

Ağ Diyagramları İle Proje Planlamasında Karşılaşılan Üç Sorun-Üç Algoritma

Orhan Yüksel¹

ÖZET

Bildiride, işlemlerin oklarla tanımlandığı proje planlama serimlerinde karşılaşılan üç önemli sorun için geliştirilen algoritmalar tanıtılacaktır.

Serimlerde oluşan döngülerin matris kullanılmadan elde edilmesi ilk algoritmayı oluşturmaktadır.

Düğümlerin, bellekte büyük yer kaplamasını engelleyen bir yöntemle, topolojik sıralanması ikinci algoritmayı ve işlemlerin değişik amaçlarla sıralanmasında, adres hesabı sıralamaya benzer bir yöntem üçüncü algoritmayı oluşturmaktadır.

Bellek kullanımı ve çalışma süresi açısından üç algoritma da önemli kazanımlar getirmektedir.

1. GİRİŞ

Proje planlamasında kullanılan ağ diyagramlarında işlemler, bilindiği gibi, ya ayrıtlarla ya da düğümlerle tanımlanmaktadır. Burada işlemlerin

¹Prof., EÜ. Mühendislik Fakültesi İnşaat Mühendisliği Bölümü, 35100, Bornova, İzmir.

ayrıntlarla tanımlandığı diyagramlarda karşılaşılan sorunlardan üçüne ilişkin önerilen algoritmalar tanıtılacaktır.²

İlk sorun, serim hazırlanırken oluşabilecek döngülerdir. Döngü bulmak için düğüm sayısı boyutunda bir kare matris kullanılabilir (Özsu,1986). Düğüm matrisi (adjacency matrix, vertex incidence matrix) olarak da adlandırılan (Ceyhun,1976; Busacker & Saaty,1965) bu matrisin boyutu özellikle kişisel bilgisayarlarda problemin bellekte çözümünü güçleştirmektedir. 2000 düğümlü küçük bir serimde döngü aramak için $2000^2 \times 2 = 8MB$ 'lık yere gereksinilir ve daha serimin hazırlık aşamasında çıkmaza girilir. Aynı serim (aşağıda görüleceği gibi) 2000 elemanlı iki vektörle de çözülebilir ve bu yaklaşım bellekte $2000 \times 2 \times 2 = 8KB$ yer kaplar ki bu da ilk çözümün 1/1000 i olmaktadır.

İkinci problem ayrıtların başına ve kuruğuna konulan numaraların topolojik olarak ($i < j$ olacak şekilde) verilmesidir. Böylece düğüm zamanlarının hesaplanması oldukça kısılacaktır. Topolojik sıralama ya da ranking olarak adlandırılan bu işleme ilişkin ilk algoritma olarak genelde Fulker-son (1962) gösterilir (Moder & Phillips,1970). Aşağıda önerilen yöntem, başlangıçta gelişigüzel numaralandırılmış bir serim, düğüm sayısı boyutunda bir vektör kullanarak topolojik olarak sıralanacaktır.

Söz konusu algoritmalar, pekçok yüksek lisans ve doktora çalışmasında (Eyeci,1984; Banar,1987; Kocatürk,1987; Özdemir, 1988; Öztuna,1988; Ulusoy,1992; Pusat,1997) denenmiştir.

Sonuncu sorun sıralamadır. Planlamada milyonlara ulaşabilen işlemler serimler ve bunların defalarca sıralanması; çıktı elde ederken, kaynak atarken, optimum süreyi bulmak için uygun ayrıt ararken ya da en basitinden düğüm zamanlarını bulurken işlemlerin, defalarca ve değişik şekillerde sıralanması gerekmektedir. Kullanılan bilgisayar ne kadar hızlı olursa olsun, bellek ne kadar gelişmiş olursa olsun, sözü edilen işlemlerin süresi gözardı edilemeyecek kadar büyük olmaktadır. Proje planlamalarında paket program kullananlar, işlemler 5-10.000 sayısını geçince so-

² Burada anlatılacak olanlar, kendi planlama programını hazırlayanlara yardımcı olsun, sorunlarını daha hızlı ve daha az bellek kullanarak çözsün diye verilmektedir. Geleneksel düğüm esaslı ya da ok esaslı tekniklerin tartışması konu dışında bırakılmış olup, anlatılanların düğüm esaslı planlama teknikleri savunucularına bir üstünlük sağlamaya-çağı, yazarın uygulamalarından aldığı sonuçlara göre kesin olarak söylenebilir.

runun ne olduğunu iyi bilirler Hele kaynak tahsisi problemi de çözümün içinde düşünülecekse (Banar, 1987; Özdemir, 1988; Ulusoy,1992; Pusat, 1997) bilgisayarın çalışma süresinin önemi daha da artmaktadır.

Sıralamaya ilişkin bilgilere bilgisayar kitaplarında (Thesen,1978; Scheid, 1982) rastlanabilir. Dilimize kazandırılan bir eser (Martin,1973) temel bir kaynak niteliğindedir. Sıralamada amaç, işlemin olabildiğince kısa sürede ve az bellek kullanarak yapılmasıdır (Topçu,1985). Aşağıda ok diyagramlarıyla planlamada kullanılmak üzere özel olarak geliştirilmiş bir teknik (Yüksel,1986) tanıtılacaktır.

2. SERİM HAZIRLANIRKEN KARŞILAŞILAN SORUNLAR

2.1 Döngü Bulma

Serimlerin boyutları büyüdüğünde karşılaşılan ilk sorun döngü oluşmasıdır. Bu, genelde iki şekilde olabilir;

- Bir işlemin bitişi kendisinden önce biten işlemlerden birisinin başlangıç düğümüne bağlanır.
- Serim doğru çizilmiştir, ancak, aynı numara birden fazla düğüme verilmiştir.

Her iki durum da aynı sonucu verir. Serimlerin kullanıldığı diğer alanlarda (optimizasyon problemleri) modelin bir parçası olabilen döngüler proje planlamasında sorun olur, hesaplar sonsuz döngüye girer ve oluşumuna izin verilmez.

Hazır yazılımlarda aranan ilk koşullardan birisi programın döngü taraması yapıp yapmadığıdır.

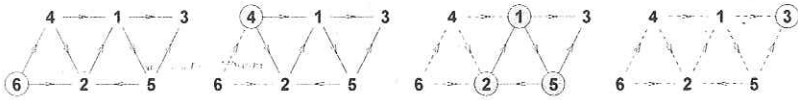
Serimlerde kendisinden ayrıtların çıktığı ve hiç bir ayrıtın girmediği düğümlere *kaynak düğümü*, *başlangıç düğümü* (source node) denilir. Aşağıda açıklaması yapılacak algoritmada *serimdeki başlangıç düğümlerini bulup bunları ve bunlardan çıkan ayrıtları elimine ederek yeni başlangıç düğümleri oluşturmakta ve aynı eliminasyonu bunlar için de yaparak bitiş düğümüne* (sink node, terminal node) ulaşılmaktadır.

İşlemi tanımlayan i, j notasyonu (başlangıç ve bitiş düğümleri) aşağıda da aynı anlamda kullanılacaktır.

- 1.A vektöründe j düğümleriyle tanımlanan elemanlar işaretlenir. A vektöründeki eleman sayısı P sayacına saydırılır.
- 2.A(i) işaretli ise B(j) de işaretlenir. B vektöründeki eleman sayısı Q sayacıyla saydırılır. $Q=0$ olursa algoritma sona erer.
- 3.B vektörünün elemanları A'ya aktarılır. B vektörü sıfırlanır. Eğer $P \neq Q$ ise $P=Q$ ve $Q=0$ alınır ve 2. adıma dönülür. Aksi halde $Q=0$ alınır.
- 4.A vektöründe işaretli ilk satır bulunur. B vektörünün aynı satırı işaretlenir.
- 5.A(i) ve B(j) işaretli ise B(i) de işaretlenir. B vektöründeki eleman sayısı Q sayacı ile sayılır.
- 6.Eğer $P < Q$ ise $P=Q$ alınır, Q sıfırlanır ve 5.nci adıma dönülür.
- 7.Döngülerin sayısını belirten R sayacı bir artırılır. B vektöründe işaretli satırlar R .nci döngünün düğümlerini gösterir. Bu satırlar A ve B vektöründe sıfırlanır.
- 8.B vektörü sıfırlanır, $P=0$, $Q=0$ yapılır. Varsa izleyen döngüleri bulmak için 2. adıma dönülür.

Algoritmanın uygulaması 9 işlemlilik küçük bir serimde (Tablo.1) görülmektedir.

Tablo.1 Serimlerde döngü bulunması



	1	2	3	2	3	4	5	6	5	7	2	
	A	B	A	B	A	B	A	B	A	B	A	B
6-4	1	0	1	1	1	0	1	1	1	1	1	1
5-3	1	0	1	1	1	0	1	1	1	1	1	1
2-1	1	0	1	1	1	0	1	1	1	0	1	0
1-5	1	0	1	0	0	0	0	0	0	0	0	0
4-2	1	0	1	1	1	0	1	1	1	1	1	1
1-3	0	0	0	0	0	0	0	0	0	0	0	0
6-2	P	5	-	5	-	4	-	4	-	4	-	3
4-1	Q	-	0	-	4	-	0	-	4	-	0	-
5-2	R	-	-	-	-	-	-	-	-	-	-	-

2.2 Topolojik Sıralama

İşlemler, kendileri için çizilen okların baş ve sonundaki düğümlerin numaralarıyla tanımlanır. Düğüm no.ları ok yönünde artan ve ardışık şekilde tanımlanmışsa düğüm zamanları bilgisayarda kolay ve hızlı hesaplanır. Uygulamada düğüm sayısı birkaç yüzü geçen serimlerde, düğümlerin yukarıda tanımlanan düzende numaralandırılması pek mümkün olamaz, düğümler gelişigüzel ve ardışık olmayan bir düzende numaralandırılabilir. Numaraların gelişigüzel tanımlandığı serimlerde düğüm zamanlarının bilgisayar tarafından hesaplanması iki şekilde yapılabilir;

1. Düğüm numaraları ardışık ve artan şekilde değiştirilir (*topolojik sıralama ranking*), sonra hesaplar yapılır.
2. Düğüm numaraları değiştirilmez, hesaplar bu haliyle yapılır.

İlk yaklaşımda tanımlanan bir algoritmaya göre bilgisayar düğüm numaralarını değiştirir. Bu, izleyen hesapları da oldukça kısaltır. Diğerinde hesaplama oldukça uzun sürer.

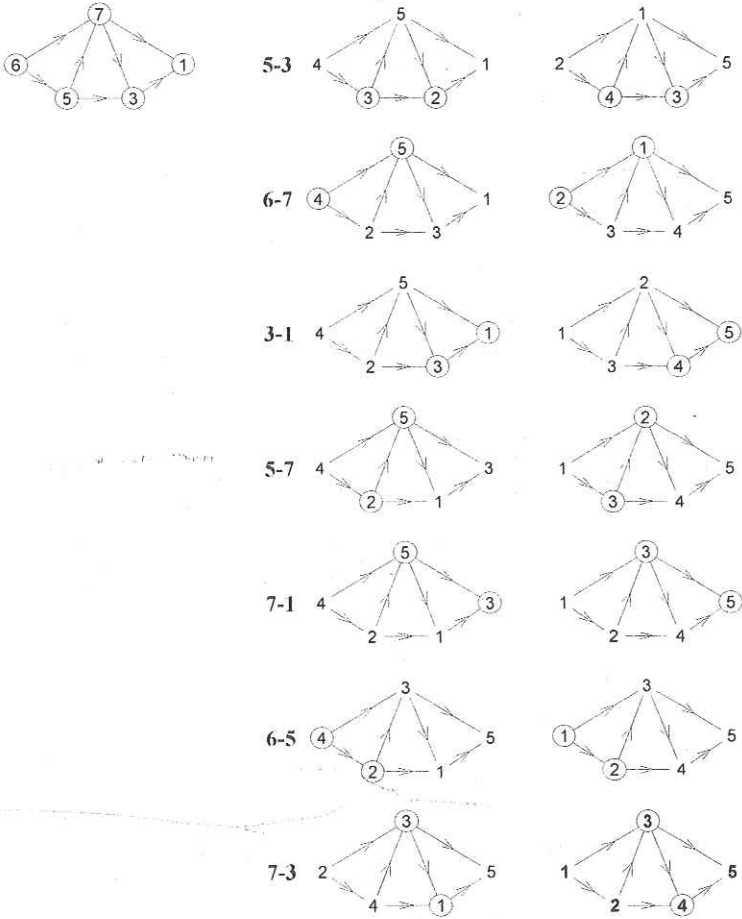
Büyük serimlerde düğümler gelişigüzel numaralandırılıp daha sonra yeniden düzenlenir ve düğüm zamanlarının hesaplanması ancak bundan sonra yapılır.

Düğümlerin düzenli olabilmesi, işlemi tanımlayan okun ucundaki düğüm no.sunun kuyruğundakinden büyük olması ile tanımlandığına göre ($i < j$) kullanılacak yöntem de bunu sağlayabilmelidir. Aşağıda *okun başındaki düğümün sonundakinden küçük olduğu durumlarda ($i > j$), kullanıldığı bütün işlemler için geçerli olmak üzere bu numaraların yerlerini değiştirme kuralına dayanan bir yöntem* tanıtılmaktadır.

1. Bir A vektöründe bütün işlemler için i ve j .nci satırlar işaretlenir.
2. Vektörde işaretli satırlara 1 den (ya da istenilen bir sayıdan) başlanarak ardışık numaralar (daha sonra düğümlerin yeni numaraları olarak kullanılacak o -an) konur.
3. İşlemden $A(i) > A(j)$ olması halinde vektörün i .nci satırındaki sayı ile j .nci satırındaki değiştirilir. Bütün işlemler için bu yapılır.
4. Vektörün elemanlarında değişiklik yapılmışsa 3. adıma dönülür.
5. Her işlemin i ve j düğümleri vektörün ilgili satırındaki değer olarak yeniden atanır.

İşlemler j düğümlerine (eşit olması halinde i düğümlerine) göre küçük-ten büyüğe doğru sıralanmışsa hesaplama süresi hızla azalmaktadır. Algoritmanın 7 işlemlerle bir serimde uygulaması Şekil.1'de görülmektedir.

				5-3	6-7	3-1	5-7	7-1	6-5	7-3	5-3	6-7	3-1	5-7	7-1	6-5	7-3
5-3	1	1	1	1	1	3	3	5	5	5	5	5	5	5	5	5	5
6-7	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
3-1	3	1	2	3	3	1	1	1	1	3	4	4	4	4	4	4	4
5-7	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7-1	5	1	3	2	2	2	2	2	4	4	3	3	3	2	2	2	2
6-5	6	1	4	4	4	4	4	4	2	2	2	1	1	1	1	1	1
7-3	7	1	5	5	5	5	5	3	3	1	1	2	2	3	3	3	3



Şekil.1 Topolojik sıralama için önerilen algoritmanın çalışması

2.3 Sıralama

İşlemlerin oklarla tanımlandığı proje planlaması serimlerinde, her işlemde, i kuyruk ve j baş düğümü bulunmaktadır. Burada kullanılacak algoritmada işlemler i düğümüne ve onların da eşit olması durumunda j düğümüne göre küçükten büyüğe doğru sıralanacaktır. Sıralamada i düğümlerinin ilk sütunda, j düğümlerinin de ikinci sütunda yer aldığı, işlem sayısı uzunluğunda bir A matrisi ve işlemlerin A matrisinde bulunduğu yerleri tanımlamakta yararlanılacak, en büyük düğüm no.su uzunluğunda bir B vektörü kullanılacaktır. Matris ve vektörün tüm elemanları tam sayıdır.

1. B vektörünün, A matrisinin birinci sütununda belirtilen, i düğümünce tanımlanan satırındaki, değerler bir artırılır (Tablo.2, sütun 2).
 2. B vektörünün ilk elemanından başlanarak elde edilen kümülatif değerler satır değeri olarak alınır. Bu değerler bir artırılmış olarak bir sonraki sıraya yazılır. İlk elemanın değeri 1 olarak alınır (Tablo 2, 3. sütun)
 3. A matrisindeki ilk işlem alınır, p değişkeninde saklanır. Bulunulan satır s değişkenine atanır.
 4. B vektöründe işlemin i düğümüne karşı gelen elemanın değeri p değişkeninde saklanan işlemin A matrisinde olması gereken yeri gösterir (satır no.su x olsun). Bu değer s değişkenindeki değere eşitse A matrisinin bir sonraki satırına gidilir, s değişkeni gelinen satırı göstermek üzere değiştirilir. B vektöründe bu satırın değeri 1 artırılır. s değişkeninin değeri A matrisinin son elemanını gösteriyorsa işlem sona erer.
 5. A matrisinin x . satırındaki işlem q değişkenine alınır. Eldeki işlem bu satıra yerleştirilir. Bu satırda aynı i düğümü ile başlayan başka bir işlem varsa, yerleştirilecek elemanın j düğümü bulunan ile kıyaslanır, küçük olan yukarı alınır. Kıyaslama ve yer değiştirme üstteki işlemin i düğümü aynı olduğu sürece devam eder.
- Eldeki işlem p değişkenine aktarılır. Aynı işlem sürdürülmek üzere 4. adıma gidilir.

Algoritmanın 17 işlemlerli bir örneğe uygulaması Tablo.2'de görülmektedir. A matrisinin ilk satırındaki 8-10 işlemi ($i=8$) B vektörünün 3. Sütundaki bilgiye göre 15. sırada bulunması gerekmektedir. A matrisinin 15. satırındaki 6-9 işlemi alınır ve yerine bu işlem, 8-10, yerleştirilir. 8 düğümüyle başlayan bir diğer işlemin yeni adresi artık 16 olacaktır (B vektörü). Aynı işlemler 6-9 ve izleyen elemanlar için yapılır.

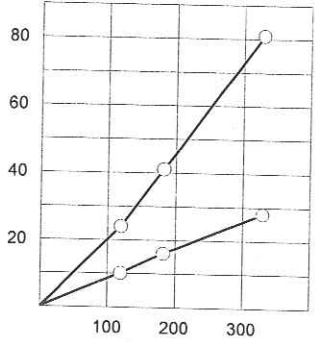
Tablo.2 A matrisi ve B vektöründe algoritmanın akışı

A matrisi				B vektörü			
1	2	3	4	1	2	3	4
1	8-10	1-4,1-3,1-2	3,7,11	1	4	1	2,3,4,5
2	6-8	1-4,1-3	7,11	2	1	5	6
3	7-9	1-4	11	3	1	6	7
4	3-5	1-5	15	4	2	7	8,9
5	2-4		16	5	2	9	10,11
6	4-6	3-5	12	6	2	11	12,13
7	5-7	4-7,4-6	9,13	7	2	13	14,15
8	9-10	4-7	13	8	2	15	16,17
9	1-2	5-7	10,17	9	1	17	18
1	5-6	5-7	17				
0							
1	1-4	6-9,6-8	2,4				
1							
1	7-8	6-9	4				
2							
1	8-9	7-8	5				
3							
1	4-7	7-9	8				
4							
1	6-9	8-10,8-9	1,6				
5							
1	1-3	8-10	6				
6							
1	1-5	9-10	14				
7							

Algoritmanın değişik serimlerde Quick Sort 2 sıralaması ile yapılan kıyaslaması Şekil.2'de görülmektedir³ (Yüksel,1986).

³ Araştırmanın yapıldığı zamandaki bilgisayarın kapasitesi nedeniyle süre olarak verilen değerler oldukça yüksektir (OY).

		İşlem sayısı		
		119	183	330
Quick sort 2	Süre	24	41	81
Önerilen yöntem		10	16	28



Şekil.2 Sıralama sürelerinin grafik görünümü

3. SONUÇ

Serimlerle planlamalar proje tipi üretimlerde vazgeçilmez araç olarak kullanılmaktadır. Yukarıda verilen üç algoritma ile işlemlerin oklarla tanımlandığı proje planlama serimlerindeki üç soruna, süre ve kapasite yönünden kazanım getirilmeye çalışılmıştır.

Bilgisayarlar donanım olarak ne kadar gelişmiş olurlarsa olsunlar, hız ve kapasite yönünden, projelerin büyüklükleri ve planlanmak istenen kaynakların karmaşıklığını çözecek boyutlara ulaşması olanaksızdır. Proje planlaması yazılımlarıyla uğraşanların, yazdıkları her satırda, bilgisayar kapasitesini ve hesaplama süresini azaltacak çözümleri aramaları ve bu sorunları gözden uzak tutmamaları gerekmektedir.

YARARLANILAN KAYNAKLAR

Banar, S., 1987, İnşaat Yatırımlarında Sınırlı Kaynak Kullanımı, Anadolu Üniversitesi Fen Bilimleri Enstitüsü, Eskişehir, 106s, (yüksek lisans tezi).

- Busacker, R.G. and Saaty, T.L.**, 1965, Finite Graphs and Networks and Introduction With Applications, McGraw Hill, 294p.
- Ceyhun, Y.**, 1976, Çizge Kuramı, ODTÜ Mühendislik Fakültesi Elektrik Mühendisliği Bölümü Yayın No:54. 265s.
- Eyeci, Y.**,1985, Birden Fazla İnşaat İlgili İşgücü Planlaması, Anadolu Üniversitesi Fen Bilimleri Enstitüsü, Eskişehir, 86s, (yüksek lisans tezi).
- Fulkerson, D.R.**, 1962, Expected Critical Path Lengths in PERT Networks, Operations Research, Vol.10, No.6, pp,808-817.
- Kocatürk, N.**,Cost Scheduling in Construction Projects Using CPM, Middle East Technical University, 132p, Ankara, (MS thesis).
- Martin, W.**, 1973, Elektronik Hesap Makinaları İle Sıralama, Çeviri: N. Yücel, İTÜ Elektronik Hesap bilimleri Enstitüsü Yayınları: 15, İstanbul, 44s.
- Moder, J.M. and Phillips, C.R.**, 1970, Project Management with CPM and PERT, Van Nostrand Reinhold Company, 360p.
- Özdemir, İ.**, 1988, Ödemelerin Periyodik ve Değişken, İşgücü ve Makine-gücünün Sabit Kısıtlı Olduğu Yatırımlarda En Uygun Yatırım Süresinin Bulunması, Anadolu Üniversitesi Fen Bilimleri Enstitüsü, Eskişehir, 139s, (doktora tezi).
- Özsu, M.T.**, 1986, Proje Planlama ve Denetim Teknikleri, Türkiye Bilişim Derneği Yayınları, Ankara, 316s.
- Öztuna, Ş.**, 1987, Optimization of Land Levelling by Out Of Kilter Algorithm (OKA), Middle East Technical University, 88p, Ankara, (MS thesis).
- Pusat, A.**, 1997, Bir Toplu Konut Projesi Uygulamasında Ağ Diyagramlarıyla İşgücü, Malzeme, Gelir-Gider ve Süre Planlanması, Ege Üniversitesi Fen Bilimleri Enstitüsü, İzmir, 62s, (yüksek lisans tezi).
- Scheid, F.**, 1982, Computers and Programming, McGraw Hill Book Company, 270p.
- Thesen, A.**, 1978, Computer Methods in Operations Research, Academic Press, NewYork, 268p.
- Topçu, A.**, 1985, Sayısal ve Alfanümerik Vektörlerin Bilgisayarda Sıralama Yöntemleri, Anadolu Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi, Cilt II, Sayı 2, Eskişehir, s.29-39.
- Ulusoy, N.**, 1992, Konfeksiyon İşletmelerinde Kısıtlı Kaynak Kullanım Tekniğiyle Üretim Planlaması, Ege Üniversitesi Fen Bilimleri Enstitüsü, İzmir, 77s, (yüksek lisans tezi).

Yüksel, O., 1986, Mikrobilgisayarlar İçin Bir Sıralama Yöntemi, Anadolu Üniversitesi Mühendislik Mimarlık Fakültesi Yayınları, No:36, Eskişehir, s.27-33.

İNŞAAT MÜHENDİSLERİ ODASI
İZMİR ŞUBESİ

