

TESİS DÜZENLEMESİ PROBLEMLERİNE KARŞILAŞTIRMALI İKİ YAKLAŞIM: (GRAFTEORİSİ (HAMILTON TEKNİĞİ) VE DAL-SINIR TEKNİĞİ UYGULAMASI)

Orhan TÜRKBEY

*Gazi Üniversitesi, Mühendislik Mimarlık Fakültesi, Endüstri Mühendisliği Bölümü,
Maltepe, 06570, Ankara, Türkiye, Fax: (312) 230-8434
e-mail: turkbey@mmf.gazi.edu.tr,*

ÖZET

Bu çalışmada, mevcut bir tesisin Malzeme Yönetimi Maliyet Giderleri (MYMG)'ni enazlamak ve yerleşim düzenlemesini seçenekli olarak elde edebilmek için birbirinden farklı iki değişik algoritma kullanılmıştır. Bu algoritmalarından ilki, Graf (Serim) Teorisi (GT)'ne dayanan ve en küçük maliyetli Hamilton Çevrimleri (HÇ)'ni esas alan Foulkes Algoritması (FA), ikincisi ise Gezgin Satıcı Problemi (GSP) gibi düşünülen probleme Dal-Sınır Algoritması (DSA) uygulamasıdır. Her iki algoritmanın bilgisayar programı tarafımızdan yazılmış (Quick Basic), paket program kullanılmamıştır. Bir grafta HÇ'nin bulunmasında kullanılan FA'nın, DSA'na nazaran, gerek ve yeter şartlar yerine getirildiğinde daha iyi sonuçlar verdiği görülmüş, elde edilen çözüm neticeleri sonuç bölümünde tartışılmıştır.

Anahtar Kelimeler: Graf teorisi, dal ve sınır tekniği, gezgin satıcı problemi, tesis düzenlemesi

AN APPLICATION OF GRAPH THEORY (HAMILTONIAN TECHNIQUE) AND BRANCH AND BOUND TECHNIQUE ON FACILITY LAYOUT PROBLEMS

ABSTRACT

In this study, two different algorithms have been used to obtain the alternative plant layout in order to minimize the materials handling cost. The first algorithm is the Foulkes Algorithm, which is essentially the Hamiltonian Cycles with minimized cost and is based on Graph Theory. The second one is the Branch and Bound Algorithm, which is considered as the Travelling Salesman Problem. The computer programs of both algorithms have been prepared specifically for the research and no package program has been used. It was observed that the Foulkes Algorithm provides better solutions compared to Branch and Bound Algorithm under necessary and satisfactory conditions. The final solutions are discussed in the last section.

Key Words: Graph theory, branch and bound technique, travelling salesman problem, facilities, layout

1. GİRİŞ

Bir ülkenin kaynaklarının rasyonel ya da verimli bir şekilde kullanılması o ülkenin kalkınmışlığı ile doğru orantılıdır. Kaynaklar olarak nitelenen üretim faktörlerini girdi olarak kullanan işletmeler, üretim faaliyetlerini öyle düzenlemelidir ki aralarında bir uyum olsun. Yani üretim faktörleri olarak nitelenen ve bunların bileşimlerinin de ürün olarak tanımlandığı insan, makina, malzeme ve sermaye faktörleri öyle bir bilgi sentezinden geçirilmelidir ki, rasyonellik ya da verimlilik ilkelerine bağlı kalınsın. İşte bu faktörlerin bileşimi olan ürün oluşturulurken, tesis içindeki hareketlerin yoğunluğu, maliyetin önemli bir kısmını oluşturur. Günümüzde, kaba bir yaklaşımla işletmelerin rekabet gücünü, kalite ve maliyet olarak nitelenen iki ana özellik belirler. Bu iki özellik rekabet ile doğrudan ilişkilidir. İşletmelerde rasyonel olarak yerleşim düzenlemesi model ve tekniklerinin kullanılmasının, verimlilik ve karlılık üzerinde doğrudan etkisinin olduğu aşıkardır.

Olayı ülke bazında önemseyen ülkeler, Tesis Düzenlemesi (TD) Tekniklerine özel bir önem vermektedirler. Genç bir nüfus yapısına sahip olan ülkemizin, bilimsel araştırma ve teknolojik gelişme seviyesinin ne yazık ki arzu edilen seviyeye ulaştığı söylenilemez. Başta sanayi sektörü olmak üzere ülkemizde kaynakların da rasyonel olarak kullanılmadığı da bir gerçektir.

Rasyonel kaynak kullanımını tanımlayan değişkenlere uygulanması gereken model ve teknikler sürekli araştırma konusudur. Yaklaşık 25 yıldır araştırmacılar, iyi bir TD elde edebilmek için çeşitli tesis tasarım algoritmaları geliştirmişlerdir (1). Araştırmaların ilk yıllarında, akış grafları (flow-graphs) (2,3) ve ilişki diyagramları (relationships bubble diagrams) (4) dikkate alınarak çözümlenmeler yapılmıştır. Son zamanlarda ise Tesis Düzenleme Problemlerine (TDP) ağırlıklı olarak; düzlemsel yakınlık grafları (planar adjacency graphs) (5), eşit bazlı yakınlık planları (matching based adjacency graphs) (6), kitlesel merkez yerleşimleri (centroid locations) (7) ve DSA uygulanmaya çalışılmaktadır. Yaklaşım bazlı tüm tasarım algoritmaları, uygulama kısıtlarının özelliği nedeniyle en iyi neticeyi elde edemezler (8, 9). Ancak, problemde değişken sayısının fazla olması nedeniyle, günümüzde bu algoritmalarından vazgeçmekte mümkün değildir.

Tesis planlaması içerisinde yer alan TD'nin, dinamik özelliği nedeniyle düzenleme faaliyetlerinin belirli periyotlarla tekrarlanması gerekir. Söz konusu faaliyetler, üretim sürecinin düzenli ve verimli akışını sağladığı gibi üretim kontrolünde de, önemli kolaylıklar sağlar.

Araştırmamızda, GT'nden yararlanarak halihazır çalışan bir tesis ele alınıp, iki farklı algoritma ile düzenlemesi yapılmış, oluşan MYMG karşılaştırılmış ve sağlanabilecek kaynak tasarrufu araştırılmıştır.

2. GRAF TEORİSİNİN TARİHÇESİ

GT konusunda bilinen ilk çalışma, 1736 yılında Euler tarafından yazılan "The Königsberg Bridge Problem" isimli makaledir (10). Bunu izleyen yıllarda GT konusundaki çalışmalar devam etmiş, 1847 yılında G.Kirchoff elektrik devreleri üzerine, 1852 yılında İrlanda'lı matematikçi W.R.Hamilton ve De Morgan kendi adı ile anılan "Hamilton Çevrimleri" isimli çalışmayı, 1987 yılında A. Cayley kombinatorik ağaç konusunda, 1900'lü yıllarda ise J.Sylvester ve D.König konuyla ilgili önemli çalışmalar yapmışlardır (11).

1900-1920 yılları arasında, F.W.Taylor'un bilimsel yönetimin temel kurallarını ortaya koyması sonucu, bugünkü anlamda organizasyon, verimlilik, TD gibi konulara sanayide GT uygulamaları ivme kazandırmıştır.

TD problemlerinde GT'ne dayalı detay formülasyonlar; Fould ve Moore tarafından incelenmiş, Fould ve Robinson ise DSA'nı kullanarak TD problemlerinin çözümünde GT'nden yararlanmışlardır (12,13).

3. GRAF TEORİSİNDE KULLANILAN ÇEVİRİMLER

GT'nde çevrim kavramı çok önemlidir. Çevrimin özelliğine bağlı olarak, graf uygulamalarında farklılık oluşur. Bu nedenle gerek Hamilton gerekse Euler çevrimleri literatürde yoğun olarak

incelenmiştir. Tesis düzenleme problemlerinde HÇ çok önemlidir.

HÇ, yöneylem araştırmasında, rota arama problemlerinde, GSP'de, hangi üretken birimlerin hangi sırada yerleştirileceğini belirleyen sıralama problemlerinde, TDP'de en az maliyeti araştırmak amacıyla kullanılabilir. HÇ'nin bulunmasında aşağıdaki kurallar uygulanır (14,15):

Kural 1: Eğer x düğümünün çıkış derecesi 2 ise, x'deki iki yay, HÇ'nin bir parçası olmalıdır.

Kural 2: Özel olmayan bir alt çevrim ya da bütün değerleri içine almayan çevrim, bir HÇ'ini oluştururken kurulabilir.

Kural 3: Oluşturulan HÇ'nin bir x düğümünden geçtiği varsayalım. Sonradan gelen tüm diğer yaylar x'de devre içinde daha sonra kullanılmadıkları sürece silinebilirler. Burada yayları silerken, diğer düğümlerin derecesini en çok 2'ye indirebiliriz ve bu durumda da kural 1 uygulanır.

Hamilton Çevrimi ile ilgili Teoremler (2, 9, 20):

Teorem 1: n sayıda düğüme sahip olan yönlendirilmiş bir grafa, yönlendirilmiş bir Hamilton yolu vardır.

Teorem 2: Varsayalım ki G grafi, n düğümlü bir bağlı graf olsun ve bu düğümler x_1, x_2, \dots, x_n olarak dizilsin. Bu durumda $\deg(x_i) \leq \deg(x_{i+1})$ olmalıdır. Eğer bütün $k \leq n/2$ ve $\deg(x_k) > k$ veya $\deg(x_{n-k}) \geq (n-k)$ ise, G bir HÇ'ne sahiptir.

Teorem 3: Örneğin bir G düzlemsel grafi, HÇ'ne sahip olsun. Yine G herhangi bir düzlemsel şekilde tanımlansın. r_i ve \hat{r}_i 'nin sırasıyla, bu tanım içinde i yayları ile sınırlandırılmış ve sınırlandırılmamış HÇ'deki bölgeleri gösterdiği varsayalım. Bu takdirde r_i ve \hat{r}_i sayıları aşağıdaki denklemi sağlar.

$$\sum_{i=3}^p (i-2)(r_i - \hat{r}_i) = 0, \text{ (Grinberg'in Teoreminden) (2, 11)}$$

q $3p - 6$ eşitsizliği, bazı grafların düzlemsel olmadığını ispat etmede kullanılmaktadır. Ancak bu teorem, bazı düzlemsel grafların HÇ'ne sahip olmayabileceğini söylemektedir (16, 17, 11).

Minimum maliyetli HÇ'nin bulunması problemi, literatürde "GSP" olarak geçmektedir. Bu problemde, olmayan yayların maliyetinin sonsuz olduğu kabul edilir. Hamilton grafini, kavram olarak Euler Grafi'na benzeyen bir tür olarak kabul edebiliriz.

4. FOULKES ALGORİTMASI

Önceden de belirtildiği gibi, uygulamamızda iki farklı algoritma kullanılmıştır. Bunlar; TD problemi, GSP gibi düşünülerek bir DSA uygulaması ve GT içinde önemli bir yere sahip FA uygulamasıdır. Biz uygulamamızda baz çalışma olarak ikinci algoritmayı seçip, bunun sonuçlarını birinci algoritma ile karşılaştırmak istediğimizden ayrıca, FA'nı inceleme gereği duyulmuştur.

FA, bir graf içerisinde mevcut çok connex (grafın tüm düğümleri arasında en az bir çevrim varsa) alt grafların ayrıştırılması için kullanılan yaklaşık bir yöntemdir. Bir grafın her x_i düğümü, herhangi bir x_j düğüme bir yay ile bağlandığında, elde edilen graf tam ve simetriktr. Grafta, x_i düğümleri arasındaki, bağıntı eşdeğerliğine (c) göre bir değer verilirse, x_i 'den x_j 'ye bir yön vardır denir.

$$x_j \in \hat{\Gamma} x_i \text{ ve } x_i \in \hat{\Gamma} x_j \text{ dir.} \quad [1]$$

$$\hat{\Gamma} x_i = \{x_i\} \cup \Gamma x_i \cup \Gamma^2 x_i \cup \dots \quad [2]$$

x_i kapalı geçişli iken;

$$\text{Yansıma (reflexive)} \quad : \forall x_i \in c ; x_i \equiv x_j \quad [3]$$

$$\text{Simetriklik (symmetrical)} \quad : \quad \forall x_i, x_j \in c \ ; \ |x_i \equiv x_j| \Rightarrow |x_j \equiv x_i| \quad [4]$$

$$\text{Geçiş (transitive)} \quad : \quad \forall x_i, x_j, x_k \in c \ ; \ |x_i \equiv x_j| \ \mathbf{ve} \ |x_j \equiv x_k| \Rightarrow |x_i \equiv x_k| \quad [5]$$

Acaba x_i 'den x_j 'ye en fazla ne kadar uzunlukta bir yol bulunabilir? sorusunu cevaplayabilmek için, Booléiennes matrisiyle (cebri ile) çarpımının yapılması gerekir (10).

$M_k = (M) (M) \dots (M)$ matrisini elde edebilmek için M 'i kendisi ile k defa çarpmak gerekir. Eğer x_i 'den x_j 'ye giden k uzunluğunda bir yol varsa, matrisin bu c_{ij} (k) bağlantısı "1" değerini alacaktır. Booléiennes hesaplama tekniği (grafın düğümleri arasındaki uzaklığı maliyet cinsinden ifade eden "etkinlik matrisi"nden elde edilecek "işlem matrisi"nin oluşturulmasında kullanılan, Booléiennes matrisiyel çarpımındaki özel matematiksel işlemi ifade eder) ile;

$$[(1) + (M)]^2 = (1) + (M) + (M)^2, \quad [6]$$

$$[(1) + (M)]^r = (1) + (M) + (M)^2 + \dots + (M)^r, \quad [7]$$

$$x_j \in \{x_i\} \cup \Gamma x_i \cup \Gamma^2 x_i \cup \dots \cup \Gamma^r x_i. \quad [8]$$

ifadeleri elde edilir. [8] ifadesinin anlamı, x_i ile x_j arasında r kadar alt uzunlukta bir yol olduğunu göstermektedir. $d_{ij}(r)$ bağlantısı matriste "1" değerlidir, aksi durumda "0" değerini alacaktır.

N düğümlerinin sayısı ise, $N-1$ sayıda alt yol olabilecektir. Çünkü, iki düğüm arasında "1" yolu mevcuttur ve indirgeme için hesaplama ard arda yapılır. Sonuçta çok connex alt graflar oluşur ve işlem durdurulur (18).

4.1 Uygulama Algoritması

Adım 1: Başlangıç maliyet (etkinlik) matrisi değerlerini oku,

Adım 2: Eğer; $c_{ij} < c_{ji}$ ise, $r_{ij} = 1$, ($i = j$ durumunda),

$c_{ij} > c_{ji}$ ise $r_{ij} = 0$, ($i = j$ durumunda),

$i = j$ ise, $r_{ij} = 0$ veya $r_{ij} = 1$ al ve işlem matrisini oluştur,

Adım 3: ($n-1$) defa matrisin kendi kendisi ile çarpım işlemi yap. Her çarpım sonunda,

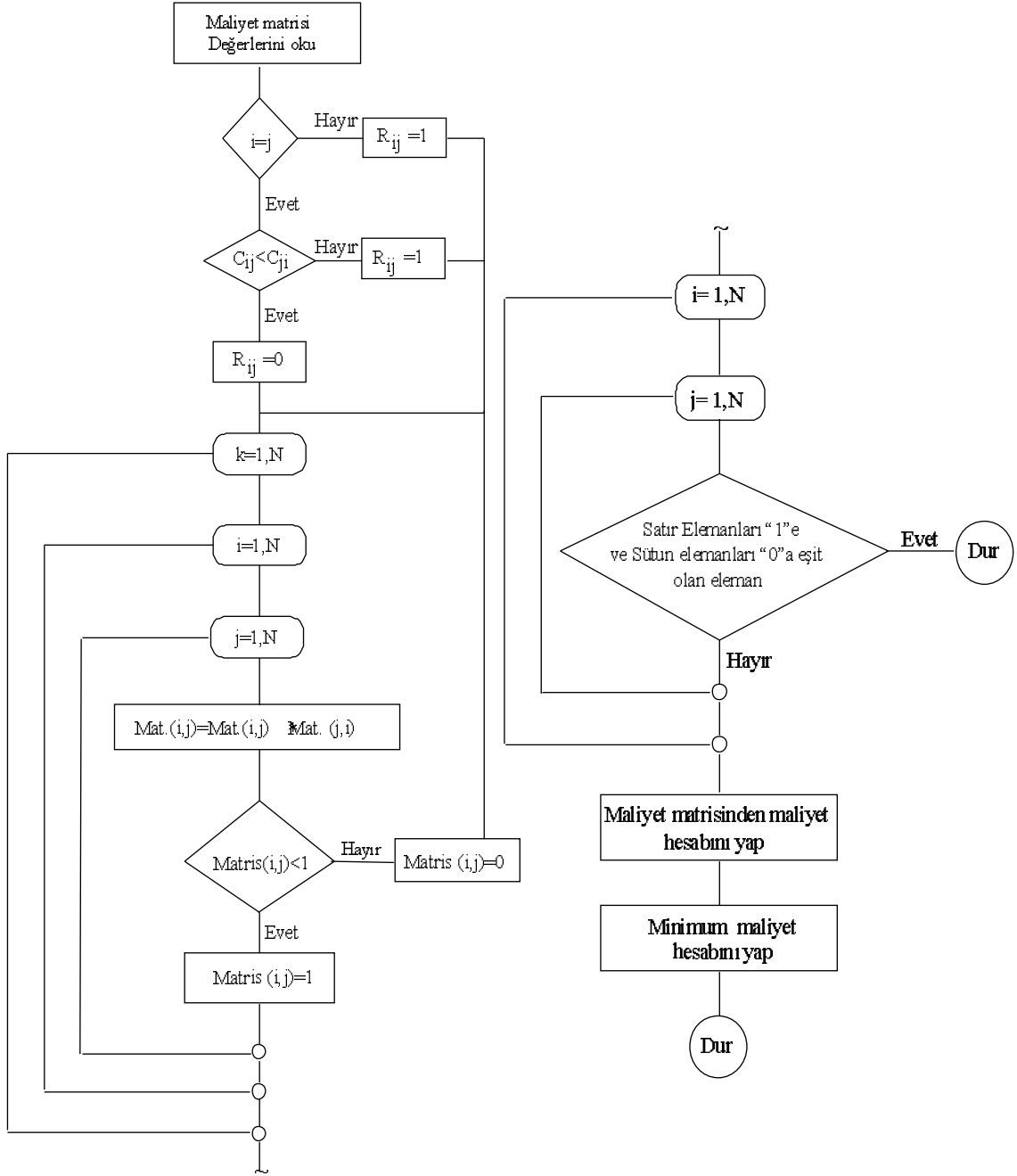
Eğer $a \geq 1$ ise, çarp (i, j) = 1

Eğer $a < 1$ ise, çarp (i, j) = 0 al ve çözüm matrisini oluştur.

Adım 4: Çözüm matrisinde satır elemanları "1"e, sütun elemanları "0"a eşit olan elemanı bul ve bu satır ve sütunu matristen çıkart. Bu şartı sağlayan eleman yoksa işlemi durdur.

Adım 5: Bulunmuş olan yörüngelere ait maliyet hesabını, maliyet matrisinden yap ve minimum maliyeti seç. Bu yörünge en düşük maliyete sahip yörüngedir.

4.2. Algoritmanın Akış Şeması



Şekil 1. Foulkes algoritması program akış şeması.

5. UYGULAMA

5.1 Genel Veriler

Uygulama yeri olarak, telli ve telsiz haberleşme sistemlerinden sorumlu bir kamu kurumunun ağaç işleri imalat atölyesi seçilmiştir. Tesis içinde 13 işlem merkezi/tezgah olup, işleme göre konumlandırılmışlardır. Üretim ise siparişe göre yapılmaktadır. Önce tesisin üretim planı incelenmiş, ürünler ve alt ürünler, bunların işlem sıraları belirlenmiş ve tesisin geçmişteki verileri de dikkate alınarak, üretim ve talep projeksiyonları yapılmıştır. Elde edilen projeksiyon verilerine göre tezgahların yükleri (malzeme akış miktarları) hesaplanmıştır. Söz konusu bilgiler Çizelge 1 ve 2'de özetlenmiştir.

Çizelge 1. tesisler ve kod numaraları

Kod	Tesisler	Kod	Tesisler
01	Şerit testere	08	Freze
02	Planya	09	Dopel tezgahı
03	Daire testere Freze	10	Toma
04	Planya	11	Özel planya
05	Daire testere	12	Özel toma
06	Freze	13	Band zımpara
07		-	-

Çizelge 2. Parça üretim ve taşıma miktarları, işlem sıraları (Üretim ve taşıma miktarları x 1000)

Parça adı	Üretim mik. (unite/dön.)	Taşıma miktarı (kg/dön.)	İşlem sırası
1. Masa yan paneller.	4.88	284.7	(1,3,6,4,7,9,13)
2. Masa üst tabla.	2.44	177.9	(1,3,6,4,7,13)
3. Masa ön panel.	2.44	113.5	(1,3,6,4,13)
4. Çekmece kutusu yan ken.	2.44	61.8	(1,3,6,4,13)
5. Çek. kutu. alt-üst parça	14.65	31.3	1,3,6,4,13)
6. Çekmece ön-arka parça	14.65	22.9	(1,3,6,4,13)
7. Çekmece çıtalrı.	14.65	18.4	(1,2,11,12,5,3,6,13)
8. Masa ayakları.	9.86	16.9	(1,2,11,12,5,3,6,13)
9. Çekmece alın parçası.	7.32	22.8	(1,2,11,8,10,13)
10. Kulp.	7.32	0.3	(6,9,10,4)
11. Ejeer yan paneller.	4.88	61.9	(1,3,6,4,7,9)
12. Ejeer alt-üst tabla	7.32	439.3	(1,3,6,4,8)
13. Ejeer ön kapaklar.	4.88	77.9	(1,3,6,4)
14. Ejeer arka panel.	2.44	38.9	(1,3,6,7)
15. Elbise askısı ana kaide.	0.40	3.7	(3,10,9)
16. Elbise askısı alt tabla.	0.80	2.4	(6,10,8)
17. Elbise askısı alt tabla.	0.80	2.4	(6,10,8)
18. Pencere çerçevesi.	0.40	181.4	(1,3,6,5,7,8,9,10,11)
19. Sehpa ana kaidesi.	0.60	4.8	(3,4,10,11)
20. Sehpa ayakları.	1.20	4.9	(3,8,5,9)
21. Sehpa tablası.	0.60	7.2	(3,6,4,12)

Çizelge 2'den yararlanarak, Çizelge 3'deki yük matrisi oluşturulur.

Çizelge 3. Yük matrisi (kg/dön.) (103).

Tesisler	1	2	3	4	5	6	7	8	9	10	11	12	13
1	-	5.1	149.3	-	1.2	-	-	-	-	-	1.2	1.2	-
2	-	-	1.1	-	-	1.1	-	1.3	-	1.3	2.4	1.1	3.9
3	-	-	-	1.2	-	269.9	-	1.2	-	3.3	-	-	2.3
4	-	-	-	-	-	-	29.6	87.9	-	0.6	0.6	1.8	64.3
5	-	1.1	1.1	-	-	1.1	4.0	4.0	4.6	4.0	4.0	1.1	-
6	-	-	3.3	158.9	20.2	-	47.3	21.5	17.6	2.6	-	-	2.2
7	-	-	-	-	-	-	-	30.4	30.4	5.0	5.0	-	49.1
8	-	1.3	-	-	0.6	-	30.4	-	30.4	6.3	6.3	-	21.7
9	-	-	-	-	-	-	30.2	30.2	-	5.1	5.0	-	13.4
10	-	1.2	-	0.1	-	-	5.0	8.9	6.2	-	6.9	-	5.7
11	-	2.4	1.1	-	1.1	1.1	5.0	6.3	5.0	6.9	-	-	3.8
12	-	-	1.1	-	1.1	1.1	-	-	-	-	1.1	-	1.1
13	-	-	-	-	-	-	-	-	-	-	-	-	-

Çizelge 4. Trafik matrisi (sıklık/dön.)

Tesisler	1	2	3	4	5	6	7	8	9	10	11	12	13
1	-	18	498	-	4	-	-	-	-	-	4	4	-
2	-	-	4	-	-	4	-	5	-	5	9	4	13
3	-	-	-	5	-	900	-	5	-	11	-	-	8
4	-	-	-	-	-	-	99	293	-	3	3	6	215
5	-	4	4	-	-	4	14	14	16	14	14	4	-
6	-	-	11	530	68	-	158	72	59	9	-	-	8
7	-	-	-	-	-	-	-	102	102	17	17	-	164
8	-	5	-	-	3	-	102	-	102	22	22	-	73
9	-	-	-	-	-	-	101	101	-	18	17	-	45
10	-	5	-	1	-	-	17	30	21	-	23	-	20
11	-	9	4	-	4	4	17	22	17	24	-	-	13
12	-	-	4	-	4	4	-	-	-	-	4	-	4
13	-	-	-	-	-	-	-	-	-	-	-	-	-

Atölye içinde taşınan parçaların zarar görmesi ve hacimlerinin büyük olması nedeniyle, bir taşıma frekansında 300 kg taşıyabilen bir transpalet kullanılmaktadır. Bu nedenle yük matrisi elemanlarının her biri 300'e bölünerek Çizelge 4'deki trafik matrisi düzenlenir. Bu matrisin elemanları tam sayılı olup, kesirli taşıma yapılmamaktadır.

Algoritmada ikinci ana veri olarak maliyet matrisinin oluşturulması gerekir. Bu matris oluşturulurken, önce Toplam Akış Miktarı (TAM) hesaplanır.

$$TAM = \sum_{i=1}^n \sum_{j=1}^n (Uzaklık)_{ij} (Trafik)_{ij} \quad (i = j = 1, 2, \dots, n) \quad [9]$$

Birim taşıma maliyeti ($c = TL/m$) ve Toplam MYMG ise;

$c = TMYMG (TL/dönem) / \text{Toplam Akış} (m/dönem)$ olacaktır. Bu durumda,

$c = 11\,279\,998\,400 / 83\,807,038 = 1\,345,949 TL/m$ bulunur (değişken giderlere göre).

Daha sonra uzaklık ve trafik matrislerinin basit çarpımı sonunda (C)'nin elemanları hesaplanır. Maliyet matrisinin (C) her elemanı; TMYMG'nin bir iş istasyonundan diğer bir iş istasyonuna

taşınacak yük miktarı ve iki iş istasyonu arasındaki uzaklığa bağlı olarak dağıtılmasından elde edilmiştir. TMYMG'ni, tesis içinde bir dönemdeki amortisman, yakıt, bakım-onarım ve işçilik giderleri oluşturmaktadır.

Çizelge 5. Maliyet matrisi (C) (x 106 TL) (TMYMG = 11 279 998 400 TL/dönem)

Tesisler	1	2	3	4	5	6	7	8	9	10	11	12	13
1	-	12.2	214.7	-	10.7	-	-	-	-	-	14.1	16.4	-
2	-	-	1.1	-	-	11.9	-	5.0	-	8.7	23.2	13.0	44.7
3	-	-	-	8.1	-	31063	-	5.8	-	25.1	-	-	28.1
4	-	-	-	-	-	-	170.1	842.5	-	5.2	4.0	10.2	740.5
5	-	7.4	8.5	-	-	4.2	27.1	11.6	57.7	54.1	65.7	20.2	-
6	-	-	37.8	1371.5	77.3	-	136.0	144.5	286.7	35.4	-	-	36.6
7	-	-	-	-	-	-	-	116.5	407.6	53.1	48.3	-	659.8
8	-	4.9	-	-	1.7	-	116.8	-	291.9	66.5	84.6	-	332.5
9	-	-	-	-	-	-	405.6	289.7	-	14.7	29.0	-	103.0
10	-	8.3	-	0.6	-	-	53.1	93.5	17.9	-	19.7	-	27.4
11	-	23.1	11.9	-	18.4	14.1	48.3	84.8	29.0	19.9	-	-	18.4
12	-	-	13.8	-	20.2	15.9	-	-	-	-	3.2	-	8.5
13	-	-	-	-	-	-	-	-	-	-	-	-	-

Şimdi de uzaklık matrisinin hesaplanması gerekir. Çizelge 6, Şekil 3'de (ek 1) görülen başlangıç yerleşim planında iş merkezleri arasındaki zigzaglı (rectilinear) uzaklıkların hesaplanması ile bulunmuştur. Yerleşim planında iş merkezlerinin koordinat değerleri hesaplanmış, $d_{ij} = x_i - x_j + y_i - y_j$ zigzaglı uzaklık ifadesinden yararlanılmıştır.

Çizelge 6. Uzaklık matrisi (m)

Tesisler	1	2	3	4	5	6	7	8	9	10	11	12	13
1	-	2.5	1.5	8.5	9.5	13.5	10.5	6.5	6.5	9.5	12.5	14.5	14.5
2		-	1.0	6.0	7.0	11.0	8.0	4.0	6.0	7.0	10.0	12.0	12.0
3			-	7.0	8.0	12.0	9.0	5.0	5.0	8.0	11.0	13.0	13.0
4				-	13.0	9.0	6.0	10.0	12.0	9.0	7.0	6.0	12.0
5					-	4.0	7.0	3.0	13.0	14.0	17.0	19.0	19.0
6						-	3.0	7.0	17.0	14.0	13.0	15.0	17.0
7							-	4.0	14.0	11.0	10.0	12.0	14.0
8								-	10.0	11.0	14.0	16.0	16.0
9									-	3.0	6.0	8.0	8.0
10										-	3.0	5.0	5.0
11											-	3.0	5.0
12												-	8.0
13													-

5.2. Foulkes Algoritması ile Tesis Düzenlemesi Uygulama Çıktıları

Önceki bölümde söz konusu algoritma ile ilgili teorik bilgiler ve uygulamada kullanılacak iki farklı algoritma için gereken genel bilgiler verilmişti.

FA, ana girdi olan başlangıç verilerini kullanmış ve HÇ yardımıyla, alt graflara ait Hamilton

yörüngeleri ile bu yörüngelerin dolaşım maliyetlerini hesaplamış ayrıca, en küçük maliyete sahip yörüneyi de bulmuştur. Bulunan yörüneye en az dolaşım maliyetini kapsadığından, üretilecek ürünlerin üretim sıra kombinasyonları esas alınarak, işlem merkezlerinin bu sırayı dikkate alacak şekilde düzenlenmesi gerekmektedir. Şekil 4'de (ek 3) ise, FA'na göre tesisin yeni yerleşim planı ve TMYMG görülmektedir.

5.3 Gezgin Satıcı Problemlerine Dal-Sınır Tekniği Uygulanması Şeklinde Düşünülen Tesis Düzenlemesi ve Çıktıları

GSP literatürde geniş olarak incelenmiştir. Söz konusu problemin çözümüne yönelik bir çok teknik ya da algoritma geliştirilmiştir. Bunlardan birisi de GSP'nin DSA ile çözümüdür. Burada tesis üretim faaliyetleri, bir GSP gibi düşünülerek, DSA ile çözümlenmeye yani düzenleme problemi çözülmeye çalışılmıştır. Anılan problemlere ait teorik bilgileri, konuyla ilgili pek çok yayında görmek mümkündür (19, 20). Bu nedenle araştırmamızda ayrıca DSA ve akış şemasına yer verilmemiştir.

Uygulamamızın bu kısmında problem, 5.1 bölümdeki genel veriler DSA kullanılarak, yine Quick Basic programlama dilinde 14 iterasyonda ve 168 sn'de çözülmüştür. Söz konusu algoritma ile elde edilen yerleşim planı Şekil 5'de (Ek 5) görülmektedir.

6. SONUÇ VE DEĞERLENDİRME

Çalışmamızda, mevcut bir tesiste düzenleme problemini çözümlenebilmek için, önceden de belirtildiği gibi "birbirinden farklı iki değişik algoritma" kullanılmıştır. İlki "graf teorisine dayanan ve en küçük maliyetli Hamilton çevrimlerini esas alan Foulkes Algoritması", ikincisi ise "Gezgin Satıcı Problemi gibi düşünülen örnek probleme Dal Sınır Algoritması" uygulamasıdır. Çözümde, bazı tesisler arasında ilişki olmadığından, bu ilişkiler sonsuz maliyetli olarak düşünülmüştür. Bu nedenle oluşan graf, tam bir graf (conneks graph) değildir.

Gerek başlangıç düzenleme planı gerekse her iki uygulama algoritması sonucunda elde edilen düzenleme planları tek boyutludur (Çizelge 7, Şekil 2). Ancak elde edilen bu düzenleme planları kesikli bir üretim şeklinde gerçekte iki boyutlu olacağından, söz konusu çıktı planları düzenleme tasarımcısı tarafından çeşitli kriterler dikkate alınarak iki boyutlu şekle dönüştürülmüştür (Şekil 3,4,5).

Yine çalışmamızda, her iki algoritma kullanılarak, bir taraftan tesis birimleri arasındaki akış ençoklanmaya çalışılırken, diğer taraftan malzeme dolaşım uzaklığı enazlanmaya çalışılmış, sonuçta "işgücü" ve "zaman" gibi iki önemli kaynaktan tasarruf sağlanmıştır. Uygulamada kullanılan her iki algoritmanın bilgisayar uygulama programı tarafımızdan yazılmış, paket program kullanılmamıştır. Bunun nedenleri; bul (Booléiennes) cebirinin özelliği, alfabetik notasyonların matematiksel işlemlerinin bilgisayar diline uyarlanabilmesi, uygulamada kullanılan problemin özelliği, ve algoritmaların aynı bazda karşılaştırılabilmesidir.

Her iki algoritmanın, mevcut yerleşim planına uygulanması sonucu elde edilen yeni yerleşim planı karşılaştırmalı olarak Çizelge 7'de görülmektedir.

Çizelge 7. Elde edilen düzenleme planına göre tesislerin karşılaştırmalı atanma şekilleri

Başlangıç düzenleme planına göre tesisler	01	02	03	04	05	06	07	08	09	10	11	12	13
Foulkes Algoritmasına göre tesis atamaları	03	12	06	11	10	04	09	07	13	08	02	05	01
Dal Sınır Alg.na göre tesis atamaları	03	12	06	13	11	04	10	09	07	08	02	05	01

Çizelge 8. Uygulamada kullanılan algoritmaların performans karşılaştırması

	Dolaşım Uzaklığı (m/dön.)	İşgücü Kullanımı (gün/dön.)	Sağlanan Tasarruf (gün; 10 ² TL; m)	CPU Time (saniye)
Başlangıç düzenlemesi	79 052	300.0	-	88
Foulkes Algoritmasına göre düzenleme planı	69 064	262.1	37.9 13 443 344 9 988	75
Dal Sınır Algoritmasına göre düzenleme planı	77 788	295.2	4.8 1 701 280 1 264	68

Foulkes algoritması kullanılarak, ürünlerin işlem sıralarına göre en küçük maliyete sahip olan çevrimin sırası (tesis düzenleme sırası) şöyle olacaktır:

13	1	3	6	4	11	2	12	5	10	8	7	9
----	---	---	---	---	----	---	----	---	----	---	---	---

Şekil 2. Foulkes algoritması kullanılarak bulunan tesis düzenlemesi.

Yine en küçük maliyete sahip olan çevrimin sırasına bağlı en düşük maliyet (TMYMG) 11.279.998.400 TL, bilgisayar CPU süresi ise 75 saniye olmaktadır.

On üç işlem merkezine sahip tesiste, uygulamada kullanılan algoritmaların performans karşılaştırması Çizelge 8'de görülmektedir. Tablodan görüleceği gibi Foulkes algoritması, Dal-Sınır algoritmasından, bilgisayar işlem zamanı hariç, gerek dolaşım uzaklığı gerekse işgücü kullanımı açısından daha iyi netice vermektedir. Söz konusu tabloda; sağlanan tasarruf, işgücü kullanımı, TMYMG ve dolaşım uzaklığı yönünden her iki algoritmanın performans sonuçları arasındaki fark büyüktür. Ayrıca Foulkes algoritmasının bir diğer üstünlüğü; Dal-Sınır algoritmasında tesis sayısında bir üst sınır (en fazla 15 tesis) olmasına rağmen, ilkinde gerek ve yeter şartlar ile bilgisayar kapasitesi uygun olduğu takdirde bu sınırın olmamasıdır.

Ülkelerin kalkınmışlık göstergelerinden biri ve bizce en önemlisi, girdilerin rasyonel kullanımınıdır. Bu nedenle ülkemiz işletmelerinin yerleşim düzenlemelerinin yapılmasında, üretime konu teşkil eden üretim planları ile üretim süreçlerinin esas alınması gerekir. Aksi durumda kaynak israfı ortaya çıkacaktır. Çünkü sonradan yapılacak yerleşim planı değişiklikleri, çok zor olduğu kadar imkansız da olabilir.

Semboller listesi

- a_{ij} : Çözüm matrisinin elemanı.
- c : Herhangi bir özelliğe göre birbirlerine denk oldukları düşünülen bütün elemanlar kümesi.
- c_{ij} : Maliyet matrisi elemanı.
- $d_i(x)$: Düğümün iç derecesi.
- $d_i(x_i)$: i düğümünün iç derecesi ($= d(x_i) = \text{deg}(x_i)$).
- $d_i(x_0)$: Düğümün dış derecesi.
- $d_i(p)$: V 'deki düğümlerin sayısının iç derecesi.
- $d_i(q)$: E 'deki kenarların sayısının dış derecesi.
- $d_f(x_i)$: i . Düğümün son derecesi.
- $d_0(p)$: V 'deki başlangıç düğümlerinin sayısı.

- $d_0(q)$: E'deki başlangıç kenarlarının sayısı.
 $d(x_i)$: i düğümünün iç derecesi (= deg (x_i)).
 $E(G)$: Kenarlar/yaylar kümesi.
 $e \in E$: Kenarlar kümesinin e elemanı.
 G : Graf.
 $G = (V, E)$: Düğümler (V) ve kenarlar (E) alt kümesi olarak tanımlanan sonlu ve boş olmayan küme.
 Γ : Düğümler kümesi (= V (G)).
 Γx_i : Düğümler kümesinde i. düğümü.
 $\hat{\Gamma}$: Tanımlanmış bir düğümden geçen bir alt grafın kenarlarının kümesi.
 $\hat{\Gamma} x_j$: Tanımlanmış bir düğümden geçen bir alt graftaki j.düğümün kenarlarının kümesi.
 HC : Hamilton çevrimi (G'deki sabit Hamilton çevrimi).
 i, j : Düğümlerin tanımlayıcıları.
 k : Hamilton çevriminde herhangi bir düğüm.
 M : Düğümler kümesi matrisi.
 n : Düğümler kümesinin eleman sayısı (= p).
 p : G'nin sırasında ya da V'deki düğümlerin sayısı.
 q : G'nin büyüklüğünde ya da E'deki kenarların sayısı.
 r : Bağlantılı x_i ve x_j düğümlerinin oluşturduğu yörüngeyi sağlayan M matrisinin kuv.
 r_{ij} : İşlem matrisinin elemanı (= ilişki matrisi).
 r_i : Hamilton çevrimi içinde yer alan i kenarlarının tamamını kapsayan G grafindaki bölgelerin sayısı.
 \hat{r}_i : Hamilton çevrimi dışında yer alan i kenarlarının tamamını kapsayan G grafindaki bölgelerin sayısı.
 $V(G)$: Düğümler kümesi.
 $v \in V$: Düğümler kümesi elemanı.
 $x \in G$: x, G grafinin bir düğümü.
 x_0 : Keyfi olarak seçilen bir başlangıç düğümü.

$$\left[\begin{array}{l} 1+1 = 0+1 = 1+0 = 1 \\ |a|+|b| = |a+b| \end{array} \right] : \text{Booléiennes matrisiyel çarpımı.}$$

KAYNAKLAR

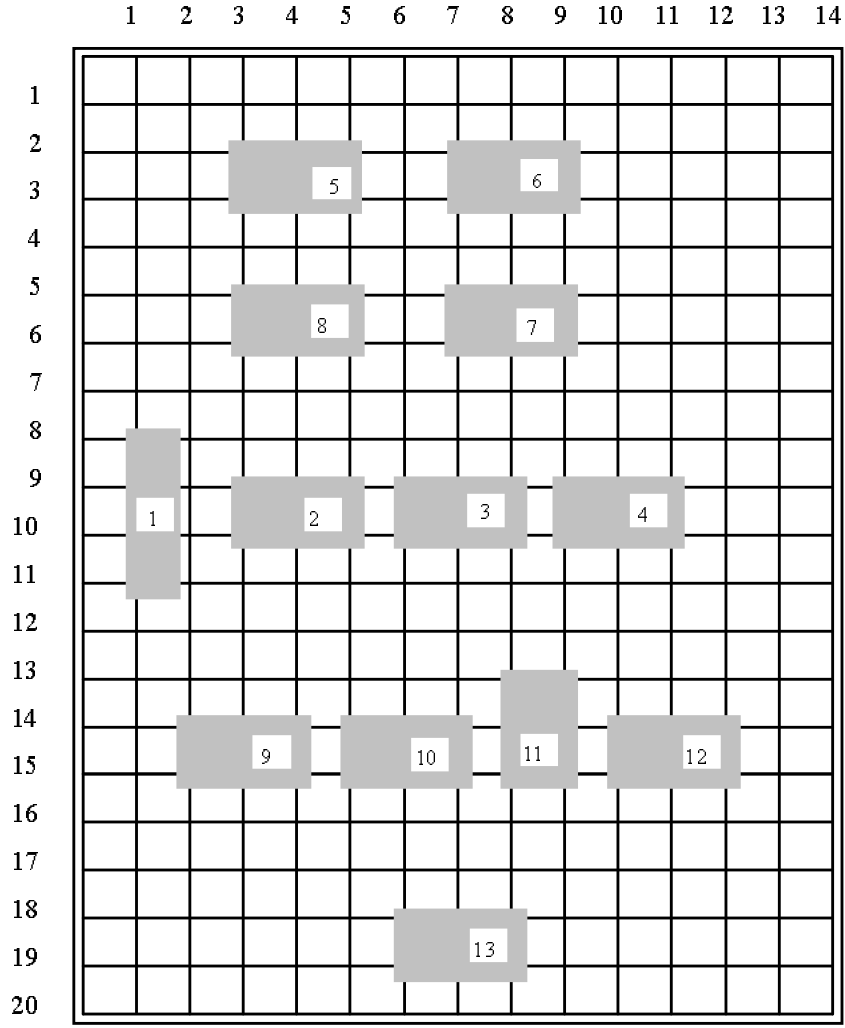
1. Baybars, I. and Eastman, C.M., "Enumerating architectural arrangements by generating their underlying graphs", *Building and Environment*, 7(2): 289-310 (1980).
2. Bondy, J.A. and Murty, U.S.R., "Graph theory with applications", North Holland, Amsterdam (1976).
3. Broersma, H.J. and Veldman, H.J., "Around three lemmas in hamiltonian graph theory", in: Topics in Combinatorics and Graph theory, *Physica-Verlag*, Heidelberg (1990).
4. Carrie, A.S., Moore, J.M., Rocznik, R. and Seppanen, J.J., "Graph theory and computer aided facilities design", *Omega*, 6(4): 353-361 (1978).
5. Clark, J. and Holton, D.A., "A first look at graph theory", *World Scientific Pub.Co.Pte.Ltd.*, Singapore (1991).
6. Drezner, Z., "DISCON: A new method for the layout problem", *Operations Research*, 28(6): 1375-1384 (1980).
7. Foulds, L.R. and Robinson, D.F., "Graph theoretical heuristic for the plant layout problem", *I. J. of Prod. Res.*, 16(1): 27-37 (1978).
8. Hammouche, A. and Webster, D.B., "Evaluation of an application of graph theory to the layout problem", *I. J. of Prod. Res.*, 23(5): 987-1000 (1985).
9. Harary, F., Graph theory, *Addison Wesley Publishing Co.*, Boston (1972).

10. Kaufmann, A., "Methodes et modèles de la recherche operationnelle (Les Mathematiques de l'entreprise)", **Dunod**, Paris (1968).
11. Krasikov, I. and Roditty, Y., "A Las-vergnas type theorem for trees", in: Topics in Combinatorics and Graph Theory, **Physica-Verlag**, Heidelberg (1990).
12. Lal, T.H. and Sprague, A., "Performance of parallel branch and bound algorithms", **IEEE Transactions on Computations**, 14(10): 79-81 (1985).
13. Levin, P.H., "Use of graphs to decide the optimum layout of buildings", **The Architects' J. of Infrastructure Library**, 4: 809-815, 1223-1224 (1964).
14. Little, J.D.C., Murty, K.G., Sweeney, D.W. and Karel, C., "An slgorithm for the travelling salesman problem", **Operations Research**, 11(6): 972-989 (1963).
15. Montreuil, B., Ratliff, H.D. and Goetschalckx, M., "Matching based interactive facility layout." **IIE Transactions**", 19(3): 271-279 (1987).
16. Muther, R., "Systematic layout planning (2nd ed.)", **Cahners Books**, Boston (1974).
17. Reed, R., "Plant Layout: factors, principles and techniques", Richard D. Irwin Inc., Homewood, **Illinois**, (1961).
18. Roth, J., Hashimshory, R. and Wachman, A., "Turning a graph into a rectangular floor plan", **Building and Environment**, 17(3): 163-173, (1982).
19. Tompkins, J.A., "Modularity and flexibility: Dealing with future shock in facilities design", **Industrial Engineering**, 12(3): 78-81 (1980).
20. Tucker, A., "Applied combinatorics", **John Wiley and Sons Inc.**, New York (1980).

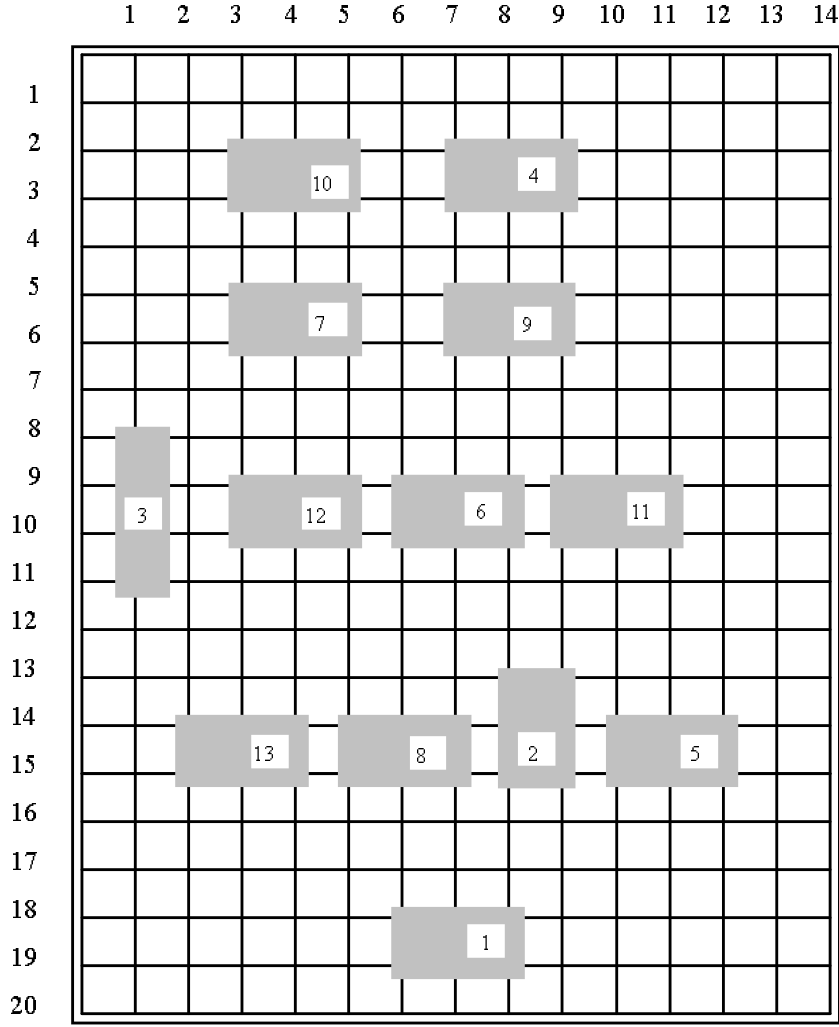
Geliş Tarihi:17.10.2001

Kabul Tarihi:05.11.2002

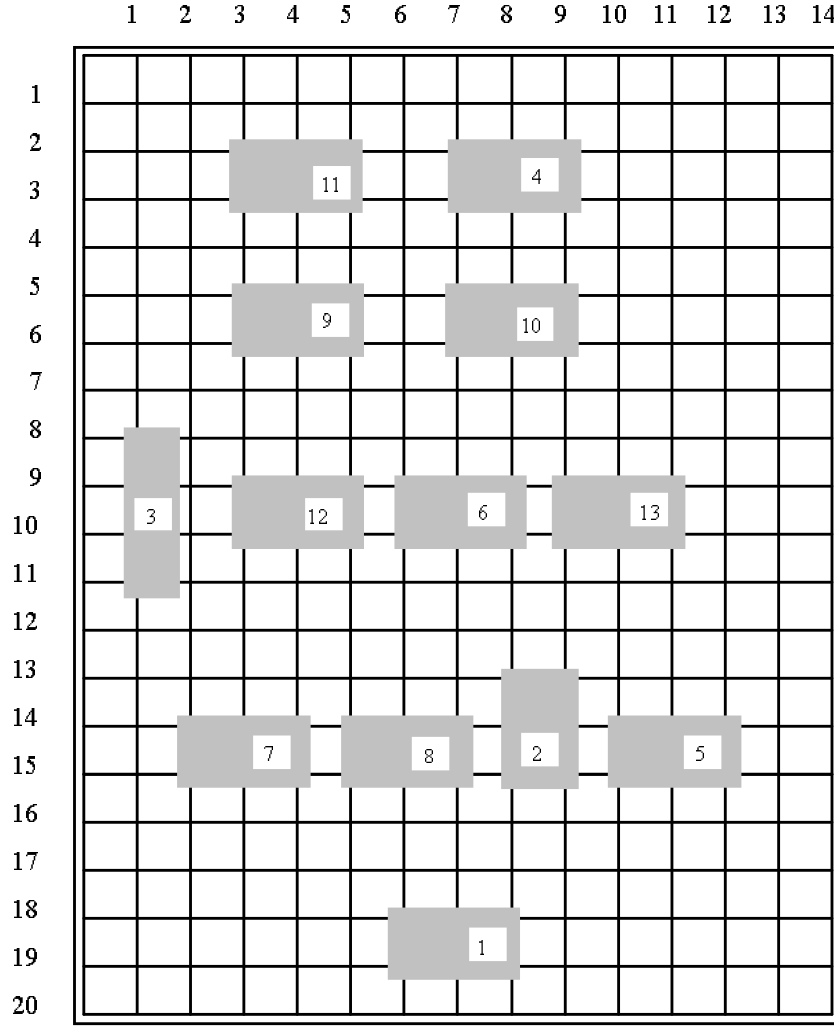
Ekler



Şekil 3. Başlangıç düzenlemesi.



Şekil 4. Foulkes algoritmasına göre elde edilen yeni düzenleme.



Şekil 5. Dal- Sınır algoritmasına göre elde edilen yeni düzenleme.

