

**ATANACAK EKİP ÖZELLİKLERİNE GÖRE  
DEĞİŞKEN SÜREYE SAHİP PROJELERİN  
PLANLANMASI İÇİN GENETİK ALGORİTMA  
BAZLI BİR YÖNTEMİN GELİŞTİRİLMESİ**

**Ömer Nuri ÇAM**



T.C.  
ULUDAĞ ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

**ATANACAK EKİP ÖZELLİKLERİNE GÖRE DEĞİŞKEN SÜREYE SAHİP  
PROJELERİN PLANLANMASI İÇİN GENETİK ALGORİTMA BAZLI BİR  
YÖNTEMİN GELİŞTİRİLMESİ**

**Ömer Nuri ÇAM**

Doç.Dr. Cenk ÖZMUTLU  
(Danışman)

YÜKSEK LİSANS TEZİ  
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

BURSA-2011  
**Her Hakkı Saklıdır.**

## TEZ ONAYI

Ömer Nuri ÇAM tarafından hazırlanan “Atanacak Ekip Özelliklerine Göre Değişken Süreye Sahip Projelerin Planlanması İçin Genetik Algoritma Bazlı Bir Yöntemin Geliştirilmesi” adlı tez çalışması aşağıdaki jüri tarafından oy birliği/oy çokluğu ile Uludağ Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

**Danışman :** Doç.Dr. Cenk ÖZMUTLU

<b>Başkan:</b>	Doç.Dr. Cenk ÖZMUTLU U.Ü. Mühendislik Fakültesi Endüstri Mühendisliği Anabilim Dalı	İmza
<b>Üye:</b>	Doç.Dr. Nursel ÖZTÜRK U.Ü. Mühendislik Fakültesi Endüstri Mühendisliği Anabilim Dalı	İmza
<b>Üye:</b>	Prof.Dr. Kemal SEZEN U.Ü. İ.İ.B Fakültesi Ekonometri Anabilim Dalı	İmza
<b>Üye:</b>	Prof.Dr. Ahmet ÖZTÜRK U.Ü. İ.İ.B Fakültesi Ekonometri Anabilim Dalı	İmza
<b>Üye:</b>	Yard.Doç.Dr. Tülin Gündüz CENGİZ U.Ü. Mühendislik Fakültesi Endüstri Mühendisliği Anabilim Dalı	İmza

**Yukarıdaki sonucu onaylarım**

**Prof. Dr. Kadri ARSLAN**

**Enstitü Müdürü**

**../../....(Tarih)**

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

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

**beyan ederim.**

.././....

İmza

Ömer Nuri ÇAM

## ÖZET

Yüksek Lisans Tezi

### ATANACAK EKİP ÖZELLİKLERİNE GÖRE DEĞİŞKEN SÜREYE SAHİP PROJELERİN PLANLANMASI İÇİN GENETİK ALGORİTMA BAZLI BİR YÖNTEMİN GELİŞTİRİLMESİ

**Ömer Nuri ÇAM**

Uludağ Üniversitesi

Fen Bilimleri Enstitüsü

Endüstri Mühendisliği Anabilim Dalı

**Danışman:** Doç.Dr. Cenk ÖZMUTLU

Her iş kullanılan kaynaklara göre farklı sürelerde bitirilebilmektedir. Tüm işletmeler en kısa zamanda en az kaynak kullanarak en kaliteli üretimi ve hizmeti verebilmeyi hedeflemektedirler. Bu nedenle daha pahalı makineler alınmakta ve daha kalifiye işçiler çalıştırılmaktadır. Bu tür kaynaklar kısıtlı olduğu için kaynak dağılımının iyi bir şekilde yapılması gerekmektedir. Birbirine bağlı işler arasında kalan zamanlarda mümkünse kaynakların farklı amaçlarla da kullanılması randımanı artıracaktır.

Projeler içerisinde benzer bilgi, kaynak ve başka benzer ihtiyaçlar içeren işler bulunabilmektedir. Bu işler arasında benzer işleri yapabilen kaynakların atanması mümkün olabilmektedir. Bu durumda işgücünün daha iyi paylaşılacağı kesindir.

Bu işleri gerçekleştiren kaynaklar içinde insan en önemli faktörlerden biridir. İnsanlar tek bir yeteneğe de sahip değildirler. Bu farklı yönlerinin kullanılması işçilerin de motivasyonunu artıracaktır.

Bu çalışma ile, birden çok yeteneği bulunan işçilerin veya kaynakların daha etkin olarak kullanılabilmesi gösterilmiştir. Kullanılan genetik algoritma ile heterojen kaynakların / işgücünün iş atamaları yapılmış ve daha az işçi veya kaynak ile işlerin yapılabilmesi ve maliyetlerin düşürülebileceği gösterilmiştir.

**Anahtar Kelimeler:** Kaynak kısıtlı proje çizelgeleme, Esnek işgücü planlama, Ekip atama, Uzmanlık alanına göre kaynakları atama, Genetik algoritma  
**2011, viii + 119 sayfa**

**ABSTRACT**

MSc Thesis

**DEVELOPMENT OF GENETIC ALGORITHM METHOD FOR PROJECT  
PLANNING WITH VARIABLE TIME DEPENDS ON ASSIGNED CREW  
PROPERTIES****Ömer Nuri ÇAM**

Uludağ University

Graduate School of Natural and Applied Sciences

Department of Industrial Engineering

**Supervisor:** Assoc.Prof.Dr. Cenk ÖZMUTLU

Finishing time of jobs varies based on type of resources. All businesses aim to give best quality as soon as possible to their productions and services while using least resource. For this reason, more expensive and more skilled workers/machines are employed. This kind of resources are limited so needs to be balanced in a good way. Interconnected processes between the difference purposes may share same resources. This will result good efficiency.

Within projects can be found similar needs and informations. So it is possible to assign same resources to these jobs. This will result the best share of the workforce.

One of the most important factors in bussiness sector is the human. People do not have a single skill. So if they use their different skills they may be more happy. This will increase their motivation.

This study proves that more than one skilled workers can be assigned more efficient. Genetic algorithm is used with the heterogeneous resources/workforces job assignments that shows fewer workers or resources may be done same jobs while reducing costs.

**Key words:** RCPSPP, Flexible workforce planning, crew assignment, assigning the resources according to area of expertise, Genetic algorithm  
**2011, viii + 119 pages.**

## ÖNSÖZ VE TEŞEKKÜR

Yüksek lisans eğitimine başlama aşamasında ve eğitimi aşamasında karşılaştığım zorluklarda bana sonsuz yardımları dokunan Doç. Dr. Cenk ÖZMUTLU ve Doç. Dr. Seda ÖZMUTLU hocam başta olmak üzere Uludağ Üniversitesi Endüstri Mühendisliği Bölümü hocalarıma ve Prof. Dr. Kasım ÖZLÜK hocama sonsuz teşekkürlerimi arz ederim.

Bana yardımcı dokunan yüksek lisans ve hazırlık aşamasında beraber ders aldığım lisans arkadaşlarıma da teşekkürlerimi sunarım. Aldığımız eğitime adaptasyon aşamasında yardımlaştığımız Ümit Yılmaz arkadaşıma ve diğer sınıf arkadaşlarım ile değerli teyzemin kızı Elif Nadide AKAY'A da teşekkür ederim.

Başta ablam Betül ÇAM ETÖZ ve eşim Özlem Hanım olmak üzere anneme, babama ve abime gösterdikleri yardımlar ile beni teşvik ettikleri için karşılığımı veremeyeceğimi bilerek, teşekkürler ederim.

Ömer Nuri ÇAM

.. / .. / .....

## İÇİNDEKİLER

	<b>Sayfa</b>
ÖZET.....	i
ABSTRACT.....	ii
ÖNSÖZ VE TEŞEKKÜR.....	iii
İÇİNDEKİLER.....	iv
KISALTMALAR DİZİNİ.....	v
ŞEKİLLER DİZİNİ.....	vii
ÇİZELGELER DİZİNİ.....	viii
1. GİRİŞ.....	1
2. KURAMSAL BİLGİLER.....	3
2.1. Proje Çizelgeleme ve Sıralama.....	4
2.1.1. Temel kavramlar.....	5
2.1.2. Öncüllük ilişkileri.....	9
2.1.3. Temel Proje hedefleri.....	10
2.2. Kaynaklar.....	23
2.2.1. Kaynak çeşitleri.....	23
2.3. Kaynak Kısıtlı Proje Çizelgeleme Problemleri (RCPPSP).....	25
2.4. Paralel makine problemleri.....	29
2.5. Çok Modlu Kaynak Kısıtlı Proje Çizelgeleme Problemleri (MRCPPSP).....	31
2.6. Ekip Oluşturma ve Atama.....	34
2.7. Servis/Hizmet Merkezleri.....	37
2.8. Önceki Çalışmalar.....	41
3. MATERYAL VE YÖNTEM.....	44
3.1. Tanımlama.....	44
3.2. Verilerin Oluşturulması.....	46
3.3. Kullanılan Problem Setlerinin Özellikleri.....	46
3.3.1. Kullanılan problem setlerinin çalışmaya ait ek özellikleri.....	48
3.4. Uygulamada Kullanılan Problemler Ve Parametreler.....	54
3.4.1. Kullanılan problemler.....	55
3.5. Matematiksel Model.....	57
3.5. Genetik Algoritma parametreleri.....	60
3.7. Kaynak Atama.....	69
3.8. Uygulama.....	70
3.7.1. Çözüm süresi.....	73
4. BULGULAR.....	76
4.1. Deterministik algoritmaların karşılaştırılması.....	76
4.2. Sonuçların analizi.....	78



4.3. Sonuçların iyileştirilmesi .....	82
5. TARTIŞMA VE SONUÇ .....	90
KAYNAKÇA .....	91
EKLER .....	96
EK-1 .....	97
EK-2 .....	98
EK-3 .....	99
EK-4 .....	100
EK-5 .....	101
EK-6 .....	104
EK -7 .....	110
EK-8 .....	112
EK-9 .....	114
EK-10 .....	116
ÖZGEÇMİŞ .....	119

## KISALTMALAR DİZİNİ

<b>Kisaltmalar</b>	<b>Açıklama</b>
AoA	: (Activity on Arc) Serim üzerindeki aktivite.
AoN	: (Activity on Node) Düğüm üzerindeki aktivite.
CPM	: (Critical Path Method) Kritik yol metodu.
GPRs	: (Generalised precedence relations) Genel öncüllük ilişkileri.
FF	: (Finish-finish relation) Bitiş-bitiş ilişkisi
FS	: (Finish-start relation) Bitiş-başlama ilişkisi
MRCPSP	: (Multi Mode Resource Constrained Project Scheduling Problem) Çok Modlu Kaynak kısıtlı proje çizelgeleme problemi
NP-hard, NP-zor	: (Non Polinomial – hard) Zorluk derecesi çizgisel olarak artmayan, zor problemler.
NP-Tam	: (Non Polinomial – Complete) Çok sayıda işlem ile uzun sürede çözülebilecek problemler.
PSP	: (Project Scheduling Problem) Proje Çizelgeleme Problemi
RCPSP	: (Resource Constrained Project Scheduling Problem) Kaynak kısıtlı proje çizelgeleme problemi
SF	: (Start-finish relation) Başlama-Bitiş ilişkisi
SS	: (Start-start relation) Başlama-Başlama ilişkisi
SWPSP	: (The Skilled Workforce Project Scheduling Problem) Uzman işgücü proje çizelgeleme problemi
PSPLIB	: (Project Scheduling Problem Library) Makalelerde kullanılan proje çizelgeleme problemlerinin bulunduğu kütüphane.

## ŞEKİLLER DİZİNİ

	<b>Sayfa</b>
Şekil 2.1. İşlerin serim üzerinde gösterim (AoA) .....	8
Şekil 2.2. Aktivitelerin düğüm üzerinde gösterimi .....	8
Şekil 2.3. Yerel ve Global En iyi Çözüm.....	15
Şekil 2.4. Ağaç kromozom gösterimi.....	19
Şekil 2.5. Örnek problemin görsel olarak öncüllerinin gösterimi.....	27
Şekil 2.6. Örnek RCPSP problemin kaynaklar üzerinde gösterimi .....	28
Şekil 2.7. Çok modlu kaynak kısıtlı proje çizelgeleme problemi örnek gösterimi .....	32
Şekil 2.8. MRCPSPP örnek problemin gantt şeması ile gösterimi .....	33
Şekil 2.9. Havayolu problemleri çizelgeleme iş akışı.....	35
Şekil 3.1. Geliştirilen algoritmanın temel akış gösterimi.....	45
Şekil 3.2. Çalışmada kullanılan algoritmanın akış çizelgesi.....	72
Şekil 3.3. Bilgisayar işlem süresi grafiği .....	74
Şekil 4.1. İşçi sayısı değişimi.....	87
Şekil 4.2. Bitiş süresi değişimi.....	88
Şekil 4.3. Süre ve Maliyet iyileştirme oranı.....	88
Şekil 4.4. Toplam maliyet değişimi .....	89

## ÇİZELGELER DİZİNİ

### Sayfa

Çizelge 2.1. Öncüllük ilişkili proje çizelgeleme sınıfları (Reyck ve ark,1999).....	11
Çizelge 2.2. Kromozom gösterimi .....	18
Çizelge 2.3. Permutasyon kodlama ile kromozomun gösterimi .....	18
Çizelge 2.4. Değer kodlama ile kromozomun gösterilmesi .....	18
Çizelge 2.5. Tek noktadan çaprazlama .....	20
Çizelge 2.6. Çift noktadan çaprazlama .....	21
Çizelge 2.7. İkili kodlamalı mutasyon gösterimi .....	22
Çizelge 2.8. Değiştirme mutasyon öncesi ve sonrası .....	22
Çizelge 2.9. Lokal arama tabanlı mutasyon gösterimi .....	22
Çizelge 2.10. Sağa kaydırma mutasyon gösterimi .....	22
Çizelge 2.11. Örnek RCPSP problemi .....	26
Çizelge 2.12. Örnek paralel makine problemi .....	29
Çizelge 2.13. Özdeş olmayan makine tiplerine örnek problem .....	31
Çizelge 2.14. Örnek MRCPSPP problemi .....	33
Çizelge 3.1. Proje Özellikleri (problem no: c154_3) .....	47
Çizelge 3.2. Öncüllük ilişkileri (problem no: c154_3) .....	47
Çizelge 3.3. Kaynak tüketimi ve iş süresi (problem no: c154_3) .....	48
Çizelge 3.4. Beceri düzeyleri .....	48
Çizelge 3.5. Beceri düzeyleri .....	48
Çizelge 3.6. Uzmanlık alanları .....	48
Çizelge 3.7. İşçi özellikleri,(uzmanlık alanı , beceri düzeyi karşılığı sırasıyla) .....	49
Çizelge 3.8. İş özellikleri (MRCPSPP'ye uyumlu gösterim) .....	49
Çizelge 3.9. İki kaynak için işçi kombinasyonu .....	52
Çizelge 3.10. Önce belirlenen işçilik seviyesine göre iş tanımı .....	52
Çizelge 3.11. Belirlenen işçi ihtiyacına göre işçi kombinasyonu .....	53
Çizelge 3.12. Kolaylaştırılmış ek sütunlarla olan gösterim .....	54
Çizelge 3.13. Kullanılan orjinal problemlerin özellikleri .....	55
Çizelge 3.14. Çaprazlama oranı karşılaştırma tablosu .....	66
Çizelge 3.15. Mutasyon oranı karşılaştırma tablosu .....	67
Çizelge 3.16. Popülasyon boyutu karşılaştırma tablosu .....	68
Çizelge 3.17. İşçi seçim öncelik kuralları .....	70
Çizelge 3.18. Bilgisayar hesap süresi tablosu .....	74
Çizelge 4.1. Deterministik algoritmalarla oluşan bulunan en iyi uygulama sonuçları ...	77
Çizelge 4.2. Orijinal problemler için maliyetler .....	80
Çizelge 4.3. Yeni problemler için maliyetler .....	81
Çizelge 4.4. Minimum işçi sayısı sonrası maliyetler .....	83
Çizelge 4.5. II. Optimizasyon sonrası tamamlanma süresi .....	85
Çizelge 4.6. Optimizasyonlar sonrası iyileştirme oranları .....	86

## 1. GİRİŞ

Üretim yapan ve hizmet veren işletmelerin karşılaştıkları problemlerden en önemlilerinden biri insan gücünün doğru bir şekilde değerlendirilmesidir. İşgücünün maliyette en önemli parametrelerden biri olduğu gerçeği, tüm dünyada, üretimin, işgücünün düşük olduğu ülkelere yönlendirilmesinden anlaşılabilir. Gelişmiş ülkeler, daha düşük işgücü maliyeti olan ülkelere üretimlerini yönlendirebilirler. Ancak en azından hizmet birimleri için bu mümkün olamamaktadır. Üreterek güçlenen ülkeler de işgücü maliyetlerini düşürmek için yeni hesaplar yapmaktadırlar.

Bu çalışma ile benzer işler için benzer işgücü ihtiyacı söz konusu olduğu durumlarda işgücünün esnek olarak kullanılabilmesi ve çok etkin bir işgücü planlamasının yapılabileceği gösterilmeye çalışılmıştır. Örnek olarak arama hizmeti veren bir işletmede, her işçi telefona cevap verebilecek altyapıya sahiptir. Bazı işçiler ise birden fazla yeteneğe ve uzmanlığa sahip oldukları için, aşırı talep durumunda çalıştığı birimler arası kaydırma yapılabilir.

İşgücünün optimize edilmesi problemleri genel olarak NP-zor problemlerdendir. İşgücü başlı başına bir kaynak kısıdı olarak bulunmaktadır. Bu çalışma ile gerçek hayatı daha iyi temsil eden örnekler geliştirilmiştir. Kaynaklara uzmanlık alanları ve beceri seviyeleri gibi yeni bir kısıt daha eklenmektedir. Ekip oluşturma aşamasında bu yeni kısıtlar göz önüne alınarak oluşturulduğu için ek hesaplama yükü getirmektedir. Yetenekli işçilerin işleri daha kısa sürede yapabilmeleri planlama hesaplarını daha da zor kılmaktadır. İşler arasında öncelik, sonralık gibi ilişkiler söz konusudur.

Bu çalışma en genel anlamda bir proje çizelgeleme problemidir. Esnek işgücü planlaması, işgücünün kaynak kısıdı oluşturması sebebiyle RCPS (Kaynak kısıtlı proje çizelgeleme) problemlerine, uzmanlıkların birer mod olarak kabul edilmesi sebebiyle MRCPSP (Çok modlu proje çizelgeleme) problemlerinin yaklaşımlarına benzemektedir. Ekip oluşturma gerekliliği sebebiyle ekip planlama (crew rostering) problemlerine, bazı ekiplerin işleri daha kısa ya da uzun sürede bitirebilmeleri sebebiyle aynı olmayan (uniform, unrelated, non-identical) paralel makine problemlerine ve işleme süresi

ayarlanabilen problemlere ve esnek işgücünün birimler arası aktarılmasının en çok kullanıldığı çağrı merkezi problemlerine benzemektedir. Bu çalışmalar ışığında literatüre yeni bir yaklaşım sunulmuştur. Bu yaklaşımlar neticesinde aşağıdaki özellikler getirilmiştir.

- Gerçek hayatı daha iyi temsil eden proje verileri,
- Her işin kendine özel bilgi alanları gerektirmesi ve bu alanları farklı seviyelerde yapabilen ya da bilen kaynakların veya işçilerin atanması,
- Heterojen (aynı anda iyi ve kötü uzman kaynakların bulunması) kaynak ataması,
- Birden fazla kaynağın veya işçinin, işin süresi üzerindeki etkileşimleri,
- İşlerin uzmanlık seviyelerine göre atanabilmesi,
- Minimum işçi sayısının bulunması,
- Fason işçi alımının sonuçlar üzerindeki etkisi.

## 2. KURAMSAL BİLGİLER

Bu bölümde çalışmaya temel teşkil eden konular hakkında kavramsal bilgiler ve açıklamalar verilmeye çalışılacaktır.

Bu çalışma en temel anlamda bir proje çizelgeleme çalışmasıdır. Bu nedenle kuramsal bilgilerde ilk olarak proje çizelgelemeye değinilecektir. Proje çizelgeleme konusu altında öncüllük ilişkileri, temel kavramlar ve hedef fonksiyonlarından kısaca bahsedilmeye çalışılmıştır.

Proje çizelgeleme konusunda en önemli kaynaklardan biri işgücüdür. Bu çalışmanın temel kısıtı olan işgücü optimizasyonu bir kaynak atama problemidir. Bu nedenle ikinci olarak, proje çizelgeleme konusunun daha spesifik bir örneği olan, kaynak kısıtlı proje çizelgeleme konusuna değinilecektir.

Giriş kısmında bahsedildiği üzere, bu çalışmada tek bir işe birden fazla işçi/kaynak atanmaktadır. Bu yaklaşım ekip atama problemlerini anımsatmaktadır. Üçüncü başlık olarak ekip atama problemlerinden bahsedilecektir. Bu aşamada, ekiplerin oluşturulması için literatürde uzmanlık alanlarına göre işçilerin dağıtılması problemlerine, özellikle havacılık sektöründeki pilot ve uçak ekibinin belirlenmesi ile sağlık sektöründe hemşirelerin gelen hastalara yönelik dağıtılması problemlerine değinilecektir. Bu çalışma ile ekip atama problemlerindeki temel farklılık açıklanacaktır.

İşçilerin farklı seviyelerde iş görmeleri ve süre üzerindeki farklı etkileri, farklı tipteki makine çizelgeleme problemlerini anımsatmaktadır. Kaynak kısıtlı proje çizelgeleme problemlerinin bu etkiyi göze aldığı çok modlu kaynak kısıtlı proje çizelgeleme problemleri anlatılacaktır.

Son olarak, çalışmanın temelini teşkil eden esnek işgücü ihtiyacının en çok kullanıldığı servis ve çağrı merkezi problemleri anlatılacaktır. Çağrı ve hizmet merkezlerinde benzer

işlerin çokluğu nedeniyle literatürde en çok esnek işgücü terimi bu konularda geçmektedir.

Bu çalışmalar ışığında literatürdeki çalışmalar ile yapılan bu çalışma arasında benzerlikler, farklılıklar ve geliştirmeler ilgili konular içerisinde anlatılacaktır.

## 2.1. Proje Çizelgeleme ve Sıralama

Proje ifadesinin farklı kişiler için farklı anlamları vardır. Tüm projeler için geçerli tek bir uygun tanım bulunmamaktadır. Fakat ISO 8402 (1990) tanımlaması büyük bir oranla kabul görmüştür (Demeulemeester, ve ark., 2002). Bu tanımlamaya göre;

Proje, bir hedefi gerçekleştirmek için, kısıtları özel ihtiyaçlarına göre zaman, maliyet ve kaynaktan oluşan, başlama ve bitiş zamanları kontrol edilebilen tek bir işlemdir. 1997’de düzeltilen haliyle ISO 10006 Proje yönetiminde kalite rehberi maddesi, bu tanımlamayı geliştirerek maddeleştirmiştir (Demeulemeester, ve ark., 2002)

Tüm proje tanımları için genel ifadeleri Slack ve arkadaşları (1998) aşağıdaki gibi özetlemişlerdir.

- **Amaç veya hedef:** Çıktı olarak bir sonuca ulaşmaktır. Genellikle maliyet, kalite ve bitiş süresi ile ilgilidir.
- **Benzersizlik:** Proje, kendi süresi içerisinde bir kerelik oluşan bir olgudur. Tekrarlanması benzersizlik ifadesini bozmaz. Çünkü her tekrar, yeni kaynaklar ve çevresel faktörlerle yeni bir proje anlamına gelir.
- **Karmaşıklık:** Çeşitli görevler arasındaki, öncüllük, maliyet gibi ilişkiler sebebiyle karmaşıktırlar.
- **Geçici niteliklilik:** Projelerde tanımlanan başlama ve bitiş süreleri vardır. Bu süreç sonrasında proje biter.
- **Kesin sonuçsuzluk:** Projeler hedefleneni gerçekleştirirken bir risk taşırlar. Gerçek hayatta olduğu gibi, projeler için de kesin konuşulamaz.
- **Hayat döngüsü:** Her projenin başlama, ilerleme ve bitiş aşamaları vardır.



Endüstri mühendisliğinin alt dallarından önemli bir kısmı projelerin çizelgenmesi problemleridir. Çizelgeleme tanımını yaptıktan sonra proje çizelgelemenin temel kavramları ve özelliklerine değinilecektir.

- **Çizelgeleme:** Projedeki aktivitelerin hangi zamanda hangi işlemin, hangi kaynakla yapılacağını tespit edilmesidir.
- **Sıralama:** Projedeki görevlerin, aktivitelerin hangi sıraya göre yapılacağını bulunması işlemidir. Bu noktada öncüllük ilişkileri önem kazanmaktadır.

Bu çalışma temel nitelikte bir proje sıralama araştırmasıdır. Ancak kaynak atama nedeniyle proje çizelgeleme ile de ilgilidir. Bu çalışmanın temel problem kümesi ise, literatürde çözümlerin karşılaştırılabilmesi amacıyla geliştirilmiş örnek problemlerdir. Bu problemler ile uygulanan algoritmalar ve en iyi çözümler karşılaştırılma imkanı bulabilmektedir. (PSPLIB)

### 2.1.1. Temel kavramlar

Her proje kendine özgü bir ihtiyaç neticesinde doğar. Her ihtiyacın da kendine özgü kısıtları bulunmaktadır. Tüm projeler için geçerli en temel kısıt süre kısıtıdır. Her projenin başlama ve bitiş süresi bulunmaktadır. Proje içerisinde bulunan her işlem için de bir başlama ve bitiş zamanı bulunmaktadır. Bunlara ek olarak, her işlemin öncelikleri, öncülleri, son süresi, başlayabilme süresi, işlem süresi ve hazırlık zamanı gibi kısıtları bulunabilir. Bu başlık altında kısaca bu tanımlar açıklanacak ve proje tanımlarına ve çizimlerine ait terminolojik terimler verilecektir.

Proje çizelgeleme problemlerinin sınıflandırılmasını anlatmak için bir notasyon geliştirilmiştir. Bu notasyon şu şekildedir.  $\alpha | \beta | \delta$ . Sırasıyla  $\alpha$  ifadesi, makine problemlerinin cinsini göstermektedir. Akış, açık, iş makine problemleri vb. Örnek notasyonlara aşağıda değinilmiştir (Demeulemeester, ve ark., 2002).

$\alpha$  parametresinin bulunduğu yerde birkaç tanım içerebilir. Bu tanımlar,  $\circ$  tanımı makineden başka bir şey kullanıldığını ifade eder.  $PI$ : Tek makine problemlerini,  $P$  benzer paralel makineleri,  $Q$  düzgün makine tanımlarını,  $R$  benzer olmayan makineleri,  $F$  akış tipi iş tanımı,  $O$  açık iş atölyesi ve  $J$  iş atölyesi tanımlarını ifade eder. İkinci sıradaki alfa tanımı kaç kaynak olduğunu gösterir. Üçüncü sırada kaynak cinsini ifade eder. “1” yenilenebilir kaynağı, “2” yenilenemeyen ve “3” çift kısıtlı kaynakları temsil eder. Dördüncü olarak ise ne kadar kaynağın mümkün olduğunu gösterir.

$\beta$  ise işlem karakteristiğini ifade etmektedir. Yine bu alanda da birden fazla tanım içerebilir. Birinci tanım işlerin bölünebilir olup olmadığını tanımlar. Örneğin,  $pmtn$  tanımı işlerin bölünebilir olduğunu tanımlamaktadır. İkinci tanım ise öncelik ilişkilerini gösterir,  $cpm$ ,  $gpr$  vb. Üçüncü sırada hazır oldukları zaman tanımlarını gösterir. Buna göre hazırlık zamanı sıfır, deterministik, stokastik veya bulanık olabilir. Dördüncü sırada işlerin süre tipleri tanımlanır. Buna göre süre tipleri tamsayı, devamlı, stokastik veya bulanık süre tanımlamaları olabilmektedir. Beşinci sırada, işlerin kesin bitiş süresinin tanımını verir. Örneğin kesin bitiş süresi olmayabilir. Altıncı sırada projenin kaynak ihtiyacını tanımlar. Bu tanıma göre kaynağı belli bir zaman aralığı talep edebilir veya kaynak talebi kesikli(*disc*) veya sürekli(*cont*) bir formüle göre tanımlanır. Yedinci sırada işlerin yapım modu verilmektedir. Bu notasyonda tek modlu, çok modlu gibi işlenme modları bulunmaktadır. Çok modlu işlemde, işin birden fazla şekilde işletilebileceğini tanımlamaktadır. Sekizinci sırada projenin, pozitif para akışı tanımlaması gibi finansal tanımı verilmektedir. Dokuzuncu sırada işlerin zamanlarının değiştiği ifadesi verilmektedir. “*sjk*” ifadesi sıralamaya göre işlerin süresinin değiştiğini göstermektedir.

$\delta$  parametresi ise amaç fonksiyonunu göstermektedir. Örneğin amaç tamamlanma süresinin minimizasyonu ise  $C_{\max}$  olarak yazılmaktadır.

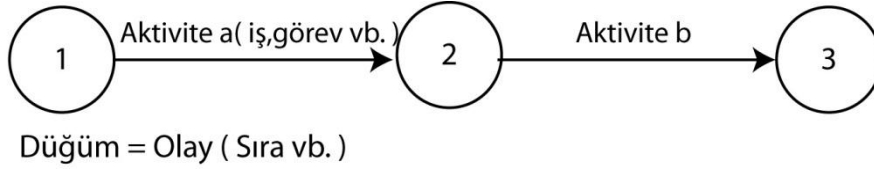
Örneğin ; “**P2 | prec,rj,pj=1 |  $C_{\max}$** “ tanımı, iki benzer paralel makine problemi olup, işlerin öncüllerinin olduğunu hazırlık ve işlem zamanları altında, yenilenebilir kaynak ihtiyacı altında maksimum tamamlanma süresinin minimize edilmesi problemi olarak tanımlanır.

Yukarıda bahsedilen notasyonlar proje tanımlaması için kullanılmaktadır. Proje bilgileri projeyi oluşturan aktivitelerinin tipine göre oluşturulmaktadır. Aktivitelerin tanımlanması için ve projede yerinin belirlenebilmesi için aşağıdaki bilgilerin bilinmesi gerekmektedir.

- **İşlem Süresi:** İşin kendi başlama ve bitiş süresi farkıdır.
- **Önem Derecesi:** İşin önemine ait çarpandır. Hesaplama ağırlığı olarak ta tanımlanabilir.
- **Teslim Süresi:** İşin teslim edilmesi gereken zamandır.
- **Kesin Teslim Süresi:** İşin kesinlikle bitmesi gereken zamandır.
- **Hazırlık Süresi:** İşler arası geçişlerde ya da ilk işin başlaması için gerekli hazırlıkların yapılması için gereken süredir. Problemin tipine göre ulaşma zamanı olarak da tanımlanmaktadır.
- **Hazır Zamanı:** İşlerin önceliklerine göre ya da makinelerin durumuna göre bir işin ilk başlayabilme zamanını temsil eder. İşin öncüllerinin tamamlandığı ve ihtiyacı olan kaynağın müsait olduğu zaman tanımıdır.
- **Akış Süresi:** İşin projenin başlamasından sonra bitiş zamanına kadar geçen süredir.
- **Tamamlanma Süresi:** İşin bittiği zamandır.
- **Geç Kalma Süresi:** Teslim süresi ile bitiş zamanı arasındaki farktır. Negatif olduğu durumlar için iş teslim süresinden erken bitmiş anlamını taşır.
- **Gecikme Süresi:** Geç kalma süresinin sadece pozitif olduğu durumlardır.

Yukarıdaki bilgiler doğrultusunda proje oluşturulmaktadır. Projenin tüm bilgilerinin tablolar halinde rakamlarla yazılması, okunabilirliğini zorlaştırmaktadır. Bunun için proje sıralamalarının grafiksel olarak temsil edilmesine imkan veren ve şebeke modellerinde de kullanılan yöntem kullanılmıştır. Proje temsilleri için gantt şemaları (Clark, 1954) , iz planlama (Herroelen, ve ark., 1998), denge çizgisi (Lumsden, 1968) gibi yöntemler kullanılmıştır. Ancak işlerin, ilişkilerini ve sıralarını temsilen en ideal olarak gösteren ağ modelleridir. Ağ modelinde en iyi bilinen modeller PERT ve CPM'dir. İki temel gösterim vardır.

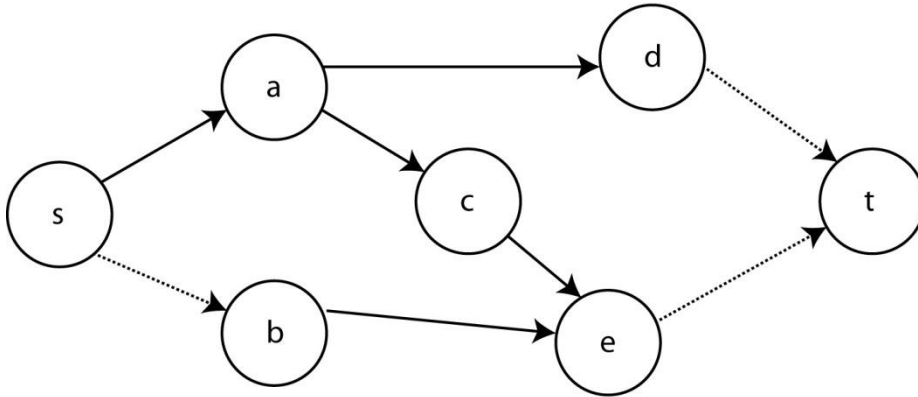
**İşlerin serimde olduğu gösterim:** (AoA) Düğümlerde olayların, serimde ise aktivitelerin temsil edildiği gösterimdir.



Şekil 2.1. İşlerin serim üzerinde gösterim (AoA)

Bu gösterimin avantajları ve özellikleri için ilgili araştırmaya bakılabilir. (bkz. Elmaghraby, 1977, Moder, ve ark., 1983)

**Aktivitelerin düğümde olduğu gösterim:** (AoN) Her aktivite nod diye de ifade edilen, düğümler üzerinde gösterilir. Serimler aktiviteler arası ilişkileri gösterir.



Düğüm = İş

Şekil 2.2. Aktivitelerin düğüm üzerinde gösterimi

Aktiviteler arası ilişkiler çok karışık ilişkilere sahip olabilirler. Ancak tek yönlü oklarla gösterilmeye çalışılır. Böylece hesaplama kolaylaşır ve projeyi daha rahat anlama imkanı olur. Bir sonraki konuda öncüllük ilişki tipleri anlatılacaktır.

### 2.1.2. Öncüllük ilişkileri

Bir önceki konuda bir işin başlaması, diğer işin bitmesine bağlı olan ilişki tipi gösterilmiştir. Ancak gerçek hayatta daha farklı ilişkiler söz konusudur. Ayrıca PERT ve CPM için aktiviteler arası ilişkilerde, gecikme zamanı genel olarak sıfır olarak kabul edilmektedir. Ancak gerçek yaşamda böyle bir konu söz konusu değildir. Elmaghraby ve Kamburowski (1989) bu gerçek yaşamdaki ilişkileri Genel öncüllük ilişkileri (*GPRs*) olarak tanımlamışlardır. En temel anlamda dört çeşit öncüllük ilişkisi vardır.

- **Bitiş-Başlama ilişkisi:** (FS) İlk işin bitmesinden sonra diğer iş başlayabilir. PERT/CPM gösteriminde AoA üzerindeki gösterilebilecek tek tanımdır. En yaygın kullanılan ilişkidir. Örneğin montaj yapılması için, montaj yapılacak parçaların bitmiş olması gerekmektedir.
- **Başlama-Başlama ilişkisi:** (SS) İki iş arasında başlama süresine göre diğer işin başlama süresi ile olan ilişkisi tanımlanır. Örnek olarak beton dökme işleminin başlaması için, beton karma makinesinin başlaması gerekmektedir.
- **Bitiş-Bitiş ilişkisi:** (FF) İki iş arasında bitiş süresine göre diğer işin bitmesi gereken süreyi tanımlar. Örneğin duvarın bitişi ile elektrik tesisatı bitirilebilir.
- **Başlama-Bitiş ilişkisi:** (SF) En az kullanılan ilişkidir. İki iş arasında ikincinin bitişi birincinin başlamasına bağlıdır. Moder ve arkadaşlarının (1983) gösterdiği örnek şu şekildedir. Örneğin, araba vites kutusunun tasarımının başlayabilmesi için, arabanın şase modelinin yapılmış olması gerekmektedir.

Tanımlama şu şekilde yapılır.  $SF_{ij}^{min}(x)$   $i$  işinin başlayabilmesi için, en erken  $j$  işinin bitirilmesi için  $x$  kadar zaman gereklidir.

En iyi bilinen öncelik ilişkisi bitiş-başlama ilişkisidir. En büyük hedef bu aralığın sıfır olmasıdır. Bu durumda boş zaman kalmamış olur. Ancak her zaman böyle olmaz. Çünkü hazırlık süreleri, bitiş sonrası gerekli süreler bu zamanı artırabilirler.

Son olarak, bu ilişkilerin kombine olarak kullanıldığı problemlere ihtiyaç duyulmuştur ve **kombine edilmiş ilişkiler** olarak tanımlanmıştır. (Hajdu, 1997)  $ZZ_{ij}^{min}(x)$  notasyonu ile ifade edilmiştir. Bu ilişki ile, SS ve FF ilişkilerinin her ikisi birden öncüllük ilişkisi olarak alınmaktadır. Örnek olarak da, duvar dökme ve tesisat döşeme verilebilir. SS ilişkisi olarak, tesisatçılar ancak yeterli oranda duvarın dökümüyle başlayabilirler ve FF olarak da, yine belli oranda duvarın dökümü sonrasında işlerini bitirebilirler.

Bu noktaya kadar anlatılan bilgiler var olan bir projenin temel tanımlarını ve tanımlamaları oluşturmaktadır. Her projenin de bir hedefi vardır. Bir sonraki konuda bunlara değinilecektir.

### 2.1.3. Temel Proje hedefleri

Her projenin bir hedefi vardır. Projeler hedeflerine göre bir ya da birden fazla amacı gerçekleştirmek durumuna göre çok sayıda çeşitlilik göstermektedirler. En yaygın hedefler aşağıda verilmiştir.

**Zaman Tabanlı Hedefler:** En yaygın olarak kullanılan amaç fonksiyonu, projenin tamamlanma süresini en iyi bir şekilde kısaltma hedefidir. Bu sayede projenin kesin tamamlama süresinin aşılma riskini azaltma ve kaynakları en iyi kullanma ihtimalin elde edilmiş olur. Böylece iyi bir finansal akış da sağlanmış olur. (Goldratt, 1997) Zamansal hedeflerin içerisinde başka hedefler de bulunabilir. En yaygın diğer hedefler boş zamanı, geç kalan işleri, fazla mesaiyi en küçüklemedir. İşlerin teslim tarihine göre, kesin bitmesi gereken sürelerle göre de zaman tabanlı hedefler yaygın olarak kullanılmaktadır.

**Kaynak Tabanlı Hedefler:** Kaynak kullanımı maliyet açısından önemli bir hedeftir. Bu anlamda maliyet tabanlı hedeflemeye de girebilir. Ancak kaynakların sınırlı olması ve az bulunması tüm maliyet hesaplarının önüne geçebilmektedir. Bu nedenle kaynakların en iyi şekilde kullanılması ve boşa harcanmaması temel hedeflerden biridir. Diğer bir hedef ise, iş başına düşen kaynak miktarının ortalamasının diğer işlerce de standart sapmasını aşmaması için en küçüklemedir. (Burgess, ve ark., 1962) Özellikle

yenilenemeyen kaynaklar var ise, zaman tabanlı hedeflerden daha önem kazanmaktadır. İyi kullanılması maliyet tabanlı hedeflerin de gerçekleştirilmesini sağlayabilmektedir.

**Maliyet Tabanlı Hedefler:** Tüm işletmelerin temel hedefi karlarının en büyüklenmesidir. (Herroelen, ve ark., 1997) Aslında diğer tüm hedefler içinde bu maliyet hedefi bulunmaktadır. Envanterin azaltılması, müşteri memnuniyetinin artırılması, kaynakların doğru kullanılması, işçilerin düzgün dağılımı vb. hedefler maliyeti doğrudan etkilemektedir. Bu başlıklar maliyet hedefi içinde de alınabilir.

**Kalite Tabanlı Hedefler:** Ürünlerde ve üretimde kalitenin artırılması hedeflenmektedir. Kalitenin artırılması ile yeniden işleme ihtimalleri azaltılır, müşteri memnuniyeti hedefleri tutturulabilir. Bu sayede maliyetin düşmesi de sağlanır.

Yukarıda bahsedilen temel proje hedefleri bir arada düşünülerek çok hedefli fonksiyonlar da oluşturulmaktadır. Bu nedenle çok sayıda proje hedefleri bulunmaktadır. Reyck ve arkadaşları (1999) öncüllük ilişkilerinin gözetildiği proje çizelgeleme problemlerinin sınıflandırılmasını Çizelge 2.1'deki gibi özetlemişlerdir.

Çizelge 2.1. Öncüllük ilişkili proje çizelgeleme sınıfları (Reyck ve ark,1999)

	Tek Modlu		Çok Modlu		
	Kaynak tipleri yok, Tercih etme yok.	Çok yenilenebilir kaynaklı, Tercih etme yok.	Bir adet yenilenemeyen kaynak Zaman/maliyet tercih etme var.	Bir adet yenilenebilir kaynak, Zaman/kaynak tercihi var.	Çoklu yenilenebilir ve yenilenemeyen kaynak tipleri, Zaman/maliyet, zaman/kaynak, kaynak/kaynak tercihleri var.
<b>ZERO-LAG</b>	<b>CPM/PERT</b> cpm Cmax	<b>RCPSP</b> m,1 cpm Cmax	<b>DTCTP</b> 1,T cpm,disc,mu Cmax	<b>DTRTP</b> 1,1 cpm,disc,mu Cmax	<b>MRCPS</b> m,1T cpm,disc,mu Cmax
<b>FS MIN</b>	<b>PDM</b> min Cmax	<b>GRCPS</b> m,1,va min,qi,di	<b>GDTCTP 1,</b> T min,qi,disc,mu Cmax	<b>GDTRTP 1,</b> 1 min,qi,disc,mu Cmax	<b>GMRCPS</b> m,1T min,qi,disc,mu Cmax
<b>SS, SF, FS, FF</b>	<b>MPM</b> gpr Cmax	<b>Cmax RCPSP-GPR</b> m,1,va gpr,qi,di,vr Cmax	<b>DTCTP-GPR</b> 1,T gpr,qi,di,disc,mu Cmax	<b>DTRTP-GPR</b> 1,1,va gpr,qi,di,disc,mu Cmax	<b>MRCPS-GPR</b> m,1T,va gpr,qi,di,disc,mu Cmax

### 2.1.3.1. Hedef fonksiyonlarda yaygın kullanılan deterministik algoritmalar

Bu bölümde çözüm için kullanılan deterministik algoritmalara değinilmiştir. Bu algoritmalar iterasyonların daha iyi çözüme yaklaşmasını sağlamak amacıyla sezgisel algoritmaların içerisinde de kullanılabilir (Pinedo, 2008).

#### a) Aktivite öncelikli algoritmalar:

Proje içindeki işlerin veya aktivitelerin önceliğinin dikkate alındığı algoritmadır. Bu önceliğe göre teslim süresi en erken olan iş veya en uzun sürecek iş projede işler sıralanırken süre, maliyet gibi avantajları gözönüne alınarak, en başa veya en sona konulurlar.

- **EDD:** (Earliest Due Date) İşlerin en erken teslim tarihine göre sıralanmasının projede geç kalan işleri minimuma indirdiği algoritmadır. (J. R. Jackson, 1955)
- **SPT:** (Shortest Processing Time) En kısa işlem süresine sahip olan işin ilk sıraya alındığı algoritmadır.
- **LPT:** (Longest Processing Time) En uzun işlem süresine sahip olan işin ilk sıraya alındığı algoritmadır.

#### b) Şebeke tabanlı algoritmalar

Projedeki aktiviteler öncüllük ilişkileri nedeniyle veya kaynakların durumuna göre diğer aktiviteleri beklemek zorunda kalabilirler. Buna göre bir sonraki başlayacak iş sayısının çok yüksek olduğu işlerin geciktirilmesi aynı zamanda diğer işlerin de ötelenmesine neden olmaktadır. Bu algoritmalar, öncüllük, ardıllık veya işin doğrudan kendisinin önemine binaen işlerin öne alındığı veya ötelendiği işlemlerdir.

- **MIS:** (Most Immediate Successors) Ardından gelen ilk işte en müsait olanın ilk olarak yerleştirildiği algoritmadır.
- **MTS:** (Most Total Successors) Ardılı olan iş sayısı fazla olan işin ilk olarak sıraya alındığı algoritmadır.



- **LNRJ:** (Least Non-Related Jobs) Öncülü ya da ardılı olan iş sayısının en az olduğu işin ilk olarak sıraya alındığı algoritmadır.
- **GRPW:** (Greatest Rank Positional Weight) Ağırlığı en yüksek olana göre sıralamanın olduğu algoritmadır.

**c) Kritik yol tabanlı algoritmalar**

Aktivitelerin öncüllükleri nedeniyle, projenin ilk başlangıcında ve belli bir zamanda tüm işler işlenmek için hazır olmayabilirler. Bu nedenle işlenmeye hazır olan işler içerisinde başlayabilme zamanlarına göre aktivitelerin işleme alındığı algoritmalarıdır.

- **EST:** (Earliest Start Time) En erken başlama zamanına sahip olan işin ilk sıraya alındığı algoritmadır.
- **EFT:** (Earliest Finish Time) En erken bitiş zamanına sahip olan işin ilk sıraya alındığı algoritmadır.
- **LST:** (Latest Start Time) En geç başlama zamanına sahip olan işin ilk sıraya alındığı algoritmadır.
- **LFT:** (Latest Finish Time) En geç bitiş zamanına sahip olan işin ilk sıraya alındığı algoritmadır.
- **MSLK:** (Minimum Slack) En az bol vakte sahip olan işin ilk sıraya alındığı algoritmadır.
- **RSM:** (Resource Scheduling Method) Kaynak durumunu baz alarak sıralamanın yapıldığı algoritmadır.

**d) Kaynak tabanlı algoritmalar**

Tüm aktiviteler bir şekilde kaynak tüketirler. Belli bir zaman biriminde tüketilecek kaynak sınırlıdır ve tüm talepleri aynı anda karşılayamayabilir. Bu gibi durumlar için hazır işler içerisinde alınan işler ile kaynağın en iyi şekilde kullanılmasını hedefleyen algoritmalar olarak kaynak tabanlı algoritmalar kullanılır.

- **GRD:** (Greatest Resource Demand) En yüksek kaynak talebine sahip olan işin ilk sıraya alındığı algoritmadır.
- **GCUMRD:** (Greatest Cumulative Resource Demand) En yüksek toplam kaynağı talep eden işin önce sıraya alındığı algoritmadır.
- **RED:** (Resource Equivalent Duration) Kaynak talebi eşit olan aktivitelerin önce sıralandığı algoritmadır.
- **CUMRED:** (Cumulative Resource Equivalent Duration) Toplam kaynak talebi eşit olan aktivitelerin önce sıralandığı algoritmadır.
- **RND:** (Random) Rastgele sıralama algoritması tüm algoritma çeşitleri için kullanılabilir. Buna göre rasgele alınan iş numarasına sahip iş ilk olarak sıralamaya alınır.

Yukarıda bahsedilen algoritmaların ağırlıklı işlem oranları olarak alınabilecek, öncelik çarpanları ile hesaplandığı ve birden fazla aşama içeren çözümler için *kombine* olarak kullanıldığı algoritmalar da bulunmaktadır.

### 2.1.3.2. Hedef fonksiyonlarda yaygın kullanılan sezgisel algoritmalar

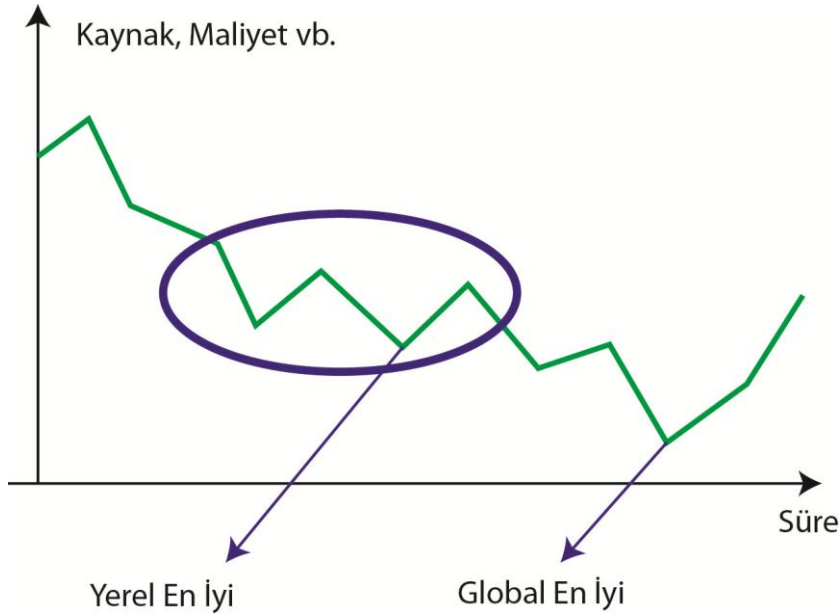
Proje çizelgeleme konuları genel olarak Np-zor problemler sınıfındadır. Bu sebeple kabul edilebilir bir zaman içinde, kabul edilebilir bir iyileşme sağlayan sezgisel algoritmalar kullanılmasını mecbur kılar. Bu sezgisel algoritmalar ile en iyi sonuç çalışılır. Çok sayıda sonuç üretildiği için her sonuç için önceki ve en iyi sonuç birer komşu sonuç olarak adlandırılır. Bu çok sayıdaki komşu sonuçlar arasında en iyisini bulmak için sezgisel algoritmalar komşu sonuç arama ya da yerel arama teknikleri kullanmaları gerekmektedir. Yerel arama tekniklerinde genel olarak şu dört kuralın olması gerekir (Pinedo, 2005).

- i. Sıralanmış bir seri
- ii. Komşu sonuç üreten fonksiyon
- iii. Komşu sonuçları karşılaştıran arama işlemi
- iv. Kabul ya da ret kriterleri.

Sezgisel algoritmalar arasında yukarıdaki özellikleri yapılarında bulunduran en bilinen ve kullanılan algoritmalara aşağıda kısaca değinilmiştir.

### a. Tabu Arama

Aslında tabu arama tekniği insanların günlük hayatta kullandığı bir yöntemdir. Kaybedilen bir şeyi ararken bakılan bir yere bir daha bakılmaması bir tabu kuralıdır. Ancak iyi bakılmadığı düşünülerek geri dönme ihtimali de vardır. Bu nedenle belli seviyede tabu listesi tutulur. Bu yaklaşım fonksiyon haline getirilerek büyük problemler için sonuç arayabilir hale getirilmiştir. İpucu verebilecek çözümlerden ilerler. Komşu çözümler arasında mevcutlar içerisinde en iyisini alarak, sırada oluşturulacak komşu çözümlerin daha iyiye gideceği ihtimalini kullanır. Bununla beraber yerel en iyi ihtimalini aşabilmek için, yeni oluşan komşu çözüm kümesinden daha kötü olanı da seçebilir (Glover, 1990)



Şekil 2.3. Yerel ve Global En iyi Çözüm

Tabu arama tekniği literatüre (Glover, 1989) kazandırılmasından bu yana yirmi yılı aşkın bir süre geçmiş olmasına rağmen esnek kullanım imkânı sebebiyle, büyük problemlerde en iyi sonucu üretecek algoritmalarından biri olarak kullanılmaktadır. Kısaca akış algoritması EK-1 ve EK-2'de grafiksel olarak gösterilmiştir.

### **b. Tavlama Benzetimi**

Tavlama işlemindeki zamanla soğumanın örnek alınarak yapıldığı bir algoritmadır. Sistem en başta çok sıcak ve kararsız haldedir. Algoritma için bu kötü durumları da kabul edebilir anlamını taşımaktadır. Zamanla soğuyarak daha kararlı hale gelir ve daha az kötüyü seçmeye doğru yönelir ve en son olarak da kararlı hale gelmiştir ve sadece en iyi sonuçları seçmeye başlar. Tepe tırmanma algoritmalarındaki yerel en iyiye takılı kalma ihtimali kararsızlığı veya çok sıcak olması sebebiyle aşılmıştır. Daha sonrasında ise mevcut yereller içerisinde en iyisini seçmeye doğru ilerlemelidir. Tüm sezgisel algoritmalarda olduğu gibi mevcut bulunanın en iyi olduğu söylenemez. Tavlama benzetimi algoritması Kirkpatrick ve arkadaşları (1983) tarafından literatüre kazandırılmıştır.

Bu algoritmada temel olarak iki parametre vardır. Birincisi başlangıç sıcaklığının tanımlandığı parametre, diğeri ise soğuma parametresidir. Soğuma parametresi sıfır ve bir arası bir reel sayıdır. Bu sayı iterasyon sayısının üstel derecesi ile güncellenir ve sıfır-bir arası bir değer olduğu için zamanla sıfıra daha yaklaşır. Bu da soğumayı, yani algoritmadaki tanımına göre daha kararlı hale geçmesini sağlar. Yineleme sayısı çok büyük bir rakama ulaştığında sadece en iyiyi seçecek ve bir müddet sonra aynı sonuç tekrarlanmaya başlayacaktır.

Durdurma kriteri olarak maksimum yineleme sayısı, sıcaklık değerinin kabul edilen bir değere ulaştığı durum veya kendini tekrar eden sonuçların belli bir sayıyı aştığı durumlar kullanılır. Akış algoritması EK-3' de verilmiştir.

### **c. Genetik Algoritma**

Genetik algoritma temellerini Darwin'in evrim teorisinden almaktadır. John Holland geçmişi 1960'lara kadar dayanan evrim stratejilerini geliştirerek, Genetik Algoritmalar adını verdiği çalışmayı literatüre kazandırmıştır (Koza, 1992).

Her organizma hücrelerden oluşmaktadır. Ve her hücrede kromozomlar tanımlıdır. Kromozomlar DNA kodlarının sıralandığı bir dizindir. Kromozomlar da genlerden oluşmaktadır. Her gende özel proteinler bulunmaktadır. Her genin kromozom üzerinde kendi yeri bulunmaktadır.

Tüm kromozomların olduğu genetik yapıya da genom adı verilmektedir. Bu genomlar içerisinde alt dizilerin her birine ise genotip denmektedir. Genotipler, fenotiplerden oluşturulmaktadır. Fenotipler, anne ve babanın genotipleridir. Bunlardan yeni genotipler oluşmaktadır. Fenotiplerden yeni genotiplerin oluşması için rekombinasyon gerçekleşmektedir. Bu işlem genetik algoritmada çaprazlama olarak tanımlanmaktadır. Yeni genotipler sadece bu fenotiplerden değil, ayrıca mutasyona da uğrayıp değişerek yeni bireyler oluşturmaktadır. Mutasyonlar genomlarda olan bozulmalardır.

Evrim teorisinde her şeyin daha iyiye gitmesinin bir sebebi vardır. Bu da doğal seleksiyon sayesinde olur. Doğa oluşan yeni organizmayı şartlarına uyamazsa yaşatmamaktadır. Genetik algoritmada bunun karşılığı uygunluk değeridir.

Şekil 2,4'te de gösterildiği üzere global en iyiyi bulabilmek gerekmektedir. Yerel en iyiye takılma ihtimali algoritmanın özelliği nedeniyle düşük bir ihtimaldir. Ancak bu ihtimali de mutasyonlar azaltabilecektir.

Algoritma bir dizi kromozom ile başlar. Bu kromozomlara başlama kromozomları denir ve bu diziyeye de popülasyon denmektedir. Bu popülasyon içerisinde uygunluk değerini geçenler hayatta kalırlar ve kabul edilirler. Sırasıyla bu işlemler kromozom kodlama, çaprazlama, mutasyon, seçim yöntemi ve uygunluk testi başlığı altında beş aşamada açıklanacaktır.

### **i. Kromozom Kodlamaları**

Kromozomdaki her gen bir bilgi tutar. Genetik algoritma nasıl uygulanacağına bağlı olarak bu bilgilerin nasıl kodlanacağı çok büyük önem taşır. Ayrıca bu kodlamalara göre uygunluk testi yapılacaktır. Yeni kodlamalar geliştirilebilir ancak en çok bilinen dört kodlama tekniği aşağıdaki gibidir ( Lab, 2004).

• **İkili Kodlama:** Tüm genler 0 ve 1'lerden oluşurlar. Problemin tipine göre 0 ve 1 ile temsil edilebilir problemler için kullanılır. Bir matematiksel çözümün ikilik karşılığına göre kodlama yapılarak sonuca ilerlenebilir. Bilgisayar alt yapısının da ikili tabanda çalışması sebebiyle bilgisayar algoritmalarında bayt hesaplamalarında kolaylıklar sağlamaktadır. Örnek gösterimi şöyledir;

Çizelge 2.2. Kromozom gösterimi

Kromozom:	1101100100110110
-----------	------------------

• **Permutasyon Kodlama:** Genler işlerin ya da görevlerin, belirlenen numaraların oluşturduğu rakamlardır. Özellikle iş sıralama problemleri için tercih edilir. Genlerdeki rakamların matematiksel bir değeri yoktur. Gendeki işin bulunduğu sıra, gerçek problemdeki istenilen sırayı temsil etmektedir. Bu çalışmada kullanılacak kodlama permutasyon kodlama olacaktır.

Çizelge 2.3. Permutasyon kodlama ile kromozomun gösterimi

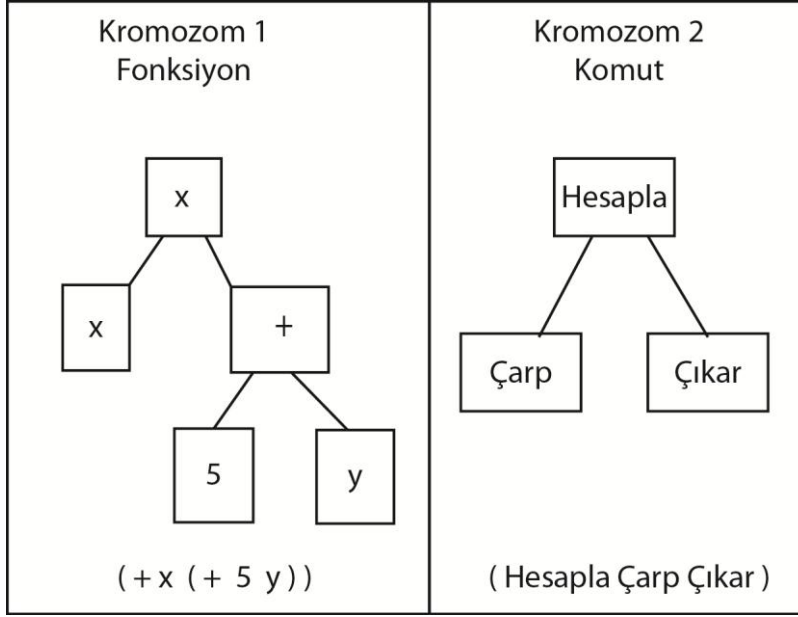
Kromozom	1 5 3 2 6 4 7 9 8
----------	-------------------

• **Değer Kodlama:** Her problem tipi yukarıda bahsedilen, ikili ve permutasyon kodlamaya uygun değildirler. Bu nedenle gerçek değerleri doğrudan kullanmayı zorunlu kılar. Bu nedenle çaprazlama ve mutasyonlarda, problemin özelliğine göre uyulması gereken kurallar bu kodlamayı kullanan geliştirici tarafından tanımlanmak zorundadır. Bu da kodlamanın kullanımını daha zorlaştırmaktadır.

Çizelge 2.4. Değer kodlama ile kromozomun gösterilmesi

Kromozom 1	1.2324 5.3243 0.4556 2.3293 2.4545
Kromozom 2	ABDJEIFJDHDIERJFDLDFLFEGT
Kromozom 3	(sağ,sol,sol,sol,ön, arka,sol,sol,sağ)

• **Ağaç Kodlama:** Ağaç kodlama daha çok, program tanımlamaları ve matematiksel hesaplar için kullanılırlar. Her gen bir fonksiyon ya da komutlar bütünüdür. Değer kodlamadaki gibi çaprazlama ve mutasyon kuralları, bu kodlamayı kullanan geliştiriciye bağlıdır.



Şekil 2.4. Ağaç kromozom gösterimi

Ağaç kodlama kurgulanması ve okunması zor olan bir kodlama türüdür. Herhangi bir tasarım kısıtı da bulunmamaktadır. Kurgusu tamamen tasarımcıya aittir. Örneğin, Şekil 2.4'te verilen, Kromozom 1'de bir fonksiyon dizilimi görülmektedir. Buna göre x değeri sol kolda olduğu gibi alınır veya sağ koldan ilerlenirse, y veya 5 değeri eklenir.

## ii. Seçim Yöntemi

Seçim yöntemi, algoritmanın mevcut sonuçlardan daha iyi bir sonuca götürecek çözümün tercih edilmesi aşamasıdır. Bu sayede daha iyi sonuçlar üretecek bireyler bırakılır ve yeni nesil bunlarla üretilir. Birçok yöntem geliştirilmiştir. Geliştirilen seçim yöntemleri içerisinde en çok bilinenleri ve kullanılanlar, elitist strateji, rulet çarkı ve turnuva seçim yöntemleridir (Agyei, ve ark., 1998).

- **Rulet çarkı yöntemi:** Tüm çözümler hesaplanır ve toplam uygunluk değeri bulunur. Daha sonra her bireyin uygunluk değeri rastgele ya da toplam uygunluk değerine bölünerek elde edilen değerler ile yüzdesel oranları bulunur. Yüzdesel oranları hesaplanan sonuçlar içinden oranı daha yüksek olanlar seçilir.

- **Elitist yöntem:** Kötü nesili daha çabuk yok edebilmek için, kötü sonucu iyi sonuç ile değiştirerek popülasyonun iyileşmesi sağlanır. En iyi sonuç saklanarak yeni nesile aktarılması sağlanır.

- **Turnuva seçim yöntemi:** Rastgele üretilen kromozom numaraları ile o anki kromozom arasında bir kıyaslama yapılır. İki kromozomdan daha iyi sonuç üreten kromozom tercih edilir. Eğer en kötü sonuç, rastgele alınan rakam ile yine kendisi ile karşılaştırılırsa yaşayacaktır. En iyi sonuç ise her ihtimalde seçilecektir.

### iii. Çaprazlama

Fenotiplerden yeni genotip elde edilebilmesi için çaprazlama işlemi uygulanır. Anne ve babadan gelen genlerin bazı kısımları alınarak yani çaprazlanarak yeni çocuk genler elde edilir. Çaprazlama genetik algoritmanın en önemli aşamasından biridir. Çok iyi tasarlanmalıdır. (Alcaraz ve ark. 2001) Konunun daha iyi izah edilmesi için en iyi bilinen çaprazlama yöntemlerine örnekler aşağıda gösterilmiştir.

- **Tek noktadan çaprazlama:** Rastgele alınan bir sıra numarasına göre iki ebeveyn kromozom bölünür, ilk çocuk için ilk bölünen kısmı birinci kromozomdan kalan kısmı diğerinden alınır. Diğer çocuk için işlemin tersi yapılır.

Çizelge 2.5. Tek noktadan çaprazlama

Kromozom 1	<b>11011</b>   <b>00100110110</b>
Kromozom 2	<u>11011</u>   <u>11000011110</u>
Çocuk 1	<b>11011</b>   <u>11000011110</u>
Çocuk 2	<u>11011</u>   <b>00100110110</b>

- **Çift noktadan çaprazlama:** En fazla kromozom uzunluğu kadar iki rastgele rakam alınır, buna göre kromozomlar ikiye ayrılır, iki çocuk için de arada kalan kısım diğer kromozom ile değiştirilerek yeni kromozomlar üretilir. Daha fazla değişik çocuklar istenirse, geliştirilmiş çift noktadan çaprazlama için, aradaki değiştirilen genler ters çevrilerek diğerlerine kopyalanırlar.



Çizelge 2.6. Çift noktadan çaprazlama

Ana Kromozom 1	<b>11011</b>   <b>0010011</b>   <b>0110</b>
Baba Kromozom 2	<u>11011</u>   <u>1100001</u>   <u>1110</u>
Çocuk 1	<b>11011</b>   <u>1100001</u>   <b>0110</b>
Çocuk 2	<u>11011</u>   <b>0010011</b>   <u>1110</u>

• **Diğer çaprazlama yöntemleri:** Çaprazlamanın kalitesi sonucun kalitesine doğrudan etkilidir. Bu nedenle bahsedilen iki temel ve yaygın çaprazlama metodundan farklı olarak çok sayıda çaprazlama teknikleri geliştirilmiştir. Çok noktadan çaprazlama, düzgün çaprazlama, lineer çaprazlama, rasgele çaprazlama gibi çaprazlama yöntemleri vardır.

Çaprazlama yöntemlerinin de kendine mahsus kısıtları olabilmektedir. Örneğin permutasyon kodlama ile yapılan iş sıralama algoritmalarında öncüllük ilişkileri söz konusu olduğundan, çaprazlama yapılırken, bu öncüllük ilişkilerine göre düzenlemeler yapılmak zorundadır. Yoksa algoritma mümkün olmayan anlamsız sonuçlar üretmeye başlayacaktır.

#### iv. Mutasyon

Tepe tırmanma algoritmalarının en büyük problemi yerel iyide çakılmalarıdır. Genetik algoritma için de aynı bu durum söz konusudur. Bu sorunu algoritma mutasyonlar ile aşabilmektedir. Problem çözümleri daha iyiye giderken, mutasyon sonucu oluşan çözümleri de kabul ederek daha iyi global çözümlere ilerlemeye çalışır. Her mutasyonun çözümü daha iyiye götürmeyebilir. Ancak uzun yinelemeler sonucunda iyiye götürdüğü literatürdeki çalışmalar tarafından kanıtlanmıştır. Örneğin RCPSP problemleri için 0,05 oranındaki bir mutasyonun sonuçları iyiye götürdüğü gösterilmiştir. (Hartmann, 1997)

Her kodlama ve programa göre mutasyonların da bazı kısıtlamaları vardır. Öncüllük ilişkilerinin bozulma ihtimali dikkat edilmesi gereken en önemli kriterdir. Böyle bir

durum söz konusu olursa mutasyon yapılmaz. Mutasyon işlemi kodlamalara göre değişmektedir ve aşağıda birkaç örnek gösterilmiştir.

Çizelge 2.7. İkili kodlamalı mutasyon gösterimi

Orijinal Kromozom	110111+0000111+0
Mutasyona uğrayan kromozom	110110 <b>0</b> 1001101 <b>00</b>

En fazla kullanılan mutasyonlardan biri rastgele mutasyon yöntemleridir. Rastgele alınan numaraya göre ikili kodlamalı bitin durumu tersine çevrilir, rastgele alınan rakama göre ilk benzer bit tersine çevrilebilir, düzgün, gaussian vb. rastgele mutasyon yöntemleri uygulanabilmektedir.

Bu çalışmaya temel teşkil edecek olan permutasyon kodlama için mutasyon teknikleri en genel anlamda değiştirme mutasyonu ya da lokal arama tabanlı mutasyondur. Örnek olarak aşağıdaki şekilde gösterilmeye çalışılmıştır.

Çizelge 2.8. Değiştirme mutasyon öncesi ve sonrası

10	11	9	5	3	<b>4</b>	8	7	<b>6</b>	2	1
10	11	9	5	3	<b>6</b>	8	7	<b>4</b>	2	1

Çizelge 2.9. Lokal arama tabanlı mutasyon gösterimi

10	11	9	5	3	<b>4</b>	8	7	6	2	1
----	----	---	---	---	----------	---	---	---	---	---

Çizelge 2.10. Sağa kaydırma mutasyon gösterimi

10	11	9	5	3	<b>4</b>	8	7	6	2	1
----	----	---	---	---	----------	---	---	---	---	---

Bu çalışmada öncüllük ilişkileri nedeniyle sağa kaydırma mutasyonu uygulanacaktır. Sağa kaydırma mutasyonunda rastgele seçilen iş, öncüllük ilişkisini bozmayacak kadar sağa kaydırılır. Bu sağa kaydırma maksimum kaydırma ile minimum kaydırma miktarı arasında rastgele alınan rakamla gerçekleştirilir.

## v. Uygunluk Fonksiyonu

Uygunluk değeri hesaplama yöntemleri hedeflenen amaca göre değişmektedir. Uygunluk değerinin hesaplanması için mutlaka bir matematiksel yöntem gereklidir. (Chen, ve ark., 1997) Örneğin sadece öncüllük uygunlukları denetlenebilir.

Her sezgisel algoritmada olduğu gibi bu algoritmada da durdurma kriterleri belirlenmek zorundadır. Bu kriterler iterasyon sayısı, aynı sonuçları tekrarlama sayısı, ve süre olabilir.

En genel anlamda genetik algoritmanın akışı EK-4'te verilmiştir.

## 2.2. Kaynaklar

Projede yapılmayı bekleyen aktiviteler hariç, tüm aktiviteler bir kaynağa ihtiyaç duyar. Projenin yapılabilmesi için, ihtiyaç duyduğu çeşitli kaynaklardan, farklı oranlarda tüketim yapar. Her kaynak sınırsız değildir. Çok kaynak kullanımı da maliyeti oluşturan büyük etkenlerdendir. İşin yapılabilmesi için gereken tüm ihtiyaçlar birer kaynak olarak tanımlanabilir.

### 2.2.1. Kaynak çeşitleri

Genel olarak kaynak tipleri yenilenebilir, yenilenmeyen ve ikili kısıtlı kaynaklar olarak üçe ayrılırlar. (Blazewicz, ve ark., 1986) Ancak zamanla bu kaynak çeşitlerinin gerçek hayatı tam temsil etmediği düşünülerek bu kaynak tiplerinden başka tipler de tanımlanmıştır. Bu yeni tanımlar, yukarıda bahsedilen üç kaynak tipinin ek özellikleri gibidir.

- **Yenilenebilir kaynaklar:** Bu kaynak tipi projenin ihtiyaç duyduğu her an, proje tarafından meşgul edilmemişse, ulaşılabilir olan kaynaklardır. Belli bir zaman

içerisinde ancak belli bir miktarda olabilirler. İşgücü, makineler, araçlar, yer bu yenilenebilir kaynağın en genel örneklerindedir.

- **Yenilenemeyen kaynaklar:** Projenin tüm zamanı içerisinde tüketilen ve yerine konulamayan kaynaklardır. Para bu kaynak tipi için iyi bir örnektir. Enerji ve ham maddeler de yenilenemeyen kaynaklardandır.

- **Çift kısıtlı kaynaklar:** Projenin tüm zamanı içerisinde tükenmekle birlikte, belli bir zaman diliminde belli bir miktarda olan kaynaktır. Yenilenemeyen ve yenilenen kaynakların ikisinin birlikte olduğu durum gibidir. Sosyal güvenceler ile sınırlandırılan çalışma saatleri nedeniyle işgücü bu kaynaklar için verilebilecek en güzel örneklerden biridir.

- **Kısmî yenilenebilir kaynaklar:** En son tanımlanan kaynak tiplerinden biridir. (Böttcher, ve ark., 1996, Schirmer, ve ark., 1996, Drexl, 1997) Güncel hayatta en çok karşılaşılan kaynak tipi denilebilir. Proje süresi içerisinde belli bir zaman aralığında yenilenebilir bir kaynak olup, diğer süresi için diğer kaynak tiplerinden olan kaynaklardır. En önemli örnek, hafta içi çalışan işçilerin hafta sonu izinli olmaları ya da hafta sonu çalışanların daha az olmaları, maliyetlerinin farklı olduğu örneğidir.

- **Kesintisiz kaynaklar:** Projenin başlamasından itibaren ya da belli bir zaman aralığında kesintisiz kaynaklardır. Örnek olarak elektrik, gaz vb. enerji kaynakları verilebilir.

- **Atanmış kaynaklar:** Bir iş tarafından meşgul edilen kaynaklardır. (Bianco, ve ark., 1998) Bir makineye tek bir kişinin verildiği örneği gösterilebilir.

- **Diğer kaynaklar:** Birden fazla aktivite ya da işin kullandığı tek bir kaynak tipi için *uzay kaynaklar* denir. Bu kaynaklara depo, makine ve üretim tesisi içerisinde bulunan toplam alanlar örnek olarak verilebilir.

Proje çizelgelemede kaynaklar bir kısıt oluşturmaktadır. Bu kısıtlara göre özel alt konular oluşturulmuştur. Bir sonraki bölümde bu konu ele alınacaktır.

### 2.3. Kaynak Kısıtlı Proje Çizelgeleme Problemleri (RCPSP)

$n$  kadar işlenmesi gereken işlerden oluşmuş ve öncüllük ilişkileri kısıtlamaları olan, belli oranda kaynak isteyen bu işlerin sıralanması esasına dayanan problemlerdir. Örnek tanımlaması  $m, I | cpm | C_{max}$  şeklindedir. Np-zor problemlerdendir. (Blazewicz, ve ark., 1983)

Bu problemlerin en fazla kullanılan hedef fonksiyonu proje süresinin minimuma indirilmesidir. ( $C_{max}$ ) Buna bağlı olarak, maliyet, gecikme süreleri ve kârın en büyüklenmesi de hedef fonksiyonlardandır.

İlk olarak tek modlu problem incelenecektir. Problemin çok modlu versiyonu diğer bir başlıkta incelenecektir. Basit kaynak kısıtlı çizelgeleme problemlerinin matematiksel gösterimi aşağıdaki şekildedir: (Ulusoy, 1996)

$t$  : zaman ( $t=1...T$ ),

$j$  : İş,

$r$  : kaynak ( $R = \{ r_1, r_2, \dots, r_n \}$ ),

$d_j$  :  $j$  işinin süresi,

$P_j$  :  $j$  işinin öncülleri kümesi

$EFT_j$  :  $j$  işinin en erken bitiş zamanı

$LFT_j$  :  $j$  işinin en geç bitiş zamanı

$k_{jr}$  :  $j$  işinin  $r$  kaynağından kullanım miktarı

$K_r$  :  $r$  kaynağının kapasitesi

olmak üzere;

#### Değişkenler:

$$X_{jt} = \begin{cases} 1, & \text{eğer } j \text{ işi } t \text{ süresi sonunda bitiyorsa} \\ 0, & \text{diğer} \end{cases}$$

Bu  $j$  işi  $t$  zamanında yapılmış ise bir daha işleme alınmayacaktır.

**Amaç Fonksiyonu**

$$\min Z = \sum_{t=EFTj}^{LFTj} tX_{jt}$$

Hedef fonksiyonu minimum tamamlanma süresi.

**Kısıtlar:**

$$\sum_{t=EFTj}^{LFTj} X_{jt} = 1$$

$$j = 1, \dots, J$$

Her iş mutlaka sıralanmak zorundadır.

$$\sum_{t=EFTi}^{LFTi} tX_{jt} \leq \sum_{t=EFTj}^{LFTj} (t - d_j)X_{jt}$$

$$J=2, \dots, J, i \in P_j$$

$J$  işinin öncüllerinden sonra yapılmasını sağlar.

$$\sum_{j=1}^J k_{jr} \sum_{\tau=t}^{t+d_j-1} X_{jt} \leq K_r$$

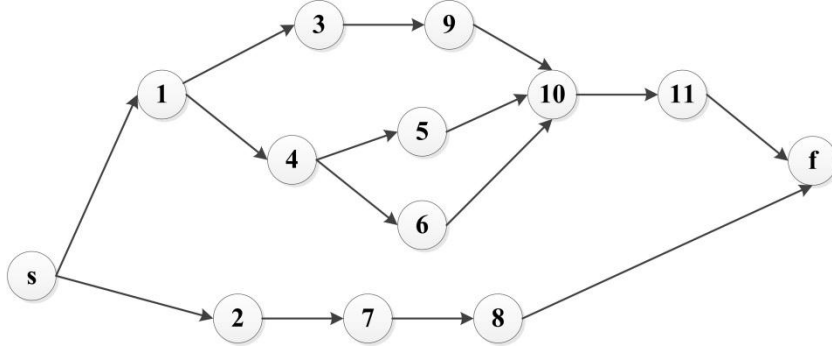
$$r \in R, t = 1, \dots, T$$

Birim zamanda en fazla kullanılacak kaynak miktarını göstermektedir.

Konunun daha iyi anlaşılması için örnek bir problem verileri Çizelge 2.11'de verilmiştir. Bu verilere ek olarak problemde 3 kaynak bulunduğu ve her kaynağın kapasitesinin de 6 olduğu verilmiştir. Bu nedenle herhangi bir kaynak 6 brimden fazla kullanılamayacaktır. Öncüllük ilişkilerinin daha kolay anlaşılmasını sağlayan şebeke modeli AON(İşlemler düğümler/nodlar üzerinde) gösterimi ile Şekil 2.5'te gösterilmiştir.

Çizelge 2.11. Örnek RCPSP problemi

İş No	Süresi (gün)	1 Numaralı Kaynak Tüketimi	2 Numaralı Kaynak Tüketimi	3 Numaralı Kaynak Tüketimi	İşin Öncülleri
1	3	3	2	1	-
2	5	2	4	2	-
3	6	3	1	2	1
4	2	4	3	1	1
5	3	2	0	3	4
6	3	1	1	1	4
7	4	3	1	1	2
8	5	2	2	2	7
9	4	3	2	3	3
10	2	4	1	0	5,9
11	3	5	4	2	10



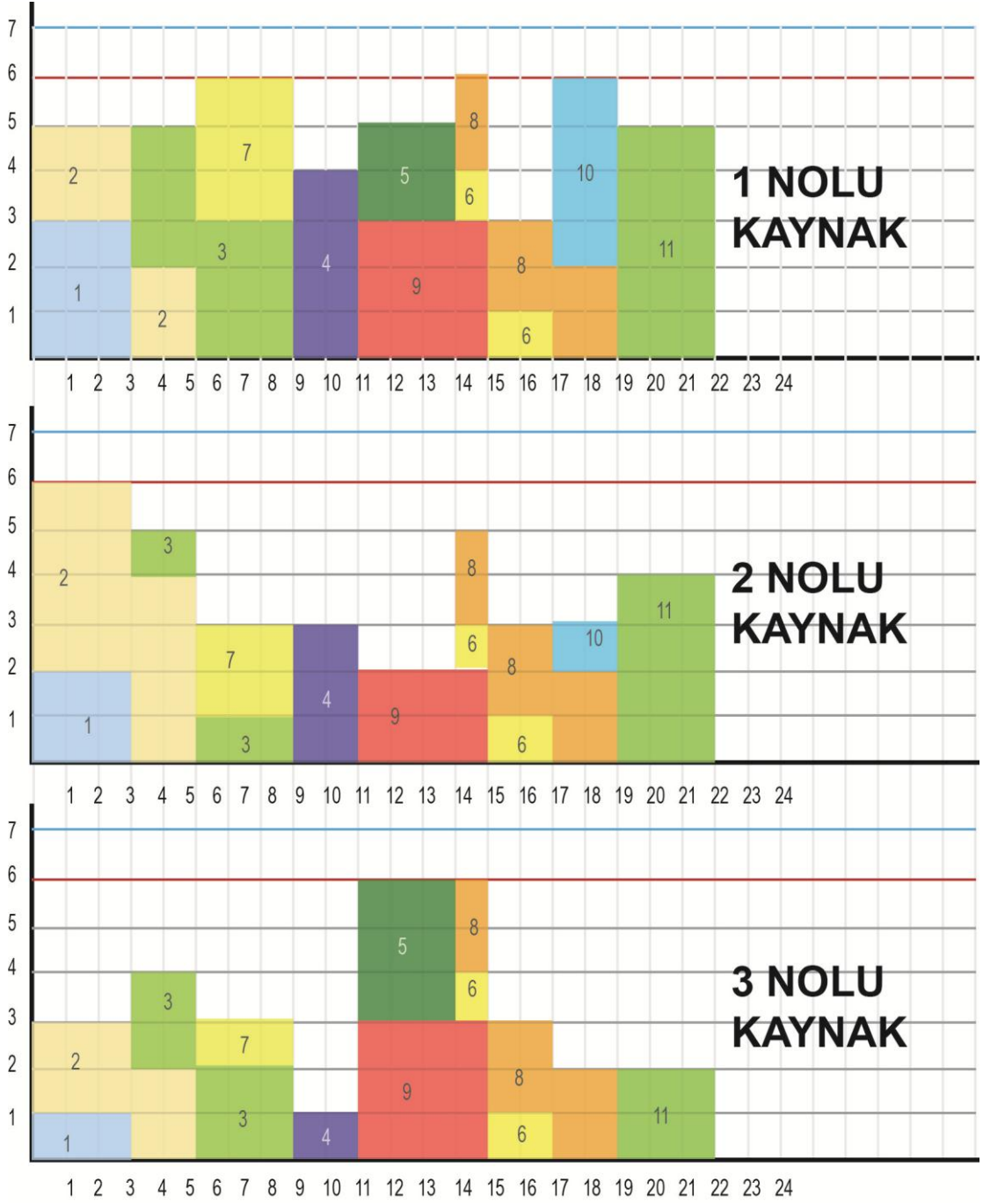
Şekil 2.5. Örnek problemin görsel olarak öncüllerinin gösterimi

Projede görüldüğü üzere, başlama  $s$  ve bitiş  $f$  işleri bulunmaktadır. Buna göre işlerin önceliği bozulmadan örnek bir sıralama yapılırsa, şu şekilde bir örnek verilebilir: 1-2-3-4-7-9-5-6-8-10-11

Şekil 2.5 üzerinden de anlaşılacağı üzere, 10. İşin yapılabilmesi için 9, 5 ve 6'nın bitmiş olması gerekmektedir.

Problemin çözümünün daha kolay anlaşılması için, Şekil 2.6 'da ise işlerin kaynaklar üzerine dağıtıldıktan sonraki hali gantt şeması üzerinde gösterilmiştir. Üzerinde rakam bulunan kutucuklar iş numarasını temsil etmektedir. Maksimum kaynak kullanımı 6 ile sınırlandırılmış 3 kaynağa, proje çizelgelemesi yapılmıştır.

Burada gösterilen üç kaynak, aynı anda tüketilen ve kullanılan kaynaklardır. Örneğin; insan gücü, hammadde ve maliyet olarak düşünülebilir.



Şekil 2.6. Örnek RCPSP problemin kaynaklar üzerinde gösterimi



## 2.4. Paralel makine problemleri

İşgücü yenilenebilir kaynak olan makine tiplerine de benzetilebilir. Bu noktada literatürdeki konuyla ilgili makine problemlerine de temas etmek gerekmektedir. Gerçek hayatta işlerin paralel makine ya da kaynaklarca işletildiği görülmektedir. Bu işletmelerde tamamlanma süresini minimize etmeye çalışırken, makineler arası iyi bir denge kurulması hedeflenmektedir.

İşlerin tamamlanma süresinin minimizasyonuna dayalı örnek bir problem üzerinde paralel makine dağılımı aşağıdaki örnek problem üzerinde gösterilmiştir.

Çizelge 2.12. Örnek paralel makine problemi

İş	1	2	3	4	5
Süre	2	3	1	4	2

İki paralel makine olsun, buna göre toplam tamamlanma süresinin makine sayısına göre aritmetik ortalaması alınır ve hedeflenen dağılım bu rakama göre oluşturulmaya çalışılır. Burada makine başına düşmesi gereken iş süresi:

$M^*$  : Ortalama makine başına düşmesi gereken ideal yük / iş

$P_j$  : j işinin süresi

$m$ : makine sayısı

$n$ : iş sayısı olmak üzere,

$$M^* = \frac{1}{m} \sum_{j=1}^n p_j \Rightarrow M^* = \frac{1}{2} * (2+3+1+4+2) = 6$$

Buna göre (öncüllük ilişkileri yoksa) rastgele alınan işler konularak makineler dengelenerek en kısa süre de elde edilmeye çalışılır.

1.Makine = 1 -2 -3 işler => 6 birim süre

2.Makine = 4-5 işler => 6 birim süre

Paralel makinelerde makine tipleri her zaman aynı değildir ve özdeş, özdeş olmayan makineler olarak ikiye ayrılırlar.

### Özdeş makine tiplerine göre sıralama:

Özdeş makineler, aynı işlerin tamamlanma sürelerinin aynı olduğu makinelerdir. Bu sebeple Çizelge 2.12’de verilen probleme tam olarak uymaktadır. Tamamlanma süresinin minimize edildiği hedef fonksiyonu aşağıda gösterilmiştir. (Hongyi., ve ark., 2003)

$i = 1, \dots, n$  makine sayısı

$j = 1, \dots, m$  iş sayısı

$t_j = j$  işinin süresi

$$x_{ij} = \begin{cases} 1, & j \text{ işi } i \text{ makinesinde işlenmiş ise} \\ 0, & \text{diğer durum} \end{cases}$$

$$C_{max} = \max_{i=1}^m \left\{ \sum_{j=1}^n x_{ij} * t_j \right\}$$

### Özdeş olmayan makine tiplerine göre sıralama:

Özdeş olmayan makineleri sıralamak daha zordur. Çünkü her iş makine tipine göre farklı işlem süresine sahiptir. Hızı, kapasitesi gibi özellikleri özdeş olmadığı için işlerin işlem süresi üzerinde büyük etkileri vardır. Bir önceki özdeş olan makine tipi için kullanılan matematiksel modelden farklı olarak, işlendiği makineye bağlı olan süre dikkate alınmıştır. Bir önceki matematiksel model ile aynı hedef fonksiyonuna sahip model aşağıdaki gibidir. (Balin, ve ark., 2008)

$i = 1, \dots, n$  makine sayısı

$j = 1, \dots, m$  iş sayısı

$t_{ij} = j$  işinin  $i$  makinesinde yapılma süresi

$$x_{ij} = \begin{cases} 1, & j \text{ işi } i \text{ makinesinde işlenmiş ise} \\ 0, & \text{diğer durum} \end{cases}$$

$$C_{max} = \max_{i=1}^m \left\{ \sum_{j=1}^n x_{ij} * t_{ij} \right\}$$

Çizelge 2.13'te daha kolay anlaşılması için bir örnek verilmiştir.

Çizelge 2.13. Özdeş olmayan makine tiplerine örnek problem

İş	1	2	3	4	5
Süre 1.Makine	2	3	1	4	2
Süre 2.Makine	1	4	2	2	1

Rastgele veya hangi makinede işlem süresi daha kısa ise önceliğin o olduğu sıralamaya göre atama yapılırsa;

1.makine = 1 – 2 işleri, => (2 + 3) 5 birim

2.makine = 3 -4-5 işleri => (2 + 2 + 1) 5 birim sürmüştür.

## 2.5. Çok Modlu Kaynak Kısıtlı Proje Çizelgeleme Problemleri (MRCPSP)

Bu problem tipi, gerçek hayatı daha iyi temsil edebilmek için RCPSP problemlerinin geliştirilmiş halidir. Aynı işlemin farklı kaynak tipleri ile yapılabildiği projeler için çözümler üretir. Mevcut işin hangi kaynak tarafından yapıldığı çok önemlidir. Farklı kaynak tipleri ile işlenen işlerin süreleri de, maliyetleri de farklı olmaktadır. Bu farklı kaynaklar birer mod olarak tanımlanırlar. Örneğin farklı uzmanlıktaki kişiler ve iş yapma becerileri farklıdır. RCPSP problem tanımından farklı olarak mod parametresi dâhil edilmiştir.

### Değişkenler:

$$X_{jmt} = \begin{cases} 1, & \text{eğer } m \text{ modunda yapılan } j \text{ işi,} \\ & t \text{ süresi sonunda bitiyorsa} \\ 0, & \text{diğer} \end{cases}$$

Bu j işi t zamanında m modunda yapılmış ise bir daha işleme alınmayacaktır.

### Amaç Fonksiyonu

$$\min Z = \sum_{t=EFT_j}^{LFT_j} tX_{jmt}$$

Hedef fonksiyonu minimum tamamlanma süresi.

m mod tanımı  
Her iş mutlaka sıralanmak zorundadır.

**Kısıtlar:**

$$\sum_{m=1}^{M_j} \sum_{t=EFT_i}^{LFT_i} t X_{jmt} \leq \sum_{m=1}^{M_j} \sum_{t=EFT_j}^{LFT_j} (t - d_{jm}) X_{jmt}$$

$$\sum_{j=1}^J k_{jr} \sum_{m=1}^{M_j} k_{jmr} \sum_{\tau=t}^{t+d_{jm}-1} X_{j\tau} \leq K_r$$

$$\sum_{j=1}^J k_{jr} \sum_{m=1}^{M_j} k_{jmr} = \sum_{t=EFT_j}^{LFT_j} X_{jmt} \leq K_r$$

$$J = 2, \dots, J, i \in P_j$$

$J$  işinin öncüllerinden sonra yapılmasını sağlar.

$$r \in R, t = 1, \dots, T$$

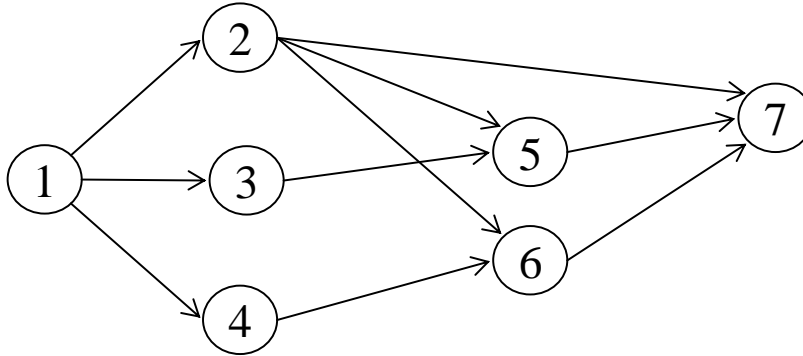
Birim zamanda en fazla kullanılacak kaynak miktarını göstermektedir.

$$r \in N$$

$N$  yenilenemez kaynak kümesi

Yenilenemez kaynakları göstermektedir.

Kaynak modlarının sayısı arttıkça hesap yükü de yüksek oranda artmaktadır. Bu problemi de yine bir örnek üzerinde gösterilirse;



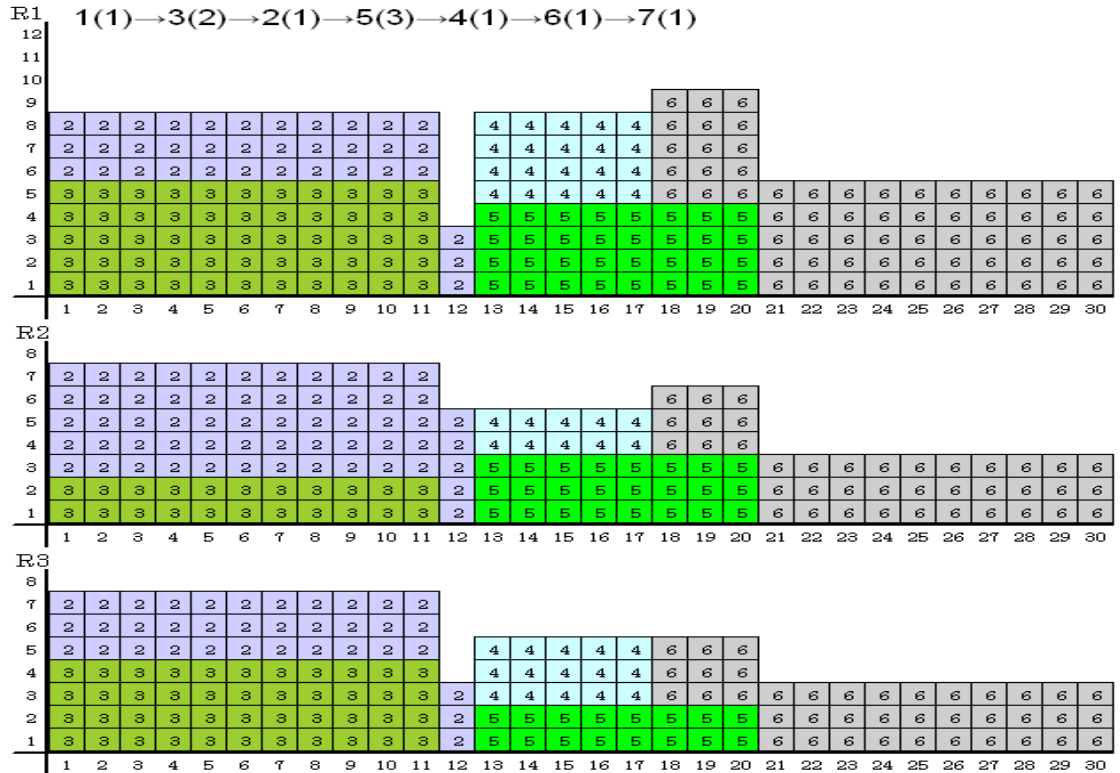
Şekil 2.7. Çok modlu kaynak kısıtlı proje çizelgeleme problemi örnek gösterimi(Lab, 2004)

Kaynakların limitleri sırası ile 12 – 8 – 8 olmak üzere Çizelge 2.14’te örnek bir MRCPSPP problemi gösterilmiştir. Şekil 2.8’de ise, işlerin kaynaklarca dağıtıldıktan sonraki durumu gantt şeması ile gösterilmektedir. 3(2) notasyonu, 3 no’lu işin 2 no’lu modda yapıldığını göstermektedir. Buna göre Çizelge 2.14’te 3 no’lu işin 2 no’lu modda 11 birim süreceği ve sırasıyla 5,2,4 birim kaynak tüketeceği anlaşılmaktadır.

Çizelge 2.14. Örnek MRCPSP problemi

İş	Mod	Süre	Yenilenemeyen		Yenilenebilir Kaynak 3	Öncülleri
			Kaynak 1	Kaynak 2		
1	1	Başlangıç				
2	1	12	3	5	3	1
	2	15	4	4	2	
	3	18	3	3	4	
3	1	5	2	5	3	1
	2	11	5	2	4	
	3	13	4	3	2	
4	1	5	4	2	3	1
	2	14	5	4	2	
5	1	15	2	3	4	2,3
	2	12	5	2	3	
	3	8	4	3	2	
6	1	13	5	3	3	2,4
	2	12	6	4	2	
	3	15	2	3	3	
7	1	Bitiş				2,5,6

Örnek, 1 – 3 – 2 – 5 – 4 – 6 – 7 iş sıralamasına göre görsel olarak çizimi Şekil 2.8’de gösterilmektedir.



Şekil 2.8. MRCPSP örnek problemin gantt şeması ile gösterimi, (Lab, 2004)

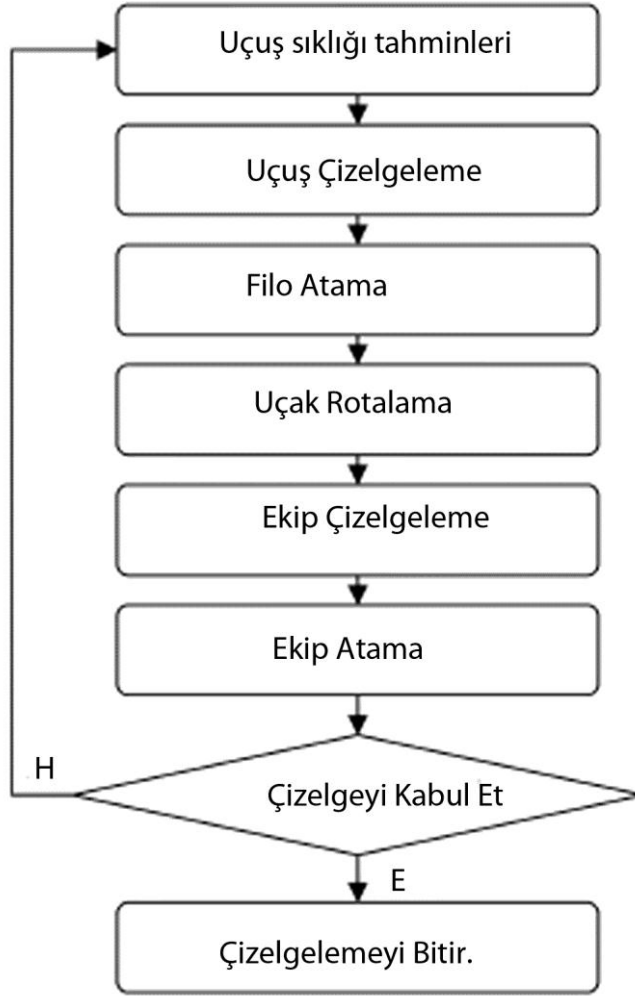
Parantez içindeki bilgiler, hangi modda işlendiğini göstermektedir. Örneğin 5 no'lu iş 3. modda işlenmiştir. Eğer 2.modda işlenmiş olsa idi 12 birim sürede işlenebilecek ve örnekte gösterildiği üzere her kaynak için tüketimi farklı olacaktı.

## **2.6. Ekip Oluşturma ve Atama**

Ekip oluşturma ve atama konusu özellikle hava yolu şirketleri için büyük önem taşıyan problemlerdendir. Her iş birer uçuşu temsil etmektedir. Her uçuş belli bir zaman almakta, gideceği yere göre farklı ekipler gerektirmektedir. Örneğin, turistik bir yer var ise, gezi rehberi olmak zorundadır. Bir ekip birden fazla uçuşu gerçekleştirebilmektedir. Bu sebeple yapılan her hesap için, uçuş çizelgesi, filo atama, uçak rotalama, ekip atama gibi problemlerin üstesinden gelmek gerekmektedir. Her uçuş için bu ekibe bağlı maliyetler oluşmaktadır. Tüm bunlar göz önüne alınarak, maliyetlerin en aza indirgenmesi hedeflenmektedir.

Bunlara ek olarak, yasaların getirdiği zorunluluklara da uyulmak zorundadır. Bu yasalar genel olarak, belli bir saatin üzerinde daha fazla uçulamayacağı, dinlenmeleri için belli sürelerin verilmesi gerektiği, belli uçuşları birden fazla kişi ile yapma zorunluğu gibi yükümlülükler de problemin parametrelerini artırmaktadır.

Bu çalışmada da basit anlamda bir ekip atama yapılması gerekmektedir. Bu nedenle Havayolu şirketlerinin kullandığı yaklaşımda, uçak ekibi atama problemlerine önerilen çözümlerin kullanılması düşünülerek işçi ataması ve iş çizelgesi yapılması hedeflenmiştir. Ancak uçak ekibi atama problemlerinin gerçekleştirilebilmesi için, şekil 2.9'da da gösterildiği üzere uçuş seferlerinin hesaplanmış olması gerekmektedir. Seferlerin oluşturulması bu çalışmada işlerin sıralanması anlamına gelmektedir. İşin sıralaması önce yapılması gerektiği için, her işe verilecek kaynak sınırı da önceden belirlenmesi gerekmektedir. Aksi takdirde olurlu olacak sonuç sayısı çok azalacaktır. Bu yaklaşım ile programın dinamikliği de kısıtlanmaktadır. Çünkü kaynak sayısı sabit kabul edilmesi zorunluluğu, esnek üretimde hedeflenen esnek işçi kullanımını sınırlamaktadır. Bununla birlikte ekibin işin süresi üzerinde bir etkisi bulunmamaktadır. Bu nedenlerden dolayı bu çalışma ekip atama problemlerine uymamaktadır.



Şekil 2.9. Havayolu problemleri çizelgeleme iş akışı (Lucic, ve ark., 2007)

Ekip atama problemleri ile temelde aynı parametrelere sahip hemşire atama problemleri de literatürde büyük yer tutmaktadır. Her hemşire uzmanlık alanlarına göre atanmalıdırlar. Hava yolu işletmelerinden farklı olarak buradaki çalışanlar 24 saat üzerinden haftanın 7 günü hesaplanarak çizelgelenmelidirler. Yine yasal zorunluluklar sebebiyle, tatil ve dinlenme hakları da göz önüne alınmalıdır. 24 saat üzerinden hesaplar yürütüldüğü için vardiya değişimleri yapılmak zorundadır. Kısıtlar kesinlikle olması gerekenler ile olması duruma bağlı olanlar olarak ikiye ayrılırlar. En sık karşılaşılan kısıtlar aşağıda listelenmiştir. (Cheang, ve ark., 2003)

- Hemşire iş yükü (minimum / maksimum)
- Birbirini izleyen aynı işlerin vardiyaları (minimum / maksimum / tam rakam)
- Birbirini izleyen iş/vardiyalar (minimum / maksimum / tam rakam)
- Sınıfına ve seviyesine göre hemşire uzmanlık alanları
- Hemşire tercihleri veya ihtiyaçları
- Boş/izinli günleri (minimum / maksimum / peş peşe boş günler)
- Vardiyalar arası boş zaman (minimum)
- Vardiya türleri atamaları (maksimum vardiya tipi, her vardiya için ihtiyaçlar)
- İzinli ve tatil günleri
- Çalışılan hafta sonları
- Beraber çalışma durumları bulunan ve bulunmayan hemşireler
- Vardiya kısıtları, vardiyalar aynı zamanda bir insana atanamaz.
- Her vardiya için gerekli personel ihtiyacı (minimum / maksimum / tam rakam.)

Bu liste duruma göre daha da uzatılabilmektedir. Problemin çözümü için çeşitli matematiksel modeller sunulmuş olup, problemin tarihler üzerinde atanabilir ya da atanamaz tanımları içermesi sebebiyle tamsayı programlamaya daha yatkındır. (Trilling, ve ark., 2006) Diğer çözüm yöntemleri problemin Np-zor olması sebebiyle sezgisel algoritmaların kullanılmasıyla yapılmaktadır. Hedef fonksiyonu daha çok, sağlık hizmetinin doğru karşılanması veya maliyetlerdir.

Bu çalışmanın ekip oluşturma aşamasına hemşire çizelgeleme problemleri de uymamaktadır. Hemşire atama problemleri için literatürde işin doğası gereği bireysel görev atama yaparak çözümler oluşturulmuştur. İşlerin önceden oluşturulması gerektiği sebebiyle ve yine kişi atamasının işin süresi ile bir ilgisi bulunmaması nedeniyle bu problemler de bu çalışmaya uymamaktadır. Bu çalışmada bahsedilen çalışma saatleri ve vardiya gibi kısıtlar göz ardı edilmiştir.



## 2.7. Servis/Hizmet Merkezleri

Gelişmiş ülke olmanın en önemli göstergelerinden biri, hizmet sektörünün ülkenin gayri safi milli hasılasına olan katkı oranının yüksek olmasıdır. Üretim teknikleri ve artan hizmet ihtiyacı sebebiyle bu sektör çok hızlı ilerlemektedir. Bu çalışmaya en uygun örnekler, çağrı ve servis merkezleri problemlerdir. Buradaki yapılan işler beceriye göre atanır ve bazı çalışanlar birden çok beceriye sahip olabilmektedirler. Çağrı merkezleri problemlerinin, havayolu şirketleri ekip atama problemlerinden en önemli farkı ise talep durumunun bilinmemesidir. Ayrıca mesafelerin karıştığı bir problem olmaması yönüyle ekip atama problemlerinden bu yönüyle daha kolaydır. (Ernst, ve ark., 2004)

Çalışmada talepler belirli olduğu için, servis merkezlerinde karşılaşılan işler deterministik olarak kabul edilmiştir. Bu problemlere iyi bir örnek olarak, yapılan bu çalışmaya en yakın nitelikte olan ve son yıllarda yapılan bir çalışma verilecektir. (Valls, ve ark., 2009) Kişiler farklı yeteneklere sahip, heterojen bir karmaya sahiptir.

### Görevler:

$j : 1, \dots, n$  (işlerin sayısı, 1 ve  $n$  başlama ve bitişi gösterir.) İşler bölünemez.

$ms_j$ :  $j$  işi için maksimum başlama zamanı.

$mf_j$ :  $j$  işi için maksimum bitiş zamanı.

$sc_j - fc_j$  = başlama – bitiş kaynaklı bu zamanları aşma durumunda oluşan maliyet.

$d_j$  : işlem süresi, tek bir kişi ile sadece çalışma saatlerindeki izinler sebebiyle bölünebilir. (öğle yemeği vb.)

$c_j$  : önem derecesi. Ne kadar büyük ise o kadar önemlidir. İşin maksimum başlama ve bitiş zamanlarına bağlı hesaplanır.

$t$  : Süre, maliyet, önem derecesi ve bitiş süreleri negatif olmayan tam sayılardır.

### İşgücü:

$W = \{ W_1, W_2, \dots, W_{nw} \}$  iş gücü  $nw$  kadar kişiden oluşur.

$A = \{ A_1, \dots, A_{na} \}$  uzmanlık alanları,  $na$  kadar beceriden oluşur.  $A$  tanımı o işin yapılabilmesi için gerekli beceriyi gösterir.  $A_j$   $j$  işinin yapılabilmesi için gerekli uzmanlığı gösterir.

$F_j$ , işin bitiş süresini gösterir.

$W(Ah)$  : bu beceriye sahip işgücünü gösterir.

$W(j)$  :  $j$  işini yapabilmesini gösterir.

$A(wk)$ ,  $ekh$  ile tanımlanan, bu işi bu kişinin ne kadar iyi yapabileceğinin tanımıdır.  $ekh$  bir tamsayı olup, 1 den büyük ve eşittir.  $ekh$  ne kadar düşük ise o kadar uygun demektir.  $ekh = 1$  (uzman) ,  $ekh=2$  (standart) ve  $ekh = 3$  (çırak) olmak üzere üç tanımı yapılmıştır. Kıdemli ya da çırağın o işe atanması o işin tamamlanma süresini etkilemektedir.

$RP$  = azaltma oranı

$IP$  = artırma oranı

$d_{jk} = j$  işinin  $wk$  işçisince yapılmış süresini temsil etmektedir. Eğer standart bir işçi ise  $d_{jk} = d_j$  dir.

$av_{kt} = 0, \dots, PDUR$  ( $PDUR$  çizelgesi yapılacak zaman dilimi.)

Her işçinin kendi zaman çizelgesi olup,  $av_{kt}$  ile temsil edilmektedir.  $w_k$  işçisinin  $t$  zamanında çalışabilir olup olmadığını gösterir. 1 ise çalışabilir, 0 ise çalışamaz.

Çalışmaya başlamış bir kişinin işi sadece çalışma çizelgesinde bir mola var ise bölünebilir ve kesinlikle hemen mola sonrası devam edilir. Bu durumda işin bitiş süresi şu şekilde hesaplanır.

$$f_j(s_j, w_k) = \begin{cases} \min \left\{ t * \mid \sum_{t=s_j}^{t*-1} av_{kt} = d_{jk} \right\} & \text{eğer } t * \text{ var ise} \\ +\infty & \text{Yok ise} \end{cases}$$

Doğal olarak şu eşitlik oluşur;  $f_j(s_j, w_k) \geq s_j + d_{jk}$  , yani işin süresi mola süresine eklenmiştir.

### Öncüllük İlişkisi:

Öncüllük arası gecikmeler, 0, negatif veya pozitif tamsayı olabilirler. Öncüllük ilişkileri şu şekilde tanımlanarak hesaplanmaktadır.  $\delta_{i,j} = (\tilde{O}I, \alpha, \beta)$  .  $i$  ve  $j$  işleri arasındaki ilişkinin tanımında,  $\tilde{O}I$  öncüllük ilişkisini göstermektedir.  $\alpha$  parametresi gecikme süresini,  $\beta$  parametresi de  $\alpha$  parametresinin nasıl tanımlanacağını gösterir. Örneğin “2” olması durumu yüzde olarak hesaplanacağını gösterir.  $\delta_{2,3} = (SS, 5, 2)$  . Bunun anlamı, “3” numaralı iş, “2” numaralı işin başlamasından %5 zaman sonra başlayabilir. İlk ve

son işlerin öncüllük ilişkilerindeki parametreler 0 olacaktır. Bu tanım kısaca bu çalışmada  $SS_{ij}$  olarak gösterilmiştir.  $\beta$  sıfır ise  $SS_{ij} = \alpha$  olacaktır. Diğer durum hesaplanarak bulunacaktır.

$$SS_{ij} = \begin{cases} \min \left\{ t * \mid \sum_{t=s_i}^{t*-1} av_{kt} = \left\lceil \frac{\alpha \cdot d_{ik}}{100} \right\rceil \right\} & \text{eğer } t * \text{ var ise} \\ +\infty & \text{Yok ise} \end{cases}$$

Aynı şekilde  $SF_{ij}$ ,  $FS_{ij}$  ve  $FF_{ij}$  durumları da hesaplanır.

### Matematiksel Modeli:

$S = (s, w)$  ;  $S$  çözüm kümesi bu şekilde tanımlanmaktadır.  $s$  negatif olmayan tamsayı olup her işin başlama süresini göstermektedir.  $w$  de, bu işi yapacak olan her işçiyi tanımlamaktadır.

### Amaç Fonksiyon:

Sırasıyla birden çok hedef fonksiyonu vardır. Optimum sonuç leksokografal (sırasıyla hedef fonksiyonlar denenir. Eğer ilki aynı ise, ikinci denkleme göre karşılaştırma yapılarak en iyi sonuç sıralaması yapılır.) sırayla belirlenir.  $Z_1 \rightarrow Z_2 \rightarrow Z_3$  ,  $\min_{Lex}(Z)$  olarak tanımlanır.

1-Önem derecesine göre;

$$Z_1(S) = Kritik(S) = \sum_{j=1}^n c_j s_j$$

2-Her işe en iyi çalışanı atama;

$$Z_2(S) = AD(S) = \sum_{i=1}^n e_{w(i)A(i)}$$

3-İyi dağılmış iş gücü;

$$Z_3(S) = \text{İşYükü}(S) = \sum_{k=1}^{ns} \frac{|yük(k) - ortalama\_yük|}{ortalama\_yük}$$

$$yük(k) = \frac{\sum_{i \in J | w(i)=k} dik}{\sum_{t=0}^{f^{n-1}} avkt}$$

$$ortalama\_yük = \frac{1}{ns} \sum_{k=1}^{ns} yük(k)$$

### Kısıtlar:

$w(i) \in W(i)$	$\forall i \in V,$	Her $i$ işi sadece tek bir $w(i)$ çalışanına verilmiştir.
$s_j - s_i \geq SS_{ij}$	$\forall (i, j) \in Ess$	Öncüllük ilişki kısıtlarının sağlanması.
$f_j - s_i \geq SF_{ij}$	$\forall (i, j) \in Esf$	Öncüllük ilişki kısıtlarının sağlanması.
$s_j - f_i \geq FS_{ij}$	$\forall (i, j) \in Efs$	Öncüllük ilişki kısıtlarının sağlanması.
$f_j - f_i \geq FF_{ij}$	$\forall (i, j) \in Eff$	Öncüllük ilişki kısıtlarının sağlanması.
$s_i \leq ms_i$	$\forall i \in V$	Maksimum zaman kısıtları
$f_i \leq mf_i$	$\forall i \in V,$	Maksimum zaman kısıtları
$ \{i \in V \mid w(i) = wk \text{ ve } si \leq t \leq fi \leq 1 \forall wk \in W, 0 \leq t < PDUR, \}$		Bir işçi birden fazla işi bir anda yapamaz.
$av_{w(i)si} = 1$	$\forall i \in V$	Çizelgede çalışabileceği zaman
$0 \leq si \leq PDUR$	$\forall i \in V$	Hedeflenen zaman sınırı içinde yapılması
$s_1 = 0,$		Hedeflenen zaman sınırı içinde yapılması
$fi = fi(si, w(i)) \forall i \in V$ parametreleri ile tanımlıdır.		

Yukarıda matematiksel modeli verilen Valls, ve arkadaşlarının (2009) yaptığı çalışma ile bu çalışma arasında bazı farklılıklar bulunmaktadır. Valls, ve arkadaşlarının yaptığı çalışmada işler bölünemez tiptedir, hedef fonksiyonu işlerin en kısa sürede tamamlanmasıdır, bir işe birden fazla kişi heterojen olarak atanmaktadır ve işçilerin uyacakları bir zaman çizelgesi bulunmamaktadır.

## 2.8. Önceki Çalışmalar

Bu bölümde kuramsal bilgilerde genişçe anlatımı yapılan konuların, bu çalışma ile ortak yanları bulunan araştırmalara atıflar yapılacaktır.

Literatürde yetenekli işgücü, işletme ve iktisat bilimleri perspektifinden bakıldığında, maliyet, planlama, proje çizelgeleme ve temel olarak tahmini istatistikler üzerinden organizasyon, motivasyon ve insan psikolojisi üzerine yoğunlaşmış durumdadır. Bu alanlarda yapılan araştırmalarda esnek işgücünün öneminin giderek arttığını ve rekabetçi ortamda yeniliğe açık vasıflı işçilerden oluşması gerektiği gerçeği anlatılmaktadır (Selamoğlu, 1998, Ansel). Bu çalışmanın kapsamında ise işgücü optimizasyonu hedefli, özellikle yöneylem araştırmaları ve proje çizelgeleme konularını içeren, endüstri mühendisliği alanlarındaki araştırmalar bulunmaktadır.

Ernst ve arkadaşlarının (2004) yaptığı çalışma, personel çizelgeleme ve görev listelemesi üzerine 700'ü aşkın makalenin gruplandırıldığı bir kaynakça araştırmasıdır. Buradan bakıldığında literatürde 1950'den bu yana giderek önemi artan personel çizelgeleme problemlerinde yeteneğin göz önüne alındığı çalışma dalları görev atama ve talebe yönelik personel çizelgeleme problemlerinde görülmektedir. Talep tabanlı ve ekip atama problemleri, havayolu ekip çizelgelemeden, otel ve turizm, hemşire atamaya kadar kullanıldığı görülmektedir. Bu çalışmalarının daha sonra revizyonu da yayınlanmıştır.

Blazewicz ve arkadaşlarının (1983) gösterdiği üzere RCPSPP problemleri Np-zor problemlerdendir. Kolish ve Hartman'ın (1999) RCPSPP üzerine oldukça fazla çalışmaları olup, çalışmalarının birinde bu problemler için sezgisel yaklaşımları anlatmakta ve sınıflandırmaktadır. Daha sonraki bir çalışma ile Hartman ve Briskorn (2010) RCPSPP problemlerinin uzantılarını, nerelerde kullanıldıklarını, hedeflerini ve çeşitliliklerini göstermişlerdir.

Yang ve arkadaşları (2001) yaptıkları çalışmada MRCPSPP problemlerinin makine çizelgeleme problemleri olarak tanımlandığını ve makine-iş çizelgeleme karakteristiği

olarak da; paralel özdeş makineler, hızları farklı paralel makineler, özdeş olmayan paralel makineler, akış ve iş atölyesi tipine uygun çizelgelemeler olarak tanımlamışlardır. (Yang, ve ark., 2001)

RCPSP problemlerinin çözümleri Np-Zor problemler olduğu için MRCPSP problemleri de Np-Zor problemler olup, çok sayıda sezgisel algoritma ile çözüm teknikleri önerilmiştir. (Reyck, ve ark., 1999) Eğer birden fazla yenilenmeyen kaynak içerirse bu problemler Np-Tam problemler sınıfına girmektedir. (Peteghem, ve ark., 2009) Tabu arama, tavlama benzetimi (Jozefowska, ve ark., 2001) ile çözümler sunulmuş olmakla beraber literatürün büyük çoğunluğuna genetik algoritma (Mori, ve ark., 1996, Peteghem, ve ark., 2009, Barrios, ve ark., 2009) ile ilgili çözüm teknikleri sahiptir. Vincent ve arkadaşının yaptığı çalışmada son yıllarda öne çıkan MRCPSP problemleri için sunulan sezgisel algoritmaların listesini EK-5'te verilmiştir. (Peteghem, ve ark., 2009) Agustin ve arkadaşları ise iki aşamalı genetik algoritma metodunu kullanarak genetik algoritma uygulamasını geliştirmişlerdir (Barrios, ve ark., 2009).

Vikram ve arkadaşlarının yaptığı çalışmada özdeş olmayan (heterojen) kaynakların çizelgelenmesinde zaman ve kaliteye dikkat edilmesi konusuna değinilmiştir. İşler arası farkların olduğu ve uzmanlık gerektiren durumlar için, karşılıklı eğitimlerin verilmesi ile iş esnekliği oluşturulmuştur. MRCPSP problemlerinin biraz modifikasyonu ile çözüm önermişlerdir. Buna göre MRCPSP problemlerindeki modlara göre bir iş yeniden işlemeyi gerektirebilir. Böylece kalite sağlanırken zaman konusunda negatif bir durum oluşmaktadır. Örneğin her iş için iki mod olduğu öngörülün. Eğer mod "1" ile o iş yapılır ise yeniden çalışma gerektirsin, aksi takdirde yeniden çalışma gerektirmesin. Çalışma sonunda küçük çaplı işler için çok da iyi bir iyileştirme gerçekleştirmediği gösterilmekle beraber, karşılıklı eğitimin göz önüne alınması gerektiği gösterilmiş ve kalite hedefleri tutturulabilmiştir (Tiwari, ve ark., 2009).

John W. Fowler ve arkadaşlarının (2007) yaptığı araştırmada ise, karşılıklı eğitimlerin alınması ile farklı işgüçleri oluşturulmuştur. Maliyet ve zaman hedeflerinin karşılanması hedeflenmiş ve karışık tamsayılı programlama tavsiye edilmiştir. Birkaç sezgisel

yaklaşım ile kabul edilebilir bir zaman içinde kabul edilebilir sonuçlar üretmeyi sağlamışlardır.

Literatürde beceriye dayalı işçi atama problemleri en çok servis ve çağrı merkezi konularında görülmektedir. Bunun nedeni, işler arası bilgi ihtiyacının ve işin içeriğinin benzerliğinin çok olmasıdır. Bu çalışmalara ait çözüm yöntemlerine Lerzan Örmeci ve arkadaşlarının (2008) yaptığı projede değinilmişlerdir. Ekip atama problemi olarak düşünüldüğünde havayolu ekip atama problemleri ile aynı görülebilir. Ancak çağrı/servis merkezleri ile havayolu ekip atama problemlerinin en temel ayrımı, ilk olanda talep belirsiz ve esnek iken, ikinci olanda talep belirlidir. (Ernst, ve ark., 2004)

Vicente ve arkadaşlarının (2009) yaptığı çalışmada, SWPSP tanımı getirilmiştir. Uzman/becerikli/yetenekli iş gücü çizelgeleme problemi olarak tanımlanabilir. Yaptıkları çalışma ile üç farklı beceri seviyesine sahip işçiler, işin süresi üzerine etkisini incelemişlerdir. Problemin MRCPSP problem olarak ta ele alınabileceğini bahsetmişler ve çözüm için sezgisel algoritma kullanmışlardır. Hibrit genetik algoritma ile ulaştıkları çözümler, ilk üretilen çözümlerden daha iyi sonuçlar üretmiştir. Literatürde ne kadar iyileştirme yaptıkları, karşılaştırma yapabilecekleri bir problem olmaması sebebiyle gösterilememiştir. Ancak işçi çizelgeleme problemlerine yeni bir yaklaşım getirmişlerdir. Bu konuya bir önceki başlıkta daha ayrıntısı ile değinilmiştir.

Esnek işgücünün olabilmesi için işçilerin birkaç konuda uzman, yetenekli ya da becerikli olmaları gerekmektedir. Bu anlamda literatürde 'çoklu-beceri 'ye sahip işgücü üzerine araştırmalar yapılmıştır. Araştırmalar sonucunda, proje çizelgelemede işçi üzerinden optimizasyon yaklaşımlarının olduğu, ancak ekiplerin belirlenmiş işler üzerine atamalarının yapıldığı görülmüştür. Bu çalışmanın temel düşüncesinde yer alan ekibin işin üzerindeki etkisi konusunda bir yaklaşım olmadığı sonucuna ulaşılmıştır. Bu çalışma ile ise esnek işgücü temelli üretim noktasında, gerçek hayatı daha iyi temsil edebilecek bir çözüm düşüncesi gerçekleştirilmeye çalışılmıştır.

### 3. MATERYAL VE YÖNTEM

#### 3.1. Tanımlama

Bu çalışma ile benzer işler için benzer beceri gerektiren işlerin bulunduğu işletmelerde çalışan işçi ve iş yükü dağılımının optimize edilmesi amaçlanmıştır. Benzer işleri yapan kişilerin bu işler arasında dağıtılabileceği ve bu işleri yerine getirebilecekleri bilinmektedir. Çağrı merkezlerinde, kendi alanlarında çağrılara cevap veren personelden ziyade, beraberce iş de yapılabilen atölye tipi üretim sistemlerinde çalışan işçiler, bu çalışmanın materyalini oluşturmaktadır. Çalışmaya temel teşkil eden işletmelerin tüm ortak özellikleri aşağıda listelenmiştir.

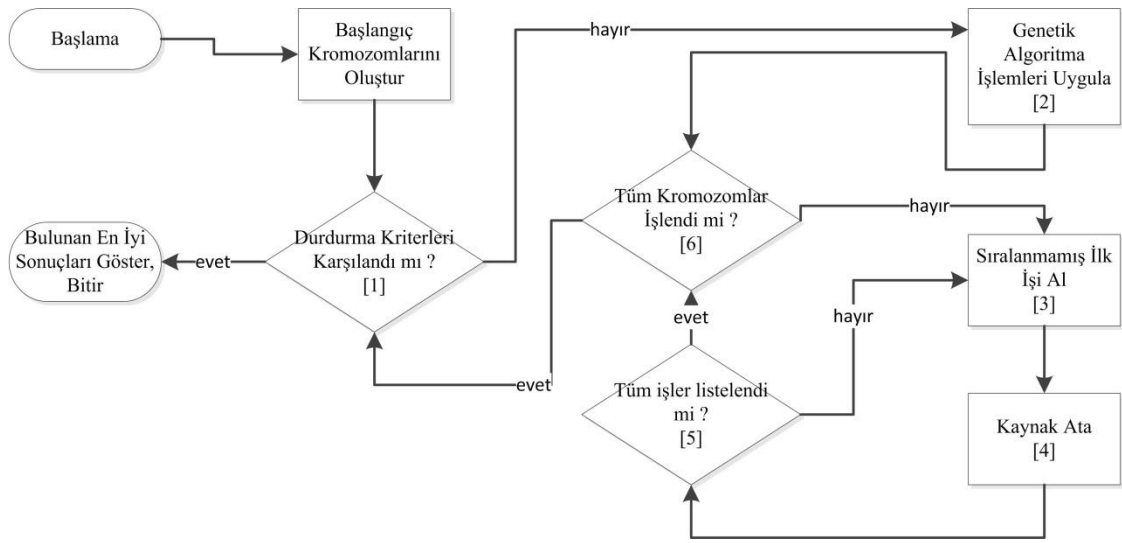
- Esnek üretim. (Farklı üretimler yapabilmeye müsait olmak.)
- Benzer işler. (İşin içeriğinin aynı ya da benzer olmasıdır. Örneğin matkap ile delme ya da sıcak mil ile yakarak delme işlemleri.)
- Belirli, kesin üretim çizelgesi. (Ne kadar, ne üretileceğinin baştan hesaplanmış olması)
- Farklı yeteneğe sahip kişiler. (Stajyer, Standart, Usta vb.)
- Bireysel ve/veya birden çok işçi gerektiren işler. (Aynı iş noktasında tek ya da daha fazla kişinin yapabildiği işlerdir.)
- Bölünemeyen işler (Başlandıktan sonra ara verilmeden tamamlanması gereken işlerdir.)
- İşgücünün yenilenebilir kaynak olması. (Sosyal güvenlik haklarından kaynaklanan ve vardiyalar, hafta sonları gibi işçinin çalışmayacağı durumların olmaması gerekmektedir.)

Son madde biraz daha detaylandırılırsa, bu çalışma ile toplamda ne kadar kaynak kullanıldığı incelenmektedir. Bu nedenle kaynaklar devamlı ve yenilenebilir kaynaklar sınıfındadır.

Çalışmanın temel problem kümesi, işçilerin beceri seviyelerinin birer mod olduğu ve işçilerin de birer kaynak olarak tanımlanacağı, RCPSP problemleri olacaktır.



Kullanılacak RCPSP problemleri standart PSPLib olarak literatürde geçen problemlerdir. (PSPLIB) Ancak bu çalışmada gerçek hayatı daha iyi temsil eden ek bilgiler bulunmaktadır. Bu nedenle orijinal problemi ve yaklaşımı biraz değiştirmek gerekmektedir. Örneğin gerekli kaynak miktarı, ilgili alanı bilen ekip sayısına denk gelmektedir. Problem değiştirildikten sonra MRCPSP problemlerine benzer hale gelecektir. Ancak MRCPSP tanımları da tam olarak karşılamamaktadır. Bu yeni yaklaşımda modlar baştan bilinen değil, seçilecek ekibin işi yapma süreleri farklılık göstereceği için dinamik olarak hesaplanan değerler olacaktır.



Şekil 3.1. Geliştirilen algoritmanın temel akış gösterimi

MRCPSP problemlerinin Np-zor olması ve problemin içerisinde gerçekleşecek olan işçi atamalarının da üstel olarak artması nedeniyle çözüm için sezgisel algoritma kullanmak gerekmektedir. Sezgisel algoritmalar içerisinde proje çizelgeleme problemlerine uygunluğu sebebiyle genetik algoritma tercih edilmektedir. ( Lab,S.C. 2004). Yapılan bu çalışmada da genetik algoritma kullanılmıştır. Şekil 3.1’de geliştirilen algoritmanın genel akışı gösterilmiştir.

Şekil 3.1’de genel akışı çizilen algoritmanın alt süreçleri bulunmaktadır. Şekil 3.1 üzerindeki, 1 ve 2 no’lu kutulardaki alt süreç ve parametreler Bölüm 3.6’da açıklanmıştır. Bu çalışma ile literatüre yapılan katkı 4 no’lu kutuda gösterilen, kaynak atama aşamasıdır. Kaynakların heterojen olması, birbirleri ile kullanımlarında etkileşimde bulunmaları ve sınırlı sayıda olmaları gibi kısıtları içinde barındırmaktadır.

Bu aşama Bölüm 3.7’de ayrıntılı olarak gösterilmiştir. Ancak daha öncesinde o an boşta (müsait) olan alternatif kaynaklardan hangisinin seçileceği kararlaştırılmış olmalıdır. Bu çalışmada müsait olan kaynak alternatiflerinin tercih edilmeleri için deterministik algoritmalar kullanılmıştır.

### **3.2. Verilerin Oluşturulması**

İşletmelerin bahsi geçen işçi tanımlarını ve özelliklerini veren verileri oluşturması bir süreç ve tecrübe gerektirmektedir. Bir işçinin verilen işi hangi sürede ve ne kalitede yapabileceği, konuya olan yatkınlığı ve becerisi, aldığı eğitimler göz önüne alınarak işletme tarafından bu kişiye, görevlendirmesi düşünülen iş için bir değer/çarpan verilir. Verilerin oluşması yine bu işleri daha önce ne kadar sürede yaptıkları ölçülerek bir ortalama ve üzerine bir risk standart sapması verilerek belirlenebilir.

İşçiler gruplandırma yapılarak, gruba bir çarpan verilerek işlem süreleri hesaplanabilir. Örneğin bu çalışmada, daha önce yapılan bir çalışma (Valls ve ark., 2009) ile aynı üç temel beceri düzeyi tanımlanmıştır. Bunlar stajyer, standart ve usta düzeyinde işçilerdir. Bu gruplar için örneğin, 0,8 veya 1,2 gibi bir değer verilebilir. Buna göre işin normal süreden %20 daha iyi veya %20 daha uzun sürede yapılabileceği öngörülebilmektedir. Bu işçilerin maliyetleri de özelliklerine göre değişmektedir.

### **3.3. Kullanılan Problem Setlerinin Özellikleri**

MRCPSp problemleri, iş sayısına, kaynakların moduna göre literatürde geçen problemler üzerinden karşılaştırma imkanına sahiptir. PSPLib olarak geçen bu kütüphanede ilgili örnek problemler bulunmaktadır. Bu problemlerin temel özelliklerine MMRCPSp konusunun anlatıldığı Bölüm 2.5’de değinilmiştir. Literatürdeki problemlerden bir tanesi, gösterimi kolaylaştırmak için kısaltılarak Çizelge 3.2, 3.3 ve 3.4’de verilmiştir. Problemin orijinal gösterimi EK-6’ya konulmuştur.

Problemlerin orjinalinde de süre ve maliyetler için birimler konulmamıştır. Bunun nedeni işin doğasına göre süre gün, dakika ya da saniye olarak alınabilir. Aynı şey maliyetler için de geçerlidir.

$R_1$  : Yenilenebilir kaynak no 1,

$R_2$  : Yenilenebilir kaynak no 2,

$N_1$  : Yenilenemeyen kaynak no 1,

$N_2$  : Yenilenemeyen kaynak no 2.

Çizelge 3.1. Proje Özellikleri (problem no: c154\_3)

Proje Sayısı	1 adet	-
İş Sayısı (başlama ve bitiş nodları dahil)	18 adet	-
Proje süresi	122 birim süre	-
Kaynaklar	Yenilenebilir Yenilenemeyen Çift Kısıtlı	2 adet 2 adet 0 adet
Teslim tarihi	22 birim süre	-
Gecikme maliyeti	2 birim lira	-
Kaynak Durumları	$R_1$ $R_2$ $N_1$ $N_2$	20 birim kapasite 26 birim kapasite 23 birim kapasite 36 birim kapasite

Çizelge 3.2. Öncüllük ilişkileri (problem no: c154\_3)

İş no	Mod	Ardıl iş sayısı	Ardıl işler
1	1	3	2 3 4
2	3	2	5 10
3	3	3	8 9 16
4	3	3	5 7 11
5	3	2	6 14
....	.....	....	....
18	1	0	....

Mod “3” olması, üç kaynak tipini de kullanabileceğini göstermektedir. Bu çalışmada yenilenemeyen kaynak olarak, bir üretim çizelgesi boyunca en fazla çalıştırılması gereken, örneğin usta sayısı, gösterilebilir. Böylece ustalar için ödenecek maliyet başta kısıtlanmış olur.

Çizelge 3.3. Kaynak tüketimi ve iş süresi (problem no: c154\_3)

İş No	Mod	Süre	R1	R2	N1	N2
1	1	0	0	0	0	0
2	1	1	6	0	0	1
	2	1	0	10	8	0
	3	9	0	8	8	0
3	1	4	6	0	0	5
	2	5	0	10	5	0
	3	7	0	10	0	4
....	....	....	....	...	...	...
18	1	0	0	0	0	0

### 3.3.1. Kullanılan problem setlerinin çalışmaya ait ek özellikleri

Çalışmaya özel problemde, normal RCPSP problemlerinden farklı olarak, beceri seviyeleri, bilgi alanları bulunmaktadır. Ayrıca hangi işçinin hangi alanı bildiği ve işlerin yapılabilmesi için hangi alanların bilinmesi gerektiği bilgileri de probleme eklenmiştir. Bu ek veriler örnek bir problem üzerinde gösterilmiştir. Anlatımını kolaylaştırmak için örnek problem boyutu küçük olması tercih edilmiştir.

Çizelge 3.4. Beceri düzeyleri

İşçi Sayısı	12 adet
İş Sayısı	18 adet
Uzmanlık alanı sayısı (A)	3 adet (A <sub>1</sub> ,A <sub>2</sub> ,A <sub>3</sub> )
Beceri seviyeleri (B)	3 adet (B <sub>1</sub> ,B <sub>2</sub> ,B <sub>3</sub> )

Çizelge 3.5. Beceri düzeyleri

Beceri Seviyesi	Kodu	Çarpan
Stajyer	B <sub>3</sub>	1,25
Standart	B <sub>2</sub>	1
Usta	B <sub>1</sub>	0,80

Bu çarpanlara göre bir işe, stajyer ve usta verildiğinde;  $(1,25) \times (0,80) = 1$  yani, stajyerin açığını usta olan kişi kapatmıştır.

Çizelge 3.6. Uzmanlık alanları

Uzmanlık Alan Adı	Kodu
Alan 1	A <sub>1</sub>
Alan 2	A <sub>2</sub>
Alan 3	A <sub>3</sub>

Çizelge 3.7. İşçi özellikleri,(uzmanlık alanı , beceri düzeyi karşılığı sırasıyla)

İşçi No	Uzmanlık Alanı	Beceri Düzeyi	Maliyet
1	A <sub>1</sub>	B <sub>3</sub>	50
2	A <sub>2</sub> , A <sub>3</sub>	B <sub>2</sub> ,B <sub>2</sub>	100
3	A <sub>1</sub> , A <sub>2</sub> ,A <sub>3</sub>	B <sub>2</sub> ,B <sub>3</sub> ,B <sub>1</sub>	120
4	A <sub>2</sub>	B <sub>3</sub>	50
5	A <sub>1</sub> , A <sub>3</sub>	B <sub>3</sub> ,B <sub>1</sub>	120
6	A <sub>3</sub>	B <sub>1</sub>	120
7	A <sub>1</sub>	B <sub>1</sub>	120
8	A <sub>3</sub>	B <sub>2</sub>	100
9	A <sub>3</sub>	B <sub>3</sub>	50
10	A <sub>2</sub> ,A <sub>3</sub>	B <sub>3</sub> ,B <sub>2</sub>	100
11	A <sub>2</sub>	B <sub>2</sub>	100
12	A <sub>1</sub> ,A <sub>3</sub>	B <sub>3</sub> ,B <sub>3</sub>	50

Çizelge 3.7 analiz edilirse, “1” no’lu işçi stajyer seviyesinde A1 alanında bilgilidir. Çizelge 3.5’ten bakılarak, bu işçinin iş süresini %25 uzatacağı anlaşılmaktadır. Bununla birlikte maliyeti ucuzdur. “3” no’lu işçi ise çoklu beceriye sahiptir. A3 alanında usta seviyesinde bilgili iken, A2 seviyesinde stajyer kadar bilgilidir. Buna göre bu işçi, A2 alanında olan bir işe atanırsa işin süresini %20 uzatacakken, A3 alanında olan işe atanırsa, %20 kısaltacaktır. Çizelge 3.8’de ise ilgili işin yapılabilmesi için hangi alanda bilgili olması gerektiği bilgisini içerir.

Çizelge 3.8. İş özellikleri (MRCPSp’ye uyumlu gösterim)

İş No	Uzmanlık Alanı	Örnek kaynak ihtiyaçları	Açıklama
1	A <sub>1</sub> ,A <sub>2</sub> ,A <sub>3</sub>	0	Başlangıç düğümü
2	A <sub>3</sub>	2	İş
3	A <sub>2</sub>	1	İş
4	A <sub>1</sub>	4	İş
5	A <sub>1</sub>	1	İş
6	A <sub>3</sub>	3	İş
7	A <sub>3</sub>	2	İş
8	A <sub>3</sub>	3	İş
9	A <sub>1</sub>	3	İş
10	A <sub>1</sub>	1	İş
11	A <sub>1</sub>	4	İş
12	A <sub>3</sub>	2	İş
13	A <sub>2</sub>	2	İş
14	A <sub>3</sub>	1	İş
15	A <sub>3</sub>	1	İş
16	A <sub>2</sub>	2	İş
17	A <sub>2</sub>	3	İş
18	A <sub>1</sub> ,A <sub>2</sub> ,A <sub>3</sub>	0	Bitiş Düğümü

İşçi maliyetleri birbirlerine görece değişken herhangi bir değer olarak standart işçi için 100 birim alınmıştır. Stajer için yarısı ödeme yapıldığı düşünülerek 50 birim maliyet ve usta için de %20 daha fazla maaş alacağı varsayımı ile 120 birim olarak düşünülmüştür. Maliyet hesabı için alınan bu arakmlar sonuçların birbirlerine göre değişimlerini göstermek için birer araçtırlar. Gerçek işçi maliyeti yaklaşımlarını yansıtmayabilirler.

Buraya kadar, bu çalışmada kullanılacak veri setlerinde olması gereken ek bilgiler tanıtılmıştır. Bu ek bilgiler doğrultusunda aynı parametreler kullanılarak uygun problemler geliştirmek için aşağıdaki kurallar geliştirilmiştir.

Problemin oluşturulmasında çok modlu kaynak kısıt problemleri kullanılması yerine RCPSP problemleri kullanılacaktır. Bunun nedeni, çok modlu problemlerinde başlangıçta işlerin bitiş süresi tanımlanmıştır. Bu çalışmada ise, seçilecek işçi tipine göre bu süre dinamik şekilde hesaplanacağı için baştan bilinemez. Bu süreler algoritmada anlık olarak hesaplanacaktır.

EK-7'de örnek gösterilen RCPSP probleminde, kaynak tipleri olarak gösterilen ( $R_1, \dots, R_N$ ) tanımlar yeni oluşturulan problemdeki iş için gerekli alanlara denk gelmektedir. Bu probleme ek olarak işçi özellikleri eklenecektir. İşçi sayısı toplam her kaynak tipi için gerekli kaynak sayısının toplamına eşit olacaktır.

$$\text{İşçi Sayısı} = \sum_{n=1}^k R_k$$

Her işçi beceri seviyesi, birer kaynağa denk geldiği için, en az işçi sayısı, anlık talep edilen en yüksek beceri seviyesi olması sağlanacaktır. Böylece problem için alan başına gerekli en az işçi sayısı belirlenmiş olur ve çözülebilir olması sağlanır. Özet olarak bir örnekle açıklanması gerekirse;

Orijinal problemde kaynak için gerekli birim tanımı şu şekilde geçmektedir.

RESOURCE AVAILABILITIES:

$R_1$	$R_2$	$R_3$	$R_4$
12	13	4	12

Bu çalışmada bu bilgi, 4 adet iş alanı olarak kabul edilecektir. Örneğin  $R_1$  alanı için  $R_2$  işçi gerekmektedir. Çalışmada toplam işçi sayısı ;

Toplam işçi sayısı =  $12 + 13 + 4 + 12 = 41$ 'dir.

Esnek üretim için çoklu beceriye sahip işçiler gerekmektedir. En az işçi sayısı ve en az işçi sayısının %50 fazlası kadar en fazla işçi sayısı aralığında bir rastgele rakam alınarak işçilerin bildikleri diğer beceriler rastgele şekilde oluşturulabilmiştir. Bu rastgele değeri üretmek için, orijinal problemdeki tohum değeri kullanılmaktadır. Tohum değeri ile, program her defa yeniden başlatılsa bile aynı rastsal değerler oluşturulabilmektedir. Böylece her defasında program yeni bir veri kümesi oluşturmayacaktır. Yapılanlar kısa bir örnek ile aşağıda açıklanmıştır.

Min  $R_1$ : 12 işçi

Maks  $R_1$ :  $12 * 1,5 = 18$  işçi

$R_1$  işçi sayısı =  $RND(12,18,tohum)$  , yani 12 ile 18 işçi arasında bir sayı olacaktır.

Yukarıda temel tanımları verilen örnek problemdeki maliyet alanı hesabı ise, her beceri alanı için verilen taban maaşların hesabı üzerine yapılmaktadır. Bir işçi en uzman olduğu alanın taban maaşı üzerinden fiyatlandırılacaktır.

Bazı problem kümeleri sadece tek bir kaynak kullanmaktadır. Fakat bu kaynağı hangi beceri seviyesinde kullanması gerektiği bilgisi eklenmemiştir. Örnek verilere bu tanım da eklenmek istenebilir. Bu ek tanım, durum 2 olarak açıklanacaktır.

#### a) Durum 1:

Yapılacak işin kimin yaptığıının önemsenmediği yaklaşımdır. İşin yapılması için toplam gereken kaynak miktarı verilmesi yeterli olacaktır.

İşçi beceri seviyelerinin kombinasyonu aşağıdaki şekilde hesaplanabilmektedir;

$n$ : beceri seviyesi sayısı

$r$ : talep edilen kaynak sayısı

$$n = \frac{(r + 2)(r + 1)}{2}$$

$n$  kadar beceri seviyesi için  $r$  kadar kaynak sayısına göre oluşacak farklı dağılım sayısı.

Buna göre, 3 beceri düzeyinin, 4 kaynak adedince oluşacak farklı sonuç sayısı 15'tir.

$$\frac{(4 + 2)(4 + 1)}{2} = 15$$

Görüldüğü üzere çözüm uzayını oldukça artıran parametrelerden biri de farklı beceri sayısı ve talep edilen kaynak sayısıdır. Çözüm uzayının boyutu üstel olarak artmakta ve problemi Np-zor hale getirmektedir.

Örneğin 7 numaralı iş için 2 birim kaynak gerektiği varsayalım. Bu durumda kaynaklar, 6 farklı sonucu söz konusu olur. Çizelge 3.9'da bu varsayımın ayrıntıları ve iş sürelerine etkileri gösterilmiştir.

Çizelge 3.9. İki kaynak için işçi kombinasyonu

Kombinasyon no.	İşçi kombinasyonu	Süreye etkisi	İşin Süresi (Birim)
0	Tek standart işçi için	$1 = 1$	10
1	Stajyer – Stajyer	$(1,25) \times (1,25) = 1,5625 \approx 1,56$	$10 \times 1,56 = 15,6$
2	Stajyer – Standart	$(1,25) \times 1 = 1,25$	$10 \times 1,25 = 12,5$
3	Stajyer – Usta	$(1,25) \times (0,8) = 1$	$10 \times 1 = 10$
4	Standart – Standart	$1 \times 1 = 1$	$10 \times 1 = 10$
5	Standart – Usta	$1 \times (0,8) = 0,8$	$10 \times 0,8 = 8$
6	Usta – Usta	$(0,8) \times (0,8) = 0,64$	$10 \times 0,64 = 6,4$

### b) Durum 2:

Bu yaklaşımda işi kimin yaptığı yaklaşımı önemsenmiştir. Bu noktada işin kalitesi öncelenmiş olabilir. Buna göre Çizelge 3.10'da 7 numaralı işi Çizelge 3.10'daki şekilde tanımlayabiliriz.

Çizelge 3.10. Önce belirlenen işçilik seviyesine göre iş tanımı

İş No	Uzmanlık Alanı	Usta	Standart	Stajyer	Diğer	Açıklama
7	A <sub>3</sub>	1	....	...	1	



Bu yaklaşıma göre bazı işlerin başına kesinlikle usta, stajyer veya standart işçi vermek imkanı doğmaktadır. Çizelge 3.10'u yorumlanırsa, diğer ile biri usta olmak üzere iki işçi ihtiyacı olduğu anlaşılmaktadır. Diğer işçi herhangi bir özelliğe sahip olabilir. Bu durumda işçi kombinasyonu azalmaktadır.

Oluşacak kombinasyon sayısını aşağıdaki şekilde hesaplanır.

$z$ : belirli kaynak sayısı

$r$ : tüm kaynak sayısı

$X$ : kadar beceri seviyesi  $z$  adet belirli kaynağa yönelik oluşacak farklı kaynak sayısı

$$X = \frac{(r + 1 - z)(r + 2 - z)}{2}$$

Örneğin 7 numaralı iş için,  $r = 2$ ,  $z = 1$  olarak verilirse, 3 adet kombinasyon olduğu bulunur. Çizelge 3.11'de bu kombinasyonlar ayrıntılı gösterilmiştir.

$$\frac{(2 + 1 - 1)(2 + 2 - 1)}{2} = 3$$

Çizelge 3.11. Belirlenen işçi ihtiyacına göre işçi kombinasyonu

Kombinasyon numarası	İşçi kombinasyonu	Süreye etkisi	İşin Süresi (Birim)
0	Tek standart işçi için	$1 = 1$	10
1	Stajyer – Usta	$(1,25) \times (0,8) = 1$	$10 \times 1 = 10$
2	Standart – Usta	$1 \times (0,8) = 0,8$	$10 \times 0,8 = 8$
3	Usta – Usta	$(0,8) \times (0,8) = 0,64$	$10 \times 0,64 = 6,4$

Bu yaklaşım ile işin üzerindeki işçilik etkisi daha kontrol edilebilir olmaktadır. Bununla birlikte hesap edilecek tüm ihtimaller çok daha azalsa bile, işlem hesabındaki karışıklık nedeniyle, çözüm süresinde bir iyileştirme sağlamamaktadır.

Bu tanımları orijinal probleme entegre edebilmek için aşağıdaki gösterilen alanların eklenmesi gerekmektedir. Orijinal probleme konulması düşünülen alanın örnek gösterimi aşağıdaki gibidir. Beceri taleplerini temsil eden B parametrelerinin bulunduğu sütun probleme eklenmiştir.

Orijinal Gösterim:

REQUESTS/DURATIONS:

jobnr. mode duration  $R_1$   $R_2$   $R_3$   $R_4$

2	1	8	4	0	0	0
3	1	4	10	0	0	0
4	1	6	0	0	0	3

Çizelge 3.12. Kolaylaştırılmış ek sütunlarla olan gösterim

İş no	Mod	Süre	$R_1$	$R_2$	$R_3$	$R_4$	$B_1(R_1, R_2, R_3, R_4)$	$B_2(R_1, R_2, R_3, R_4)$	$B_3(R_1, R_2, R_3, R_4)$
2	1	8	4	0	0	0	2 - - -	1 - - -	- - - -
3	1	4	10	0	0	0	5 - - -	- - - -	- - - -
4	1	6	0	0	0	3	- - - -	- - - 3	- - - -

Çizelge 3.12'den "2" numaralı iş için  $R_1$  uzmanlık alanında  $B_1$  beceri seviyesine sahip 2 kişinin bulunması gerektiği anlaşılmaktadır.

### 3.4. Uygulamada Kullanılan Problemler Ve Parametreler

Gerçek hayatta toplanan verilerin çok az ortak noktaları bulunmaktadır. Bu farklılıklar işletmenin hedefleri ve eldeki kaynakları ile ilgilidir. Geliştirilen algoritmaların da bu farklılıkları kullanabilmesi için örnek veri kümeleri gerekmektedir. Bu çalışmada örnek "veri üreten program" sayesinde, literatürdeki problemler temel alınarak, farklı problem kümeleri oluşturma imkanı oluşmuştur. Ayrıca yeni parametrelerle oluşturulan problemler de kendi aralarında aynı mantığa sahip olarak üretildikleri için sonuçlar üzerinde yorum yaparken hangi parametrelerin öne çıktığı daha kolay anlaşılabilir. Literatürdeki farklı problemlerin seçilmesi ile gerçek hayatta karşılaşılabilecek farklı örnekler (Çok sayıda pahalı işçi ya da tam tersi, çalışmayan işçiler gibi durumlar.) üretilerek çözüm aranmıştır. Bu temel noktalar haricinde örnek veriler sonuç olarak birer girdidir ve her zaman değiştirilebilirler. EK-5'te örnek problem üreten program ve EK-6' da çözüm üreten program kullanımı anlatılmıştır.

### 3.4.1. Kullanılan problemler

Bu çalışmada kullanılacak örnek problemlerin üretilmesi için “veri oluşturma programına” literatürde kullanılan problemler yüklenmiş ve Bölüm 3.3.1’de anlatılan ek parametrelerin de olduğu formata çevrilmişlerdir. Kullanılan problemler ve özellikleri Çizelge 3.13’te gösterilmiştir.

Çizelge 3.13. Kullanılan orjinal problemlerin özellikleri

Problem No	İş Sayısı (adet)	İşçi Sayısı / Kaynaklar (Sırasıyla)(adet)	Kaynak / Alan Sayısı (adet)	Bilinen En İyi Sonuç(birim süre)
J301_1	30	12-13-4-12	4	43
J302_1	30	9-11-11-16	4	38
J303_1	30	15-14-15-16	4	72
J304_1	30	10-22-26-13	4	49
J305_1	30	13-13-12-15	4	53
J601_1	60	13-11-12-13	4	77
J602_1	60	21-16-15-21	4	65
J603_1	60	20-21-18-21	4	60
J604_1	60	33-23-26-26	4	84
J605_1	60	15-18-17-16	4	76
J901_1	90	12-14-17-13	4	73
J902_1	90	15-26-26-22	4	96
J903_1	90	16-26-31-25	4	81
J904_1	90	33-32-40-21	4	93
J905_1	90	18-21-26-16	4	78
J1201_1	120	14-12-13-9	4	105
J1202_1	120	16-19-15-17	4	87
J1203_1	120	23-21-17-19	4	80
J1204_1	120	15-27-22-25	4	74
J1205_1	120	26-20-21-28	4	92

Çizelge 3.13’te kullanılan problemlerde, problem rakamları arttıkça karmaşıklıkları da artmaktadır. Örneğin 301 numaralı olan problem aynı anda sadece tek bir kaynak talep ederken, 305 numaralı problemde aynı anda birden fazla kaynak talebinde bulunmaktadır. Böylece bu çalışmanın farklı problemlerde ne kadar başarılı olduğu karşılaştırılabilecektir.

Örnek veriler oluşturulduktan sonra daha zengin tanımlar ve kısıtlar içeren yeni örnek problemler, daha az tanımlı ve kısıtlı orijinal problemden ayrılmaktadır. Bu ayrım

sonrasında 3 farklı yeni problem oluşmaktadır. Bu problemler ve tanımları aşağıdaki şekilde olacaktır.  $X$  tanımı problemdeki iş sayısını gösteren parametredir.

JX1\_1: Orijinal RCPSP problem kümesi. Örneğin 30 iş için J301\_1 no'lu iş olarak gösterilir.

JX1\_1a: İşçi ve becerilerin temel kaynak sayıları baz alınarak üretilen yeni problem kümesi.

JX1\_1b: a tipi problemin çözümünden sonra ilk optimizasyon olarak minimum işçi sayısı bulunarak, işçi sayısının azaltıldığı problem kümesi.

JX1\_1c: b tipi problemin çözümünden sonra ikinci bir optimizasyon ile teslim süresinin yakalanmaya çalışıldığı ve fason işçilerin eklendiği problem kümesi.

### 3.4.2. Yeni problem kümesi parametreleri

Yeni problemler oluşturulurken, yüklenen problemdeki rasgele oluşturmak için verilen tohum(seed) değer, tüm rastgele değişkenler için kullanılmıştır. Bir işçiye birden fazla alan tipi verilmesi, bir işçinin daha ne kadar alanlarda çalışabileceği ve ilgili alan için en fazla işçi sayısı gibi ihtimaller bu değere göre hesaplanmıştır. Tüm örnek problemlerde aynı altyapıyı sağlayabilmek için veri üretme programında varsayılan olarak, işçi sayısında gerekli değerlerin en fazla kişi sayısı %150'si olabilme ihtimali eklenmiştir. Örneğin  $R_1$  alanında J301\_1 numaralı problemde 12 işçi ihtiyacı vardır. Örnek problemde ise bu alanı bilen en fazla  $12 * 1,5 = 18$  işçi olabilecektir. Buna göre 12 ve 18 işçi arasında rastgele bir rakam alınarak ilgili alanı kaç işçinin bildiği bilgisi oluşturulacaktır.

Beceri çarpanları daha önce yapılan bir çalışmadaki değerlerle benzer olarak verilmiştir. ( $B_1$  0,8 –  $B_2$  1 –  $B_3$  1,25) Bu verilere ek olarak,  $B_1$  seviyesinde ama maaşı daha yüksek olan fason üretici de kaynak işçi olarak eklenebilmektedir.  $B_0$  tanımı ile eklenen fason işçi, tüm alanlarda çalışabilecek ve usta niteliğinde iş yapabilecektir.

İşçi maliyetlerinde ise, en kötü işçi stajyer düşünülmüş ve bir stajyer normal maaşın yarısını alabilir olarak değerlendirilmiştir. Usta ise normal ücretin %20 fazlasını aldığı

varsayılarak, temel maaş değerlerini 120 – 100 ve 50 birim lira olarak kullanılmıştır. En son, her işi yapabilecek ve ustadan daha masraflı olan fason işçiler için de 140 birim lira maliyet konulmuştur. Burada verilen işçi maliyetleri değişken ve değiştirilebilir olup, bu çalışma için sadece bağıl bir karşılaştırmaya yardımcı olmak amacıyla seçilmiştir.

Erken bitirme maliyetleri tüm problemler için sıfır olarak alınmıştır. Bunun nedeni işi erken bitirmenin, işçi dağılımının daha iyi yapılamamasına sebep verebilme ihtimalidir. Erken bitirmenin de amaç fonksiyona dahil edilmesi durumunda çoğunlukla fason işçi talebi olacaktır. Buradaki amaç ise mevcut işçileri optimize bir şekilde dağıtabilmektir. Ayrıca temel hedef işi zamanında bitirmektir, daha önce bitirmek değildir.

### 3.5. Matematiksel Model

Bu çalışmada klasik MMRCPSPP problemlerinden daha fazla sayıda değişken ve kısıt bulunmaktadır. Bu sayının artmasında temel olarak, işlerin ve işçilerin beceri seviyelerine göre kısıtlarının bulunması ve işlem sürelerinin heterojen kaynaklara göre hesaplanmasının getirdiği ek işlemlerin etkisi bulunmaktadır.

#### İndeks Elemanları:

$j, v$ : iş sayısı indis elemanı,

$i$ : işçi sayısı indis elemanı,

$b$ : beceri indis elemanı

$\forall i, j, b, v \in N$

#### Parametreler:

$J = 1, \dots, j$  ( $j$  kadar iş sayısı vardır.) İşler bölünemez.

$L = 1, \dots, i$  ( $i$  kadar işçi sayısı vardır.)

$B = 1, \dots, b$  ( $b$  adet beceri sayısı vardır.)

$A_r = \{ a_{r1}, \dots, a_{rb} \}$  uzmanlık alanları çarpanları.

$A_c = \{ a_{c1}, \dots, a_{cb} \}$  beceri seviyelerine göre maliyetleri.

$A_j = \{ k * a_b \forall k, b \in N \}$   $j$  işinin ihtiyaç duyduğu beceri tipleri ve sayısı

$A_i = \{ k * a_b \forall k, b \in N \}$  i işçisinin becerileri

$D_v$  = işlerin öncüllüklerini gösteren küme.

$p_j$  = j işinin normal işlem süresi,

$dd$  = Teslim zamanı

$T$  = Gecikme maliyeti

$E$  = Erken bitirme maliyeti

$M_i$  = i işçisinin maaşı

### Karar Değişkenleri:

$d_j$  = j işinin hesaplanan işlem süresi,

$s_j$  = j işinin başlama süresi.

$f_j$  = j işinin bitiş süresi.

$X_{i,j}$  : i işçisinin  $s_j$  süresinde müsait olup olmadığını gösterir.

$L_{i,j,b}$  : i işçisinin j işinde b becerisi olduğunu gösterir. 0 ise bu alanda becerisi yoktur.

$\tau$  : pozitif teslim süresi

$\delta$  : pozitif erken bitirme süresi

$C_{max}$  = Tamamlanma süresi

$\forall s, f, C, d \in R \quad X \in \{0,1\}$

### Kısıtlar:

$$X_{i,j} = \begin{cases} 1, & \text{eğer } i \text{ işçisi } s_j \text{ anında müsait ise} \\ 0, & \text{değil ise} \end{cases} \quad (3.1)$$

$$L_{i,j,b} = \{ L_i, i \mid X_{i,j} * a_{i,b} * A_{i,b} > 0 \} \quad (3.2)$$

$$\sum_{t=\min s_j}^{\max f_j} X_{i,j} = 1 \quad (3.3)$$

$$A_{j,b} \geq \sum_{i=1}^L L_{i,j,b} \quad (3.4)$$

$$A_j \geq \sum_{i=1}^L L_{i,j} \quad (3.5)$$

$$d_j = \prod_{b=1}^B \prod_{i=1}^L L_{i,j,b} * ar_b * p_j \quad (3.6)$$

$$s_j \geq s_v + d_j \quad v \in Dv \quad (3.7)$$

$$Cmax = \sum_{j=1}^n d_j \quad (3.8)$$

$$\tau = \begin{cases} \text{Eğer } Cmax \geq dd & Cmax - dd \\ \text{Eğer } Cmax < dd & 0 \end{cases} \quad (3.9)$$

$$\delta = \begin{cases} \text{Eğer } Cmax < dd & dd - Cmax \\ \text{Eğer } Cmax \geq dd & 0 \end{cases} \quad (3.10)$$

### Amaç Fonksiyonu:

$$\min Z = \tau * T + \delta * E + \sum_{n=1}^k M_k$$

3.1 numaralı kısıt, bir işin başlama zamanında müsait olan işçileri gösteren kısıttır. Bu sayede başka bir işe atanmış işçiler bu işe atanmayacaklardır. 3.2 numaralı kısıt ile,  $J$  işinde,  $b$  becerisine sahip çalışacak işçilerin listesi verilmektedir. Böylece  $j$  işinin talep ettiği  $b$  becerisine sahip olmayan işçilerin atanması engellenecektir. 3.3 numaralı kısıt ile bir işin minimum başlama ve maksimum bitiş süresi içerisinde o işçinin müsait olmadığını gösterir. 3.4 numaralı kısıt ile,  $j$  işinin ihtiyaç duyduğu  $b$  beceri tipinden olan kaynak ihtiyacının karşılanmasını sağlar. Ayrıca bir işçi birden fazla beceriye sahipse bile, ilk atandığı beceri tipinde kalacak ve aynı iş için bir daha atanabilir olamayacaktır. 3.5 kısıtı ile, beceri tipi belirtilmemiş kaynak ihtiyaçlarının da karşılanmasını sağlar. 3.6 numaralı kısıt ile, normal olarak  $p_j$  sürecek bir  $j$  işinin işçilerin yeteneklerine göre yeniden hesaplanmasını sağlar. Normal olarak tüm kaynakların standart olduğu düşünülen bir  $j$  işi 3 birim sürecek iken, örneğin usta atanması sonucu 2,25 birim sürecektir. 3.7 numaralı kısıt ile ise, öncüllük ilişkileri kısıtı sağlanmaktadır. Tüm işler bitiş-başlama önceliğine çevirilmiştir. Başlama için bir hazırlık veya gecikme zamanı bulunmamaktadır. Bu kısıt, işlerin öncüllerinden sonra başlamasını sağlar. 3.8 numaralı formülde ise, tamamlanma süresinin hesaplanması gösterilmektedir. 3.9 ve 3.10 da ise, pozitif erken bitirme ya da geç bitirme süresi hesaplanmaktadır. Aksi takdirde erken bitirme ya da geç bitirme maliyetleri 0 olarak alınacaktır.

Son olarak amaç fonksiyon ise maliyet ve tamamlanma sürelerini göze alarak maliyetin minimizasyonunu amaçlamaktadır. Bilindiği gibi, tüm projelerin teslim tarihi bulunmaktadır. Bu tarihi geçen her birim zaman için, ceza maliyeti bulunmaktadır. Erken bitirme durumu da söz konusu olabilir. Erken ya da geç bitirme durumlarında negatif ya da pozitif bir çarpan verilerek hedef fonksiyon, bitirmenin ne yönde olacağını istenmesine göre etkilenebilir. Çalışan işçilerin maliyetleri de eklenerek maliyet tabanlı hedef fonksiyon oluşturulacaktır.

En ucuz işçi seçimi veya en kötü işçi seçimi gibi işçi seçiminde tercihte bulunulmak istenildiği durumlar için, işçiler en düşük maaşlı, en pahalı maaşlı veya en becerikli gibi bir listeye sıralanabilir. Bu listeden sıra ile alınacak işçiler, istenilen seçime göre işlere atanmış olurlar.

### **3.5. Genetik Algoritma parametreleri**

RCPSp problemleri üzerine yapılan bir çalışmada, 1000 çizelgeleme için mutasyon oranının en iyi %5 olması durumunda, popülasyon boyutunun 25-40 adet arasında olduğu, çift noktalı çaprazlama yapılmasını ve elitist seçim yapılması durumunun en iyi çözümleri ürettiği sonuçları alınmıştır. (Hartmann, 1998) Adı geçen çalışmanın temel parametreleri denenerek, kullanılan iterasyon sayısı, mutasyon oranı ve tipi, durdurma kriteri olarak kabul edilen rakamlar ileride bahsedilecek nedenlerle seçilmişlerdir. İlk kromozomlar oluşturulduktan sonra, herhangi bir yeni üreme yapılmamaktadır. Uygunluk değeri, en küçük bitirme süresidir.

Başlangıç kromozomları “rastgele”, “en çok ardılı olan işe göre”, “en az ardılı olan işe göre” ve “hemen sonraki ardıl işlerin adetine göre” üretilmektedir. Her biri %25 ihtimale sahiptirler. Üretilen rastgele rakam hangi aralığına düşerse o algoritma kullanılarak kromozomlar üretilirler. Bu sayede olabildiğince farklı zenginlikte kromozom üretilme imkanı doğmaktadır.

Bu noktada kromozomun nasıl oluşturulduğu, çaprazlamanın nasıl yapıldığı ve mutasyonun nasıl gerçekleştirildiği anlatılacaktır. Çözüm aşamalarının daha kolay



anlaşılması için örnek olarak Çizelge 2.11’de bilgileri verilen problem kullanılacaktır. Öncüllük ilişkisi görsel olarak ise Şekil 2.5’te verilmiştir.

### **Kromozom Oluşturma**

Kromozom uzunluğu iş sayısı kadardır. Örneğin 30 iş için kromozom uzunluğu 30’dur. (Örnek problemde iş sayısı 11 olduğu için kromozom uzunluğu 11 olarak alınmıştır.) Kromozomdaki iş sırası algoritma tamamlandığında sırasıyla yapılacak işler için en ideal sıra olacaktır. Bazı işlerin işletme bazında öncelikleri söz konusudur. Bu nedenle ideal olan sıra bilinse bile, ilk olarak yapılmak istenen işler olabilir. Kromozom oluşturulurken hem bu talepler, hem de popülasyonda çok değişik kromozomlar üretebilmek için, çok ardılı olan işlere göre, az ardılı olan işlere göre, rastgele alınan sıralamalara göre veya önem derecesine göre işler sırayla atanırlar. Bu çalışmada rastgele alınan bir rakama göre işler, en çok ardılı olan, en az ardılı olan ve rastgele alınarak kromozomlara sırasıyla atanmışlardır.

*S*: Sıralanmamış işler kümesi ( İlk iterasyon için tüm işlerin bulunduğu listedir. ).

*E*: Müsait işler kümesi. ( Sıralanmamış işler kümesinden oluşturulur. )

*K*: Kromozom ( İş sayısı kadar genenden oluşmaktadır. Başlangıç ve bitiş genlerini, yani iş numaralarını üzerinde bulundurmamaktadır.)

*i*: iterasyon sayısı.

*r*: rastgele üretilen sayı

Müsait işler kümesi program tarafından hesaplanır ve doldurulur. Öncül kısıtı olmayan işler ile *E* listesi doldurulur. Sıralanmamış işler kümesi algoritmanın başlangıcında tüm işlerin bulunduğu listedir. İşler kromozoma yerleştirildikçe bu kümeden kaldırılırlar.

**Aşama 1:** İş adeti kadar kromozom uzunluğu olan kromozom dizisi oluştur. *S* kümesini doldur.

**Aşama 2:** Öncüllük kısıtı ve başlama zamanı kısıtı olmayan işlerden *E* kümesini doldur.

**Aşama 3:** Rastgele bir sayı üret. Müsait işler kümesi 0-100 arası bir yüzdeliğe göre böl ve yüzdelerle sırasıyla bir önceki yüzde toplamları ile kümülatif olarak ilgili işlere ata. Üretilen rastgele rakam hangi aralığa geliyorsa o işi al.

**Aşama 4:**  $S$  kümesinde bir iş var ise, **Aşama 2'**e dön. Yoksa kromozom oluşturulmuştur, algoritma kromozom oluşturma sürecini bitir.

$i=0$ ;

$S: \{ 1,2,3,4,5,6,7,8,9,10,11 \}$

$E: \{1,2\}$

K: 

-	-	-	-	-	-	-	-	-	-	-
---	---	---	---	---	---	---	---	---	---	---

$r = 0,25$

$E$  kümesinden rastgele aralığa düşen işi al.

$$işno = \begin{cases} r < 0,50 & işno = 1 \\ 0,50 < r < 1 & işno = 2 \end{cases}$$

K: 

1	-	-	-	-	-	-	-	-	-	-
---	---	---	---	---	---	---	---	---	---	---

$i=1$

$S: \{ 2,3,4,5,6,7,8,9,10,11 \}$

$E: \{2,3,4\}$

$r = 0,55$

$$işno = \begin{cases} r < 0,33 & işno = 2 \\ 0,33 < r < 0,6 & işno = 3 \\ 0,6 < r < 1 & işno = 4 \end{cases}$$

K: 

1	3	-	-	-	-	-	-	-	-	-
---	---	---	---	---	---	---	---	---	---	---

$i=2$

$S: \{ 2,4,5,6,7,8,9,10,11 \}$

$E: \{2, 4,9\}$

$r = 0,95$

$$işno = \begin{cases} r < 0,33 & işno = 2 \\ 0,33 < r < 0,6 & işno = 4 \\ 0,6 < r < 1 & işno = 9 \end{cases}$$

K: 

1	3	9	-	-	-	-	-	-	-	-
---	---	---	---	---	---	---	---	---	---	---

Yukarıda gösterilen iterasyon  $S$  kümesinde iş kalmayınca kadar devam ederek kromozom oluşturulur. ( $i=10$ , örnek için iterasyon sayısı 10 olunca durur.) Bu şekilde oluşturulmuş iki adet kromozom çaprazlanırken öncüllük ilişkilerine dikkat etmek gerekmektedir. Bu çalışmada kullanılan iki noktadan çaprazlama tekniği kullanılmıştır. Yukarıda anlatılan şekilde üretilen iki kromozom çift noktadan çaprazlama yapılmış ve aşağıda işlem aşamaları gösterilmektedir.

### Çaprazlama İşlemi

Bu çalışmada iki noktadan çaprazlama yöntemi kullanılmıştır. Bu çaprazlama yöntemine göre rastgele alınan iki rakam arasında kalan alan, diğer kromozom ile değiştirilir. Ancak öncüllük ilişkileri nedeniyle birebir kopyalanamazlar. Algoritma aşağıdaki aşamalardan oluşmaktadır:

**Aşama 1:** İki adet en az 1, en fazla kromozom uzunluğu kadar rastgele rakam üret. Birinci çaprazlama noktasına denk gelen kısımdan öncesini yeni genlere değiştirmeden aktar. Değiştirilecek gen sayısını tutan bir değişken( $m$ ) oluştur.

**Aşama 2:** Değiştirilecek aralıkta sıralanmamış iş var ise sıradaki ilk işi al. Yoksa tüm kromozoma sırası ile bak ve olmayan işi al. Yerleştirilecek kromozomda bu atanacak işin olup olmadığına bak.

**Aşama 2.1:** Eğer iş var ise, Aşama 2'ye dön. Eğer iş yok ise, Aşama 3'e geç.

**Aşama 3:** Yerleştirilecek işin öncülü olup olmadığına bak. Eğer öncülü var ise, önce o öncülü olan işi veya işleri yerleştir. Taşınan iş yerleştirildi ise  $m$  değişkeninden 1 azalt.

**Aşama 4:**  $m$  değişkeni 0'dan büyük ise Aşama 2'ye dön, yoksa Aşama 5'e geç.

**Aşama 5:** İkinci çaprazlama noktasından sonraki kalan işleri yerleştir. Algoritmayı durdur.

$r_1, r_2$  : Rastgele üretilen sayılar

$m$ : Yerleştirilecek gen sayısı

K1:	1	3	4	9	5	6	2	7	10	8	11
K2:	2	1	7	4	5	6	3	9	10	11	8

$$r_1 = \text{Rand}(1,11) = 3$$

$$r_2 = \text{Rand}(1,11) = 7$$

$m = 7-3+1 = 5$  ( 5 adet gen yer değiştirecektir. Bir artırılmasının nedeni ilgili genlerin sayıya dahil olmasıdır. )

Üretilen iki sayı, kromozomdaki 3. ve 7. sıradaki noktalardan çaprazlama yapılacağını göstermektedir.

K1: 

1	3	4	9	5	6	2	7	10	11	8
---	---	---	---	---	---	---	---	----	----	---

K2: 

2	1	7	4	5	6	3	9	10	11	8
---	---	---	---	---	---	---	---	----	----	---

Yeni üretilecek kromozomların adı sırası ile C1 ve C2 olsun. İlk nokta öncesi bırakılır.

C1: 

1	3	-	-	-	-	-	-	-	-	-
---	---	---	---	---	---	---	---	---	---	---

C2: 

2	1	-	-	-	-	-	-	-	-	-
---	---	---	---	---	---	---	---	---	---	---

K1 kromozomu incelenecek olursa, ilk 4 no'lu iş C2 kromozomuna aktarılacaktır. Öncüllerine bakılır bir kısıt yoksa konulur. m sayısı 1 azaltılır.

C2: 

2	1	4	-	-	-	-	-	-	-	-
---	---	---	---	---	---	---	---	---	---	---

$$m = m - 1 = 4$$

Ancak sıradaki 9 no'lu iş için öncelikle 3 no'lu işin de yapılması gerekmektedir. Bu nedenle, 9 no'lu işten önce 3 no'lu iş konulur.

C2: 

2	1	4	3	9	-	-	-	-	-	-
---	---	---	---	---	---	---	---	---	---	---

$$m = m - 1 = 3$$

Sıradaki 5 ve 6 nolu işler için bir kısıt bulunmamaktadır.

C2: 

2	1	4	3	9	5	6	-	10	11	8
---	---	---	---	---	---	---	---	----	----	---

$$m = m - 2 = 1$$

K1 kromozomundan alınacak 2 no'lu iş C2 kromozomunda zaten vardır. Bu nedenle yeri değişmeyecektir. Geriye kalan iş numarası çaprazlanan kromozomda olup, çocuk kromozomda olmayan genlerdir. Sırasıyla bu genler doldurularak C2 kromozomu tamamlanacaktır.

C2: 

2	1	4	3	9	5	6	7	10	11	8
---	---	---	---	---	---	---	---	----	----	---

$m=m-1 = 0$

Aynı şekilde C1 kromozomu da doldurularak öncüllük ilişkileri olurlu kromozomlar üretilmiş olur.

C1: 

1	3	2	7	4	5	6	9	10	8	11
---	---	---	---	---	---	---	---	----	---	----

$r_1$  ve  $r_2$  rakamları problemin türüne göre değiştirilemeyecek nokta aralığı seçse idi, bu durumda yeni kromozomlar, doğrudan eski kromozomların aynısı olacaktı. Bu durum  $r_1=9$  ve  $r_2=11$  için görülmektedir. Çaprazlama oranı daha düşük değerler verilse idi, bu kısıtlar nedeniyle daha da düşük çaprazlamaya girilmiş olacak ve başlangıç değerlerini yansıtmayacaktı.

Çaprazlama oranı için testler yapılmıştır. Yapılan testler Çizelge 3.14'te gösterilmiştir. Bu sonuçlara göre çaprazlama oranı 1 olarak alınmıştır. Bunun anlamı hiçbir birey doğrudan kendini yeni nesile doğrudan aktaramayacaktır. Ancak bu ve benzeri çalışmalar için tüm bireyler çaprazlamaya alınsa bile, tamamen değiştiği söylenememektedir. Öncüllük ilişkilerinden dolayı çaprazlama yapılsa bile kromozom değişmeyebilmektedir. Bu durum çaprazlamanın nasıl yapıldığı yukarıdaki bölümde gösterilmiştir. Ayrıca düşük çaprazlama oranında hesaplama süresinin de azaldığı farkedilmiştir. Bunun nedeni ise, çaprazlama işlemi yapılmadığı için daha az işlem yapılmakla birlikte, daha az sayıda nüfus çeşitliliği sebebiyle algoritma durdurma kriterlerinden olan, son 50 sonucun aynı olması nedeniyle algoritmanın hesaplamayı sonlandırmasıdır.

Çizelge 3.14. Çaprazlama oranı karşılaştırma tablosu

Çaprazlama Oranı	Problem No	Tekrarlama Sayısı	İterasyon Sayısı	Ortalama Sonuç	Bulunan En İyi Sonuç	Bulunan En Kötü Sonuç	Bilinen En İyi Sonuçtan Ortalama Sapma
<b>1</b>	<b>J301_1a</b>	<b>10</b>	<b>1000</b>	<b>56,9</b>	<b>43,95</b>	<b>70,92</b>	<b>%132</b>
0,95	J301_1a	10	1000	57,1	45,89	72,06	%132
0,90	J301_1a	10	1000	57,3	49,32	77,81	%132
0,85	J301_1a	10	1000	57,2	46,35	79,16	%132
0,80	J301_1a	10	1000	58,1	45,89	82,66	%135
<b>1</b>	<b>J1201_5a</b>	<b>10</b>	<b>1000</b>	<b>145,7</b>	<b>126,64</b>	<b>234,64</b>	<b>%138</b>
0,95	J1201_5a	10	1000	148,7	131,54	195,47	%141
0,90	J1201_5a	10	1000	146,9	133,05	212,76	%139
0,85	J1201_5a	10	1000	146,9	133,26	197,42	%139
0,80	J1201_5a	10	1000	148,8	135,59	233,64	%141

## Mutasyon

Mutasyon tipi olarak, yer değiştirme mutasyonu yukarıda bahsi geçen(Hartmann,1998) çalışmada tavsiye edilmiştir. Ancak bu çalışmada öncül kısıtlarının bulunması nedeniyle bu mutasyon kullanılmamıştır. Bunun yerine mutasyona girildiğinde kesinlikle mutasyon işlemi yapacak olan sağa kaydırma mutasyonu yapılmıştır. Bu mutasyon tipine Çizelge 2.10'da bahsedilmiştir. Böylece kullanılacak mutasyon oranı gerçek mutasyon oranı olabilecektir. Mutasyonu diğer yöntemlerle olmaya zorlamak da bir çözüm olabilir. Ancak rakamların rasgele oluşturulması nedeniyle, algoritmanın kabul edeceği mutasyonu kaç adet iterasyonda bulabileceği kestirilememektedir. Bu da çok uzun işlem zamanı anlamına gelebilir. Mutasyon aşamaları aşağıda anlatılmıştır.

**Aşama 1:** Kromozom uzunluğu ile 1 arasında rastgele bir rakam alınır. Bu rakam ile mutasyon yapılacak gen bulunur.

**Aşama 2:** Mutasyon yapılacak gen üzerindeki iş numarasına bakılır ve öncüllük ilişkilerini bozmadan en çok ne kadar kaydırılabileceğine bakılır.

**Aşama 3:** 1 ile en çok kaydırılabilecek rakam arasında bir rastgele sayı alınır ve iş çıkan rakam kadar sağa kaydırılır.

$$r_1 = \text{Rand}(1,11) = 4$$

C1: 

1	3	2	7	4	5	6	9	10	8	11
---	---	---	---	---	---	---	---	----	---	----


7 no'lu iş, en fazla 8 no'lu işin önüne kadar kaydırılabilir. Çünkü 8 no'lu işin yapılabilmesi için 7 no'lu işin bitmiş olması gerekmektedir. Buna göre,

$r_{2max} = 6$ 'dır. Yani 7 no'lu iş en fazla 6 gen sağa kaydırılabilir.

$r_2 = \text{Rand}(1,6) = 3$ , 3 gen sağa kaydırılacak anlamına gelir.

C1: 

1	3	2	7	4	5	6	9	10	8	11
---	---	---	---	---	---	---	---	----	---	----



C1: 

1	3	2	4	5	6	9	10	8	11
---	---	---	---	---	---	---	----	---	----



C1: 

1	3	2	4	5	6	7	9	10	8	11
---	---	---	---	---	---	---	---	----	---	----

Bu bilgiler doğrultusunda mutasyon oranları denenmiş ve en iyisi bulunmaya çalışılmıştır. Sonuçlar Çizelge 3.15'te gösterilmiştir.

Çizelge 3.15. Mutasyon oranı karşılaştırma tablosu

Mutasyon Oranı	Problem No	Tekrarlama Sayısı	İterasyon Sayısı	Ortalama Sonuç	Bulunan En İyi Sonuç	Bulunan En Kötü Sonuç	Bilinen En İyi Sonuçtan Ortalama Sapma
0,01	J301_1a	10	1000	56,6	49,15	70,92	%131
0,05	J301_1a	10	1000	56,5	43,84	72,06	%131
<b>0,1</b>	<b>J301_1a</b>	<b>10</b>	<b>1000</b>	<b>51,6</b>	<b>49,32</b>	<b>67,87</b>	<b>%12</b>
0,2	J301_1a	10	1000	57,5	46,35	69,06	%133
0,5	J301_1a	10	1000	56,2	43,84	72,06	%13
0,01	J1201_5a	10	1000	148,6	126,64	234,64	%161
0,05	J1201_5a	10	1000	148,7	110,84	195,47	%161
<b>0,1</b>	<b>J1201_5a</b>	<b>10</b>	<b>1000</b>	<b>144,6</b>	<b>113,89</b>	<b>212,76</b>	<b>%157</b>
0,2	J1201_5a	10	1000	146,9	124,26	188,42	%159
0,5	J1201_5a	10	1000	145,8	117,54	203,85	%158

Problemler fason işçi kullanılmadan, bilinen en iyi süre ile çözüm sonucu çıkan tamamlanma süresi üzerinden hesaplanmıştır. Tablodaki sonuçlara göre mutasyon

oranının çok az ya da fazla olması problemin daha iyiye gittiğini göstermemektedir. Bu sebeple %10 mutasyon oranı kullanılmıştır.

Popülasyon boyutunun bahsi geçen çalışmada 25 ile 40 boyutunda olması tavsiye edilmiştir. Buna göre aynı problemler için popülasyon boyutları denenmiştir.

Çizelge 3.16. Popülasyon boyutu karşılaştırma tablosu

Popülasyon Boyutu	Problem No	Tekrarlama Sayısı	İterasyon Sayısı	Ortalama Sonuç	Bulunan En İyi Sonuç	Bulunan En Kötü Sonuç	Bilinen En İyi Sonuçtan Ortalama Sapma
25	J301_1a	10	1000	56,9	49,64	65,18	%132
<b>30</b>	<b>J301_1a</b>	<b>10</b>	<b>1000</b>	<b>56,2</b>	<b>49,32</b>	<b>61,05</b>	<b>%130</b>
35	J301_1a	10	1000	56,6	49,32	70,92	%131
40	J301_1a	10	1000	56,1	49,32	72,84	%130
25	J1201_5a	10	1000	147,7	112,17	236,34	%160
30	J1201_5a	10	1000	145	112,17	234,64	%157
<b>35</b>	<b>J1201_5a</b>	<b>10</b>	<b>1000</b>	<b>143,8</b>	<b>112,17</b>	<b>220,88</b>	<b>%156</b>
40	J1201_5a	10	1000	148,2	120,89	234,64	%161

Çizelge 3.16'daki sonuçlardan da anlaşılacağı üzere, popülasyon boyutu olarak 30-35 arası bir rakam seçilmesi uygun olacaktır. 30 popülasyon boyutunun işlem süresinin daha kısa olacağı nedeniyle popülasyon boyutu olarak 30 seçilmiştir.

Durdurma ölçütleri olarak “süre”, “maksimum iterasyon sayısı” ya da sonuçların daha iyiye gitmediği durumlar için “tekrar edecek maksimum iterasyon sayısı” kuralları vardır. Maksimum iterasyon sayısı oldukça yüksek olarak 1500 olarak seçilmiştir. Bu maksimum iterasyon sayısına ulaşmadan önce program aynı sonuçları üretmeye başlarsa ve bir geliştirme olmazsa durdurulacaktır. Değişiklik olması mutasyon oranına bağlıdır. Bu nedenle tüm kromozomların aynı olduğu noktaya gelindiğinde, son olarak mutasyon olmamışsa sistem durdurulur. Bu rakam da popülasyon boyutundan büyük bir rakam olmalıdır. Bu sebeple son 50 sonuç aynı ise algoritma durdurulacaktır. İstenildiği takdirde durdur butonuna basılarak veya ayarlanan süreyi aşması durumunda da durdurulabilir.



### 3.7. Kaynak Atama

Kaynakların atanabilmesi için müsait konumda olmaları gerekmektedir. Bir başka deyişle herhangi bir işe atanmamış olmaları gerekmektedir. Yapılan çalışmada müsait işler için tutulan bir değişkenler kümesi, her iş yerleştirildikten sonra güncellenmektedir. Bir işe birden fazla kaynak aynı anda atandığı için, aynı zaman aralığında ilgili kaynakların tümünün müsait olması gerekmektedir. Eğer işin yapılabilmesi için tek alternatif var ise, bu işlemi yapabilecek tüm kaynakların müsait olması beklenir ve müsait oldukları an işe atanırlar.

Klasik kaynak kısıtlı proje çizelgeleme problemlerinde her işin talep ettiği kaynak bellidir ve hangi modda yapılacaksa iş süresi baştan bilinmektedir. Bu çalışmada ise, iş süreleri atanacak kişilerin becerilerine göre değişmektedir. Ayrıca birden fazla işçinin aynı işe atanabilmesi nedeniyle çok sayıda farklı iş süreleri çıkabilmektedir. Çok sayıdaki alternatif kaynak kombinasyonları nedeniyle algoritma istenilen derecede iyi sonuçlar üretemeyebilir. Çünkü gerçek hayatta alternatif kaynaklardan yapılan ideal seçimler, çok sayıda oluşan kombinasyondan küçük bir kısımdır. Çok sayıdaki kombinasyonu engellemek amacıyla SPT, LPT algoritmalarına benzer deterministik algoritmalar kullanılmıştır. Dört adet deterministik algoritma kullanılmıştır. Bu algoritmalar,

- **En iyi işçiyi seçme (Eİİ)** : Müsait olan işçiler/kaynaklar arasından beceri seviyesi en yüksek olan işçiyi seçmek için kullanılan algoritmadır.
- **En kötü işçiyi seçme (EKİ)** : Müsait olan işçiler/kaynaklar arasından beceri seviyesi en düşük olan işçiyi seçmek için kullanılan algoritmadır.
- **En ucuz işçiyi seçme (EUİ)** : Müsait olan işçiler/kaynaklar arasından maliyeti en düşük olan işçiyi seçmek için kullanılan algoritmadır.
- **En pahalı işçiyi seçme (EPI)** : Müsait olan işçiler/kaynaklar arasından maliyeti en yüksek olan işçiyi seçmek için kullanılan algoritmadır.

Bu işçi seçimlerinde algoritmanın deterministik olması çözüm kümemizi istediğimiz yönde daraltacaktır. Örneğin en iyi işçi algoritması uygulandığı varsayılırsa, atanan

işçiler çoğunlukla iyi işçiler arasından olacak ve âtil olan işçiler çalışmadıkları için çıkarılabilecektir. Ayrıca çözüm için algoritmada rastgele işçi ataması sonrası çıkacak kombinasyon sayısını da azaltacaktır. Bu da çözüm süresinin kısalmasına ve algoritmanın daha çabuk bitirme parametrelerine ulaşmasını sağlayacaktır. Örnek verilerin sonuçlarının çözümünde, bu algoritmalar ile de sonuçlar karşılaştırılacaktır.

Çizelge 3.17. İşçi seçim öncelik kuralları

Kısa Adı	Öncelik Kuralı
Eİİ	En iyi işçi ( $Min B_i$ )
EKİ	En kötü işçi ( $Max B_i$ )
EUİ	En ucuz işçi ( $Min M_i$ )
EPI	En pahalı işçi ( $Max M_i$ )

$B_i$ :  $i$  no'lu işçinin beceri seviyesi çarpanı

$M_i$ :  $i$  no'lu işçinin maaşı

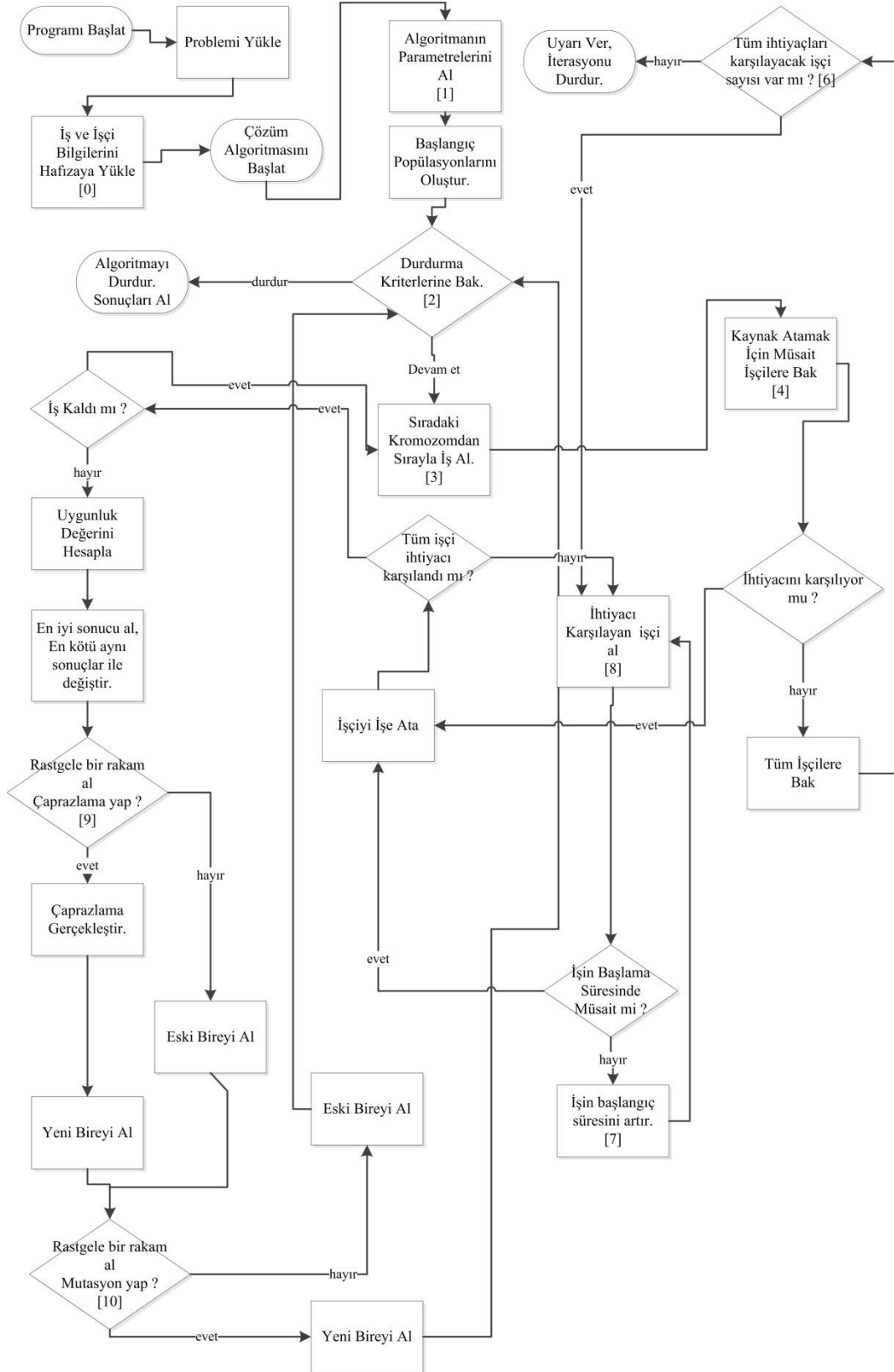
### 3.8. Uygulama

İşlem gücü bilgisayar ortamlarında değişiklikler gösterdiği için, algoritmalar kaç iterasyon sonunda iyi sonuca ulaştırdığı yönüyle karşılaştırılırlar. İşlem süresinin karşılaştırma yapılmasındaki bu zorluk nedeniyle yazılan programın çalıştırıldığı bilgisayar ortamının karşılaştırmada bir önemi yoktur. Bununla birlikte gerçek hayatta çözümün kabul edilebilir bir süre içerisinde verilmesi önemlidir. Bu nedenle problem çözümünün, iş ve işçi artışına bağlı olarak sürenin ne kadar arttığı da gösterilecektir. Bu sayede iş ve işçi sayısına göre programın çözüm süresi tahmin edilebilir.

Bu uygulamanın gerçekleştirilmesi için c#.net programı kullanılmıştır. Çözüm üreten programın akış çizelgesi Şekil 3.2'de gösterilmiştir.

Şekil 3.2'deki kutular üzerinde görülen tüm komutları bilgisayar gerçekleştirmektedir. Bölüm 3'ün başından buraya kadar anlatılan konuların akış şeması şeklinde gösterilmeye çalışılmıştır. Şekil üzerindeki bazı kritik noktalar izah edilecek ve konu özetlenmeye çalışılacaktır.

Şekil üzerinde 0 no'lu kutuda iş ve işçi bilgilerinin hafızaya yüklenmesi sağlanmaktadır. Hafızada iş ve işçi sayıları kadar uzunlukta sınıflar bulunmakta ve her iş ve işçiler için diğer bilgiler tutulmaktadır. Bu bilgiler işler için öncül işleri, süresi, kaynak ihtiyacı gibi bilgilerdir. İşçiler için ise, bildikleri alanlar ve bilgi dereceleri bulunmaktadır. 1 no'lu kutuda ise algoritmanın diğer girdileri olan, genetik algoritma parametreleri yüklenmektedir. Bunlar mutasyon oranı, iterasyon sayısı gibi parametrelerdir. 2 no'lu kutuda algoritmanın durdurma kriterlerinin sağlanıp sağlanmadığına bakılmaktadır. Bu kriterler, maksimum iterasyon sayısı, elle durdurma, son belli bir sayıda sonucun kendini yinelemesi durumlarıdır. 9 no'lu kutuda çaprazlama oranına göre kromozomun çaprazlanıp çaprazlanmayacağı, 10 no'lu kutuda ise mutasyon oranına göre mutasyon yapılıp yapılmayacağı denetlenmektedir. Çalışma ile ilgili bu parametrelere Bölüm 3.6' da değinilmiştir.



Şekil 3.2. Çalışmada kullanılan algoritmanın akış çizelgesi

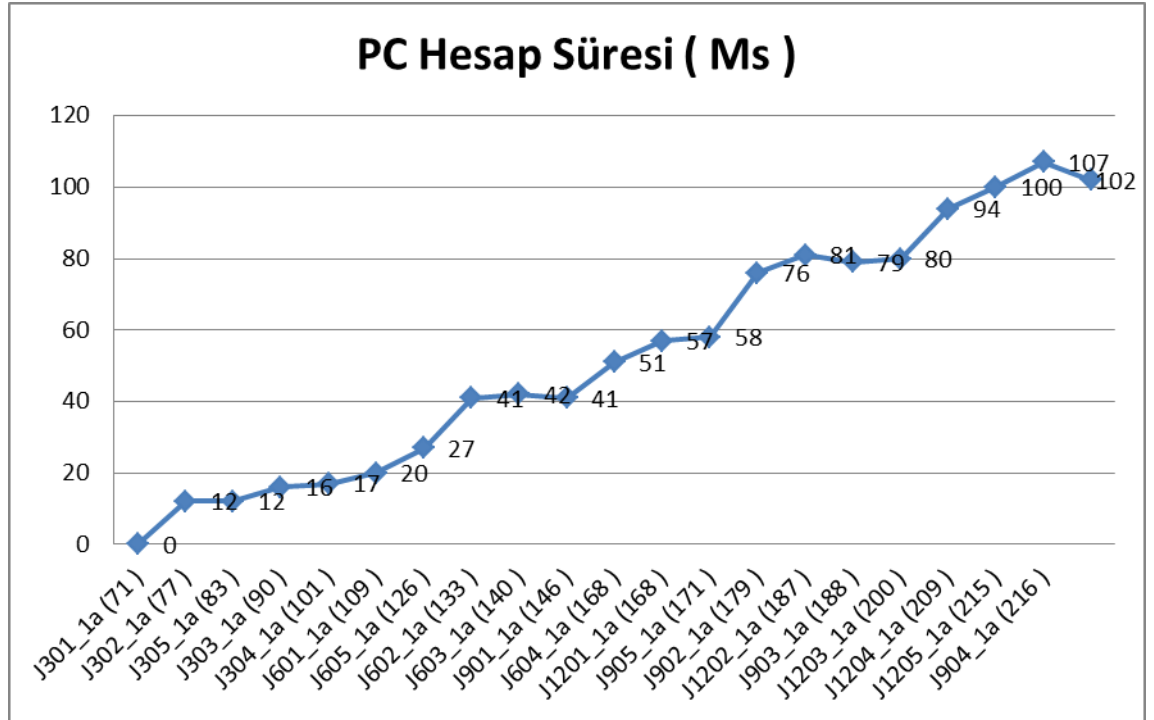
3 no'lu kutuda, mevcut kromozomun üzerindeki işler sırasıyla alınır. Alınan iş için, müsait kaynak olup olmadığına bakılır. Bu 4 no'lu kutudaki kaynak denetleme işleminde, işin ihtiyaç duyduğu beceri seviyesine ve bilgi alanına sahip müsait işçilere bakılır. Eğer müsait bir işçi yok ise, tüm işçilere bakılarak, bu işin yapılıp yapılmayacağı anlaşılır. 6 no'lu kutudaki bu denetleme ile algoritma durdurulmaya ya da devam ettirilmeye karar verir. Problemler içerisinde mevcut kaynaklardan daha fazla sayıda talepte bulunan bir iş yazılmış olabilir. Bu durumda algoritma işlem yapmayı durduracak ve ilgili girdilerin kontrol edilmesi için uyarı verecektir. Eğer tüm işçiler içerisinde işin yapılabilirdiği görülmüş ise, o işçilerin müsait olması beklenecektir. Bunun için atanacak işin başlangıç süresi artırılacaktır. Bunu gösteren 7 no'lu kutuda, işlem süresini artırmak için, tüm işçiler kontrol edilerek, en son müsait olan işçinin zamanı ilgili işin başlayabilme zamanı olarak güncellenecektir. Müsait olduktan sonra o işe atanacaklardır. Bu şekilde kromozomdaki tüm işler sıralanana kadar işlem devam edecektir.

### **3.7.1. Çözüm süresi**

Sonuçlar üretilirken hesap süresi iş ve işçi sayısına, iş başına talep edilen ayrı ayrı tanımlanmış beceri ihtiyaçları vb. tüm parametreler hesap süresini artırmaktadır. Buna göre problem başına hesap süresi tablosu Çizelge 3.17'de verilmiştir. Sonuçlar çalıştırılan bilgisayar ortamından bağımsız olarak, iş ve işçi sayısının problemin çözümüne yaptıkları etki incelenmiştir. Bunun nedeni, yeni bilgisayarların daha hızlı çalışmaları nedeniyle, hesaplanan işlem süreleri devamlı düşmektedir. Bunun yerine bağıl olarak iş ve işçi sayısına göre işlem süresinin ne kadar arttığı gösterilmesi hedeflenmiştir.

Çizelge 3.18. Bilgisayar hesap süresi tablosu

Problem No	Süre (Milisaniye)
J301_1a	12
J302_1a	12
J303_1a	17
J304_1a	20
J305_1a	16
J601_1a	27
J602_1a	42
J603_1a	41
J604_1a	57
J605_1a	41
J901_1a	51
J902_1a	81
J903_1a	80
J904_1a	102
J905_1a	76
J1201_1a	58
J1202_1a	79
J1203_1a	94
J1204_1a	100
J1205_1a	107



Şekil 3.3. Algoritmanın hesap süresi grafiği

Burada elde edilen süreler bilgisayarların işlemcilerinin farklılıkları nedeniyle, arka planda çalışan programlar vb. nedenlerle farklılık gösterebilir. Ancak bağıl olarak bir fikir vermektedir. Sonuçlar milisaniye olarak verilmiştir. Problem adlarının yanındaki parantez içerisinde işçi sayısı ve iş sayısı toplamlarıdır.

Dikkat edilirse iş ve işçi sayısı toplamı arttıkça problemin çözüm süresi de ona bağlı olarak üstel olarak artmaktadır. Burada elde edilen veriler ortalama değerlerdir. Çok kötü bir senaryoda iş sayısı iki kat arttığında işlem süresi yaklaşık dört kat artacaktır. Ancak kodlamada kullanılan özel teknikler nedeniyle bu ihtimal çok düşüktür.

Daha önce de değinildiği üzere, problem tiplerinde rakam arttıkça problem kompleksleşmektedir. Bu nedenle iş sayısı aynı kalsa bile yapılan hesapların artması nedeniyle işlem süresi daha uzun sürmektedir. Örnek veri oluşturma bölümünde “durum 2” olarak bahsedilen başlıkta anlatılan, çözüm kümesi için alternatif kümelerin azalmasına karşın işlem adetinin artması nedeniyle çözüm süresini kısaltmayacağı durumu burada görülmektedir.

En bariz olarak görülen diğer bir örnek, J904\_1 problemi ile J1205\_1 probleminin süresi neredeyse aynı olduğu görülmüştür. Bunun nedeni, işçi sayısı ve iş sayısının toplamlarının birbirlerine eşit olmasıdır. Ayrıca çözüm için işler yerleştirildiğinde çok sayıda işçi alternatifleri oluşmakta ve program çözüm için bu işçileri değerlendirmektedir. Bu sonuç veri oluşturma bölümünde durum 1 başlığında anlatılmıştır.

## 4. BULGULAR

Tüm araştırma sonuçları 10'ar kez çalıştırılarak ortalama sonuçlar üzerinden değerlendirme yapılmıştır. Ayrıca her denemede 1500 iterasyon ve bu iterasyonların tümünde bulunan sonuçların ortalaması üzerinden sonuçlar elde edilmiştir.

Araştırma sonuçları iki şekilde ele alınacaktır. Öncelikle problem çözümünde kullanılan deterministik algoritmalarından hangisinin daha iyi sonuç ürettiği ve birbirlerine karşı üstünlük ya da avantajları tartışılacaktır. Ardından orijinal RCPSP problemleri ile geliştirilerek özel verilerle elde edilen yeni problemlerin çözümünden elde edilen sonuçlar karşılaştırılacak ve buna göre yapılan geliştirmeler anlatılacaktır.

### 4.1. Deterministik algoritmaların karşılaştırılması

Deterministik algoritmalarından hangisinin daha iyi sonuç ürettiği ölçülmek üzere aşağıdaki sonuçlar karşılaştırılmıştır. Erken bitirme maliyetinin sıfır olarak alındığı sebebiyle karşılaştırma tamamen tamamlanma süresi üzerinden yapılacaktır. Tüm problemler üzerinde tüm algoritmalar uygulanacak, ancak listeyi kısa tutmak için, en iyi çözümü veren algoritma aşağıdaki tabloda listelenecektir.

Çizelge 4.1'de de görüldüğü üzere hesap yapılmadan önce de bekleneceği üzere, müsait olduğunda en iyi işçiyi kullanmak tüm sonuçlar arasında en iyi sonucu vermiştir.

Bu en iyi işçi aynı zamanda en pahalı işçi anlamına da gelebilir. Ancak o an gerekli alan için en iyi işçi olduğu anlamına gelmez. Örneğin  $R_1, R_2$  alanlarında  $B_1, B_2$  seviyesinde bilgili bir işçi maaşı usta seviyesindedir. Pahalı olarak bu işçi önceliğe sahip olacaktır. Ancak o an  $R_2$  alanında işçiye ihtiyaç duyulmakta ise algoritma pahalı fakat bilgi açısından standart seviyedeki işçiyi seçmiş olacaktır. Bu nedenle en pahalı işçi algoritması her zaman en iyi sonucu vermemekte ve en iyi işçi algoritması ile de aynı sonucu üretmemektedir.



Çizelge 4.1. Deterministik algoritmalarla oluşan bulunan en iyi uygulama sonuçları

Problem No	En iyi ortalama tamamlanma süresi	Ortalama tamamlanma süresi için en iyi algoritma	En iyi tamamlanma süresi	En iyi sonucu bulan algoritma
J301_1a	46,8	En İyi	40,52	En Pahalı
J302_1a	35,1	En İyi	28,58	En İyi
J303_1a	52,7	En İyi	33,12	En İyi
J304_1a	30,7	En İyi	23,05	En İyi
J305_1a	49,5	En İyi	44,46	En İyi
J601_1a	62,6	En İyi	42,11	En İyi
J602_1a	79,5	En İyi	69,41	En İyi
J603_1a	74,7	En İyi	55,63	En İyi
J604_1a	59,2	En İyi	42,64	En İyi
J605_1a	48,5	En İyi	41,09	En İyi
J901_1a	113,7	En İyi	89,71	En İyi
J902_1a	100	En İyi	87,05	En İyi
J903_1a	75,1	En İyi	53,63	En İyi
J904_1a	98,3	En İyi	75,68	En İyi
J905_1a	213,9	En Pahalı	171,61	En Pahalı
J1201_1a	181,2	En İyi	166,02	En İyi
J1202_1a	207,6	En İyi	184,88	En İyi
J1203_1a	92,5	En İyi	73,8	En İyi
J1204_1a	110	En İyi	95,43	En İyi
J1205_1a	94,4	En İyi	73,9	En İyi

En ucuz işçi seçimi süreleri oldukça uzatmıştır. Böyle olacağı da öngörülebilir. Ancak önce sıralanan işlerin ucuz işçilerce yapılıp, sıradaki işlerde daha pahalı işçi alternatiflerinin kullanılma ihtimali nedeniyle daha iyiye götürebilme durumu da bulunmaktadır. En iyi işçi ve en pahalı işçi arasındaki ilişki aynı şekilde en ucuz işçi ve en kötü işçi arasında da geçerlidir. Ancak burada benzerlik daha yüksektir.

Sonuçlar üretilirken program bazı algoritmalar için “mevcut kaynaklarla bu problem çözülemez.” uyarısı vermektedir. Bunun nedeni şu şekilde açıklanabilir. Bir iş için sırasıyla gereken beceriler  $R_1, B_1$  ve  $R_2, B_3$  olsun. 2 kaynak istesin ve müsait 3 kaynak olsun. Bu kaynak özellikleri de, 1. işçi için  $R_1, B_1$ , 2. işçi  $R_1, B_1, R_2, B_3$ , 3. işçi de  $R_1, B_1$  olsun. Eğer sıralamada 2 numaralı işçiyi önce seçip  $R_1$  alanı için atama yaparsa program kalan talep için kaynak bulamayacak ve bu hatayı verecektir. Çözüm için başka bir algoritma tercih edilmelidir.

## 4.2. Sonuçların analizi

Bu çalışma ile kaynak kısıtlı proje problemlerine esnek kaynak atama yapılabileceği gösterilmiştir. Bunun sağlanması için ek parametrelere ihtiyaç duyulmuştur. Bu parametreler, her işin ihtiyaç duyduğu bilgi alanı, her işçinin bilgi alanları ve bu alanlardaki beceri seviyeleridir. Bu sayede gerçek hayatı daha iyi temsil edebilen örnek problemler üretilmiştir.

Oluşturulan örnek problemlerin temel karakteristiklerinin farklı olması nedeniyle çok farklı verilerin çözümü denenebilmiştir. Problemler arası temel karakteristik farkları şunlardır.

- İş sayısının, işçi sayısına oranı,
- Farklı beceri seviyelerindeki işçilerin işe atanma oranları,
- Aynı anda talep edilen işçilerin farklı bilgi alanlarındaki dağılımıdır.

İş sayısı ve işçi sayısının farklı oranlarda olması çözülecek problemler için en kritik parametredir. Örneğin, 90 iş barındıran J901\_1a problemi için 56 işçi verilirken, J904\_1a numaralı problemde aynı iş sayısı için 126 işçi verilmiştir. Bu sayede çok sayıda alternatif çözümler oluşabilecektir. İşçi sayısının çok olması doğru orantılı olarak projenin daha çabuk yapılması anlamına gelmemektedir. Bunun nedeni bir işe atanabilecek işçi sayısı bellidir. Tüm işçilerin aynı anda kullanımları da mümkün değildir. Çünkü öncüllük bağlantıları nedeniyle, bir işin tamamlanmasını beklemek zorundadır. O iş tamamlandıktan sonra da işi yapan işçiler atanabilir hale geldiği için bazı işçiler işçi seçim algoritması gereği daha az çalışacak veya daha çok tercih edileceklerdir. Bu da çok sayıda çalışmayan işçi anlamına gelebilir. Böyle bir ihtimal nedeniyle minimum işçi sayısını bulan bir optimizasyon yapılabilir.

Farklı beceri seviyesindeki işçilerin işe atanma oranları, işin teslim süresi ile doğru orantılıdır. Usta atamaları ile işin erken bitirilebilmesi mümkündür. Bu nedenle kesinlikle teslim süresi karşılanmış olacaktır. Ancak bu durumda çok fazla kalifiye

eleman çalıştırmaya neden olacaktır. Maliyetlerin artması bir yana, bu kadar kalifiye eleman bulmak da başlı başına bir problemdir, hatta mümkün değildir denilebilir.

Bir iş için talep edilen bilgi seviyeleri de problemlerde farklılıklar göstermektedir. Bu talepler neticesinde işçi seçiminde daha spesifik işçi seçilip alternatif işçi ihtimalleri azalırken, bilgisayar hesaplaması için ekstra işleme neden olmaktadır. Bu noktaya çözüm süresi başlığında değinilmiştir. Örnek problemlerde genel olarak 1\_1 ile biten problemlerde aynı anda sadece tek bir bilgi alanından işçi talebi yapılırken, 5\_1 ile biten problemlerde ise aynı anda birden fazla bilgi alanından işçi talebi yapılmaktadır.

Geliştirilen algoritma ile farklı nitelikteki problemler için çözümler üretilebilmiştir. Böylece algoritmanın bu kısıtları içerebilecek tüm gerçek hayattaki farklı ölçeklerdeki problemleri de çözebileceği gösterilmiştir.

Sonuçların analizi aşamasında öncelikle esnek işgücü atamasının yapılabildiği ve algoritmanın anlamlı çözümler üretebildiği gösterilmiştir. Gerçek hayatta da işletmelerde çok farklı yeteneğe sahip işçi veya kaynakların bulunması nedeniyle bu yeni örnek kümesi standart kaynak kısıtlı proje çizelgeleme problemlerine göre gerçek hayatı daha iyi temsil edebilmektedir. Bu çalışma ile çok farklı özelliklerdeki kaynakların beraberce etkin bir şekilde kullanılması mümkün olmuştur.

Ancak yine de bu çalışmanın gerçek hayata olacak katkısının daha iyi gözlenebilmesi için, normal ve sabit işçi atamasının , esnek işçi atamadan ne kadar iyi olup olmadığı karşılaştırılması yapılan çalışmanın faydası anlamında bir fikir verebilecektir. Bu nedenle 3.3.1 no'lu bölümde bahsedildiği şekilde ek parametreler ile geliştirilen problemlerin, orjinal kaynak kısıtlı proje çizelgeleme problemleri ile elde edilen sonuçlar arasında bir karşılaştırma yapılması anlamlı olacaktır. Bunun için normal kaynak kısıtlı proje çizelgeleme problemlerindeki kaynakların normal beceri seviyesinde ve tek bir alanı bildikleri varsayılmıştır.

Buna göre bulunan sonuçlar, normal kaynak kısıtlı proje çizelgeleme problemlerinde bulunan sonuçlara çok yakın çıkmıştır. Sezgisel bir algoritma kullanılması nedeniyle

bulunan sonuçların en iyi sonuç olduğu söylenememektedir. Bu sebeple literatürde şu zamana kadar ilgili problemde bulunmuş en iyi sonuç bitirilme süresi olarak alınarak karşılaştırma yapılmıştır. Buna göre elde edilen maliyetler çizelge 4.2’de verilmiştir.

Çizelge 4.2. Orijinal problemler için maliyetler

Problem No	İşçi Sayısı / Kaynaklar (Sırasıyla)	Orijinal Prblm. İşçi Toplam Maliyet	Bilinen En İyi Sonuç – Teslim Tarihi	Gecikme maliyeti	Toplam Maliyet
J301_1	12-13-4-12	41*100=4100	43-38 = 5	26*5=130	4230
J302_1	9-11-11-16	47*100=4700	38-34 = 4	5*4=20	4720
J303_1	15-14-15-16	60*100=6000	72-72 = 0	3*0=0	6000
J304_1	10-22-26-13	71*100=7100	49-49 = 0	21*0=0	7100
J305_1	13-13-12-15	57*100=5700	53-41 = 8	24*8=192	5892
J601_1	13-11-12-13	49*100=4900	77-77 = 0	50*0=0	4900
J602_1	21-16-15-21	73*100=7300	65-65 = 0	49*0=0	7300
J603_1	20-21-18-21	70*100=7000	60-60 = 0	13*0=0	7000
J604_1	33-23-26-26	108*100=10800	84-84 = 0	5*0=0	10800
J605_1	15-18-17-16	66*100=6600	76-59 = 17	9*17=153	6753
J901_1	12-14-17-13	56*100=5600	73-67 = 6	9*6=54	5654
J902_1	15-26-26-22	89*100=8900	96-96 = 0	88*0=0	8900
J903_1	16-26-31-25	98*100=9800	81-81 = 0	27*0=0	9800
J904_1	33-32-40-21	126*100=12600	93-93 = 0	7*0=0	12600
J905_1	18-21-26-16	81*100=8100	78-66 =12	31*12=372	8472
J1201_1	14-12-13-9	48*100=4800	105-99 = 6	44*6=264	5064
J1202_1	16-19-15-17	67*100=6700	87-70 = 17	110*17=1870	8570
J1203_1	23-21-17-19	80*100=8000	80-79 = 1	22*1=22	8022
J1204_1	15-27-22-25	89*100=8900	74-70 = 4	75*4=300	9200
J1205_1	26-20-21-28	95*100=9500	92-92 = 0	32*0=0	9500

Orjinal problem için işçi maliyetleri ve diğer maliyetler Çizelge 4.2’de gösterilmiştir. Yeni üretilen problemlerle, orijinal problemler arasında en temel fark, iki işletmenin, işçi atamaları için tutundukları tavidir. Örneğin orjinal problem için işletme, tüm işçileri için standart işçi atamaları yapmıştır. İşçiler de tek bir alan bilmekte ve farklılıkları bulunmadığı kabul edilmiştir. Yeni problemler için ise işçi seviyeleri farklılıklar gösteren bir atama planlanmıştır. Çalışmaya özgü üretilen bu yeni problemler ile orijinal problemler arasındaki ilk fark burada ortaya çıkmaktadır. İşletme çoğunlukla işçi maliyetleri açısından daha avantajlı olmuştur. Gerçek hayatta da işçilerin tümü aynı maaşı almadıkları için yeni problem kümesi çok daha gerçekçi

olduğu bir daha görülmektedir. Yeni üretilen problemlerde birçok ek parametreler bulunmaktadır. En önemli özellik olarak içerisinde, farklı bilgi ve beceri seviyelerinde işçiler bulundurmaktadır. Bu nedenle işçi maliyetleri de değişiklik göstermektedir.

Çizelge 4.3. Yeni problemler için maliyetler

Problem No	İşçi Sayısı	İşçi Maliyeti	Bulunan En İyi Sonuç – Teslim Tarihi	Gecikme maliyeti	Toplam Maliyet
J301_1a	41	4180	40,52-38 = 2,52	<b>26*2,52=65,52</b>	4245,52
J302_1a	47	4940	28,58-34 = -5,42	0*-5,42=0	4940
J303_1a	60	6220	33,12-72 = -38,88	0*-38,88=0	6220
J304_1a	71	6940	23,05-49 = 0	0*-25,95=0	6940
J305_1a	57	4920	44,46-41 = 8	<b>24*3,46=83,04</b>	5003,04
J601_1a	49	5190	42,11-77 = -34,89	0*-34,89=0	5190
J602_1a	73	7320	69,41-65 = 4,41	<b>49*4,41=216,09</b>	7536,09
J603_1a	70	7650	55,63-60 = -4,37	0*-4,37=0	7650
J604_1a	108	10520	42,64-84 = -41,36	0*-41,36=0	10520
J605_1a	66	6650	41,09-59 = -17,91	0*-17,91=0	6650
J901_1a	56	5600	89,71-67 = 22,71	<b>9*22,71=204,39</b>	5804,39
J902_1a	89	8640	87,05-96 = -8,95	0*-8,95=0	8640
J903_1a	98	9950	53,63-81 = -27,37	0*-27,37=0	9950
J904_1a	126	12040	75,68-93 = -17,32	0*-17,32=0	12040
J905_1a	81	7530	171,61-66 = 105,61	<b>31*105,61=3273,91</b>	10803,91
J1201_1a	48	4920	141,23-99 = 67,02	<b>44*67,02=2948,88</b>	7868,88
J1202_1a	67	6570	184,88-70 = 114,88	<b>110*114,88=12636,8</b>	19206,8
J1203_1a	80	8280	73,8-79 = -5,2	0*-5,2=0	8280
J1204_1a	89	8780	95,43-70 = 25,43	<b>75*25,43=1907,25</b>	10687,25
J1205_1a	95	10080	73,9-92 = -18,1	0*-18,1=0	10080

Orjinal problemler ile yeni problemler arası farklılıklara bakıldığında, işçi maliyet farkının sifıra yakın olması işçi dağılımının varsayılan problem için standart işçi kullanımı planına yakın yapıldığı fikrini verebilir. Ancak pahalı ve ucuz işçi kombinasyonu da kullanılmış olabilir. Buna mukabil, işçi maliyeti farkında negatif sayı ne kadar büyük ise, o kadar usta seviyesinde işçi çalıştırıldığı söylenebilir. Bu nedenle çözüm programı daha çalıştırılmadan oluşturulacak çözümlerin orijinal problem için bilinen çözümlerden çok iyi olacağı öngörülebilmektedir. Tam tersi olarak da bu fark pozitif yönde ne kadar büyük ise o kadar stajyer çalıştığı söylenebilir. Buradan da işlerin süresinin uzun olacağı ve ilk problemin çözüm süresinden daha uzun süreceği öngörülebilmektedir. Bu iki

problem arasındaki plan farklarından oluşan erken bitirme ya da geç bitirme maliyetleri ile işçi maliyetlerindeki görünen mutlak avantajları azalacaktır.

Yapılan çalışma sonucu bulunan sonuçlar incelendiğinde, teslim sürelerinde bazı işlerin çok daha kötü olduğu görülmektedir. Buna göre işçi talepleri uygun yapılmış ya da yapılmamış olduğu anlaşılabilir. Örneğin, j1201-2\_1a, j905\_1a numaralı problemlerde çok sayıda stajyer talebi bulunması nedeniyle teslim sürelerini çok fazla geçmiş durumdadırlar. J303\_1a gibi problemlerde ise usta talebi çok olduğu için iş süreleri çok kısalmıştır. Teslim süresinin çok altında bir sonuç üretilmiştir. Bu planlama minimum işçi bulunurken, işçi seçimi üzerinde doğrudan etkilidir. Diğer problemlerde de iyi bir işçi ataması planlanmadığı anlaşılmaktadır.

Sonuçlardan da anlaşılacağı üzere, verilerin nasıl elde edildiği ve işlere kimlerin atanacağı planlamasının çok iyi yapılması gerekmektedir.

Bu sonuçların işletme refleksi ile hareket edilirse daha da iyileştirilebileceği görülmektedir. Bu amaçla iki adet optimizasyon gerçekleştirilecektir. Birincisi minimum işçi sayısını hesaplamak, diğeri ise teslim süresini karşılayabilmek olacaktır. Teslim süresinin karşılanabilmesi için fason işçi alımı yapılacaktır.

### **4.3. Sonuçların iyileştirilmesi**

Program çıktılarına bakılırsa, problemler maliyet ve teslim süresini karşılamaları yönüyle optimizasyon yapılabilirler. Bulunan en iyi sonuçlara bakıldığında toplam 8 adet işin geç kaldığı görülmektedir. Bu da işletmeye ekstra bir maliyete ve imajında zedelenmeye neden olmaktadır.

Optimizasyonun bir sonu yoktur. Her zaman daha iyi bir geliştirme mümkün olmaktadır. Ancak bu çıkan sonuçlar içerisinde iki adet optimizasyon hedeflenmektedir. Birinci optimizasyon maliyetin düşürülmesi, ikinci optimizasyon ise teslim süresinin altında projenin tamamlanmasını sağlamaktır.

Daha önce de bahsedildiği gibi, öncüllük ilişkileri nedeniyle tüm işler aynı anda işlenememektedir. Belli bir zaman aralığına düşen işlerin sayısı sınırlıdır. Bu nedenle işçilerin birçoğu çalışmadan beklemektedirler. İlk iyileştirme olarak bir işletme refleksi ile hareket edilerek, öncelikle minimum işçi sayısı bulunmaya çalışılacaktır. Daha sonra minimum işçi sayısı ile yapılan çözümlerde teslim tarihi gecikirse, fason işçi alımı yapılacaktır. Bu noktada sadece optimizasyona ihtiyaç duyulan yani geciken problemler kullanılacaktır.

Çizelge 4.4. Minimum işçi sayısı sonrası maliyetler

Prb. No	Çıkarılan işçi sayısı	Yeni İşçi Maliyeti	Yeni İşçi Maliyet Farkı (JX_1a-JX1b)	(Bulunan Tamamlanma Süresi- Teslim Süresi) * Gecikme Maliyeti	1.Optimizasyon Sonrası Toplam Maliyet Farkı (JX_1a - JX1b)
J301_1b	12	3320	860	(41-38)*26 = 130	795,52
J302_1b	12	3980	960	(31,76-34)*0 = 0	960
J303_1b	15	5180	1040	(33,12-72)*0 = 0	1040
J304_1b	22	5490	1450	(27,49 - 49) * 0 = 0	1450
J305_1b	7	4570	350	(45,36 - 41) *24 = 104,64	328,4
J601_1b	7	4490	700	(66,08 - 77) *0 = 0	700
J602_1b	9	6820	500	(72,03 -65)*49=344,47	371,62
J603_1b	29	5600	2050	(69,39-60)*13 = 122,07	1927,93
J604_1b	52	6290	4230	(42,64 - 84)*0 = 0	4230
J605_1b	10	6150	500	(41,09 - 59) * 0 = 0	500
J901_1b	1	5400	200	(89,71 - 67)*9 = 204,39	200
J902_1b	21	7060	1580	(95,4 - 96) * 0 = 0	1580
J903_1b	20	7730	2220	(69,66 - 81) * 0 = 0	2220
J904_1b	50	8010	4030	(82,21 - 93) *0 = 0	4030
J905_1b	5	7080	450	(179,95 - 66) * 31 = 3532,45	191,46
J1201_1b	1	4990	-70	(165,63 - 99) *44 = 2931,72	-52,84
J1202_1b	4	6370	200	(187,88 - 70) *110 = 12966,8	-130
J1203_1b	19	6810	1470	(76,54 - 79) * 0 = 0	1470
J1204_1b	24	7290	1490	(87,41 - 70) *75 = 1305,75	2091,5
J1205_1b	40	6480	3600	(74,85 - 92) * 0 = 0	3600

Her şeyden önce şu unutulmamalıdır ki, çözüm yöntemimiz bir sezgisel algoritma olup, bulunan sonuçlardan daha iyi bir sonuç bulma ihtimali her zaman vardır.

İşçi sayısının azaltılması üretilen sonuçların daha düşük aralıklarda olmasına, yani alternatif ihtimallerin düşmesine sebep olmaktadır. Hesaplama süresi de doğal olarak azalmaktadır.

Çıkarılan işçilerin daha çok kötü işçiler olması beklenebilir. Ancak işletmenin ihtiyacı doğrultusunda örnek veriler oluşturulacağı için, her zaman kötü işçinin çıkarılması söz konusu olmamaktadır. Örneğin bu çalışmada temel alınan problemlerde problem cinsine göre en iyi işçi sınıfında da çok kişi işten çıkarılmıştır. Bunun temel nedeni işçi sayısının fazlalığı ve o sınıfa ihtiyacın az olmasıdır. Örneğin bu çalışmada kullanılan J903\_1 numaralı problemde tüm alanları en iyi seviyede bilen işçiler çıkarılmıştır. J302\_1 numaralı problemde ise çıkarılan tüm işçiler stajyer olanlardır. Bununla birlikte teslim süresinden daha düşük sürede bitiş zamanının elde edildiği problemlerde stajyer işçi sayısı az ve usta işçi sayısı fazladır. Bu da öngörülebileceği üzere, kalifiye işçinin bir işletme için ne kadar önemli olduğunun bir başka göstergesidir.

Sonuçlara bakılacak olursa, maliyet açısından sadece bir problem minimum işçi sayısı konusunda faydası olmamıştır. Bunun nedeni çok basittir. İş sayısı çok ve işçi sayısı ise orantısız olarak azdır. Bu nedenle sadece bir işçi çıkarılabilmektedir. Gecikme süresi maliyetinin de çok yüksek olması nedeniyle kalifiye olmayan her eleman için büyük bir ceza söz konusudur. Projenin en başında işleri hangi seviyede işçilerin yapacağı seçilirken çok sayıda stajyere iş ataması planlanmıştır. Bunun nedeni işletmede çok sayıda stajyer bulunması ya da ucuza işçi çalıştırmanın daha uygun olacağı varsayımı olabilir. Maliyetin düşürülmesi için projelendirmenin daha mantıklı yapılması zorunludur. Bu aşamadan sonra ancak teslim süresine yetiştirilmesi için fason işçi kullanıldığında maliyetin ne olacağına bakılabilir. Ancak daha iyi bir maliyet elde etmek mümkün değildir.

İşçi sayılarının azaltılması teslim süresi yönünde olumlu bir etkiye sahip değildir. İşletmelerin ayakta kalabilmesi için teslim sürelerine uyduklarını gösterebilmeleri gerekmektedir. Sonuçlara bakılır ise, sekiz problem teslim süresini geciktirmiştir. Teslim süresinin karşılanamadığı işler için bu işleri yetiştirmek adına, fason olarak üretirmek gerekmektedir.

Bu nedenle yapılan maliyet geliştirmeleri ve kaynak atamalarından sonra teslim süresini karşılamak için fason iş yaptırılmıştır. Yani fason işçi alındığı varsayılmıştır. Fason işçi sayısı, teslim süresini sağlayabilen ilk işçi sayısı kadardır. Bu işçi sayısının



bulunabilmesi için birden başlayarak en ideal olanına kadar artırılarak denenmiştir. Belki algoritmada, daha önceki bir aşamada daha az fason işçi alımı ile problem çözüm verebilirdi. Ancak sezgisel algoritma nedeniyle bu çalışma yapımı aşamasında bu değerler elde edilmiştir. Çizelge 4.5'te alınan fason işçi sayıları gösterilmiştir.

Çizelge 4.5. II. Optimizasyon sonrası tamamlanma süresi

Prb. No	Fason İşçi sayısı	Yeni İşçi Maliyeti	(Bulunan Tamamlanma Süresi- Teslim Süresi) * Gecikme Maliyeti)
J301_1c	1	3460	( <b>37,05</b> -38)*0 = 0
J305_1c	1	4710	( <b>37,22</b> – 41) *0 = 0
J602_1c	2	7100	( <b>59,80</b> -65)*0=0
J603_1c	3	6020	( <b>59,60</b> -60)*0 = 0
J901_1c	6	6240	( <b>62,64</b> – 67)*0 = 0
J905_1c	11	8620	( <b>66,08</b> - 66) * 31 = 3532,45
J1201_1c	8	6110	( <b>99,67</b> – 99) *44 = 2931,72
J1202_1c	15	8470	( <b>66,10</b> – 70) *0 = 0
J1204_1c	10	8690	( <b>62,97</b> – 70) *0 = 0

Yeniden temas etmek gerekirse, çözümler sezgisel bir algoritma yardımıyla yapıldığı için belki daha az fason işçi gerektirmesine rağmen bu çözüm üretilmemiş olabilir. Ancak fason işçi eklemenin üretilen çözümlerin istenilen hedef doğrultusunda yönlendirdiği çok açık görülmektedir. Ayrıca işçi atamalarına bakıldığında eklenen tüm fason işçilerin en fazla çalışan oldukları görülmektedir.

İşletmenin yetiştiremediği işleri dışarıda yaptıracaklarını yani fason işçi eklenmesi durumunda mevcut işçilerin daha az çalıştığı ve yeniden optimizasyon yapılması gerektiği düşünülebilir. Ancak işletmeler işi asgari seviyede de olsa yapabilecek işçi sayısını sağlamak zorundadırlar. Fasona güvenerek işçi çıkarımı yapmazlar.

Problemlerdeki çeşitlilik (işletmelerdeki farklılıklar) nedeniyle tüm problemlerin aynı mantık ile optimizasyonu zaten mümkün değildir. Ancak yapılan iyileştirmelerin ne kadar faydalı olduğunu göstermeleri açısından iyi birer örnektirler.

Sonuçları analiz etmek gerekirse, J901\_1 numaralı problem için maliyet daha kötüye gitmiştir. Bunun nedeni işçi sayısının az olmasıdır. J1202\_1 numaralı problemde ise 15

tane fason işçi alınmasına rağmen toplam maliyette düşmüştür. Bunun nedeni gecikme maliyetinin çok yüksek olmasıdır.

Diğer tüm problemler işletmenin teslim süresine uyduğu imajı nedeniyle ve maliyetlerdeki iyileştirmeler sebebiyle katlanılabilir maliyetlerdir. J905\_1, J1201\_1 ve J1202\_1 numaralı işler için gecikme süresi maliyeti alınacak işçiden düşük olduğundan ve erken bitirmenin de bir faydası olmadığından bulunan sonuç en ideal sonuçtur ve bu şekilde bırakılmışlardır.

Çizelge 4.6’da bilinen en iyi süre ve maliyete karşın yapılan geliştirmeler yüzde olarak görülmektedir. Koyu işaretli olanlar sonucun daha kötü olduğunu göstermektedir.

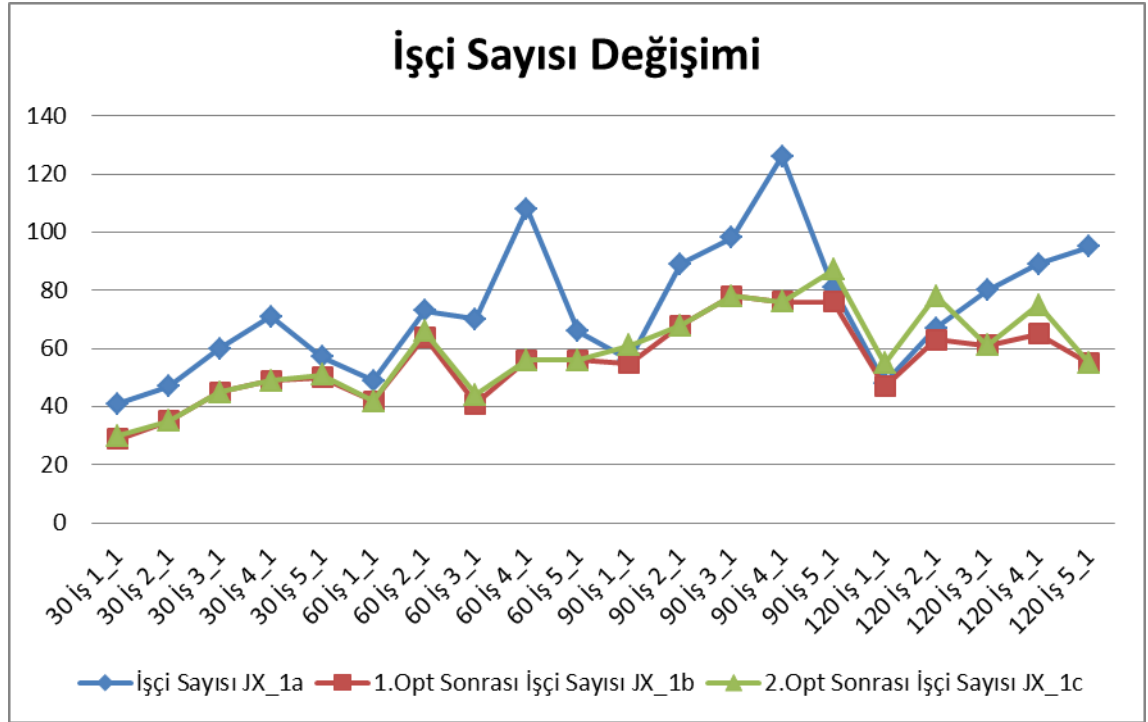
Çizelge 4.6. Optimizasyonlar sonrası iyileştirme oranları

Prb. No	2.Opt Sonrası	1.Opt sonrası Maliyet İyileştirme	2.Opt. Sonrası Maliyet iyileştirme
	Süre İyileştirme Oranı		
30 İş 1 1	0,90	0,81	0,81
30 İş 2 1	1,00	0,81	0,81
30 İş 3 1	1,00	0,83	0,83
30 İş 4 1	1,00	0,79	0,79
30 İş 5 1	0,82	0,93	0,94
60 İş 1 1	1,00	0,87	0,87
60 İş 2 1	0,83	0,95	0,94
60 İş 3 1	0,86	0,75	0,79
60 İş 4 1	1,00	0,60	0,60
60 İş 5 1	1,00	0,92	0,92
90 İş 1 1	0,70	0,97	<b>1,08</b>
90 İş 2 1	1,00	0,82	0,82
90 İş 3 1	1,00	0,78	0,78
90 İş 4 1	1,00	0,67	0,67
90 İş 5 1	0,37	0,98	0,80
120 İş 1 1	0,60	1,00	0,78
120 İş 2 1	0,38	1,00	0,44
120 İş 3 1	1,00	0,82	0,82
120 İş 4 1	0,72	0,80	0,81
120 İş 5 1	1,00	0,64	0,64

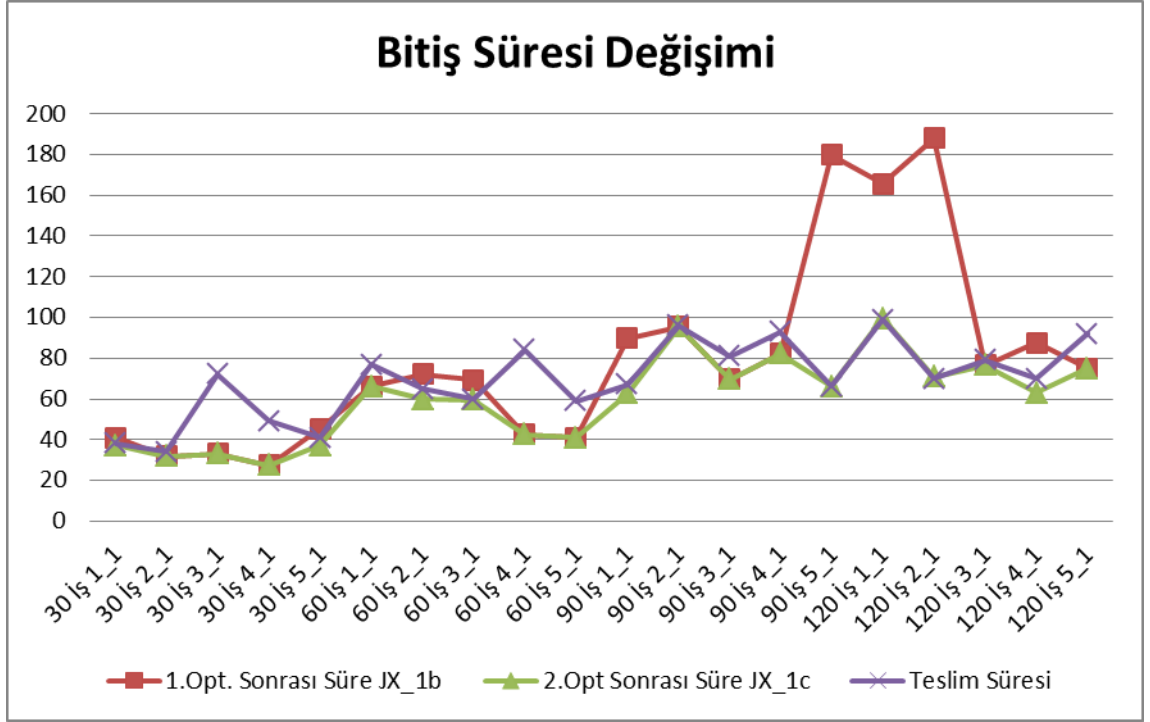
Sürelerin iyileştirilmesi zaten temel hedefimiz olduğu için, doğal olarak teslim süresini aşan işler için iyileştirme yapılmış ve sonuç olarak hiçbir kötü sonuç elde edilmemiştir.

Maliyetler için bakıldığında, 2. Optimizasyon sonrası bir adet işin maliyeti artmıştır. Bunun nedeni, işçi sayısının az olması nedeniyle, işçi çıkarımı az yapılabilmiş ve yeni alınan fason işçiler doğrudan işçi maliyetini artırmıştır. Ancak gecikme maliyetinin çok yüksek olması nedeniyle, gecikme maliyetini düşüren her fason işçi, gecikme maliyetinden gelen iyileştirmeye karşılık gelmiştir. Bu nedenle işçi alımından kaynaklanan çok yüksek bir maliyet, gecikme maliyetini düşürdüğü için, maliyetini oradan karşılamıştır.

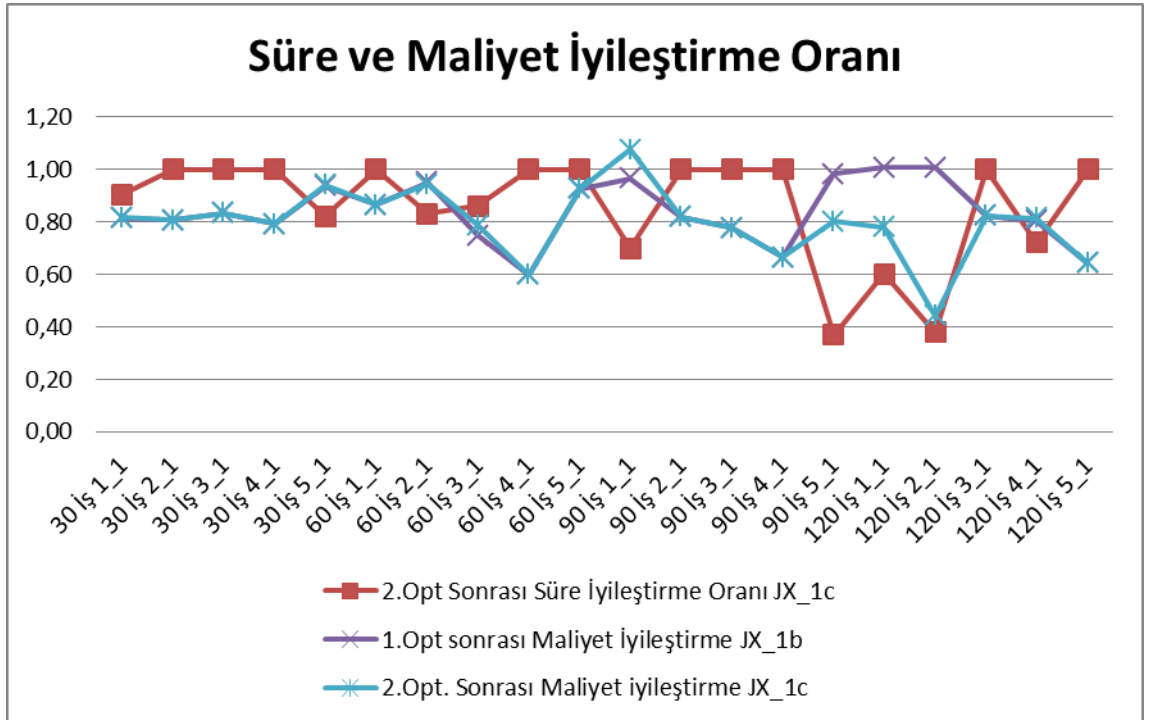
Sonuçların tamamı grafiksel olarak ise aşağıda gösterilmiştir.



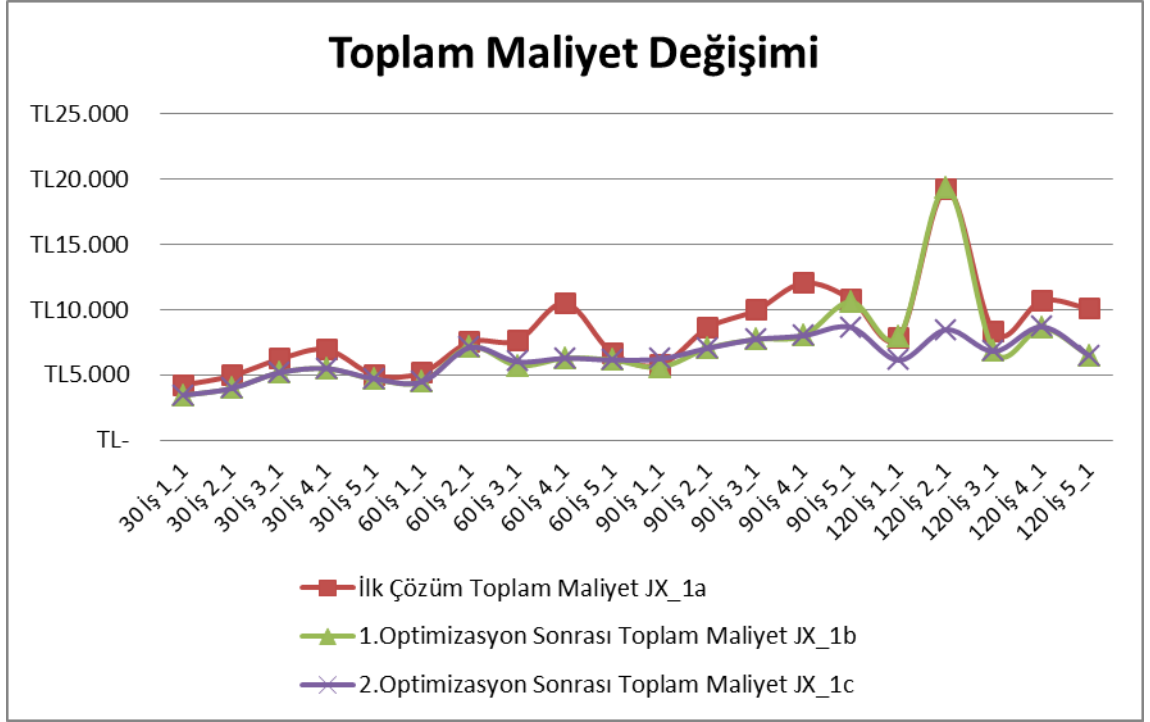
Şekil 4.1. İşçi sayısı değişimi



Şekil 4.2. Bitiş süresi değişimi



Şekil 4.3. Süre ve Maliyet iyileştirme oranı



Şekil 4.4. Toplam maliyet değişimi

Grafiklerden de açıkça görüldüğü üzere fason işçilerin alınarak yapıldığı ikinci optimizasyon sonrasında maliyetler ve süreler düşürülmüştür. Böylece teslim süresi hedefi ve maliyet hedefleri aynı zamanda tutturulabilmiştir.

Buradaki işçi sayıları bir projede kullanılan toplam değerdir. Örneğin bir işçinin hangi zaman aralıklarında kullanıldığına değil, proje süresi sonuna kadar kaç adet işçi kullanıldığına bakılmıştır. Aynı şekilde fason işçiler için de aynı kural geçerlidir. Yaklaşım, projede kullanılan temel maliyetler üzerinedir. Bu maliyeti oluşturan temel girdilerin, zaman aralıklarına göre atanması bu çalışmanın amacı ve konusu içerisinde değildir.

## 5. TARTIŞMA VE SONUÇ

Bu çalışmanın amacı, gerçek hayatı daha iyi temsil eden kaynak kısıtlı proje çizelgeleme problemlerini çözmek olmuştur. Gerçek hayatta kaynakların veya işçilerin yerleri benzer işler için kaydırılabilmektedir. Ayrıca her kaynağın farklı seviyelerde iş görme kabiliyetleri vardır. Bu yeteneklerin aynı anda uygulanması işlem süresi üzerine sinerjik bir etkide bulunmaktadır. Literatürde olmayan böyle bir yaklaşıma bu çalışma ile çözüm üretilebildiği gösterilmiştir.

Çıkan sonuçların analizinde de görüleceği üzere, bir işletme için en önemli şeylerden bir tanesi kalifiye eleman çalıştırmaktır. Ancak her işletmenin ilk önceliği bu yönde olmayabilir. Örneğin ürün arzı değişken olmayıp, teslim tarihi konusunda daha rahat çalışabilecek işletmeler için daha ucuz işçilik kullanılabilir ve kabul edilebilir bir ürün üretebilir. Gerekirse her iş başına bir iyi bilen atama ile işin kalitesinden de vazgeçilmemiş olabilir. Kullanılan problemlerde bu tarz etkiler açık olarak ortaya çıkmıştır. Bu nedenle erişilen sonuçlar çok anlamlıdır.

Her ne kadar bu araştırmada bahsi geçen kaynaklar işçi olarak ifade edilse de, kuramsal bilgiler bölümünde değinildiği üzere, işlem süresi ayarlanabilir,özdeş olmayan birer kaynak makine de olabilirler. Uzman işçi gibi gelişmiş makine kaynağı da elektrik masrafları, amortismanı vb. nedenlerden dolayı masraflıdır. Böylece kalifiye elemanın yüksek maaşı gibi, böyle bir makinenin de maliyeti yüksek olacaktır. Bu nedenle yapılan çalışma makine kaynakları için de kullanılabilir.

Bu çalışmayı daha ileri götürmek gerekirse, ekip atama ve hemşire atama bölümlerinde de değinildiği gibi çalışma saatleri, sosyal güvenlik sınırlamaları ve vardiyalar ile işgücü atanarak ve maliyet hesapları yapılarak daha gerçek hayata yakın çözümler üretilebilir. Maliyetler çalıştığı saat bazında hesaplanabilir ve bir iş için en fazla atanacak işçi sayısı, en az atanabilecek işçi sayısı verilebilir. İş süresi belli bir sürenin üstüne çıkamayacağı veya inemeyeceği gibi kısıtlar da verilebilir.

## KAYNAKÇA

- Blazewicz, J., Ecker, K.H., Pesch, E., Schmidt, G., Weglarz, J. (2007).** Motivations for Deterministic Scheduling Problems. *Handbook on Scheduling From Theory to Applications*. Springer.
- Agyei, M. F., & YEO, E. O. (1998).** Optimising engineering problems using genetic algorithms. *Engineering Computations*, 268-280.
- Alcaraz, J. , Marato C. (2001),** A Roboust Genetic Algorithm for Resource Allocation in Project Scheduling, *Annals of Operation Research* 102, 82-109
- Ansel, H. (1998).** Esnek üretimde İşçiler ve sendikalar (post-fordizm’de üretim esnekleşirken İşçiye neler oluyor). *Birleşik Metal-İş Sendikası*.
- Balin, S., Başgil, H., & Alcan, P. (2008).** Non-identical parallel machine scheduling using genetic algorithm. *Computers & Industrial Engineering*.
- Barrios, A., Ballestin, F., & Valls, V. (2009).** A double genetic algorithm for the MRCPSP/max. *Article In Press*.
- Bianco, L. P., Dell’Olmo, & Speranza, M. (1998).** Heuristics for Multimode Scheduling Problems with Dedicated Resources. *European Journal of Operational Research*, 107,260-271.
- Blazewicz, J., Cellary, W., Slowinski, R., & Weglarz, J. (1986).** Scheduling under Resource Constraints Deterministic Models. *Baltzer, Basel*.
- Blazewicz, J., Lenstra, J., & Rinnooy Kan, A. (1983).** Scheduling subject to resource constraints: Classification and complexity. *Discrete Applied Mathematics*, 11-22.
- Böttcher, J., Drexl, A., & Kolisch, R. (1996).** A Branch-and-Bound Procedure for Project Scheduling with Partially Renewable Resource Constraints,. *Proceedings of the Fifth Workshop on Project Management and Scheduling*, 48-51.
- Burgess, A., & Killebrew, J. (1962).** Variation in Activity Level on a Cyclic Arrow Diagram. *Industrial Engineering*, March-April, 76-83.
- Cheang, B., Li, H., Lim, A., & Rodrigues, B. (2003).** Nurse rostering problems—a bibliographic survey. *European Journal of Operational Research* 151, 447-460.

- Chen, M., & Zalzal, A. M. (1997).** A genetic approach to motion planning of redundant mobile manipulator systems considering safety and configuration. *Journal of Robotic Systems*, 529-544.
- Clark, W. (1954).** The Gantt Chart. *Pitman*.
- Demeulemeester, E. L., & Herroelen, W. S. (2002).** *Project Scheduling A Research Handbook* (s. 1-3). New York, Boston, Dordrecht, London, Moscow: Kluwer Academic Publishers.
- Dr. Scott J. Amos, P. (2007).** *Skills & Knowledge of Cost Engineering*. 209 Prairie Avenue, Suite 100, Morgantown, WV 26501 USA: AACE Education Fund.
- Drexl, A. (1997).** Local Search Methods for Project Scheduling under Partially Renewable Resource Constraints. *INFORMS San Diego Spring Meeting*, (s. 4-7).
- Elmaghraby, S. (1977).** Activity Networks – Project Planning and Control by Network Models. *John Wiley & Sons Inc*.
- Elmaghraby, S., & Kamburowski, J. (1989).** The Analysis of Activity Networks under Generalized Precedence Relations (GPR). Parts I and II. *OR Reports No. 231 and 232*. NC. State University, Raleigh, NC.
- Ernst, A., Jiang, H., Krishnamoorthy, M., & Sier, D. (2004).** An Annotated Bibliography of Personnel Scheduling and Rostering. *Annals of Operations Research* 127, 21-144.
- Ernst, A., Jiang, H., Krishnamoorthy, M., & Sier, D. (2004).** Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research* 153, 3–27.
- Fowler, J. W., Wirojanagud, P., & Gel, E. S. (2007).** Heuristics for workforce planning with worker differences. (190).
- Glover, F. (1989).** Tabu Search. *ORSA Journal on Computing*, 190-206.
- Glover, F. (1990).** Tabu Search: A Tutorial. *Programming Integer Algorithms, Heuristics* (s. 74-94). The Institute of Management Science.
- Goldratt, E. (1997).** *Critical Chain*. Great Barrington, U.S.A.: The North River Press.
- Gopalakrishnan, B., & Johnson, E. L. (2005).** Airline Crew Scheduling: State-of-The-Art. (140).
- Hongyi, S., & Wang G. (2003).** Parallel machine earliness and tardiness scheduling with proportional weights. *Computers & Operations Research*.



- Hajdu, M. (1997).** Network Scheduling Techniques for Construction Project Management. *Kluwer Academic Publishers*.
- Hartman, S., & Briskorn, D. (2010).** A survey of variants and extensions of the resource-constrained project scheduling problem. *207*.
- Hartmann, S. (1997).** A Competitive Genetic Algorithm for Resource-Constrained Project Scheduling. *John Wiley & Sons, Inc.*
- Hartmann, S. (1998).** A Competitive Genetic Algorithm for Resource-Constrained Project Scheduling.
- Herroelen, W., Dommelen, P. V., & Demeulemeester, E. (1997).** Project Network Models with Discounted Cash Flows - A Guided Tour through Recent Developments. *European Journal of Operational Research*, 97-121.
- Herroelen, W., Reyck, B. D., & Demeulemeester, E. (1998).** A Classification Scheme for Project Scheduling. *EURO XVI Conference*. Brussels, Belgium.
- J. Buddhakulsomsiri, D. K. (2006).** Properties of multi-mode resource-constrained project scheduling problems with resource vacations and activity splitting. *European Journal of Operational Research*, 279-295.
- J. R. Jackson. (1955).** Scheduling a production line to minimize maximum tardiness. *Management Sci. Res. Project, UCLA*, Research Report 43.
- Jozefowska, J., Mika, M., Rozycki, R., Waligora, G., & Wezglarz, J. (2001).** Simulated annealing for multi-mode resource-constrained project scheduling problem. *102*, 137-155.
- Kirkpatrick, S., Gelatt, C., & Vecchi, M. (1983).** Optimization by simulated annealing. *Science*, 671-679.
- Kolisch, R., & Hartman, S. (1999).** Heuristic algorithms for solving the resource-constrained project scheduling problem: Classification and computational analysis. *Handbook on recent advances in project scheduling*.
- Koza, J. R. (1992).** *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press.
- Lab, S. C. (2004).** 16. Resource-constrained Project Scheduling. *Graduate School of Information, Production and Systems*. WASEDA UNIVERSITY, IPS.
- Lucic, P., & Teodorovic, D. (2007).** Metaheuristics approach to the aircrew rostering problem. *Springer Science, Annual Operations Research*, 311-338.

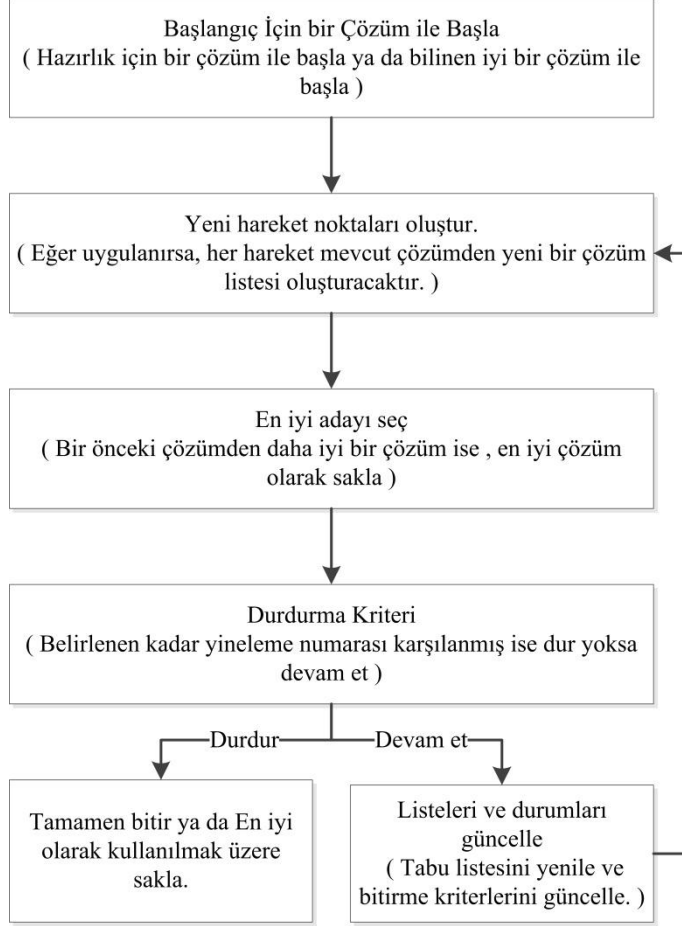
- Lumsden, P. (1968).** Line of Balance Method. *Pergamon Press, Exeter.*
- Moder, J. C. (1983).** Project Management with CPM, PERT and Precedence Diagramming. Van Nostrand Reinhold Company.
- Moder, J., Phillips, C., & Davis, E. (1983).** *Project Management with CPM, PERT and Precedence Diagramming, Van Nostrand Reinhold Company, Third Edition.*
- Mori, M., & Tseng, C. C. (1996).** A genetic algorithm for multi-mode resource constrained project scheduling problem. *100.*
- Örmeci, L., Aksen, D., Erdem, A., Salman, F., & Savaş, S. (2008).** Çağrı merkezlerinin iş gücü planlaması. (*106MI75*). İstanbul: Tübitak.
- Peteghem, V. V., & Vanhoucke, M. (2009).** A genetic algorithm for the preemptive and non-preemptive multi-mode resource-constrained project scheduling problem. (201), 409-418.
- Pinedo, L.M. (2005).** Local Search. *Planning and scheduling in manufacturing services* (s. 422). Springer Science Business Media, Inc.
- Pinedo, L.M. (2008).** Part I, *Theory, Algorithms, and Systems* (Third Edition). Springer Science Business Media, Inc.
- Anonim, PSPLIB.** *Project scheduling problem library - PSPLIB.* 2010 tarihinde Project scheduling problem library - PSPLIB: <http://129.187.106.231/psplib> adresinden alındı
- Reyck, B. D., & Herroelen, W. (1999).** The multi-mode resource-constrained project scheduling problem with generalized precedence relations. *European Journal of Operational Research*, 538±556.
- Schirmer, A., & Drexl, A. (1996).** Partially Renewable Resources – A Generalisation of Resource-Constrained Project Scheduling. *IFORS Triennial Meeting*, (s. July 8-12.). Vancouver.
- Selamoğlu, A. (1998).** Yönetim ve üretim anlayışında değişim, japon modelinin artan etkinliği ve İnsan unsuru. *ÇİMENTO İŞVEREN DERGİSİ*, 8-25.
- Slack, N., Chambers, S., Harland, C., Harrison, A., & R., J. (1998).** *Operations Management.* Pitman Publishing.
- Tiwari, V., Patterson, J. H., & Mabert, V. A. (2009).** Scheduling projects with heterogeneous resources to meet time and quality objectives. (193).
- Trilling, L., Guinet, A., & Le Magny, D. (2006).** Nurse scheduling using integer linear programming and constraint programming. *IFAC.*

- Ulusoy, G. (1996).** Proje Planlama Yönelimleri. *Tübitak Yayınları*.
- Valls, V., Pe´rez, A., & Quintanilla, S. (2009).** Skilled workforce scheduling in Service Centres. *European Journal of Operational Research* 193, 791–804.
- Anonim (2006),** Waseda University. 16. Resource Constrained Project Scheduling. *Graduate School of Information, Production and Systems*. IPS.
- Yang, B., Genunes, J., & J.O. O'Brien, W. (2001).** *Resource-Constrained Project Scheduling: Past Work and New Directions*. Department of Industrial and Systems Engineering, University of Florida.
- Yu, G., & Thengvall, B. (2009).** Airline Optimization. C. A. Floudas, & P. M. Pardalos, *Encyclopedia of Optimization, Second Edition*. Springer.
- Zeghal, F., & Minoux, M. (2006).** Modeling and solving a Crew Assignment Problem in air transportation. (175).

**EKLER**

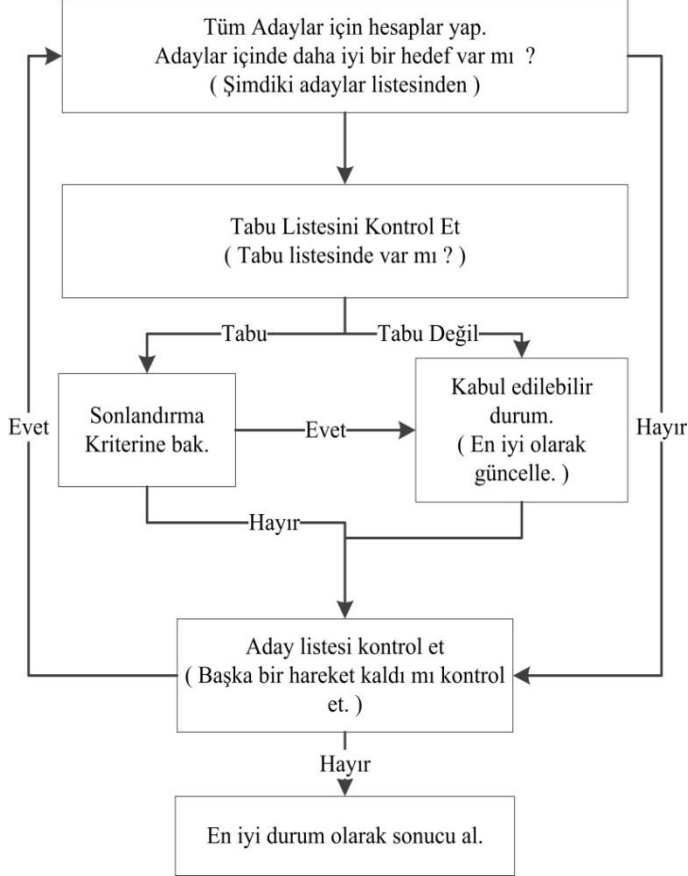
**EK-1**

## Tabu Arama Algoritması (Glover, 1990)



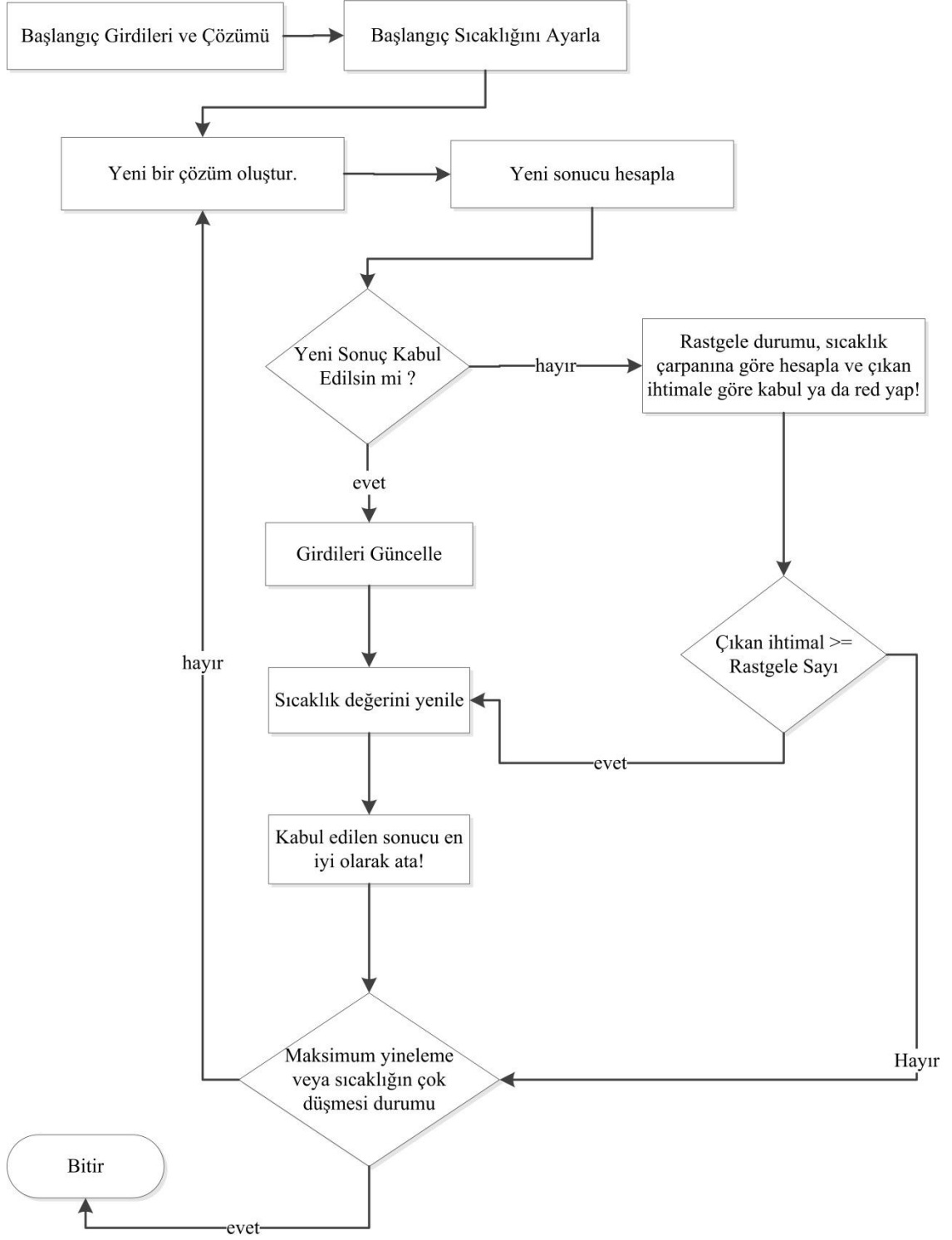
## EK-2

## Tabu Algoritması En İyiyi Seçme (Glover, 1990)



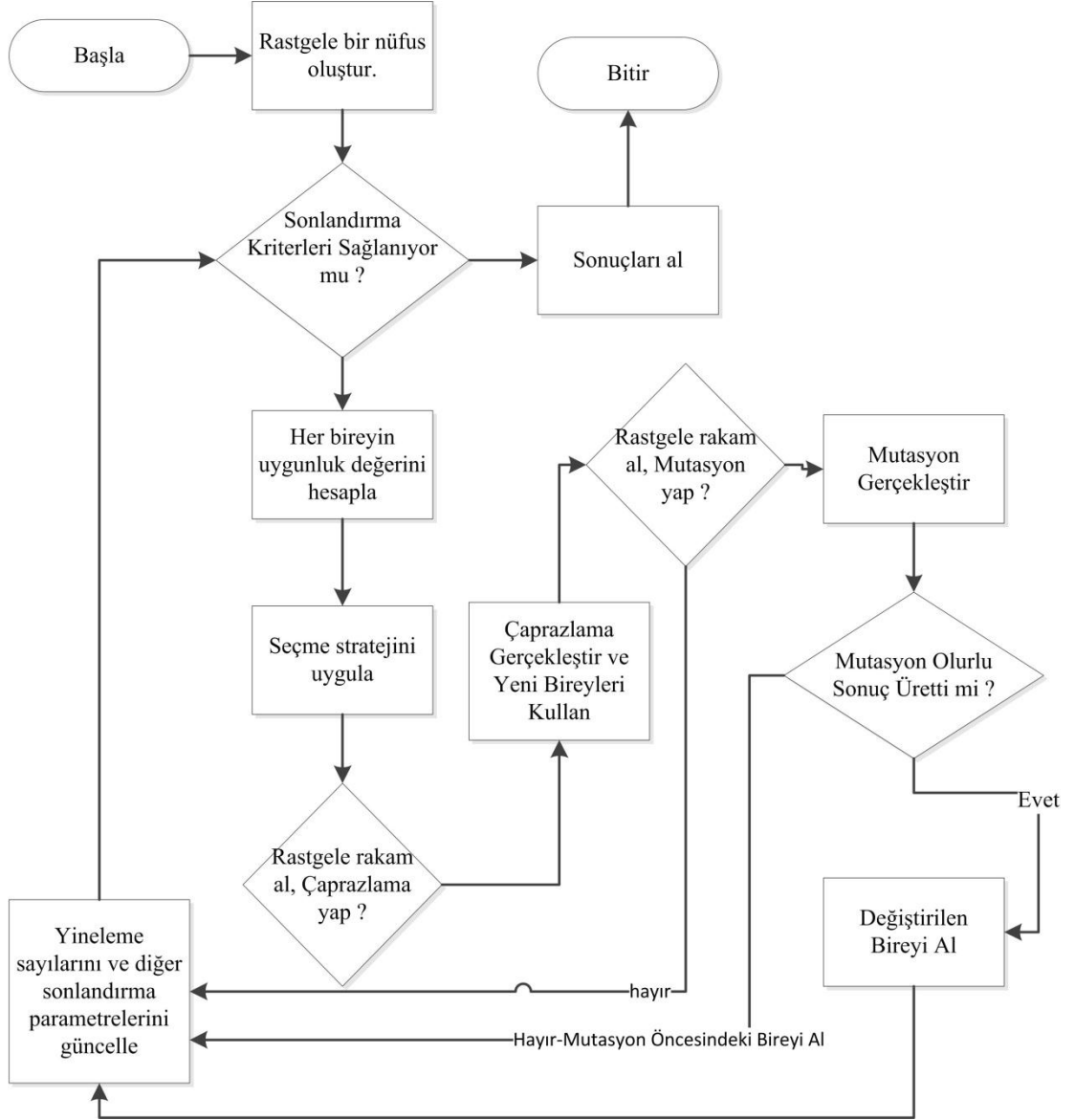
## EK-3

## Tavlama benzetimi algoritması



## EK-4

## Genetik algoritma akışı





**EK-5****Örnek Veri Oluşturan Program**

Örnek verilerin oluşturulması için gerekli tüm altyapı, verilerin oluşturulması başlığı altında açıklanmıştır. Bu açıklamalara uygun olarak bir “veri oluşturma programı” geliştirilmiştir. Böylece, literatürdeki tüm RCPSP problemleri örnek problem haline getirilebilmektedir. İstenirse bağımsız problemler de geliştirilebilir.

Problemin oluşturulması için literatürde geçen şekilde formatla düzenlenmiş problem sisteme yüklenmelidir. Bu dosyalar sm uzantılıdır. Problemin yüklenmesi için kaç adet beceri seviyesi istenirse, o kadar adet beceri seviyelerinin çarpanları sisteme girilecektir. Ardından aynı sıralama ile bu işçi tiplerine ait taban fiyatlar verilmelidir. Son olarak erken bitirme maliyetini pozitif ya da negatif verilebilir. Bu veriler girildikten sonra problem yükle butonuna basılarak sm uzantılı problem sisteme yüklenmeye hazırdır.

Bu program ile üretilen verilerin herhangi bir zamanda her iş için gerekli beceri ve alana sahip işçi bulabilme anlamlılığını/ihtimalini koruyarak geliştirilmiştir. Böylece elde edilen kaynaklar kesinlikle bu işleri yapabilecek seviyededir.

Programda örnekler üzerinde değişiklik yapabilmeye izin verilmektedir. Bu izin sonucu programda kaynaklar ve işler üzerinde bir anlamsızlık oluşursa, çözecek program bu tutarsızlık sebebiyle çalışmayacaktır. Bu noktaya ileride yeniden değinilecektir.

Şekil EK5-1. Örnek Veri Oluşturma Modülü

Bu girilen verilere göre oluşturulan bilgiler yeni bir pencerede oluşturulmaktadır. Rastgele alınan değerlerin kök başlangıç rakamları (tohum değerleri) problemde tercih edilen rakamdır. Bu verilere göre oluşturulan değerler, istenildiği gibi değiştirilebilir.

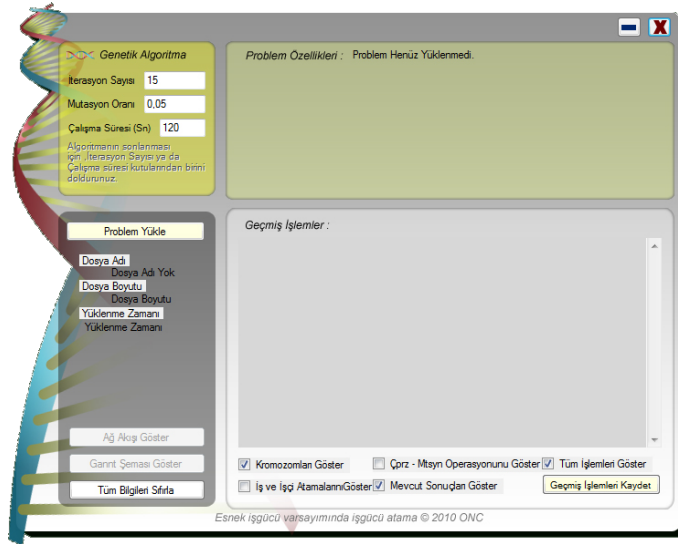
Şekil EK5-2. Örnek Veri Oluşturma Modülü 2. Pencere

Daha sonra tamam tuşuna basılarak yeni veri tabanımız oluşturulmuş olmaktadır. Örnek veriyi kaydet diyerek, çözüm için problem saklanır. EK-9'de verilen orijinal problem, bu program aracılığı ile örnek veriye çevrilmiştir ve EK-10 olarak gösterilmiştir.

## EK-6

### Çözüm Üreten Programın Kullanılması

Program yüklendikten sonra açıldığında aşağıdaki gibi bir pencere karşımıza gelmektedir.



Şekil EK6-1. Çözme programı ilk görüntü

Program açıldıktan sonra genetik algoritma için gerekli maksimum iterasyon sayısı, mutasyon oranı veya en fazla kaç saniye çalışacağını gireceğiniz bir alan bulunmaktadır. Bu alanlar daha farklı sonuçlar üretebilmek adına seçimli olarak bırakılmıştır.

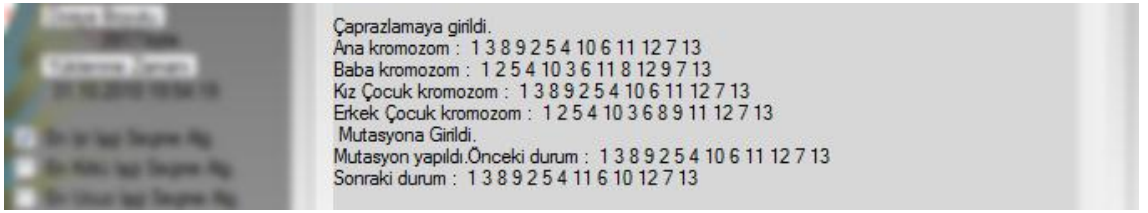
Programın aşağısında, kromozomları göster, “çprz-mtsln operasyonunu göster”, “iş ve işçi atamaları göster”, “mevcut sonuçları göster” seçenekleri bulunmaktadır. “Tüm işlemler” seçeneği işaretlendiğinde, programda yapılan işlemler listelenecektir. Diğer seçenekler algoritmanın akışı ile ilgili bilgilerdir. Aşağıda verilen şekillerde ilgili bilgi haricindeki yerler bulanık hale getirilmiştir. Böylece altında bilgisi verilen yer öne çıkarılmıştır.

Kromozomları göster seçeneği işlem yapılan sıralamayı göstermektedir. Örnek verdiği çıktı Şekil EK6-2’teki gibidir.



Şekil EK6-2. Kromozomları göster seçeneği işaretli

Mutasyon operasyonunu göster seçeneği tıklandığı durumda, çaprazlama işlemi Şekil EK6-3'de gösterilmektedir.



Şekil EK6-3. Çprz-Mtsyn operasyonu göster seçeneği işaretli

İş ve işçi atamalarını göster seçeneği işaretlendiğinde, işçi ve iş atamalarının listesi görülmektedir. Örnek çıktı Çizelge EK6-1'de gösterilmiştir.

Çıktı formatı şu şekildedir:

İşçi No : x

[Başlama süresi-Bitiş süresi]-[Başlama süresi-Bitiş süresi]-.....

Bu başlama ve bitiş içerisinde hangi işte çalıştığı çıktının aşağısında yer almaktadır.

Buna göre aynı aralığa düşen işte çalıştığı anlaşılmaktadır.

İş Numarası :x

Başlangıç süresi : x,xx-x,xx İş Süresi x,xx Hesap Süresi x,xx

Atanan beceriler: Beceri-X(Çarpanı) x adet,.....

Atanan İşçiler: x

## Çizelge EK6-1. İş ve işçi atamaları göster seçeneği işaretli

İşçi No: 1 [0,00: 1,64] - [4,20: 6,25] - [11,96: 14,01] - [14,01: 16,51] - [20,26: 22,26] - [22,26: 23,79] -
İşçi No: 2
İşçi No: 3 [0,00: 1,64] - [1,64: 4,20] - [4,20: 6,25] - [8,89: 11,96] - [11,96: 14,01] - [14,01: 16,51] - [16,51: 20,26] - [20,26: 22,26] - [22,26: 23,79] -
İşçi No: 4 [0,00: 1,64] - [1,64: 4,20] - [4,20: 6,25] - [8,89: 11,96] - [11,96: 14,01] - [16,51: 18,43] - [22,26: 23,79] -
İşçi No: 5
İşçi No: 6 [4,20: 8,89] - [11,96: 14,01] - [16,51: 20,26] -
İşçi No: 7 [0,00: 1,64] - [4,20: 6,25] - [8,89: 11,96] - [11,96: 14,01] - [16,51: 20,26] - [22,26: 23,79] -
İşçi No: 8 [4,20: 8,89] - [14,01: 16,51] - [20,26: 22,26] -
İşçi No: 9 [4,20: 8,89] - [14,01: 16,51] - [20,26: 22,26] - [22,26: 23,79] -
İşçi No: 10 [4,20: 8,89] - [14,01: 16,51] -
İşçi No: 11 [22,26: 23,79] -
İşçi No: 12 [0,00: 1,64] - [1,64: 4,20] - [4,20: 6,25] - [8,89: 11,96] - [16,51: 18,43] - [22,26: 23,79] -
İşçi No: 13 [1,64: 4,20] - [4,20: 8,89] - [8,89: 11,96] - [11,96: 14,01] - [16,51: 18,43] - [22,26: 23,79] -
İşçi No: 14 [0,00: 1,64] - [1,64: 4,20] - [4,20: 6,25] - [8,89: 11,96] - [11,96: 14,01] - [16,51: 20,26] - [20,26: 22,26] - [22,26: 23,79] -
İşçi No: 15 [4,20: 8,89] - [14,01: 16,51] - [16,51: 20,26] -
İşçi No: 16 [0,00: 1,64] - [11,96: 14,01] - [14,01: 16,51] - [22,26: 23,79] -
İşçi No: 17 [0,00: 1,64] - [14,01: 16,51] - [22,26: 23,79] -
İşçi No: 18
İş Numarası: 1 Başlangıç-Bitiş: 0,00-0,00 İş Süresi: 0,00 Hesap Süresi: 0,00 Atanan beceriler: Beceri-1: (0,8) 0 adet Beceri-2: (1) 0 adet Beceri-3: (1,25) 0 adet Atanan işçiler:
İş Numarası: 2 Başlangıç-Bitiş: 4,20-8,89 İş Süresi: 3,00 Hesap Süresi: 4,69 Atanan beceriler: Beceri-1: (0,8) 1 adet Beceri-2: (1) 2 adet Beceri-3: (1,25) 3 adet Atanan işçiler: 13 , 8 , 15 , 9 , 6 , 10 ,
İş Numarası: 3 Başlangıç-Bitiş: 0,00-1,64 İş Süresi: 5,00 Hesap Süresi: 1,64 Atanan beceriler: Beceri-1: (0,8) 5 adet Beceri-2: (1) 3 adet Beceri-3: (1,25) 0 adet Atanan işçiler: 14 , 1 , 12 , 16 , 3 , 4 , 7 , 17 ,
İş Numarası: 4 Başlangıç-Bitiş: 8,89-11,96 İş Süresi: 6,00 Hesap Süresi: 3,07 Atanan beceriler: Beceri-1: (0,8) 3 adet Beceri-2: (1) 3 adet Beceri-3: (1,25) 0 adet Atanan işçiler: 14 , 3 , 13 , 4 , 7 , 12 ,
İş Numarası: 5 Başlangıç-Bitiş: 14,01-16,51 İş Süresi: 2,00 Hesap Süresi: 2,50

<p>Atanan beceriler: Beceri-1: (0,8) 2 adet Beceri-2: (1) 3 adet Beceri-3: (1,25) 3 adet  Atanan işçiler: 9 , 17 , 8 , 3 , 1 , 16 , 15 , 10 ,  İş Numarası: 6  Başlangıç-Bitiş: 16,51-20,26 İş Süresi: 3,00 Hesap Süresi: 3,75  Atanan beceriler: Beceri-1: (0,8) 1 adet Beceri-2: (1) 2 adet Beceri-3: (1,25) 2 adet  Atanan işçiler: 14 , 3 , 7 , 6 , 15 ,  İş Numarası: 7  Başlangıç-Bitiş: 16,51-18,43 İş Süresi: 3,00 Hesap Süresi: 1,92  Atanan beceriler: Beceri-1: (0,8) 2 adet Beceri-2: (1) 1 adet Beceri-3: (1,25) 0 adet  Atanan işçiler: 13 , 4 , 12 ,  İş Numarası: 8  Başlangıç-Bitiş: 1,64-4,20 İş Süresi: 4,00 Hesap Süresi: 2,56  Atanan beceriler: Beceri-1: (0,8) 2 adet Beceri-2: (1) 3 adet Beceri-3: (1,25) 0 adet  Atanan işçiler: 14 , 3 , 13 , 4 , 12 ,  İş Numarası: 9  Başlangıç-Bitiş: 4,20-6,25 İş Süresi: 5,00 Hesap Süresi: 2,05  Atanan beceriler: Beceri-1: (0,8) 4 adet Beceri-2: (1) 2 adet Beceri-3: (1,25) 0 adet  Atanan işçiler: 14 , 1 , 3 , 4 , 7 , 12 ,  İş Numarası: 10  Başlangıç-Bitiş: 11,96-14,01 İş Süresi: 4,00 Hesap Süresi: 2,05  Atanan beceriler: Beceri-1: (0,8) 4 adet Beceri-2: (1) 3 adet Beceri-3: (1,25) 1 adet  Atanan işçiler: 14 , 3 , 1 , 13 , 4 , 7 , 6 , 16 ,  İş Numarası: 11  Başlangıç-Bitiş: 20,26-22,26 İş Süresi: 2,00 Hesap Süresi: 2,00  Atanan beceriler: Beceri-1: (0,8) 1 adet Beceri-2: (1) 3 adet Beceri-3: (1,25) 1 adet  Atanan işçiler: 9 , 14 , 3 , 8 , 1 ,  İş Numarası: 12  Başlangıç-Bitiş: 22,26-23,79 İş Süresi: 3,00 Hesap Süresi: 1,54  Atanan beceriler: Beceri-1: (0,8) 5 adet Beceri-2: (1) 4 adet Beceri-3: (1,25) 2 adet  Atanan işçiler: 9 , 14 , 3 , 13 , 1 , 12 , 16 , 17 , 4 , 7 , 11 ,  İş Numarası: 13  Başlangıç-Bitiş: 23,79-23,79 İş Süresi: 0,00 Hesap Süresi: 0,00  Atanan beceriler: Beceri-1: (0,8) 0 adet Beceri-2: (1) 0 adet Beceri-3: (1,25) 0 adet  Atanan işçiler:</p>
--

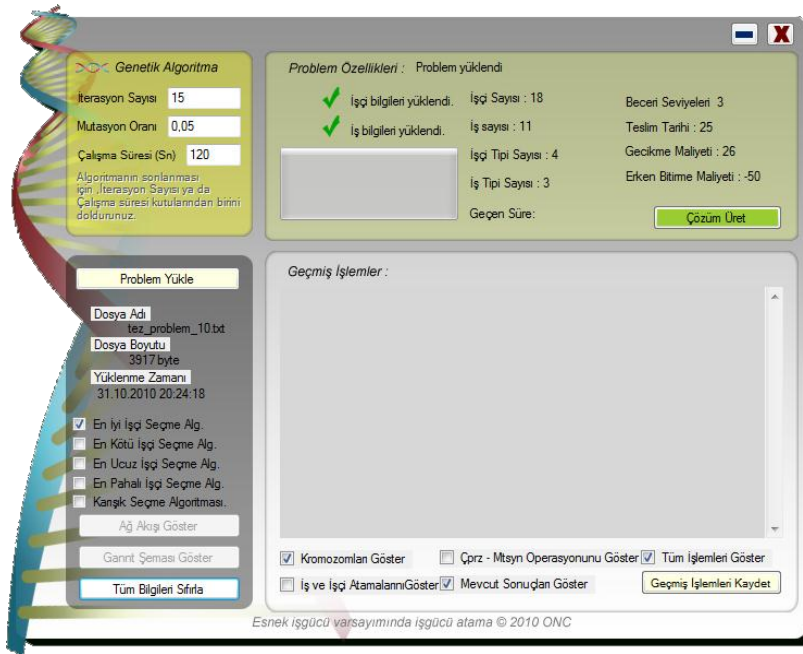
İşlerin yapılış süresi ve o işlere hangi işçilerin atandığı bilgisi, bir işin ne kadar sürdüğü bilgisi bu seçenek işaretli ise mevcut sonuçlar olarak görülmektedir. Eğer işçi çalışmamış ise hiçbir zaman aralığı görülmeyecektir. Bu özelliği sayesinde, en iyi çözüm için çalışmayan işçiler çıkarılarak minimum işçi sayısı bulunabilmektedir.

Son seçenek ise, bulunan en iyi sonucu göster seçeneğidir. Bu seçenek işaretli olduğu zaman, mevcut bulunan en iyi bitiş süresini, maliyeti ve bu işlem için pc hesaplama süresini göstermektedir. Örnek olarak, Şekil EK6-4'deki çıktı görülmektedir.



Şekil EK6-4. En iyi sonucu göster seçeneği işaretli

Problem yükledikten sonra aşağıdaki ekran görülmektedir. Burada probleme ait bilgiler gösterilmekte ve çözüm için tercih edilecek alternatif işçi seçim algoritmaları bulunmaktadır. Varsayılan olarak “en iyi işçi seçme algoritması” seçilmiştir. Daha önce de bahsedildiği üzere, “karışık seçme algoritması” seçildiği takdirde program çok daha iyi sonuçlar üretebilir ancak bunun hangi algoritmalar ile üretildiği bilinemeyeceği için, “iş ve işçi atamalarını göster” seçenekleri ile takip edilmesi uygundur.



Şekil EK6-5. Problem yükledikten sonraki ekran görüntüsü

Tüm bilgileri sıfırla seçeneği işaretlendiği takdirde, çözüm ve bilgi gösteren yazı ekranı, bilinen en iyi sonuçlar silinmektedir. Böylece istenilirse programa aynı problem yeniden yüklenmeden yeni algoritmalar denenebilmektedir.



Ayrıca iterasyon sayısına göre işlemin ne kadar kaldığını gösteren bir işlem süreci kontrolü de eklenmiştir.

Çözüm üret seçeneği eğer mevcut herhangi bir iş için, yeterli işçi özelliği ve yeteneği bulamaz ise, kullanıcıya “mevcut kaynaklarla ya da becerilerle problemin çözülemez” uyarısını yapmaktadır. Veri oluşturma problemi ile oluşturulan tüm problemlerde böyle bir ihtimal bulunmamaktadır. Ancak elle düzeltme gibi durumlarda anlamsız istemler oluşursa, program problemin düzeltilmesi için uyarıda bulunacaktır.

Program son 50 iterasyonda daha iyi bir sonuç bulamaz ise çözümü durduracaktır. Problem işlem halinde iken seçme algoritmaları seçeneklerinin değiştirilmesi programın çalışmasını etkileyecektir. Program çözme işlemi yaparken istenilen bilgiler için göster/gösterme seçenekleri kullanılabilir.

Program en son olarak işlem tamamlandığında ya da bir sebeple durdurulursa, o zaman kadar elde edilen değerlerden hesaplanan ortalama, en iyi ve en kötü tamamlanma süresi ve maliyet bilgilerini göstermektedir.

**EK -7**

MRCPSP problemleri için yapılan çözüm yöntemleri listesi(Peteghem, ve ark., 2009)

Yazar	Yıl	Yöntem	R/NR	Veri Seti	Aktivite Numaraları	m	R	NR
Slowinski	1980	LP	RNR	Kendi Veriseti	–	–	–	–
Talbot	1982	Enum	RNR	Kendi Veriseti	10,20,30	1–3	3	0
Patterson ve arkadaşları.	1989	Enum	RNR	–	–	–	–	–
Speranza ve Vercellis	1993	B&B	RNR	Kendi Veriseti	10–20	P 2	1–6	1
Boctor	1993	Heur	R	Kendi Veriseti	50,100	1–4	1,2,4	0
Drexl ve Grünwald	1993	Sezgisel	RNR	Kendi Veriseti	10/10	2–4/2–4	3/3	1/3
Özdamar ve Ulusoy	1994	Sezgisel	RNR	Kendi Veriseti	20–57	1–3	1–6	1–6
Slowinski ve arkadaşları.	1994	SA	RNR	Kendi Veriseti	30	2	3	3
Boctor	1996a	SA	R	Boctor (1993)	50,100	1–4	1,2,4	0
Boctor	1996b	Sezgisel	R	Boctor (1993)	50,100	1–4	1,2,4	0
Sprecher ve arkadaşları	1997	B&B	RNR	PSPLIB	10	3	2	2
Mori and Tseng	1997	GA	R	Own	20,30,40,50,60,70	2–4	4	0
Kolisch ve Drexl	1997	Sezgisel	RNR	PSPLIB	10,30	3	2	2
Hartmann ve Drexl	1998	B&B	RNR	PSPLIB	10,12,14,16	3	2	2
Sprecher ve Drexl	1998	B&B	RNR	PSPLIB/Own	10,12,14,16,18,20	3/1–5/3	2/1–5/2	2/1–3/0
Özdamar	1999	GA	RNR	PSPLIB/Own	10/90	3/2	2/2	2/2
Knotts ve arkadaşları.	2000	Sezgisel	R	Maroto and Tormos (1994)	50	2	3	0
Nonobe ve Ibaraki	2001	TS	R	PSPLIB	30	3	2	2
Jozefowska ve arkadaşları.	2001	SA	RNR	PSPLIB	10,12,14,16,18,20,30	3	2	2
Hartmann	2001	GA	RNR	PSPLIB	10,12,14,16,18,20,30	3	2	2
Bouleimen ve Lecocq	2003	SA	RNR	PSPLIB	10,12,14,16,18,20,30	3	2	2

## MRCPSP problemleri için yapılan çözüm yöntemleri listesi(Peteghem, ve ark., 2009)' Devam

Alcaraz ve arkadaşları.	2003	GA	RNR	PSPLIB/Boctor (1993)	10,12,14,16,18,20,30/50,100	3/1-4	2/1,2,4	2/0
Zhang ve arkadaşları.	2006	PS	RNR	PSPLIB	10,12,14,16,18,20	3	2	2
Zhu ve arkadaşları.	2006	B&C	RNR	PSPLIB	20,30	3	2	2
Lova ve arkadaşları.	2006	Sezgisel	R	Boctor	50,100	1-4	1,2,4	0
Jarboui ve arkadaşları.	2008	PS	RNR	PSPLIB	10,12,14,16,18,20,30	3	2	2
Ranjbar ve arkadaşları.	2008	SS	RNR	PSPLIB	10,12,14,16,18,20	3	2	2
Lova ve arkadaşları.	2009	GA	RNR	PSPLIB/Boctor (1993)	10,12,14,16,18,20,30/50,100	3/1-4	2/1,2,4	2/0

R/NR = Yenilenebilir kaynaklara uygun veri setleri (R: yenilenebilir) veya hem yenilenir ve hem yenilenmeyen (RNR) kaynaklar;

m = Mod sayısı; R = Yenilenebilir kaynak sayısı ve NR = Yenilenemeyen kaynak sayısı

**EK-8**

```

*****
file with basedata      : c154_.bas
initial value random generator: 802306808
*****

projects                : 1
jobs (incl. supersource/sink): 18
horizon                  : 122
RESOURCES
- renewable              : 2 R
- nonrenewable           : 2 N
- doubly constrained     : 0 D
*****

PROJECT INFORMATION:
prnr. #jobs rel.date duedate tardcost MPM-Time
  1  16  0  22   2  22
*****

PRECEDENCE RELATIONS:
jobnr. #modes #successors successors
  1   1   3   2 3 4
  2   3   2   5 10
  3   3   3   8 9 16
  4   3   3   5 7 11
  5   3   2   6 14
  6   3   1   12
  7   3   1   9
  8   3   1   10
  9   3   2  15 17
 10   3   2  11 13
 11   3   1   15
 12   3   1   16
 13   3   1   17
 14   3   1   16
 15   3   1   18
 16   3   1   18
 17   3   1   18
 18   1   0
*****

REQUESTS/DURATIONS:
jobnr. mode duration R 1 R 2 N 1 N 2
-----
  1  1  0  0  0  0  0
  2  1  1  6  0  0  1
    2  1  0 10  8  0
    3  9  0  8  8  0
  3  1  4  6  0  0  5
    2  5  0 10  5  0
    3  7  0 10  0  4
  4  1  1  3  0  0  7
    2  3  3  0  8  0
    3  4  3  0  0  3
  5  1  3  0  1  0  9

```

```

    2 6 10 0 6 0
    3 7 8 0 2 0
6   1 2 9 0 0 8
    2 6 6 0 0 5
    3 10 0 2 0 5
7   1 2 1 0 0 6
    2 4 0 8 0 6
    3 7 0 7 0 6
8   1 3 8 0 0 6
    2 5 0 8 0 4
    3 6 5 0 3 0
9   1 1 2 0 0 7
    2 3 0 4 0 6
    3 6 2 0 0 6
10  1 8 0 10 0 5
    2 8 0 9 1 0
    3 9 0 9 0 6
11  1 1 2 0 8 0
    2 3 0 5 8 0
    3 8 0 3 6 0
12  1 4 0 8 3 0
    2 9 0 8 0 6
    3 10 0 6 2 0
13  1 5 2 0 0 5
    2 6 0 9 0 4
    3 10 0 7 0 2
14  1 5 0 6 0 5
    2 7 3 0 6 0
    3 10 3 0 5 0
15  1 3 8 0 10 0
    2 4 0 7 7 0
    3 9 6 0 0 5
16  1 4 0 7 10 0
    2 6 0 7 0 8
    3 7 5 0 0 1
17  1 2 0 4 0 4
    2 2 0 5 5 0
    3 3 0 2 5 0
18  1 0 0 0 0 0

```

\*\*\*\*\*

RESOURCEAVAILABILITIES:

```

R1 R2 N1 N2
20 26 23 36

```

\*\*\*\*\*

**EK-9**

## RCPSP için literatürde geçen örnek problem

```

*****
file with basedata      : j30_17.bas
initial value random generator: 28123
*****

projects              : 1
jobs (incl. supersource/sink): 32
horizon               : 158
RESOURCES
- renewable           : 4 R
- nonrenewable        : 0 N
- doubly constrained  : 0 D
*****

PROJECT INFORMATION:
prnrr. #jobs rel.date duedate tardcost MPM-Time
  1    30  0   38   26   38
*****

PRECEDENCE RELATIONS:
jobnr. #modes #successors successors
  1     1     3     2 3 4
  2     1     3     6 11 15
  3     1     3     7 8 13
  4     1     3     5 9 10
  5     1     1     20
  6     1     1     30
  7     1     1     27
  8     1     3     12 19 27
  9     1     1     14
 10     1     2     16 25
 11     1     2     20 26
 12     1     1     14
 13     1     2     17 18
 14     1     1     17
 15     1     1     25
 16     1     2     21 22
 17     1     1     22
 18     1     2     20 22
 19     1     2     24 29
 20     1     2     23 25
 21     1     1     28
 22     1     1     23
 23     1     1     24
 24     1     1     30
 25     1     1     30
 26     1     1     31
 27     1     1     28
 28     1     1     31
 29     1     1     32
 30     1     1     32
 31     1     1     32
 32     1     0
*****

REQUESTS/DURATIONS:
jobnr. mode duration R 1 R 2 R 3 R 4

```

---

1	1	0	0	0	0	0
2	1	8	4	0	0	0
3	1	4	10	0	0	0
4	1	6	0	0	0	3
5	1	3	3	0	0	0
6	1	8	0	0	0	8
7	1	5	4	0	0	0
8	1	9	0	1	0	0
9	1	2	6	0	0	0
10	1	7	0	0	0	1
11	1	9	0	5	0	0
12	1	2	0	7	0	0
13	1	6	4	0	0	0
14	1	3	0	8	0	0
15	1	9	3	0	0	0
16	1	10	0	0	0	5
17	1	6	0	0	0	8
18	1	5	0	0	0	7
19	1	3	0	1	0	0
20	1	7	0	10	0	0
21	1	2	0	0	0	6
22	1	7	2	0	0	0
23	1	2	3	0	0	0
24	1	3	0	9	0	0
25	1	3	4	0	0	0
26	1	7	0	0	4	0
27	1	8	0	0	0	7
28	1	3	0	8	0	0
29	1	7	0	7	0	0
30	1	2	0	7	0	0
31	1	2	0	0	2	0
32	1	0	0	0	0	0

\*\*\*\*\*

RESOURCE AVAILABILITIES:

R 1 R 2 R 3 R 4

12 13 4 12

\*\*\*\*\*

**EK-10**

```

*****
file with basedata      : j30_17.bas
initial value random generator: 28123
*****

projects                : 1
jobs (incl. supersource/sink): 32
horizon                 : 158
Labours: 41
Skills: 3
RESOURCES
- renewable             : 4 R
- nonrenewable         : 0 N
- doubly constrained   : 0 D
*****

PROJECT INFORMATION:
pronr. #jobs rel.date duedate tardcost MPM-Time Due Date Cost
  1   30  0  38   26   380
*****

PRECEDENCE RELATIONS:
jobnr. #modes #successors successors
  1     1     3     2 3 4
  2     1     3     6 11 15
  3     1     3     7 8 13
  4     1     3     5 9 10
  5     1     1     20
  6     1     1     30
  7     1     1     27
  8     1     3     12 19 27
  9     1     1     14
 10     1     2     16 25
 11     1     2     20 26
 12     1     1     14
 13     1     2     17 18
 14     1     1     17
 15     1     1     25
 16     1     2     21 22
 17     1     1     22
 18     1     2     20 22
 19     1     2     24 29
 20     1     2     23 25
 21     1     1     28
 22     1     1     23
 23     1     1     24
 24     1     1     30
 25     1     1     30
 26     1     1     31
 27     1     1     28
 28     1     1     31
 29     1     1     32
 30     1     1     32
 31     1     1     32
 32     1     0
*****

REQUESTS/DURATIONS:

```



jobnr.	mode	duration	R 1	R 2	R 3	R 4	B1	B2	B3
1	1	0	0	0	0	0	0;0;0;0	0;0;0;0	0;0;0;0
2	1	8	4	0	0	0	3;0;0;0	0;0;0;0	0;0;0;0
3	1	4	10	0	0	0	8;0;0;0	0;0;0;0	0;0;0;0
4	1	6	0	0	0	3	0;0;0;0	0;0;0;0	0;0;0;0
5	1	3	3	0	0	0	2;0;0;0	0;0;0;0	0;0;0;0
6	1	8	0	0	0	8	0;0;0;0	0;0;0;0	0;0;0;0
7	1	5	4	0	0	0	3;0;0;0	0;0;0;0	0;0;0;0
8	1	9	0	1	0	0	0;0;0;0	0;0;0;0	0;0;0;0
9	1	2	6	0	0	0	5;0;0;0	0;0;0;0	0;0;0;0
10	1	7	0	0	0	1	0;0;0;0	0;0;0;0	0;0;0;0
11	1	9	0	5	0	0	0;3;0;0	0;0;0;0	0;1;0;0
12	1	2	0	7	0	0	0;4;0;0	0;0;0;0	0;1;0;0
13	1	6	4	0	0	0	3;0;0;0	0;0;0;0	0;0;0;0
14	1	3	0	8	0	0	0;5;0;0	0;0;0;0	0;1;0;0
15	1	9	3	0	0	0	2;0;0;0	0;0;0;0	0;0;0;0
16	1	10	0	0	0	5	0;0;0;0	0;0;0;0	0;0;0;0
17	1	6	0	0	0	8	0;0;0;0	0;0;0;0	0;0;0;0
18	1	5	0	0	0	7	0;0;0;0	0;0;0;0	0;0;0;0
19	1	3	0	1	0	0	0;0;0;0	0;0;0;0	0;0;0;0
20	1	7	0	10	0	0	0;6;0;0	0;0;0;0	0;2;0;0
21	1	2	0	0	0	6	0;0;0;0	0;0;0;0	0;0;0;0
22	1	7	2	0	0	0	1;0;0;0	0;0;0;0	0;0;0;0
23	1	2	3	0	0	0	2;0;0;0	0;0;0;0	0;0;0;0
24	1	3	0	9	0	0	0;5;0;0	0;0;0;0	0;2;0;0
25	1	3	4	0	0	0	3;0;0;0	0;0;0;0	0;0;0;0
26	1	7	0	0	4	0	0;0;1;0	0;0;1;0	0;0;0;0
27	1	8	0	0	0	7	0;0;0;0	0;0;0;0	0;0;0;0
28	1	3	0	8	0	0	0;5;0;0	0;0;0;0	0;1;0;0
29	1	7	0	7	0	0	0;4;0;0	0;0;0;0	0;1;0;0
30	1	2	0	7	0	0	0;4;0;0	0;0;0;0	0;1;0;0
31	1	2	0	0	2	0	0;0;0;0	0;0;1;0	0;0;0;0
32	1	0	0	0	0	0	0;0;0;0	0;0;0;0	0;0;0;0

\*\*\*\*\*

KNOWLEDGE LEVEL

B1 B2 B3

0,8 1 1,25

\*\*\*\*\*

LABOR PROPERTIES

Lbr_no	Area	Knowledge	Cost
1	R1	B3	50
2	R1;R3	B2;B1	120
3	R1;R2;R4	B1;B2;B1	120
4	R4	B1	120
5	R1;R2	B3;B2	100
6	R2;R4	B1;B3	120
7	R4	B3	50
8	R2	B2	100
9	R2	B3	50
10	R1	B1	120
11	R2;R4	B1;B1	120
12	R1	B2	100
13	R1;R3;R4	B1;B1;B2	120
14	R1;R2	B1;B1	120
15	R1;R4	B1;B2	120
16	R2;R4	B1;B3	120
17	R2	B1	120

18	R4	B3	50
19	R1;R2;R4	B1;B1;B2	120
20	R2	B1	120
21	R4	B2	100
22	R1	B1	120
23	R2	B1	120
24	R1;R4	B1;B2	120
25	R1	B1	120
26	R1;R2	B1;B3	120
27	R3	B3	50
28	R1;R2;R4	B1;B3;B1	120
29	R3	B2	100
30	R4	B3	50
31	R2;R4	B2;B1	120
32	R1;R4	B2;B2	100
33	R3	B1	120
34	R2;R4	B1;B2	120
35	R2	B2	100
36	R1;R2	B1;B3	120
37	R4	B2	100
38	R1	B1	120
39	R3;R4	B2;B3	100
40	R1	B3	50
41	R2	B3	50

\*\*\*\*\*

**ÖZGEÇMİŞ**

Adı Soyadı : Ömer Nuri ÇAM  
Doğum Yeri ve Tarihi : Akkuş/ORDU 13.05.1982  
Yabancı Dili : İngilizce

Eğitim Durumu (Kurum ve Yıl)

Lise : Ünye Lisesi 1999  
Lisans : Ege Üniversitesi Deri Mühendisliği 2005  
Yüksek Lisans : Uludağ Üniversitesi Endüstri Mühendisliği 2011

Çalıştığı Kurum/Kurumlar ve Yıl : Uludağ Üniversitesi Bilgi İşlem Daire Başkanlığı  
2011

İletişim (e-posta) : [onc@uludag.edu.tr](mailto:onc@uludag.edu.tr)  
: omernuricam@gmail.com

Yayınları\* :

## ULUDAĞ ÜNİVERSİTESİ

## TEZ ÇOĞALTMA VE ELEKTRONİK YAYIMLAMA İZİN FORMU

Yazar Adı Soyadı	Ömer Nuri ÇAM
Tez Adı	Atanacak Ekip Özelliklerine Göre Değişken Süreye Sahip Projelerin Planlanması İçin Genetik Algoritma Bazlı Bir Yöntemin Geliştirilmesi
Enstitü	Uludağ Üniversitesi Fen Bilimleri Enstitüsü
Anabilim Dalı	Endüstri Mühendisliği
Tez Türü	Yüksek Lisans Tezi
Tez Danışman(lar)ı	<b>Doç.Dr. Cenk ÖZMUTLU</b>
Çoğaltma (Fotokopi Çekim) izni	<p>Tezimden fotokopi çekilmesine izin veriyorum</p> <p>Tezimin sadece içindekiler, özet, kaynakça ve içeriğinin % 10 bölümünün fotokopi çekilmesine izin veriyorum</p> <p>Tezimden fotokopi çekilmesine izin vermiyorum</p>
Yayımlama izni	<p>Tezimin elektronik ortamda yayımlanmasına izin Veriyorum</p> <p>Tezimin elektronik ortamda yayımlanmasının ertelenmesini istiyorum</p> <p>1 yıl</p> <p>2 yıl</p> <p>3 yıl</p> <p>Tezimin elektronik ortamda yayımlanmasına izin vermiyorum</p>

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

Tarih :13.07.2011

İmza :