

ATÖLYE TİPİ ÜRETİM SİSTEMLERİNDE İŞ ÇİZELGELEME PROBLEMİNE YÖNELİK SEZGİSEL İKİ YÖNTEM : OPTSIRA ve IssıraÇizelge

Doç. Dr. Mehmet ÇINAR*

GİRİŞ

Bir endüstri işletmesinde ana üretim planlama modeli, ayrıntılı planlama modelinin temel girdilerini sağlar ve eldeki kaynakların planlama sürecinde en uygun kullanımını sağlayacak temel üretim politikalarını belirler. Ayrıntılı üretim planlama modeli ise ana planda belirlenen politikalar temelinde, yapılacak işlerin sırasını ve bu işlere ayrılacak en iyi kaynak miktarlarını belirler. Dolayısı ile ayrıntılı planlama daha kısa bir dönem için tasarlanır. Bu süreç içerisinde iş programları ve üretim çizelgeleri hazırlanır.

Öte yandan üretim planlama ve kontrol işlevinin; atölye tipi üretim sistemlerinde, ürün hattı tipi üretim sistemlerine kıyasla daha zor olduğu bilinen bir gerçektir.

İşlerin belirlenen öncelik değerlerine göre sıraya sokulması çalışmalarına iş sıralaması adı verilir. Böylelikle, bir tezgah boşaldığı zaman tezgaha yüklenecek iş daha önceden yapılan iş sıralamasına göre seçilir. İş çizelgesinde ise işlerin bir tezgahta hangi zamanlarda başlaması ve bitirilmesi gerektiği planlanır. Diğer bir deyişle, elde edilen iş sıralarına zaman boyutunun da eklenmesiyle iş çizelgesi hazırlanmış olur.

ATÖLYE TİPİ ÜRETİM SİSTEMLERİNDE SIRALAMA PROBLEMİ

Atölye sıralama problemi genel olarak, belli bir değerlendirme ölçütü temelinde en iyi iş akışını (işlerin işlenme sırasını) belirlemek olarak tanımlanmaktadır. Klasik atölye sıralama problemi dört faktöre bağlı olarak farklı sınıflara ayrılabilir: [Baker, 1976: Acar, 1985]

İşlerin Geliş Biçimi: Belirli bir dönem için iş listesi bilinmektedir ve işler boş olan atölyeye hemen işlenmek üzere düzenli olarak gelirler. Genellikle iş listesi değişmez. Bu tür yapı statik problem olarak tanımlanır. Eğer iş listesi sürekli ve rasgele değişiyor ve işler düzensiz aralıklarla atölyeye gelmekte ise, problem dinamik olarak nitelendirilir. Bu tür problemlerde, herhangi bir zamanda gelebilecek işin özellikleri sıralamanın sürekli değişmesini gerektirebilir. Bu nedenle dinamik sıralama yerine çizelgeleme terimi yaygınlıkla kullanılmaktadır.

Tezgah Sayısı: Atölyede yer alan tezgah sayısına göre de sıralama problemi, tek tezgahlı veya çok tezgahlı olmak üzere iki türde incelenebilir. Tezgah sayısı, sıralama probleminin çözümünü etkileyen, zorlaştırıcı bir faktör olarak bilinmektedir.

* Erciyes Üniversitesi, İktisadi ve İdari Bilimler Fakültesi, Üretim Yönetimi ABD. Öğretim Üyesi.

İş Akışı: Eğer gelen bütün işler aynı sırayı izliyor ise sıralama problemi, akış tipi olarak nitelendirilir. Yanı sıra işler, farklı sıraları izliyor ise, karışık iş akışından söz edilir.

Değerleme Ölçütü: Atölye performansını değerlendirmek için kullanılan ölçüt, sıralama problemlerinin yapılandırılmasında önemli rol oynamaktadır. Başlıcaları;

- Ürünlerin sistem içinde üretim ve bekleme zamanlarını içeren akış süresi,
- Kuyrukta bekleyen ortalama iş sayılarının oluşturduğu üretim içi stok düzeyi,
- İşlerin tezgahlarda bekleme süreleri,
- Sipariş başına ortalama gecikme süresi,
- Geciken iş sayısı,
- Tezgah ve işgücü kullanım oranları,
- Bu ölçütlerin ortalama ve standart sapma gibi istatistiksel ölçümleri vb.

Çizelgeleme problemlerinin genel gösterimi A/B/C/D olup, burada;

A= İşlerin geliş biçimi; gelişler toplu (n), belirli zamanlarda ya da tek tek belirsiz (rassal),

B= Tezgah sayısı (m),

C= İşleme biçimi; işlerin belirli bir akışı izlediği akış atölyesi (F) veya rassal bir akışı izlediği iş atölyesi (G),

D= Değerlendirme (Performans) ölçütü.

Örneğin, n/3/F/F anlatımı ile; n işin 3 tezgahtan oluşan bir akış atölyesinde işlenebileceği ve değerlendirme ölçütünün ortalama akış süresinin (F) enküçüklenmesi olduğu anlaşılır.

Parametreler:

r_i = i işinin işlenmesi için hazır olduğu zaman,

d_i = i işinin teslim tarihi,

k = 1,2,...,m makina sayısı,

i = 1,2,...,n iş sayısı,

j = 1,2,...,g_i i işinin işlem sayısı,

a_j = $d_i - r_i$ i işinin yapılması için eldeki süre,

k_{ij} = i işinin j işleminin atandığı tezgah,

P_{ij} = i işinin j işleminin işlenme süresi,

$P_i = \sum P_{ij}$ = i işinin toplam işlenme süresi,

w_{ij} = i işinin j işleminden önceki bekleme süresi,

$W_i = \sum W_{ij}$ = i işinin toplam bekleme süresi,

C_i = $r_i + w_i + P_i$, i işinin tamamlanma zamanı,

$F_i = C_i + r_i = W_i + P_i$, i işinin akış zamanı, (Atölye veya üretim süresi)

$L_i = C_i - d_i$, i işinin gecikme durumu,

$L_i = F_i + r_i - d_i = F_i - a_i$; $L_i > 0$ iş gecikmeli, $L_i < 0$ iş erken,

$T_i = \text{Enb } [0, L_i]$, i işinin gecikmeli olması,

$E_i = \text{Enb } [0, L_i]$, i işinin erken olması,

$F_{\text{enb}} = \text{Enbüyük akış süresi}$,

$C = S C_i / n$,

$O_{ij} = i$ işinin j işlemi, (k_{ij}, P_{ij})

$P_{(i)} = \text{Çizelgede } i.\text{nci sıradaki işin işlem süresi}$,

$C_{(i)} = \text{Çizelgede } i.\text{nci sıradaki işin tamamlanma zamanı}$,

olarak tanımlandığında, ortalama ya da enbüyük C , F , L_i , W_i ve T_i birer değerlendirme ölçütü olarak alınır ve enküçüklenmeye çalışılır [Baker, 1976: Çınar, 1990].

AYNI TEKNOLOJİK SIRALI İŞLERİN ÇİZELGELENMESİNE YÖNELİK SEZGİSEL BİR ALGORİTMA - OPTSIRA -

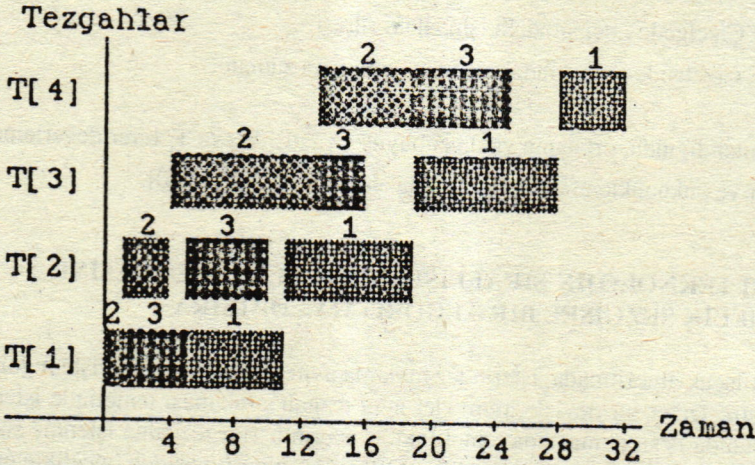
Tanımlanacak algoritmada, işlerin tezgahlarda aynı teknolojik sırada işlem görecekları varsayılmıştır. Diğer bir deyişle, tüm işler aynı tezgah sıralaması temelinde işleneceklerdir. Bu durumda iş sıralamasında temel faktör, işlerin herbir tezgahta işlenme süreleri olup, toplam akış süresi enküçüklenmelidir. OPTSIRA algoritmasında öncelikle en kısa akış süresi gerektiren iş sıralaması bulunmakta, ardısıra işlenme sürelerini de gösteren Gantt diyagramı oluşturulmaktadır. Algoritmada gerekli tek girdi, işlerin herbir tezgahdaki işlem sürelerini gösteren süre setidir. Set uzunluğu iş sayısını, herbir alt küme de varolan tezgah sayısı temelinde işlenme sürelerini göstermektedir. Mathematica dili ile kodlanan OPTSIRA algoritması izleyen adımlardan oluşur:

- Adım 1.** Birinci tezgah temelinde işleri sırala. Elde edilen sıra temelinde iş sayısı kadar sıra permutasyonu oluştur. (Eniyi sıralama; eğer set küçükten büyüğe sıralanarak oluşturulmuşsa, olası tüm sıralamalar arasından iş sayısı kadar seçenek türetildiğinde, yeni set içine düşmektedir. Bu işlem aynı zamanda algoritma işlem süresini de önemli ölçüde azaltır. Tüm sıralama seçenekleri sınanmak istendiğinde, (sor=sır) dönüştürmesi yapılabilir.)
- Adım 2.** Her bir seçenek için, sıralamanın gerektirdiği her bir işin her bir tezgahta sürece başlama ve tamamlama sürelerini bul.
- Adım 3.** En kısa toplam akış süresi gerektiren sıralama seçeneğini ve ilişkin iş ve tezgah başlama-tamamlanma süreleri setlerini belirle.
- Adım 4.** İş ve tezgahlara ilişkin süre matrislerini koordinat matrisine dönüştür.
- Adım 5.** Çizim modülünü kullanarak iş çizelgesini görüntüle.
- Adım 6.** İlişkin hesaplama ve istatistikleri listele, dur.

Örnek 1: Aynı teknolojik sırada 4 tezgah üzerinde süreçlenecek 3 iş ve bunların her bir tezgahta işlenme süreleri izleyen biçimde olsun [Hofeman,1976, s.69: Hottenstein, 1978, s.439]. OPTSIRA algoritması izleyen çıktığı üretir.

In[1]:=
dat={{6,8,9,4},{1,3,9,6},{4,5,3,6}};OPTSIRA[dat]

Out[1]:=

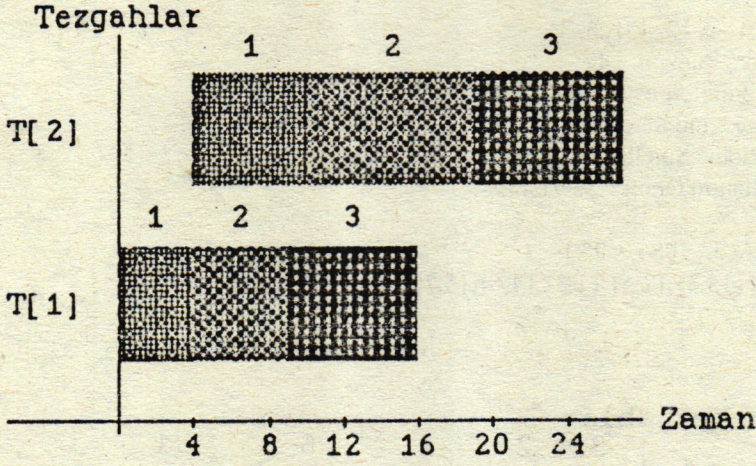


İş Sıralaması --> {2, 3, 1}
Toplam Süreç Zamanı = 32
İş Bekleme Süreleri = {0, 6, 0}
Ortalama Bekleme Süresi = 2.
Tezgah Bekleme Süreleri = {0, 2, 3, 3}
Tezgah Kullanım Oranı = % 75

Çıktıdan izlenebileceği gibi; eniyi sıralama (2,3,1) biçiminde elde edilmektedir. Bu sıralamanın gerektirdiği toplam süreç zamanı 32 olup, 2 numaralı iş 2. tezgahta 1, 3. tezgahta 2 ve 4. tezgahta da 3 birim olmak üzere toplam 6 birim süre beklemektedir. Diğer işler, hiçbir tezgahta süreçlenmek için beklememektedir. Öte yandan 1. tezgah hiç boş kalmamakta, diğerleri ise sırasıyla 3, 7 ve 16 birimlik toplam boş kalma süreleri içermektedir.

Örnek 2: [Özgüven, 1993 s.17, tezgah sıralaması dikkate alınmamıştır.]

In[2]:=sure={{4,6},{5,9},{7,8}};OPTSIRA[sure];
Out[2]:=



İş Sıralaması --> {1, 2, 3}

Toplam Süreç Zamanı = 27

İş Bekleme Süreleri = {0, 1, 3}

Ortalama Bekleme Süresi = 1.33333

Tezgah Bekleme Süreleri = {0, 4}

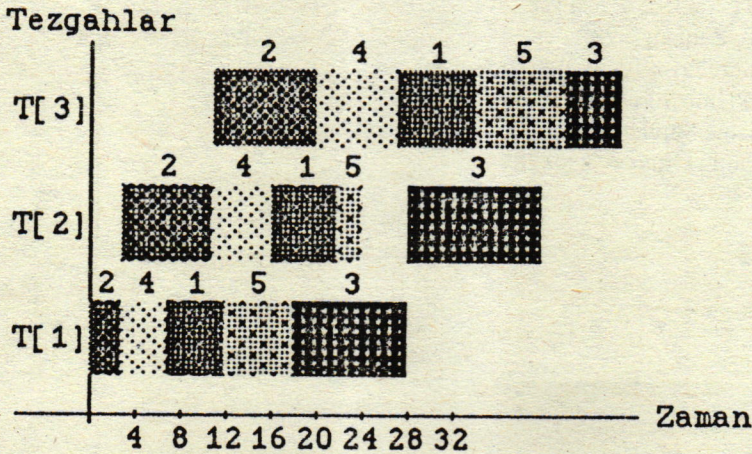
Tezgah Kullanım Oranı = % 85.1852

Örnek 3 : [Çınar, 1990, s. 252]

In[3]:=

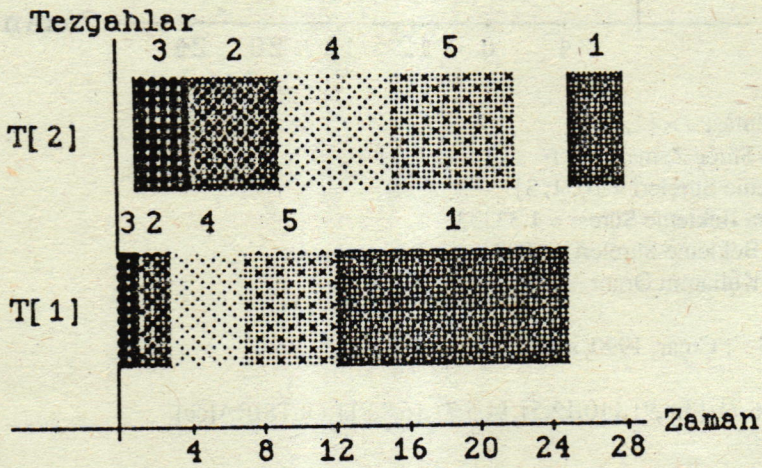
cc={{5,6,7},{3,8,9},{10,12,5},{4,5,7},{6,2,8}};OPTSIRA[cc]

Out[3]:=



İş Sıralaması --> {2, 4, 1, 5, 3}
 Toplam Süreç Zamanı = 47
 İş Bekleme Süreleri = {0, 8, 9, 14, 2}
 Ortalama Bekleme Süresi = 6.6
 Tezgah Bekleme Süreleri = {0, 4, 0}
 Tezgah Kullanım Oranı = % 91.4894

Örnek 4: [Acar, 1985, s.93]
 In[4]:=p1={{13,3},{2,5},{1,3},{4,6},{5,7}}; OPTSIRA[p1]
 Out[4]:



İş Sıralaması --> {3, 2, 4, 5, 1}
 Toplam Süreç Zamanı = 28
 İş Bekleme Süreleri = {0, 1, 2, 3, 0}
 Ortalama Bekleme Süresi = 1.2
 Tezgah Bekleme Süreleri = {0, 3}
 Tezgah Kullanım Oranı = % 89.285

n/2/G/Fenk ve n/3/G/Fenk ÇİZELGELEME PROBLEMLERİ İÇİN SEZGİSEL BİR ALGORİTMA - İssıraCizelge -

Tezgahlar üzerinde işlem sıralamaları farklı teknolojik sıra gerektiren problemlerin çözümüne yönelik olarak geliştirilen ve Mathematica dili ile kodlanarak İssıraCizelge adı verilen bu algoritmada iki tür girdiye gerek duyulur. Birincisi, işlerin her bir tezgahtaki işlem süreleri seti. İkincisi ise tezgah üzerinde işlem teknolojik sıraları seti. Algoritma, en kısa toplam işleme süresi temelinde iş sıralamasını ve ilişkin çizelgeyi, kimi istatistiksel bilgiler ile birlikte çıktı olarak verir. Çözümü araştırılan 3/2/G/Fenk ve 3/3/G/Fenk problemlerinde çözüm süresi sırasıyla 16 ve 20 saniyedir. Algoritma adımları örnekler üzerinde açıklanacaktır .

Örnek 5: 2 tezgah üzerinde 2 işlemlilik 3 iş sıralanacaktır. İşlerin tezgahlarda işlenme süreleri ve işlem sıraları izleyen biçimdedir [Özgüven, 1993, s.17].

tiss={{1,2},{2,1},{1,2}};(* Tezgah sıraları *)
sur={{4,6},{5,9},{7,8}};(* İşlem süreleri *)

Adım 1. Önce tezgahlar üzerinde en kısa tamamlanma süresi oluşturacağı varsayılan iş sıralaması bulunur. Bu amaçla, önce işlerin birinci işlemleri artan sırada, ikinci işlemleri ise azalan sırada sıralanır. Elde edilen sıralama 3. Adımda kullanılan İssırala alt programı yardımıyla en son biçimini alacaktır.

İssıra[tiss,sur];
{{1, 2, 3}, {2, 3, 1}}

Böylelikle, 1.tezgahta işlerin{1,2,3}sırasıyla, 2.tezgahta ise{2,3,1}sırasıyla yapılacağı kabul edilir.

Adım 2. Adım 1'deki sıralama temelinde hangi işin hangi işleminin hangi tezgahta yapılacağını gösteren tezgah-işlem sıralama matrisi geliştirilir.

TezSıra[tiss];
{{{1, 1}, {2, 2}, {3, 1}}, {{2, 1}, {3, 2}, {1, 2}}}

Adım 3. Çizelgelemeye temel koordinat matrisi geliştirilerek birinci işlemleri gösteren m1 seti seçilir. Bu adımda kullanılan alt-program İSSırala olarak kodlanmıştır.

m1= {{1, 1}, {2, 1}, {3, 1}}

Adım 4. m1 seti elemanlarının gerektirdiği toplam işlenme süreleri hesaplanarak, en büyüğü seçilir (m11) ve gerekli işlem süresi hem tezgah durum setine (tezsür), hem de başlangıç süre matrisine (bassur) işlenir. Ataması yapılan (çizelgeleyen) işin yerine, o işin varsa sonraki işlemi m1 setine atanarak, m1 seti boşalınca kadar Adım 4 yinelenir. Yineleme sonuçları izleyen biçimdedir.

kasu= {10, 14, 15}
m11= {3, 1}
tezsür = {7, 0}
bassur = {{(0), (0), (0, 7)}, {(0), (0), (0)}}

m1= {{(1, 1), (2, 1), (3, 2)}}
kasu= {10, 14, 8}
m11= {2, 1}
tezsür = {7, 5}
bassur = {{(0), (0), (0, 7)}, {(0, 5), (0), (0)}}

m1= {{(1, 1), (2, 2), (3, 2)}}
kasu= {10, 9, 8}
m11= {1, 1}
tezsür = {11, 5}
bassur = {{{(7, 11), (0), (0, 7)}, {(0, 5), (0), (0)}}

m1= {{(1, 2), (2, 2), (3, 2)}}
kasu= {6, 9, 8}
m11= {2, 2}
tezsür = {20, 5}
bassur = {{{(7, 11), (11, 20), (0, 7)}, {(0, 5), (0), (0)}}

m1= {{(1, 2), (3, 2)}}
kasu= {6, 8}
m11= {3, 2}
tezsür = {20, 15}
bassur = {{{(7, 11), (11, 20), (0, 7)}, {(0, 5), (7, 15), (0)}}

m1= {{(1, 2)}}
kasu= {6}
m11= {1, 2}
tezsür = {20, 21}
bassur = {{{(7, 11), (11, 20), (0, 7)}, {(0, 5), (7, 15), (15, 21)}}

Adım 5. Adım 3 temelinde oluşturulan Adım 4'ün tamamlanmış süre matrisi, Adım 2'nin koordinat matrisi temelinde dönüştürülerek, iş sırası temelli çizelgenecek (rev) matrisi oluşturulur. Bu matris üzerinde İş Sıralaması, İş Bekleme Süreleri, Tezgaah Bekleme Süreleri, Tezgaah Kullanım Oranı gibi kimi istatistiksel bilgiler türetilir.

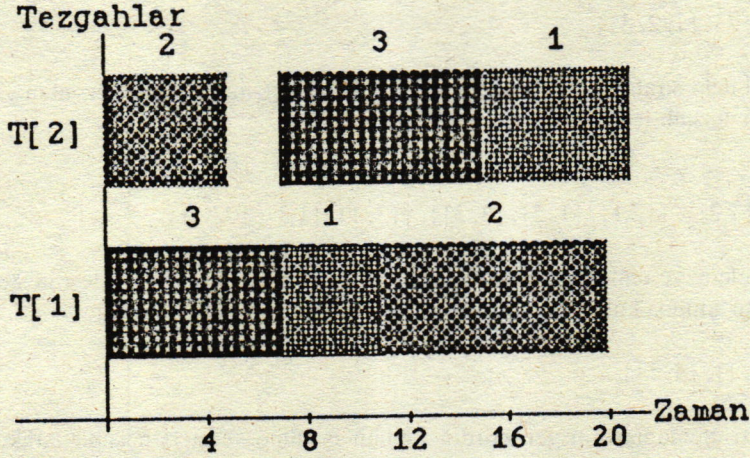
İş Sıralaması --> {{(3, 1, 2), (2, 3, 1)}}
İş Bekleme Süreleri = {4, 6, 0}
Ortalama Bekleme Süresi = 3.33333
Toplam Süreç Zamanı = 21

Tezgah Bekleme Süreleri = {0, 2}

Tezgah Kullanım Oranı = % 90.4762

Adım 6. Adım 5 matrisi (rev) Gantt benzeri çizelgeye dönüştürülür.

GANTCIZIM[rev,issıra]



Algoritma için kullanılacak tek komut;

tiss={{1,2},{2,1},{1,2}};(* İşlem sıraları *)

sur={{4,6},{5,9},{7,8}};(* İşlem süreleri *)

IsSıraCizelge[sur,tiss] biçimindedir.

Örnek 6: 3 tezgah üzerinde 3 işlemli 3 iş sıralanacaktır. İşlerin tezgahlarda işleme süreleri ve işlem sıraları izleyen biçimdedir [Çınar, 1990, s.251].

İş	1. İşlem		2. İşlem		3. İşlem	
	Tezgah	Süre	Tezgah	Süre	Tezgah	Süre
1	1	3	2	4	3	3
2	2	4	1	1	3	2
3	3	3	1	5	2	5

Çözüm için izleyen komut çalıştırılır:

tiss={{1,2,3},{2,1,3},{3,1,2}};(* Tezgah sıraları *)

sur={{3,4,3},{4,1,2},{3,5,5}};(* İşlem süreleri *)

IsSıraCizelge[sur,tiss];

Adım 1. Önce tezgahlar üzerinde en kısa tamamlanma süresi oluşturacağı varsayılan iş sıralaması bulunur. Bu amaçla, önce işlerin birinci işlemleri artan sırada, ikinci işlemleri ise azalan sırada sıralanır.

Issıra[tiss,sur];
{{2, 1, 3}, {3, 1, 2}, {1, 2, 3}}

Adım 2. Adım 1'deki sıralama temelinde hangi işin hangi işleminin hangi tezgahta yapılacağını gösteren tezgah-işlem sıralama matrisi geliştirilir.

TezSıra[tiss];
{{{1,1}, {2,2}, {3,2}}, {{2,1}, {1, 2}, {3, 3}}, {{3, 1}, {1,3}, {2,3}}}

Adım 3. Çizelgelemeye temel koordinat matrisi geliştirilerek birinci işlemleri gösteren m1 seti seçilir. Bu adımda kullanılan alt-program ISSırala olarak kodlanmıştır.

m1= {{1, 1}, {2, 1}, {3, 1}}

Adım 4. m1 seti elemanlarının gerektirdiği toplam işlenme süreleri hesaplanarak, en büyüğü seçilir (m11) ve gerekli işlem süresi hem tezgah durum setine (tezsür), hem de başlangıç süre matrisine (bassur) işlenir. Ataması yapılan (çizelgelenen) işin yerine, o işin varsa sonraki işlemi m1 setine atanarak, m1 seti boşalincaya kadar Adım 4 yinelenir. Yineleme sonuçları izleyen biçimdedir.

m1= {{1, 1}, {2, 1}, {3, 1}}
kasu= {10, 7, 13}
m11= {3, 1}
tezsür = {0, 0, 3}
bassur = {{{0}, {0}, {0}}, {{0}, {0}, {0}}, {{0,3}, {0}, {0}}}

m1= {{1, 1}, {2, 1}, {3, 2}}
kasu= {10, 7, 10}
m11= {1, 1}
tezsür = {3, 0, 3}
bassur = {{{0,3}, {0}, {0}}, {{0}, {0}, {0}}, {{0,3}, {0}, {0}}}

m1= {{1, 2}, {2, 1}, {3, 2}}
kasu= {7, 7, 10}
m11= {3, 2}
tezsür = {8, 0, 3}
bassur = {{{0,3}, {0}, {3,8}}, {{0}, {0}, {0}}, {{0,3}, {0}, {0}}}

m1= {{1, 2}, {2, 1}, {3, 3}}
kasu= {7, 7, 5}

m11= {2, 1}
tezsür = {8, 4, 3}
bassür = {{{0,3}, {0}, {3,8}}, {{0,4},{0},{0}}, {{0,3},{0},{0}}}

m1= {{1, 2}, {2, 2}; {3, 3}}
kasu= {7, 3, 5}
m11= {1, 2}
tezsür = {8, 8, 3}
bassür = {{{0,3},{0},{3,8}}, {{0,4},{4,8},{0}}, {{0,3},{0},{0}}}

m1= {{1, 3}, {2, 2}, {3, 3}}
kasu= {3, 3, 5}
m11= {3, 3}
tezsür = {8, 13, 3}
bassür = {{{0,3},{0},{3,8}}, {{0,4},{4,8},{8,13}}, {{0,3},{0},{0}}}

m1= {{1, 3}, {2, 2}}
kasu= {3, 3}
m11= {2, 2}
tezsür = {9, 13, 3}
bassür = {{{0,3},{8,9},{3,8}}, {{0,4},{4,8},{8,13}}, {{0,3},{0},{0}}}

m1= {{1, 3}, {2, 3}}
kasu= {3, 2}
m11= {1, 3}
tezsür = {9, 13, 11}
bassür = {{{0,3},{8,9},{3,8}}, {{0,4},{4,8},{8,13}}, {{0,3},{8,11},{0}}}

m1= {{2, 3}}
kasu= {2}
m11= {2, 3}
tezsür = {9, 13, 13}
bassür= {{{0,3},{8,9},{3,8}}, {{0,4},{4,8},{8,13}}, {{0,3},{8,11},{11,13}}}

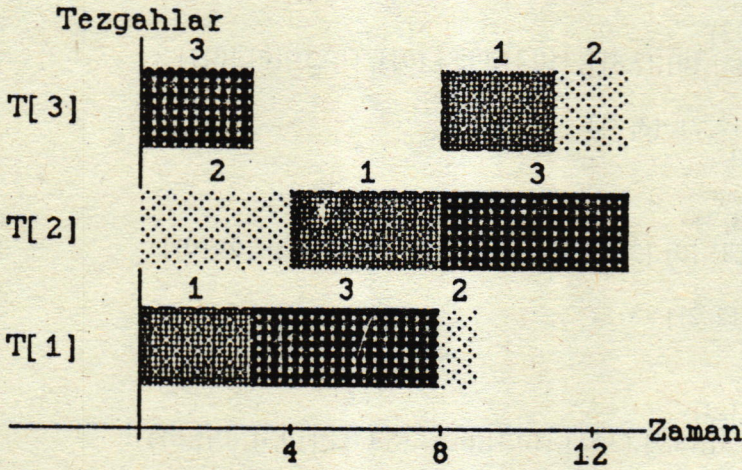
Adım 5. Adım 3 temelinde oluşturulan Adım 4'ün tamamlanmış süre matrisi, Adım 2'nin koordinat matrisi temelinde dönüştürülerek, iş sırası temelli çizelgelenecek (rev) matrisi oluşturulur. Bu matris üzerinde İş Sıralaması, İş Bekleme Süreleri, Tezgah Bekleme Süreleri, Tezgah Kullanım Oranı gibi kimi istatistiksel bilgiler türetilir.

rev= {{{0,3},{8,11},{11,13}}, {{0,4},{4,8},{8,13}}, {{0,3},{3,8},{8,9}}}
İş Sıralaması --> {{1, 3, 2}, {2, 1, 3}, {3, 1, 2}}
İş Bekleme Süreleri = {1, 6, 0}
Ortalama Bekleme Süresi = 2.33333
Toplam Süreç Zamanı = 13

Tezgah Bekleme Süreleri = {0, 0, 5}
Tezgah Kullanım Oranı = % 61.5385

Adım 6. Adım 5 matrisi (rev) Gantt benzeri çizelgeye dönüştürülür.

GANTCIZIM[rev]



SONUÇ

Bu çalışmada, akış tipi üretim atölyelerinde tezgahlara iş yüklenmesi, bilinen adıylarla iş sıralaması ve çizelgeleme problemleri incelenerek, tezgah üzerindeki iş akışlarının aynı teknolojik sırayı izleyip izlememesi durumlarına ilişkin iki yeni sezgisel algoritma tanıtılmıştır. İşlem süresi açısından oldukça doyurucu olduğu gözlenen söz konusu algoritmalar, aynı zamanda üzerinde çalışılan örnekler üzerinde de eniyi çözüme ulaşmaktadır. Yanısıra, çıktı olarak türetilen bir çok değişik değerlendirme ölçütü istatistikleri de kıyaslama için karar vericiye fikir sunabilmektedir.

KAYNAKÇA

ACAR, Nesime. Üretim Planlaması Yöntem ve Uygulamaları. MPM Yayını, No:280, Ankara, 1985.

BAKER, R. Introduction to Sequencing and Scheduling. John Wiley and Sons, NewYork, 1974:

ÇINAR, Mehmet. Yönetmel Kararlara İlişkin Sayısal Yöntemler. Erciyes Üniversitesi Yayını No: 8, Kayseri, 1990.

HOFEMAN, R.J. Production: Concepts, Analysis and Control. 3.rd Ed., Columbus, Ohio: Merrill, 1976.

HOTTENSTEIN, M.P. Models and Analysis for Production Management. International Textbook Company, Scranton, Pennsylvania, 1968.

ÖZGÜVEN, Cemal. " İki Karma Sıfır-Bir Yükleme Modeli ", İk. ve İd. Bil. Fak. Dergisi, Temmuz 1993, s. 34.

EK-1: OPTSIRA PROGRAMI

```

OPTSIRA[data ]:=
Block[{sir=data,Ti},setT={};opts={};sir1=sir;sets={};
  sir=Sort[sir];
  sor=Permutations[sir];
  For[y=1,y!=Length[sir]+1,y++,
    tor=sor[[y]];sets=Append[sets,tor];
    Ft=Transpose[tor];
    ek=0;Ti={};
  For[x=1,x!=Length[Ft]+1,x++,
    al=Ft[[x]];
    t={};bak={};cko=0;
    For[x1=1,x1!=Length[al]+1,x1++,
      bal=al[[x1]];
      If[x>=2,ek=tek[[x1]];
      If[!cko < ek,ek=cko];
      ek1=ek;ek=ek1+bal;
      t=Append[t,{ek1,ek}];bak=Append[bak,ek];
      cko=ek;
    ];
    tek=bak;Ti=Append[Ti,t];
  ];
  setT=Append[setT,Ti];opts=Append[opts,Max[Ti]];
];

sir=Position[opts,Min[opts]][[1,1]];
bb1={};
Do[tt=sir1[[i]]->i;bb1=Append[bb1,tt],{i,Length[sir1]}];
cizim=sets[[sir]];
son=cizim /.bb1;optciz=setT[[sir]];
fols={};bassur=optciz;
trba=Transpose[bassur];
fols={};
For[x3=1,x3!=Length[trba]+1,x3++,
  sir1=Sort[trba[[x3]]];fol={};
Do[fo=Abs[Max[sir1[[ct]]]-Min[sir1[[ct+1]]]];
  fol=Append[fol,fo],{ct,1,Length[sir1]-1}];
fol1=Apply[Plus,fol];
fols=Append[fols,fol1];
];(* iş bekleme*)
fark={};
For[krt=1,krt!=Length[bassur]+1,krt++,
  tezg=Sort[bassur[[krt]]];ark={};

```

```

For[krt1=1,krt1!=Length[tezg],krt1++,
  ato1=Last[tezg[[krt1]]];
  ato2=First[tezg[[krt1+1]]];
  frk=ato2-ato1;
  ark=Append[ark,frk];arko=Apply[Plus,ark]
];
fark=Append[fark,arko];
farko=Apply[Plus,fark];
etk=N[((Max[bassur]-farko)/Max[bassur])*100];

Print["İş Sıralaması --> ",son];
Print["İş Bekleme Süreleri = ",fols];
Print["Ortalama Bekleme Süresi = ",
N[Apply[Plus,fols]/Length[fols]]];
Print["Toplam Süreç Zamanı = ",Max[bassur]];
Print["Tezgah Bekleme Süreleri = ",fark];
Print["Tezgah Kullanım Oranı = % ",etk];

```

```
GANTCIZIM[bassur]
```

```
]
```

```
GANTCIZIM[data12_] :=
```

```

Block[{ basla=data12,to,
  kod=Range[1,30,3];
  kod1={RGBColor[1,0,0],RGBColor[1,1,0],
  RGBColor[0,0,1],RGBColor[1,0,1],RGBColor[0,1,1],
  RGBColor[1,1,1],RGBColor[1,1,0]};
  kutular={};eksen={};raklar={};ra={0.5,3.5,6.5,9.5,12.5,15.5};

  kodis=Table[son,{Length[bassur]}];
  to=Array[" T",Length[basla]];yesi={};
  For[x=1,x!=Length[basla]+1,x++,
    s1=basla[[x]];
    kutu1={};yes1={};rak1={};
    eks=Text[to[[x]],{-5,kod[[x]]+1}];
    For[x1=1,x1!=Length[s1]+1,x1++,
      s2=s1[[x1]];
      If[s2!={0,0},
        kutu={kod1[[kodu[[x,x1]]]],Rectangle[{s2[[1]],kod[[x]]],
        {s2[[2]],kod[[x]]+2}}];
        rak=Text[kodis[[x,x1]],{Apply[Plus,s2]/2,ra[[x+1]]}];
        kutu1=Append[kutu1,kutu];

```

```

rak1=Append[rak1,rak];
yes1=Append[yes1,kodis[[x,x1]]];
];kutular=Append[kutular,kutu1];
raklar=Append[raklar,rak1];
yesi=Append[yesi,yes1];
eksen=Append[eksen,eks];
];

Show[Graphics[kutular,Axes->{0,0},
AxesLabel->{" Zaman",Tezgahlar},
Ticks->{Range[0,32,4],None}],Graphics[eksen],
Graphics[raklar]]

]

```

EK-2: IsSıraCizelge PROGRAMI

```

ISSirala[suremat__,tezislemmat__]:=
Block[{sure=suremat,tezislem=tezislemmat},
tezsür={0,0,0};
an=Table[{i,j},{j,1,Length[sure]},
{i,1,Length[sure]}];
ant=Transpose[an];
an=Take[an,Length[tezislem]];
bassur=Table[{0},{j,1,Length[sure]},
{i,1,Length[sure]}];
bassur=Take[bassur,Length[tezislem]];
m1=Take[an,1][[1]];

Do[
kasu={};
For[c=1,c!=Length[m1]+1,c++,
tsu=m1[[c]][[1]];ver=sure[[tsu]];
al=Apply[Plus,ver];
kasu=Append[kasu,al];
t=Position[kasu,Max[kasu]];
If[Length[t]>=2,
t1=t[[1]][[1]];t2=t[[2]][[1]];
sec1=Take[m1,{t1}][[1,1]];
sec2=Take[m1,{t2}][[1,1]];
yh=sure[[sec1]];yh1=Count[yh,0];
oh=sure[[sec2]];yh2=Count[oh,0];
If[yh1<yh2,t=t1,t=t2],t=t[[1,1]]];
];

```



```

m11=m1[[t]];se11=m11[[1]];se22=m11[[2]];
secme=Position[tezislem,m11][[1]];
se1=secme[[1]];se2=secme[[2]];
If[se22>1,
  ilkin=Position[tezislem,{se1,se22-1}][[1]];
  ilk1=bassur[[ilkin[[1]],ilkin[[2]]][[2]];
  ilk2=Max[bassur[[se1]];
  ilks={ilk1,ilk2};ilk=Max[ilks],
  ilk=Max[bassur[[se1]]];
  tezsurr[[se1]]=ilk+sure[[se1,se22]];
  bassurr[[se1,se2]]=
  Flatten[{ilk,tezsurr[[se1]]}];
  sure[[se1,se22]]=0;
If[
  MemberQ[Flatten[tezislem,1],m11+(0,1)],
  m1[[t]]=m11+(0,1),m1=Drop[m1,{t}];
  sure[[m11[[1]],m11[[2]]]=0
,
{Length[Flatten[an,1]]};
  bassur=Take[bassur,Length[tezislem]];
  yenisle={};
  Do[sir1=bassur[[trt]];bb22={};
  Do[ttz=sir1[[i]]->i;bb22=Append[bb22,ttz],
  {i,Length[sir1]}];
  son3=Sort[bassur[[trt]]]/.bb22;
  yenisle=Append[yenisle,son3],
  {trt,Length[bassur]}];issira=yenisle;
  trba=Transpose[bassur];
fols={};
For[x3=1,x3!=Length[trba]+1,x3++,
  sir1=Sort[trba[[x3]]];fol={};
Do[fo=Abs[Max[sir1[[ct]]]-Min[sir1[[ct+1]]]];
  fol=Append[fol,fo],{ct,1,Length[sir1]-1}];
  fol1=Apply[Plus,fol];
  fols=Append[fols,fol1]
  ];(* iş bekleme*)
fark={};
For[krt=1,krt!=Length[bassur]+1,krt++,
  tezg=Sort[bassur[[krt]]];ark={};
For[krt1=1,krt1!=Length[tezg],krt1++,
  ato1=Last[tezg[[krt1]]];
  ato2=First[tezg[[krt1+1]]];
  frk=ato2-ato1;
  ark=Append[ark,frk];arko=Apply[Plus,ark]];

```

```

fark=Append[fark,arko];
farko=Apply[Plus,fark];
etk=N[((Max[bassur]-farko)/Max[bassur])*100];
Print["İş Sıralaması --> ",issıra];
Print["İş Bekleme Süreleri = ",fols];
Print["Ortalama Bekleme Süresi = ",
N[Apply[Plus,fols]/Length[fols]];
Print["Toplam Süreç Zamanı = ",Max[bassur]];
Print["Tezgah Bekleme Süreleri = ",fark];
Print["Tezgah Kullanım Oranı = % ",etk];
GANTCIZIM[bassur]
]

```

```

GANTCIZIM[data12_]:=
Block[{ basla=data12,to},
  kod=Range[1,30,3];
  kod1={RGBColor[1,0,0],RGBColor[1,1,0],
  RGBColor[0,0,1],RGBColor[1,0,1],RGBColor[0,1,1],
  RGBColor[1,1,1],RGBColor[1,1,0]};
  kutular={};eksen={};raklar={};ra={0.5,3.5,6.5,9.5};
  bed1={};
  Do[atk=Transpose[tezs[[mus]]][[1]];
  bed1=Append[bed1,atk,
  {mus,1,Length[tezs]}];kodis=bed1;
to=Array[" T",Length[basla]];yesi={};
For[x=1,x!=Length[basla]+1,x++,
  s1=basla[[x]];
  kutu1={};yes1={};rak1={};
  eks=Text[to[[x]],{-3,kod[[x]]+1}];
  For[x1=1,x1!=Length[s1]+1,x1++,
  s2=s1[[x1]];
  kutu={kod1[[kodis[[x,x1]]]],Rectangle[{s2[[1]],kod[[x]],
  {s2[[2]],kod[[x]]+2}}];
  rak=Text[kodis[[x,x1]],{Apply[Plus,s2]/2,ra[[x]]}];
  kutu1=Append[kutu1,kutu];
  rak1=Append[rak1,rak];
  yes1=Append[yes1,kodis[[x,x1]]];
  ];kutular=Append[kutular,kutu1];
  raklar=Append[raklar,rak1];
  yesi=Append[yesi,yes1];
  eksen=Append[eksen,eks];
Show[Graphics[kutular,Axes->{0,0},
  AxesLabel->{Zaman,Tezgahlar},

```

```
Ticks->{Range[0,32,4],None}],Graphics[eksen],  
Graphics[raklar]]  
]
```

```
IsSıraCizelge[suremat77__,tezislemmat77__]:=  
Block[{sure=suremat77,tiss=tezislemmat77},  
sur=sure;tezs={};  
Do[bel=Position[tiss,g];  
tezs=Append[tezs,bel],{g,1},  
Length[Union[Flatten[tiss]]]];  
ISSırala[sure,tezs]  
]
```