

T.C.
YÜZÜNCÜ YIL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MATEMATİK ANABİLİM DALI

**GRAF VE FUZZY GRAFLARDA EN KISA YOL PROBLEMLERİNİN BAZI
ALGORİTMALAR YARDIMI İLE İNCELENMESİ**

YÜKSEK LİSANS TEZİ

HAZIRLAYAN: Mehmet DURMUŞ
DANIŞMAN: Yrd. Doç. Dr. M. Şerif ALDEMİR

VAN – 2007

T.C.
YÜZÜNCÜ YIL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MATEMATİK ANABİLİM DALI

**GRAF VE FUZZY GRAFLARDA EN KISA YOL PROBLEMLERİNİN BAZI
ALGORİTMALAR YARDIMI İLE İNCELENMESİ**

YÜKSEK LİSANS TEZİ

HAZIRLAYAN: Mehmet DURMUŞ

VAN – 2007

ÖZET

GRAF VE FUZZY GRAFLARDA EN KISA YOL PROBLEMLERİNİN BAZI ALGORİTMALAR YARDIMI İLE İNCELENMESİ

DURMUŞ, Mehmet

Yüksek Lisans Tezi, Matematik Anabilim Dalı

Tez Danışmanı: Yrd. Doç. Dr. M. Şerif ALDEMİR

Şubat 2007, 57 Sayfa

Birinci bölümde konu ile ilgili yazılmış makaleler derlenip toparlanarak kaynak bildirişi kısmı oluşturuldu.

İkinci bölümde graf ve fuzzy graflara ilgili temel tanımlara yer verildi.

Bu çalışmanın üçüncü bölümünde, Dijkstra Algoritması, Ford Algoritması, Bellmann-Kalaba Algoritması ve Latin Birleşimi Algoritması örneklerle gösterildi ve bazı fuzzy en kısa yollar için bazı fuzzy algoritmalar incelendi.

Son bölüm, dördüncü bölümde, ise en kıya yol problemlerinin çözümünde kullanılan algoritmaların kullanım özellikleri verildi.

Anahtar kelimeler: Algoritmik yaklaşım; En kısa yollar; Fuzzy graf; Fuzzy en kısa yollar; Fuzzy yollar; Graf; Yollar.

ABSTRACT

AN ANALYSIS OF THE SHORTEST DESTINATION PROBLEMS IN GRAPH AND FUZZY GRAPH BY THE AID OF SOME ALGORITHMS

DURMUŞ, Mehmet

Master Thesis, Mathematics Main Science Branch

Supervisor: Assist. Prof. Dr. M. Şerif ALDEMİR

February 2007, 57 pages

In the first section, the articles related with the subject have been compiled and merged within the context of bibliography.

In the second section, fundamental definitions have been made related with the graphs and fuzzy graphs. In the third section, Dijkstra Algorithm, Ford Algorithm, Bellmann-Kalaba Algorithm ve Latin Combined Algorithm were shown with examples and some fuzzy algorithms were investigated in order to be suited to the shortest destinations.

In the last (forth) section the utility proper of algorithms used in the solutions for the shortest destinations were presented.

Key words: Algorithmic approach; Fuzzy graf; Fuzzy paths; Fuzzy shortest paths; Graph; Paths; Shortest paths;.

ÖNSÖZ

Bu çalışma son zamanların ekonomide daha ekonomik olma mücadelesinde önem kazanan en kısa yol problemleri olarak simgeleşen ama yol boyutuna eklenen her bir değer için yeni bir kullanım alanı doğuran graf ve fuzzy graflarda en kısa yol problemleri dikkatimizi çekti. Bu beyanda en kısa yol problemlerindeki algoritmalar ekonomide (taşımacılık, ulaşım, kapasite vb.) uygulama alanı bulan problemlerde kesin çözümler bulunduğundan bu konuyu seçtik.

Graflarda en kısa yolu bulmak için kullanılan Dijkstra Algoritması, Ford Algoritması, Bellmann-Kalaba Algoritması ve Latin Birleşimi Algoritmaları incelendi. Ayrıca bazı fuzzy en kısa yol algoritmaları da özetlendi. Uygulama ve kullanım kolaylığı sunan algoritmalar tespit edilerek sonuç bölümünde tartışıldı. Ek olarak Ford algoritmasına ait bir bilgisayar programı yapıldı.

Bu tez çalışmamda, her türlü ilgi ve yardımlarını esirgemeyen ve olabilecek en iyi çalışmayı hiç aksatmadan, ihmal etmeden, günlük hayatta bana ve tezime lazım olacak her konuyu ve her bilgiyi eksiksiz ve itina ile çok büyük fedakarlıklara katlanarak insancıl güzelliğin zirvesinde, sıkmadan, bıktırmadan, korkutmadan onlarca kitabın çeviri ve anlatımlarını, gündüz-gece, mesaiye bakmadan, bana tekrarlayarak anlatan danışmanım sayın Yrd. Doç. Dr. M. Şerif ADEMİR' e teşekkürlerimi sunarım.

Ayrıca bu çalışmam süresince bilgisayar ve çeviri konularında aynı özverili yardımlarını esirgemeyen Doç. Dr. Zeynel YALÇIN 'a teşekkürü borç bilirim.

Mehmet DURMUŞ

İÇİNDEKİLER

	Sayfa
ÖZET	i
ABSTRACT	iii
ÖN SÖZ	v
İÇİNDEKİLER	vii
SİMGELER DİZİNİ	ix
ŞEKİLLER DİZİNİ	xi
TABLolar DİZİNİ	xiii-xiv
EKLER DİZİNİ	xv
1. GİRİŞ ve LİTERATÜR BİLDİRİŞLERİ	1
2. ÖN BİLGİLER	2
3. ALGORİTMALAR	26
4. SONUÇ	54
EKLER	55
KAYNAKLAR	56
ÖZ GEÇMİŞ	58

SİMGELER DİZGİNİ

Simgeler

t_i	:	T'nin elemanları ($i=1,2,\dots,n$) ve G'nin tepeleri.
a_k	:	A'nin elemanları ($k=1,2,\dots,m$) ve G'nin ayrıtları.
λ_i	:	Ford algoritmasında en kısa uzunluklar.
\mathcal{L}	:	Latin dizisi
$[M]$:	Yollar matrisi
$[D]$:	Değerler matrisi
$(t_i, t_j)=a_{ij}$:	t_i tepesini t_j tepesine bağlayan ayrıttır.
Γ	:	Tepeler kümesi üzerinde tanımlı bir dönüşüm.
$M = [a_{ij}]$:	Tepe-tepe bağlantı matrisi.
$B = [b_{ij}]$:	Tepe-ayrıt bağlantı matrisi
μ_A	:	A fuzzy kümesinin üyelik fonksiyonu
\underline{G}	:	Fuzzy alt küme grafi.
$\hat{+}$:	Cebirsel toplam.
$\overline{\underline{R}}$:	\underline{R} bağıntısının tümleyeni yani $\mu_{\overline{\underline{R}}}(x, y) = 1 - \mu_{\underline{R}}(x, y)$.
\oplus	:	Ayrık toplamı.
$\underline{\underline{R}}$:	\underline{R} 'ye en yakın bağıntı
G_α	:	Fuzzy bağıntısının - α düzey alt kümesi

ŞEKİLLER DİZİNİ

	Sayfa
Şekil 2.1. Ayrıt ve tepe	2
Şekil 2.2. Bir Bukle	3
Şekil 2.3. Yedi tepeli, on ayrıtlı ve bir bukleli bir graf	3
Şekil 2.4. n tepeli sıralı ayrıtlı	4
Şekil 2.5. Tam graf çeşitleri	5
Şekil 2.6. Yönlendirilmiş bir graf	6
Şekil 2.7. 6 tepeli 10 ayrıtlı ve 1 bukleli bir graf.	7
Şekil 2.8. Tepeleri (B, C, A, D) olan Tablo 2.15. in grafi.	22
Şekil 3.1. Dijkstra algoritması için bir graf	27
Şekil 3.2. Birinci tekrarlamamanın sonucundaki numaralandırmalar.	28
Şekil 3.3. İkinci tekrarlamamanın sonundaki numaralandırma	29
Şekil 3.4. Son numaralandırmalar ve t_i 'ler	30
Şekil 3.5. Ford Algoritması için 6 tepeli bir graf	31
Şekil 3.6. Bellmann-Kalaba Algoritması için bir graf	36
Şekil 3.7. Latin bileşimi algoritması için bir graf	42
Şekil 3.8. Şekil 3.6.'da verilen grafın Latin harfleriyle düzenlenmiş şekli	45
Şekil 3.9. Floyd Algoritması için bir fuzzy graf	51

TABLolar DİZİNİ

	Sayfa
Tablo 2.1.Şekil 2.7. deki grafin tepe-tepe bağlantı matrisi.	7
Tablo 2.2. Şekil 2.7.deki graftın tepe-ayrıt bağlantı matrisi.	8
Tablo 2.3. Verilen \underline{R}_1 ve \underline{R}_2 bağıntılarının birleşim ($\underline{R}_1 \cup \underline{R}_2$) bağıntısı.	11
Tablo 2.4. Verilen \underline{R}_1 ve \underline{R}_2 bağıntılarının kesişim ($\underline{R}_1 \cap \underline{R}_2$) bağıntısı.	11
Tablo 2.5. Verilen \underline{R}_1 ve \underline{R}_2 bağıntılarının cebirsel çarpım ($\underline{R}_1 \cdot \underline{R}_2$) bağıntısı.	12
Tablo 2.6. Verilen \underline{R}_1 ve \underline{R}_2 bağıntılarının cebirsel toplam ($\underline{R}_1 \hat{+} \underline{R}_2$) bağıntısı	12
Tablo 2.7. Verilen \underline{R}_1 bağıntısının tümleyen ($\overline{\underline{R}_1}$) bağıntısı	13
Tablo 2.8. Verilen \underline{R}_1 ve \underline{R}_2 bağıntılarının ayrık toplam ($\underline{R}_1 \oplus \underline{R}_2$) bağıntısı .	14
Tablo 2.9. Verilen \underline{R}_1 bağıntısına en yakın bağıntı (\underline{R}).	14
Tablo 2.10. Verilen \underline{R}_1 ve \underline{R}_2 bağıntılarının bileşke ($\underline{R}_2 \circ \underline{R}_1$) bağıntısı.	15
Tablo 2.11. Verilen \underline{R}_1 ve \underline{R}_2 bağıntılarının Boole toplam ($\underline{R}_2 \circ \underline{R}_1$) bağıntısı.	16
Tablo 2.12. Bir \underline{R} bağıntısı.	17
Tablo 2.13. $\alpha \cdot R_\alpha$ ifadesini R_α nin bütün elemanları α ile çarpılarak elde edilmesi	18
Tablo 2.14. $\underline{R}_2 \circ \underline{R}_1 = \underline{R}$ ve $\underline{R}_2 \circ \underline{R}_1 = \underline{R}$ bağıntıları.	19
Tablo 2.15. Bir \underline{R} bağıntısı	21
Tablo 2.16. \underline{R} bağıntısı ve bu bağıntıdan oluşturulan \underline{R}^2 , $\underline{R} \cup \underline{R}^2$, \underline{R}^3 , $\underline{R} \cup \underline{R}^2 \cup \underline{R}^3$, \underline{R}^4 ve $\hat{\underline{R}}$ bağıntıları	22
Tablo 3.1. Şekil 3.1.de verilen grafin tepe-tepe bağlantı matrisi	27
Tablo 3.2. Şekil 3.4. teki grafta t_1 'in t_i 'lere, $i=2,3,\dots,9$, olan en kısa uzaklıkları.	30
Tablo 3.3. t_1 tepesinin bir t_i tepesine uzaklık değerleri	31
Tablo 3.4. t_1 in t_2, t_3, t_4, t_5 ve t_6 tepelerine olan uzaklıkları	32
Tablo 3.5. λ_i değerlerinin hesaplanması	33
Tablo 3.6. En elverişli olma durumunun $\lambda_j - \lambda_i \leq q_{ij}$ eşitsizliği ile test edilmesi	33

Tablo 3.7. t_1 tepesinden t_2, t_3, t_4, t_5, t_6 tepelerine en kısa yolların uzunlukları olan $\lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6$ değerlerini	34
Tablo 3.8. 6 tepeli bir matrisin simgeli tepe-tepe matrisi	36
Tablo 3.9. 6 tepeli bir matrisin tepe-tepe matrisi	36
Tablo 3.10. Bellmann-Kalaba Algoritması için $k=0,1,2,3$ ve $i=0,1,2,3,4,5$ için d_i^k değerlerinin bulunuşu	38
Tablo 3.11. d_i^k değerlerinin toplandığı tablo	39
Tablo 3.12. $[M]^1, [M']^1$ ve $[D]^1$ matrisleri	43
Tablo 3.13. $[M]^2, [D]^2, [M]^3, [D]^3, [M]^4$ ve $[D]^4$ matrisleri	43
Tablo 3.14. $[M]^1, [M']^1$ ve $[D]^1$ matrisleri	45
Tablo 3.15. $[M]^2, [D]^2, [M]^3, [D]^3, [M]^4, [D]^4, [M]^5$ ve $[D]^5$ matrisleri	47

EKLER DİZİNİ

Sayfa

Ek 1. Teorem 3.2. Ford Algoritmasının Bilgisayarın BASIC Dilinde Programı

55

1. GİRİŞ ve LİTERATÜR BİLDİRİŞLERİ

En kısa yol problemlerine ilişkin temel tanım ve özellikler Seshu ve Reed (1961), Harary (1969), Christofides (1975), Kaufmann (1975), Dubois ve Prade (1980) ve Bakoğlu (1982) tarafından detaylı olarak verilmiştir.

Daha sonra Fuzzy kısa yol problemine ilişkin bir algoritma Klein (1991) tarafından verildi. Bunu takiben ise fuzzy genel ve/veya grafları için geçmişe dönük fuzzy keşfe yarayan FBHAO araştırma algoritması Shitong ve Jiantfu (1993) tarafından verildi.

Fuzzy en kısa yol problemi ve buna ilişkin en hayati ayrıtlara ilişkin bir algoritma Lin ve Chern (1993) tarafından verildi.

Grafları ikiye bölmek için bir fuzzy küme algoritması Yan ve Hsiao (1994) tarafından verildi.

Yapısal sıkıntılara ilişkin Fuzzy küme üzerindeki çalışma Chepoi ve Dumitrescu (1999) tarafından sonuçlandırıldı.

İnsan makine etkileşmelerini kullanarak kazanılmış yada edinilmiş bireysel ve çok bireyli fuzzy uzaysal bağıntıları Robinson (2000) vermiştir.

Ticarette akıllıca stoklama için yapay sınır şebekesi ve fuzzy sınır şebekesine dayalı genetik algoritmanın tamamı ve kökenindeki fonksiyonu bulmayı destekleyen sisteme karar kılma üzerinde Kuo ve ark. (2001) çalışmıştır.

Bu çalışmalarını algoritmalar yardımıyla daha detaylı olarak Devillez(2002) ve arkadaşları Chen ve Chiang (2004), Jamci ve ark. (2004), Colliot ve ark. (2004), Dick ve ark. (2004) ve son olarak ta Ma ve ark. (baskıda) sonuçlandırmıştır.

Bizim amacımız en kısa yolu bulmak için graf teorideki bazı algoritmaları kullanmak ve mümkün olduğu kadarınca bu algoritmalara ait bilgisayar programı hazırlamaktır. Bunun içinde graf teorisindeki temel kavramları bir arada vermek ve fuzzy graflara yönelik temel kavramlara yer vererek; fuzzy kısa yol problemlerine zemin hazırlamaktır. İleriki çalışmalarımızda ise Fuzzy graflara ilişkin algoritmaları detaylı inceleyebilmektir.

2. ÖN BİLGİLER

Bu bölümde konuyla ilgili Seshu ve Reed (1961), Harary (1969), Christofides (1975), Kaufmann (1975), Dubois ve Prade (1980) ve Bakoğlu (1982) tarafından verilen bazı temel tanım ve kavramlar detaya girmeksizin bir arada verilecektir.

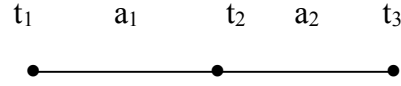
Tanım 2.1. (Graf)

T ile A ayrık iki küme ($T \neq \emptyset$) olmak üzere $G=(T,A)$ ikilisine graf denir.

Tanım 2.2. (Ayrıt ve Tepe)

T 'nin t_i , ($i=1,2,\dots,n$) elemanlarına G grafının tepeleri denir.

A 'nın a_k , ($k=1,2,\dots,m$) elemanlarına G grafının ayrıtları denir.

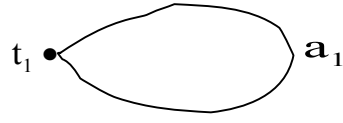


Şekil 2.1. Ayrıt ve tepe

Her ayrıtın en az bir tepesi vardır ama her tepenin bir ayrıtla bağlı olması gerekmez. Bu nedenle graf tanımında $T \neq \emptyset$ şartı konulmuştur. Şekil 2.1.deki graf ta t_1 ve t_2 tepelerine a_1 ayrıtının uç tepeleri denir. Bir ayrıtın uç tepeleri konumundaki iki tepeye bağlantılı tepeler denir. Birer tepesi ortak olan ayrıtlara bağlantılı ayrıtlar denir.

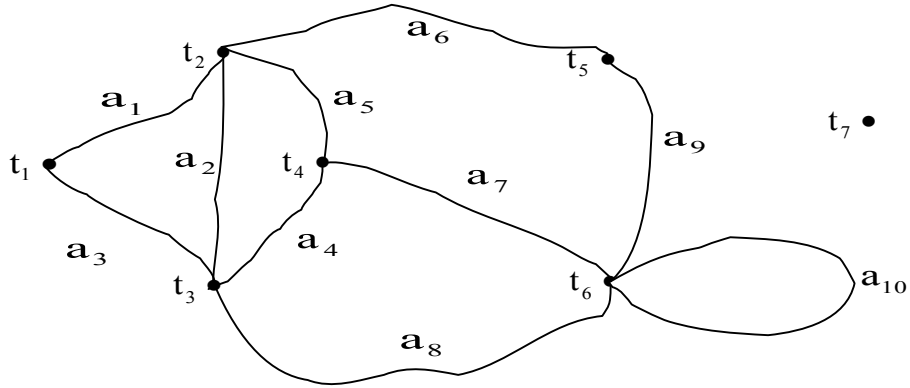
Tanım 2.3. (Bukle)

Yalnız bir tepe ile bağlantılı olan ayrıtı bukle denir. Yani; iki uç tepesi çakışık Şekil 2.2. deki gibi bir ayrıtı bukle denir.



Şekil 2.2. Bir Bukle

Yedi tepeli, on ayrıtlı ve bir bukleli bir graf aşağıda verilmiştir.



Şekil 2.3. Yedi tepeli, on ayrıtlı ve bir bukleli bir graf

Tanım 2.4. (Grafın Mertebesi ve Tepenin Derecesi)

T 'nin eleman sayısına grafın mertebesi denir. Şekil 2.3.deki grafın mertebesi 7 dir. Bir tepeye bağlı olan ayrıtların sayısına o tepenin derecesi denir. Şekil 2.3.deki grfta t_2 tepesinin derecesi 4'tür.

Tanım 2.5. (Ayrık Tepe)

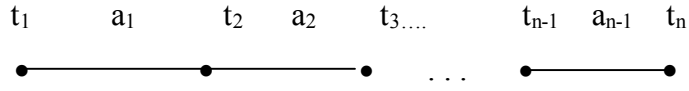
Derecesi sıfır olan tepelere ayrık (izole) tepe denir. Şekil 2.3.deki t_7 tepesi ayrık tepedir.

Tanım 2.6. (Komşuluk)

Bir G grafında bir ayrıta bağlı olan tepelere komşu tepeler denir. Bir G grafında aynı tepe ile bağlantılı olan ayrıtlara komşu ayrıtlar denir.

Tanım 2.7. (Yol)

$G=(T,A)$ grafının tepelerinin herhangi bir dizisi $\{t_0,t_1,\dots,t_{n-1},t_n\}$ olsun. Eğer her $i=1,2,3,\dots,n$ için $(t_{i-1},t_i) \in A$ ise bu diziyeye t_0 ve t_n tepelerini birleştiren bir yoldur denir. Bir grafın ayrıtları ard arda dizilerek elde edilen alt grafa sıralı ayrıt denir. Bir sıralı ayrıtta bir tepeden önce ayrıt yoksa buna başlangıç tepesi, bir tepeden sonra ayrıt yok ise buna da son tepe denir. Şekil 2.4.'de t_1 ilk tepe; t_n son tepedir.



Şekil 2.4. n tepeli sıralı ayrıt

Tanım 2.8. (Ayrıt Katarı)

Bir sıralı ayrıtta her ayrıtın tekrarı bir ise bu sıralı ayrıta ayrıt katarı denir. Bir ayrıt katarının her bir iç tepesinin derecesi 2 ve uç tepelerinin derecesi 1 ise bu ayrıt katarına yukarıda tanımlandığı gibi yol denir.

Tanım 2.9. (Çevre)

Bir ayrıt katarındaki her tepenin derecesi 2 ise bu ayrıt katarına çevre denir.

Tanım 2.10. (Elementer Yol)

$\{t_0,t_1,\dots,t_{n-1},t_n\}$ yolunda t_0 ve t_n tepelerinin dereceleri 1 ve diğer tepelerin dereceleri 2 ise bu yola elementer yol denir. Eğer $t_0=t_n$ ise bu elementer yola çevre (bukle) adı verilir.

Tanım 2.11. (Birleştirilmiş Graf)

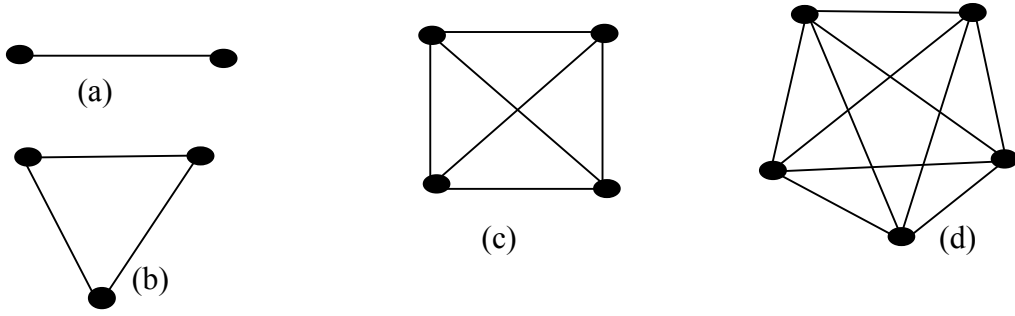
Bir grafın her farklı tepe çiftini birleştiren bir yol varsa bu grafa birleştirilmiş graf denir.

Tanım 2.12. (Lineer Graf)

Bir grafa tepeler hariç, ayrıtları hiçbir yerde kesişmeden düzleme çizilebilen graflara lineer graf denir.

Tanım 2.13. (Tam Graf)

Farklı her tepe çiftini yalnız bir ayrıt ile birleştiren grafa tam graf denir. Farklı her tepe çifti yalnız bir yönlü ayrıt ile birleştirilmiş grafa ise bir turnuva denir. Şekil 2.5.'te (a) iki tepeli, (b) üç tepeli, (c) dört tepeli ve (d) beş tepeli tam graflar oluşturulmuştur.



Şekil 2.5. Tam graf çeşitleri

Tanım 2.14. (Ağaç)

Çevre içermeyen birleştirilmiş alt grafa ağaç denir.

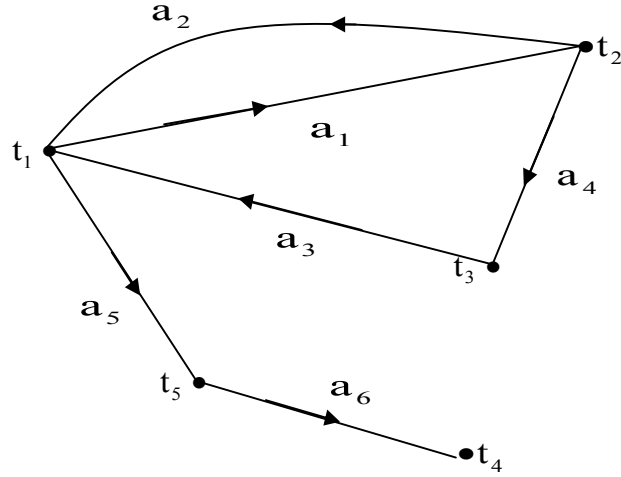
Tanım 2.15. (Yönlü Ayrıt)

Bir grafın bir ayrıtının uç noktalarının sıralanmasıyla, ayrıtta bir yön saptanmış olur. Böyle yönü saptanmış ayrıtlara yönlü ayrıtlar denir.

Tanım 2.16. (Yönlendirilmiş Graf)

Bir grafın her ayrıtı, belli bir yöne sahipse, böyle bir grafa yönlendirilmiş graf denir.

Çift yönlü ayrıtta yön belirtilmeyebilir. Yani; uç noktalarında t_1 ve t_2 olan bir ayrıt üzerinde t_1 veya t_2 ye doğru bir yön oku yoksa iki yönde yönlendirilmiş kabul edilmektedir. Yönlü bir G grafi, T tepeler kümesi ve Γ tepeler kümesi üzerinde tanımlı bir dönüşüm olmak üzere, $G=(T,\Gamma)$ ikilisi ile gösterilebilir. Şekil 2.6. da Γ 'nın kullanımı örneklenmiştir.



Şekil 2.6. Yönlendirilmiş bir graf

$\Gamma(t_1)=\{t_2,t_3,t_5\}$ yani t_1 başlangıç tepeli ayrıtların son tepeleri t_2 ve t_5 dir.

Benzer şekilde;

$$\Gamma(t_2)=\{t_1,t_3\}$$

$$\Gamma(t_3)=\{t_2\}$$

$$\Gamma(t_4)=\emptyset$$

$$\Gamma(t_5)=\{t_4\}$$
 yazılabilir.

Tanım 2.17. (Bağlantı Matrisi)

Bir grafta hangi tepelerin hangi ayrıtlarla ya da hangi tepelerin hangi tepelerle bağlantılı olduğunu göstermek grafi tümüyle görünür kılar. Böyle bir gösterim tablo veya matrisle olur. Matrisin her satırı bir tepeye ve her sütunu da bir ayrıta (veya bir tepeye) karşı getirilir. Bu karşı getirme sonunda, grafin $M = [a_{ij}]$ ile gösterilen matrisine bağlantı matrisi denir.

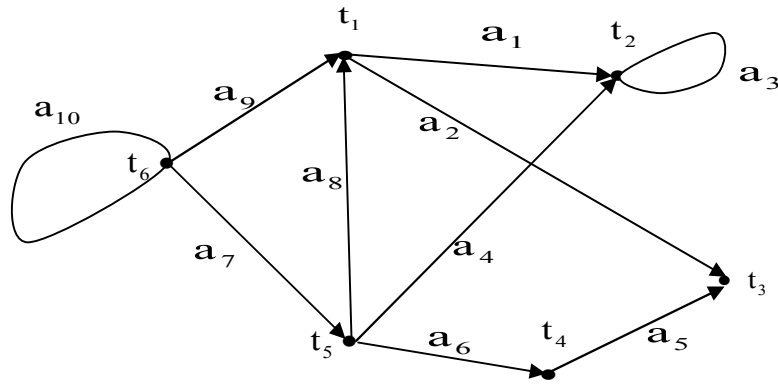
Tanım 2.18. (Tepe-Tepe Bağlantı Matrisi)

Verilin bir grafın tepe-tepe bağlantı matrisi $A = [a_{ij}]$ ise

$$A = [a_{ij}] = \begin{cases} a_{ij} = 1, & \text{eğer } (t_i, t_j) \in G \text{ ise} \\ a_{ij} = 0, & \text{eğer } (t_i, t_j) \notin G \text{ ise} \end{cases}$$

Şeklinde tanımlanır.

Bu tanımları Şekil 2.7. de verilen 6 tepeli 10 ayrıtlı ve 1 bukleli graf ile açılalım:



Şekil 2.7. 6 tepeli 10 ayrıtlı ve 1 bukleli bir graf.

$$A = \begin{matrix} & \begin{matrix} t_1 & t_2 & t_3 & t_4 & t_5 & t_6 \end{matrix} \\ \begin{matrix} t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \end{matrix} & \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{matrix}$$

Tablo 2.1.Şekil 2.7. deki grafın tepe-tepe bağlantı matrisi.

Bağlantı matrisinin ikinci kuvvetini ele aldığımızda, yani $a_{ik}^{(2)} = \sum_{j=1}^n a_{ij} a_{jk}$ ile verilen

A^2 matrisi $a_{ik}^{(2)}$ elemanı ile ifade edilir. $a_{ik}^{(2)} = \sum_{j=1}^n a_{ij} a_{jk}$ eşitliğindeki toplamada her bir

terimin değerinin 1 olması için a_{ij} ve a_{jk} nin her ikisinin aynı anda 1 olmasıyla mümkündür.

Aksi takdirde "0" değerine sahiptir. Böylece $a_{ij} = a_{jk} = 1$ olması, k_j tepesini içeren t_i den t_j ye

2 adımlı (ayrıtli) bir yolun varlığını garantiler ve $a_{ik}^{(2)}$ terimi, t_i den t_k ya 2 adımlı yolların toplamıdır.

Benzer şekilde, eğer $a_{ik}^{(p)}$, A^p nin bir elemanı ise o zaman $a_{ik}^{(p)}$, t_i den t_k ya p adımlı (ayrıtli) yolların sayısıdır.

Tanım 2.19. (Tepe-Ayrıt Bağlantı Matrisi)

Verilen n -tepeli ve m -ayrıtli bir yönlü grafinin tepe-ayrıt bağlantı matrisi

$$B = [b_{ij}] = \begin{cases} b_{ij} = 1, & \text{eğer } t_i \text{ tepesi } a_j \text{ ayrıtının bir başlangıç tepesi ise} \\ b_{ij} = -1, & \text{eğer } t_i \text{ tepesi } a_j \text{ ayrıtının bir son tepesi ise} \\ b_{ij} = 0, & \text{eğer } t_i \text{ tepesi } a_j \text{ ayrıtı ile bağlı değil ve diğer durumlar} \end{cases}$$

şeklinde tanımlanır.

$$B = [b_{ij}] = \begin{array}{c} \begin{matrix} & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 & a_9 & a_{10} \end{matrix} \\ \begin{matrix} t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \end{matrix} \left[\begin{array}{cccccccccc} 1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{array} \right] \end{array}$$

Tablo 2.2. Şekil 2.7.deki grafin tepe-ayrıt bağlantı matrisi.

Eğer yönlendirilmemiş graf olsa bağlantı matrisinde -1 yerine +1 alınır.

Tanım 2.20. (Fuzzy Küme)

X kümesi ve $[0,1]$ aralığı verilsin.

$\mu_A : X \rightarrow [0,1]$ ile tanımlanan fonksiyona Fuzzy küme denir ve

$$A = \{(x | \mu_A(x)); x \in X\}$$

ile gösterilir. $\forall x \in X$ için $\mu_A(x) \in [0,1]$ olmak üzere

$$\mu_A : X \rightarrow [0,1]$$

ile tanımlı olan fonksiyona X 'in "A fuzzy kümesi" denir.

Tanım 2.21. (Üyelik Fonksiyonu ve Üyelik Derecesi)

μ_A fonksiyonuna “A fuzzy kümesinin üyelik fonksiyonu” denir.

$\mu_A(x)$ değerine x’in “üyelik derecesi (veya değeri)” denir ve $\mu_A(x)$ kümesine de A fuzzy kümesine ait olan elemanların “üyelik derecelerinin kümesi” denir.

Tanım 2.22. (Fuzzy graf)

$x \in E_1$ ve $y \in E_2$ olmak üzere E_1 ve E_2 kümelerini göz önüne alalım. (x,y) sıralı çiftlerinin kümesi $E_1 \times E_2$ çarpım kümesinde tanımlanmış olsun. $M, E_1 \times E_2$ nin üyelik kümesi olmak üzere $\forall (x,y) \in E_1 \times E_2 : \mu_G(x,y) \in M$ olacak şekilde G fuzzy alt kümesine bir graftır denir.

Tanım 2.23. (Fuzzy Yol)

Sonlu bir $G \subset E \times E$ grafında tekrarlı veya tekrarsız r-sıralı $C = (t_{i_1}, t_{i_2}, \dots, t_{i_r})$ olsun.

Burada $t_{i_k} \in E, k = 1, 2, \dots, r$ ve $\forall (t_{i_k}, t_{i_{k+1}}) : \mu_R(t_{i_k}, t_{i_{k+1}}) \geq 0, k = 1, 2, \dots, r-1$ koşulu sağlanır.

Böyle r-sıralıya G grafında (veya β bağıntısı üzerinde) x_{i_1} den x_{i_r} ye bir “Fuzzy yoldur” denir. Burada,

1) $\forall (t_{i_1}, t_{i_2}, \dots, t_{i_r})$ yolu için

$l(t_{i_1}, t_{i_2}, \dots, t_{i_r}) = \mu_R(t_{i_1}, t_{i_2}) \wedge \mu_R(t_{i_2}, t_{i_3}) \wedge \dots \wedge \mu_R(t_{i_{r-1}}, t_{i_r}), (\wedge = \text{minimum})$ ile tanımlı bir değer karşılık gelir.

2) t_i ve t_j E’nin keyfi iki elemanı olsun. t_i ve t_j arasındaki tüm yolları ele alalım.

$C(t_i, t_j)$ böyle yolların kümesi olsun. Yani

$$C(t_i, t_j) = \left\{ c(t_i, t_j) : c(t_i, t_j) = (t_{i_1} \quad t_i, t_{i_2}, \dots, t_{i_{r-1}}, t_{i_r} \quad t_j) \right\} =$$

ile tanımlanır.

3) t_i den t_j ye kuvvetli yol $C^*(t_i, t_j)$ ile gösterilir ve

$$C^*(t_i, t_j) = \bigvee_{c(t_i, t_j)} l(t_{i1} = t_i, t_{i2}, \dots, t_{ir-1}, t_{ir} = t_j), \quad (V = \text{maximum})$$

ile tanımlanır.

Tanım 2.24.

R 'de $*$ ikili işlemi artandır denir \Leftrightarrow eğer $x_1 > y_1$ ve $x_2 > y_2$ iken $x_1 * x_2 > y_1 * y_2$ olmasıdır.

Genişletme kuralları kullanarak, M ve N fuzzy sayılarını birleştirmek için $*$ işlemi \otimes işlemine genişletilebilir. Üstelik μ_M ve μ_N , \mathbb{R} üzerinde sürekli iki fonksiyon olduğunu varsayalım. O halde

$$\mu_{M \otimes N}(Z) = \sup_{z=x*y} \min(\mu_M(x), \mu_N(y))$$

olur.

Tanım 2.25. (Fuzzy Bağıntısı)

$E_1, x \in E_2$ içinde bir fuzzy bağıntısı $x \in E_1, y \in E_2$ için $x \mathfrak{R} y$ olarak yazılır.

$$\left. \begin{array}{l} \mu_1(x) = \bigvee_y \mu(x, y) \\ \mu_1(x) = \text{MAX}_y \mu(x, y) \end{array} \right\} y' \text{e göre maksimumu temsil eder.}$$

$$\left. \begin{array}{l} \mu_2(x) = \bigwedge_y \mu(x, y) \\ \mu_2(x) = \text{MIN}_y \mu(x, y) \end{array} \right\} y' \text{e göre minimumu temsil eder.}$$

Tanım 2.26. (İki Bağıntının Birleşimi)

\mathfrak{R}_1 ve \mathfrak{R}_2 bağıntılarının bileşkesi $\mathfrak{R}_1 \cup \mathfrak{R}_2$ veya $\mathfrak{R}_1 + \mathfrak{R}_2$ ile gösterilir ve

$$\begin{aligned} \mu_{\mathfrak{R}_1 \cup \mathfrak{R}_2}(x, y) &= \mu_{\mathfrak{R}_1}(x, y) \vee \mu_{\mathfrak{R}_2}(x, y) \\ &= \text{MAX}[\mu_{\mathfrak{R}_1}(x, y), \mu_{\mathfrak{R}_2}(x, y)] \end{aligned}$$

ile tanımlanır.

Eğer $\mathfrak{R}_1, \mathfrak{R}_2, \dots, \mathfrak{R}_n$ bağıntılar ise, o zaman

$$\mu_{R_1 \cup R_2 \cup \dots \cup R_n}(x, y) = \bigvee_{R_i} \mu_{R_i}(x, y)$$

dur. Daha genel olarak

$$\underline{R} = \bigcup_i \underline{R}_i \text{ veya } \sum_i \underline{R}_i$$

dır.

\underline{R}_1	\nearrow	y_1	y_2	y_3	y_4	,	\underline{R}_2	\nearrow	y_1	y_2	y_3	y_4	,	$\underline{R}_1 \cup \underline{R}_2$	\nearrow	y_1	y_2	y_3	y_4
x_1	0,3	0,2	1	0			x_1	0,3	0	0,7	0			x_1	0,3	0,2	1	0	
x_2	0,8	1	0	0,2			x_2	0,1	0,8	1	1			x_2	0,8	1	1	1	
x_3	0,5	0	0,4	0			x_3	0,6	0,9	0,3	0,2			x_3	0,6	0,9	0,4	0,2	

Tablo 2.3. Verilen \underline{R}_1 ve \underline{R}_2 bağıntılarının birleşim ($\underline{R}_1 \cup \underline{R}_2$) bağıntısı.

Tanım 2.27. (İki Bağıntının Kesişimi)

\underline{R}_1 ve \underline{R}_2 bağıntılarının kesişimi $\underline{R}_1 \cap \underline{R}_2$ ile gösterilir ve

$$\begin{aligned} \mu_{\underline{R}_1 \cap \underline{R}_2}(x, y) &= \mu_{\underline{R}_1}(x, y) \wedge \mu_{\underline{R}_2}(x, y) \\ &= \text{MIN}[\mu_{\underline{R}_1}(x, y), \mu_{\underline{R}_2}(x, y)] \end{aligned}$$

ile tanımlanır.

Eğer $\underline{R}_1, \underline{R}_2, \dots, \underline{R}_n$ bağıntı iseler,

$$\mu_{\underline{R}_1 \cap \underline{R}_2 \cap \dots \cap \underline{R}_n}(x, y) = \bigwedge_{R_i} \mu_{R_i}(x, y)$$

ile gösterilir. Sonuç olarak

$$\underline{R} = \bigcap_i \underline{R}_i$$

olarak yazılabilir.

\underline{R}_1	\nearrow	y_1	y_2	y_3	y_4	,	\underline{R}_2	\nearrow	y_1	y_2	y_3	y_4	,	$\underline{R}_1 \cap \underline{R}_2$	\nearrow	y_1	y_2	y_3	y_4
x_1	0,3	0,2	1	0			x_1	0,3	0	0,7	0			x_1	0,3	0	0,7	0	
x_2	0,8	1	0	0,2			x_2	0,1	0,8	1	1			x_2	0,1	0,8	0	0,2	
x_3	0,5	0	0,4	0			x_3	0,6	0,9	0,3	0,2			x_3	0,5	0	0,3	0	

Tablo 2.4. Verilen \underline{R}_1 ve \underline{R}_2 bağıntılarının kesişim ($\underline{R}_1 \cap \underline{R}_2$) bağıntısı.

Tanım 2.28. (İki Bağıntının Cebirsel Çarpımı)

\underline{R}_1 ve \underline{R}_2 bağıntılarının cebirsel çarpımları $\underline{R}_1 \cdot \underline{R}_2$ ile gösterilir ve

$$\mu_{\underline{R}_1 \cdot \underline{R}_2}(x, y) = \mu_{\underline{R}_1}(x, y) \cdot \mu_{\underline{R}_2}(x, y)$$

ile tanımlanır.

\underline{R}_1	\nearrow	y_1	y_2	y_3	y_4
x_1		0,3	0,2	1	0
x_2		0,8	1	0	0,2
x_3		0,5	0	0,4	0

,

\underline{R}_2	\nearrow	y_1	y_2	y_3	y_4
x_1		0,3	0	0,7	0
x_2		0,1	0,8	1	1
x_3		0,6	0,9	0,3	0,2

,

$\underline{R}_1 \cdot \underline{R}_2$	\nearrow	y_1	y_2	y_3	y_4
x_1		0,09	0	0,7	0
x_2		0,08	0,8	0	0,2
x_3		0,3	0	0,12	0

Tablo 2.5. Verilen \underline{R}_1 ve \underline{R}_2 bağıntılarının cebirsel çarpım ($\underline{R}_1 \cdot \underline{R}_2$) bağıntısı.

Tanım 2.29. (İki Bağıntının Cebirsel Toplamı)

\underline{R}_1 ve \underline{R}_2 bağıntılarının cebirsel toplamı $\underline{R}_1 \hat{+} \underline{R}_2$ ile gösterilir ve

$$\mu_{\underline{R}_1 \hat{+} \underline{R}_2}(x, y) = \mu_{\underline{R}_1}(x, y) + \mu_{\underline{R}_2}(x, y) - \mu_{\underline{R}_1}(x, y) \cdot \mu_{\underline{R}_2}(x, y)$$

ile tanımlanır.

\underline{R}_1	\nearrow	y_1	y_2	y_3	y_4
x_1		0,3	0,2	1	0
x_2		0,8	1	0	0,2
x_3		0,5	0	0,4	0

,

\underline{R}_2	\nearrow	y_1	y_2	y_3	y_4
x_1		0,3	0	0,7	0
x_2		0,1	0,8	1	1
x_3		0,6	0,9	0,3	0,2

,

$\underline{R}_1 \cdot \underline{R}_2$	\nearrow	y_1	y_2	y_3	y_4
x_1		0,09	0	0,7	0
x_2		0,08	0,8	0	0,2
x_3		0,3	0	0,12	0

$\underline{R}_1 \hat{+} \underline{R}_2$	\nearrow	y_1	y_2	y_3	y_4
x_1		0,51	0,20	1	0
x_2		0,82	1	1	1
x_3		0,80	0,90	0,58	0,20

Tablo 2.6. Verilen \underline{R}_1 ve \underline{R}_2 bağıntılarının cebirsel toplam ($\underline{R}_1 \hat{+} \underline{R}_2$) bağıntısı

Tanım 2.30. (Bağıntıların Dağılma Özellikleri)

Fuzzy bağıntılarının dağılma özellikleri

$$\underline{R}_1 \cap (\underline{R}_2 \cup \underline{R}_3) = (\underline{R}_1 \cap \underline{R}_2) \cup (\underline{R}_1 \cap \underline{R}_3)$$

$$\underline{R}_1 \cup (\underline{R}_2 \cap \underline{R}_3) = (\underline{R}_1 \cup \underline{R}_2) \cap (\underline{R}_1 \cup \underline{R}_3)$$

$$\underline{R}_1 \cdot (\underline{R}_2 \cup \underline{R}_3) = (\underline{R}_1 \cdot \underline{R}_2) \cup (\underline{R}_1 \cdot \underline{R}_3)$$

$$\underline{R}_1 \cdot (\underline{R}_2 \cap \underline{R}_3) = (\underline{R}_1 \cdot \underline{R}_2) \cap (\underline{R}_1 \cdot \underline{R}_3)$$

$$\underline{R}_1 \hat{+} (\underline{R}_2 \cup \underline{R}_3) = (\underline{R}_1 \hat{+} \underline{R}_2) \cup (\underline{R}_1 \hat{+} \underline{R}_3)$$

$$\underline{R}_1 \hat{+} (\underline{R}_2 \cap \underline{R}_3) = (\underline{R}_1 \hat{+} \underline{R}_2) \cap (\underline{R}_1 \hat{+} \underline{R}_3)$$

şeklindedir.

Tanım 2.31. (Bir Bağıntının Tümlenyeni)

\underline{R} bağıntısının tümlenyeni $\overline{\underline{R}}$ ile gösterilir ve $\forall (x, y) \in E_1 \times E_2$ için

$$\mu_{\overline{\underline{R}}}(x, y) = 1 - \mu_{\underline{R}}(x, y)$$

ile tanımlanır.

\underline{R}	\nearrow	y_1	y_2	y_3	y_4
x_1		0,3	0,4	0,2	0
x_2		0,5	0	1	0,9
x_3		0,4	0	0,1	0,8

,

$\overline{\underline{R}}$	\nearrow	y_1	y_2	y_3	y_4
x_1		0,7	0,6	0,8	1
x_2		0,5	1	0	0,1
x_3		0,6	1	0,9	0,2

Tablo 2.7. Verilen \underline{R}_1 bağıntısının tümlenyeni ($\overline{\underline{R}}$) bağıntısı

Tanım 2.32. (İki Bağıntının Ayrık Toplamı)

\underline{R}_1 ve \underline{R}_2 bağıntılarının ayrık toplamı $\underline{R}_1 \oplus \underline{R}_2$ ile gösterilir ve

$$\underline{R}_1 \oplus \underline{R}_2 = (\underline{R}_1 \cap \overline{\underline{R}_2}) \cup (\overline{\underline{R}_1} \cap \underline{R}_2)$$

ile tanımlanır.

R_1	\nearrow	y_1	y_2	y_3	,	R_2	\nearrow	y_1	y_2	y_3	,	\bar{R}_1	\nearrow	y_1	y_2	y_3	,	\bar{R}_2	\nearrow	y_1	y_2	y_3
x_1	1	0,3	0			x_1	0,2	0,6	1			x_1	0	0,7	1			x_1	0,8	0,4	0	
x_2	0,1	1	0,8			x_2	0,8	0	0,2			x_2	0,9	0	0,2			x_2	0,2	1	0,8	

$R_1 \cap \bar{R}_2$	\nearrow	y_1	y_2	y_3	,	$\bar{R}_1 \cap R_2$	\nearrow	y_1	y_2	y_3	,	$R_1 \oplus R_2$	\nearrow	y_1	y_2	y_3
x_1	0,8	0,3	0			x_1	0	0,6	1			x_1	0,8	0,6	1	
x_2	0,1	1	0,8			x_2	0,8	0	0,2			x_2	0,8	1	0,8	

Tablo 2.8. Verilen R_1 ve R_2 bağıntılarının ayrık toplam ($R_1 \oplus R_2$) bağıntısı .**Tanım 2.33. (Bir Fuzzy Bağıntısına En Yakın Bağıntı)**

R fuzzy bağıntısı verilsin. Bu bağıntıya en yakın bağıntı \underline{R} ile gösterilir ve

$$\mu_{\underline{R}}(x,y) = \begin{cases} 0 & \mu_R(x,y) < 0,5 \\ 1 & \mu_R(x,y) > 0,5 \\ 0 \text{ veya } 1 & \mu_R(x,y) = 0,5 \end{cases}$$

ile tanımlanır .

R	\nearrow	y_1	y_2	y_3	y_4	y_5	y_6	,	\underline{R}	\nearrow	y_1	y_2	y_3	y_4	y_5	y_6
x_1	0,7	0,3	0,2	1	0	0,8			x_1	1	0	0	1	0	1	
x_2	0,5	0,4	0	0,6	0,9	0,1			x_2	0	0	0	1	1	0	
x_3	0,6	1	0,8	0	0	0,7			x_3	1	1	1	0	0	1	

Tablo 2.9. Verilen R_1 bağıntısına en yakın bağıntı (\underline{R}).**Tanım 2.34. (Max-min Bileşke İşlemi)**

$R_1 \subset X \times Y$ ve $R_2 \subset Y \times Z$ olsun. R_1 ve R_2 'nin max-min bileşkesi

$R_1 \circ R_2$ ile gösterilir. $x \in X$, $y \in Y$, $z \in Z$ olmak üzere

$$\begin{aligned} \mu_{R_1 \circ R_2}(x,y) &= \bigvee_y \left[\mu_{R_1}(x,y) \wedge \mu_{R_2}(y,z) \right] \\ &= \text{MAX}_y \left[\text{MIN} \left(\mu_{R_1}(x,y), \mu_{R_2}(y,z) \right) \right] \end{aligned}$$

ile tanımlanır.

\tilde{R}_1	\nearrow	y_1	y_2	y_3	y_4	y_5
x_1		0,1	0,2	0	1	0,7
x_2		0,3	0,5	0	0,2	1
x_3		0,8	0	1	0,4	0,3

\tilde{R}_2	\nearrow	z_1	z_2	z_3	z_4
y_1		0,9	0	0,3	0,4
y_2		0,2	1	0,8	0
y_3		0,8	0	0,7	1
y_4		0,4	0,2	0,3	0
y_5		0	1	0	0,8

$\tilde{R}_2 \circ \tilde{R}_1$	\nearrow	z_1	z_2	z_3	z_4
x_1		0,4	0,7	0,3	0,7
x_2		0,3	1	0,5	0,8
x_3		0,8	0,3	0,7	1

Tablo 2.10. Verilen \tilde{R}_1 ve \tilde{R}_2 bağıntılarının bileşke ($\tilde{R}_2 \circ \tilde{R}_1$) bağıntısı.

$$(x,z) = (x_1, z_1)$$

$$\text{MIN}(\mu_{\tilde{R}_1}(x_1, y_1), \mu_{\tilde{R}_2}(y_1, z_1)) = \text{MIN}(0,1, 0,9) = 0,1$$

$$\text{MIN}(\mu_{\tilde{R}_1}(x_1, y_2), \mu_{\tilde{R}_2}(y_2, z_1)) = \text{MIN}(0,2, 0,2) = 0,2$$

$$\text{MIN}(\mu_{\tilde{R}_1}(x_1, y_3), \mu_{\tilde{R}_2}(y_3, z_1)) = \text{MIN}(0, 0,8) = 0$$

$$\text{MIN}(\mu_{\tilde{R}_1}(x_1, y_4), \mu_{\tilde{R}_2}(y_4, z_1)) = \text{MIN}(1, 0,4) = 0,4$$

$$\text{MIN}(\mu_{\tilde{R}_1}(x_1, y_5), \mu_{\tilde{R}_2}(y_5, z_1)) = \text{MIN}(0,7, 0) = 0$$

$$\text{MAX}_{y_i} \left[\text{MIN}(\mu_{\tilde{R}_1}(x_1, y_i), \mu_{\tilde{R}_2}(y_i, z_1)) \right] = \text{MAX}(0,1, 0,2, 0, 0,4, 0) = 0,4 \text{ olur.}$$

Benzer şekilde $(x,z) = (x_i, z_j); i=1,2,3; j=1,2,3,4$ bulunarak $\tilde{R}_2 \circ \tilde{R}_1$ oluşturulur.

Tanım 2.35. (Boole Çarpımı)

Burada \cdot boole çarpımı ve $\dot{\sum}_y$ Boole toplamını göstermek üzere

$$\mu_{\tilde{R}_2 \circ \tilde{R}_1}(x,z) = \dot{\sum}_y \mu_{\tilde{R}_1}(x,y) \cdot \mu_{\tilde{R}_2}(y,z)$$

ile tanımlanır.

R_1	\nearrow	y_1	y_2	y_3	y_4	y_5
x_1		0	1	1	0	0
x_2		1	0	0	1	0
x_3		0	1	0	1	1

R_2	\nearrow	z_1	z_2	z_3	z_4
y_1		1	0	1	0
y_2		0	0	0	0
y_3		1	0	0	0
y_4		0	1	1	0
y_5		0	1	0	1

$R_2 \circ R_1$	\nearrow	z_1	z_2	z_3	z_4
x_1		1	0	0	0
x_2		1	0	1	0
x_3		0	1	1	1

Tablo 2.11. Verilen R_1 ve R_2 bağıntılarının Boole toplam ($R_2 \circ R_1$) bağıntısı.

Burada max-min bileşke işlemi bileşmeli olup,

$$(R_1 \circ R_2) \circ R_3 = R_1 \circ (R_2 \circ R_3)$$

dır.

Diğer tarafta eğer R , $E \times E$ üzerinde tanımlanmış bir bağıntı ise o zaman daha genel olarak

$$R \circ R = R^2$$

ve daha genel olarak

$$\underbrace{R \circ R \circ \dots \circ R}_{k \text{ tane}} = R^k$$

yazılabilir.

Max-min bileşke işleminin birleşim üzerinde dağılma özelliğini sağladığı halde kesişim üzerinde dağılma özelliğini sağlamamaktadır. Yani aşağıdaki özellikler yazılabilir.

$$R \circ (R_1 \cup R_2) = (R \circ R_1) \cup (R \circ R_2)$$

$$R \circ (R_1 \cap R_2) \neq (R \circ R_1) \cap (R \circ R_2)$$

Ayrıca,

$$R \circ (R_1 \cap R_2) \subset (R \circ R_1) \cap (R \circ R_2)$$

$$R_1 \subset R_2 \Rightarrow R \circ R_1 \subset R \circ R_2$$

dır.

Tanım 2.37. (Max- * Bileşkesi)

Tanım 2.34. teki \wedge (min) yerine $*$ alınarak

$$\mu_{R_2 * R_1}(x, z) = \bigvee_y \left[\mu_{R_1}(x, y) * \mu_{R_2}(y, z) \right]$$

şeklinde tanımlanır.

Tanım 2.38. (Fuzzy Bağıntısının - α Düzey Alt Kümesi)

$\alpha \in [0,1]$ olsun $\underline{R} \subset X \times Y$, Y Fuzzy bağıntısının α düzey alt kümesi

$$G_\alpha = \{(x,y) \mid \mu_{\underline{R}}(x,y) \geq \alpha\}$$

ile tanımlanır.

\underline{R}	y_1	y_2	y_3	y_4
x_1	0,3	0,8	1	0
x_2	0,5	1	0,3	0,9
x_3	1	0,2	0,6	0,7

Tablo 2.12. Bir \underline{R} bağıntısı.

O zaman

$$\begin{aligned} G_{0,3} &= \{(x,y) : \mu_{\underline{R}}(x,y) \geq 0,3\} \\ &= \{(x_1,y_1), (x_1,y_2), (x_1,y_3), (x_2,y_1), (x_2,y_2), (x_2,y_3), (x_2,y_4), \\ &\quad (x_3,y_1), (x_3,y_3), (x_3,y_4)\} \end{aligned}$$

ve

$$G_{0,8} = \{(x_1,y_2), (x_1,y_3), (x_2,y_2), (x_2,y_4), (x_3,y_1)\}$$

şeklinde yazılabilir. Burada

$$\alpha_1 \geq \alpha_2 \Rightarrow G_{\alpha_1} \subset G_{\alpha_2}$$

olduğu aşıkardır.

Teorem 2.1. (Ayrışım Teoremi)

$$\mu_{R_\alpha}(x,y) = \begin{cases} 1, & \mu_{\underline{R}}(x,y) \geq \alpha \\ 0, & \mu_{\underline{R}}(x,y) < \alpha \end{cases}$$

olmak üzere her hangi bir fuzzy \underline{R} bağıntısı

$$\underline{R} = \bigvee_{\alpha} \alpha.R_\alpha, \quad 0 < \alpha \leq 1$$

biçiminde ayrışabilir. Burada $\alpha.R_\alpha$ ifadesini R_α nin bütün elemanları α ile çarpılarak elde edilmiştir. Bunu Tablo 2.13. yardımı ile izah edelim.

$$\begin{array}{|c|c|c|c|} \hline 0,3 & 0,8 & 1 & 0 \\ \hline 0,5 & 1 & 0,3 & 0,9 \\ \hline 1 & 0,2 & 0,6 & 0,7 \\ \hline \end{array} = V \left(\begin{array}{|c|c|c|c|} \hline 1 & 1 & 1 & 0 \\ \hline 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 \\ \hline \end{array}, \begin{array}{|c|c|c|c|} \hline 1 & 1 & 1 & 0 \\ \hline 1 & 1 & 1 & 1 \\ \hline 1 & 0 & 1 & 1 \\ \hline \end{array}, \begin{array}{|c|c|c|c|} \hline 0 & 1 & 1 & 0 \\ \hline 1 & 1 & 0 & 1 \\ \hline 1 & 0 & 1 & 1 \\ \hline \end{array}, \right. \\
 \\
 \begin{array}{|c|c|c|c|} \hline 0 & 1 & 1 & 0 \\ \hline 0 & 1 & 0 & 1 \\ \hline 1 & 0 & 1 & 1 \\ \hline \end{array}, \begin{array}{|c|c|c|c|} \hline 0 & 1 & 1 & 0 \\ \hline 0 & 1 & 0 & 1 \\ \hline 1 & 0 & 0 & 1 \\ \hline \end{array}, \begin{array}{|c|c|c|c|} \hline 0 & 1 & 1 & 0 \\ \hline 0 & 1 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 \\ \hline \end{array}, \\
 \\
 \begin{array}{|c|c|c|c|} \hline 0 & 0 & 1 & 0 \\ \hline 0 & 1 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 \\ \hline \end{array}, \begin{array}{|c|c|c|c|} \hline 0 & 0 & 1 & 0 \\ \hline 0 & 1 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 \\ \hline \end{array} \left. \right)$$

Tablo 2.13. $\alpha.R_\alpha$ ifadesini R_α nin bütün elemanları α ile çarpılarak elde edilmesi

Dolayısı ile

$$\underline{R} = V(0,2.\underline{R}_{0,2}, 0,3.\underline{R}_{0,3}, 0,5.\underline{R}_{0,5}, 0,6.\underline{R}_{0,6}, 0,7.\underline{R}_{0,7}, 0,8.\underline{R}_{0,8}, 0,9.\underline{R}_{0,9}, 1.\underline{R}_1)$$

yazılabileceği aşıkardır. Ayrıca

\underline{R} , \underline{R} fuzzy bağıntısının en yakın bağıntısı ise

$$\underline{R}_2 \circ \underline{R}_1 = \underline{R} \Rightarrow \underline{R}_2 \circ \underline{R}_1 = \underline{R}$$

kolaylıkla görülür. Bu Tablo 2.14. ile gösterilmiştir.

$$\begin{array}{|c|c|c|c|c|c|} \hline \underline{R}_1 & y_1 & y_2 & y_3 & y_4 & y_5 \\ \hline x_1 & 0,1 & 0,2 & 0 & 1 & 0,7 \\ \hline x_2 & 0,3 & 0,5 & 0 & 0,2 & 1 \\ \hline x_3 & 0,8 & 0 & 1 & 0,4 & 0,3 \\ \hline \end{array} \circ \begin{array}{|c|c|c|c|} \hline \underline{R}_2 & z_1 & z_2 & z_3 & z_4 \\ \hline y_1 & 0,9 & 0 & 0,3 & 0,4 \\ \hline y_2 & 0,2 & 1 & 0,8 & 0 \\ \hline y_3 & 0,8 & 0 & 0,7 & 1 \\ \hline y_4 & 0,4 & 0,2 & 0,3 & 0 \\ \hline y_5 & 0 & 1 & 0 & 0,8 \\ \hline \end{array} = \begin{array}{|c|c|c|c|} \hline \underline{R} & z_1 & z_2 & z_3 & z_4 \\ \hline x_1 & 0,4 & 0,7 & 0,3 & 0,7 \\ \hline x_2 & 0,3 & 1 & 0,5 & 0,8 \\ \hline x_3 & 0,8 & 0,3 & 0,7 & 1 \\ \hline \end{array}$$

\underline{R}_1	y_1	y_2	y_3	y_4	y_5	\circ	\underline{R}_2	z_1	z_2	z_3	z_4	$=$	\underline{R}	z_1	z_2	z_3	z_4
x_1	0	0	0	1	1		y_1	1	0	0	0		x_1	0	1	0	1
x_2	0	0	0	0	1		y_2	0	1	1	0		x_2	0	1	0	1
x_3	1	0	1	0	0		y_3	1	0	1	1		x_3	1	0	1	1
							y_4	0	0	0	0						
							y_5	0	1	0	1						

Tablo 2.14. $\underline{R}_2 \circ \underline{R}_1 = \underline{R}$ ve $\underline{R}_2 \circ \underline{R}_1 = \underline{R}$ bağıntıları.

Sonlu bir fuzzy grafta fuzzy yolların bazı özellikleri şunlardır:

Teorem 2.2.

$\underline{R} \subset \text{ExE}$ ve $\ell_k^*(x, y)$, x'ten y'ye k uzunluklu (ayrıtılı) kuvvetli yol olmak üzere,

$$\forall (x, y) \in \text{ExE} : \mu_{\underline{R}^k}(x, y) = \ell_k^*(x, y)$$

dir.

Tanım 2.39.

ExE de tanımlı $\mu_R(x, y)$ bağıntısı verilsin.

- 1) $\forall (x, y) \in \text{ExE}$ için $\mu_R(x, y) = 1$ (yansıma)
- 2) $\forall (x, y) \in \text{ExE}$ için $\mu_R(x, y) = \mu \Rightarrow \mu_R(y, x) = \mu$ (simetri)
- 3) $\forall (x, y) \in \text{ExE}$ olsun $\forall (x, y), (y, z), (x, z) \in \text{ExE}$ için

$$\mu_R(x, z) \geq \text{MAX}_y \left[\text{MIN}(\mu_R(x, y), \mu_R(y, z)) \right] \text{ (geçişme)}$$

veya

$$\mu_R(x, z) \geq \text{V}_y \left[\mu_R(x, y) \wedge \mu_R(y, z) \right]$$

ile ifade edilir.

Tanım 2.40.

\underline{R} , ExE de tanımlı bir fuzzy bağıntı olsun. $x, y, z \in E$ için

$$\mu_{\underline{R}}(x,z) \geq \text{MAX}_y \left[\text{MIN}(\mu_{\underline{R}}(x,y), \mu_{\underline{R}}(y,z)) \right]$$

ile verilen

$$\underline{R}^2 = \underline{R} \circ \underline{R}$$

tanımlanır. Bu ifade

$$\mu_{\underline{R}^2}(x,z) = \text{V}_y \left[\mu_{\underline{R}}(x,y) \wedge \mu_{\underline{R}}(y,z) \right]$$

olarak yazılabilir.

Tanım 2.40. de verilen geçişkenlik özelliği

$$\underline{R} \circ \underline{R} \subset \underline{R}$$

biçiminde ifade edilebilir. Dolayısı ile

$$\underline{R}^2 \subset \underline{R}$$

olur. Böylece $k=1,2,3,\dots$ için

$$\underline{R}^{k+1} \subset \underline{R}^k$$

elde edilir. Netice itibari ile $k=1,2,3,\dots$ için

$$\underline{R}^k \subset \underline{R}$$

elde edilir. Böylece elde edilen.

$$\hat{\underline{R}} = \underline{R} \cup \underline{R}^2 \cup \underline{R}^3 \cup \dots$$

bağıntısı, fuzzy bağıntılarının “kapalı geçişkenliği” olarak adlandırılır.

Teorem 2.3.

Her hangi bir fuzzy bağıntısının kapalı geçişkenliği, geçişken bir bağıntıdır.

İspat: Fuzzy bağıntısının kapalı geçişkenliğinin de

$$\hat{\underline{R}}^2 = \hat{\underline{R}} \circ \hat{\underline{R}} \quad \underline{R}^2 \cup \underline{R}^3 \cup \underline{R}^4 \cup \dots \text{ yazılır. Ve dolayısıyla } \hat{\underline{R}}^2 \subset \hat{\underline{R}} \text{ yazılabilir.}$$

Bu ise geçişkenliği gösterir. Yani \underline{R} bağıntısı geçişkendir.

Dolayısıyla şu özellikleri sıralayabiliriz.

$$1) (\underline{R} \supset \underline{R}^2) \Leftrightarrow (\underline{R} = \hat{\underline{R}}) \Leftrightarrow (\underline{R} \text{ geçişlidir})$$

$$2) (\underline{R} = \underline{R}^2) \Rightarrow (\underline{R} = \hat{\underline{R}}) \Leftrightarrow (\underline{R} \text{ geçişlidir})$$

3) Teorem 2.3.den yararlanarak her hangi bir bağıntıdan bir geçişken bağıntı oluşturulur.

Teorem 2.4.

\underline{R} herhangi bir fuzzy bağıntı olsun. Eğer bazı k 'lar için

$$\underline{R}^{k+1} = \underline{R}^k \text{ ise}$$

$$\hat{\underline{R}} = \underline{R} \cup \underline{R}^2 \cup \dots \cup \underline{R}^k$$

dir. Tersisi doğru değildir.

Teorem 2.5.

$\underline{R} \subset \text{ExE}$ ve $\hat{\underline{R}}$, \underline{R} 'nin kapalı geçişken bağıntısı olsun. O zaman

$$\forall (x, y) \in \text{ExE} : \mu_{\hat{\underline{R}}}(x, y) = \ell_k^*(x, y)$$

olur.

Teorem 2.6.

$n = \text{card}(E)$ olsun. Eğer $k > n = \text{card}(E)$ olmak üzere x_i den x_j ye bir yolun uzunluğu k ise zincirin tüm elemanları tek (bir tek) değildir. Yol içinde en az bir kapalı yol vardır. Eğer bu kapalı yolu (yolları) hareket ettirirsek elde edilen yol n ye eşit veya küçüktür. Ve

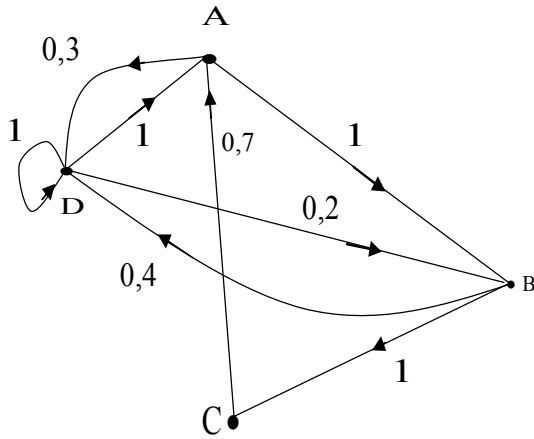
$$\ell_k^*(x, y) = \ell_{k \leq n}^*(x, y)$$

dır. Burada $\ell_{i \leq n}^*(x, y)$, x 'den y 'ye n 'den küçük veya eşit uzunluklu kuvvetli yolların değeridir.

\underline{R} bağıntısı Tablo 2.15. deki gibi verilsin. (B, C, A, D) bir yol olsun. Şimdi $\ell(B, C, A, D)$ değerini hesaplayalım.

\underline{R}	A	B	C	D
A	0	1	0	0,3
B	0	0	1	0,4
C	0,7	0	0	0
D	1	0,2	0	1

Tablo 2.15. Bir \underline{R} bağıntısı



Şekil 2.8. Tepeleri (B, C, A, D) olan Tablo 2.15. in grafi.

\tilde{R}	A	B	C	D
A	0	1	0	0,3
B	0	0	1	0,4
C	0,7	0	0	0
D	1	0,2	0	1

\tilde{R}

\tilde{R}^2	A	B	C	D
A	0,3	0,2	1	0,4
B	0,7	0,2	0	0,4
C	0	0,7	0	0,3
D	1	1	0,2	1

\tilde{R}^2

$\tilde{R} \cup \tilde{R}^2$	A	B	C	D
A	0,3	1	1	0,4
B	0,7	0,2	1	0,4
C	0,7	0,7	0	0,3
D	1	1	0,2	1

$\tilde{R} \cup \tilde{R}^2$

\tilde{R}^3	A	B	C	D
A	0,7	0,3	0,2	0,4
B	0,4	0,7	0,2	0,4
C	0,3	0,2	0,7	0,4
D	1	1	1	1

\tilde{R}^3

$\tilde{R} \cup \tilde{R}^2 \cup \tilde{R}^3$	A	B	C	D
A	0,7	1	1	0,4
B	0,7	0,7	1	0,4
C	0,7	0,7	0,7	0,4
D	1	1	1	1

$\tilde{R} \cup \tilde{R}^2 \cup \tilde{R}^3$

\tilde{R}^4	A	B	C	D
A	0,4	0,7	0,3	0,4
B	0,4	0,4	0,7	0,4
C	0,7	0,3	0,2	0,4
D	1	1	1	1

\tilde{R}^4

\hat{R}	A	B	C	D
A	0,7	1	1	0,4
B	0,7	0,7	1	0,4
C	0,7	0,7	0,7	0,4
D	1	1	1	1

$\tilde{R} \cup \tilde{R}^2 \cup \tilde{R}^3 \cup \tilde{R}^4 = \hat{R}$

Tablo 2.16. \tilde{R} bağıntısı ve bu bağıntıdan oluşturulan \tilde{R}^2 , $\tilde{R} \cup \tilde{R}^2$, \tilde{R}^3 , $\tilde{R} \cup \tilde{R}^2 \cup \tilde{R}^3$, \tilde{R}^4 ve \hat{R} bağıntıları

$$\begin{aligned}\ell(B,C,A,D) &= \mu_{\mathbb{R}}(B,C) \wedge \mu_{\mathbb{R}}(C,A) \wedge \mu_{\mathbb{R}}(A,D) \\ &= 1 \wedge 0,7 \wedge 0,3 \\ &= 0,3 \text{ olur.}\end{aligned}$$

Uzunluğu 3'eşit ve üçten küçük olan B'den D'ye tüm yollar incelenirse, bunlar (B, D), (B, D, D), (B, D, D, D) dir.

$$\ell(B,D) = \mu_{\mathbb{R}}(B,D) = 0,4$$

$$\ell(B,D,D) = \mu_{\mathbb{R}}(B,D) \wedge \mu_{\mathbb{R}}(D,D) = 0,4 \wedge 1 = 0,4$$

$$\ell(B,D,D,D) = \mu_{\mathbb{R}}(B,D) \wedge \mu_{\mathbb{R}}(D,D) \wedge \mu_{\mathbb{R}}(D,D) = 0,4 \wedge 1 \wedge 1 = 0,4$$

O halde

$$\begin{aligned}\ell^*(B,D) &= \ell(B,C,A,D) \cup \ell(B,D) \cup \ell(B,D,D) \cup \ell(B,D,D,D) \\ &= 0,3 \cup 0,4 \cup 0,4 \vee 0,4 \\ &= 0,4\end{aligned}$$

Buradan da

$$\mu_{\hat{\mathbb{R}}}(B,D) = 0,4 \text{ olur.}$$

Diğer taraftan B ve D arasında 3 uzunluklu iki yol vardır. Bunlar (B,C,A,D) ve (B,D,D,D) dir ve

$$\begin{aligned}\ell_3^*(B,D) &= \ell(B,C,A,D) \cup \ell(B,D,D,D) \\ &= 0,3 \cup 0,4 \\ &= 0,4\end{aligned}$$

elde edilir. Dolayısıyla

$$\mu_{\mathbb{R}^3}(B,D) = 0,4 \text{ dir. Şimdi}$$

(C,A,B,D,A,D) yolunu ele alalım. Bu yol (D,A,D) çevresini (kapalı yolu) içerir.

$$\begin{aligned}\ell_5^*(C,D) &= \ell_{j \leq 4}^*(C,D) \\ &= \ell_1^*(C,D) \cup \ell_2^*(C,D) \cup \ell_3^*(C,D) \cup \ell_4^*(C,D) \\ &= \mu_{\mathbb{R}}(C,D) \cup \mu_{\mathbb{R}}(C,D) \cup \mu_{\mathbb{R}}(C,D) \cup \mu_{\mathbb{R}}(C,D) \\ &= 0 \cup 0,3 \cup 0,4 \cup 0,4 \\ &= 0,4\end{aligned}$$

0,3 de bulabilirdik, C ve D arasında 5 uzunluklu kuvvetli yol (C,A,B,D,A) değil fakat (C,A,B,D,D,D) dir. Üstelik eğer kapalı yollar çıkarılırsa bu ikisi (C,A,B,D) ye indirgenir.

Tanım 2.41.

$$\ell(x_{i_1}, \dots, x_{i_r}) = \mu_{\mathbb{R}}(x_{i_1}, x_{i_2}) \wedge \dots \wedge \mu_{\mathbb{R}}(x_{i_{r-1}}, x_{i_r})$$

yardımıyla tanımlanan diğer birleşme ve monoton özelliğine sahip " \wedge " başka işlemlere genişletilebilir. Eğer $*$ böyle operatör ise

$$\ell(x_{i_1}, x_{i_2}, \dots, x_{i_r}) = \mu_{\mathbb{R}}(x_{i_1}, x_{i_2}) * \mu_{\mathbb{R}}(x_{i_2}, x_{i_3}) * \dots * \mu_{\mathbb{R}}(x_{i_{r-1}}, x_{i_r})$$

ile tanımlanır. Eğer $*$ operatörü " \bullet " çarpma işlemi olarak alınırsa

$$\ell(x_{i_1}, x_{i_2}, \dots, x_{i_r}) = \mu_{\mathbb{R}}(x_{i_1}, x_{i_2}) \bullet \mu_{\mathbb{R}}(x_{i_2}, x_{i_3}) \bullet \dots \bullet \mu_{\mathbb{R}}(x_{i_{r-1}}, x_{i_r})$$

olduğu görülür. Eğer $a, b \in [0, 1]$ ise $a \bullet b \leq a \wedge b$ dir.

Uyarı 2.2.

1) " \wedge " operatörünün geçişme özelliği " \bullet " nın geçişme özelliğini verir. Yani

$$\mathbb{R} \circ \mathbb{R} \subset \mathbb{R} \Rightarrow \mathbb{R} \bullet \mathbb{R} \subset \mathbb{R}$$

dir.

2) Bu geçişmenin tersi doğru değildir.

Tanım 2.42.

Genellikle L veya R ile gösterilen bir fonksiyon fuzzy sayılarının bir fonksiyonu olması için \Leftrightarrow

$$1) L(x) = L(-x)$$

$$2) L(0) = 1$$

$$3) L, [0, +\infty) \text{ üzerinde artan olmasıdır.}$$

Örneğin,

$$L(x) = \begin{cases} 1, & x \in [-1, +1] \\ 0, & \text{diğer durumlar} \end{cases}$$

$$L(x) = \max(0, 1 - |x|^p), p > 0,$$

$$L(x) = e^{-|x|^p}, p > 0 \text{ ve}$$

$$L(x) = \frac{1}{1+|x|^p}, p > 0$$

Fuzzy sayı fonksiyonları verilebilir. M fuzzy sayısının bir L-R tipi fuzzy sayısı olabilmesi için

$$\Leftrightarrow \mu_M(x) = \begin{cases} L\left(\frac{m-x}{\alpha}\right), x \leq m \text{ ve } \alpha > 0 \text{ için} \\ R\left(\frac{x-m}{\beta}\right), x > m \text{ ve } \beta > 0 \text{ için} \end{cases}$$

olmasıdır.

Burada L sol için ve R sağ için kullanılır. m, M nin ortalama değeridir. α ve β sırası ile sol ve sağ dağılımı gösterir. Dağılım 0 olduğu zaman M geleneksel olarak fuzzy olmayan bir sayı olarak kabul edilir. Dağılım arttıkça M git gide fuzzyleşir. Sembolik olarak

$$M = (M, \alpha, \beta)_{LR}$$

olarak yazılır.

3. ALGORİTMALAR

Bu bölümde graflarda en kısa yolu bulmak için kullanılan Dijkstra Algoritması, Ford Algoritması, Bellman-Kalaba Algoritması ve Latin Birleşimi Algoritması verilecektir. Ayrıca bazı fuzzy en kısa yol algoritmaları da özetlenecektir.

Teorem 3.1. (Dijkstra Algoritması)

$l(t_i)$, t_i tepesi üzerinde numaralanmış olsun.

Başlangıç:

Adım1: $l(s)=0$ alınır ve daimi olarak işaretlenmiş numara

$\forall t_i \neq s$ için $l(t_i) = \infty$ alınır ve bu numaralama geçici işaretlenir. $p=s$ alınır

Numaralandırmayı güncelleştirmek (uygun hale getirmek)

Adım2: $\forall t_i \in \Gamma(p)$ için geçici numaralanır

$$l(t_i) = \min[l(t_i), l(p) + l(p, t_i)]$$

ye göre numaralanmalar güncelleştirilir.

Bir numarayı daimi bir şekilde sabitleştirmek.

Adım3: $l(t_i^*) = \min[l(t_i)]$ olacak şekildeki, geçici numaralanmış olan tepelerinin her birisinin t_i^* 'i bulunur.

Adım4: Daimi t_i^* numarası işaretlenir ve $p=t_i^*$ olur.

Adım5:

(i) (Eğer sadece s den t ye bir yol işaretleniyor ise)

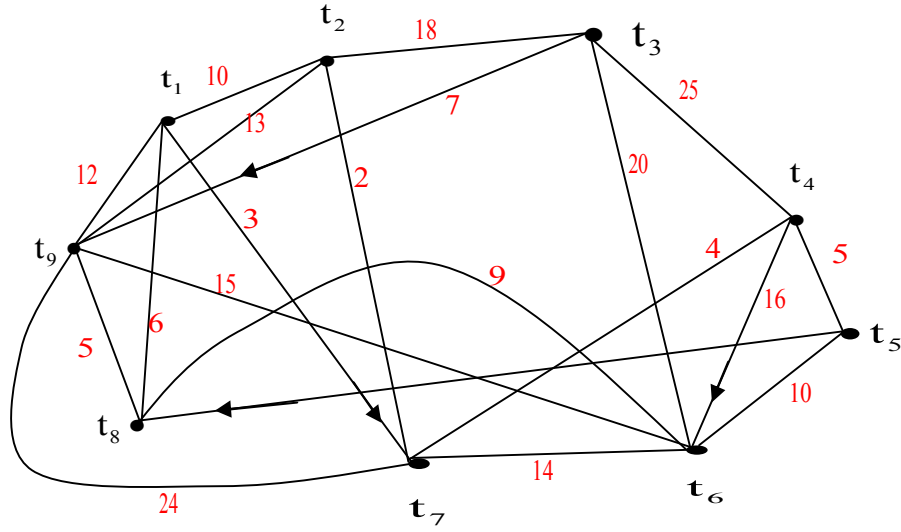
Eğer $p=t$ ise $l(p)$ istenen en kısa yolun uzunluğudur.

Eğer $p \neq t$ ise adım 2 ye gidilir.

(ii) (Eğer s den diğer tüm tepelere olan yol talep edilirse)

Eğer bütün tepeler daimi numaralanmış ise numaralamalar, en kısa yolun uzunluğudur.

Şekil 3.1.de verilen grafa en kısa yolun uzunluğunu Dijkstra algoritması yardımıyla bulalım. Bu grafin tepe-tepe bağlantı matrisi Tablo 3.1. de görülmektedir.



Şekil 3.1. Dijkstra algoritması için bir graf

	t ₁	t ₂	t ₃	t ₄	t ₅	t ₆	t ₇	t ₈	t ₉
t ₁		10					3	6	12
t ₂	10		18				2		13
t ₃		18		25		20			7
t ₄			25		5	16	4		
t ₅				5		10			
t ₆			20		10		14	15	9
t ₇		2		4		14			24
t ₈	6				23	15			5
t ₉	12	13				9	24	5	

Tablo 3.1. Şekil 3.1.de verilen grafın tepe-tepe bağlantı matrisi

Birinci tekrarlama**Adım 1:**

$$l(t_1) = 0 \stackrel{+}{=} l(t_1) \quad \infty, \quad \forall t_i \neq t_1, \quad p=t_1$$

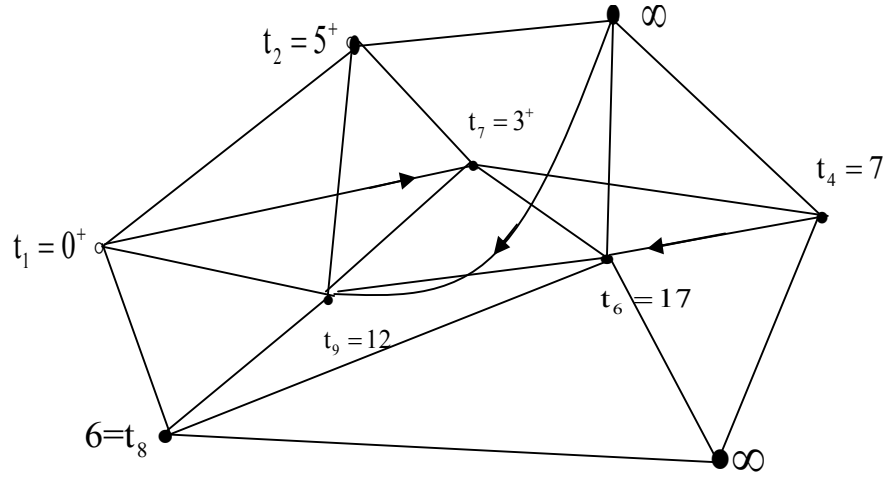
Adım 2:

$\Gamma(p) = \Gamma(t_1) = \{t_2, t_7, t_8, t_9\}$ tümü ile geçici numaralanır. İlkini t_2 'yi göz önüne alalım

$$l(t_2) = \min[\infty, 0^+ + 10] = 10$$

benzer şekilde $l(t_7) = 3, l(t_8) = 6, l(t_9) = 12$.

Adım 3:



Şekil 3.3. İkinci tekrarlamamın sonundaki numaralandırma.

Adım 3:

$$\min \left[\begin{array}{c} \underline{5}, \underline{7}, \underline{17}, \underline{6}, \underline{12}, \underline{10} \\ t_2 \quad t_4 \quad t_6 \quad t_8 \quad t_9 \quad (t_3, t_5) \end{array} \right] = 5, \text{ bu ise } t_2 \text{ karşılık gelir.}$$

Adım 4: t_2 , şimdi daimi numaralandı. $l(t_2) = 5^+$, $p = t_2$

Adım 5: Adım 2'ye gidilir.

Üçüncü tekrarlama:

Adım 2: $\Gamma(p) = \Gamma(t_2) = \{t_1, t_3, t_7, t_9\}$ sadece $\{t_3, t_9\}$ geçici numaralanır.

$$l(t_i) = \min [l(t_i), l(p) + l(p, t_i)]$$

eşitliğinden

$$l(t_3) = \min [\infty, 5^+ + 18] = 23$$

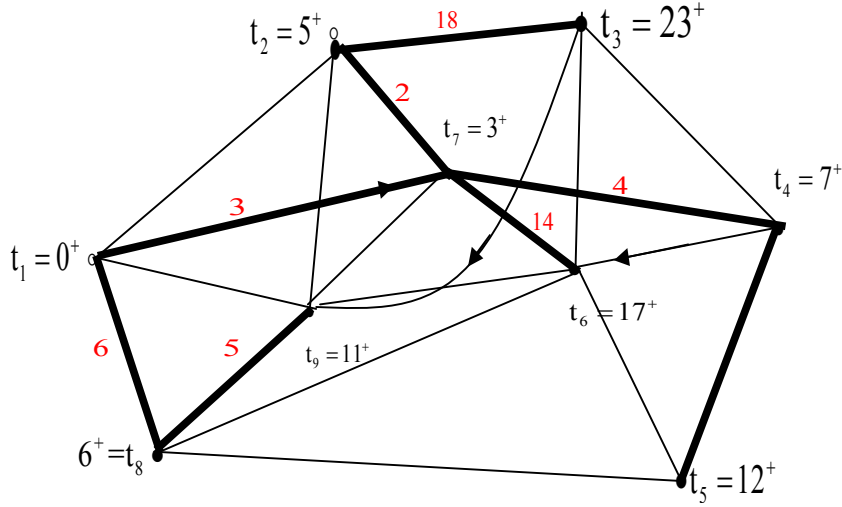
bulunur ve benzer şekilde $l(t_9) = 12$ bulunur.

$$\text{Adım 3: } \min \left[\begin{array}{c} \underline{23}, \underline{7}, \underline{17}, \underline{6}, \underline{12}, \underline{\infty} \\ t_3 \quad t_4 \quad t_6 \quad t_8 \quad t_9 \quad t_5 \end{array} \right] = 6, \text{ bu ise } t_8 \text{ 'e karşılık gelir.}$$

Adım 4: t_8 , şimdi daimi numaralandı. $l(t_8) = 6^+$, $p = t_8$

Adım 5: Adım 2'ye gidilir.

Bu şekilde devam edilerek Şekil 3.4.deki son numaralandırmayı elde ederiz.

Şekil 3.4. Son numaralandırmalar ve t_i 'ler

t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9
	5	23	7	12	17	3	6	11

Tablo 3.2. Şekil 3.4. teki grafiğe t_1 'in t_i 'lere, $i=2,3,\dots,9$, olan en kısa uzaklıkları.**Uyarı 3.1.**

- 1) Dikkat edilirse yapılacak en kısa yol bulma işleminde grafiğin bukle içermemesi durumunda Dijkstra Algoritması kullanılabilir.
- 2) Eğer t_1 tepesinin dışında herhangi bir tepenin diğer tepelere olan en kısa yollarını bulmak istersek, bu kez tablodaki t_1 tepesi yerine istenen tepe yazılarak aynı işlemler tekrarlanarak bulunur.
- 3) Sonuç olarak diyebiliriz ki
 - 1) Dijkstra Algoritması sadece bir tepeden diğer tüm tepelere en kısa yolları buluyor.
 - 2) Sadece bir tepeden diğer tepelere en kısa yol birden fazla ise bunlardan sadece bir tanesini buluyor.

Teorem 3.2. (Ford Algoritması)

Biz burada en kısa yol problemi çözümünde önemli olan Ford Algoritmasını inceleyeceğiz. Bir grafiğin herhangi iki tepesi arasındaki Minimum uzunlukta yolun bulunması problemi öncelikle bir optimizasyon (en elverişli yol) problemi olduğu açıktır. En kısa yol problemlerine ilişkin temel tanım ve özellikler incelenerek graflarda en kısa yol probleminin

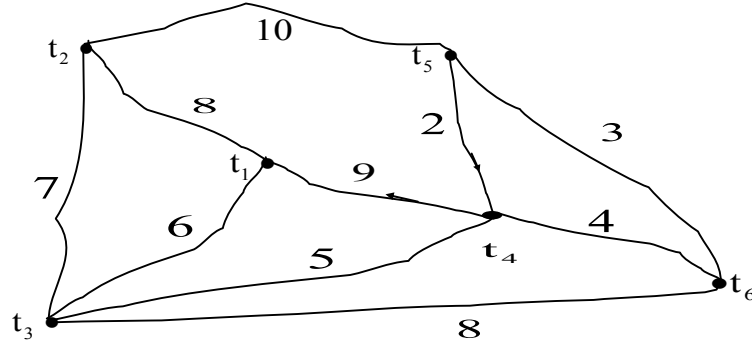
bir çözümlü olarak Ford algoritması incelenecektir.

n tepeli bir grafta t_1 tepesinden diğer tüm t_n tepelerine olan en kısa uzunluklu yolların bulunmasını deneme yanılma yerine, her bir t_i tepesinin bu tepe ile bağlantılı tepelere olan uzunlukları q_{ij} değerleri alınarak Tablo 3.3. elde edilir.

$t_i \setminus t_j$		t_1	t_j	t_n
	$\lambda_i \setminus \lambda_j$	λ_1	λ_j	λ_n
t_1	λ_1	q_{11}	q_{1j}	q_{1n}
.....
t_i	λ_i	q_{i1}	q_{ij}	q_{in}
.....
t_n	λ_n	q_{nj}	q_{nj}	q_{nn}

Tablo 3.3. t_i tepesinin bir t_i tepesine uzaklık değerleri

Daha sonra bu tablodan $\lambda_j = \min(\lambda_i + q_{ij})$ hesaplamasıyla elde edilen λ_j değerleri bulunarak ikinci bir tablo oluşturulur. İkinci bir tabloda en elverişli olma durumu olan $\lambda_j - \lambda_i \leq q_{ij}$ eşitsizliğini sağlamayan λ_j değerleri için yeni $\lambda'_{j_0} = \lambda_{i_0} + q_{i_0 j_0}$ değerleri en kısa yollar olacaktır. Bu şekilde tanımlanan Ford Algoritmasını Şekil 3.5.de verilen bir grafla izah edelim.



Şekil 3.5. Ford Algoritması için 6 tepeli bir graf

Şekil 3.5.de verilen grafin t_1 tepesinden t_2, t_3, t_4, t_5, t_6 tepelerine olan en kısa yolların uzunluğunu Ford Algoritması yardımı ile bulalım.

Uyarı 3.2.

Örneğimizdeki grafta t_5 ile t_4 ve t_4 ile t_1 arasındaki yollar tek yönlü olduğundan tablodaki yerleri boş bırakılacak.

$t_i \setminus t_j$		t_1	t_2	t_3	t_4	t_5	t_6
	$\lambda_i \setminus \lambda_j$	0	8	6	11	18	14
t_1	0		8	6			
t_2	8	8		7		10	
t_3	6	6	7		5		8
t_4	11	9		5			4
t_5	18		10		2		3
t_6	14			8	4	3	

Tablo 3.4. t_1 in t_2, t_3, t_4, t_5 ve t_6 tepelerine olan uzaklıkları

Şekil 3.5.deki graftan (t_i, t_j) yollarının uzunlukları Tablo 3.4.de yerlerine yazılarak tablo doldurulur. Sonra en kısa uzunluklar olan λ_i değerleri Tablo 3.5.deki gibi hesaplanarak Tablo 3.4.deki yerlerine yazılır.

$\lambda_j = \min(\lambda_i + q_{ij})$ $\lambda_1 = 0$ dir $\lambda_2 = \min(\lambda_1 + q_{12})$ $\lambda_2 = 0 + 8 = 8$	$\lambda_j = \min(\lambda_i + q_{ij})$ $\lambda_3 = \min(\lambda_1 + q_{13})$ $\lambda_3 = \min(0 + 6, 8 + 7)$ $\lambda_3 = 6$	$\lambda_j = \min(\lambda_i + q_{ij})$ $\lambda_4 = \min(\lambda_1 + q_{14}, \lambda_2 + q_{24}, \lambda_3 + q_{34})$ $\lambda_4 = \min(0 + q_{14}, 8 + q_{24}, 6 + 5)$ $\lambda_4 = 11$	$\lambda_j = \min(\lambda_i + q_{ij})$ $\lambda_5 = \min(\lambda_1 + q_{15}, \lambda_2 + q_{25}, \lambda_3 + q_{35}, \lambda_4 + q_{45})$ $\lambda_5 = \min(0 + q_{15}, 8 + 10, 6 + q_{35}, 11 + q_{45})$ $\lambda_5 = 18$	$\lambda_j = \min(\lambda_i + q_{ij})$ $\lambda_6 = \min(\lambda_1 + q_{16}, \lambda_2 + q_{26}, \lambda_3 + q_{36}, \lambda_4 + q_{46}, \lambda_5 + q_{56})$ $\lambda_6 = \min(0 + q_{16}, 8 + q_{26}, 6 + 8, 11 + 4, 18 + 3)$ $\lambda_6 = 14$
--	---	---	---	---

Tablo 3.5. λ_i değerlerinin hesaplanması

Son olarak ta en elverişli olma durumu olan $\lambda_j - \lambda_i \leq q_{ij}$ eşitsizliğine göre Tablo 3.7. yeniden düzenlenecek.

$\lambda_2 - \lambda_1 \leq q_{12}$ 8-0 ≤ 8	$\lambda_3 - \lambda_1 \leq q_{13}$ 6-0 ≤ 6	$\lambda_1 - \lambda_2 \leq q_{21}$ 0-8 ≤ 8	$\lambda_3 - \lambda_2 \leq q_{23}$ 6-8 ≤ 7	$\lambda_5 - \lambda_2 \leq q_{25}$ 18-8 ≤ 10	$\lambda_1 - \lambda_3 \leq q_{31}$ 0-6 ≤ 6	$\lambda_2 - \lambda_3 \leq q_{32}$ 8-6 ≤ 7	$\lambda_4 - \lambda_3 \leq q_{34}$ 11-6 ≤ 5
$\lambda_3 - \lambda_1 \leq q_{36}$ 14-6 ≤ 8	$\lambda_1 - \lambda_4 \leq q_{41}$ 0-11 ≤ 9	$\lambda_3 - \lambda_4 \leq q_{43}$ 6-11 ≤ 5	$\lambda_6 - \lambda_4 \leq q_{46}$ 14-11 ≤ 4	$\lambda_2 - \lambda_5 \leq q_{52}$ 8-18 ≤ 10	$\lambda_4 - \lambda_5 \leq q_{54}$ 11-18 ≤ 2	$\lambda_6 - \lambda_5 \leq q_{56}$ 14-18 ≤ 3	$\lambda_3 - \lambda_6 \leq q_{63}$ 6-14 ≤ 8
$\lambda_4 - \lambda_6 \leq q_{64}$ 11-14 ≤ 4	eşitsizlikleri en elverişli olma şartını sağlarken (t_6, t_5) dikdörtgeni için						$\lambda_5 - \lambda_6 \leq q_{65}$ 18-14 > 3

sağlanmamaktadır.

Tablo 3.6. En elverişli olma durumunun $\lambda_j - \lambda_i \leq q_{ij}$ eşitsizliği ile test edilmesi

Bu nedenle $\lambda'_{j_0} = \lambda_{i_0} + q_{i_0 j_0}$

formülü ile hesaplanan

$$\lambda'_5 = \lambda_6 + q_{65} \Rightarrow \lambda'_5 = 14 + 3 \Rightarrow \lambda'_5 = 17$$

değeri tabloda $\lambda_5 = 18$ yerine yazılmalıdır. Bu denetlemeden sonra tablonun en elverişli olma koşulunu sağlayan, bir düzeltilmiş tabloya erişilir. Bu son durumu Tablo 3.7.te görmekteyiz.

$t_i \setminus t_j$		t_1	t_2	t_3	t_4	t_5	t_6
	$\lambda_i \setminus \lambda_j$	0	8	6	11	17	14
t_1	0		8	6			
t_2	8	8		7		10	
t_3	6	6	7		5		8
t_4	11	9		5			4
t_5	17		10		2		3
t_6	14			8	4	3	

Tablo 3.7. t_1 tepesinden t_2, t_3, t_4, t_5, t_6 tepelerine en kısa yolların uzunlukları olan $\lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6$ değerlerini

Uyarı 3.3.

Eğer t_1 tepesinin dışında herhangi bir tepenin diğer tepelere olan en kısa yolları bulmak istersek, bu kez tablodaki t_1 tepesi yerine istenen tepe yazılarak aynı işlemler tekrarlanarak bulunur.

Ford algoritmasına ait bir bilgisayar programı BASIC programlama dili kullanılarak yazılmıştır. Bu program ek bölümünde verilmiştir.

Teorem 3.3. (Bellmann-Kalaba Algoritması)

Buklesiz bir grafın en kısa yolunun bu algoritma ile bulunması, en elverişli bir yolun sadece en elverişli alt yollardan oluştuğu fikrine dayanır. Yani seçilen alt yol; başlangıç noktasından son noktaya dek en elverişlidir. Bu esastan hareket ederek, r tane ayrıtlı en kısa uzunluklu yol $k \leq r$ koşuluna uyan en kısa alt yollardan oluşmuştur denilebilir.

$\{t_0, t_1, \dots, t_{n-1}, t_n\}$ tepeleri ile verilen bir grafın her hangi iki t_i ve t_j tepeleri arasında bir ayrıtlı varsa bu ayrıtlı $a_{ij} \geq 0$ değeri, bir ayrıtlı yoksa $a_{ij} = \infty$ ve (t_i, t_i) için de $a_{ii} = 0$ olmak

üzere a_{ij} değerleri ($i, j=0, 1, 2, \dots, n$) alınır.

Buna göre bir $\mu(t_0, t_{i_1}, t_{i_2}, \dots, t_{i_k}, t_n)$ yoluna ait $a_{0i_1} + a_{i_1 i_2} + a_{i_2 i_3} + \dots + a_{i_k n}$ değerleri toplamının minimum olması istenir.

Bu amaçla i tepesinden n son tepesine giden en elverişli yolun d_i değeri

$$d_i = \min_{j \neq i} (d_j + a_{ij}) \quad (i=0, 1, 2, \dots, n-1)$$

$$d_n = 0$$

Denklemler sisteminin çözümünden bulunur. Burada verilen $n+1$ tepeleri 0 dan n ye kadar numaralanır. Böylece,

$$d_i^{(0)} = a_{in} \quad (i=0, 1, 2, \dots, n-1)$$

$$d_n^{(0)} = 0$$

olsunlar. Buradan

$$d_i^{(1)} = \min_{j \neq i} (d_j^{(0)} + a_{ij}) \quad (i=0, 1, 2, \dots, n-1), (j=0, 1, 2, \dots, n)$$

$$d_n^{(1)} = 0$$

hesaplanır. Bu işlemlere devamla, her ($k=1, 2, 3, \dots$) değeri için,

$$d_i^{(k)} = \min_{j \neq i} (d_j^{(k-1)} + a_{ij}) \quad (i=0, 1, 2, \dots, n-1), (j=0, 1, 2, \dots, n)$$

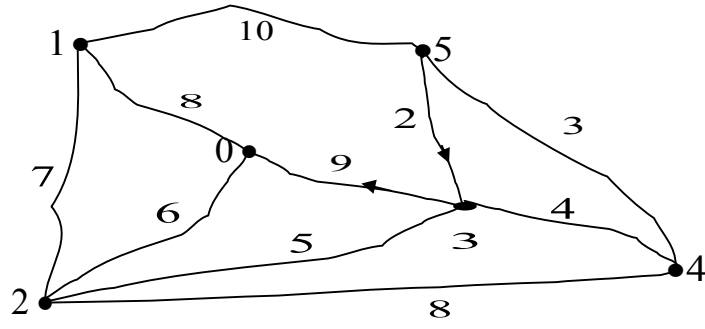
$$d_n^{(k)} = 0$$

biçiminde bir indirgeme formülü bulunur ve

$$d_i^{(k)} = d_i^{(k-1)} \quad (i=0, 1, 2, \dots, n)$$

olunca ardışık işleme son verilir. Bu halde $d_0^{(k)}$ nın değeri, t_0 ile t_n tepeleri arasındaki en elverişli yolun değerini ifade eder. Bu değer geriye doğru gidilerek hesaplanır.

$n+1$ tepeli bir grafta, en elverişli bir duruma ulaşmak için $n-1$ ardışık işlemin yeterli olduğu görülmektedir. Bunu Şekil 3.6. da verilen graf için uygulayalım.



Şekil 3.6. Bellman-Kalaba Algoritması için bir graf

Şekil 3.6. de verilen grafın tepeleri 0,1,2,3,4,5 ile numaralandırarak bu grafa ait tepe-tepe bağlantı matrisini yazalım.

Uyarı 2.4.

Şekil 3.6. daki grafı t_5 ile t_4 ve t_4 ile t_1 arasındaki yollar tek yönlü olduğundan tablodaki yerleri boş bırakılır.

$$A = \begin{matrix} & \begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} & a_{04} & a_{05} \\ a_{10} & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{20} & a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{30} & a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{40} & a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{50} & a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{bmatrix} \end{matrix}$$

Tablo 3.8. 6 tepeli bir matrisin simgeli tepe-tepe matrisi

$$A = \begin{matrix} & \begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{bmatrix} 0 & 8 & 6 & \infty & \infty & \infty \\ 8 & 0 & 7 & \infty & \infty & 10 \\ 6 & 7 & 0 & 5 & 8 & \infty \\ 9 & \infty & 5 & 0 & 4 & \infty \\ \infty & \infty & 8 & 4 & 0 & 3 \\ \infty & 10 & \infty & 2 & 3 & 0 \end{bmatrix} \end{matrix}$$

Tablo 3.9. 6 tepeli bir matrisin tepe-tepe matrisi

Bundan sonra, $i=0,1,2,3,4,5$ ile ardışık işlemlere başlanır.

1⁰)

$$d_0^{(0)} = \infty$$

$$d_1^{(0)} = a_{15} = 10$$

$$d_2^{(0)} = \infty$$

$$d_3^{(0)} = \infty$$

$$d_4^{(0)} = a_{45} = 3$$

$$d_5^{(0)} = a_{55} = 0$$

$$d_i^{(k)} = \min_{j \neq i} (d_j^{(k-1)} + a_{ij}) \quad (i=0,1,2,\dots,n-1), (j=0,1,2,\dots,n)$$

$$d_n^{(k)} = 0$$

2⁰)

$$d_0^{(1)} = \min \begin{pmatrix} d_1^{(0)} + a_{01} = 10 + 8 \\ d_2^{(0)} + a_{02} = \infty + 6 \\ d_3^{(0)} + a_{03} = \infty + \infty \\ d_4^{(0)} + a_{04} = 3 + \infty \\ d_5^{(0)} + a_{05} = 0 + \infty \end{pmatrix} = 18$$

$$d_1^{(1)} = \min \begin{pmatrix} d_0^{(0)} + a_{10} = \infty + 8 \\ d_2^{(0)} + a_{12} = \infty + 0 \\ d_3^{(0)} + a_{13} = \infty + \infty \\ d_4^{(0)} + a_{14} = 3 + \infty \\ d_5^{(0)} + a_{15} = 0 + 10 \end{pmatrix} = 10$$

$$d_2^{(1)} = \min \begin{pmatrix} d_0^{(0)} + a_{20} = \infty + 6 \\ d_1^{(0)} + a_{21} = 10 + 7 \\ d_3^{(0)} + a_{23} = \infty + 5 \\ d_4^{(0)} + a_{24} = 3 + 8 \\ d_5^{(0)} + a_{25} = 0 + \infty \end{pmatrix} = 11$$

$$d_3^{(1)} = \min \begin{pmatrix} d_0^{(0)} + a_{30} = \infty + 9 \\ d_1^{(0)} + a_{31} = 10 + \infty \\ d_2^{(0)} + a_{32} = \infty + 5 \\ d_4^{(0)} + a_{34} = 3 + 4 \\ d_5^{(0)} + a_{35} = 0 + \infty \end{pmatrix} = 7$$

$$d_4^{(1)} = \min \begin{pmatrix} d_0^{(0)} + a_{40} = \infty + \infty \\ d_1^{(0)} + a_{41} = 10 + \infty \\ d_2^{(0)} + a_{42} = \infty + 8 \\ d_3^{(0)} + a_{43} = \infty + 4 \\ d_5^{(0)} + a_{45} = 0 + 3 \end{pmatrix} = 3$$

$$d_5^{(1)} = \min \begin{pmatrix} d_0^{(0)} + a_{50} = \infty + \infty \\ d_1^{(0)} + a_{51} = 10 + 10 \\ d_2^{(0)} + a_{52} = \infty + \infty \\ d_3^{(0)} + a_{53} = \infty + 2 \\ d_4^{(0)} + a_{54} = 3 + 3 \end{pmatrix} = 0$$

3⁰)

$$d_0^{(2)} = \min \begin{pmatrix} d_1^{(1)} + a_{01} = 10 + 8 \\ d_2^{(1)} + a_{02} = 11 + 6 \\ d_3^{(1)} + a_{03} = 7 + \infty \\ d_4^{(1)} + a_{04} = 3 + \infty \\ d_5^{(1)} + a_{05} = 0 + \infty \end{pmatrix} = 17$$

$$d_1^{(2)} = \min \begin{pmatrix} d_0^{(1)} + a_{10} = \infty + 8 \\ d_2^{(1)} + a_{12} = 11 + 0 \\ d_3^{(1)} + a_{13} = 7 + \infty \\ d_4^{(1)} + a_{14} = 3 + \infty \\ d_5^{(1)} + a_{15} = 0 + 10 \end{pmatrix} = 10$$

$$d_2^{(2)} = \min \begin{pmatrix} d_0^{(1)} + a_{20} = 18 + 6 \\ d_1^{(1)} + a_{21} = 10 + 7 \\ d_3^{(1)} + a_{23} = 7 + 5 \\ d_4^{(1)} + a_{24} = 3 + 8 \\ d_5^{(1)} + a_{25} = 0 + \infty \end{pmatrix} = 11$$

$$d_3^{(2)} = \min \begin{pmatrix} d_0^{(1)} + a_{30} = 18 + 8 \\ d_1^{(1)} + a_{31} = 10 + \infty \\ d_2^{(1)} + a_{32} = 11 + 5 \\ d_4^{(1)} + a_{34} = 3 + 4 \\ d_5^{(1)} + a_{35} = 0 + \infty \end{pmatrix} = 7$$

$$d_4^{(2)} = \min \begin{pmatrix} d_0^{(1)} + a_{40} = 18 + \infty \\ d_1^{(1)} + a_{41} = 10 + \infty \\ d_2^{(1)} + a_{42} = 11 + 8 \\ d_3^{(1)} + a_{43} = 7 + 4 \\ d_5^{(1)} + a_{45} = 0 + 3 \end{pmatrix} = 3$$

$$d_5^{(2)} = \min \begin{pmatrix} d_0^{(1)} + a_{50} = 18 + \infty \\ d_1^{(1)} + a_{51} = 10 + 10 \\ d_2^{(1)} + a_{52} = 11 + \infty \\ d_3^{(1)} + a_{53} = 7 + 2 \\ d_4^{(1)} + a_{54} = 3 + 3 \end{pmatrix} = 0$$

4⁰)

$$d_0^{(3)} = \min \begin{pmatrix} d_1^{(2)} + a_{01} = 10 + 8 \\ d_2^{(2)} + a_{02} = 11 + 6 \\ d_3^{(2)} + a_{03} = 7 + \infty \\ d_4^{(2)} + a_{04} = 3 + \infty \\ d_5^{(2)} + a_{05} = 0 + \infty \end{pmatrix} = 17$$

$$d_1^{(3)} = \min \begin{pmatrix} d_0^{(2)} + a_{10} = \infty + 8 \\ d_2^{(2)} + a_{12} = \infty + 0 \\ d_3^{(2)} + a_{13} = \infty + \infty \\ d_4^{(2)} + a_{14} = 3 + \infty \\ d_5^{(2)} + a_{15} = 0 + 10 \end{pmatrix} = 10$$

$$d_2^{(3)} = \min \begin{pmatrix} d_0^{(2)} + a_{20} = 18 + 6 \\ d_1^{(2)} + a_{21} = 10 + 7 \\ d_3^{(2)} + a_{23} = 7 + 5 \\ d_4^{(2)} + a_{24} = 3 + 8 \\ d_5^{(2)} + a_{25} = 0 + \infty \end{pmatrix} = 11$$

$$d_3^{(3)} = \min \begin{pmatrix} d_0^{(2)} + a_{30} = 17 + 9 \\ d_1^{(2)} + a_{31} = 10 + \infty \\ d_2^{(2)} + a_{32} = 11 + 5 \\ d_4^{(2)} + a_{34} = 3 + 4 \\ d_5^{(2)} + a_{35} = 0 + \infty \end{pmatrix} = 7$$

$$d_4^{(3)} = \min \begin{pmatrix} d_0^{(2)} + a_{40} = 17 + \infty \\ d_1^{(2)} + a_{41} = 10 + \infty \\ d_2^{(2)} + a_{42} = 11 + 8 \\ d_3^{(2)} + a_{43} = 7 + 4 \\ d_5^{(2)} + a_{45} = 0 + 3 \end{pmatrix} = 3$$

$$d_5^{(3)} = \min \begin{pmatrix} d_0^{(2)} + a_{50} = 17 + \infty \\ d_1^{(2)} + a_{51} = 10 + 10 \\ d_2^{(2)} + a_{52} = 11 + \infty \\ d_3^{(2)} + a_{53} = 7 + 2 \\ d_4^{(2)} + a_{54} = 3 + 3 \end{pmatrix} = 0$$

Tablo 3.10. Bellmann-Kalaba Algoritması için $k=0,1,2,3$ ve $i=0,1,2,3,4,5$ için

d_i^k değerlerinin bulunuşu

Bu değerlerle Tablo 3.11. oluşturulabilir.

$d_i^{(k)}$ k	$d_0^{(k)}$	$d_1^{(k)}$	$d_2^{(0,1,2,3)}$	$d_3^{(k)}$	$d_4^{(k)}$	$d_5^{(k)}$
0	∞	10	∞	∞	3	0
1	18	10	11	7	3	0
2	17	10	11	7	3	0
3	17	10	11	7	3	0

Tablo 3.11. d_i^k değerlerinin toplandığı tablo

Bu tablodan $d_i^{(2)} = d_i^{(3)}$, ($i=0,1,2,3,4,5$) olduğu kolayca görülür. Ve burada işlemlere son verilir. Bu durumda 0 tepesinden 5 tepesine en kısa yol ve bu yolun değeri $d_0^{(3)}$ ifadesi geriye doğru gidilerek saptanır. Yani $d_0^{(3)}$ 'in minimum olmasına neden olan $d_i^{(k)}$ 'lar bulunarak adım adım geriye doğru, şu işlemlerle en kısa yol bulunur.

$$\begin{aligned}
 d_0^{(3)} &= d_2^{(2)} + a_{02} \\
 &= d_4^{(1)} + a_{24} + a_{02} \\
 &= d_5^{(0)} + a_{45} + a_{24} + a_{02} \\
 &= 0 + a_{45} + a_{24} + a_{02} \\
 &= a_{02} + a_{24} + a_{45} \\
 &= 6 + 8 + 3 \\
 &= 17
 \end{aligned}$$

Buna göre en kısa yolun

$\mu(0,2,4,5) = \mu(t_0, t_2, t_4, t_5)$ biçiminde bir yol ve bunun değerinin de $q(\mu) = d_0^{(3)} = 17$ olduğu görülür.

Uyarı 3.5.

Bu yöntemle verilen iki tepe arasındaki en kısa yolun uzunluğu bulunur.

Teorem 3.4. (Latin Birleşimi Algoritması)

Herhangi bir G grafinin yollarının hesabına girmeden önce, bir genel hesap yöntemi olan, Latin bileşimi algoritmasını açıklayalım:

Bir G grafinde belli \mathcal{L} özelliğini taşıyan bir yolu oluşturan bir tepeler dizisine bu G grafinin bir \mathcal{L} -latin dizisi adı verilir. Buna göre ikisi de \mathcal{L} özelliğini taşıyan ve biri p ayrıtlı diğeri de q ayrıtlı olan G grafinin iki yolunu göz önüne alalım. Bu yollardan her biri sırasıyla şu iki \mathcal{L} -latin dizisi ile gösterilebilir:

$$S_1 = (t_{i_1}, t_{i_2}, \dots, t_{i_p}, t_k)$$

$$S_2 = (t_1, t_{j_2}, t_{j_2}, \dots, t_{j_p})$$

Bu iki dizinin Latin bileşimi denilen ve * ile gösterilen ikili işlemi

$$S_1 * S_2 = \begin{cases} (t_{i_1}, t_{i_2}, \dots, t_{i_p}, t_k, t_1, t_{j_2}, t_{j_2}, \dots, t_{j_p}) & \text{eğer } t_k = t_p \text{ ve bu dizi bir } L\text{-Latin dizisi ise} \\ \emptyset & \text{(boş dizi), eğer yukarıdaki iki koşuldan biri gerçekleşmiyorsa} \end{cases}$$

ile tanımlanır ve böylece p ve q uzunluklu iki \mathcal{L} -latin dizisinden hareket ederek p+q uzunluğunda yeni bir latin dizisi oluşturulmuş olur. O halde, bu işlemin sistemli bir biçimde kullanılması ile, bir grafta. verilen uzunluktaki yollardan başlayarak daha uzun yollar bulmak mümkündür.

Şimdi, t_k tepesinde biten ve t_i tepesinde başlayan p+1 tepeli \mathcal{L} -Latin dizilerinin alt kümesini C_{t_i, t_k}^p ile gösterelim. Benzer biçimde C_{t_k, t_j}^q ile de t_k tepesinden başlayan ve t_j tepesinde biten \mathcal{L} -latin dizilerinin alt kümesi gösterilir. Yani;

$$C_{t_i, t_k}^p = \{s_1, s_2, \dots, s_H, \dots, s_\alpha\}$$

$$C_{t_k, t_j}^q = \{r_1, r_2, \dots, r_\nu, \dots, r_\beta\}$$

dır. Buradan bileşim kümesi

$$C_{t_i, t_k}^p * C_{t_k, t_j}^q = \{(s_1 * r_1), (s_1 * r_2), \dots, (s_1 * r_\beta), (s_2 * r_1), \dots, (s_\alpha * r_\beta)\}$$

biçiminde tanımlanmaktadır. Yazılışı basitleştirmek için yeni bir simge kullanılır ve bileşim işlemi (dizinin boş olmaması kaydıyla) ikinci dizinin başlangıcını kaldırıp bu iki diziyi birleştirmektedir. Yani,

$s_1 * s_2 = s_1 * s'_2$ ($s_1 * s_2$ dizisi boş değil) dir. Burada s'_2 dizisi, başlangıç tepesi hariç tutulmuş s_2 dizisini ifade etmektedir.

Bu gösteriliş kullanılarak

$$C^p_{t_i \succ t_k} * C^q_{t_k \succ t_j} = C^p_{t_i \succ t_k} \cdot C^q_{t_k \succ t_j}$$

yazılır, burada

$$C^q_{t_k \succ t_j} = \{r'_1, r'_2, \dots, r'_v, \dots, r'_\beta\}$$

dir.

Bir G grafında, aşağıdaki bileşimlerle verilen ve Latin bileşimi işlemlerini ayrıca uygulayarak 3,4,5,...,n tepeli \mathcal{L} -latin dizileri elde edilmektedir.

$$C^2_{t_i \succ t_j} = \bigcup_{k=1}^n (C^1_{t_i \succ t_k} \cdot C^1_{t_k \succ t_j})$$

$$C^3_{t_i \succ t_j} = \bigcup_{k=1}^n (C^2_{t_i \succ t_k} \cdot C^1_{t_k \succ t_j})$$

$$C^4_{t_i \succ t_j} = \bigcup_{k=1}^n (C^3_{t_i \succ t_k} \cdot C^1_{t_k \succ t_j})$$

$$C^n_{t_i \succ t_j} = \bigcup_{k=1}^n (C^{n-1}_{t_i \succ t_k} \cdot C^1_{t_k \succ t_j})$$

burada n, G grafının tepeleri sayısıdır. * işlemi değişmeli olup

$$C^{q+r}_{t_i \succ t_j} = \bigcup_{k=1}^n (C^q_{t_i \succ t_k} \cdot C^r_{t_k \succ t_j})$$

yazılabilmektedir. Bu formül ise matris çarpımını hatırlatmaktadır. Eğer $[M]^p$, i. Satır ve j. sütündeki elemanı

$$C^p_{t_i \succ t_j} = \bigcup_{k=1}^n (C^{p-1}_{t_i \succ t_k} \cdot C^1_{t_k \succ t_j})$$

olan bir matris ise bundan yararlanarak, daha basit olarak

$$[M]^{q+r} = [M]^q \cdot [M]^r$$

biçiminde yazılır. Burada $[M]^r$ elemanları $C^r_{t_k \succ t_j}$ lerdir. Bunlar Latin bileşimi

algoritmasının esasını oluşturmaktadır. Burada \mathcal{L} özelliği, “dizimin bir yol olması”

varsayıldığına göre, bir grafın yollarını belirtmek için bu grafın \mathcal{L} -latin dizilerini belirtmek yeterli olacaktır. Böylece, ardışık olarak

$$[M]^2 = [M]^1 \cdot [M']^1,$$

$$[M]^3 = [M]^2 \cdot [M']^1,$$

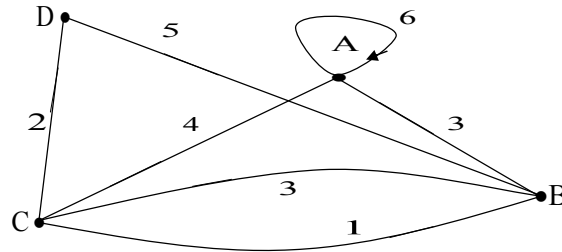
$$[M]^4 = [M]^3 \cdot [M']^1,$$

$$[M]^n = [M]^{n-1} \cdot [M']^1$$

elde edilir. $[M]^1$ matrisi, tepeleri Latin harfleriyle gösterilmiş bir grafın ayrıtlarının bu harflerle belirtildiği bir tepe-tepe matrisidir. Yani, grafın herhangi bir A tepesinde bir bukle varsa bu grafın matrisinde A ile A'nın kesiştiği yere AA yazılarak, grafın A ile B tepeleri arasında bir ayrıt varsa, bu matriste A ile B'nin kesiştiği yere AB yazılarak, eğer bir ayrıt yoksa matriste bu yer boş bırakılarak, $[M]^1$ matrisi elde edilmektedir. $[M']^1$ matrisi ise $[M]^1$ matrisinin elemanlarının sadece son tepeleri alınarak yazılmış şeklidir.

Ayrıca, $[M]^1$ matrisinde grafın ayrıtlarına bağlanan $a_{ij} \geq 0$ değerleri bu matristeki yerlerine yazılarak bir $[D]^1$ matrisi ve daha sonra $[M]^n$ matrisinden de $[D]^n$ ($n=1,2,3,\dots$) matrisleri oluşturulur. Bu matrisin elemanları, grafın tüm tepeleri arasındaki n ayrıtlı yolların değerlerini gösterir.

Latin bileşimi algoritmasını Şekil 3.7.'de verilen grafa uygulayalım:



Şekil 3.7. Latin bileşimi algoritması için bir graf

Önce grafa ait matrisler Tablo 3.12.'deki gibi yazılır.

$$[M]^1 = \begin{matrix} & A & B & C & D \\ A & \begin{bmatrix} AA & AB & - & - \end{bmatrix} \\ B & \begin{bmatrix} - & - & BC & BD \end{bmatrix} \\ C & \begin{bmatrix} CA & CB & - & CD \end{bmatrix} \\ D & \begin{bmatrix} - & - & - & - \end{bmatrix} \end{matrix}, \quad [M']^1 = \begin{matrix} & A & B & C & D \\ A & \begin{bmatrix} A & B & - & - \end{bmatrix} \\ B & \begin{bmatrix} - & - & C & D \end{bmatrix} \\ C & \begin{bmatrix} A & B & - & D \end{bmatrix} \\ D & \begin{bmatrix} - & - & - & - \end{bmatrix} \end{matrix}, \quad [D]^1 = \begin{matrix} & A & B & C & D \\ A & \begin{bmatrix} 6 & 3 & - & - \end{bmatrix} \\ B & \begin{bmatrix} - & - & 3 & 5 \end{bmatrix} \\ C & \begin{bmatrix} 4 & 1 & - & 2 \end{bmatrix} \\ D & \begin{bmatrix} - & - & - & - \end{bmatrix} \end{matrix},$$

Tablo 3.12. $[M]^1$, $[M']^1$ ve $[D]^1$ matrisleri

Bundan sonra bu grafin \mathcal{L} -Latin dizilerini belirtmek için matrisler ardışık olarak oluşturulur. Bu matrisler ($n=4$ 'e kadar) Tablo 3.13.'te verilmiştir.

$$[M]^2 = [M]^1 \cdot [M']^1 = \begin{matrix} & A & B & C & D \\ A & \begin{bmatrix} AAA & AAB & ABC & ABD \end{bmatrix} \\ B & \begin{bmatrix} BCA & BCB & - & BCD \end{bmatrix} \\ C & \begin{bmatrix} CAA & CAB & CBC & CBD \end{bmatrix} \\ D & \begin{bmatrix} - & - & - & - \end{bmatrix} \end{matrix}, \quad [D]^2 = \begin{matrix} & A & B & C & D \\ A & \begin{bmatrix} 12 & 9 & 6 & 8 \end{bmatrix} \\ B & \begin{bmatrix} 7 & 4 & - & 5 \end{bmatrix} \\ C & \begin{bmatrix} 10 & 7 & 4 & 6 \end{bmatrix} \\ D & \begin{bmatrix} - & - & - & - \end{bmatrix} \end{matrix},$$

$$[M]^3 = \begin{matrix} & A & B & C & D \\ A & \begin{bmatrix} AAAA & AAAB & AAABC & AAABD \\ ABCA & ABCB & ABCBC & ABCBD \end{bmatrix} \\ B & \begin{bmatrix} BCAA & BCAB & BCBC & BCBD \end{bmatrix} \\ C & \begin{bmatrix} CAAA & CAAB & CABC & CBCD \\ CBCA & CBCB & CABCD \end{bmatrix} \\ D & \begin{bmatrix} - & - & - & - \end{bmatrix} \end{matrix}, \quad [D]^3 = \begin{matrix} & A & B & C & D \\ A & \begin{bmatrix} 18 & 15 & 12 & 14 \\ 10 & 7 & - & 8 \end{bmatrix} \\ B & \begin{bmatrix} 13 & 10 & 7 & 9 \end{bmatrix} \\ C & \begin{bmatrix} 16 & 13 & 10 & 6 \end{bmatrix} \\ D & \begin{bmatrix} 8 & 5 & - & 12 \end{bmatrix} \end{matrix},$$

$$[M]^4 = \begin{matrix} & A & B & C & D \\ A & \begin{bmatrix} AAAAA & AAAAB & AAABC & AAABD \\ ABCAA & ABCAB & ABCBC & ABCBD \\ AABCA & AABCB & AABCD \end{bmatrix} \\ B & \begin{bmatrix} BCAA & BCAAB & BCABC & BCBCD \\ BCBCA & BCBCB & BCABD \end{bmatrix} \\ C & \begin{bmatrix} CAAAA & CAAAB & CAABC & CAABD \\ CBCAA & CBCAB & CBCBC & CBCBD \\ CABCA & CABCB & CABCD \end{bmatrix} \\ D & \begin{bmatrix} - & - & - & - \end{bmatrix} \end{matrix}, \quad [D]^4 = \begin{matrix} & A & B & C & D \\ A & \begin{bmatrix} 24 & 21 & 18 & 20 \\ 16 & 13 & 10 & 12 \\ 16 & 13 & 10 & 14 \end{bmatrix} \\ B & \begin{bmatrix} 19 & 16 & 13 & 9 \\ 11 & 8 & - & 15 \end{bmatrix} \\ C & \begin{bmatrix} 22 & 19 & 16 & 18 \\ 14 & 11 & 8 & 10 \end{bmatrix} \\ D & \begin{bmatrix} 14 & 13 & - & 12 \end{bmatrix} \end{matrix},$$

Tablo 3.13. $[M]^2$, $[D]^2$, $[M]^3$, $[D]^3$, $[M]^4$ ve $[D]^4$ matrisleri

Bu işlemlere istenirse devam edilebilir. Biz burada işlemleri kesip, elde edilebilecek sonuçlar üzerinde durmak istiyoruz.

\int özelliği, alınan dizilerin uzunluğunun en kısa olması ise, bu grafın her hangi iki tepesi arasındaki, n ayrıtlı en kısa yollar, $[M]^n$ ve $[D]^n$ ($n=1,2,3,4,\dots$) matrislerinden kolayca bulunabilmektedir.

A dan D ye koşulsuz olarak giden

2 ayrıtlı en kısa yol ABD olup değeri 8 dir.

3 ayrıtlı en kısa yol ABCD olup değeri 8 dir.

4 ayrıtlı en kısa yol ABD olup değeri 12 dir.

\int özelliği, alınan dizilerin bir çevre oluşturmaları ise, bu halde dizinin başlangıç ve sonu aynı olacağından grafın çevreleri, $[M]^n$ ($n=1,2,3,4,\dots$) matrisinin tepegen elemanları n ayrıtlı çevreleri verir.

Örneğin C nin 2 ayrıtlı çevresi CBC olup değeri 4 dir.

C nin 3 ayrıtlı çevresi CABC olup değeri 10 dir.

C nin 4 ayrıtlı çevresi CAABC olup değeri 16 dir.

C nin 4 ayrıtlı çevresi CBCBC olup değeri 8 dir.

Yukarıda verilen örneklerde, koşulsuz yolların ve çevrelerin bulunmasını gördük. Şimdi yol ve çevreye bazı koşulların bağlanması halini inceleyelim:

Elementer yollar. $[M]^n$ matrisinin elemanları (dizileri) içinden şöyle saptanır. Bir dizide harfler sadece bir kez bulunuyorsa böyle bir diziye ait yol elementer bir yoldur. Örneğin, A dan D ye giden üç ayrıtlı ABCD yolu elementer bir yoldur.

Bir grafın aynı ayrıtlını iki kez geçmeyen bir yoluna basit yol denir. Örneğin A dan, D ye giden dört ayrıtlı ABCBD yolu basit bir yoldur, fakat bu yol B tepesini iki kez geçtiğinden elementer yol değildir

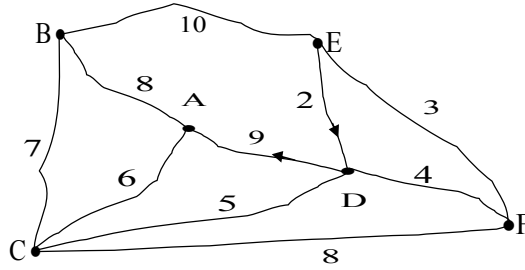
Bir grafın, başlangıç ve son tepesi dışında kalan tüm tepeleri ayrıtlı olan çevresine elementer çevre denir. Örneğin dört ayrıtlı CBCBC çevresi bir elementer çevre değil fakat üç ayrıtlı ABCA çevresi bir elementer çevredir.

Bu tanımlarla getirilen kısıtlayıcı koşulların biri ya da birkaçı altında, verilen bir grafın yol ve çevrelerini yine Latin birleşimi algoritması yardımıyla bulmak mümkündür. Bu algoritma hangi amaçla kullanılacaksa, $[M]^1$ matrisinin bu amca yönelik olarak oluşturulması ve $[M]^n$ ($n=1,2,3,4,\dots$) matrisinde amaca uygun

olmayan dizilerin dikkate alınmaması, yapılacak işlemlerde kolaylık sağlamaktadır.

Şekil 3.6.'da verilen grafi Şekil 3.8.'deki gibi düzenlenirse A dan E ye giden en kısa elementer yolların bulunmasını bir kez de Latin bileşimi algoritması ile yapalım.

∫ özelliği Latin dizilerin minimum uzunlukta olmasıdır. Böyle bir özellik, $[M]^n$ ($n=2,3,4,\dots$) matrisleri oluşturulurken, grafin herhangi iki tepesi arasında bulunan diziler içinden sadece en kısa uzunluktaki elementer yollar seçilir ve diğer yolları, matriste ayrıca göstermek gerekmez. Böylece verilen graf, tepelerine Latin harfleri koyarak yeniden çizilir ve grafa ait matrisler yazılır.



Şekil 3.8. Şekil 3.6.'da verilen grafin Latin harfleriyle düzenlenmiş şekli

$$[M]^1 = \begin{matrix} & \begin{matrix} A & B & C & D & E & F \end{matrix} \\ \begin{matrix} A \\ B \\ C \\ D \\ E \\ F \end{matrix} & \begin{bmatrix} - & AB & AC & - & - & - \\ BA & - & BC & - & BE & - \\ CA & CB & - & CD & - & CF \\ DA & - & DC & - & - & DF \\ - & EB & - & ED & - & EF \\ - & - & FC & FD & FE & - \end{bmatrix} \end{matrix}, \quad [M']^1 = \begin{matrix} & \begin{matrix} A & B & C & D & E & F \end{matrix} \\ \begin{matrix} A \\ B \\ C \\ D \\ E \\ F \end{matrix} & \begin{bmatrix} - & B & C & - & - & - \\ A & - & C & - & E & - \\ A & B & - & D & - & F \\ A & - & C & - & - & F \\ - & B & - & D & - & F \\ - & - & C & D & E & - \end{bmatrix} \end{matrix},$$

$$[D]^1 = \begin{matrix} & \begin{matrix} A & B & C & D & E & F \end{matrix} \\ \begin{matrix} A \\ B \\ C \\ D \\ E \\ F \end{matrix} & \begin{bmatrix} - & 8 & 6 & - & - & - \\ 8 & - & 7 & - & 10 & - \\ 6 & 7 & - & 5 & - & 8 \\ 9 & - & 5 & - & - & 4 \\ - & 10 & - & 2 & - & 3 \\ - & - & 8 & 4 & 3 & - \end{bmatrix} \end{matrix},$$

Tablo 3.14. $[M]^1$, $[M']^1$ ve $[D]^1$ matrisleri

Bundan sonra Latin dizilerini içeren matrislerin oluşturulmasına geçilir ve

$$[M]^2 = \begin{matrix} & \begin{matrix} A & B & C & D & E & F \end{matrix} \\ \begin{matrix} A \\ B \\ C \\ D \\ E \\ F \end{matrix} & \begin{bmatrix} - & ACB & ABC & ACD & ABE & ACF \\ BCA & - & BAC & BCD & - & BEF \\ CDA & CAB & - & CFD & CFE & CDF \\ DCA & DCB & DFC & - & DFE & DCF \\ EDA & - & EDC & EFD & - & EDF \\ FDA & FEB & FDC & FED & - & - \end{bmatrix} \end{matrix}, [D]^2 = \begin{matrix} & \begin{matrix} A & B & C & D & E & F \end{matrix} \\ \begin{matrix} A \\ B \\ C \\ D \\ E \\ F \end{matrix} & \begin{bmatrix} - & 13 & 15 & 11 & 18 & 14 \\ 13 & - & 14 & 12/12 & - & 13 \\ 14 & 14 & - & 12 & 11 & 9 \\ 11 & 12 & 12 & - & 7 & 13 \\ 11 & - & 7 & 7 & - & 6 \\ 13 & 13 & 9 & 5 & - & - \end{bmatrix} \end{matrix},$$

$$[M]^3 = \begin{matrix} & \begin{matrix} A & B & C & D & E & F \end{matrix} \\ \begin{matrix} A \\ B \\ C \\ D \\ E \\ F \end{matrix} & \begin{bmatrix} - & - & - & ABCD & ACFE & ACDF \\ BCDA/BEDA & - & BEDC & BEFD & - & BCDF/BEDF \\ CFDA & CFEB & - & CFED & CDFE & - \\ DFCA & DFEB & - & - & DCFE & - \\ EDCA & EDCB & EFDC & - & - & EDCF \\ FEDA & FDCB & FEDC & - & - & - \end{bmatrix} \end{matrix},$$

$$[D]^3 = \begin{matrix} & \begin{matrix} A & B & C & D & E & F \end{matrix} \\ \begin{matrix} A \\ B \\ C \\ D \\ E \\ F \end{matrix} & \begin{bmatrix} - & - & - & 16 & 17 & 15 \\ 21/21 & - & 17 & 17 & - & 16/16 \\ 21 & 21 & - & 13 & 12 & - \\ 18 & 17 & - & - & 16 & - \\ 13 & 14 & 12 & - & - & 15 \\ 14 & 16 & 10 & - & - & - \end{bmatrix} \end{matrix},$$

$$[M]^4 = \begin{matrix} & \begin{matrix} A & B & C & D & E & F \end{matrix} \\ \begin{matrix} A \\ B \\ C \\ D \\ E \\ F \end{matrix} & \begin{bmatrix} - & ACFEB & - & ACFED & ACDFE & ABCDF \\ BEDCA & - & BEFDC & - & BCDFE & BEDCF \\ CFEDA & CDFEB & - & - & - & - \\ DFEB A & DFCAB/DCFE B & DFEB C & - & - & - \\ EFDCA & EFD C B & - & - & - & - \\ FEDCA & FED C B & FEDAC & - & FDCBE & - \end{bmatrix} \end{matrix},$$

$$[D]^4 = \begin{matrix} & \begin{matrix} A & B & C & D & E & F \end{matrix} \\ \begin{matrix} A \\ B \\ C \\ D \\ E \\ F \end{matrix} & \begin{bmatrix} - & 27 & - & 19 & 18 & 20 \\ 23 & - & 22 & - & 19 & 25 \\ 22 & 23 & - & - & - & - \\ 25 & 26/26 & 24 & - & - & - \\ 18 & 19 & - & - & - & - \\ 16 & 17 & 20 & - & 26 & - \end{bmatrix} \end{matrix},$$

$$[M]^5 = \begin{matrix} & \begin{matrix} A & B & C & D & E & F \end{matrix} \\ \begin{matrix} A \\ B \\ C \\ D \\ E \\ F \end{matrix} & \begin{bmatrix} - & ACD FEB & - & - & ABCDFE & - \\ BEFDCA & - & - & - & - & - \\ CDFEDA & CFEDAB & - & - & - & - \\ DFEBCA & - & DFEBAC & - & DFCABE & - \\ EFDCBA & EFDCAB & - & - & - & - \\ FEDCBA & FEDCAB & - & - & - & - \end{bmatrix} \end{matrix},$$

$$[D]^5 = \begin{matrix} & \begin{matrix} A & B & C & D & E & F \end{matrix} \\ \begin{matrix} A \\ B \\ C \\ D \\ E \\ F \end{matrix} & \begin{bmatrix} - & 28 & - & - & 27 & - \\ 28 & - & - & - & - & - \\ 31 & 30 & - & - & - & - \\ 30 & - & 31 & - & 36 & - \\ 27 & 26 & - & - & - & - \\ 25 & 24 & - & - & - & - \end{bmatrix} \end{matrix}$$

Tablo 3.15. $[M]^2$, $[D]^2$, $[M]^3$, $[D]^3$, $[M]^4$, $[D]^4$, $[M]^5$ ve $[D]^5$ matrisleri

Verilen grafin altı tepesi bulunduğundan, ancak altı harfli elementer diziler (elementer yollar) söz konusu olabilir. Bu nedenle işlemlere burada son verilir. Buradan, A tepesinden E tepesine giden n-ayrıtli elementer yollar $[M]^n$ matrisinden ve bu yolların değerleri de $[D]^n$ matrisinden kolayca görülür. Örneğin,

A tepesinden E tepesine 2 ayrıtli elementer yol ABE olup uzunluğu 18 dir,

A tepesinden E tepesine 3 ayrıtli elementer yol ACFE olup uzunluğu 17 dir,

A tepesinden E tepesine 4 ayrıtli elementer yol ACD FE olup uzunluğu 18 dir,

A tepesinden E tepesine 5 ayrıtli elementer yol ABCDFE olup uzunluğu 27 dir.

Bu yollar içinden, değeri en küçük olan yol problemin cevabını oluşturur. Yani, A dan E ye giden 3 ayrıtli ACFE en kısa yoldur.

Teorem 3.5. (Fuzzy Ağırlıklı Graflar İçin En Kısa Yol Floyd Algoritması)

$G=(V,E)$ fuzzy olmayan bir graf olsun. w_{ij} , E'ye ait olan (v_i, v_i) ayrıtının ağırlığı olsun. ℓ_{ij} , (v_i, v_i) tepelerinin her birine atanmış değeri olsun.

Başlangıçta, $\forall v_i \in V$ için $i \neq j$ ise

$$\ell_{ij} = \begin{cases} w_{ij}, & (v_i, v_i) \in E \text{ için} \\ \infty, & (v_i, v_i) \notin E \text{ için} \end{cases}$$

ve $i=j$ ise $\ell_{ij} = 0$ olarak alırız. Bundan sonraki işimiz ℓ_{ij} değişikliğini içerir, V üzerinde v_k düzenlenmesi için $\min(\ell_{ij}, \ell_{ik} + \ell_{kj})$ ile ℓ_{ij} yer değiştirilir, $V - \{v_k\}$ üzerinde v_i düzenlenir ve v_i 'yi sabitleştirmek için v_j , V üzerinde düzenlenir. İşlemin sonunda, ℓ_{ij} , v_i ve v_j arasındaki en kısa yolun uzunluğu olmuş olur. Açıkça, bu yolda, \tilde{w}_{ij} , pozitif sayılar kabul ettiğimizde de fuzzy ağırlıklı \tilde{w}_{ij} , uygulanarak genişletilebilir. Başlamak için yukarıda yaptığımız gibi $(v_i, v_i) \in E$ için $\tilde{\ell}_{ij} = \tilde{w}_{ij}$, alınır. k, i, j yukarıda belirttiğimiz gibi indisler olmak üzere $\min(\tilde{\ell}_{ij}, \tilde{\ell}_{ik} \oplus \tilde{\ell}_{kj})$ ile $\tilde{\ell}_{ij}$ yer değiştirir.

$P(i, j)$, i ve j arasındaki bütün yolların kümesi ve ℓ_k , k yolunun uzunluğu olmak üzere, v_i ve v_j arasındaki en kısa yolun uzunluğu $\ell_{ij} = \min_{k \in P(i, j)} \ell_k$ olarak tanımlanabilir. ℓ_{ij} , w_{kl} 'nin artan bir fonksiyonudur. Floyd algoritmasının sonucu da w_{kl} nin artan bir fonksiyonudur. Her iki fonksiyon fuzzy olmayan w_{kl} ye uygundur. Böylece \tilde{w}_{kl} fuzzy sayılarına karşılık gelen bu fonksiyonların genişlemesine de uygun düşer. Böylece fuzzy Floyd algoritması tepeler arasındaki fuzzy en kısa yol uzunluğunu da bulur.

v_i ve v_j arasındaki fuzzy en kısa uzunluklu $\tilde{\ell}_{ij}$ yine elde edilir olmasına rağmen, uzunluğu $\tilde{\ell}_{ij}$ olan en kısa yol (yahut en kısa yolların kümesi) artık illaki mevcut olmayabilir. $\tilde{\ell}_{ij} = \min_{k \in P(i, j)} \tilde{\ell}_k$ eşitliği sağlanır. Fakat böyle fuzzy sayılarının \min söz konusu sayılardan illaki birisini sağlaması gerekmediğinden, hiçbir yolun fuzzy uzunluklu $\tilde{\ell}_{ij}$ ye sahip olmaması mümkündür .

k yolunun bir kritik değeri $\text{hgt}(\tilde{\ell}_k \cap \tilde{\ell}_{ij})$ olabilir. Burada $\text{hgt}(\tilde{\ell}_k \cap \tilde{\ell}_{ij}) = \sup_{x \in X} \min(\mu_{\tilde{\ell}_k}(x), \mu_{\tilde{\ell}_{ij}}(x))$ dır.

Bu durum Klein (1991) çalışmasında detaylandırılmıştır. Bu çalışmayı özetle verelim:

Bunun için gerekli olacak bazı temel kavramları verelim. $G(V,E)$ grafi verilmiş olsun. Burada V kümesi grafin tepelerine ve E ise grafin ayrıtlarına karşılık gelir. Bir ayrıt (i,j) tepelerinin bir çifti ile gösterilir. Eğer E, V nin fayırlı elemanlarının sıralı bir çiftinin bir kümesine dönüştürülürse, $G(V,E)$ yönlendirilmiş bir graf olur ve $E, (i,j)$ sıralı çiftinin bir kümesidir. (i,j) sıralı çifti i tepesinden j tepesine giden bir ayrıta karşılık gelir. (i,i) ayrıtı bir bukleye karşılık gelir.

s tepesinden t tepesine bir yol $(s,i_1), (i_1,i_2), \dots, (i_k,t)$ şeklindeki ayrıtların bir dizisidir. Başka bir deyişle s tepesinden t tepesine varılabilir. s den t ye bir yol (s,t) şeklinde gösterilir. Eğer $s \neq t$ ise (s,t) yolu açık, $s=t$ ise (s,t) yolu kapalı bir yoldur. Çevre s hariç hiçbir tepesi tekrarlanmayan ve en az bir ayrıtı olan kapalı (s,s) yoludur. Hiçbir çevre içermeyen bir grafa çevresizdir denir. Bir grafta hem bir başlangıç tepesi hem de bir son tepesi atanabilir. Bunlar bazı aktivitelerin başladığı ve bittiği uç tepeler olarak yorumlanabilir. N tepeli çevre içermeyen yönlü bir grafta, başlangıç 1 tepesi olarak ve bitiş N tepesi olarak numaralanır ve diğer tüm tepeler, $i < j$ olmak üzere her hangi bir (i,j) ayrıtı için numaralanabilir.

Çevre içermeyen yönlendirilmiş grafin bir özel tipi, katlı (layer) graftır. Eğer $|V_k| > 1$ ise bu bir çevre içermeyen yönlendirilmiş bir grafin, tepeler kümesi $V; V_1, V_2, \dots, V_M$ şeklindeki M altlanmaya ayrıştırılabilir. V_k daki tepeler $i, j \in V_k$ için bir (i,j) ayrıtı mevcut olmaması (yani V_k daki iki tepelyi bağlayan bir ayrıt yoktur), sonucu olarak numaralanmıştır. Katlı graflar için genelde V_1 başlangıç tepesidir ve V_M son tepedir.

Eğer her bir (i,j) ayrıtı, bir c_{ij} ağırlığı veya uzunluğu ile temsil edilir ise o zaman bir (s,t) yolu, yoldaki ayrıtların ağırlığının toplamına eş değer olan uzunlukla veya ağırlıkla temsil edilir. Bu s ve t tepeleri arasında min ağırlıklı yolu bulma problemi en kısa yol problemine eşdeğerdir.

Bir grafın en kısa yolunu bulmanın değişik şekilleri vardır. Dinamik programlama (DP) ile şekillenen numaralama algoritmalarını kullanmak en genel metotlardan biridir. Bu modelin çalışabilmesi için grafın yönlendirilmiş ve çevre içermeyen graflar olmalıdır.

1 başlangıç ve N son tepe olmak üzere 1'den N'ye kadar numaralanmış n tepeli bir çevre içermeyen yönlendirilmiş $G(V,E)$ grafi için bir DP en kısa yol problemi formülü

$$f(N) = 0$$

$$f(i) = \min_{i < j} \{c_{ij} + f(j) \mid (i,j) \in E\},$$

olarak verilir. Burada c_{ij} , (i,j) ayrıtının uzunluğunu ve $f(i)$ ise i tepesinden N tepesine olan yollar veya en kısa yolun uzunluğudur. Tamamıyla algoritma i tepesi $f(i)$ ile numaralanmıştır ve numaralar yolun belirlenmesine izin verir.

Bellman iyimserlik prensibinden dolayı bunun tekrarı çok esnek ve bir çok uygulamaya sahiptir. Görülebilir esnekliklerden biri

$$f(N) = 0$$

$$f(i) = \min_{i < j} \{c_{ij} + f(j) \mid (i,j) \in E\},$$

deki toplama işlemi yerine herhangi bir ikili işlemin alınabilmesidir. Bu en kısa yol problemidir ve modellerin temelini oluşturan

$$f(N) = 0$$

$$f(i) = \min_{i < j} \{c_{ij} + f(j) \mid (i,j) \in E\},$$

eşitliğinde verilen DP tekrarlama aşağıda veriliyor.

Genel olarak fuzzyleştirme; tepe sınırlamaları, veya ayrıt uzunlukları, ayrıt kapasiteleri vasıtasıyla şebekeye dahil olmuştur. Literatürde en az kullanılan modellerden biri olmasına rağmen en basit fuzzy en kısa yol problemidir. Fuzzyleştirme bir dizi değişik şekilde şebekeye dahil edilebileceğinden dolayı biz kendimizi iki özel kategori ile sınırlayacağız.

1. Her bir ayrıt uzunluğunun bir fuzzy sayısı olduğu klasik fuzzy en kısa yol problemi
2. Yolun uzunluğunun bir fuzzy sayısı olması ve grafta da her bir ayrıt bir üyelik değerine sahip olması

Klasik fuzzy en kısa yol problemi Dubois ve Prade (1991) tarafından ilk kez ortaya atılmış gibi gözüküyor.

Dubois ve Prade genişletilmiş minimum ortalama ve maksimum ortalama \oplus genişletilmiş toplama kullanılarak bu problemin çözümünü tartışıyorlar. Problemi çözmek için Floyd's ve Ford'un algoritmalarını kullanıyorlar. Var olan bir problemin durumu çözümü

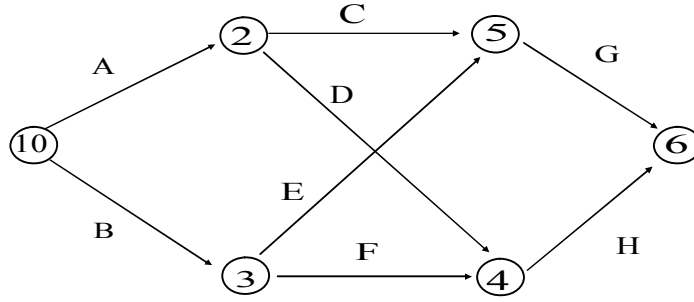
bulabilecek bir fuzzy en kısa yol boyudur. Fakat bazı fuzzy sayıların genişletilmiş minimumları bu sayılardan bazısı olmayabileceğinden söz konusu en kısa yolun en kısa şekli mevcut olmayabilir.

Bu fuzzy en kısa yol problemleri Wilen DP tekrarlamasının terimlerine bağlı olarak incelenebilir. Bu tekrarlamaya Ford algoritmasına oldukça yakındır. Fuzzy sayılarına kolaylıkla genişletilebilir. Fuzzy DP programlaması tekrarı aşağıdaki gibi verilir.

$$\begin{aligned}\tilde{f}(N) &= \{0/1\}, \\ \tilde{f}(i) &= \min_{i < j} \{ \tilde{c}_{ij} + \tilde{f}(j) \mid (i,j) \in E \},\end{aligned}$$

Burada \tilde{c}_{ij} fuzzy ayırıt boyudur, x/y x elemanını ve onun üyelik değeri olan y'yi temsil eder.

Ve $\tilde{f}(i), i \rightarrow N$ olan fuzzy en kısa yolun uzunluğudur.



Şekil 3.9. Floyd Algoritması için bir fuzzy graf

Bu $\tilde{f}(i)$ rutin numaralama olarak görülebilir. Ve her i tepesi bir fuzzy numaralı $\tilde{f}(i)$ ile verilebilir. Yönlendirilmiş çevre içermeyen bir şebekede, $\tilde{f}(1)$ 1 numaralı başlangıç tepesinden N numaralı son tepeye bağlanan en kısa fuzzy mesafesini verir. Dubois ve Prade(1991)'nin belirttiği gibi $\tilde{f}(1)$ fuzzy sayısına karşılık gelen bir yol olmayabilir. Bu durum Şekil 3.9.'da gösterilmiştir. Burada A dan H'a kadar olan harfler \tilde{C}_{ij} ayırıtlarının uzunluklarını temsil ediyor ve aşağıdaki gibi verilir.

$$\tilde{C}_{12}=A=\{1/0.3,2/0.2,3/0.5\},$$

$$\tilde{C}_{13}=B=\{2/0.4,3/0.6\},$$

$$\tilde{C}_{25}=C=\{1/0.5,2/0.8,3/0.2\},$$

$$\tilde{C}_{24}=D=\{3/0.2,4/0.3\},$$

$$\tilde{C}_{35}=E=\{5/0.8,6/0.4\},$$

$$\tilde{C}_{34}=F=\{4/0.6,5/0.2,6/0.8\},$$

$$\tilde{C}_{56}=G=\{3/0.2,4/0.6\},$$

$$\tilde{C}_{46}=H=\{2/0.8,3/0.9\}$$

Bu durumda

$$\tilde{f}(N)=\{0/1\},$$

$$\tilde{f}(i)=\min_{i < j} \{\tilde{c}_{ij} + \tilde{f}(j) \mid (i,j) \in E\},$$

tekrarlaması kullanılarak aşağıdakiler elde edilir.

$$\tilde{f}(6)=\{0/1\},$$

$$\tilde{f}(5)=\{2/0.8,3/0.9\},$$

$$\tilde{f}(4)=\{3/0.2,4/0.3\},$$

$$\tilde{f}(3)=\min\{C+\tilde{f}(4), D+\tilde{f}(5)\},$$

$$=\min\{\{4/0.2,5/0.5,6/0.6,7/0.2\}, \{5/0.2,6/0.3,7/0.3\}\}$$

$$=\{4/0.2,5/0.5,6/0.3,7/0.2\}$$

Bununla birlikte $\tilde{f}(3)$ değeri ayrıtların bir dizisine karşılık gelmez. Mesela 3. tepeden 6. tepeye yol olmayabilir.

Klasik fuzzy en kısa yol mesafesinin bulunması zor değildir. Fakat eğer gerçek bir yola karşılık gelmiyorsa karar vermek için kullanacağımız şey nedir. Bu probleme çözüm getirmek amacıyla fuzzy yol probleminin 2. bir kategorisi dikkate alınmalıdır. Her bir ayrıtı bir fuzzy sayısı olarak görmek yerine her bir mümkün ayrıt uzunluğunun ve yol uzunluğunu fuzzy kümesi olarak ele alalım bu durumda her bir ayrıt içerisinde bir uzunluğa fuzzy kümesinin karşılık getirildiği bir üyelik derecesi vardır. Daha sonra en kısa yol boyunu bulmak için DP tekrarlaması vasıtasıyla bu mümkün olur. Gerçek bir yola karşılık gelmeyip pozitif üyelik dereceli fuzzy kümesindeki her bir değerın şebekede bir yola karşılık gelecek bir fuzzy sayısı olacaktır. Temsili basitleştirmek için ilgilendiğiniz şebekenin katlı olduğunu varsayalım öyle ki M tane kat olsun . Negatif çevreli olmayan herhangi bir şebekenin Bellmann-Ford metodu yardımı ile katlı bir şebekeye dönüşülebileceğini de vurgulayalım.

Her bir ayrıtın 1 ve R arasındaki uzunluk için bir tam sayı değerini alabileceğini de belirtelim. 1'den R'ye kadar tam değerler fuzzy kümeler olsun ve her bir her bir fuzzy kümesindeki I ,her bir a_{ij} ayrıtlarının üyelik derecesinin büyüklüğü $\mu_1(i, j)$ ile gösterilmiş

olsun. Burada $1 \leq I \leq R$ dır. Her bir ayırıt, üyelik değerinin sıralı bir R -sıralısı ile bağlantılıdır. Bu problemi çözmek için değiştirilmiş DP yaklaşımı kullanılır .

Teorem 3.6. (Fuzzy Ağırlıklı Graflar İçin En Kısa Yol Ford Algoritması)

Diğer bir örnek tepelerin ağırlandırıldığı bükle olmaksızın birleştirilmiş yönlendirilmiş grafa uygulanan Ford algoritmasıdır. p_i, V_i nin pozitif ağırlığı olsun tepeler yapılacak işlerin bir kümesini teşkil eder. E ise tepeler arasındaki ilişkileri belirler. V_i in bir fuzzy ağırlığa sahip olduğu varsayılırsa yani V_i işinin \tilde{p}_i nin i bilinmeyen işlem zamanıdır. $P(i)$ ve $S(i)$, sırasıyla V_i den hemen önce ve hemen sonra gelen tepelerin kümesi olsun. V_i işinin en erken başlama zamanı r_i ve en son bitirme zamanı d_i yi veren klasik formül aşağıdaki gibi olur;

$$\tilde{r}_i = \mathop{\text{m\~{a}x}}_{V_j \in P(i)} (\tilde{r}_j \oplus \tilde{p}_j), \quad \tilde{d}_i = \mathop{\text{m\~{i}n}}_{V_j \in S(i)} (\tilde{d}_j \ominus \tilde{p}_j)$$

burada \tilde{r}_i ve \tilde{d}_i sırasıyla V_i işinin fuzzy en erken başlama zamanı ve fuzzy en son bitirme zamanıdır. İşlemi başlatmak için kullanılan öncelik olmak sizin işlerin fuzzy en erken başlama zamanı da fuzzy olabilir. Ford algoritmasının yukarda bahsedilen fuzzylaştırılması Dubois ve Prade (1980)'in (III.3. B.d.) maddesi altında verdikleri anlamıyla geçerlidir. $L - R$ fuzzy sayılarının kullanılması nedeniyle Floyd ve Ford algoritmalarının fuzzy versiyonunun çok fazla hesap gerektirmediğini vurgulayalım.

4. SONUÇ

Ford algoritmasının en kullanışlı taraflarından biri her hangi bir tepenin diğer tüm tepelere olan en kısa yol uzaklığını bir tablo vasıtasıyla bir arada vermesidir. Fakat Ford algoritması bulunan en kısa yolların hangi tepelerden geçtiğini ve kaç ayrıtla en kısa yolların bulunduğunu açık bir şekilde vermemektedir. Yine bu algoritma bukleli yollarda işlem görmediğinden Latin birleşimi algoritmasına ihtiyaç vardır. Bellmann kalaba algoritması ise bir noktanın sadece bir noktaya olan en kısa yol uzaklığını vermektedir. Yani bu algoritma ile verilen iki tepe arasındaki en kısa yolun uzunluğu bulunur. Bu durum Bellmann kalabaya göre Ford algoritmasının daha kullanışlı olduğunu bize gösterir.

Dijkstra Algoritması ise Ford algoritması gibi sadece bir tepeden diğer tüm tepelere olan en kısa yolları buluyor. Ayrıca bir tepeden diğer tepelere en kısa yol birden fazla ise bunlardan sadece bir tanesini buluyor

Latin bileşimi yöntemi, elemanları arasında ikili bir bağıntı bulunan kümelerde ortaya çıkan problemlerin hepsine uygulanabilen genel bir yöntemdir. Ayrıca bu yöntemin, makine dilinde kolayca programlanabilmesi, elle yapılması güç olan hesap işlemlerini ortadan kaldırmaktadır. Bu yönüyle de bu algoritmanın önemi artmaktadır. Bu algoritma Ford algoritmasında olduğu gibi bir noktanın diğer tüm noktalara olan en kısa yol uzunluklarını vermekle birlikte; tüm noktaların diğer tüm noktalara olan en kısa yol uzunluklarını hem kaç ayrıtla bulunduğunu hem de hangi tepelerden geçildiğini belli ettiğinden çok daha kullanışlıdır.

EK 1.

Teorem 3.2. Ford Algoritmasının bilgisayarın BASIC dilinde Programı

```
DECLARE SUB MIN (P( ), EK!)
OPTION BASE 1
INPUT "matris boyutunu giriniz:"; N
DIM Q(N, N), LAMBDA (N), P(N)
FOR I = 1 TO N
    FOR J = 1 TO N
        INPUT Q(I, J)
    NEXT J, I
LAMBDA(I) = 0
FOR J = 2 TO N
    JJ = J: SAY% = 0
    FOR I = 1 TO J - 1
        IF Q(I, J) <> 0! THEN
            SAY% = SAY% + 1
            P(SAY%) = LAMBDA(I) + Q(I, J)
        END IF
    NEXT I
    CALL MIN(P( ), EK)
    LAMBDA(J) = EK
NEXT J
FOR I = 1 TO N
    FOR J = 1 TO N
        IF Q(I, J) <> 0! THEN
            IF LAMBDA(J) - LAMBDA(I) > Q(I, J) THEN
                LAMBDA(J) = LAMBDA(I) + Q(I, J)
            END IF
        END IF
    NEXT J, I
NEXT J, I
CLS
PRINT "LAMBDA DEGERLERI"
FOR I = 1 TO N
    PRINT LAMBDA(I) ;
NEXT I
END

SUB MIN (P( ), EK)
    EK = P (1)
    FOR I = 2 TO JJ - 1
        IF EK > P(I) THEN EK = P (I)
    NEXT I
END SUB
```

KAYNAKLAR

Bakoğlu, H., 1982. *Doğrusal Programlama*. Ege Üniversitesi Yayınları, 385.

Chen, L. H., Chiang, C. H., 2004. An intelligent control system with a multi-objective self-exploration process. *Fuzzy Sets and Systems*, 143: 275-294.

Chepoi, V., Dumitrescu, D., 1999. Fuzzy clustering with structural constraints. *Fuzzy Sets and Systems*, 105: 91-97.

Christofides, N., 1975. *Graph Theory An Algorithmic Approach*. Academic Press Inc. 400.

Colliot, O., Tuzikov, A. V., Cesar, R. M., Bloch, I., 2004. Approximate reflectional symmetries of fuzzy objects with an application in model-based object recognition. *Fuzzy Sets and Systems*, 147: 141-163.

Devillez, A., Billaudel, P., Lecolier, G. V., 2002. A fuzzy hybrid hierarchical clustering method with a new criterion able to find the optimal partition. *Fuzzy Sets and Systems*, 128: 323-338.

Dick, S., Meeks, A., Last, M., Bunke, H., Kandel, A., 2004, Data mining in software metrics databases. *Fuzzy Sets and Systems*, 145: 81-110.

Dubois, D., Prade, H., 1980. *Fuzzy Sets and Systems Theory and Applications*. Academic Pres. 393.

Harary, F., 1969. *Graph Theory*. Addison- Wesley. 274.

Jamei, M., Mahfoul, M., Linkens, D. A., 2004. Elicitation and fine-tuning of fuzzy control rules using symbiotic evolution. *Fuzzy Sets and Systems*, 147: 57-74.

Kaufmann, A., 1975. *Introduction to the Theory of Fuzzy Subsets*. Aeademie Pres. 416.

- Klein, C. M., 1991. Fuzzy shortest paths. *Fuzzy and Systems*, 39 (1):27-41.
- Kuo, R. J., Chen, C. H., Hwang Y. C., 2001. An intelligent stock trading decision support system through integration of genetic algorithm based fuzzy neural network and artificial neural network. *Fuzzy Sets and Systems*, 118: 21-45.
- Lin, K., Chern, M., 1993. The Fuzzy shortest path problem and its most vital arcs. *Fuzzy Sets and Systems*, 58 (3): 343-353.
- Ma, J., Fan, Z. P., Jiang, Y. P., Mao J. Y., Ma, L., (baskıda), A method for repairing the inconsistency of fuzzy preference relations. *Fuzzy Sets and Systems*.
- Robinson, V. B., 2000. Individual and multipersonal fuzzy spatial relations acquired using human-machine interaction. *Fuzzy Sets and Systems*, 113: 133-145.
- Seshu, S., Reed, B. M., 1961. *Linear Graphs and Electrical Networks*. Addison-Wesley Publishing Company Inc. 315.
- Shitong, W., Jianfu, C., 1993. Backward fuzzy heuristic search algorithm Fbhao for fuzzy general and/or graph. *Fuzzy Sets and Systems*, 60,(1): 67-75.
- Yan, J. T., ve Hsiao, P. Y., 1994. A fuzzy clustering algorithm for graph bisection. *Information processing letters*, 52: 259-263.

ÖZGEÇMİŞ

1975 yılında Artvin'in Borçka ilçesine bağlı Güzelyurt köyünde doğdu. İlkokulu Güreşen Selimiye Köyünde, ortaokulu Borçka ilçe merkezinde, liseyi Hopa ilçe merkezinde tamamladı. Yüksek öğrenimini ise Dokuz Eylül Üniversitesi Eğitim Fakültesi Matematik Bölümünde tamamladı. 1996'dan beri Milli Eğitimin ilköğretim okullarında matematik öğretmenliği yapmaktadır. 2004 yılında Yüzüncü Yıl Üniversitesi Fen Bilimleri Enstitüsü Matematik Ana Bilim Dalında Yüksek Lisansa başladı. Şu anda Hopa 14 Mart İlköğretim okulunda matematik öğretmeni olarak çalışmaktadır. Evli ve iki çocuk babasıdır.

Mehmet DURMUŞ