

PARALEL MAKİNELERDE İŞLERİN VE MAKİNE OPERATÖRLERİNİN BİRLİKTE ÇİZELGELENMESİ PROBLEMİNE ÇÖZÜM YAKLAŞIMLARI

Emrah B. Edis*, **Ceyda Oğuz****, **İrem Özkarahan*****

*Celal Bayar Üniversitesi, Mühendislik Fakültesi, Endüstri Mühendisliği Bölümü, 45140 Muradiye, Manisa

**Koç Üniversitesi, Mühendislik Fakültesi, Endüstri Mühendisliği Bölümü, 34450, Sarıyer, İstanbul

***Troy Üniversitesi, Bilgisayar Bilimleri (Computer Science), Montgomery, AL 36104, A.B.D.

emrah.edis@cbu.edu.tr, iozkarahan@troy.edu, coguz@ku.edu.tr

(Geliş/Received: 07.03.2011; Kabul/Accepted: 13.03.2012)

ÖZET

Literatürde ele alınan paralel makine çizelgeleme problemlerinin çoğunluğu, işlerin ihtiyaç duyduğu tek kaynağın makine olduğu varsayımına dayanır ve yalnızca iş-makine çizelgeleme problemi ile ilgilidir. Oysa gerçek yaşam problemlerinde işler, makineler dışındaki ek kaynaklara da gereksinim duymaktadır. Ek kaynakların en çok karşılaşılan örneği, farklı makinelerde farklı işlerle çalışabilme yeteneğine sahip olan makine operatörleridir. Bu çalışmada işlerin ve operatörlerin paralel makineler boyunca çizelgelemesini gerektiren bir gerçek yaşam problemi ele alınmıştır. Operatörler, makineleri izlemek, ürünleri çıkarmak, ürün üzerindeki fazlalıkları almak vb. işlemlerle sorumludur. Bu işlemler, herhangi bir ürünün işlenme süresinin tamamı boyunca bir operatörün makine başında durmasını gerektirmeyebilir. Bu kapsamda, ele alınan problemin yazındaki benzer çalışmalardan önemli bir farkı; bir operatörün belli zaman dilimleri boyunca birden fazla makineye atanabilmesi özelliğidir. Operatörün aynı zaman diliminde ilgileneceği makineler belirlenirken, makinelerin fiziksel yakınlıkları da dikkate alınmalıdır. Ele alınan problem için en son işin bitiş süresini en küçükleme amaçlı tamsayı ve kısıt programlama modelleri geliştirilmiştir. Geliştirilen modeller, farklı parametreler içeren test problemlerine uygulanmış ve performansları değerlendirilmiştir. Son olarak, kısıt programlama modelinin gerçek yaşam problemine uygulanabilirliği gösterilmiştir.

Anahtar Kelimeler: paralel makineler, iş ve operatör çizelgeleme, tamsayı programlama, kısıt programlama

SOLUTION APPROACHES FOR SIMULTANEOUS SCHEDULING OF JOBS AND OPERATORS ON PARALLEL MACHINES

ABSTRACT

In literature, most of the studies related to parallel machine scheduling problems assume that jobs require only machines as the processing resources and accordingly deal with simply job-machine scheduling problem. However, in real-life manufacturing environments, jobs may also require additional resources. A common example of additional resources is cross-training workers that perform tasks related with different machines. This study handles a real-life problem that requires simultaneous scheduling of jobs and operators over the parallel machines. Operators are responsible for monitoring the machines, unloading the parts and trimming extra material. These tasks may not require an operator's full attention during the processing of a job at one machine. In this context, a significant distinguishing feature of the investigated problem is that an operator can be assigned to more than one machine through the specified time periods. While determining the machines that an operator has to deal with during the same scheduling periods, the physical closeness of the machines should also be taken into account. For the problem on hand, with the aim of minimizing the completion time of the last job, integer and constraint programming models have been developed. The models have been evaluated through the test problems with different parameters, and their performances have been discussed. Finally, the applicability of the proposed constraint programming model on the real-life problem has been shown.

Keywords: parallel machines, job and operator scheduling, integer programming, constraint programming

1. GİRİŞ (INTRODUCTION)

Paralel makine çizelgeleme (PMÇ) problemlerinin büyük çoğunluğunda, işlerin gereksinim duyduğu tek kaynağın makine olduğu varsayılmaktadır [1]. Oysa gerçek yaşam problemlerinde işler, işlem süreleri boyunca, makine operatörleri, ürün kalıpları, robotlar vb. ek kaynaklara da ihtiyaç duymaktadırlar [2]. Ek kaynakların en çok karşılaşılan örneği, farklı makinelerde farklı işlerle çalışabilen makine operatörleridir [3]. Bu çalışmada işlerin ve operatörlerin paralel makineler boyunca çizelgelenmesini gerektiren bir gerçek yaşam problemi ele alınmıştır. Operatörler, makineleri izlemek, ürünleri çıkarmak, ürün üzerindeki son işlemleri yapmak vb. işlemlerle sorumludur. Bu işlemler, herhangi bir ürünün işlenme süresinin tamamı boyunca bir operatörün makine başında durmasını gerektirmeyebilir. Bu kapsamda, ele alınan problemin yazındaki benzer çalışmalardan önemli bir farkı; bir operatörün, işlerin operatör gereksinimleri doğrultusunda, belli zaman dilimleri boyunca birden fazla makineye atanabilmesi özelliğidir. Operatörün aynı zaman diliminde ilgileneceği makineler belirlenirken makinelerin fiziksel yakınlıkları da dikkate alınmalıdır [4]. Ele alınan problem için en son işin bitiş süresini (makespan) en küçükleme amaçlı tamsayı ve kısıt programlama modelleri geliştirilmiştir. Geliştirilen modeller, farklı parametrelere dayalı test problemlerine uygulanmış ve modellerin performansları değerlendirilmiştir. Son olarak kısıt programlama modelinin gerçek yaşam problemlerine uygulanabilirliği gösterilmiştir.

2. YAZIN TARAMASI (LITERATURE REVIEW)

Çizelgeleme yazınında iş ve makinelerin operatör gereksinimleri çoğu kez dikkate alınmamış, mevcut operatör sayısının sonsuz olduğu (kısıt yaratmadığı) varsayılmıştır [1]. İş ve makinelerin ek kaynak gereksinimlerinin dikkate alındığı çalışmaların büyük bölümünde ise ek kaynakların mevcut miktarlarının sınırlı olduğu üzerinde durulmuş, ek kaynakların (örn. operatör) çizelgelenmesi ile ilgilenilmemiştir. Edis [5] ek kaynak gerektiren PMÇ problemleri ile ilgili genel bir yazın tarama çalışması yapmıştır. Daniels vd. [6] ve Chen [7] işlerin işlem zamanlarının atanan ek kaynak sayısına bağlı olduğu ve ek kaynağın mevcut miktarının sınırlandığı özdeş PMÇ problemi üzerinde çalışmışlardır. Grigoriev vd. [8, 9, 10] işlem zamanları atanan ek kaynak miktarına bağlı olan ilgisiz PMÇ problemini ele almıştır. Hu [11, 12] özdeş paralel makineler için problemi iki alt problem olarak incelemiş, önce işlerin çizelgesini oluşturmuş, daha sonra ise toplam operatör sayısı kısıdı altında operatörleri makinelere atayarak çizelgeyi güncellemiştir. Hu [11, 12]'de ele alınan problemler için Chaudry ve Drake [13] ve Chaudry [14] genetik algoritma tabanlı yaklaşımlar geliştirmiştir. Yine ek kaynak gerektiren PMÇ problemleri için, Edis vd.

[15] ve Edis ve Ozkarahan [16, 17] işlem sürelerinin eşit olduğu ve eşit olmadığı durumları ele almıştır. Diğer yandan, PMÇ yazınında operatör ve makine çizelgelenmesini birlikte ele alan çok az sayıda çalışma vardır. Bourland ve Carl [18] plastik parçaların enjeksiyon makinelerinde çizelgelenmesi problemi ile ilgilenmiş ve ek kaynak olarak ele alınan operatörün - parçaların gereksinimleri doğrultusunda- birden fazla komşu makineye atanabildiği problemi tanımlamıştır. Problemi planlama ve kontrol problemi olarak iki aşamada ele alıp ayar süreleri ve elde stok bulundurma kaynaklı toplam maliyeti en küçüklemeyi amaçlamışlardır. Zouba vd. [19] operatörün faaliyetlerinin birden fazla makineye paylaştırıldığı özdeş PMÇ problemi ile ilgilenmiş ve probleme farklı bir bakış açısı getirerek işlem zamanlarını etkileyen operatör-makine dağıtım modları oluşturmuştur. Ele aldığımız problemin özellikleri olan operatörü paylaşacak işlerin komşu makinelere atanması ve işlerin operatörlerle birlikte çizelgelenmesi problemi en son işin bitiş süresini (makespan) en küçükleme amacı kapsamında önceki çalışmalarda ele alınmamış ve şu ana kadar tamsayı ve kısıt programlama modelleri geliştirilmemiştir.

3. PROBLEM TANIMI (PROBLEM DEFINITION)

Ele alınan problem, elektrik malzemeleri üreten bir firmanın plastik enjeksiyon kalıplama bölümündeki gerçek bir çizelgeleme probleminden ortaya çıkmıştır. Bu bölümde, montaj bölümüne gönderilecek plastik parçalar imal edilmektedir. Her plastik parça kendisine uygun bir kalıba sahiptir ve bu kalıpla birlikte bir makinede tek aşamada üretilmektedir. Bir kalıp, farklı hammadde kullanılarak birden fazla plastik ürünün imalatında kullanılabilir. Her kalıp tipinden yalnızca bir adet mevcuttur. Bu durum, aynı kalıbı kullanan plastik ürünlerin aynı anda üretilmeyeceği gerçeğini beraberinde getirir. Kullanılan enjeksiyon makineleri de farklı özelliklere sahiptir ve her kalıp tipi yalnızca kendisine uygun makine kümesinden bir makinede kullanılabilir. Bu durum 'makine elverişliliği kısıtları' olarak tanımlanabilir.

Öte yandan, belirli sayıda makine operatörü, makineleri izlemek, üretilen parçaları makineden çıkarmak, fazla çapaklarını almak vb. işlemleri yapmakla sorumludur. Bu işlemler herhangi bir ürün veya kalıp için operatörün tüm zamanını almayabilir (bkz. [18]). İmalat ortamı incelendiğinde üç tip işlem mevcuttur:

- (1) *Manüel operasyonlu işlemler*: Ürünün imalat süresi boyunca bir operatör makine başında bulunmalıdır.
- (2) *Yarı-otomatik işlemler*: Bir operatör aynı anda iki makine ile ilgilenebilmektedir (İşlemdaki operatör ihtiyacı 1/2 olarak tanımlanmaktadır).

(3) *Tam-otomatik işlemler*: Ürünün imalatı boyunca operatöre ihtiyaç yoktur.

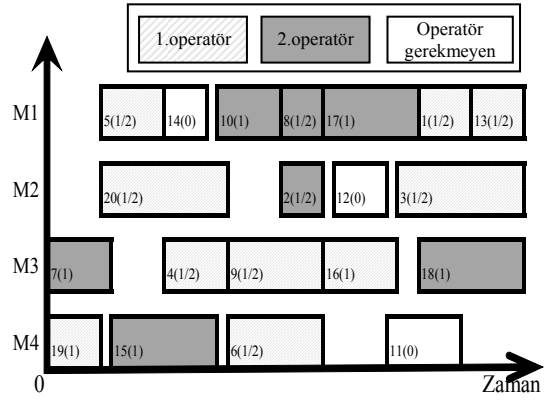
Tanımlanan problemde, kalıpların kullanımı konusunda problemi kolaylaştırıcı bir varsayım yapılabilir. Önceden bahsedildiği gibi, atölyede her tip kalıptan bir adet bulunmakta ve bir tip kalıp birden fazla ürünün imalatında kullanılabilir. Bu açıdan, aynı kalıbı kullanan plastik parçaların bir iş grubu oluşturduğu varsayımı yapılmıştır. Böylece kalıplar ek kaynak olarak değil, çizelgenecek işler olarak düşünülmektedir. Bu varsayım, ek kaynak olarak yalnızca operatörler kaldığı için, ek kaynak tipi sayısını bire düşürmektedir. Bu çalışmada önerilen tamsayı ve kısıt programlama modelleri, ek olarak aşağıdaki varsayımları dikkate almaktadır:

- Bir makine aynı anda en fazla bir adet kalıp üzerinde işlem yapabilmektedir.
- Makineler ve belli sayıda operatör belirlenen çizelgeleme periyodu boyunca kesintisiz olarak hizmet verebilmektedir.
- İşlerin bölünmesine izin verilmemektedir.
- İşlerin işlem zamanları kesin olarak belirlenmiştir.
- İşlerin (kalıpların) işlem zamanları boyunca operatör gereksinimleri sabit değerler olarak alınmıştır (0, 1/2, 1).
- Her operatör tüm makinelerdeki işlerle aynı uzmanlıkta ilgilenebilmektedir.
- Çizelgeleme periyodu bir saatlik eşit dilimlere bölünmüştür.

Ele alınan sistemin daha iyi anlaşılabilmesi için 20 iş ($i = 1, \dots, 20$), dört makine (M1, M2, M3, M4), mevcut sayısı iki adet ($R = 2$) olan bir ek kaynaktan (örn. operatör) oluşan bir PMÇ probleminin aşağıdaki parametrik değerlere sahip olduğu düşünülün:

- İşlem zamanları $p_i = [5, 4, 12, 6, 6, 9, 6, 4, 9, 6, 7, 5, 5, 4, 10, 7, 9, 10, 5, 12]$;
- Ek kaynak gereksinimleri $res_i = [1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1, 1/2, 1/2, 1, 0, 0, 1/2, 0, 1, 1, 1, 1, 1/2]$
- M_i her işin (veya kalıbın) işlenebileceği makine kümesi $M_i = [[1,1,1,1], [1,1,1,0], [0,1,0,1], [1,1,1,0], [1,1,0,1], [1,1,1,1], [0,0,1,1], [1,0,0,1], [0,1,1,0], [1,1,1,1], [1,1,0,1], [1,1,1,1], [1,0,1,1], [1,0,1,0], [1,0,0,1], [1,0,1,0], [1,1,0,0], [0,1,1,0], [0,0,1,1], [1,1,1,0]]$
- Fiziksel yakınlığı olan makineler: M1-M2, M1-M3, M2-M3, M3-M4
- Fiziksel yakınlığı olmayan makineler: M1-M4, M2-M4

Şekil 1, örnek probleme ait en iyi çizelgeyi vermektedir. Her iş, 'işin indeksi (operatör ihtiyacı)' ile belirtilmektedir.



Şekil 1. Örnek probleme ait en iyi çizelge (Optimal schedule of the given example)

Şekil 1'deki çizelge, probleme ait tüm kısıtları karşılamaktadır:

- ✓ Hiçbir zaman diliminde mevcut operatör sayısı limiti aşılmamıştır.
- ✓ Hiçbir zaman diliminde işler üst üste gelmemektedir.
- ✓ Hiçbir zaman diliminde bir operatöre kapasitesinden fazla iş atanmamıştır.
- ✓ İşler sadece işlenebilecekleri makine kümesinden bir makineye atanmıştır.
- ✓ İşlere gereksinimleri doğrultusunda operatör atanmıştır.
- ✓ Bir operatörün aynı zaman diliminde ilgileneceği birden fazla iş varsa, bu işler, fiziksel yakınlıkları olan makinelere atanmıştır.

Ele alınan problemin zorluğu analiz edilecek olursa, özdeş makinelerden oluşan ve en son işin bitiş zamanının en küçüklenmesi amaçlı PMÇ problemleri NP-zor sınıftadır [20]. Daniels vd. [6] mevcut sayısı iki adet ($R = 2$) olan bir ek kaynaktan oluşan ve ek kaynak ihtiyaçları $res_i \in \{1, 2\}$ olan çizelgeleme probleminin NP-zor olduğunu göstermiştir. Buna göre tanımlanan problem de NP-zor sınıftadır.

4. ÇÖZÜM YAKLAŞIMLARI (SOLUTION APPROACHES)

Bu bölümde, geliştirilen tamsayı ve kısıt programlama modelleri verilmiştir. Parametreler verilirken, işlerin operatör gereksinimleri iki ile çarpılarak tamsayı hale getirilmiş, tutarlılığı sağlamak üzere mevcut operatör sayısı da iki kat olarak alınmıştır.

4.1 Tamsayı Programlama Modeli (Integer Programming Model)

İndisler

- i, i' iş indisleri. $i, i' = 1, \dots, n$
- j, j' makine indisleri. $j, j' = 1, \dots, m$
- t zaman periyodu indisi. $t = 0, 1, \dots, T$
- r operatör indisi. $r = 0, 1, \dots, R$

(0: yapay operatör)

Parametreler

- res_i i . işin ihtiyaç duyduğu operatör miktarı,
 $res_i \in \{0, 1, 2\}$
- R mevcut toplam operatör sayısı
- p_i i . işin işlem süresi
- M_i i . işin işlenebileceği elverişli makine kümesi
- $|M_i|$ i . işin işlenebileceği toplam elverişli makine sayısı
- $neig_{jj'}$ eğer j ve j' fiziksel yakınlığı olan makinelerse 1, aksi halde 0.

Karar değişkenleri

- x_{ijtr} eğer i . iş j . makinede t . zamanda r . operatör ile yapılmaya başlarsa 1, aksi halde 0.
- C_{max} makespan, en son işin tamamlanma zamanı.

Amaç, en son işin tamamlanma zamanını en küçüklemeektir.

$$\min C_{max} \quad (1)$$

(2) no'lu kısıt grubu, her işin bitiş zamanının C_{max} 'tan küçük olacağını belirtmektedir.

$$\sum_{j \in M_i} \sum_{t=0}^{T-1} \sum_{r=0}^R x_{ijtr} (t + p_i) \leq C_{max} \quad i = 1, \dots, n \quad (2)$$

(3) no'lu kısıt grubu, her bir makineye atanan işlerin toplam işlem süresinin C_{max} değerinden küçük olması gerektiğini vurgulamaktadır.

$$\sum_{i=1}^n \sum_{t=0}^{T-1} \sum_{r=0}^R x_{ijtr} p_i \leq C_{max} \quad j = 1, \dots, m \quad (3)$$

(4) no'lu kısıt, operatör başına düşen iş yükünün C_{max} değerinden küçük olması gerektiğini belirtmektedir.

$$\sum_{i=1}^n res_i p_i \leq C_{max} R \quad (4)$$

(5) no'lu kısıt grubu, her işin mutlaka işleneceğini vurgulamaktadır.

$$\sum_{j \in M_i} \sum_{t=0}^{T-p_i} \sum_{r=0}^R x_{ijtr} = 1 \quad i = 1, \dots, n \quad (5)$$

(6) no'lu kısıt grubu, operatör ihtiyacı olan her işin mutlaka bir operatöre atanması gerektiğini göstermektedir.

$$\sum_{j \in M_i} \sum_{t=0}^{T-p_i} \sum_{r=0}^R x_{ijtr} = 1 \quad i = 1, \dots, n \mid res_i > 0 \quad (6)$$

(7) no'lu kısıt grubu, herhangi bir makinede herhangi bir zaman aralığında birden fazla işin işlem göremeyeceğini belirtmektedir.

$$\sum_{t=1}^n \sum_{r=0}^R \sum_{s=\max\{0, t-p_i\}}^{t-1} x_{ijtr} \leq 1 \quad j = 1, \dots, m; t = 0, 1, \dots, T-1 \quad (7)$$

(8) no'lu kısıt grubu, herhangi bir zaman aralığında herhangi bir operatöre kapasitesinden fazla iş atanamayacağını vurgulamaktadır.

$$\sum_{i|res_i>0} \sum_{j \in M_i} \sum_{s=\max\{0, t-p_i\}}^{t-1} res_i x_{ijtr} \leq 2 \quad t = 0, 1, \dots, T-1, r = 1, \dots, R \quad (8)$$

(9) no'lu kısıt grubu, herhangi bir zaman aralığında aynı operatöre atanan iki işin fiziki yakınlığı olmayan makinelerde işlenemeyeceğini belirtmektedir.

$$\sum_{s=\max\{0, t-p_i\}}^{t-1} (x_{ijtr} + x_{i'j'sr}) \leq neig_{jj'} + 1 \quad \forall i, i' \mid res_i = 1 \ \& \ res_{i'} = 1 \ \& \ i' > i, \forall t, r = 1, \dots, R, \forall j, j' \mid j' \neq j, \quad (9)$$

(10) no'lu kısıt grubu, herhangi bir zaman aralığında çizelgelenen işlerin toplam işgücü gereksiniminin mevcut operatör sayısını geçemeyeceğini vurgulamaktadır.

$$\sum_{j=1}^n \sum_{i \in M_j} \sum_{r=1}^R \sum_{s=\max\{0, t-p_i\}}^{t-1} res_i x_{ijtr} \leq R \quad t = 1, \dots, T-1 \quad (10)$$

(11) no'lu kısıt grubu, operatör gereksinimi olmayan işlerin mutlaka tanımlanan yapay operatöre atanması gerektiğini belirtmektedir.

$$\sum_{j \in M_i} \sum_{t=0}^{T-1} x_{ijtr} = 1 \quad i = 1, \dots, n \mid res_i = 0 \quad (11)$$

(12) no'lu kısıt grubu, karar değişkenlerinin 0 ya da 1 değeri alacağını vurgulamaktadır.

$$x_{ijtr} \in \{0, 1\} \quad i = 1, \dots, n; j \in M_i; t = 0, \dots, T-1; r = 0, \dots, R \quad (12)$$

4.2 Kısıt Programlama Modeli (Constraint Programming Model)

Kısıt programlama, son yıllarda tamsayı programlamaya alternatif olarak kullanılmaya başlamıştır. Bir kısıt programlama modeli; bir değişkenler kümesi, bu değişkenler kümesindeki her bir değişkenin alabileceği sonlu sayıda değerleri gösteren bir değer kümesi ve değişkenlere ait kısıtlar şeklinde ifade edilir. Kısıt programlama en iyileme

modeli ise ek olarak en iyilenmesi istenen bir amaç fonksiyonu içermektedir. Kısıt programlama en iyileme modellerinde çözüm yöntemleri, verilen problem kısıtlarını tutarlı duruma getirecek şekilde, değişkenlerin alabileceği değerler kümesini küçülterek olurlu çözümler bulma ve çözümü iyileştirme üzerinde yoğunlaşmaktadır.

Tamsayı programlama ve kısıt programlama yaklaşımları bir dizi çizelgeleme problemi üzerinde uygulanarak karşılaştırılmıştır (örn., [21, 22, 23, 24]). Bu çalışmalarda; tamsayı programlama modellerinin, doğrusal programlama gevşetmesinin (LP relaxation) amaç fonksiyonu için sıkı alt sınır değerleri sağladığı durumlarda (örn. atama ve dağıtım problemlerinde) daha iyi sonuçlar ürettiği; kısıt programlama modellerinin ise çizelgelemenin, sıralamanın ve katı olurluluğun ön planda olduğu kısıt ve problem tiplerinde üstün olduğu sonuçlarına varılmıştır.

Tanımlanan problem için geliştirilen kısıt programlama modeli aşağıda verilmiştir:

Karar değişkenleri:

$start_i$	i işinin başlangıç zamanını gösteren değişken, $start_i \in \{1, 2, \dots, T-p_i\}$
$machine_i$	i işinin atandığı makineyi gösteren değişken, $machine_i \in \{1, 2, \dots, m\}$
C_{max}	makespan, en son işin tamamlanma zamanı

Amaç, en son işin bitiş zamanını en küçükmektir.

$$\min C_{max} \quad (13)$$

(14) no'lu kısıt grubu, her işin bitiş zamanının C_{max} 'tan küçük olacağını belirtmektedir.

$$start_i + p_i \leq C_{max} \quad i = 1, \dots, n. \quad (14)$$

(15) no'lu kısıt grubu, her bir makineye atanan işlerin toplam işlem süresinin C_{max} değerinden küçük olması gerektiğini vurgulamaktadır.

$$\sum_{j \in M_i} (machine_i = j)p_j \leq C_{max} \quad j = 1, \dots, m. \quad (15)$$

(16) no'lu kısıt, her bir operatörün toplam iş yükünün C_{max} değerinden küçük olması gerektiğini belirtmektedir.

$$\sum_{i=1}^n res_i p_i \leq C_{max} R \quad (16)$$

(17) no'lu kısıt grubu, herhangi bir işin kendisine ait elverişli makine kümesinde olmayan bir makineye atanamayacağını göstermektedir.

$$machine_i \neq j \quad i = 1, \dots, n; j \notin M_i. \quad (17)$$

(18) no'lu kısıt grubu, elverişli makine kümesinde sadece bir makine olan işin kesinlikle o makineye atanması gerektiğini göstermektedir.

$$machine_i = j \quad i = 1, \dots, n \text{ ve } |M_i|=1; j \in M_i \quad (18)$$

(19) no'lu kısıt grubu, herhangi bir makinede herhangi bir zaman aralığında birden fazla işin işlem göremeyeceğini göstermektedir.

$$(machine_i = machine_{i'}) \Rightarrow (start_i \geq start_{i'} + p_{i'}) \text{ OR } (start_{i'} \geq start_i + p_i) \quad (19)$$

$$i = 1, \dots, n-1, i' = i + 1, \dots, n.$$

(20) no'lu kısıt grubu, operatör gereksinimi olmayan işlerin yapay operatöre atanması gerektiğini vurgulamaktadır.

$$operator_i = 0 \quad \forall i \mid res_i = 0 \quad (20)$$

(21) no'lu kısıt grubu, operatör gereksinimi olan işlerin yapay operatör dışındaki bir operatöre atanması gerektiğini vurgulamaktadır.

$$operator_i \neq 0 \quad \forall i \mid res_i > 0 \quad (21)$$

(22) no'lu kısıt grubu, tam işçilik gerektiren iki iş aynı operatöre atanmışsa bu işlerin birbiriyle örtüşen zaman aralıklarında işlenemeyeceğini belirtmektedir.

$$(operator_i = operator_{i'}) \Rightarrow (start_i \geq start_{i'} + p_{i'}) \text{ OR } (start_{i'} \geq start_i + p_i) \quad (22)$$

$$i = 1, \dots, n-1, i' = i + 1, \dots, n \mid res_i = 2 \ \& \ res_{i'} = 2$$

(23) no'lu kısıt grubu, yarı otomatik işlem gören iki iş aynı operatöre atanmışsa ve birbiriyle örtüşen zaman aralıklarında işleniyorsa, bu iki işin birbirine komşu makinelere atanmış olması gerektiğini belirtmektedir.

$$(operator_i = operator_{i'}) \text{ AND } [(start_i \geq start_{i'} + p_{i'}) \text{ OR } (start_{i'} \geq start_i + p_i)] \Rightarrow \quad (23)$$

$$neig_{machine_i, machine_{i'}} = 1$$

$$i = 1, \dots, n-1, i' = i + 1, \dots, n \mid res_i = 1 \ \& \ res_{i'} = 1$$

(24) no'lu kısıt grubu, herhangi bir zaman aralığında herhangi bir operatöre kapasitesinden fazla iş atanamayacağını göstermektedir.

$$\sum_{i=1}^n \sum_{s=\max\{0, t-p_i\}}^{t-1} res_i (operator_i = r) (start_i = k) \leq 2 \quad (24)$$

$$t = 0, 1, \dots, T-1, r = 1, \dots, R$$

(25) no'lu kısıt grubu, herhangi bir zaman aralığında işlerin toplam işgücü gereksiniminin mevcut operatör sayısını geçmeyeceğini belirtmektedir.

$$\sum_{i=1}^n \sum_{s=\max\{0, t-p_i\}}^{t-1} res_i(start_i = s) \leq R \quad t=0, \dots, T-1 \quad (25)$$

Yukarıdaki modelden anlaşılacağı gibi, kısıt programlama modeli, tamsayı programlama modeline göre çok daha az sayıda değişken içermektedir. Ayrıca kısıtlarda 've', 'veya', 'eğer...ise' gibi mantıksal ifadelerin kullanılabilmesi problemin tanımlanmasında kolaylıklar sağlamaktadır.

5. HESAPLAMA SONUÇLARI (COMPUTATIONAL RESULTS)

Tamsayı ve kısıt programlama modelleri; 20 iş, 4 makine, 2 operatör; 30 iş, 4 makine, 2 operatör ve 30 iş, 6 makine, 3 operatör test problemlerine uygulanmıştır. Her gruptan 10 olmak üzere toplam 30 problem çözülmüştür. İşlere ait işlem zamanları [4,12] arasında kesikli tekdüze bir dağılımdan türetilmiştir.

Tüm problemler için model çalışma süresi 1000 saniye ile sınırlanmıştır. Önerilen modeller, Intel Core 2 Duo 2.2 Ghz 2 GB RAM özellikli bilgisayarda ILOG OPL IDE 6.3 [25] yazılımı ile çözülmüştür. Tüm test problemlerine ait sonuçlar Tablo 1'de verilmiştir. Tablodaki alt sınır (AS) değerleri aşağıdaki şekilde hesaplanmıştır (bkz. [26, 27]):

$$AS = \text{en büyük} \left\{ \sum_{i=1}^n p_i / m, \sum_{i=1}^n res_i p_i / R \right\} \quad (26)$$

Sapma yüzdeleri dikkate alındığında, küçük boyutlu problemler için tamsayı programlamanın göreceli olarak iyi sonuçlar verdiği; problem büyüklüğü arttıkça ise kısıt programlamanın açık üstünlüğü görülmektedir. Şekil 2'de her bir problem grubu için tamsayı ve kısıt programlamadan elde edilen sonuçların alt sınır değerlerinden sapma yüzdelerinin ortalamaları verilmiştir.

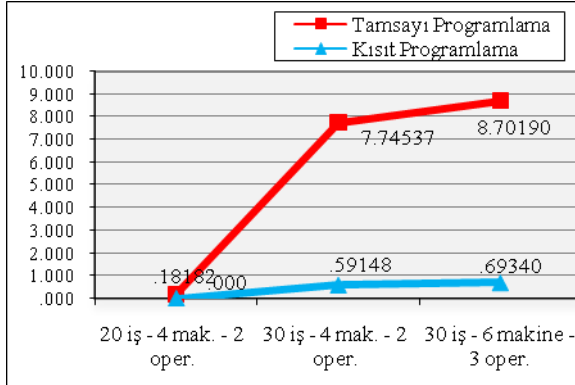
Görüldüğü üzere, problem büyüklüğü arttıkça, tamsayı programlama modeline ait sapma yüzdeleri çok hızlı artmaktayken, kısıt programlamada bu

Tablo 1. Hesaplama sonuçları (Computational results)

N	m	R	Test Problem	Alt Sınır (AS)	Tamsayı Programlama Modeli			Kısıt Programlama Modeli			
					C _{max}	Çözüm Süresi (saniye)	Sapma % (C _{max} -AS)/AS	C _{max}	Çözüm Süresi	Sapma % (C _{max} -AS)/AS	En iyi Sonuç Ulaşma Süresi (saniye)
20	4	2	1	43	43	141,48	0,00	43	15,13	0,00	15,11
			2	45	45	212,06	0,00	45	65,70	0,00	65,69
			3	48	48	503,19	0,00	48	37,05	0,00	37,03
			4	46	46	65,25	0,00	46	6,94	0,00	6,92
			5	43	43	134,06	0,00	43	15,34	0,00	15,34
			6	56	56	105,52	0,00	56	32,89	0,00	32,88
			7	43	43	70,25	0,00	43	5,36	0,00	5,34
			8	55	56	*	1,82	55	24,61	0,00	24,59
			9	39	39	202,63	0,00	39	14,02	0,00	14,02
			10	47	47	200,20	0,00	47	26,91	0,00	26,89
30	4	2	1	62	63	*	1,61	62	129,11	0,00	129,08
			2	60	61	*	1,67	60	528,83	0,00	528,80
			3	66	74	*	12,12	67	*	1,52	199,45
			4	58	63	*	8,62	58	964,75	0,00	964,73
			5	68	69	*	1,47	68	75,13	0,00	75,09
			6	77	86	*	11,69	78	*	1,30	375,55
			7	63	68	*	7,94	63	54,38	0,00	54,36
			8	65	71	*	9,23	66	*	1,54	164,13
			9	64	72	*	12,50	65	*	1,56	538,06
			10	66	73	*	10,61	66	29,95	0,00	29,92
30	6	3	1	41	44	*	7,32	42	*	2,44	57,42
			2	40	45	*	12,50	40	56,73	0,00	56,70
			3	44	49	*	11,36	45	*	2,27	114,03
			4	39	41	*	5,13	39	158,63	0,00	158,59
			5	45	49	*	8,89	46	*	2,22	106,64
			6	45	49	*	8,89	45	856,02	0,00	855,98
			7	42	45	*	7,14	42	222,27	0,00	222,24
			8	41	41	997,06	0,00	41	77,13	0,00	77,11
			9	40	44	*	10,00	40	329,45	0,00	329,42
			10	38	44	*	15,79	38	300,41	0,00	300,39

* 1000 sn. model çalışma limiti aşılmıştır.

artışın hızı çok daha az olmaktadır. Özellikle 30 iş 6 makine problemlerinde kısıt programlamanın göreceli üstünlüğü en üst düzeye çıkmaktadır.



Şekil 2. Sapma yüzdeleri (Percentage of deviations)

Çözüm süreleri dikkate alındığında, tamsayı programlama modelinin 20-iş test problemleri dışındaki problemlerde 1000 sn. çözüm süresi limitini aştığı görülmektedir. Bu açıdan 20-iş test problemleri için çözüm süreleri karşılaştırıldığında, kısıt programlama modelinin ortalama sekiz dokuz kat daha hızlı şekilde çözüme ulaştığı görülmektedir. Tablo 1'deki son sütun, kısıt programlama için bir performans ölçütü olan 'en iyi sonuca ulaşma süresini' göstermektedir. Problem büyüklüğü arttıkça modelin elde ettiği en iyi sonuçlara ulaşma sürelerinin de arttığı görülmektedir.

Son olarak kısıt programlama modelinin gerçek yaşam problemine uygulanabilirliği araştırılmıştır. Gerçek çizelgeleme ortamı verilerini içeren 100 iş, 36 makine, 10 operatör test problemleri kullanılmıştır. Örnek problemlerde, daha önce bahsedildiği gibi her iş bir kalıpla eş değer olarak alındığından; işler, işletmede bulunan 374 adet kalıptan rastgele olarak seçilmiştir. Daha önce üçüncü bölümde verilen varsayımlar gerçek problem için de geçerlidir. İşlere ait işlem zamanları, işletmeden alınan farklı haftalardaki iş emirleri kümelerinin değerlendirilmesi sonucunda [5, 20] arasında kesikli tekdüze bir dağılımdan türetilmiştir. İş - makine uygunluk matrisi ve makinelerin fiziki yakınlıkları gerçek veriler olarak alınmıştır. Makinelere ait yerleşim planı ve makinelerin fiziki yakınlıklarına ait matris ekte

verilmiştir. Tablo 2'de, ele alınan beş farklı örnek probleme ait kısıt programlama sonuçları verilmiştir. Alt sınır değerleri yine 26 no'lu eşitlik kullanılarak hesaplanmıştır. Tüm problemler için modelin çalışma süresi 3600 sn ile sınırlandırılmıştır. Tablodan görüleceği gibi her problem için etkin olurlu çözümler elde edilebilmiş, alt sınırdan sapma yüzdeleri ortalama %7-8 dolaylarında gerçekleşmiştir. Geliştirilen kısıt programlama modelinin gerçek yaşam problemi için oldukça iyi sonuçlar sağlayabildiği görülmektedir.

6. SONUÇ (CONCLUSION)

Bu çalışmada işlerin ve operatörlerin paralel makineler boyunca çizelgenmesini gerektiren bir gerçek yaşam problemi ele alınmıştır. Bu problem için tamsayı ve kısıt programlama modelleri geliştirilmiştir. Kısıt programlama modeli küçük boyutlu test problemlerinde etkin çözümlere hızla ulaşmayı başarmış, gerçek boyutlu problemlerde ise uygulanabilir ve etkin sonuçlar üretmiştir. Ele alınan problem, içerdiği özellikler ve önerilen çözüm yaklaşımları açısından yazına aşağıdaki katkılarda bulunmaktadır:

- Literatürde çok seyrek ele alınan operatörlerin ve makinelerin bir arada çizelgenmesi problemi üzerinde durulmuştur.
- Çizelge oluşturulurken makinelerin fiziki yakınlıkları ilk kez dikkate alınarak, aynı operatörü paylaşacak işlerin komşu makinelere atanması üzerinde durulmuştur.
- Bu derece karmaşık bir probleme ilk kez bir tamsayı programlama modeli geliştirilmiş ve en iyi/olurlu çözümler alınmıştır.
- Etkin bir kısıt programlama modeli geliştirilmiş, orta ve büyük boyutlu problemler için oldukça iyi sonuçlar elde edilmiştir.

Devam eden araştırmada, probleme özgü arama algoritmaları geliştirilerek, kısıt programlamada çözüm süreleri azaltılmaya ve model performansı artırılmaya çalışılacaktır. Ayrıca, atama ve çizelgeleme tipi alt problemlerin sırasıyla tamsayı ve kısıt programlama ile ele alındığı bir ayrıştırma yaklaşımı üzerinde çalışılmaktadır.

Tablo 2. Gerçek boyutlu problemler için hesaplama sonuçları (Computational results for real sized problems)

İş sayısı	Makine Sayısı	Operatör Sayısı	Örnek Problem	Alt Sınır	C_{max}	Çözüm Süresi	Sapma %	En İyi Sonuca Ulaşma Süresi
100	36	10	1	75	79	*	5,33	2346,390
			2	72	78	*	8,33	2680,875
			3	68	74	*	8,82	2132,266
			4	73	79	*	8,22	2086,156
			5	67	74	*	10,45	488,031

* 3600 sn. model çalışma limitine erişilmiştir.

TEŞEKKÜR (ACKNOWLEDGEMENTS)

Bu makalenin bir bölümü, ilk yazar Emrah B. Edis'in TÜBİTAK-2218 programı desteği ile Koç Üniversitesi'nde yürüttüğü doktora sonrası araştırma çalışmasında tamamlanmıştır.

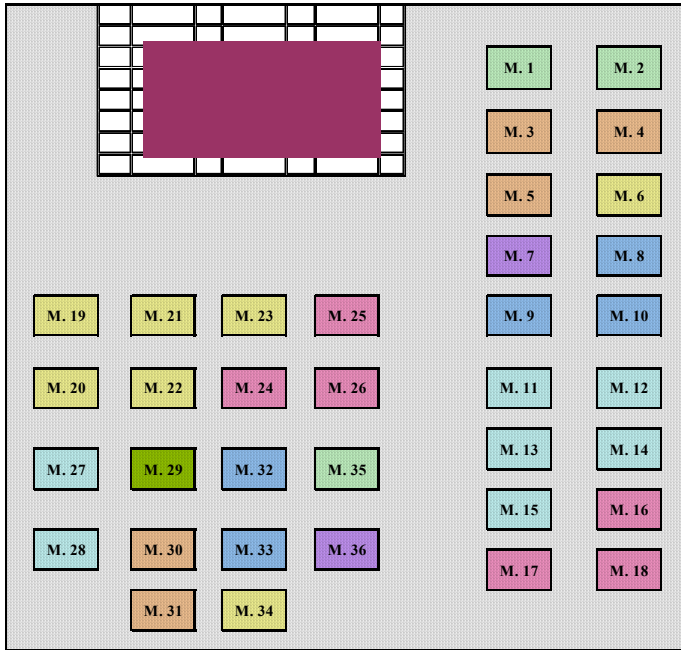
KAYNAKLAR (REFERENCES)

1. Pinedo M., **Scheduling: Theory, Algorithms, and Systems**, 2. basım, Prentice-Hall, Englewood-Cliffs, NJ, A.B.D., 2002.
2. Ventura, J.A. ve Kim, D., "Parallel machine scheduling with earliness-tardiness penalties and additional resource constraints", **Computers & Operations Research**, Cilt 30, 1945–1958, 2003.
3. Daniels, R.L., Hua, S.Y., ve Webster, S., "Heuristics for Parallel-Machine Flexible-Resource Scheduling Problems with Unspecified Job Assignment", **Computers & Operations Research**, Cilt 26, 143-155, 1999.
4. Edis, E.B., Oğuz, C. ve Özkarahan, İ., "Paralel Makinelerde İşlerin ve Makine Operatörlerinin Birlikte Çizelgelenmesi Problemine Çözüm Yaklaşımları", **Yöneylem Araştırması ve Endüstri Mühendisliği 30. Ulusal Kongresi Bildiri Özetleri Kitabı**, Sabancı Üniversitesi, İstanbul, 338, 30 Haziran-2 Temmuz 2010.
5. Edis, E.B., "Resource Constrained Parallel Machine Scheduling Problems with Machine Eligibility Restrictions: Mathematical and Constraint Programming Based Approaches", Doktora Tezi, Dokuz Eylül Üniversitesi, Fen Bilimleri Enstitüsü, Ocak 2010.
6. Daniels, R.L., Hoopes, B.J. ve Mazzola J.B., "Scheduling Parallel Manufacturing Cells with Resource Flexibility", **Management Science**, Cilt 42, No 9, 1260-1276, 1996.
7. Chen, Z., "Simultaneous Job Scheduling and Resource Allocation on Parallel Machines", **Annals of Operations Research**, Cilt 129, 135–153, 1999.
8. Grigoriev, A., Sviridenko, M. ve Uetz, M., "LP rounding and an almost harmonic algorithm for scheduling with resource dependent processing times", **Lecture Notes in Computer Science**, Cilt 4110, Editörler: Diaz, J., Jansen, K., Rolim, J. ve Zwick, U., Springer-Verlag, Berlin, Almanya, 140-151, 2006.
9. Grigoriev, A., Sviridenko, M. ve Uetz, M., "Machine scheduling with resource dependent processing times", **Mathematical Programming Series B**, Cilt 110, 209-228, 2007.
10. Grigoriev, A., Sviridenko, M. ve Uetz, M., "Unrelated parallel machine scheduling with resource dependent processing times", **Lecture Notes in Computer Science**, Cilt 3509, Editörler: Jünger, M. ve Kaibel V., Springer-Verlag, Berlin, Almanya, 182-195, 2005.
11. Hu, P.C., "Minimizing Total Tardiness for the Worker Assignment Scheduling Problem in Identical Parallel-Machine Models", **International Journal of Advanced Manufacturing Technology**, Cilt 23, No 5–6, 383–388, 2004.
12. Hu, P.C., "Minimizing Total Flow Time for the Worker Assignment Scheduling Problem in the Identical Parallel-Machine Models", **International Journal of Advanced Manufacturing Technology**, Cilt 25, No 9–10, 1046–1052, 2005.
13. Chaudry, I.A. ve Drake, P.R., "Minimizing total tardiness for the machine scheduling and worker assignment problems in identical parallel machines using genetic algorithms", **International Journal of Advanced Manufacturing Technology**, Cilt 42, 581–594, 2009.
14. Chaudry, I.A., "Minimizing flow time for the worker assignment problem in identical parallel machine models using GA", **International Journal of Advanced Manufacturing Technology**, Cilt 48, 747–760, 2010.
15. Edis, E.B, Araz, C. ve Ozkarahan, I. "Lagrangian-Based Solution Approaches for a Resource-Constrained Parallel Machine Scheduling Problem with Machine Eligibility Restrictions", **Lecture Notes in Computer Science**, Cilt 5027, Editörler: Nguyen, N.T. vd., Springer-Verlag, Berlin, Almanya, 337-346, 2008.
16. Edis, E. B. ve Ozkarahan, I. "A combined integer/constraint programming approach to a resource-constrained parallel machine scheduling problem with machine eligibility restrictions", **Engineering Optimization**, Cilt 43, No 2, 135-157, 2011.
17. Edis, E.B. ve Ozkarahan, I. "Solution Approaches for a Real Life Resource Constrained Parallel Machine Scheduling Problem", **International Journal of Advanced Manufacturing Technology**, Cilt 58, No 9-12, 1141-1153, 2012.
18. Bourland, K.E. ve Carl, L.K., "Parallel machine scheduling with fractional operator requirements", **IIE Transactions**, Cilt 26, No 5, 56-65, 1994.
19. Zouba, M, Baptiste, P. ve Rebaine, D., "Scheduling identical parallel machines and operators within a period based changing mode", **Computers & Operations Research**, Cilt 36, 3231-3239, 2009.
20. Garey, M.R. ve Johnson, D.S., **Computers and Intractability: A Guide to the Theory of NP-Completeness**, W. H. Freeman and Company, A.B.D., 1979.
21. Darbi-Dowman, K.D., Little, J., Mitra, G. ve Zaffalon, M., "Constraint Logic Programming and Integer Programming Approaches and Their Collaboration in Solving an Assignment Scheduling Problem", **Constraints**, Cilt 1, 245-264, 1997.
22. Smith, B.M., Brailsford, S.C., Hubbard, P.M. ve Williams, H.P., "The Progressive Party Problem: Integer Linear Programming and Constraint Programming Compared", **Constraints**, Cilt 1, 119-138, 1997.

23. Darbi-Dowman, K.D. ve Little, J., "Properties of Some Combinatorial Optimization Problems and Their Effect on the Performance of Integer Programming and Constraint Logic Programming", **Informs Journal on Computing**, Cilt 10, No 3, 276-286, 1998.
24. Lustig, I.J. ve Puget J.F., "Program Does Not Equal Program: Constraint Programming and Its Relationship to Mathematical Programming", **Interfaces**, Cilt 31, No 6, 29-53, 2001.
25. IBM, **ILOG OPL IDE 6.3., User's Manual**, IBM Corp., 2009.
26. McNaughton, R., "Scheduling with deadlines and loss functions", **Management Science**, Cilt 6, No 1, 1-12, 1959.
27. Li, Y., Wang, F. ve Lim A., "Resource constraints machine scheduling: A genetic algorithm approach", **CEC: 2003 Congress on Evolutionary Computation**, Cilt 1-4, 1080-1085, 2003.

EKLER (ATTACHMENTS)

A- Makine yerleşim planı (Machine layout)



B- Makinelere ait fiziki yakınlık matrisi (Physical closeness of the machines)

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15	M16	M17	M18	M19	M20	M21	M22	M23	M24	M25	M26	M27	M28	M29	M30	M31	M32	M33	M34	M35	M36			
M1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
M2	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
M3	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
M4	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
M5	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
M6	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
M7	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
M8	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
M9	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
M10	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
M11	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
M12	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
M13	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
M14	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
M15	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
M16	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
M17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
M18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
M19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
M20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	
M21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
M22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	
M23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
M24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	
M25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	
M26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
M27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	
M28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	
M29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	1	0	1	0	0	0	0	
M30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1	0	0	0	0	
M31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	
M32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0
M33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0
M34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	
M35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
M36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

