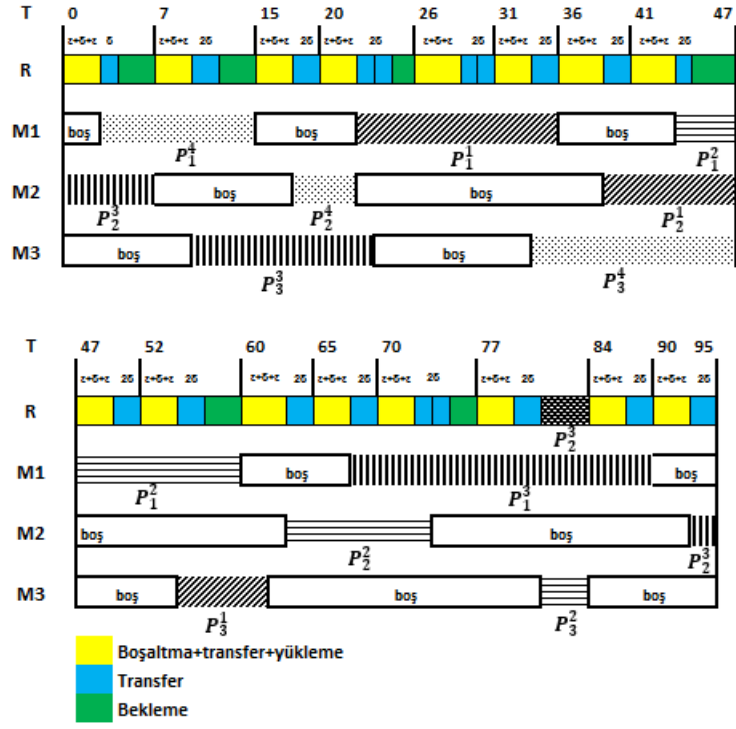
m\n	1	2	3	4
1	13	16	22	12
2	7	11	9	5
3	8	4	14	13

Çizelge 1.3: Robot hareket sıralaması ve aktivite başlangıç zamanları.

σ_{RMS}	A_0	A_2	A_1	A_0	A_3	A_2	A_1	A_0	A_3	A_2	A_1	A_0	A_3	A_2	A_3	A_1
T	0	7	15	20	26	31	36	41	47	52	60	65	70	77	84	90

Tüm çözümü görselleştirmek amacı ile çizilen Gant şeması Şekil 1.3’te verilmiştir. Transfer zamanı, yükleme/boşaltma zamanları, robotun bekleme zamanları, makinelerin parçalar üstündeki işlem zamanları ve boş zamanları gant şemasında gösterilmiştir. A_i aktivitesi, ε (M_i makinasını boşaltma zamanı) + δ (transfer zamanı M_i ’den M_{i+1} ’e) + ε (M_{i+1} makinasını yükleme zamanı) bu işlemlerden ibarettir, bu sırada M_{i+1} ’de parçanın işlem göreceği zaman başlamıştır. Şekil 1.3’de, robot ve makineler; R , M_1 , M_2 ve M_3 olarak gösterilmiştir.

Şekil 1.3’de görüleceği üzere tanımlanan döngüde başlangıçta M_2 ’de söz konusu parçanın işlemi devam etmektedir. İlk aktivite A_0 ile başlayan döngüde, Transfer(δ) ve yükleme/boşaltma (ε) zamanlarının ardından M_1 ’de parça sıralamasında ilk sırada olan parça 4 işlenmeye başlanmıştır. İkinci aktivite olan A_2 ’nin başlangıç zamanı 7 olarak verilmiştir. Çünkü parça 3’ün işlemi henüz bitmemiş ve robot işlem bitene kadar beklemiştir. M_2 ’de işlemi biten parça bu anda boşaltılır ve M_3 ’e transfer edilir ve yüklenir. M_3 ’e yüklenen parçanın işlenmesi başlamıştır. Üçüncü aktivite A_1 ’in başlaması için makine 1’de parça 4’ün işleminin bitmesi gerekmektedir. M_1 ’e 3’de işlenmesi başlanan parça 4’ün M_1 ’deki işlem zamanı 12 ve 15’de işlenmesi bitmiştir. Bu anda M_1 ’i boşaltıp, M_2 ’yi yüklemektedir. Bu şekilde devam eden döngüde orataya çıkacak robot ve makinelerin durumu Şekil 1.3’de verilmiştir.



Şekil 1.3: Gant şeması.

2. MATEMATİKSEL MODEL

Döngü zamanını en küçükleyecek robot hareket sıralaması ve üretilecek parçaların üretim sıralamasını eş zamanlı bulması için geliştirilen formülasyonda kullanılan parametreler, karar değişkenleri ve matematiksel model aşağıda verilmiştir:

Parametreler:

$N = \{1, 2, \dots, n\}$: Parça kümesi

$M = \{0, 1, 2, \dots, m\}$: Makine kümesi

$P = \{1, 2, \dots, n(m+1)\}$: Pozisyon Kümesi

P_m^n : $n \in N$ parçasının makine $m \in M$ 'deki işlem süresi

ε : Yükleme/boşaltma süreleri

δ : İki ardışık makine arasındaki transfer süresi

B : Büyük bir sayı

Karar Değişkenleri:

$$x_{jl}^h = \begin{cases} 1, & \text{Eğer aktivite } A_j \text{ 'nin } l. \text{ tekrarı pozisyon } h \in P \text{ 'ye atanır ise,} \\ 0, & \text{Diğer durumlarda} \end{cases}$$

$$z_{jl}^n = \begin{cases} 1, & \text{Eğer aktivite } A_j \text{ 'nin } l. \text{ tekrarı } n \in N \text{ parçasına ait ise,} \\ 0, & \text{Diğer durumlarda} \end{cases}$$

$$y_{jlm} = \begin{cases} 1, & \text{Eğer aktivite } A_j \text{ 'nin } n \in N \text{ parçasına ait } l. \text{ tekrarı pozisyon } h \in P \text{ 'ye atanır ise} \\ 0, & \text{Diğer durumlarda} \end{cases}$$

T_h : $h \in P$ pozisyonuna atanan aktivitenin başlangıç zamanı

CT : Döngü zamanı

Min CT

Öyle ki

$$\sum_{h=1}^p x_{jl}^h = 1 \quad \forall j \in \mathcal{M}, \forall l = 1 \dots k \quad (2.1)$$

$$\sum_{l=1}^k \sum_{j=0}^m x_{jl}^h = 1 \quad \forall h \in \mathcal{P} \quad (2.2)$$

$$\sum_{l=1}^k z_{jl}^n = 1 \quad \forall j \in \mathcal{M}, \forall n \in \mathcal{N} \quad (2.3)$$

$$\sum_{n=1}^k z_{jl}^n = 1 \quad \forall j \in \mathcal{M}, \forall l = 1 \dots k \quad (2.4)$$

$$y_{jlmn} \geq x_{jl}^h + z_{jl}^n - 1 \quad \forall j \in \mathcal{M}, \forall n \in \mathcal{N}, \forall l = 1 \dots k, \forall h \in \mathcal{P} \quad (2.5)$$

$$x_{jl}^h \geq y_{jlmn} \quad \forall j \in \mathcal{M}, \forall n \in \mathcal{N}, \forall l = 1 \dots k, \forall h \in \mathcal{P} \quad (2.6)$$

$$z_{jl}^n \geq y_{jlmn} \quad \forall j \in \mathcal{M}, \forall n \in \mathcal{N}, \forall l = 1 \dots k, \forall h \in \mathcal{P} \quad (2.7)$$

$$T_h \geq T_r + 2\varepsilon + \delta + \sum_{l=1}^k \sum_{j=0}^m \sum_{n=1}^k y_{jlmn} \cdot P_j^n + B \left(\sum_{l=1}^k x_{jl}^h + \sum_{l=1}^k x_{(j-1)l}^r - 2 \right) \quad \forall h, r \in \mathcal{P} : h > r, \forall j \in \mathcal{M} \quad (2.8)$$

$$CT + T_h \geq T_r + 2\varepsilon + \delta + \sum_{l=1}^k \sum_{j=0}^m \sum_{n=1}^k y_{jlmn} \cdot P_j^n + B \left(\sum_{l=1}^k x_{jl}^h + \sum_{l=1}^k x_{(j-1)l}^r - 2 \right) \quad \forall h, r \in \mathcal{P} : h < r, \forall j \in \mathcal{M} \setminus \{0\} \quad (2.9)$$

$$T_{h+1} \geq T_h + 2\varepsilon + \delta + \delta|q - (j+1)| + B \left(\sum_{l=1}^k x_{jl}^h + \sum_{l=1}^k x_{ql}^{h+1} - 2 \right) \quad \forall j, q \in \mathcal{M} : j \neq q, \forall h \in \mathcal{P} \setminus \{p\} \quad (2.10)$$

$$CT \geq T_h + 2\varepsilon + \sum_{j=0}^m \sum_{l=1}^k (j+2) \delta \cdot x_{jl}^h \quad \forall n \in \mathcal{N}, \forall h \in \mathcal{P} \quad (2.11)$$

$$T_1 = 0 \quad (2.12)$$

$$\sum_{h=1}^p h \cdot x_{jl}^h \leq \sum_{h=1}^p h \cdot x_{js}^h \quad \forall j \in \mathcal{M}, \forall l = 1 \dots (k-1), s = (l+1) \dots k \quad (2.13)$$

$$\sum_{g=1}^k \sum_{h=r+1}^{s-1} x_{(j-1)g}^h \geq 1 - B(2 - x_{jl}^r - x_{j(l+1)}^s) \quad \forall r, s \in \mathcal{P} : r < s, \forall j \in \mathcal{M} \setminus \{0\}, \forall l = 1 \dots (k-1) \quad (2.14)$$

$$\sum_{g=1}^k \sum_{h=r+1}^{s-1} x_{(j+1)g}^h \geq 1 - B(2 - x_{jl}^r - x_{j(l+1)}^s) \quad \forall r, s \in \mathcal{P} : r < s, \forall j \in \mathcal{M} \setminus \{m\}, \forall l = 1 \dots (k-1) \quad (2.15)$$

$$\sum_{n=1}^k y_{011n} = 1 \quad (2.16)$$

$$\sum_{l=1}^k \sum_{s=h+1}^{r-1} y_{(j+1)l sn} \geq 1 - B \left(2 - \sum_{l=1}^k y_{jllm} - \sum_{l=1}^k y_{(j+1)l r} \right) \quad \forall j \in \mathcal{M} \setminus \{m\}, \forall n, t \in \mathcal{N} : n \neq t, \forall h, r \in \mathcal{P} : h < r \quad (2.17)$$

$$x_{jt}^h, z_{jl}^n, y_{jllm} \in \{0, 1\} \quad \forall j \in \mathcal{M}, \forall n \in \mathcal{N}, \forall h \in \mathcal{P}, \forall l = 1 \dots k \quad (2.18)$$

$$T_h, CT \geq 0 \quad \forall h \in \mathcal{P} \quad (2.19)$$

Bu formülasyonda, Kısıt (2.1) herhangi bir aktivitenin herhangi bir tekrarının yalnızca 1 pozisyona atanması şartını sağlamaktadır. Her pozisyona bir aktivitenin bir tekrarının atanması gerektiği şartı kısıt (2.2) ile sağlanmaktadır. Üretilecek her bir parçaya ait her aktivite bir defa tekrar edilmeli şartı kısıt (2.3) ile sağlanmaktadır. Kısıt (2.4) ile her bir aktivitenin her bir tekrarının yalnızca 1 parçaya ait olması sağlanmaktadır. Kısıtlar (2.5), (2.6) ve (2.7) x_{jl}^h ve z_{jl}^n değişkenlerinin çarpımına eşit olan y_{jlm} değişkeninin doğrusal kısıtlarla çarpım değerini sağlaması için yazılmıştır. Ardışık A_{j-1} ve A_j aktivite başlangıç zamanları arasında yükleme/boşaltma zamanı $(2\varepsilon + \delta)$ ve j makinesine atanan parçanın işlem zamanı kadar sürenin geçmesi gerektiği şartı (2.8) ile sağlanmaktadır. A_{j-1} başladıktan sonra $2\varepsilon + \delta$ kadar zaman sonunda j makinesine yüklenmesi tamamlanır ve j makinesinde parçanın işlemi bitene kadar beklenmeli, en az bu süre kadar sonra A_j aktivitesi başlayabilir. Eğer A_{j-1} aktivitesi A_j aktivitesinden daha sonra gerçekleştirildi ise döngü A_j aktivitesi yapılmadan sona erecektir. Bu durumda başlangıç zamanları arasında bir önceki kısıta ilave olarak döngü zamanı kadar daha zaman geçmesi gerekmekte ve bu şart (2.9) ile sağlanır. Kısıt (2.10) ardışık pozisyonlara atanan aktivitelerin başlama zamanları arasındaki ilişkiyi belirlemek amacıyla yazılmıştır. Bir önceki kısıttan farklı olarak ardışık aktiviteler olduğu için aradaki transfer zamanı $\delta|q - (j + 1)|$ kadar zaman daha geçmeli şartı da bu kısıt ile sağlanmaktadır. Döngü zamanı gerçekleştirilen son aktivitenin tamamlanma zamanından büyük olmalıdır. Tamamlanma zamanına parçayı söz konusu makineye yükledikten sonra giriş stoğuna dönüş zamanı da eklenir ve bu şart da kısıt (2.11) ile sağlanmaktadır. Kısıt (2.12), ilk atanan aktiviteye başlangıç zamanı olarak 0 atamaktadır. Kısıt (2.13) ile herhangi bir aktivitenin daha önceki bir tekrarının daha önceki bir pozisyona atanma şartı sağlanır. Bir döngünün olurlu olabilmesi için A_j aktivitesinin herhangi iki tekrarı arasında mutlaka bir A_{j-1} ve bir A_{j+1} aktiviteleri yer almalıdır. Bu olurluluk şartı ise kısıt (2.14) ve kısıt (2.15) yardımı ile sağlanmaktadır. Problemin varsayımlarından biri olan tüm döngülerin A_0 aktivitesi ile başlama şartını sağlamak amacıyla kısıt (2.16) yazılmıştır. Diğer bir olurluluk şartı ise kısıt (2.17) ile sağlanmaktadır. Buna göre, n parçasına ait A_j aktivitesinden sonra gelen A_{j+1} aktivitesi de n parçasına ait olmalıdır. Kısıt (2.18) ve (2.19) karar değişkenlerine ait işaret kısıtlarıdır.

Geliştirilen matematiksel model GAMS'te kodlanmış ve CPLEX 12.6.2 kullanılarak farklı makine ve parça sayılarına sahip veri kümeleri türetilerek test edilmiştir. Test örneklerine ait makine ve parça sayıları ve çözüm süreleri Çizelge 2.1'de verilmiştir.

Çizelge 2.1'de δ sütununda S ve L ile gösterilen satırlar δ 'nın iki farklı değerini göstermektedir. Farklı δ değerlerine göre ayrıştırılmasının sebebi çözüm üzerindeki δ 'nın çözüme etkisinin daha net anlaşılabilmesidir. Eğer çok büyük δ değeri kullanılırsa probleme ait makinelerdeki işlem zamanlarının etkisi ortadan kalkmakta ve belli bir döngü çeşidi optimal olmaktadır. Benzer bir durum çok küçük bir δ değeri kullanıldığında ortaya çıkmaktadır. Bu defa δ değerinin çözüm üzerindeki etkisi ortadan kalkmakta ve belli bir döngü çeşidi optimal olmaktadır. Bu durumda işlem zamanları, makine ve parça sayıları dikkate alınarak tek başına çözümü belirlemeyen δ değeri bulunması amaçlanmıştır. Çözümü bu şekilde etkilemeyen δ değerini bulmak amacı ile aşağıda gösterilen aralıktan yararlanılarak iki farklı δ değeri türetilmiştir. Bu aralık, birçok defa yapılan denemeler sonucu ortaya çıkmıştır. Her bir problem için bir büyük (L) bir de küçük (S) δ değeri türetilmiştir.

P_{ij} : i makinesinde j parçasının işlem zamanı

δ_L : üretilen büyük δ değeri

δ_S : üretilen küçük δ değeri

$$\delta_L = U((P_{ij}/(m \times n))/(6 \times (m - 1)), (P_{ij}/(m \times n))/(2 \times (m - 1)))$$

$$\delta_S = U((P_{ij}/(m \times n))/(4 \times (m - 1)), (P_{ij}/(m \times n))/(m - 1))$$

Matematiksel modelin çözüm sürecinde, çözücü bulduğu alt sınır değeri ile elde ettiği en iyi olurlu çözüm arasında bulunan yüzde sapma değerini hesaplar. 1 saat zaman limiti verilerek alınan bu çözümlerde elde edilen bu yüzde sapma değerleri çözücünün verdiği değerlerdir.

Çizelge 2.1: Matematiksel modelden elde edilen sonuçlar.

Problem	δ	Parça (n)	Makine Döngü (m)	Döngü Zamanı	Altsınır	Çözüm Zamanı	Sapma (%)
1	S	3	3	3557	3557	2,56	0,00
	L	3	4	6695	6695	2,28	0,00
2	S	3	5	4532	4532	77,34	0,00
	L	3	5	10522	10522	44,26	0,00
3	S	3	6	6572	6572	1384,75	0,00
	L	3	6	13930	13930	1808,18	0,00
4	S	4	3	7095	4590	ZL	54,58
	L	4	3	13464	2783	ZL	383,79
5	S	4	4	5070	5070	79,61	0,00
	L	4	4	7840	7840	84,89	0,00
6	S	4	5	6705	844	ZL	694,43
	L	4	5	9687	7282	ZL	33,03
7	S	5	3	10715	6020	ZL	77,99
	L	5	3	19718	11620	ZL	69,69
8	S	5	4	ÇY	ÇY	ZL	ÇY
	L	5	4	ÇY	ÇY	ZL	ÇY
9	S	5	5	10058	1190	ZL	745,21
	L	5	5	15366	1627	ZL	844,44
10	S	6	3	13998	802	ZL	1645,39
	L	6	3	18932	1162	ZL	1529,26
Altsınır: Matematiksel modelin bulduğu altsınır değeri							

Çizelge 2.1’de görüldüğü üzere makine sayısı ve parça sayısı arttıkça çözüm süreleri artmakta ve belli bir noktadan sonra verilen zaman limiti (1 saat) tamamlandığında hala çözüm alınamamaktadır. Zaman limiti sınırına takılan problemler ZL ile gösterilmiş, verilen zaman limiti dolduğunda herhangi bir çözüm bulunamayan problemler ÇY ile gösterilmiştir. Elde edilen çözümlerin döngü zamanları ve çözücünün bulduğu altsınır değerleri verilmiştir. Bulunan altsınır değerleri, bulunan çözümlerin döngü zamanından bir hayli uzak olduğu çözümler de görülmektedir. Özellikle problem 9 ve 10 için bu durum gözlemlenmektedir. Bu gibi durumlarda bulunan çözümün kalitesi hakkında bir fikre sahip olunamamaktadır. Alınan sonuçlar, küçük boyutlu problemlerde dahi geliştirilen matematiksel modelin çözüm vermediğini göstermektedir. Matematiksel modelin performansını iyileştirerek, çözülemeyen problemleri çözmek için çeşitli yollar araştırılmıştır. Geliştirilen modelin performansını artırmak amacı ile probleme özgü bir altsınır geliştirilerek, modele entegre edilmiş ve tekrar aynı problemler çözümlenerek performansa etkisi incelenmiştir.

2.1 Altsınır Hesaplanması

Literatürde döngü zamanı için tek tip parça ve 1-birim döngüleri ele alan problemlere ait bir altsınır aşağıdaki gibi verilmiştir [6]:

$$\max\{2(m+1)(\varepsilon + \delta) + \sum_{j=1}^m (\min(p_j, \delta)), \max_{1 \leq j \leq m} (p_j) + (4\varepsilon + 4\delta)\}$$

$p_j = j$ makinesindeki işlem zamanı

Bahsedildiği gibi tek tip ürün ve 1-birim döngü varsayımı altında geliştirilen bu altsınırın ele alınan problemde kullanılabilmesi için güncellenmesi gerekmektedir. Literatürde bulunan bu altsınır formülasyonundan yola çıkarak, değişik tip parça üretimi ve n -birim döngüleri ele alan problemler için aşağıdaki önteoremden bir altsınır hesaplanmaktadır.

Önteorem 1: Sistem, değişik tip parça üreten robotik hücre olsun. Bu durumda n -birim döngüler ele alındığında,

$$n \times \max\{2(m+1)(\varepsilon + \delta) + \frac{1}{n} \sum_{j=1}^m \sum_{i=1}^n \min(p_{ij}, \delta), 4\varepsilon + 4\delta + \frac{1}{n} \sum_{i=1}^n \max_{1 \leq j \leq m} (\hat{p}_{ij})\}$$

ifadesi döngü zamanı için bir altsınır tanımlar. Bu formülde, i parçasının j makinesindeki işlem zamanı p_{ij} ve j makinasında büyükten küçüğe doğru sıralanmış bütün p_{ij} değerleri \hat{p}_{ij} ile ifade edilir.

İspat:

Verilen altsınır formülasyonunda maksimum fonksiyonunun içinde iki kısım yer almaktadır. İlk kısım, robotun bir parçayı üretmek için yapması gereken minimum hareketlerin sonucunda ortaya çıkacak zamanı ifade etmektedir. İkinci kısım ise, döngü zamanının bir başka tanımı olan robotun aynı makineyi ardışık iki yüklemesi arasında geçen zamanı ifade etmektedir.

Robotun bir parçayı üretmek için yapması gereken minimum hareketlerin sonucunda ortaya çıkacak zamanı ifade ederken bir döngüde olması gereken işlemlerin asgari zamanları dikkate alınmaktadır. Herhangi bir parça, her bir makinede işlem görmesi gerektiği için bir döngüde her bir makinede yüklenme ve boşaltma işlemleri gerçekleştirilmelidir. Girdi stoğu yalnızca boşaltılır ve çıktı stoğu yalnızca yüklenir böylece ikisi bir makine

olarak düşünölmektedir. Bu durumda $m + 1$ adet makinede bu işlemler yapılacaktır. Bu işlemler dışında makineler arası transferin gerçekleşmesi gerekmektedir. Her bir makine (girdi ve çıktı stoğu dahil) arası transfer δ kadar zaman almaktadır. Bir döngüde asgari ardışık makineler arasında yapılan transfer işlemi $\delta(m + 1)$ ve en son çıktı stoğuna parça yükledikten sonra döngünün sonlanması için girdi stoğuna dönölmesi $\delta(m + 1)$ kadar zaman almaktadır. Sonuçta toplamda en az $2\delta(m + 1)$ transfer zamanı ortaya çıkmaktadır. Ayrıca, bir robot herhangi parçayı makineye yükledikten sonra ya o parçanın işlem zamanı kadar makinenin önünde beklemekte ya da bir sonraki makineye hareket etmektedir. Bu süre en az işlem zamanı ile transfer zamanının düşük olanı kadar sürebilir. Tüm bu işlemlerin sonunda yapılması gereken işlemlerin zamanı

$$2(m + 1)(\varepsilon + \delta) + \frac{1}{n} \sum_{j=1}^m \sum_{i=1}^n \min(p_{ij}, \delta)$$

şeklinde hesaplanır. Burada $2(m + 1)(\varepsilon + \delta)$ ifadesi sadece bir ürünün üretiminde gereken yükleme ve boşaltma işlemlerini içerdği için tüm parçaların işlem zamanlarını topladığımız ifadenin n 'e bölünmesi gerekmektedir. Diğer taraftan çevrim zamanı herhangi bir makinenin ardışık iki yüklemesi arasında geçen zamandan daha büyük olmalıdır. Max fonksiyonundaki ikinci ifade bir makinenin ardışık iki yüklemesi arasında geçen en düşük değeri ifade etmektedir. Örneğin j makinesindeki (M_j) ardışık iki yükleme arasında,

- M_j 'nin ε zamanda yüklenmesi ve işlem bittikten sonra yine ε zamanda boşaltılması
- M_{j+1} 'e δ zamanda transfer edilmesi
- Robotun makine M_{j-1} 'e 2δ zamanda hareket etmesi
- Bir sonraki parçayı ε zamanda boşaltması
- M_j 'ye δ zamanda transfer etmesi
- bu parçayı M_j 'ye ε zamanda yüklemesi gerçekleşmelidir.

Toplamda bu işlemler için $(4\varepsilon + 4\delta)$ kadar zaman geçmektedir. Bu zaman içerisinde hesaba katılması gereken bir unsur da M_j 'deki söz konusu parçanın işlem zamanıdır.

Değişik tip parçaların makinelerdeki işlem zamanları birbirlerinden farklıdır ve tüm parçalar tüm makinelerde işlem görmeli varsayımı vardır. Aynı anda bütün makineler dolu olduğunda, bir turda ardışık iki yükleme arası geçen zaman makinelerde yüklü olan parçaların yüklü oldukları makinelerdeki işlem zamanlarının en uzun olanı olacaktır. Fakat n -birim döngü ele alındığı için, bu durum n defa tekrar edecektir. Dolayısıyla bütün işlerin her bir makinedeki işlem zamanı büyükten küçüğe doğru sıralanarak \hat{p}_{ij} elde edilir. Daha sonra makine bazında sıralanmış bu işlem zamanları arasından bütün makinelerde en büyük olanı seçilir. Seçilen bu işlem zamanına, ikinci, üçüncü, ... n 'inci sıradaki işlem zamanları için bulunan en büyük işlem zamanı değerleri eklenir. Dolayısıyla n parça için toplam en büyük süre bulunmuş olur. Bu değer n 'e bölünerek bir parça için ortalama süre elde edilmiş olur. Sonuç olarak, $\frac{1}{n} \sum_{i=1}^n \max_{1 \leq j \leq m} (\hat{p}_{ij})$ denklemi elde edilir. ■

Altsınır hesaplanmasının daha net anlaşılması için aşağıdaki örnek verilmiştir.

Örnek 2: Çizelge 2.2'de probleme ait veriler verilmiştir. $n=6$ ve $m=4$ olan problemde $\varepsilon=1$ ve daha önce anlatılan aralıktan üretilen $\delta=80$ olarak alınmıştır.

Çizelge 2.2: Altsınır hesaplama örneği verileri.

n/m	1	2	3	4
1	120	103	57	50
2	15	46	133	109
3	102	72	119	117
4	134	62	137	108
5	145	27	26	7
6	63	148	81	11

Altsınır hesaplanması yapılmadan önce her bir makinedeki işlem zamanları Çizelge 2.3'te gösterildiği gibi büyükten küçüğe doğru sıralanmıştır.

Çizelge 2.3: Altsınır hesaplama örneği verilerinin sıralanmış hali.

1	2	3	4
145	148	137	117
134	103	133	109
120	72	119	108
102	62	81	50
63	46	57	11
15	27	26	7

Bu durumda, $\frac{1}{n} \sum_{i=1}^n \max_{1 \leq j \leq m} (\hat{p}_{ij}) \rightarrow \frac{1}{6} \sum_{i=1}^6 (148 + 134 + 120 + 102 + 63 + 27)$ olmaktadır.

Altsınır fonksiyonunun ilk kısmı;

$$2(4 + 1)(1 + 80) + \frac{1}{6} \sum_{j=1}^4 \sum_{i=1}^6 \min(p_{ij}, 80) = 1076,16$$

İkinci kısmı ise;

$$4 \times 1 + 4 \times 80 + \frac{1}{6} \sum_{i=1}^6 \max_{1 \leq j \leq 4} \{\hat{p}_{ij}\} = 457,67$$

olur.

Bu durumda sonucu robotun bir parçayı üretmek için yapması gereken minimum hareketlerin sonucunda ortaya çıkacak zaman belirlemekte ve sonuç $6 \times 1076,16 = 6457$ olmaktadır.

Çizelge 2.4: Altsınır eklendikten sonra elde edilen sonuçlar.

Problem	Delta	Parça (n)	Makine Döngü (m)	Döngü Zamanı	Altsınır	Çözüm Zamanı	Sapma (%)
1	S	3	3	3557	3557	2,48	0,00
	L	3	4	6695	6695	3,06	0,00
2	S	3	5	4532	4532	44,20	0,00
	L	3	5	10522	10522	87,11	0,00
3	S	3	6	6572	6572	2763,77	0,00
	L	3	6	13930	13930	1324,56	0,00
4	S	4	3	7095	6582	ZL	7,79
	L	4	3	13464	12810	ZL	5,11
5	S	4	4	5070	5070	118,53	0,00
	L	4	4	7840	7840	88,00	0,00
6	S	4	5	6705	6452	ZL	3,92
	L	4	5	9687	9632	ZL	0,57
7	S	5	3	10715	10170	ZL	5,36
	L	5	3	19718	19110	ZL	3,18
8	S	5	4	ÇY	ÇY	ZL	ÇY
	L	5	4	ÇY	ÇY	ZL	ÇY
9	S	5	5	10058	9771	ZL	2,94
	L	5	5	15366	15204	ZL	1,07
10	S	6	3	13998	12884	ZL	8,65
	L	6	3	18932	18376	ZL	3,03

Geliştirilen bu altsınır formülasyonu ile veri kümeleri için elde edilen tüm altsınır değerleri matematiksel modele ek bir kısıt olarak eklenerek yukarıda bahsedilen veri kümeleri kullanılarak testler tekrarlanmıştır. Elde edilen sonuçlar Çizelge 2.4’te verilmiştir. Görülebileceği gibi altsınır eklendikten sonra çözüm süreleri azalmış ve çözüm alınamayan problemler için 1 saat sonunda optimal çözüme olan uzaklıklar (sapma) azalmıştır. Ama yine de istenen performans değerleri sağlanamamış ve sezgisel bir yöntem geliştirilmesi kararlaştırılmıştır. Bir sonraki bölümde geliştirilen bu yöntemden bahsedilecektir.





3. SEZGİSEL YÖNTEM

Matematiksel modelin küçük çaplı problemlerde dahi çözüm verememesi sebebi ile sezgisel bir yöntem geliştirilmiştir. Ele alınan problemde verilmek istenen kararlar; üretilcek olan parçaların üretim sırası ve robot hareket sırasının belirlenmesidir. Bu iki ayrı kararın eş zamanlı verilmesi istenmektedir. Döngü zamanını en küçükleme amaçlandığında, her iki karar da birbirine göre şekil alır. Ayrı ayrı karar vermek en iyi çözüme ulaşma garantisini ortadan kaldırmaktadır. Fakat bu iki kararın kendilerine özgü yapıları nedeni ile ikisi için farklı metasezgisellerin kullanılmasına karar verilmiştir.

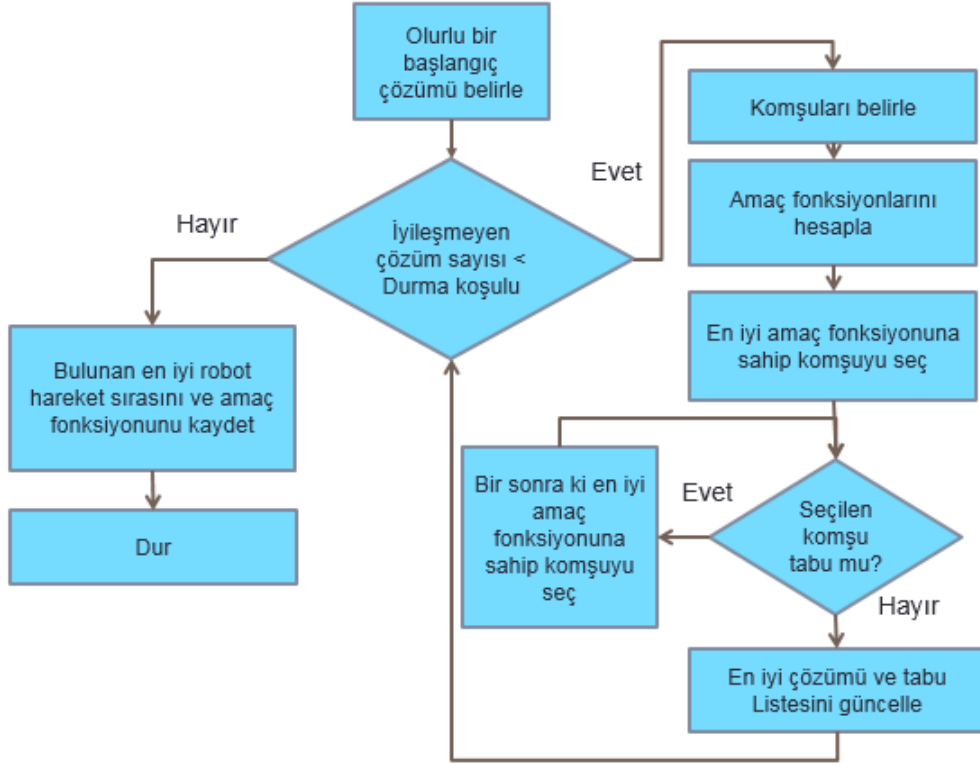
Her iki algoritmada da parça sıralamasını belirlemek için bir Genetik Algoritma geliştirilmiştir. Çizelgeleme problemlerinde Genetik Algoritmanın parça sıralamasını belirlemek için sıklıkla kullanıldığı bilinmektedir. Diğer taraftan, verilen bir parça sıralamasına karşılık gelen en iyi robot hareket sıralamasını belirlemek için Genetik Algoritma çok da uygun değildir. Bunun sebebi, Genetik Algoritmada yer alan çaprazlama kurallını uygularken robot hareket sıralamasının belirli olurluluk kurallarına uymasının gerekmesidir. Dolayısıyla çaprazlama esnasında bu olurluluğun kaybedilmesi durumu ortaya çıkabilmektedir. Olurlu olmayan bir hareket sıralamasını daha sonra tekrar olurlu hale getirmek de kolay olmayan bir işlemdir. Bu sebeple, robot hareket sıralamasını belirlemek için bir Tabu Arama algoritması geliştirilmiştir. Sonuçta, parça sıralamasını belirleyen bir Genetik Algoritma ve belirlenen parça sıralamasına karşılık gelen en iyi robot hareket döngüsünü belirleyen bir Tabu Arama Algoritmasının iç içe çalıştığı hibrit bir metasezgisel yöntem tasarlanmıştır.

3.1 *n*-birim Döngüler için Geliştirilen Algoritma

Problemin çözümüne yönelik iki sezgisel algoritma geliştirilmiştir. Bunlardan birincisi anlatılan problem tanımına uygun olarak *n*-birim döngüler arasında çözüm aramaktadır. Robot hareket sıralaması ve parça üretim sırasını belirlemek için kullanılan Tabu Arama Algoritması Genetik Algoritma bir sonraki bölümlerde verilmiştir.

3.1.1 Tabu arama algoritması

Geliştirilen Tabu Arama algoritmasının akış şeması Şekil 3.1’de verilmiştir. Tabu arama algoritmasında bulunan *Amaç fonksiyonlarını hesapla* adımında, genetik algorithmada bulunan parça sıralamasını kullanarak belirlenen robot hareket sıralamasını kullanılarak döngü zamanı hesaplanır.



Şekil 3.1: Tabu Arama Algoritmasının Akış Şeması.

Başlangıç çözümünün belirlenmesi:

Tabu Arama algoritmasının önemli unsurlarından bir tanesi de bir başlangıç çözümünün gerekmesidir. Başlangıç çözümü olarak kullanılmak üzere olurlu bir robot hareket sıralamasını belirlemek için bir algoritma geliştirilmiştir. Geliştirilen algoritmanın sözde kodu aşağıda verilmiştir. Bu algoritma A_0 aktivitesini ilk sıraya atadıktan sonra arkasına gelecek aktiviteleri tek tek rasgele seçerek yerleştirmektedir. Seçilen her yeni aktiviteden sonra Bölüm 1.3’te bahsedilmiş olan olurluluk şartlarının sağlanıp sağlanmadığı kontrol edilmektedir.

Algoritma 1 Robot Hareket Sıralaması için Başlangıç Çözümü Belirleme Algoritması

Girdi: k =parça sayısı, m =makine sayısı

Çıktı: Olurlu Robot Hareket Sıralaması

```
1:  $k(m+1)$  boyutunda bir vektör tanımla,  $i$  pozisyon indisi olsun.  $1 < i < k(m+1)$ 
2: döngü ( $v = 0$ ' den  $m$ 'ye ) yap
3:    $v$  boyutunda bir dizi tanımla
4:   döngü ( $i = 1$ 'den  $k(m+1)$ )'e yap
5:     eğer  $i = 1$  ise
6:        $P_1 \leftarrow 0$ 
7:       devam et
8:     değilse eğer  $i = 2$  ise
9:        $P_2 \leftarrow \text{rand}(1, m)$ 
10:    değilse eğer  $i > 2$  ise
11:       $v = \text{rand}(0, m)$ 
12:      eğer  $\text{dizi}(v)$ 'nin boyutu  $< k$  ise
13:        eğer  $v = 0$  ise
14:           $P_j = 0$  olan ilk  $j$  pozisyonu bul
15:          eğer  $\exists k : P_k = 1, j < k < i$  ise
16:             $P_i \leftarrow 0$ 
17:             $i \leftarrow i + 1$ 
18:             $\text{dizi}(0)$  boyutu  $\leftarrow \text{dizi}(0)$  boyutu+1
19:          son eğer
20:        değilse eğer  $v = m$  ise
21:           $P_j = m$  olan ilk  $j$  pozisyonu bul
22:          eğer  $P_j \neq m \forall j < i$  ise
23:             $P_i \leftarrow m$ 
24:             $i \leftarrow i + 1$ 
25:             $\text{dizi}(m)$  boyutu  $\leftarrow \text{dizi}(m)$  boyutu+1
26:          değilse eğer  $\exists k : P_k = m - 1, j < k < i$  ise
27:             $P_i \leftarrow m$ 
28:             $i \leftarrow i + 1$ 
29:             $\text{dizi}(m)$  boyutu  $\leftarrow \text{dizi}(m)$  boyutu+1
30:          son eğer
31:        değilse
32:           $P_j = v$  olan ilk  $j$  pozisyonu bul
33:          eğer  $P_j \neq v \forall j < i$  ise
34:             $P_i \leftarrow v$ 
35:             $i \leftarrow i + 1$ 
36:             $\text{dizi}(v)$  boyutu  $\leftarrow \text{dizi}(v)$  boyutu+1
37:          değilse eğer  $\exists k : P_k = v - 1$  and  $\exists h : P_h = v + 1, j < k, h < i$  ise
38:             $P_i \leftarrow v$ 
39:             $i \leftarrow i + 1$ 
40:             $\text{dizi}(v)$  boyutu  $\leftarrow \text{dizi}(v)$  boyutu+1
41:          son eğer
42:        son eğer
43:      değilse eğer  $\sum_v \text{dizi}(v)$  boyutu  $= k(m+1)$  ise
44:        döngüden çık
45:      son eğer
46:    son eğer
47:  son döngü
48: son döngü
```

Aşağıda parça sayısı $n = 4$ ve makine sayısı $m = 4$ olan bir problem için olurlu bir robot hareket sıralaması (D) verilmiştir. Bu döngü için $n(m + 1) = 20$ adet pozisyon gerekmektedir. Görülebileceği gibi herhangi iki A_j aktivitesi arasında bir A_{j-1} ve bir A_{j+1} aktivitesi yapılmıştır.

$$D = \{A_0, A_2, A_1, A_3, A_2, A_0, A_4, A_1, A_0, A_3, A_4, A_2, A_1, A_0, A_3, A_2, A_1, A_4, A_3, A_4\}$$

Komşuların belirlenmesi: Verilen başlangıç çözümünün komşularının belirlenmesi için aşağıdaki yöntem geliştirilmiştir.

- İlk aktiviteden sonuncuya doğru sırasıyla aktiviteleri ele al. Ele alınan aktivite A_j olsun:
- A_j 'nin bir sonraki tekrarını bul. Bu aktivite A_j^* olsun. Bu aktiviteyi bir önceki pozisyona kaydır ve ortaya çıkan hareket dizisini yeni bir komşu olarak kaydet. Bu işleme A_{j-1} veya A_{j+1} aktivitesine ulaşana kadar devam et (A_0 için sadece A_1 'i kontrol et, A_m için de sadece A_{m-1} 'i kontrol et).

Verilen D döngüsü için, ilk pozisyonda bulunan A_0 aktivitesiyle başlanır. Bu aktivitenin bir sonraki tekrarı 6'ncı pozisyondadır. Olurluluk şartlarına göre bu iki aktivite arasında bir adet A_1 aktivitesi bulunmalı şartı vardır. Bu yüzden A_1 aktivitesi arada kalacak şekilde ikinci A_0 aktivitesini ön pozisyonlara kaydırarak yer değiştirilebilir. Bu işlemler yapıldığında 2 yeni komşu şu şekilde türetilmiş olur:

$$D^1 = \{\mathbf{A}_0, A_2, A_1, A_3, \mathbf{A}_0, A_2, A_4, A_1, A_0, A_3, A_4, A_2, A_1, A_0, A_3, A_2, A_1, A_4, A_3, A_4\}$$

$$D^2 = \{\mathbf{A}_0, A_2, A_1, \mathbf{A}_0, A_3, A_2, A_4, A_1, A_0, A_3, A_4, A_2, A_1, A_0, A_3, A_2, A_1, A_4, A_3, A_4\}$$

A_0 aktivitesi daha fazla öne alınamaz çünkü aksi takdirde olurluluk bozulacaktır. İlk pozisyondaki A_0 aktivitesi kullanılarak bu iki komşu elde edildikten sonra D döngüsünde sıralamadaki ikinci aktivite olan A_2 aktivitesi ele alınır. Bu aktivitenin ikinci tekrarı 5'inci pozisyondadır. Olurluluk şartlarına göre iki A_2 aktivitesi arasında birer adet A_1 ve A_3 yer almalıdır. Bu örnek için A_3 4'üncü pozisyonda olduğu için yeni bir komşu elde edilemez. Bir sonraki aktivite A_1 ve ikinci tekrarı 8'inci pozisyondadır. Olurluluk şartlarını ihlal etmeyecek şekilde bir önceki pozisyona kaydırılır. Bu durumda aşağıdaki D^3 döngüsü komşu olarak elde edilecektir:

$$D^3 = \{A_0, A_2, A_1, A_3, A_2, A_0, \mathbf{A}_1, A_4, A_0, A_3, A_4, A_2, A_1, A_0, A_3, A_2, A_1, A_4, A_3, A_4\}$$

Döngü Zamanının Hesaplanması

Döngü zamanını hesaplamak için iteratif bir yöntem geliştirilmiştir. Döngü zamanının hesaplanma yönteminin daha net anlaşılması için bir örnek verilmiştir.

Örnek 3: 2 makineli 3 farklı parça üretimi yapan bir hücre olduğu varsayalım. $\varepsilon = 1$, $\delta = 2$ ve işlem zamanları Çizelge 3.1’de verilmiştir. Robot hareket sıralaması ve başlangıç zamanları da aşağıda verilmiştir.

Çizelge 3.1: İşlem zamanları.

Parça Tipi	M1	M2
1	10	8
2	15	6
3	20	3

σ_{RMS}	A_0	A_2	A_1	A_2	A_0	A_1	A_0	A_2	A_1
T_h	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9

Bu yöntemde ilk iterasyonda bütün aktivite başlangıç zamanları 0 olarak ayarlanır. Bütün döngülerin A_0 aktivitesi ile başladığı varsayıldığından bu aktiviteden başlanarak robot belirlenmiş olan hareket sıralamasını takip eder. Bu esnada da bütün aktivite başlangıç zamanları aşağıda açıklanan şekilde ve denklem 3.1 veya 3.2’ye göre güncellenir. Bir döngüdeki bütün aktiviteler tamamlandığında bir önceki başlangıç zamanlarıyla yeni başlangıç zamanları birbirleriyle karşılaştırılır. Eğer ardışık iki döngüdeki zamanlar birbirleriyle aynı ise algoritma durdurulur ve çevrim zamanı hesaplanmış olur. Değilse bu eşitlik sağlanana kadar iterasyonlar devam ettirilir. Yapılan testlerde en fazla 2-3 iterasyon sonunda algoritmanın çevrim zamanını doğru bir şekilde hesaplayarak durduğu gözlemlenmiştir.

T_1 değeri daima 0’a eşittir. Her bir T_h iki faktöre bağlı olarak hesaplanmaktadır. İlk olarak T_h , T_{h-1} ’e bağlıdır. Bir önceki pozisyona atanan aktivite tamamlanmadan başlayamaz. Diğer bir yandan, A_j ’ye ait başlangıç zamanını belirlemek için, son A_{j-1} ’in bilinmesi gereklidir (bu \hat{A}_{j-1} olsun). Örnek verilen döngüde, A_2 , A_0 ’dan sonraki pozisyonda ve bu sıralamada girdi stoğundan parça alan robot birinci makineye yükleme yaptıktan sonra A_2 ’yi gerçekleştirebilmek için ikinci makineye hareket edecektir. Bir önceki döngüde yapılan son aktivite A_1 olması döngünün ikinci makine dolu olarak

sonlandığı anlamına gelmektedir ve döngü tekrar başladığında ikinci makineye yüklenen parçanın işlemi henüz bitmedi ise A_2 aktivitesini gerçekleştirebilmek için robotun işlem tamamlanana kadar beklemesi gerekmektedir. Bu durum, döngüde yer alan son A_1 'in başlangıç zamanına \hat{A}_1 bakılarak belirlenebilir.

k parçasına ait A_j^k aktivitesinin herhangi bir v pozisyonuna atanmış olduğunu varsayalım. Bu aktivitenin başlangıç zamanı aynı k parçasına ait $A_{(j-1)}^k$ aktivitesinin ve $v-1$ pozisyonuna atanmış olan aktivitenin tamamlanma zamanından büyük veya eşit olmalıdır. $A_{(j-1)}^k$ aktivitesinin s pozisyonuna atandığını varsayalım. Pozisyon v ile s arasındaki ilişkiye bağlı olarak 2 durum ortaya çıkacaktır. Bu da aşağıdaki eşitliklerle ifade edilebilir.

Eğer $v > s$ ise (Yani A_j^k aktivitesi pozisyon sıralamasında $A_{(j-1)}^k$ aktivitesinden daha sonra geliyorsa):

$$T_v = \max\{T_{v-1} + 2\varepsilon + \delta + |v - (s + 1)|\delta, T_s + 2\varepsilon + \delta + P_{kj}\} \quad (3.1)$$

Eğer $v < s$ ise:

$$T_v = \max\{T_{v-1} + 2\varepsilon + \delta + |v - (s + 1)|\delta, T_s + 2\varepsilon + \delta + P_{kj} - CT\} \quad (3.2)$$

İkinci eşitlikte başlangıçta $CT = 0$ olarak alınır. Fakat sonraki iterasyonlarda $CT = \max\{T_j + 2\varepsilon + \delta + |j + 1|\delta\}$ formülü ile hesaplanmaktadır. CT bütün aktivitelerin tamamlanma zamanlarına girdi stok alanına dönüş zamanı eklenerek elde edilen değerlerden en büyük olanına eşit olmalıdır.

Bu örnek için robot hareket sıralaması, parça sıralaması ve başlangıç zamanlarının her bir pozisyon için iteratif olarak nasıl hesaplandığı aşağıda gösterilmektedir:

$$\sigma_{RMS} = \{A_0, A_2, A_1, A_2, A_0, A_1, A_0, A_2, A_1\} \text{ ve } \sigma_{ps} = \{1, 2, 3\}$$

$$T_1 = 0$$

$$T_2 = \max\{T_1 + 2\varepsilon + \delta + |2 - (0 + 1)|\delta, T_0 + 2\varepsilon + \delta + P_{32} - CT\} = 6$$

$$T_3 = \max\{T_2 + 2\varepsilon + \delta + |1 - (2 + 1)|\delta, T_0 + 2\varepsilon + \delta + P_{11}\} = 14$$

$$T_4 = \max\{T_3 + 2\varepsilon + \delta + |2 - (1 + 1)|\delta, T_2 + 2\varepsilon + \delta + P_{12}\} = 26$$

$$T_5 = T_4 + 2\varepsilon + \delta + |0 - (2 + 1)|\delta = 36$$

$$T_6 = \max\{T_5 + 2\varepsilon + \delta + |1 - (0 + 1)|\delta, T_5 + 2\varepsilon + \delta + P_{21}\} = 55$$

$$T_7 = T_6 + 2\varepsilon + \delta + |0 - (2 + 1)|\delta = 63$$

$$T_8 = \max\{T_7 + 2\varepsilon + \delta + |2 - (0 + 1)|\delta, T_6 + 2\varepsilon + \delta + P_{22}\} = 69$$

$$T_9 = \max\{T_8 + 2\varepsilon + \delta + |1 - (2 + 1)|\delta, T_7 + 2\varepsilon + \delta + P_{31}\} = 87$$

Bu algoritmanın durma koşulu, ardışık iki iterasyondaki T_h değerlerinin birbiri ile aynı değeri almasıdır. Bu durumda yukarıda verilen hesaplamalar bu koşul sağlanana dek tekrarlanmaktadır. Yapılan hesaplamalarda formüller içerisinde T_h değerleri yerine önceki iterasyonda bulunan değerler kullanılmaktadır.

Son olarak döngü zamanı son pozisyon z (bu örnekte A_1) ve döngü zamanı aşağıdaki formül ile hesaplanır:

$$CT = T_{n(m+1)} + 2\varepsilon + \delta + (z + 1)\delta = 95 \rightarrow CT = T_9 + 2 + 2 + (1 + 1)2 = 95$$

T_2 için yazılan eşitliğe bakıldığında \max fonksiyonunun ikinci kısmında T_9 'u görülmektedir. Daha önce de bahsedildiği üzere bir önceki döngüde yapılan son aktivite A_1 ve A_2 aktivitesinin başlayabilmesi ikinci makinede parçanın işleminin bitmesi gerekir. Parçanın ikinci makinede işleminin başlaması ise A_1 'in başlangıç zamanına yani T_9 'a bağlıdır.

Tabu Hareketi Tanımı: Tabu Arama algoritmasında belirlenmesi gereken tanımlardan birisi de tabu olarak kabul edilecek harekettir. Bu algorithmada tabu hareketi, iki komşunun pozisyonlarına atanan aktivitelerin karşılaştırılması ile belirlenir. Bu karşılaştırma esnasında ilk karşılaşılan aynı pozisyonda farklı aktivite atanması durumunda pozisyon bilgisi tabu olarak kaydedilir. Komşuluk tanımında pozisyonlardaki aktiviteler değiştirilerek yeni komşular üretilen bu algorithmaya uygun olarak devamlı aynı komşular üzerinde durulmasını engelleyecek bir tabu hareketi tanımlanması amaçlanmıştır.

Tabu Arama algoritmasında kullanılan parametreler:

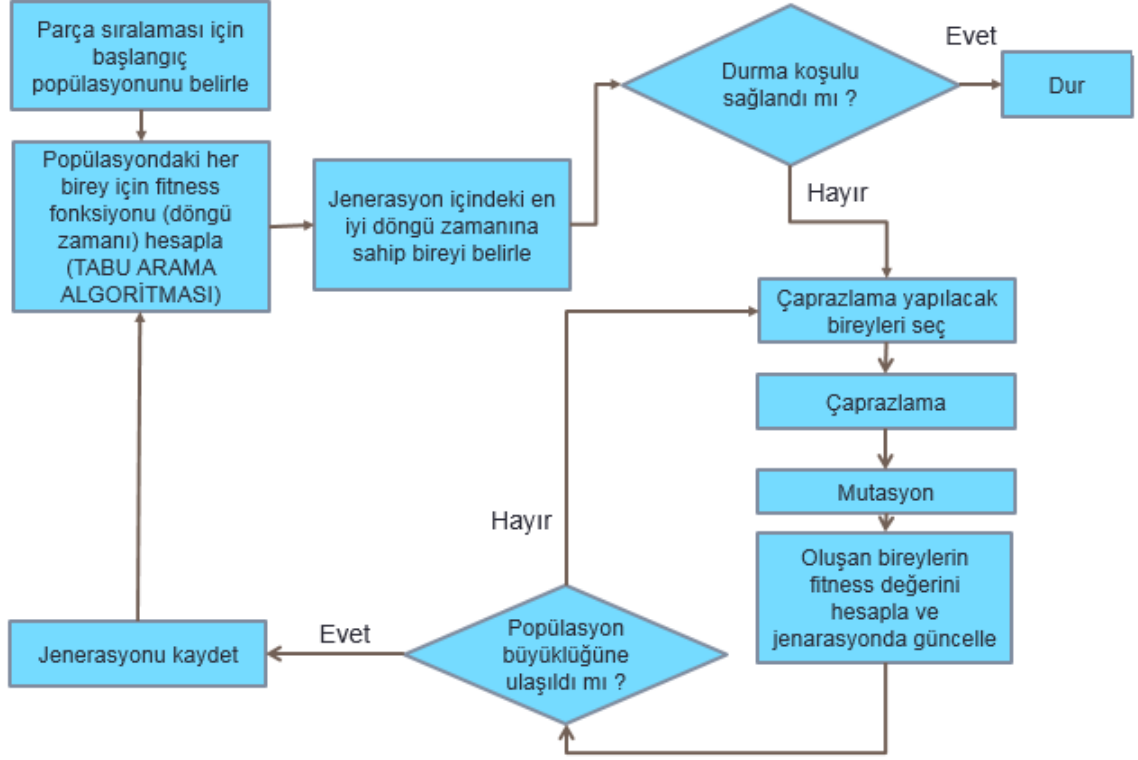
- **Tabu Listesi (TL) Uzunluk Katsayısı:** Tabu Arama algoritması robot hareket sıralamasını belirlemek için geliştirilmiştir. Problemden yer alan parça sayısı n ve

makine sayısı m değerine bağlı olarak bir döngü toplamda $n(m + 1)$ adet aktivite içermektedir. Tabu aramada bu aktivitelerin bulunduğu pozisyonlar yasaklanmaktadır. Dolayısıyla tabu listesi pozisyonları kaydetmektedir. Sabit bir tabu listesi uzunluğu belirlenmesi durumunda, küçük boyutlu problemler için çok fazla hareket yasaklanacak ve algoritma çok kısa sürede iyi sonuçlar bulamadan sonuçlanabilecektir. Diğer taraftan büyük boyutlu problemlerde de çok az sayıda aktivite yasaklanacağı için algoritma döngüye girerek olurlu bölgenin tamamını araştırmadan sonuçlanabilecektir. Bu problemin üstesinden gelebilmek için tabu listesinin uzunluğu problemin boyutuna göre daha uzun veya daha kısa seçilebilir bir şekilde "tabu listesi uzunluk katsayısı kullanılarak" ayarlanmıştır. Örnek olarak, söz konusu parametre 0.2 seçildi ise ve problem 4 parça, 4 makineden oluşuyor ise tabu listesi uzunluğu $0.2 \times ((4 + 1)4)$ 'den 4'tür. İşlemin sonucunun tamsayı olmaması durumunda en yakın yukarıdaki tamsayıya yuvarlanmaktadır.

- **Durma Koşulu:** Algoritma durma koşulu olarak, iyileşmeyen çözüm sayısını baz almaktadır. İyileşmeyen çözüm sayısı algoritma içerisinde bir sayaç ile hafızada tutulmaktadır. Her iyileşmeyen çözümde bu sayaç değeri 1 artırılır ve bu sayaç başlangıçta belirlenen değerden küçük olduğu sürece algoritma çalışmaya devam eder, aksi takdirde durdurulur. Çözümde herhangi bir iyileşme olduğunda bu sayaç sıfırlanır.

3.1.2 Genetik algoritma

Genetik algoritma ile parçaların üretim sıralaması belirlenecektir. Dolayısıyla en basit ve kullanışlı kodlama parça sıralamasının gösterildiği kodlamadır. Örneğin, $n = 6$ olan bir örnek için $\{3, 4, 1, 2, 6, 5\}$ bir çözümü belirtmektedir. Bu çözümde 3 numaralı parça ilk sırada, 5 numaralı parça ise son sırada üretilecektir. Şekil 3.2'de akış şeması verilmiştir.



Şekil 3.2: Genetik algoritma akış şeması.

Başlangıç popülasyonunun belirlenmesi :

Belirlenen popülasyon büyüklüğü parametresi adedince olurlu çözüm rasgele bir şekilde belirlenmektedir. n parçanın olduğu bir problem için toplam $(n - 1)!$ kadar olurlu çözüm vardır. Parçaların sıralamadaki permütasyonlarının her biri bir çözümü ifade etmektedir. Döngü olduğu için de $(n - 1)!$ adet sıralama elde edilmektedir.

Fitness (Uygunluk) değerinin hesaplanması:

Popülasyonda bulunan her birey için fitness değerinin hesaplanması gerekmektedir. Döngü zamanı algoritimada fitness (uygunluk) değerini ifade etmektedir. Bu aşamada her bir birey için Tabu Arama Algoritması çağrılır ve bu bireyin ifade ettiği parça sıralaması için robot hareket sıralaması belirlenir. Tabu Arama Algoritması'nın anlatıldığı Bölüm 3.1.1'de detayları anlatılan iteratif yöntem ile döngü zamanı (fitness) hesaplanmaktadır.

Çaprazlama yapılacak bireylerin seçilmesi :

Çaprazlama yapılacak bireyler rulet tekerleği seçme kriterine göre seçilmektedir. Bu kriterde her bireyin seçilme olasılığı o bireyin uygunluk değeriyle orantılı olarak ağır-

lıklandırılır. Örneğin j bireyine ait uygunluk değeri f_j ile gösterilirse bu bireyin çaprazlama için seçilme olasılığı aşağıdaki şekilde hesaplanır:

$$p_j = \frac{f_j}{\sum_{j \in J} f_j}$$

Her bireyin seçilme olasılığı belirlendikten sonra bir sonraki jenerasyondaki popülasyon sayısına ulaşılan kadar bireyler seçilir ve çaprazlama yapılır.

Çaprazlama:

Kullanılan çaprazlama operatöründe seçilen bireyler ortadan bölünerek çaprazlanmaktadır. İlk çocuk için ilk ebeveyne ait ilk kısım alınarak kalan kısım ikinci ebeveyndeki sıralamaya göre tamamlanır. İkinci çocukta ise ikinci ebeveynin ilk kısmı alınıp kalan kısım ilk ebeveyndeki sıralamaya göre tamamlanır. Örneğin, seçilen ilk birey $\{2, 1, 3, 5, 4, 6\}$, ikinci birey, $\{3, 4, 1, 2, 6, 5\}$ olsun. İlk bireyin elemanlarının yarısı alınır $\{2, 1, 3\}$ ve daha sonra ilk bireyin ilk yarısında olmayan elemanlar için ikinci bireyin eleman sırasına göre alınır ve çaprazlama sonucunda üretilen yeni birey $\{2, 1, 3, 4, 6, 5\}$ olur. Bu örnek için ikinci birey ise $\{3, 4, 1, 2, 5, 6\}$ şeklinde belirlenmiş olur.

Mutasyon :

Mutasyon; belirlenmiş mutasyona uğrama olasılığına uygun olarak yapılır. Eğer bir bireye mutasyon yapılacaksa rasgele belirlenen iki pozisyondaki elemanların yerleri değiştirilir. Örneğin, mutasyona uğrayacak olan birey: $\{2, 1, 3, 5, 4, 6\}$ olsun. Rasgele 2 ve 4 numaralı pozisyonlar seçilmiş olsun. Bu durumda bu iki pozisyondaki hareketlerin yerleri değiştirilir ve $\{2, 5, 3, 1, 4, 6\}$ bireyi elde edilir.

Geliştirilen Genetik Algoritmada kullanılan parametrelerin detayları şu şekildedir:

- **Popülasyon Büyüklüğü:** Popülasyonda bulunabilecek birey sayısını belirler. Popülasyon büyüklüğü β parametresi ile parça sayısı n 'e bağlı olarak hesaplanmıştır. $(\beta \times n)$
- **Durma Koşulu:** Genetik algoritmada kullanılan durma koşulu ardışık iyileşmeyen jenerasyon sayısı ile belirlenmiştir. Üretilen jenerasyondaki en iyi fitness değerine sahip olan birey mevcut en iyi çözümden daha iyi değil ise bu bir sayaç

yardımı ile tutulmaktadır. Çözümde iyileşme olduğu zaman sayaç sıfırlanır. Parametre olarak verilen ardışık iyileşme olmayan çözüm sayısına ulaşıldığında algoritma durur.

- **Elit çözüm sayısı:** Diğer jenerasyon oluşturulurken çaprazlama veya mutasyona uğramadan doğrudan diğer jenerasyona geçecek birey sayısını belirleyen katsayıdır. Örnek olarak, popülasyon büyüklüğü 30, elit çözüm sayısını belirleyen parametre 0.1 ise en iyi uygunluk değerine sahip $0.1 \times 30 = 3$ birey doğrudan bir sonraki jenerasyona aktarılır. Bu bireyler aynı zamanda çaprazlamada da kullanılabilir.
- **Mutasyon olasılığı:** Genetik algoritmanın bir parçası olan mutasyon verilen bu olasılığa göre gerçekleştirilmiştir.

3.2 1-birim Döngüler için Geliştirilen Algoritma

Çizelge 1.1’de de görüleceği üzere literatürde değişik tip parça üretimi söz konusu olduğunda CRM (Concatanated robot move cycles) döngüleri olarak binilen 1-birim döngülerin tekrarından meydana gelen döngüler sıklıkla ele alınmıştır. Bu çalışmada da döngü tipinin çözüme etkisini görebilmek amacı ile n -birim döngüleri ele alan algoritmaya ek olarak CRM döngüleri ele alan bir algoritma daha geliştirilmiştir.

Geliştirilen algoritmanın çalışma mekanizması n -birim döngüleri ele alan hibrit algoritma ile aynıdır. Fakat çözüm 1-birim döngülerden meydana geleceği için robot hareket sıralamasını belirleyen Tabu arama algoritmasında başlangıç çözümü bulunması ve komşuların türetilme yönteminde değişiklikler yapılmıştır. Başlangıç çözümü için sadece 1-birim döngüler ele alınıp aynı olurluluk şartları baz alınarak oluşturulur. Komşuluk tanımı ise sadece 1-birim döngü üzerinde diğer algoritmanın komşuluk tanımı ile komşular türetilip ve üretilen 1-birim döngülerin n defa tekrarlanıp yeni komşular elde edilir. 3 makineli 4 farklı parçanın üretildiği bir sistem için üretilen 1-birim döngülerin oluşturduğu robot hareket sıralamasına aşağıda örnek verilmiştir:

$$\sigma_{RHS} = \{A_0, A_2, A_3, A_1, A_0, A_2, A_3, A_1, A_0, A_2, A_3, A_1, A_0, A_2, A_3, A_1\}$$

Görüldüğü üzere $\{A_0, A_2, A_3, A_1\}$ döngüsünün $n = 4$ defa tekrar ettiği görülmektedir.



4. GELİŞTİRİLEN ÇÖZÜM YÖNTEMLERİNİN KARŞILAŞTIRILMASI

Geliştirilen algoritmaların performansı analiz edilirken iki önemli kriter, elde edilen çözümün en iyi çözüme yakınlığı ve çözüm zamanıdır. Yazılım kodları tasarlanırken aynı işlemi daha kısa sürede yapan yöntemler tercih edilir. Daha iyi çözümler elde etmek daha fazla zaman gerektirdiği için olabildiğince etkili bir kodlama yöntemi kullanılmaktadır. Bu amaçla kullanılan yöntemlerden biri ise paralel kodlama yöntemidir. Algoritmada yapılması gerekenleri bilgisayarda bulunan çekirdekleri paralel olarak kullanarak yerine getirmek çözüm zamanını azaltmaktadır. Bu programlama yöntemine çoklu çekirdek programlama (multi core programming) denmektedir.

İç içe çalışan hibrit sezgisel algoritmada, bahsedildiği üzere genetik algoritmanın popülasyonunda bulunan her birey için tabu arama algoritmasının çalışması gerekmektedir. Her bir jenerasyonda popülasyon büyüklüğü kadar tekrar edilmesi gereken bu işlem çözüm zamanında büyük etkiye sahiptir. Bu yüzden bireyler için tabu arama algoritmasının çekirdeklerde paralel olarak çalıştırılmasına karar verilmiştir.

Uygulanan paralel kodlama yönteminin algoritma üzerinde etkisini görebilmek amacı ile bazı problemler üzerinde testler yapılmıştır. Çizelge 4.1’de paralel kodlama yöntemi kullanılmadan (Normal) ve yöntem kullanıldıktan sonra (Paralel) aynı problemler için oluşan çözüm zamanları verilmiştir.

Yapılan testlerde 16 çekirdeğe sahip bir bilgisayar kullanılmıştır. Paralel programlama yöntemi ile çalışan algoritma ortalamada 10 kattan daha hızlı çözüm verdiği görüldükten sonra algoritmanın paralel çalışan hali ile devam etmeye karar verilmiştir. Paralleleştirilen algoritmanın performansını test etmek amacı ile n -birim döngüler için geliştirilen algoritma kullanılmıştır. Etkisini inceledikten sonra her iki algoritmaya da paralel programlama yöntemi uygulanmıştır.

Çizelge 4.1: Paralel ve normal kodlanan algoritmanın karşılaştırılması.

Parça(n)	Makine (m)	Çözüm Zamanı	
		Normal	Paralel
4	4	4,25	0,44
4	4	4,64	0,45
4	5	6,27	0,84
4	5	7,71	0,72
5	3	12,98	1,67
5	3	15,68	1,28
5	4	23,38	2,39
5	4	20,20	1,55
5	5	7,19	0,56
5	5	7,84	0,53
6	3	14,40	1,53
6	3	14,16	1,50
6	4	22,73	2,09
6	4	25,99	2,34
6	5	60,87	4,36
6	5	57,97	3,80

4.1 Parametre Kalibrasyonu

Performans analizine geçmeden önce Tabu Arama ve Genetik algoritmada kullanılan parametreler için en uygun değerlerin hangileri olduğuna karar verilmelidir. Bu değerleri başlangıçta bilmek mümkün değildir. Çözüm zamanı ve çözüm kalitesi açısından iyi sonuçlar elde etmek için parametre kalibrasyon çalışmaları yapılmıştır. İç içe çalışan Tabu Arama ve Genetik algoritmalarının parametre kalibrasyonu beraber yapılmıştır. Algoritmaların beraber ele alınmasının sebebi birbirlerini etkileyen parametrelerinin olmasıdır. Mesela populasyon büyüklüğü ile tabu listesinin uzunluğu problemin çözüm süresini birlikte etkileyen parametrelerdir. Çünkü popülasyondaki her birey için Tabu Arama algoritması çağrılacağı için burada da Tabu Arama Algoritmasının parametrelerinin etkisi devreye girecektir.

Her iki algoritma için kalibrasyon çalışması kapsamında makine çizelgeleme çevrimiçi kütüphanesinde bulunan bazı veri kümeleri kullanılmıştır [url-2]. Veri kümelerinde işlem zamanları yer almaktadır. Tüm problemler için $\varepsilon=1$ alınmış ve δ ise işlem zamanları dikkate alınarak üretilmiştir. Yapılan testlerde bahsedilen veri kümelerinden $m = 7$, $n = 7$ olan bir problem ve işlem zamanları $U(0,10000)$ arasında üretilen $m = 5$, $n = 10$ olan bir problem kullanılmıştır. Bu şekilde iki farklı problemin ele alınmasının nedeni

işlem zamanları arasındaki varyansın çözüm üzerindeki etkisini görmektir. Makine çizelgeleme çevrimiçi kütüphanesinden alınan problemin işlem zamanları arasındaki varyans düşük ve üretilen problemin aralığını yüksek tutmanın amacı ise varyansı büyük olan bir problem elde edilmesinin istenmesidir. Diğer yandan kullanılan delta değerlerinin de çözümü etkilediği bildiğimiz için iki farklı delta seviyesi üretilip bu iki problem için kullanılmıştır. Sonuç olarak 2 farklı problem için Çizelge 4.2 ve Çizelge 4.5'te verilen parametre seviyeleriyle çözüm alınmıştır. Başlangıç çözümü oluşturulması ve çarpazlanacak bireylerin seçilmesinde rasgelelik etkili olduğu için bu durumun etkisinden kurtulmak amacıyla her kombinasyondan 5 tekrar alınmıştır.

Çizelge 4.2'de verilen 6 farklı parametre ve δ değeri için iki farklı seviye denenmiştir. Bu durumda $2^7 \times 2 \times 5 = 1280$ defa n -birim döngüler için geliştirilen algoritma koşuturup bahsedilen parametre değerlerine karar verilmiştir.

Çizelge 4.2: n -birim algoritmasında kullanılan parametreler ve denenilen seviyeler.

Parametreler	Seviyeler	
	Küçük	Büyük
δ	Küçük	Büyük
β	10	15
GA için durma kriteri (γ_g)	50	100
TAA için durma kriteri (γ_t)	30	50
Mutasyon olasılığı (ρ)	0,1	0,05
Elit Çözüm Seçme Katsayısı (σ)	0,1	0,2
TL Uzunluk Katsayısı (λ)	0,05	0,1

Yapılan testler sonucunda parametre seviyelerinin her kombinasyonu için Çizelge 4.3 ve 4.4 oluşturulmuştur. Bu seviyeler ile çözülen iki farklı problem için kullanılan iki farklı δ değeri hesaba katıldığında 4 farklı problem ortaya çıkmaktadır. Her bir problem için tüm testler beşer defa tekrar edilmiştir. 1280 koşuturumun $4 \times 5 = 20$ koşuturumu aynı parametre seviyeleri için alınmıştır. Bu durumda 64 farklı seviye için bu alınan koşuturumların döngü zamanının tüm koşuturumlarda bulunan en iyi döngü zamanından % sapma değerleri hesaplanmıştır. Tüm koşuturumlardan elde edilen tüm döngü zamanlarının ortalama ve maksimum sapma değerleri hangi parametre değerinin seçileceğini belirlemektedir. Çizelge 4.3 ve 4.4'te ortalama sapma değerine göre küçükten büyüğe doğru sıralanmış, en küçük maksimum sapma değerini de bu parametre değerleri ile elde edildiği görülmüştür. Çözüm zamanının en büyük çaplı problemlerde dahi 2 dakikayı geçmediği görülmüş ve bu parametre değerleri ile devam etmeye karar verilmiştir.

Çizelge 4.3: n -birim algoritmasının parametre kalibrasyon koşuturlarının sonuçları-1.

No	β	γ_g	γ_i	ρ	σ	λ	Ortalama % Sapma	Max % Sapma	Çözüm Zamanı
1	15	100	50	0,1	0,1	0,05	1,38	2,52	67,66
2	15	100	50	0,1	0,1	0,1	1,44	2,65	67,60
3	10	100	50	0,1	0,1	0,05	1,46	2,52	69,68
4	10	100	50	0,1	0,1	0,1	1,48	2,64	63,22
5	15	100	30	0,05	0,1	0,05	1,52	3,10	51,81
6	10	100	50	0,05	0,1	0,05	1,53	2,55	71,29
7	15	100	30	0,05	0,1	0,1	1,56	2,99	50,22
8	10	100	30	0,05	0,1	0,05	1,56	2,97	50,19
9	15	100	50	0,1	0,2	0,05	1,56	3,58	48,64
10	15	100	50	0,05	0,1	0,05	1,57	3,31	68,42
11	10	100	50	0,1	0,2	0,1	1,58	3,24	66,15
12	10	100	50	0,1	0,2	0,05	1,59	3,58	60,46
13	15	50	30	0,05	0,1	0,05	1,61	3,12	21,14
14	15	50	50	0,1	0,1	0,1	1,62	3,33	34,16
15	10	100	30	0,05	0,1	0,1	1,63	2,99	54,84
16	10	100	30	0,05	0,2	0,05	1,64	3,83	51,53
17	15	100	30	0,05	0,2	0,05	1,64	3,83	35,46
18	10	50	50	0,1	0,1	0,05	1,64	4,16	35,24
19	10	100	30	0,1	0,2	0,1	1,65	4,32	46,64
20	10	50	50	0,05	0,2	0,1	1,65	3,33	31,73
21	10	50	50	0,05	0,1	0,1	1,66	3,32	32,18
22	10	100	50	0,05	0,2	0,1	1,66	5,07	61,97
23	10	100	30	0,1	0,1	0,05	1,68	3,18	43,88
24	15	50	50	0,05	0,1	0,1	1,68	3,32	33,73
25	15	50	50	0,1	0,2	0,05	1,68	3,64	26,32
26	15	100	50	0,05	0,2	0,1	1,69	5,07	52,28
27	10	100	50	0,05	0,2	0,05	1,69	3,38	58,14
28	10	100	50	0,05	0,1	0,1	1,69	3,18	77,07
29	15	100	50	0,05	0,1	0,1	1,69	3,18	68,98
30	15	100	50	0,05	0,2	0,05	1,70	3,38	43,26
31	15	50	50	0,1	0,1	0,05	1,72	4,16	32,83
32	10	50	30	0,05	0,1	0,05	1,73	3,12	22,96

Çizelge 4.4: n -birim algoritmasının parametre kalibrasyon koşuturlarının sonuçları-2.

No	β	γ_g	γ_t	ρ	σ	λ	Ortalama % Sapma	Max % Sapma	Çözüm Zamanı
33	15	100	50	0,1	0,2	0,1	1,73	4,03	46,12
34	10	50	50	0,1	0,2	0,05	1,74	3,64	32,01
35	10	50	50	0,05	0,1	0,05	1,75	2,92	31,06
36	10	100	30	0,1	0,1	0,1	1,75	3,02	38,08
37	10	100	30	0,1	0,2	0,05	1,75	4,19	52,90
38	10	50	50	0,1	0,2	0,1	1,75	3,31	37,78
39	15	50	50	0,1	0,2	0,1	1,75	3,31	23,65
40	15	100	30	0,1	0,1	0,1	1,75	3,50	44,62
41	15	100	30	0,1	0,2	0,1	1,76	4,32	36,71
42	10	50	50	0,1	0,1	0,1	1,76	3,21	32,92
43	15	100	30	0,1	0,2	0,05	1,77	4,19	37,93
44	15	50	50	0,05	0,2	0,1	1,79	3,85	23,50
45	15	50	50	0,05	0,1	0,05	1,79	2,68	26,43
46	15	100	30	0,05	0,2	0,1	1,81	3,52	31,38
47	10	50	30	0,1	0,2	0,05	1,81	4,50	23,17
48	10	50	50	0,05	0,2	0,05	1,82	3,07	28,02
49	10	100	30	0,05	0,2	0,1	1,82	3,52	44,51
50	10	50	30	0,1	0,1	0,05	1,82	3,41	31,34
51	15	50	30	0,1	0,1	0,05	1,84	3,41	22,25
52	15	100	30	0,1	0,1	0,05	1,84	3,81	52,90
53	15	50	50	0,05	0,2	0,05	1,88	3,24	21,79
54	15	50	30	0,1	0,2	0,05	1,92	4,50	15,29
55	15	50	30	0,05	0,2	0,05	1,96	3,69	15,96
56	15	50	30	0,1	0,1	0,1	1,97	3,58	20,42
57	10	50	30	0,1	0,1	0,1	1,98	3,41	33,80
58	10	50	30	0,05	0,1	0,1	1,99	4,55	24,39
59	10	50	30	0,1	0,2	0,1	1,99	6,15	18,76
60	15	50	30	0,05	0,1	0,1	2,00	4,55	19,56
61	15	50	30	0,1	0,2	0,1	2,01	6,15	17,37
62	10	50	30	0,05	0,2	0,1	2,02	3,86	19,24
63	10	50	30	0,05	0,2	0,05	2,04	3,69	17,54
64	15	50	30	0,05	0,2	0,1	2,08	4,32	17,82

Bir önceki durumla paralel olarak 1-birim döngüler için geliştirilen algoritma da Çizelge 4.5'te verilen 6 farklı parametre ve δ değeri için iki farklı seviye denenmiştir. Yine 1280 defa bahsedilen algoritma koşuturulup tercih edilecek parametre değerlerine karar verilmiştir. Burada bir önceki algoritmadan farklı alınan seviye tabu listesi (TL) uzunluk katsayısıdır. Bir önceki algoritmada $n(m+1)$ adet pozisyona atanan aktiviteler incelenip olası komşuluklar üretilirken burada $(m+1)$ adet pozisyon incelenip olası 1-birim döngüler üretilip n defa tekrarlanmaktadır. Tabu listesi uzunluk katsayısı bu problemde $(m+1)$ ile çarpıldığı için daha büyük seviyelerin kullanılması gerektiğinden 0,05 ve 0,1 yerine 0,1 ve 0,3 kullanılmıştır.

Çizelge 4.5: 1-birim algoritmasında kullanılan parametreler ve denenen seviyeler.

Parametreler	Seviyeler	
	Küçük	Büyük
δ	Küçük	Büyük
β	10	15
GA için durma kriteri (γ_g)	50	100
TAA için durma kriteri (γ_t)	30	50
Mutasyon olasılığı (ρ)	0,1	0,05
Elit Çözüm Seçme Katsayısı (σ)	0,1	0,2
TL Uzunluk Katsayısı (λ)	0,1	0,3

Çizelge 4.6 ve 4.7’de ortalama sapma değerine göre küçükten büyüğe doğru sıralanmış, en küçük maksimum sapma değerini de bu parametre değerleri ile edildiği görülmüştür. n -birim ve 1-birim algoritmalarının ortalama sapma değerlerine bakıldığında kısmen farklı parametreler ile daha iyi sonuçlar verdiği görülmektedir. Mesela β değeri her ikisi için de 15 iken γ_g parametresi n -birim algoritması için 100 ve 1-birim algoritması için 50 olduğu belirlenmiştir. Ortalama sapma değerleri için en iyi performansın sergilendiği bu değerlerin maksimum sapma oranı için de en iyi seviyeler olduğu görülmektedir. Bu değerlerin çözüm zamanı da iki dakikayı geçmediği için bu parametre değerleri ile devam etmeye karar verilmiştir.

Çizelge 4.6: 1-birim algoritmasının parametre kalibrasyon koşuturlarının sonuçları-1.

No	β	γ_g	γ_t	ρ	σ	λ	Ortalama % Sapma	Max % Sapma	Çözüm Zamanı
1	15	50	50	0,1	0,1	0,3	2,75	7,67	34,16
2	15	100	50	0,1	0,1	0,1	2,83	8,63	67,66
3	10	100	30	0,1	0,1	0,1	2,84	9,65	43,88
4	10	100	30	0,1	0,1	0,3	3,01	8,63	38,08
5	15	100	30	0,1	0,2	0,1	3,07	9,09	37,93
6	15	50	30	0,1	0,1	0,1	3,14	9,04	22,25
7	15	50	50	0,1	0,2	0,3	3,16	10,07	23,65
8	15	50	30	0,1	0,2	0,1	3,17	10,37	15,29
9	15	100	30	0,1	0,1	0,1	3,22	9,04	52,90
10	10	50	30	0,1	0,2	0,1	3,24	9,04	23,17
11	15	50	50	0,05	0,1	0,3	3,27	9,04	33,73
12	15	50	50	0,1	0,2	0,1	3,29	8,74	26,32
13	15	50	30	0,05	0,2	0,3	3,30	7,91	17,82
14	15	50	50	0,05	0,2	0,1	3,31	7,79	21,79
15	10	50	50	0,05	0,2	0,1	3,33	8,04	28,02
16	15	50	30	0,05	0,1	0,1	3,35	7,74	21,14
17	15	100	50	0,1	0,2	0,3	3,39	12,01	46,12
18	10	50	50	0,1	0,1	0,3	3,39	8,94	32,92
19	15	100	30	0,1	0,2	0,3	3,40	9,88	36,71
20	15	50	30	0,1	0,1	0,3	3,42	10,02	20,42
21	15	50	30	0,1	0,2	0,3	3,50	11,20	17,37
22	15	50	50	0,1	0,1	0,1	3,50	11,87	32,83
23	15	100	50	0,1	0,2	0,1	3,54	10,95	48,64
24	10	50	50	0,1	0,2	0,1	3,61	11,87	32,01
25	15	100	50	0,1	0,1	0,3	3,64	11,20	67,60
26	10	50	30	0,1	0,1	0,3	3,66	10,22	33,80
27	10	100	30	0,1	0,2	0,1	3,67	9,04	52,90
28	15	50	50	0,05	0,1	0,1	3,68	12,10	26,43
29	15	50	30	0,05	0,2	0,1	3,70	10,85	15,96
30	10	50	50	0,1	0,2	0,3	3,75	9,04	37,78
31	10	100	50	0,1	0,1	0,1	3,76	10,37	69,68
32	15	100	30	0,1	0,1	0,3	3,82	14,37	44,62

Çizelge 4.7: 1-birim algoritmasının parametre kalibrasyon koşuturlarının sonuçları-2.

No	β	γ_g	γ_i	ρ	σ	λ	Ortalama % Sapma	Max % Sapma	Çözüm Zamanı
33	10	50	30	0.05	0.2	0.1	3,82	10,85	17,54
34	10	50	50	0.05	0.1	0.3	3,83	10,07	32,18
35	10	50	30	0.05	0.2	0.3	3,83	8,91	19,24
36	15	50	30	0.05	0.1	0.3	3,86	8,91	19,56
37	10	50	50	0.1	0.1	0.1	3,86	10,22	35,24
38	15	50	50	0.05	0.2	0.3	3,88	9,65	23,50
39	10	100	50	0.1	0.2	0.1	3,95	11,09	60,46
40	15	100	30	0.05	0.2	0.3	3,97	10,47	31,38
41	15	100	50	0.05	0.1	0.1	3,98	10,27	68,42
42	10	100	50	0.1	0.1	0.3	3,99	9,87	63,22
43	10	50	30	0.05	0.1	0.1	4,00	10,37	22,96
44	10	50	30	0.1	0.2	0.3	4,01	11,09	18,76
45	15	100	30	0.05	0.2	0.1	4,01	10,77	35,46
46	10	100	50	0.1	0.2	0.3	4,09	12,01	66,15
47	15	100	50	0.05	0.2	0.2	4,14	11,93	52,28
48	10	50	30	0.1	0.1	0.1	4,14	8,94	31,34
49	10	50	30	0.05	0.1	0.3	4,15	11,20	24,39
50	10	50	50	0.05	0.2	0.3	4,28	13,39	31,73
51	10	100	30	0.05	0.2	0.3	4,28	12,07	44,51
52	15	100	30	0.05	0.1	0.3	4,30	11,73	50,22
53	10	100	30	0.1	0.2	0.3	4,37	14,37	46,64
54	10	100	50	0.05	0.1	0.1	4,42	12,74	71,29
55	10	100	30	0.05	0.2	0.1	4,43	14,89	51,53
56	15	100	50	0.05	0.1	0.3	4,50	11,33	68,98
57	15	100	30	0.05	0.1	0.1	4,50	14,89	51,81
58	10	50	50	0.05	0.1	0.1	4,52	12,10	31,06
59	10	100	30	0.05	0.1	0.1	4,59	11,98	50,19
60	10	100	50	0.05	0.2	0.1	4,68	14,03	58,14
61	10	100	50	0.05	0.1	0.3	4,77	10,63	77,07
62	15	100	50	0.05	0.2	0.1	4,86	14,03	43,26
63	10	100	50	0.05	0.2	0.3	4,93	11,93	61,97
64	10	100	30	0.05	0.1	0.3	5,04	12,85	54,84

4.2 Test Sonuçları

Matematiksel model ve geliştirilen sezgisel algoritmaların performanslarını karşılaştırmak amacı ile işlem zamanları $U(0,10000)$ arasında üretilen 12 farklı problem ve bahsedilen çevrimiçi kütüphaneden [url-2] alınan 8 farklı problem kullanılmıştır. Toplamda 20 farklı problem, bu problemlerin işlem zamanlarına bağlı olarak türetilen iki farklı δ değeri kullanılarak test edilmiştir. Bu problem verilerinin özellikleri Ek 1’de gösterilmiştir. Elde edilen çözümlerin bulunan en iyi çözümden veya matematiksel model kullanılarak herhangi bir çözüm elde edilememiş ise geliştirilen altsınır formülasyonu yardımı ile hesaplanan altsınır değerinden uzaklığı ile yüzde sapma değeri hesaplanmıştır.

GAMS ile gösterilen sütun 1 saat zaman limiti ile çalıştırılan matematiksel modelin sonuçlarını, n -birim ve 1-birim ile gösterilen sütunlar ise yine bu probleme özel geliştirilen algoritmalar ile elde edilen sonuçları göstermektedir. Çizelge 4.8 ve 4.9’da matematiksel model ve algoritmalarından elde edilen çözümlerin sapma oranları ve çözüm zamanları gösterilmiştir. Burada hesaplanan sapma değerleri, optimal çözümün bulunduğu Çizelge 4.8’de gösterilen iki problem hariç diğer problem verileri için matematiksel model verilen zaman limiti içerisinde bir çözüme ulaşmıştır. 8 problem verisi için optimal çözüme hem matematiksel model hem de sezgisel algoritma kullanarak ulaşılmıştır. Ortalama sapma değerlerine bakıldığında n -birim döngüler için geliştirilen algoritmanın matematiksel modelden daha iyi bir performans gösterdiği görülmektedir. 1-birim döngüler için geliştirilen algoritma ise sapma değerleri açısından her iki çözüm yönteminden de daha kötü performans göstermiştir.

Çizelge 4.8: Matematiksel model ve sezgisel algoritma sonuçları-1.

Problem	δ	Döngü Zamanı (% Sapma)			Çözüm Zamanı		
		GAMS	n -birim	1-birim	GAMS	n -birim	1-birim
1	S	0,00	0,00	1,88	2,22	0,28	0,33
	L	0,00	0,00	3,61	2,84	0,23	0,34
2	S	0,00	0,00	2,34	54,95	0,55	0,46
	L	0,00	0,00	1,10	92,66	0,53	0,54
3	S	0,00	0,00	9,31	2385,39	0,64	0,56
	L	0,00	0,00	5,02	2415,50	0,61	0,67
4	S	7,23	6,56	10,98	ZL	1,03	0,67
	L	4,86	4,79	12,79	ZL	0,97	0,62
5	S	0,00	0,00	0,00	90,41	0,44	0,61
	L	0,00	0,00	2,17	80,41	0,45	0,81
6	S	3,77	3,15	6,91	ZL	0,84	0,79
	L	0,57	0,57	5,07	ZL	0,72	0,95
7	S	5,09	4,28	9,94	ZL	1,67	0,94
	L	3,08	0,68	6,49	ZL	1,28	1,15
8	S	ÇY	2,51	6,79	ZL	2,39	1,16
	L	ÇY	1,31	8,63	ZL	1,55	0,91
9	S	2,85	2,26	5,85	ZL	0,56	0,92
	L	1,05	2,05	5,91	ZL	0,53	1,19
10	S	7,96	2,70	10,97	ZL	1,53	1,19
	L	2,94	1,66	9,15	ZL	1,50	1,47
Ortalama %		2,19	1,63				

Matematiksel modelin özellikle Çizelge 4.9’da gösterilen problem verileri için verilen zaman limiti içerisinde herhangi bir çözüme ulaşamadığı görülmektedir. Burada hesaplanan sapma değerleri, bulunan döngü zamanının geliştirilen altsınır formülasyonu ile hesaplanan altsınır değerinden sapma değeridir. Bunun yanında, matematiksel modelin çözüm verdiği problem verilerindeki sezgisel algoritmadan elde edilen sonuçlar karşılaştırıldığında çok daha kısa sürede bir kaç problem hariç matematiksel modelden daha iyi sonuçlar verdiği gözlemlenmektedir. Ayrıca ortalama sapma değerine bakıldığında yine geliştirilen sezgisel algoritmanın, verilen zaman limiti içerisinde kesin çözüm yöntemine göre daha iyi performans gösterdiği açıkça görülmektedir.

Çizelge 4.9: Matematiksel model ve sezgisel algoritma sonuçları-2.

Problem	δ	Döngü Zamanı (% Sapma)			Çözüm Zamanı		
		GAMS	n -birim	1-birim	GAMS	n -birim	1-birim
11	S	ÇY	2,05	9,24	ZL	2,09	1,46
	L	ÇY	2,86	8,18	ZL	2,34	1,65
12	S	ÇY	3,16	11,30	ZL	4,36	1,64
	L	ÇY	4,18	10,43	ZL	3,80	2,74
13	S	ÇY	6,85	14,36	ZL	30,61	6,47
	L	ÇY	5,04	9,83	ZL	26,61	6,63
14	S	ÇY	7,65	16,25	ZL	33,79	5,09
	L	ÇY	3,79	5,72	ZL	19,66	4,97
15	S	ÇY	2,14	7,94	ZL	33,41	8,09
	L	ÇY	2,70	7,13	ZL	27,55	7,94
16	S	ÇY	3,11	4,90	ZL	23,11	5,62
	L	ÇY	1,26	3,41	ZL	20,69	5,53
17	S	ÇY	5,74	7,50	ZL	33,31	10,41
	L	ÇY	2,02	3,61	ZL	30,10	10,50
18	S	ÇY	5,05	7,39	ZL	74,78	32,88
	L	ÇY	4,23	7,25	ZL	68,64	29,13
19	S	ÇY	5,75	7,87	ZL	20,81	10,53
	L	ÇY	2,36	7,62	ZL	18,21	10,41
20	S	ÇY	4,03	3,19	ZL	53,21	18,53
	L	ÇY	2,75	3,72	ZL	52,87	18,41
Ortalama %			3,84	7,84			

Matematiksel modelin gösterdiği performansı iyileştirmek amacıyla ek çalışmalar yapılmıştır. Sezgisel algoritmanın her bir problem için verdiği çözümü bir başlangıç çözümü olarak matematiksel modele vererek çözüm sürelerinin ve % sapma değerlerinin nasıl değiştiği incelenmiştir. Bu sonuçlar Çizelge 4.10 ve 4.11’de verilmiştir.

Görüleceği üzere başlangıç çözümü verilmesinin matematiksel modeli çözerek elde edilen sapma değerleri üzerinde etkisi olmamıştır. Ortalama sapma miktarlarına bakıldığında n -birim döngüler için geliştirilmiş algoritma ve matematiksel model birbirine çok yakın performans göstermektedir. Bununla birlikte matematiksel model 40 örneğin ancak 8 tanesini çözebilirken bunlardaki ortalama süre 640,55 saniye olmuştur. n -birim için geliştirilen algoritma aynı problemleri ortalama 16,58 saniyede çözüm bulmuştur. Bunun yanında, n -birim ve 1-birim döngüler için geliştirilen algoritmaların sapma değerleri karşılaştırıldığında n -birim algoritmasının daha iyi olduğu fakat çözüm zamanlarına bakıldığında ise 1-birim algoritması daha iyi performans gösterdiği görülmektedir. Sonuç olarak, n -birim döngüler için geliştirilen algoritmanın çözüm zamanının da makul olduğunu göz önüne alarak çözüm kalitesi açısından daha iyi olduğuna karar verilmiştir.

Çizelge 4.10: Başlangıç çözümü verilen matematiksel model ve sezgisel algoritma sonuçları-1.

Problem	δ	Döngü Zamanı (% Sapma)			Çözüm Zamanı		
		GAMS	n -birim	1-birim	GAMS	n -birim	1-birim
1	S	0,00	0,00	1,88	2,22	7,11	3,33
	L	0,00	0,00	3,61	2,84	6,74	3,43
2	S	0,00	0,00	2,34	54,95	14,81	4,58
	L	0,00	0,00	1,10	92,66	23,14	5,44
3	S	0,00	0,00	9,31	2385,39	22,69	5,62
	L	0,00	0,01	5,02	2415,50	24,29	6,69
4	S	7,02	7,02	10,98	ZL	31,62	6,66
	L	5,04	5,04	12,79	ZL	25,31	6,18
5	S	0,00	0,00	0,00	90,41	12,43	6,07
	L	0,00	1,24	2,17	80,41	21,40	8,06
6	S	3,25	3,25	6,91	ZL	22,68	7,87
	L	0,12	0,57	5,07	ZL	34,47	9,50
7	S	4,47	4,47	9,94	ZL	34,54	9,39
	L	1,70	1,70	6,49	ZL	36,68	11,51
8	S	2,95	2,95	6,79	ZL	57,19	11,60
	L	0,55	0,55	8,63	ZL	57,83	9,12
9	S	2,31	2,31	5,85	ZL	27,59	9,20
	L	1,07	1,07	5,91	ZL	27,92	11,94
10	S	2,77	2,77	10,97	ZL	47,13	11,86
	L	2,76	2,76	9,15	ZL	39,10	14,69

Çizelge 4.11: Başlangıç çözümü verilen matematiksel model ve sezgisel algoritma sonuçları-2.

Problem	δ	Döngü Zamanı (% Sapma)			Çözüm Zamanı		
		GAMS	n -birim	1-birim	GAMS	n -birim	1-birim
11	S	2,09	2,09	9,24	ZL	53,11	14,56
	L	1,75	1,75	8,18	ZL	67,16	16,54
12	S	3,26	3,26	11,30	ZL	103,31	16,39
	L	5,16	5,16	10,43	ZL	106,00	27,42
13	S	7,36	7,36	14,36	ZL	370,68	27,00
	L	5,31	5,31	9,83	ZL	302,86	20,03
14	S	8,28	8,28	16,25	ZL	263,93	20,53
	L	3,94	3,94	5,72	ZL	291,75	28,83
15	S	2,19	2,19	7,94	ZL	450,44	29,68
	L	2,78	2,78	7,13	ZL	297,81	21,38
16	S	3,21	3,21	4,90	ZL	468,54	21,49
	L	1,28	1,28	3,41	ZL	433,18	35,93
17	S	6,10	6,10	7,50	ZL	359,09	36,56
	L	2,06	2,06	3,61	ZL	643,38	68,25
18	S	5,31	5,31	7,39	ZL	951,60	67,91
	L	4,42	4,42	7,25	ZL	584,00	35,90
19	S	6,10	6,10	7,87	ZL	307,72	35,32
	L	2,42	2,42	7,62	ZL	336,93	53,09
20	S	3,19	4,20	3,19	ZL	721,02	52,42
	L	2,82	2,82	3,72	ZL	813,09	89,23
Ortalama		2,88	2,89	7,04			

5. SONUÇ VE ÖNERİLER

Bu çalışmada, çeşitli sayıda makine ve bir adet tek tutuculu robotun bulunduğu bir üretim hattı ele alınmıştır. Ele alınan problem, değişik tip parça üreten robotik hücrelerin tasarımı problemidir. Sistemde parçalar makinelerde işlenirken, makineler arası parça transferi ve makinelerin yüklenmesi ve boşaltılması ise sistemdeki tek bir robot tarafından yapılmaktadır. Üretilecek her tip parçanın sistemdeki makinelerde işlenmek üzere belirli sayıda işlemi ile parça ve makineye özgü işlem zamanları bulunmaktadır. Bu tür sistemlerde hem robot belirli bir bilgisayar kodunu takip ederek aktivitelerini yaptığı için hem de uygulaması ve denetlemesi kolay olduğu için dögüsel üretim tercih edilmektedir. Yapılan çalışmada incelenen, n adet parça üretilmesi ile oluşan n -birim dögüsü adı verilen dögü çeşididir.

Değişik tip parça üretimi ve n -birim dögüleri ele alan çalışmaların yanı sıra değişik tip parça ve 1-birim dögüleri ele alan veya tek tip parça ve n -birim dögüleri ele alan çalışmalar da incelenmiştir. Literatür araştırmasında da üstünde durulduğu üzere, m makineli bir robotik hücrede n adet değişik tip parçanın üretildiği ve n -birim dögülerini inceleyen bir çalışmanın olmadığı görülmüştür. Bunun sonunda problem tanımı detaylı bir şekilde yapıp, çözüm yöntemi geliştirme aşamasına geçilmiştir.

Ele alınan problem için bir kesin çözüm yöntemi olan matematiksel model ve sonrasında performansını arttırmak amacı ile altsınır formülasyonu geliştirilmiştir. Problem boyutu büyüdükçe beklenildiği üzere çözüm alınamayan bu yöntem ek olarak bir sezgisel algoritma tasarlanmıştır. Literatürde sıklıkla kullanılan iki metasezgisel yöntem hibrit bir şekilde tasarlanmıştır. Tabu Arama algoritması verilen bir parça sıralaması için robot hareket sıralaması bulmaktadır. Genetik algoritma ise parça sıralamasını Tabu Arama algoritması ile belirlenen robot hareket sıralamasını kullanarak hesaplanan uygunluk değerine (dögü zamanına) göre belirlemektedir. Bu şekilde iç içe çalışan bir mekanizmaya sahip olan bir sezgisel algoritmaya ek olarak 1-birim dögüleri ele alan aynı mekanizmaya sahip bir sezgisel algoritma daha geliştirildi. Burada amaç, n -birim dögüler ile 1-birim dögülerin performansını değişik tip parçalar üzerinde test etmek-

döngüler ile 1-birim döngülerin performansını değişik tip parçalar üzerinde test etmektedir.

Ardından literatürden alınan ve rasgele üretilmiş işlem zamanlarını içeren toplamda 20 çeşit veri dosyası ile geliştirilen matematiksel model GAMS 12.2.6 çözücüsünde çözdürülmüştür. C++ programlama dili ile kodlanan her iki sezgisel algoritmanın çözüm zamanını iyileştirmek amacı ile paralel programlama yöntemine başvurularak güncelenen bilgisayar kodu öncekine göre çok daha kısa zamanda çözüm vermiştir. Buna ek olarak, algoritmalar çeşitli parametre değerleri içerdiği için kalibrasyon çalışmaları yapılmış, her bir algoritma için 1280 adet koşurum alınmıştır. Parametre değerlerine karar verildikten sonra, her iki algoritma da matematiksel modelde kullanılan veri dosyaları için çalıştırılmıştır. 3 yöntem hem döngü zamanı hem de çözüm zamanı açısından karşılaştırılmıştır.

Sonuç olarak, n -birim döngüleri için geliştirilen sezgisel algoritmanın m makineli ve n adet parça üreten bir sistemdeki problem için hem çözüm kalitesi hem de çözüm süresi bakımından oldukça iyi sonuçlar verdiği görülmüştür. Problem boyutu büyüdükçe, matematiksel modelin çözüm süresi artarken, bahsedilen algoritmanın çözüm süresinde ciddi bir artış gözlemlenmemekte ve en fazla yaklaşık bir dakikada çözüm vermektedir. 1-birim döngüleri için geliştirilen sezgisel algoritma ise sapma oranları açısından iyi performans göstermemekte fakat büyük çaplı problemler için çözüm zamanı açısından her iki yöntemden daha kısa zamanda çözüm verebilmektedir. Optimal çözüme ulaşılan problemlerde, n -birim döngüleri için geliştirilen sezgisel algoritma da optimal çözümleri bulmuştur. Çözüm zamanı açısından ise karşılaştırılmayacak ölçüde daha kısa zamanda cevap verebilmiştir.

Bu çalışmanın sonunda, n -birim döngülerin bu problem tipi için 1-birim döngülere göre daha iyi çözümler olduğu ortaya konulmuştur. Bu şekilde bir yaklaşımla, literatürde daha önce hiç çalışma yapılmamış olan bu alandaki boşluk doldurulmuştur.

Gelecekte yapılabilecek çalışmalara ilk olarak önerilebilecek olan problem tipi, sabit olarak alınan n sayısını çeşitlendirerek problemin gidişatını incelemektir. n -birim döngüler ele alınırken n sayısının çözüme etkisi olduğu bilinmemekte fakat bu tezde bu konu üstünde durulmamıştır.

Hat dengeleme bir çok üretim sektöründe ele alınan bir problem olup, bu problem çeşidi için de uygulanabilir. Makinaya özgü bir işlem zamanı tanımlamadan birbirine benzer nitelikte makinaların bulunduğu esnek bir üretim hattı için bu problem çözülebilir. Aynı şekilde ek olarak, bu parçaların birbirleri arasında öncelik ilişkileri olabilir.

Son olarak, tek tutuculu bir robotun ele alındığı robotik hücrede çift tutuculu bir robot kullanıldığında nasıl sonuçların ortaya çıktığı incelenebilir. Bu iki tip robot performans karşılaştırması yapılarak, maliyete katlanarak çift tutuculu robotun alınıp alınmamasına karar verilebilir.



KAYNAKLAR

- [1] **Hall, N. G., Kamoun, H., ve Sriskandarajah, C.,** (1997). Heat transfer Classification, two and three machine cells. *Operations Research* 45, 3 (1997), 421-439.
- [2] **Sethi, S. P., Sriskandarajah, C., Sorger, G., Blazewicz, J.,ve Kubiak, W.** (1992). Sequencing of parts and robot moves in a robotic cell. *International Journal of Flexible Manufacturing Systems* 4, 3-4, 331–358.
- [3] **Crama, Y., Kats, V., Van De Klundert, J., ve Levner, E.** (2000). Cyclic scheduling in robotic flowshops. *Annals of operations Research* 96, 1-4, 97-124.
- [4] **Brauner, N., F. G.,** (2001). On cycles and permutations in robotic cells. *Mathematical and Computer Modelling* 34, 565-591.
- [5] **Brauner, N., F. G.,** (1999). On the conjecture in robotic cells:new simplified proof for three-machine cases. *INFOR* 37, 20-36.
- [6] **Dawande, M., Geismar, H. N., Sethi, S. P., ve Sriskandarajah, C.,** (2005). Sequencing and scheduling in robotic cells: Recent developments. *Journal of Scheduling* 8, 5 (2005), 387–426.
- [7] **Bilge, U., ve Ulusoy, G.,** (1995). A time window approach to simultaneous scheduling of machines and material handling system in an fms. *Operations Research* 43, 6, 1058-1070.
- [8] **Sriskandarajah, C., Hall, N. G., ve Kamoun, H.,** (1998). Scheduling large robotic cells without buffers. *Annals of Operations Research* 76, 287-321.
- [9] **Aneja, Y. P., ve Kamoun, H.,** (1999). Scheduling of parts and robot activities in a two machine robotic cell. *Computers & Operations Research* 26, 4, 297-312.
- [10] **Hall, N. G., Kamoun, H., ve Sriskandarajah, C.,** (1998). Scheduling in robotic cells: complexity and steady state analysis. *European Journal of Operations Research* 109, 43–65.
- [11] **Hurink, J., ve Knust, S.,** (2002). A tabu search algorithm for scheduling a single robot in a job-shop environment. *Discrete Applied Mathematics* 119, 1, 181–203.
- [12] **Soukhal, A., ve Martineau, P.,** (2005). Resolution of a scheduling problem in a flowshop robotic cell. *European Journal of Operational Research* 161, 1, 62-72.
- [13] **Carlier, J., Haouari, M., Kharbeche, M., ve Moukrim, A.,** (2010). An optimization based heuristic for the robotic cell problem. *European Journal of Operational Research* 202, 3, 636–645.

- [14] **Zahrouni, W., ve Kamoun, H.**, (2012). Sequencing and scheduling in a three-machine robotic cell. *International Journal of Production Research* 50, 10, 2823–2835.
- [15] **Nawaz, M., E. E., ve Ham, I.**, (1983). A heuristic algorithm for the machine, n-job flow-shop sequencing problem. *OMEGA* 11, 1, 91-95.
- [16] **Batur, G. D., Karasan, O. E., ve Akturk, M. S.**, (2012). Multiple part type scheduling in flexible robotic cells. *International Journal of Production Economics* 135, 2, 726–740.
- [17] **Fazel Zarandi, M., Mosadegh, H., ve Fattahi, M.**, (2013). Two-machine robotic cell scheduling problem with sequence-dependent setup times. *Computers & Operations Research* 40, 5, 1420–1434.
- [18] **Elmi, A., ve Topaloglu, S.**, (2014). Scheduling multiple parts in hybrid flow shop robotic cells served by a single robot. *International Journal of Computer Integrated Manufacturing* 27, 12, 1144–1159.

Url-1<<http://www.robotics.org>>, alındığı tarih: 20.4.2015

Url-2<<http://people.brunel.ac.uk/mastjbj/jeb/orlib/flowshopinfo.html>>, alındığı tarih: 12.11.2014

EKLER

EK 1 : Üretilen problem verilerinin özellikleri

EK 2 : Çevrimiçi kütüphaneden alınan problem verilerinin özellikleri

EK 3 : Matematiksel model ve sezgisel algoritmaların sonuçları-1

EK 4 : Matematiksel model ve sezgisel algoritmaların sonuçları-2

EK 5 : Matematiksel modele başlangıç çözümü verildiğinde alınan Sonuçlar-1

EK 6 : Matematiksel modele başlangıç çözümü verildiğinde alınan Sonuçlar-2



EK 1

Çizelge Ek1: Üretilen problem verilerinin özellikleri

Problem	δ	Parça (n)	Makine (m)
1	S	3	4
	L	3	4
2	S	3	5
	L	3	5
3	S	3	6
	L	3	6
4	S	4	3
	L	4	3
5	S	4	4
	L	4	4
6	S	4	5
	L	4	5
7	S	5	3
	L	5	3
8	S	5	4
	L	5	4
9	S	5	5
	L	5	5
10	S	6	3
	L	6	3
11	S	6	4
	L	6	4
12	S	6	5
	L	6	5

EK 2

Çizelge Ek2: Çevrimiçi kütüphaneden alınan problem verilerinin özellikleri

Problem	δ	Parça (n)	Makine (m)
13	S	11	5
	L	11	5
14	S	13	4
	L	13	4
15	S	12	5
	L	12	5
16	S	14	4
	L	14	4
17	S	10	6
	L	10	6
18	S	8	9
	L	8	9
19	S	7	7
	L	7	7
20	S	8	8
	L	8	8

EK 3

Çizelge Ek3: Geliştirilen yöntemlerden elde edilen sonuçlar-1

Problem	Delta	Döngü Zamanı (% Sapma)				Çözüm Zamanı		
		Altsınır	GAMS	<i>n</i> -birim	1-birim	GAMS	<i>n</i> -birim	1-birim
1	S	3557	3557	3557	3624	2.484	0,281	0,33
	L	6695	6695	6695	6937	3.062	0,234	0,34
2	S	4532	4532	4532	4638	44.203	0,547	0,46
	L	10522	10522	10522	10638	87.110	0,531	0,54
3	S	6572	6572	6572	7184	2.763.765	0,641	0,56
	L	13930	13930	13930	14629	1.324.562	0,609	0,67
4	S	6582	7095	7044	7305	ZL	1,032	0,67
	L	12810	13464	13455	14448	ZL	0,969	0,62
5	S	5070	5070	5070	5070	118.532	0,438	0,61
	L	7840	7840	7840	8010	88.000	0,453	0,81
6	S	6452	6705	6662	6898	ZL	0,844	0,79
	L	9632	9687	9687	10120	ZL	0,719	0,95
7	S	10170	10715	10625	11181	ZL	1,672	0,94
	L	19110	19718	19241	20350	ZL	1,282	1,15
8	S	13698	ÇY	14050	14628	ZL	2,39	1,16
	L	27120	ÇY	27479	29460	ZL	1,547	0,91
9	S	9771	10058	9997	10343	ZL	0,563	0,92
	L	15204	15366	15522	16103	ZL	0,531	1,19
10	S	12884	13998	13241	14297	ZL	1,531	1,19
	L	18376	18932	18687	20057	ZL	1,5	1,47

EK 4

Çizelge Ek4: Geliştirilen yöntemlerden elde edilen sonuçlar-2

Problem	Delta	Altsınır	Döngü Zamani (% Sapma)			Çözüm Zamani		
			GAMS	<i>n</i> -birim	1-birim	GAMS	<i>n</i> -birim	1-birim
11	S	18385	ÇY	18769	20083	ZL	2,093	1,46
	L	34835	ÇY	35861	37683	ZL	2,344	1,65
12	S	13650	ÇY	14095	15192	ZL	4,36	1,64
	L	31506	ÇY	32881	34793	ZL	3,797	2,74
13	S	17644	ÇY	18942	20177	ZL	30,605	6,469
	L	27577	ÇY	29042	30289	ZL	26,612	6,625
14	S	18720	ÇY	20270	21762	ZL	33,792	5,094
	L	27378	ÇY	28457	28945	ZL	19,664	4,97
15	S	46296	ÇY	47309	49971	ZL	33,412	8,094
	L	63672	ÇY	65440	68211	ZL	27,549	7,938
16	S	35910	ÇY	37064	37671	ZL	23,113	5,624
	L	43050	ÇY	43600	44517	ZL	20,686	5,532
17	S	29680	ÇY	31489	31905	ZL	33,305	10,408
	L	46870	ÇY	47834	48561	ZL	30,095	10,502
18	S	40912	ÇY	43086	43937	ZL	74,778	32,876
	L	54960	ÇY	57387	58947	ZL	68,642	29,126
19	S	21623	ÇY	22941	23324	ZL	20,812	10,532
	L	36477	ÇY	37358	39256	ZL	18,206	10,406
20	S	35288	ÇY	36769	36413	ZL	53,21	18,532
	L	52680	ÇY	54167	54640	ZL	52,872	18,408

EK 5

Çizelge Ek5: Matematiksel modele başlangıç çözümü verildiğinde alınan sonuçlar-1

Problem	Delta	Döngü Zamanı (% Sapma)				Çözüm Zamanı		
		Altsınır	GAMS	<i>n</i> -birim	1-birim	GAMS	<i>n</i> -birim	1-birim
1	S	3267	3557	3557	3624	2,22	7,11	3,33
	L	6345	6695	6695	6937	2,84	6,74	3,43
2	S	4320	4532	4532	4638	54,95	14,81	4,58
	L	10440	10522	10522	10638	92,66	23,14	5,44
3	S	6380	6572	6572	7184	2385,39	22,69	5,62
	L	13695	13930	13931	14629	2415,50	24,29	6,69
4	S	6582	7044	7044	7305	ZL	31,62	6,66
	L	12810	13455	13455	14448	ZL	25,31	6,18
5	S	5013	5070	5070	5070	90,41	12,43	6,07
	L	7806	7840	7937	8010	80,41	21,40	8,06
6	S	6452	6662	6662	6898	ZL	22,68	7,87
	L	9632	9644	9687	10120	ZL	34,47	9,50
7	S	10170	10625	10625	11181	ZL	34,54	9,39
	L	19110	19434	19434	20350	ZL	36,68	11,51
8	S	13698	14102	14102	14628	ZL	57,19	11,60
	L	27120	27268	27268	29460	ZL	57,83	9,12
9	S	9771	9997	9997	10343	ZL	27,59	9,20
	L	15204	15366	15366	16103	ZL	27,92	11,94
10	S	12884	13241	13241	14297	ZL	47,13	11,86
	L	18376	18884	18884	20057	ZL	39,10	14,69

EK 6

Çizelge Ek6: Matematiksel modele başlangıç çözümü verildiğinde alınan sonuçlar-2

Problem	Delta	Döngü Zamanı (% Sapma)				Çözüm Zamanı		
		Altsınır	GAMS	<i>n</i> -birim	1-birim	GAMS	<i>n</i> -birim	1-birim
11	S	18385	18769	18769	20083	ZL	53,11	14,56
	L	34835	35444	35444	37683	ZL	67,16	16,54
12	S	13650	14095	14095	15192	ZL	103,31	16,39
	L	31506	33133	33133	34793	ZL	106,00	27,42
13	S	17644	18942	18942	20177	ZL	370,68	27,00
	L	27577	29042	29042	30289	ZL	302,86	20,03
14	S	18720	20270	20270	21762	ZL	263,93	20,53
	L	27378	28457	28457	28945	ZL	291,75	28,83
15	S	46296	47309	47309	49971	ZL	450,44	29,68
	L	63672	65440	65440	68211	ZL	297,81	21,38
16	S	35910	37064	37064	37671	ZL	468,54	21,49
	L	43050	4ZL	4ZL	44517	ZL	433,18	35,93
17	S	29680	31582	31489	31905	ZL	359,09	36,56
	L	46870	47834	47834	48561	ZL	643,38	68,25
18	S	40912	43086	43086	43937	ZL	951,60	67,91
	L	54960	57387	57387	58947	ZL	584,00	35,90
19	S	21623	23206	22941	23324	ZL	307,72	35,32
	L	36477	37610	37358	39256	ZL	336,93	53,09
20	S	35288	36413	36769	36413	ZL	721,02	52,42
	L	52680	54167	54167	54640	ZL	813,09	89,23

ÖZGEÇMİŞ

Ad-Soyad : Betül ÇOBAN
Uyruğu : T.C
Doğum Tarihi ve Yeri : 12.06.1991-Almanya/Völklingen
E-posta : cobanbetull@gmail.com

ÖĞRENİM DURUMU:

- **Lisans** :2014, TOBB Ekonomi ve Teknoloji Üniversitesi, Mühendislik Fakültesi, Endüstri Mühendisliği
- **Yüksek lisans** : 2015, TOBB Ekonomi ve Teknoloji Üniversitesi, Endüstri Mühendisliği Anabilim Dalı, Endüstri Mühendisliği

MESLEKİ DENEYİM VE ÖDÜLLER:

Yıl	Yer	Görev
2014-2016	TOBB Ekonomi ve Teknoloji Üniversitesi	Burslu Yüksek Lisans Öğrencisi

YABANCI DİL: İngilizce, Arapça

TEZDEN TÜRETİLEN YAYINLAR, SUNUMLAR VE PATENTLER:

- **Coban, B.**, Gultekin, H., Akhlaghi, V.E., 2015. Değişik Tip Parça Üreten Robotlu Akış Hücrelerinde Hat Dengeleme ve Çizelgeleme, YAEM 2015: Yöneylem Araştırması ve Endüstri Mühendisliği 35. Ulusal Kongresi, Eylül 9-11, Ankara-Türkiye.
- Akhlaghi, V.E., Gultekin, H., **Coban, B.**, 2015. Shortest k-unit Cycle in a Multiple Part-Type Robotic Cell, MISTA 2015: Multidisciplinary International Conference on Scheduling, August 25 - 28, Prague-Czech Republic.
- Gultekin, H., **Coban, B.**, Akhlaghi, V.E.,2016. Robotik Akış Atölyelerinde Parça Ve Robot Hareketlerinin Beraber Sıralanması, YAEM 2016, Yöneylem Araştırması ve Endüstri Mühendisliği 36. Ulusal Kongresi, Temmuz 13-15, İzmir-Türkiye.

