



Pajek

Pajek
XXL

Pajek
3XL

**Büyük Ağların Analizi
ve Görselleştirilmesi için Yazılım**

**Pajek Kılavuzu
Komut Listesi ve Kısa Açıklamaları
versiyon 5.17**

**Andrej Mrvar ve Vladimir Batagelj
Hasan Güçlü (çeviri)**

**Ljubljana, 13 Nisan 2023
İstanbul, 15 Ağustos 2023**

©1996, 2023 A. Mrvar, V. Batagelj., Ticarî olmayan kullanımlarda ücretsizdir.

©2023 Hasan Güçlü (Çeviri)

PdfLaTeX, versiyonu 1 Ekim 2003

Andrej Mrvar
Faculty of Social Sciences
University of Ljubljana, Slovenia
<http://mrvar.fdv.uni-lj.si/>
andrej.mrvar@fdv.uni-lj.si

Vladimir Batagelj
Department of Mathematics, FMF
University of Ljubljana, Slovenia
<http://vldowiki.fmf.uni-lj.si/doku.php?id=vb>
vladimir.batagelj@fmf.uni-lj.si

Hasan Güçlü (çeviri)
Department of Biostatistics and Biomedical Informatics
Istanbul Medeniyet University, School of Medicine
<https://avesis.medeniyet.edu.tr/hasan.guclu/iletisim>
gucluh@gmail.com

İçindekiler

1	Pajek	3
2	Veri tipleri	7
3	Ana Pencere Menüsü	9
3.1	File (Dosya)	9
3.2	Network (Ağ)	15
3.3	Networks (Ağlar)	44
3.4	VertexID (DüğümKimlik)	47
3.5	Operations (İşlemler)	47
3.6	Partition (Sınıflama)	59
3.7	Partitions (Sınıflamalar)	61
3.8	Vector (Vektör)	62
3.9	Vectors (Vektörler)	64
3.10	Permutation (Permütasyon)	65
3.11	Permutations (Permütasyonlar)	65
3.12	Cluster (Küme)	66
3.13	Hierarchy (Hiyerarşi)	66
3.14	Options (Seçenekler)	66
3.15	Info (Bilgi)	71
3.16	Tools (Araçlar)	71
4	Çizim Penceresi Araçları	75
4.1	Ana Pencere Çizim (Draw) Menüsü	75
4.2	Layout (Yerleşim)	76
4.3	Layers (Tabakalar)	78
4.4	GraphOnly (Sadece Ağ)	80
4.5	Default (Varsayılan)	80
4.6	Previous (Önceki)	80
4.7	Redraw (Tekrar Çiz)	80
4.8	Next (Sonraki)	80
4.9	ZoomOut (Küçült)	80
4.10	Options (Seçenekler)	80
4.11	Export (Aktar)	86
4.12	Spin (Döndür)	89
4.13	Move (Kaydır)	89
4.14	Info (Bilgi)	90
4.15	FishEye (Balık Gözü)	90

5	EPS/SVG/X3D/VRML'e Aktarma	91
5.1	Varsayılan parametreler	91
5.2	EPS, SVG, X3D ve VRML için Varsayılan Parametreler Penceresi	91
5.3	Resimlerin EPS/SVG'ye aktarılması – parametrelerin girdi dosyasında tanıtılması	96
5.4	Pajek Resimlerinde Unicode Kullanımı	100
6	Pajek'de Makrolar	103
6.1	Makro nedir?	103
6.2	Makro nasıl kaydedilir?	103
6.3	Makro nasıl çalıştırılır?	103
6.4	Örnek	103
6.5	Kurulum dosyasında mevcut makrolar	104
6.5.1	Soyağacı ve diğer çevrimsiz ağlar için hazır olarak gelen makrolar	104
6.5.2	Akrabalık ilişkilerini hesaplamak için hazırlanmış makrolar	105
6.6	Oturumu Tekrar Etmek	105
6.7	Komut çalıştır	106
6.8	Son komutun tekrar edilmesi	106
6.9	Son komutla oluşturulan sabitlerin skalerlere kaydedilmesi	106
7	Pajek'de Blok Modelleme	108
7.1	MDL dosyaları	108
7.2	MDL dosyası örnekleri	110
7.2.1	Düzenli bloklar	110
7.2.2	Köşegen bloklar (kümelenme)	110
7.2.3	Çevrimsel olmayan model (yukarı)	111
7.2.4	Simetrik kümeli çevrimsel olmayan model (aşağı)	111
7.2.5	Merkez-çevre	111
7.2.6	Düzenli yol	111
7.2.7	Düzenli zincir	112
7.2.8	Davis.net dosyası için iki modlu 'standart model'	112
8	Pajek'de renkler	113
9	Pajek'e nasıl atıfta bulunulur?	115

1 Pajek



Pajek, binlerce hatta milyonlarca düğümü olan *büyük ağların* analizi ve görselleştirilmesi için tasarlanmış ve Windows işletim sisteminde çalışan bir uygulamadır. Sloven dilinde *pajek* kelimesi örümcek anlamına gelir. **Pajek**'in en son sürümü, ticari olmayan kullanımlar için kendine ait İnternet sayfasında ücretsiz olarak kullanıma sunulmuştur:

<http://mrvar.fdv.uni-lj.si/pajek/>

Pajek'in geliştirilmesine Kasım 1996'da başladık. Pajek, Delphi (Pascal) programlama diliyle geliştirilmiştir. **Pajek**'in bazı fonksiyonları Matjaž Zaveršnik tarafından yazılmıştır. Pajek'in geliştirmedeki temel motivasyonumuz, büyük ağ verilerinin artık bilgisayarda kullanmaya hazır dijital hale gelmiş olması ve **Pajek**'in bu ağların analizi ve görselleştirilmesi için araçlar sağlaması gerektiğiydi. Bu ağlar arasında, işbirliği ağları, kimyada organik moleküller, protein-reseptör etkileşim ağları, soybilim ağları, İnternet ağları, atıf ağları, yayılma/difüzyon (AIDS, haberler, yenilikler) ağları, veri madenciliği (2-modlu ağlar) vb. ağlar sayılabilir. Büyük ağ veri koleksiyonuna şu adresten ulaşabilirsiniz:

<http://vlado.fmf.uni-lj.si/pub/networks/data/>

Pajek'in tasarımı, daha önce geliştirdiğimiz çizge (ağ) veri yapısı ve algoritma kütüphaneleri olan Graph ve X-graph, ağ analizi ve görselleştirme programları STRAN, RelCalc, Draw, Energ ve SGML tabanlı çizge açıklama işaretleme dili NetML'deki deneyimlerimize dayanmaktadır.

<http://vlado.fmf.uni-lj.si/pub/networks/default.htm>

Pajek-XXL ve **Pajek-3XL**, Pajek programının özel sürümleridir. Bu programlar bilgisayar belleğinde daha az yer kaplarlar. Aynı ağ için Pajek'e göre en az 2-3 kat daha az fiziksel bellek gerektirirler ve böylece bilgisayar belleğini yoğun kullanan işlemler (örneğin, rassal ağların oluşturulması, çıkarma, küçültme vb.) daha hızlı gerçekleştirilir. Kılavuzun geri kalanında, **Pajek-XXL** ve **Pajek-3XL**'e özgü komutlar kırmızı kahverengi renkte yazılmıştır.

Pajek-XXL ve **Pajek-3XL** hakkında ayrıntılı bilgi için:

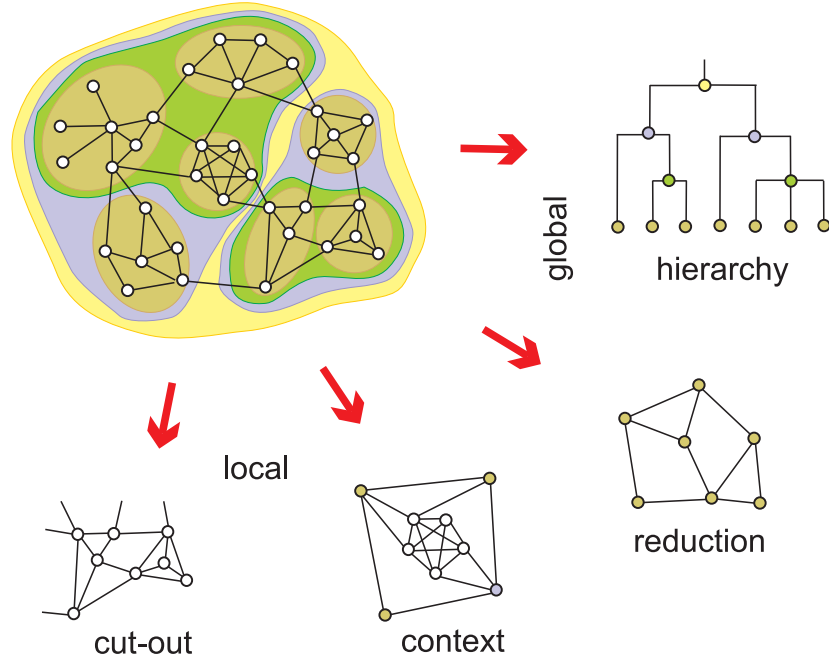
<http://mrvar.fdv.uni-lj.si/pajek/PajekXXL.htm>

veya:

<http://mrvar2.fdv.uni-lj.si/pajek/PajekXXL.htm>



Şekil 1: Pa jek/Örümcek



Şekil 2: Büyük ağlara farklı yaklaşımlar

Pa jek'in tasarımındaki temel hedefler şunlardır:

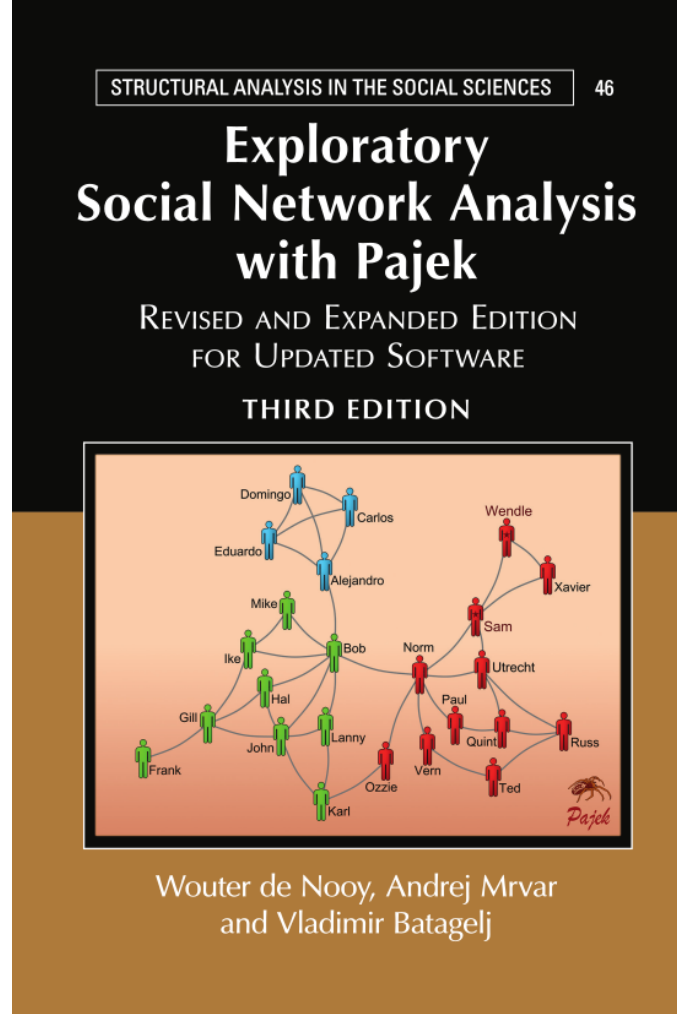
- Büyük bir ağın (*tekrarlamalı*) *ayrıştırma* yoluyla daha küçük ağlara ayrıştırılması ve bunların sonra daha özel yöntemlerle incelenmesi için soyutlama işlemini

desteklemek,

- Kullanıcıya güçlü *görselleştirme* araçları sağlamak,
- Büyük ağların analizi için *etkili (ikinci dereceden daha hızlı) algoritmaları* uygulamak

Pa jek ile yapabileceğimiz çok şey var. Örneğin bir ağda kümeleri (bileşenleri, ‘önemli’ düğümlerin komşuluklarını, çekirdekleri vb.) bulmak, aynı kümeye ait düğümleri *ayırma* ve bağlamdaki olası diğer parçalarla birlikte göstermek (detaylı yerel görünüm), kümelerdeki düğümleri *küçültmek* ve kümeler arasındaki ilişkileri göstermek (genel görünüm)...

Normal (yönlü, yönsüz, karışık) ağların yanı sıra, **Pa jek** aynı zamanda *çoklu ilişkili ağları*, *iki modlu ağları* (iki ayrık düğüm kümesi arasındaki ağırlıklı yada ağırlıksız ağlar), ve *zamansal ağları* (zaman içinde değişen dinamik ağlar) da analiz edebilmektedir.



Şekil 3: *Pajek* kitabı

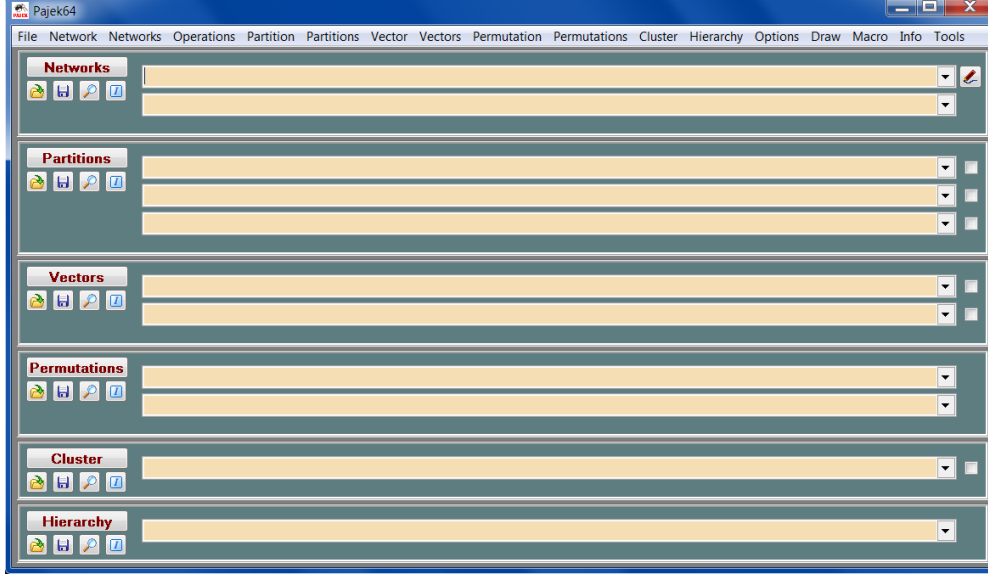
Bu kılavuz **Pajek**'in son versiyonunda yapılabilecek bütün işlemlerin kısa açıklamalarını içerir. Sosyal ağ teorisine ait her kavram açıklanmamıştır. Yeni kullanıcılara **Pajek** kitabını okumalarını tavsiye ediyoruz. [31]

de Nooy W., Mrvar A., Batagelj V. (2018) *Exploratory Social Network Analysis With Pajek*: Güncelleştirilmiş yazılım için gözden geçirilmiş ve genişletilmiş 3. baskı. Structural Analysis in the Social Sciences 46, Cambridge University Press, 2018.

Pajek ile *ağ analizi* konusunda genel bilgi edinmek için NICTA çalıştayını sunumuna bakabilirsiniz. [5]

2 Veri tipleri

Pa jek’de altı farklı veri tipi kullanılmaktadır.



Şekil 4: Pa jek Ana Penceresi

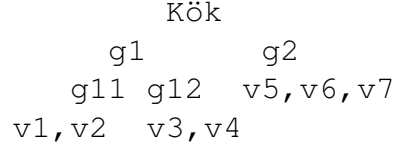
1. **Ağ** (network) – Temel veri tipi (düğümler ve bağlantılar). Çizge de denir. Bağlantılar tek yönlü *yay* olabileceği gibi yönsüz *kenar* da olabilir. Varsayılan dosya uzantısı: **.net**. Ağ, bir girdi dosyasında farklı formatlarla temsil edilebilir:

- yay/kenar formatı (örneğin 1 2 – 1’den 2’ye bağlantı)
- yay/kenar listesi formatı (örneğin 1 2 3 – 1’den 2’ye ve 1’den 3’e bağlantı)
- matris formatı (komşuluk matrisi)
- UCINET, GEDCOM, kimyasal formatlar...

Girdi dosyasına ayrıca ağ görselleştirilmesi için ekstra bilgiler eklenebilir. Bu **EPS/SVG/VRML’e dönüştürme** kısmında açıklanmıştır.

2. **Sınıflama** (partition) – Her bir düğüm için ait olduğu sınıfı gösterir. Varsayılan dosya uzantısı: **.clu**.
3. **Permütasyon** (permutation) – Düğümlerin yeniden bir sıralanışıdır. Varsayılan dosya uzantısı: **.per**.

4. **Küme** (cluster) – düğümlerin bir alt kümesidir (örneğin bir sınıf). Varsayılan dosya uzantısı: **.cls**.
5. **Hiyerarşi** (hierarchy) – Hiyerarşik bir şekilde sıralanmış düğümler. Örnek:



Burada Kök'ün iki alt grubu vardır: g1 ve g2. g2 bir yapaktır ve v5,v6 ve v7 düğümlerinden oluşan bir kümedir. g1'in iki alt grubu vardır: g11 ve g12... Varsayılan dosya uzantısı: **.hie**.

6. **Vektör** (vector) – Her bir düğüme ait sayısal özellikleri içerir (reel sayılardan oluşur). Varsayılan dosya uzantısı: **.vec**.

Seçilen ağ, sınıflama ve diğerlerinin üzerine çift tıklama yaparak ekranda gösterilmesini sağlayabilirsiniz.

Pa jek'in ana penceresindeki (bkz. Figure 4) işlemler girdi olarak kullandıkları veri tiplerine göre düzenlenmiştir.

Permütasyonlar, sınıflamalar ve vektörler düğümlerin istatistik olarak sınıflayıcı, sıralayıcı ve sayısal gibi farklı ölçüm seviyelerindeki özelliklerini tutmak için kullanılabilirler.



Şekil 5: *Örümcek ağı*; Fotoğraf: Vladimir Batagelj.

3 Ana Pencere Menüsü

3.1 File (Dosya)

Altı çeşit veri için girdi/çıkıı işlemleri.

- **Network – N (Ağ)**
 - **Read (Oku)** – Ascii veya Unicode UTF8 metin dosyasından ağ oku.
 - **Read Time Events** (zamansal olaylar formatında oku) – Dosyada zamansal olaylarla tanımlanan zaman ağını oku. Tablo 1'e bakınız. Özellikler listesi s boş da olabilir. Eğer birkaç bağlantı iki düğümü birbirine bağlıyorsa, komutun hangi satıra uygulanacağını belirlemek için :k (k. satır) gibi ek bir etiket verilmelidir. Örneğin, HE :3 14 37 komutu, 14 ve 37 düğümlerini birbirine bağlayan üçüncü bağlantıyı gizlemektedir.

Zaman olayları kullanılarak açıklanan bir zaman ağı örneği:

```
*Vertices 3
```

Tablo 1: Zamansal olaylar listesi

Olay	Açıklama
TI t	ilk olaylar – zaman noktası t başladığında
TE t	son olaylar – takip eden olaylar zaman noktası t 'de olan olaylardır
AV vns	n etiketli ve s özelliklerine sahip düğüm v 'yi ekle
HV v	düğüm v 'yi gizle
SV v	düğüm v 'yi göster
DV v	düğüm v 'yi sil
AA uvs	s özelliğine sahip (u,v) yayını ekle
HA uw	yay (u,v) 'yi gizle
SA uw	yay (u,v) 'yi göster
DA uw	yay (u,v) 'yi sil
AE uvs	s özelliklerine sahip kenar $(u:v)$ 'yi ekle
HE uw	kenar $(u:v)$ 'yi gizle
SE uw	kenar $(u:v)$ 'ti göster
DE uw	kenar $(u:v)$ 'yi sil
CV vs	düğüm özelliğini değiştir – düğüm v 'nin özelliğini s ile değiştir
CA uvs	yay özelliğini değiştir – yay (u,v) 'nin özelliğini s ile değiştir
CE uvs	kenar özelliğini değiştir – kenar $(u:v)$ 'nin özelliğini s ile deęiltir
CT uw	tip değiştir – bağlantı (u,v) 'nin yön tipini değiştir
CD uw	yay (u,v) 'nin yönünü değiştir
PE uvs	yay çifti (u,v) ve (v,u) 'yu s özelliklerine sahip tek bir kenar $(u:v)$ ile değiştir
AP uvs	s özelliklerine sahip (u,v) ve (v,u) yay çiftini ekle
DP uw	(u,v) ve (v,u) yay çiftini sil
EP uvs	$(u:v)$ kenarını s özelliklerine sahip (u,v) ve (v,u) yay çifti ile değiştir

```

*Events
TI 1
AV 2 "b"
TE 3
HV 2
TI 4
AV 3 "e"
TI 5
AV 1 "a"
TI 6
AE 1 3 1
TI 7
SV 2
AE 1 2 1
TE 7
DE 1 2
DV 2
TE 8
DE 1 3
TE 10
HV 1
TI 12
SV 1
TE 14
DV 1

```

Zaman ağını **zaman aralıkları** kullanarak tanımlamak da mümkündür.

- **Save** (Kaydet) – Seçili ağı Ascii/Unicode UTF8 koduyla dosyaya kaydet.
Eğer ağ aşağıdaki gibi 5 ilişkiden (yaydan) oluşan bir Ore ağı ise bir GEDCOM dosyası olarak kaydedilebilir: 1. Wi→Hu (Karı→Koca), 2. Mo→Da (Anne→Kız), 3. Mo→So (Anne→Oğul), 4. Fa→Da (Baba→Kız), 5. Fa→So (Baba→Oğul)
Bir diğer imkan da Pajek Ore ağıdır: 1.Fa→Ch (Baba→Çocuk), 2.Mo→Ch (Anne→Çocuk), 3.Hu-Wi (Koca→Karı) (kenar), veya 1.Pa→Ch (Ebeveyn → Çocuk), 3.Hu→Wi (Koca→Karı) (kenar).
- **Save as Time Events** (Zamansal Olaylar Olarak Kaydet) – Zamansal bir ağı zamansal olaylar formatında kaydet.
- **View/Edit** (Gör/Değiştir) – Ağı gör ve/veya değiştir. Düğümü seç, komşularını göster ve:
 - * seçili düğüme giren veya çıkan yay oluşturun (Newline yazılı satıra fareniz sol tuşu ile çift tıklayarak)
 - * bağlantıları silin (farenizin sol tuşuyla çift tıklayarak);
 - * bağlantının değerini değiştirin (farenin sağ tuşuyla tek tıklayarak);
 - * yeni bir görünmez düğüm kullanarak bağlantıyı iki dik bağlantıya dönüştürün (farenizin orta tuşuyla tek tıklayarak).
- **Change Label** (Etiket Değiştir) seçili ağın etiketini değiştirin.

- **Dispose** (Sil) seçili ağı bellekten sil.
- **Export as Matrix to [EPS, SVG + PDF]** ([EPS, SVG + PDF]'e matris olarak aktarın) – matrisi EPS (SVG ve PDF) formatlarına dönüştür:
 - * **Original** (orijinal) – varsayılan numaralandırmayı kullan (1-modlu ve 2-modlu ağlar için).
 - * **Using Partition** (sınıflamayı kullanarak) – seçili sınıflamayı kullan. Metin penceresinde sınıflar arasındaki bağlantıların sayısı ve yoğunluğu (ve seçilen iki sınıftaki düğümler) gösterilir. Ayrıca matris yoğunluğunun gölgeleme olarak gösterildiği EPS'e dönüştürülür:
 1. **Structural** (Yapısal) – Yoğunluklar sınıflar arasındaki mümkün olan en yüksek bağlantı sayısına göre normalize edilir. Bu işlem yoğun ağlar için uygundur.
 2. **Delta** (Delta) – Yoğunluklar sınıftaki en yüksek iç ve dış komşuya sahip düğümlere göre normalize edilir. Bu işlem seyrek ağlar için uygundur.
 - * **Using Permutation** (Permütasyon Kullanarak) – mevcut permütasyonu satır ve kolonları yeniden sıralamak için kullan. Bu seçenek hem bir modlu hem de iki modlu ağlar için kullanılabilir.
 - * **Using Permutation + Partition** (Permütasyon + Sınıflama'yı Kullanarak) – mevcut permütasyon ve sınıflamayı kullan. Sınıflama tarafından tanımlanan farklı sınıfları bölmek için bağlantılar çizilir. Bu seçenek hem bir modlu hem de iki modlu ağlar için kullanılabilir.
 - * **Options** (Seçenekler) – EPS'e çevirmede kullanılabilecek ekstra parametreleri değiştir.
 - **Diamonds for Negative Values, Circles for 0** (negatif değerler için elmas, 0 değeri için çember) – Pozitif değerler için kareler, negatif değerler için elmas (baklava dilimi), 0 değeri için ise çemberler kullanılır. Bu işlem siyah-beyaz yazıcı çıkışı almak için uygundur.
 - **Diamonds, Circles and Lines in GreyScale** (elmas, çember ve bağlantılar gri tonlarda) – Elmas, çember ve bağlantılar gri tonlarda çizilir (kırmızı, yeşil ve mavi yerine).
 - **Labels on Top/Right** (etiketler üst/sağda) – Etiketler matrisin üst ve sağında yazılır – uzun etiketler için uygundur.
 - **Only Black Borders** (sadece siyah sınırlar) – Matristeki bütün karelerin siyah sınırları vardır yoksa koyu kareler beyaz ve açık renkli karelerin siyah sınırları olur.
 - **Thick Boundary Line** (kalın sınır çizgileri) – Kümeleri bölmek için kalın çizgiler kullan.

- **Large Squares/Diamonds/Circles** (büyük kare/elmas/çember)
 - Daha büyük ya da daha küçük kare, elmas ve çember kullan.
- **Use Partition Colors for Vertex Labels** (düğüm etiketi olarak sınıflama renklerini kullan) – Düğümlerin etiketlerini sınıflama renklerini kullanarak göster.
- **VertexID – ID** – (düğüm belirteçleri).
 - **Read** (Oku) Bir metin dosyasından belirteçleri oku.
 - **Save** (Kaydet) Seçili belirteçleri bir Ascii dosyasına kaydet.
 - **Save to NET File as Vertex Labels** (NET dosyasına düğüm etiketleri olarak kaydet) – Belirteçleri bir NET dosyasına sonradan **Network/Transform/Add/Vertex Labels/from File(s)** (Ağ / Transform / Ekle / Düğüm Etiketleri / Dosya(lar)dan) komutu ile kullanılabilecek şekilde kaydet.
 - **View/Edit** (Gör/Değiştir) Düğüm belirteçlerini değiştir.
 - **Change Label** (Etiket Değiştir) Seçili belirteçlerin etiketlerini değiştir.
 - **Dispose** (Sil) Seçili belirteçleri bellekten sil.
- **Partition – C** (Sınıflama)
 - **Read** (Oku) Sınıflamayı bir metin dosyasından oku.
 - **Save** (Kaydet) Seçili sınıflamayı bir metin dosyasına kaydet.
 - **View/Edit** (Gör/Değiştir) Sınıflamayı gör/değiştir (düğümleri sınıflara ayır).
 - **Change Label** (Etiket Değiştir) Seçili sınıflamanın etiketini değiştir.
 - **Dispose** (Sil) Seçili sınıflamayı bellekten sil.
- **Vector – V** (Vektör)
 - **Read** (Oku) Vektörü bir metin dosyasından oku.
 - **Save** (Kaydet) Seçili vektörleri bir metin dosyasına kaydet. Eğer vektör kimlik numaralarını temsil eden bir küme mevcutsa, ilgili kimlik numaralarına sahip tüm vektörler aynı çıkış dosyasına kaydedilir. Vektörün kimliği, seçilen vektöre *V* tuşuna basarak (boş bir küme önce oluşturulmalıdır) eklenebilir. Tüm vektörlerin aynı boyutlarda olması gerekmektedir.
 - **View/Edit** (Gör/Değiştir) Vektörü gör/değiştir (Vektörün bileşenlerini değiştir).

- **Change Label** (Etiket Değiştir) Seçili vektörün etiketini değiştir.
- **Dispose** (Sil) Seçili vektör bellekten sil.
- **Permutation – P** (Permütasyon)
 - **Read** (Oku) Permütasyonu bir metin dosyasından oku.
 - **Save** (Kaydet) Seçili permütasyonu bir metin dosyasına kaydet.
 - **View/Edit** (Gör/Değiştir) Permütasyonu gör/değiştir (İki düğümün yerlerini değiştir).
 - **Change Label** (Etiket Değiştir) Seçili permütasyonun etiketini değiştir.
 - **Dispose** (Sil) Seçili permütasyonu bellekten sil.
- **Cluster – S** (Küme)
 - **Read** (Oku) Kümeyi bir metin dosyasından oku.
 - **Save** (Kaydet) Seçili kümeyi bir metin dosyasına kaydet.
 - **View/Edit** (Gör/Değiştir) Kümeyi gör/değiştir (düğüm ekle ve çıkar).
 - **Change Label** (Etiket Değiştir) Seçili kümenin etiketini değiştir.
 - **Dispose** (Sil) Seçili küme bellekten sil.
- **Hierarchy – H** (Hiyerarşi)
 - **Read** (Oku) Hiyerarşiyi bir metin dosyasından oku.
 - **Save** (Kaydet) Seçili hiyerarşiyi bir metin dosyasına kaydet.
 - **View/Edit** (Gör/Değiştir) Hiyerarşiyi gör/değiştir (seçili bir boğuma ait düğümleri (alt ağaç) göster veya tip ve etiketlerini değiştir). Düğümler hiyerarşide yukarı ve aşağı alınabilirler.
 - **Change Label** (Etiket Değiştir) Seçili hiyerarşinin etiketini değiştir.
 - **Dispose** (Sil) Seçili hiyerarşiyi bellekten sil.
 - **Export as Dendrogram to [EPS, SVG + PDF]** (Hiyerarşiyi [EPS, SVG+PDF] formatında bir dendrograma çevir) – Hiyerarşinin EPS (SVG ve PDF) formatında bir dendrogramını çiz. Sadece ikili hiyerarşilerde çalışır. Benzemezlilikler hiyerarşideki boğum adlarında ve [] işaretleri arasında depolanabilir. Hiyerarşik kümelene ve ilişkisel kısıtlılıklı kümelene elde edilirken bunlar otomatik olarak oluşturulur.
- **Pa jek Project File** (Proje Dosyası) – *.paj

- **Read** (Oku) Bir **Pa jek** proje dosyasındaki bütün veri nesnelarını yükle (Ağlar, Sınıflamalar, Permütasyonlar, Kümeler, Hiyerarşiler ve vektörler).
- **Save** (Kaydet) Mevcut bütün veri nesnelarını bir **Pa jek** proje dosyasına kaydet.
- **Ini File** (Ini Dosyası)
 - **Load** (Yükle) – **Pa jek** ayarlarını bir ini dosyasından (* .ini) yükle.
 - **Save** (Kaydet) – **Pa jek**'in mevcut ayarları bir ini dosyasına (* .ini) yaz.
- **Exit** (Çıkış) – Pa jek programı kapatılır.

3.2 Network (Ağ)

Girdi olarak yalnızca bir ağa ihtiyaç duyan işlemler. Rassal ağlar için buna da ihtiyaç yoktur.

- **Create Random Network** (Rassal Ağ Oluştur) – Seçili büyüklüğe sahip bir rassal ağ oluştur.
 - **Total No. of Arcs** (Toplam Yay Sayısı) – Seçili büyüklüğe ve verilen yay sayısına sahip yönlü ve rassal bir ağ oluştur.
 - **Vertices Output Degree** (Dış dereceli düğümler) – Seçili büyüklüğe ve her bir düğüm için verilen aralıkta dış dereceye sahip düğümlerden oluşan yönlü ve rassal bir ağ oluştur.
 - **Bernoulli/Poisson Network** (Bernoulli/Poisson Ağı) – Bernoulli/Poisson yöntemine göre yönlü/yönsüz, çevrimsiz, iki modlu rassal bir ağ oluştur. Bernoulli/Poisson yönteminde her bir bağlantı verilen bir p olasılığı ile seçilir. **Pa jek**'de ise büyük ve sığ ağlar için çok küçük bir sayı olan bu p parametresi yerine daha kullanışlı bir parametre olan *ortalama derece* \bar{d} kullanılır. Aralarındaki bağıntı aşağıdaki gibi yazılabilir: $\bar{d} = \frac{1}{n} \sum_{v \in V} \deg(v) = \frac{2m}{n}$ ve $m = pM$ burada $n = |V|$ (düğüm sayısı), $\deg(v)$ (düğüm v 'nin derecesi), $m = |L|$ (bağlantı sayısı) ve M ise ağda maksimum mümkün olabilecek bağlantı sayısıdır. Örneğin yönsüz ağlar için $M = n(n - 1)/2$.
 - **Scale Free** (Ölçeksiz Ağ) – Yönlü/yönsüz veya çevrimsiz bir ölçeksiz ağ oluşturur. Burada kullanılan yöntem *ölçeksiz ağlar* oluşturmak için [56]'de açıklanan yöntemin biraz daha geliştirilmiş halidir. Ağın büyümesi

sırasında her bir basamakta k kenarlı yeni bir düğüm ağı eklenir. Kenarların uç noktaları aşağıda verilen olasılığa göre rassal bir şekilde seçilen düğümlere eklenir.

$$\Pr(v) = \alpha \frac{\text{içderece}(v)}{|E|} + \beta \frac{\text{dışderece}(v)}{|E|} + \gamma \frac{1}{|V|}$$

burada $\alpha + \beta + \gamma = 1$. Ayrıca $\sum_{v \in V} \Pr(v) = 1$.

- **Small World** (Küçük Dünya Ağı) – Rassal bir küçük dünya ağı oluşturur. [12]'yi inceleyiniz.
- **Extended Model** (Genişletilmiş model) – Albert ve Barabasi [3] tarafından geliştirilen modele göre rassal bir ağ oluşturur.
- **Create New Network** (Yeni Ağ Oluştur) – Seçili ağdan yeni bir ağ oluşturur (boş veya tam ağ).
 - **Empty Network** (Ağı boşalt) – Verilen sayıda düğüme sahip hiç bağlantısı olmayan yeni bir ağ oluşturur.
 - **Complete Network** (Tam Ağ) – Verilen sayıda düğüme ve bütün mümkün bağlantılara sahip yönlü/yönsüz bir ağ oluşturur.
 - **with Bi-Connected Components stored as Relation Numbers İle** (İlişki Numarası Olarak Kaydedilmiş İkili-Bağlantılı Bileşenler) – Seçili ağın ikili-bağlantılı bileşenleri. Articulation points (Kesim noktaları) birçok sınıfa aittir bundan dolayı sonuç bir sınıflamaya kaydedilemez (ikili-bağlantılı bileşenler bir hiyerarşiye kaydedilir). Pejek'in yeni versiyonlarında bunlar ayrıca çoklu ilişkili ağ olarak da kaydedilebilir. Bir bileşende olması gereken en düşük düğüm sayısı seçilebilir. Sonuçta sadece bir ikili bileşene ait olan düğümleri, ikili bileşen dışındaki düğümleri ve kesim noktalarını içeren bir sınıflama elde edilir: ikili bileşen dışındaki düğümler 0 kümesinde (sınıfında), her bir ikili bileşen 1'den başlayarak toplam ikili bileşen sayısına kadar numaralandırılır ve içindeki düğümler bu numaraları sınıf olarak alırlar. Kesim noktaları ise sınıf numarası olarak 999999998 alırlar. Ayrıca kesim noktalarını içeren bir vektör de oluşturulur: her bir düğümün ait olduğu ikili-bağlantılı bileşenlerin numarası verilir.
 - **with Ring Counts stored as Line Values** (Döngü Sayıları Bağlantı Değeri Olarak Kaydedilmiş) – her bir bağlantı kaç defa tanımlı bir döngüye aittir.
 - * **3'lü Döngüler** (Halka) – Her bir bağlantı için bağlı olduğu 3'lü döngüleri sayar.

- **Undirected** (Yönsüz) – yönsüz ağlar için – yönsüz 3'lü döngüleri sayar.
- **Directed** (Yönlü) – yönlü ağlar için – çevrimsel (**cy**), geçişli (**tr**) ve bütün 3'lü döngüleri sayar veya her bir bağlantının kaç defa geçişli bir kısayol olduğu durumları sayar. Bkz. Şekil 10.
- * **4-Rings** (4'lü döngüler) – Her bir bağlantı için o bağlantının ait olduğu 4'lü döngüleri sayar.
 - **Undirected** (Yönsüz) – yönsüz ağlar için – yönsüz 4'lü döngüleri sayar.
 - **Directed** (Yönlü) – yönlü ağlar için – çevrimsel (**cy**), elmas (**dia**), soyağacı (**gen**) ve geçişli (**tr**) ve bütün 4'lü döngüleri sayar veya her bir bağlantının kaç kere geçişli olduğunu sayar. Bkz. Şekil 11.
- **SubNetwork with Paths** (Yollarla Alt Ağ)
 - * **One shortest Path between Two Vertices** (İki Düğüm Arasındaki En Kısa Yol) – İki düğüm arasındaki en kısa yolu bulur. Sonuçta yeni bir ağ elde edilir. Ağdaki bağlantı değerleri eğer düğümler arasındaki mesafeleri temsil ediyorsa hesaba katılabilir. Çizgesel teorik mesafe hesaplanacaksa (ki daha hızlı hesaplanabilir) bağlantı değerleri kullanılmaz.
 - * **All Shortest Paths between Two Vertices** (İki Düğüm Arasındaki Bütün En Kısa Yollar) – İki düğüm arasındaki bütün en kısa yolları bul. Sonuçta yeni bir ağ elde edilir. Ağdaki bağlantı değerleri eğer düğümler arasındaki mesafeleri temsil ediyorsa hesaba katılabilir. Çizgesel teorik mesafe hesaplanacaksa (ki daha hızlı hesaplanabilir) bağlantı değerleri kullanılmaz.
 - * **Walks with Limited Length between Two Vertices** (İki Düğüm Arasında Sınırlı Uzunlukta Yürüyüş) – İki düğüm arasında sınırlı maksimum uzunluğa sahip bütün yürüyüşleri bulur.
 - * **Geodesics Matrices*** (Jeodezik Matrisler*) – En kısa yol matrisi ile jeodezik sayım matrisini hesapla (*yalnızca küçük ağlar için*).
 - * **Info on Diameter** (Ağ Çapı Hakkında Bilgi) – Ağın çapını bulur – ağda düğüm çiftleri arasında bütün en kısa yollar arasında en uzun olanın uzunluğu. Bütün ağdaki yollar incelenir bundan dolayı işlem büyük ağlarda (2000'den fazla düğümü olan ağlarda) uzun sürebilir.
- **SubNetwork with Flows** (Akışlarla Alt Ağlar)
 - * **Maximum Flow between Two Vertices** (İki Düğüm Arasında Maksimum Akış) – Seçili iki düğme arasında maksimum akışı

bulur. Bu algoritma doldurulacak yollar arar ve bunlar arasında en kısa olanı seçer. Bu algoritma teknik olarak da gerçek bir akışın olduğu ve bağlantılarının kapasite değeri taşıdığı bir ağda da kullanılabilir. Bütün değerlerin 1 olduğu ağların analizinde de kullanmak mümkündür. Sonuçta iki düğümü ve aralarında maksimum akış katkıda bulunan bağlantıları de içeren yeni bir ağ oluşturulur.

- * **Maximum Flow between Pairs in Cluster*** (Bir Kümedeki Çiftler Arasında Maksimum Akış – Bir kümedeki bütün düğümlerin arasındaki maksimum akışı bulur. Sonuçta bağlantıların değerlerinin maksimum akış olduğu bir ağ oluşturulur. Bu algoritma biraz yavaştır: Bu nedenle yalnızca düğüm sayısının az olduğu küçük ağ ve küme-lerde kullanın.

– Transform (Transform)

- * **Transpose 1-Mode** (Bir Modlu Ağı Traspoze Et) – Bir modlu bir ağın bütün yaylarının yönlerini ters çevirerek transpoze eder.
- * **Remove (Sil)**
 - **Selected Vertices** (Seçili Düğümler) – Seçili düğümler ağdan silinir.
 - **All Edges** (Bütün Kenarlar) – Seçili ağdaki bütün kenarlar (yönsüz bağlantılar) silinir.
 - **All Arcs** (Bütün Yaylar) – Seçili ağdaki bütün yaylar (yönlü bağlantılar) silinir.
 - **Multiple Lines** (Çoklu bağlantılar) – Seçili ağdaki bütün çoklu bağlantıları sil.
 - 1. **Sum Values** (Değerleri Topla) – Bütün silinen bağlantıların değerleri toplanır ve yeni bir bağlantı olarak ilgili iki düğümün arasına eklenir.
 - 2. **Number of Lines** (Bağlantı Sayısı) – Bütün silinen bağlantıların sayısı yeni bir bağlantı olarak ilgili iki düğümün arasına eklenir.
 - 3. **Min Value** (Minimum Değer) – İki düğüm arasındaki bağlantılarından en küçük ağırlıklı (değerli) olanı seçilir.
 - 4. **Max Value** (Maksimum Değer) – İki düğüm arasındaki bağlantılarından en büyük ağırlıklı (değerli) olanı seçilir.
 - 5. **Single Line** (Tek Bağlantı) – Düğümler arasında değeri 1 olan yalnızca bir bağlantı oluşturulur
 - **Loops** (İlmekler) – Seçili ağdaki bütün ilmekleri sil.
 - **Lines with Value** (Bağlantı Değerleri)

1. **lower than** (belirli bir değerden az olanlar) – Seçili ağdaki bağlantı değeri belirli bir değerden düşük olan bütün bağlantıları sil.
2. **higher than** (belirli bir değerden yüksek olanlar) – Seçili ağdaki bağlantı değeri belirli bir değerden yüksek olan bütün bağlantıları sil.
3. **within interval** (bir aralıkta olanlar) – Seçili ağdaki bağlantı değeri belirli bir aralıkta olan bütün bağlantıları sil.
- **all Arcs from each Vertex except** (Her bir düğümden çıkan bütün yaylarını ancak)
 1. **k with Lowest Line Values** (en düşük ağırlıklı k tanesi hariç) – Her bir düğümün dış bağlantı değerlerini küçükten büyüğe sırala ve en küçük k tanesi hariç diğerlerini sil.
 2. **k with Highest Line Values** (en yüksek ağırlıklı k tanesi hariç) – Her bir düğümün dış bağlantı değerlerini büyükten küçüğe sırala ve en büyük k tanesi hariç diğerlerini sil.
 3. **Keep Arcs with Value equal to the k-th Value** (Değeri k. olan bağlantıları tut) – Değeri k. değere eşit olan bağlantılarla ilgili ne yapılacağına karar ver (bunları silelim mi silmeye- lim mi?)
- **Triangle** (Üçgen) – Alt veya üst üçgene ait olan yayları sil.
- * **Add** (Ekle) – Ağa yeni düğümler, bağlantılar veya düğüm/bağlantı etiketleri ekle.
 - **Vertices** (Düğümler) – Ağı yeni bir ağa kopyala. Ağ boyutu seçilen düğüm sayısı kadar artırılabilir (bağlantısı olmayan ekstra düğümler eklenir).
 - **Vertex Labels** (Düğüm Etiketleri)
 1. **Default** (Varsayılan) – Mevcut düğümlerin etiketlerini varsayılan düğüm etiketleriyle değiştir. ($v1, v2...$).
 2. **from File(s)** (Dosya(lar)dan) – Varsayılan düğüm etiketlerini ($v1, v2...$) dosyada verilen etiketlerle değiştir. İki modlu ağ durumunda etiketleri içeren iki dosya seçilebilir (her bir mod için bir dosya). Bu özellikle her bir modun etiketlerinin ayrı dosyalarda olduğu ve ağların çarpması ile elde edilen iki modlu ağlar için önemlidir.
 3. **and Descriptions from File(s)** (Düğüm Bilgilerini Dosya(lar)dan Oku) – Varsayılan düğüm etiketlerini ($v1, v2...$) etiketleri içeren bir dosyadan oku. Yine iki modlu ağ durumunda etiketleri içeren iki dosya seçilebilir (her bir mod için bir

dosya). Bu yöntemle düğüm şekilleri, renkle, vb. ekstra bilgiler de değiştirilebilir.

- **Line Labels as Line Values** (Bağlantı Etiketleri Bağlantı Değeri Olsun) – Bağlantıların etiketlerini bağlantı değerleri ile değiştir veya yoksa yenisini oluştur. Kullanılacak ondalık basamak sayısı Draw (çizim) penceresinde bağlantıları işaretlemek için kullanılan basamak sayısı kadardır.
- **Sibling edges** (kardeş kenarlar) – Aşağıdaki ortaklığa sahip düğümlere kardeş kenarı ekle
 1. **Input** (İç) – ata yayı
 2. **Output** (Dış) – çocuk yayı
- * **Edges** → **Arcs** (Kenarlar → Yaylar) – Bütün kenarları her iki yönde yaylara çevir (yönlü bir ağ oluşur).
- * **Arcs** → **Edges** (Yaylar → Kenarlar)
 - **All** (Tümünü) – Bütün yayları yönsüz kenarlara çevir (yönsüz bir ağ oluşur).
 - **Bidirected only** (İki yönlü olanları) – Yalnızca iki yönde de yay varsa bunları kenara çevir:
 1. **Sum Values** (Değerleri Topla) – Yeni kenarın değeri her iki yayın değerinin toplamıdır.
 2. **Min Value** (Minimum Değer) – Yeni kenarın değeri yay değerlerinden küçük olanıdır.
 3. **Max Value** (Maksimum Değer) – Yeni kenarın değeri yay değerlerinden büyük olanıdır.
- * **Bidirected Arcs** → **Arc** (İki Yönlü Yaylar → Yay)
 - **Select Min Value** (Minimum Değeri Seç) – Ağdaki düğümler arasında iki yönlü yaylar mevcutsa yalnızca değeri düşük olanı bırak ve değeri yüksek olanı sil. Eğer değerler eşitse iki yayı bir kenarla değiştir.
 - **Select Max Value** (Maksimum Değeri Seç) – Ağdaki düğümler arasında iki yönlü yaylar mevcutsa yalnızca değeri yüksek olanı bırak ve değeri düşük olanı sil. Eğer değerler eşitse iki yayı bir kenarla değiştir.
- * **Line Values** (Bağlantı Değerleri) – Bağlantı değerlerinin transformasyonu:
 - **Set All Line Values to 1** (Bütün Bağlantı Değerlerini 1'e Eşitle) – Bütün bağlantı değerlerini 1 yapar.

- **Recode** (Tekrar Kodla) – Seçili aralıklara göre bağlantı değerlerinin bir frekans dağılımını gösterir ve bağlantı değerlerini bu şekilde tekrar kodlar.
- **Multiply by** (Çarp) bütün değerleri bir sayı ile çarp.
- **Divide by** (Böl) bütün değerleri bir sayıya böl.
- **Add Constant** (Sabit Ekle) bütün değerlere bir sayı ekle.
- **Constant** (Sabit) – Seçili bir sayı ve bağlantı değerinin minimum veya maksimumu.
- **Absolute** (Mutlak Değer) bütün değerlerin mutlak değerini al.
- **Absolute + Sqrt** (Mutlak Değer + Karekök) – bütün değerlerin mutlak değerini ve sonra karekökünü al.
- **Truncate** (Ondalık basamakları sil) – bütün değerlerin ondalık kısımlarını at.
- **Exp** (eksponansiyelini al) – bütün değerlerin eksponansiyelini al.
- **Ln Logaritma** (Tabii Logaritmasını Al) – bütün değerlerin tabii logaritmasını al.
- **Power** (Üs Al) – bütün değerlerin belirli bir sayı ile üssünü al.
- **Normalize** (Normalize et)
 1. **Sum** (Toplam) – toplama göre normalize et. Böylece toplamları 1 olur.
 2. **Max** (Maksimum) –maksimum değere göre normalize et. Böylece en büyük değer 1 olur.
- **Replace Arc Values with Ranks** (Yay Değerlerini Sıra Numarasıyla Değiştir) – Yönlü bir ağda bütün yayları yay değerine göre artan veya azalan sıraya koy ve bağlantı değerlerini sıra numarasıyla değiştir.
- * **Reduction** (İndirgeme)
 - **Degree** (Derece) – (Özyinelemeli) bir şekilde ağın belirli bir değerden düşük dereceye sahip bütün düğümlerini sil (İç, dış ve toplam dereceye göre yapılabilir. Bu işlem seçili bir kümedeki düğümlere de uygulanabilir.
 - **Pathfinder*** (Yol Bulucu*) – bağlantı değerlerinin benzemezlilik olduğu ağırlıklı bir ağda bağlantıları sil. Yalnızca üçgen eşitsizliğini bozmayan bağlantılar bırakılır. Ekstra bir parametre de r (sıfırdan büyük bir reel sayı) verilmelidir. Bu parametre ağdaki yolların uzunluğunun hesaplanmasında kul-

lanılan ölçeğin bir parametresidir (Minkowski mesafesi):

$$a \boxed{r} b = \sqrt[r]{a^r + b^r}$$

Genelde $r = 1$, $r = 2$, veya $r = \infty$ değerleri kullanılır:

$$r = 1 \Rightarrow a \boxed{r} b = a + b$$

$$r = 2 \Rightarrow a \boxed{r} b = \sqrt{a^2 + b^2}$$

$$r = \infty \Rightarrow a \boxed{r} b = \max(a, b).$$

Algoritmanın hızına göre bu yaklaşık 10 bine kadar düğümü olan orta büyüklükte ağlara uygulanabilir. Detaylar:

http://en.wikipedia.org/wiki/Pathfinder_network

- **Hierarchy** (Hiyerarşi) – Özyinelemeli bir şekilde ağda 0 veya 1 komşusu olan bütün düğümleri sil. Sonuçta daha basit bir ağ ve silinen düğümlerin bir hiyerarşisi elde edilir. Orijinal ağ daha sonra tekrar elde edilebilir (eğer bağlantıların yönünü kaldırırsak).
- **Subdivisions** (Alt Bölümler) – Özyinelemeli bir şekilde ağda tam olarak 2 komşusu olan bütün düğümleri (ve bağlantılarını da) sil ve onların yerine bu komşular arasında doğrudan bir bağlantı ekle. Sonuçta (çizilmesi) daha basit bir ağ elde edilir. Orijinal ağ tekrar bulunamaz.
- **Design flow graph** (Dizayn Akış Ağı) Ağın yapısal parçalarını McCabe yöntemine göre indirge, kod ve akış ağları için [51].
- * **Replace Loops with Duplicates of Vertices** (İlmekleri Düğümlerin Kopyalarıyla Değiştir) – İlmeklere sahip olan bir ağı aşağıdaki gibi ilmeksiz bir ağa çevirir:
 - İlmeği olan her bir düğümün ilmeksiz bir kopyasını oluştur.
 - Kopya düğümü orijinal düğüme ilmek değerine eşit ağırlıklı bir bağlantı ile bağlar
 - *Optionally* (İsteğe Bağlı Olarak): Kopya düğümü orijinal düğümün bağlı olduğu diğer bütün düğümlere de bağla. Bağlantı değeri olarak ise orijinal düğümlerle bağlantı değerini kullan.
- * **1-Mode to 2-Mode** (Bir Modlu Ağı iki Modlu Ağa Çevir) – Herhangi bir bir modlu ağı iki modlu ağa çevir.
- * **Incidence Network** (Kesişim Ağı) – Komşuluk matrisini *kesişim matrisine* çevir. Kesişim matrisi **Çizgi Çizgesi** oluşturmak için kullanılabilir.
- * **Sort Lines** (Bağlantıları Sırala) –

- **Neighbors around Vertices** (Düğümün Etrafındaki Komşular) – Her bir düğümün bağlantılarını diğer uçtaki düğüme göre artana doğru sıraya diz
 - **Line Values** (Bağlantı Değerleri) – Bağlantıları değerine göre azalan ya da artana doğru sıraya diz.
 - **Line Values around Vertices** (Düğümün Etrafındaki Bağlantı Değerleri) – Her bir düğümün bağlantılarını bağlantı değerine göre artan ya da azalana doğru sıraya diz.
- **Create Partition** (Sınıflama Oluştur) – Ağı sınıflandırır. Sonuçta bir sıralama oluşur.
 - **Degree** (Derece)
 - * **Input** (İç) – Her bir düğümün iç derecesi.
 - * **Output** (Dış) – Her bir düğümün dış derecesi.
 - * **All** (Tümü) – Her bir düğümün bütün bağlantılarının sayısı (derecesi).
 - **Components** (Bileşenler)
 - * **Weak** (Zayıf) – Seçili ağın zayıf bileşenleri.
 - * **Strong** (Güçlü) – Seçili ağın güçlü bileşenleri.
 - * **Strong-Periodic** (Güçlü-Periyodik) – Seçili ağın güçlü-periyodik bileşenleri – güçlü şekilde bağlanmış bileşenler daha sonra periyotlarına göre tekrar ayrılır.
 - **k -Neighbors** (k -Komşular) – Aşağıdaki şarta uyan bütün düğümleri seç
 - * **Input** (İç) ...düğümün seçilen düğüme en fazla k adımda ulaşabiliriz
 - * **Output** (Dış) ...düğümün seçili düğümün en fazla k adımda ulaşılabilir
 - * **All** (Tümü) ...İç ve dış komşuların toplamı (bağlantıların yönü yok sayılır)
Sonuçta verilen düğüme uzaklığa göre belirli sınıflara ait düğümlerden oluşan bir sınıflama oluşturulur. Ulaşılamayan düğümlerin sınıf numarası 999999998 olur. Bu sınıflama elde edildikten sonra alt ağ çekilebilir.
 - **k -Core** (k -Çekirdek) – k -çekirdek verilen ağın her düğümün aynı çekirdekte en az k komşusu olduğu bir alt ağıdır:
 - * **Input** (İç) ... düğüme gelen bağlantılar.
 - * **Output** (Dış) ... düğümün çıkan bağlantılar.
 - * **All** (Tümü) ... bütün komşular.

– **Valued Core** (Değerli Çekirdek) – Genelleştirilmiş k -çekirdek: Bağlantıları (komşuları) saymak yerine bağlantıların değerini kullanır. Değerli çekirdeğin hesaplanmasında bağlantıların sayısı veya maksimum değeri kullanılabilir: Eşik değere (val) göre toplam değerli çekirdek verilen ağın bir alt ağıdır ve aynı çekirdeğe ait düğümlere (veya düğümlerden) olan bağlantıların değerlerinin toplamı en az val değerine sahiptir. Eşik değere (val) göre maksimum değerli çekirdek de verilen ağın bir alt ağıdır ve aynı çekirdeğe ait düğümlere (veya düğümlerden) olan bağlantıların değerlerinin maksimumu en az val değerine sahiptir. Eşik değerler önceden verilmelidir. Eşik değeri belirlemenin iki yolu vardır:

- * **First Threshold and Step** (İlk Eşik ve Adım) – Önce bir eşik değeri seçin sonra bu eşik değeri artırılma miktarını seçin.
- * **Selected Thresholds** (Seçili Eşikler) – Eşik değerleri (artan sayılar şeklinde) bir vektör olarak verilir.

Ayrıca (bir modlu ağlar için), iç, dış ve tüm değerli çekirdekler bulunabilir.

– **Communities** (Topluluklar) – Ağdaki toplulukları bul.

- * **Louvain Method** (Louvain Yöntemi) – Louvain yöntemini kullanarak toplulukları bul. Bağlantı değerleri dikkate alınır. Bu algoritma farklı seviyelerde çalışır. Eğer ağ bir işaretli ağ ise (en azından bir bağlantı negatif değere sahipse) algoritmanın özel bir şekli kullanılır. Bu algortmada çözünürlük (Resolution) adında ekstra bir parametre kullanılabilir: çözünürlüğün 1 olması standart Louvain metodunu çalıştırır, daha yüksek çözünürlük değerleri daha fazla sayıda küme oluşturur, daha düşük değerler ise daha düşük. İki farklı algoritma kullanılabilir:

- **Multi-Level Coarsening + Single Refinement** (Çok seviyeli kabalaştırma + Tek seviyeli inceleştirme) – İnceleştirme sadece son seviyede elde edilen sınıflama üzerinde yapılır (en kaba sınıflama).
- **Multi-Level Coarsening + Multi-Level Refinement** (Çok seviyeli kabalaştırma + Çok seviyeli inceleştirme) – özyinelemeli bir şekilde elde edilen her bir seviyede önce kabalaştırma yapar ve sonra inceleştirme. Bu algoritma sonucu elde edilen modüleriteler genelde daha yüksektir. Algoritmanın detayları için aşağıdaki çalışmaya göz atılabilir:

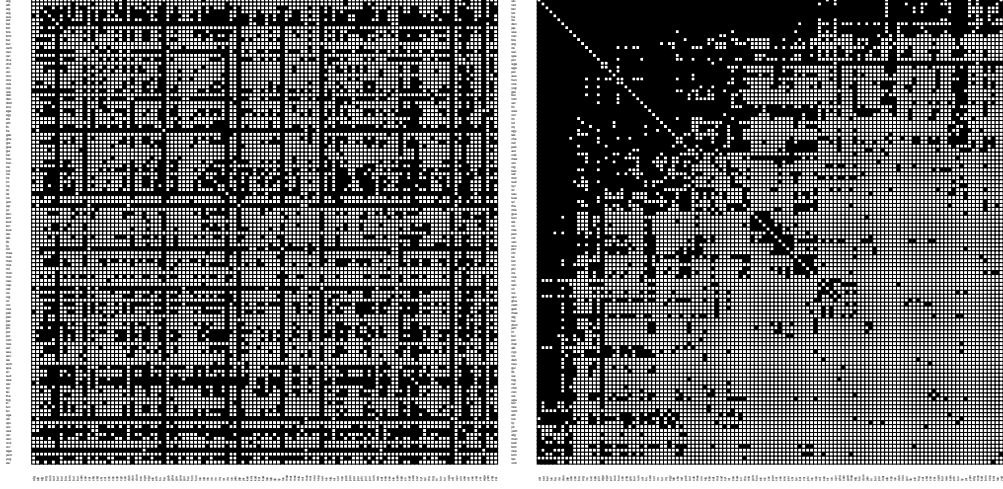
<http://dl.acm.org/citation.cfm?id=1970376>

- * **VOS Clustering** (VOS Kümelenmesi) – Louvain yöntemine ben-

zerdir. Her ikisi de çözünürlük parametres kullanır fakat Louvain yönteminde *modülerite* optimize edilirken VOS kümeleneğinde *VOS kalite fonksiyonu* optimize edilir. Detaylar için VOSviewer sayfası incelenebilir: <http://www.vosviewer.com/>.

Bağlantı değerleri her zaman pozitif alınır bundan dolayı bu algoritma işaretli ağlar için uygun değildir. Bunun için de iki farklı algoritma kullanılabilir:

- **Multi-Level Coarsening + Single Refinement** (Çok seviyeli kabalaştırma + Tek seviyeli inceleştirme) – İnceleştirme sadece son seviyede elde edilen sınıflama üzerinde yapılır (en kaba sınıflama).
 - **Multi-Level Coarsening + Multi-Level Refinement** (Çok seviyeli kabalaştırma + Çok seviyeli inceleştirme) – özyinelemeli bir şekilde elde edilen her bir seviyede önce kabalaştırma yapar ve sonra inceleştirme. Bu algoritma ile edilen VOS kaliteleri genelde daha yüksektir.
- **Islands** (Adalar) – Ağın düğümlerini bağlantı değerlerine (ağırlıklarına) göre yapışkan (cohesive) kümeler ayır ve sınıflandır (kümelerin içindeki ağırlıklar komşuluktaki ağırlıklardan fazla olmalıdır): bir düğümün yüksekliği (vektör) komşuluk bağlantılarının maksimum değeri olarak tanımlanır. Burada iki seçenek vardır:
- * **Line Weights** (Bağlantı Ağırlıkları)
 - * **Line Weights [Simple]** (Bağlantı Ağırlıkları [Basit])
- Eğer **Generate Network with Islands** (Adalı Ağ Oluştur) seçeneği seçildiyse sadece adaları oluşturan bağlantılara sahip yeni bir ağ elde edilir.
- **Label Propagation** (Etiket Yayılması) – Ağdaki kümelerin bulunmasında etiket yayılması algoritmasını kullan. İki versiyonu bulunmaktadır: *çok hızlı* and *standart*.
- * **Random Start** (Rastgele Başlangıç) – Düğümlere rastgele ilk etiketler vererek hızlı etiket yayılması algoritmasını uygula.
 - * **Optimize Partition** (Sınıflamayı Optimize Et) – Verilen bir kümeleme ile başla ve hızlı etiket yayılması algoritmasını kullanarak optimize et. Başlangıç sınıflaması sadece birkaç etikete (sınıfa) atanmış sadece birkaç düğümü olan neredeyse tamamen boş olabilir.
- **Blockmodeling*** (Blok Modellemesi) – Bir ve iki modlu ağların genelleştirilmiş blok modellemesi [7, 35]. Ayrıntılar için bölüm 7, sayfa 108. Model-



Şekil 6: Dünya Ticaret Ağı. Sıralamalar: alfabetik ve kümelenme tarafından belirlenen

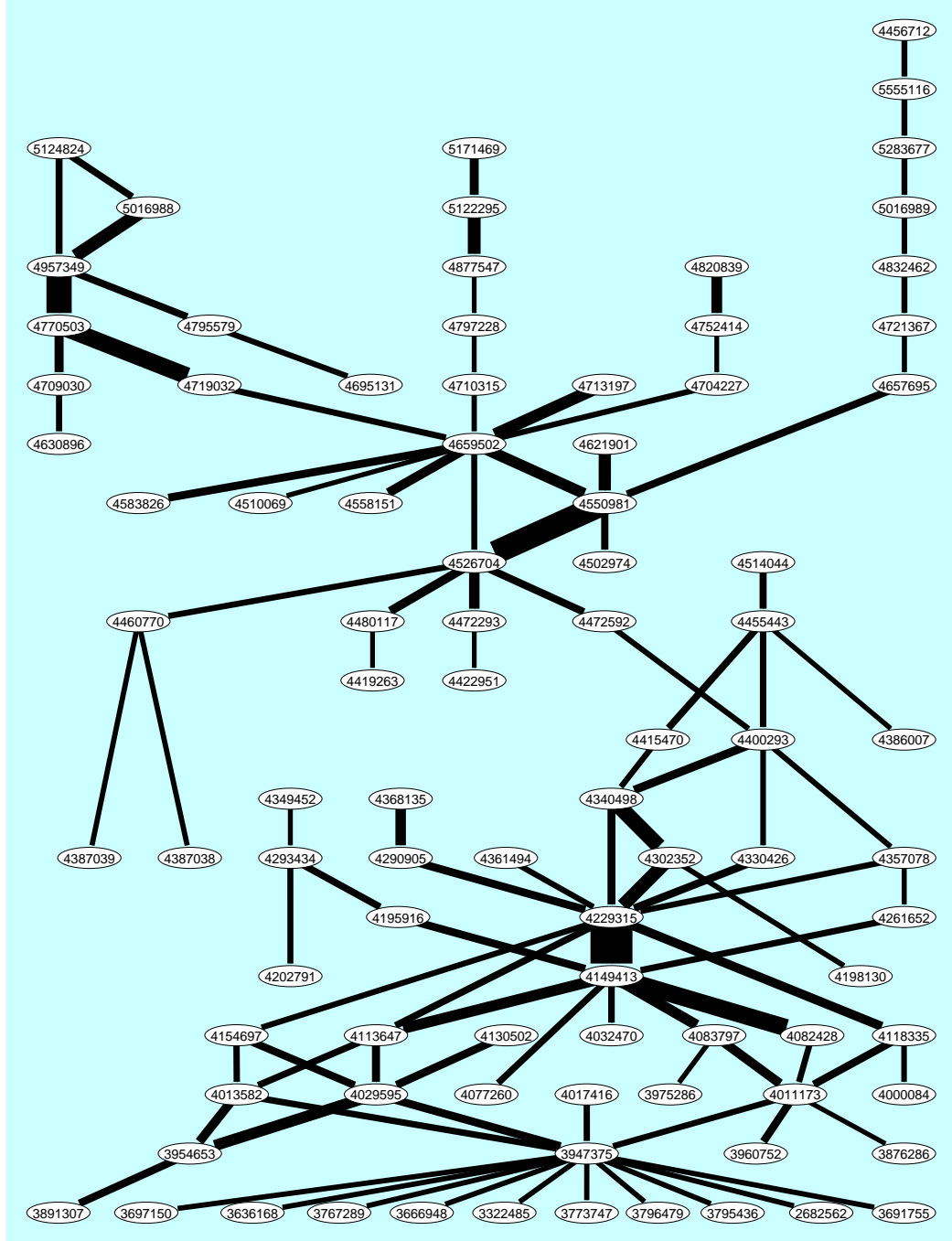
lerin tanımları MDL dosyalarında tutulur. Blok tipleri için sayfa: 69.

- * **Random Start** (Rastgele Başlangıç) – Optimizasyona rastgele sınıflama(lar)la başla.
 - * **Optimize Partition** (Sınıflamayı Optimize Et) – Seçili sınıflama için kriter fonksiyonunu göster ve optimize et.
 - * **Restricted Options** (Seçeneklerin Sınırlandırılması) – Seçeneklerin yalnızca seçilen bir kısmını göster (çoğu kullanıcı için yeterlidir) veya tüm seçenekleri göster.
 - * **Short Report** (Kısa Rapor) – Rapor penceresinde optimizasyonun yalnızca ana sonuçlarını göster (çoğu kullanıcı için yeterlidir) veya detaylı ve uzun raporu göster.
- **Vertex Labels** (Düğüm Etiketleri) – Düğümlerden aynı etikete sahip olanlar aynı sınıfta olacak şekilde bir sınıflama oluştur (molekül için uygundur).
 - **Vertex Labels matching Regular Expression** (Düzenli İfadelere Uyan Düğüm Etiketleri) – Düğüm etiketlerinin belirli bir düzenli ifadeye uyan ve uymayanlar şeklinde ikili bir sınıflama oluştur. Bazı düzenli ifade örnekleri:
 - ^D - 'D' ile başlayan etiketler;
 - a\$ - 'a' ile biten etiketler;
 - edu - 'edu' karakter dizisini içeren etiketler.

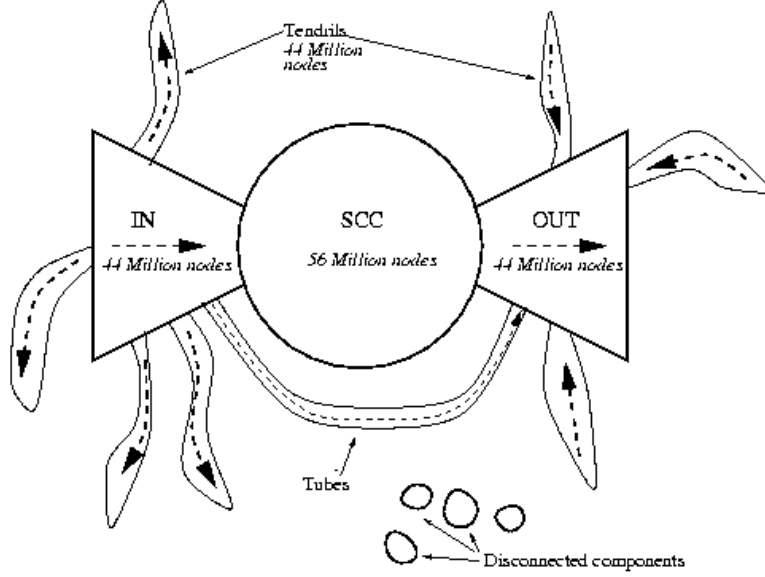
Düzenli ifadeye uyan etikete sahip düğümler 1 nolu sınıfa uymayanlar ise 0 nolu sınıfa konur

http://docwiki.embarcadero.com/RADStudio/Seattle/en/Regular_Expressions

- **Vertex Shapes** (Düğüm Şekilleri) – Düğümlerin şekillerine göre bir sınıflama oluştur yani aynı şekle sahip olanlar aynı sınıfta olacak şekilde (elips, kutu, elmas) to the same class numbers (soyağacında bu cinsiyeti gösterir).
- **Bow-Tie** (papyon) – Yönlü bir ağın (örneğin web sitelerinin ağı) düğümlerini papyon yapısına uygun şekilde sınıflama yap. Bu yapıda sınıflar (kümeler): 1 – LSCC (büyük güçlüce bağlantılı bileşen), 2 – IN (iç bileşen), 3 – OUT (dış bileşen), 4 – TUBES (tüpler), 5 – TENDRILS (filizler), 0 – OTHERS (diğerleri).
- **Default Labels Partition** (Varsayılan Etiketler Sınıflaması) – Bu işlemde girdi varsayılan düğüm etiketlerine (örneğin: v_3, v_9, \dots) sahip bir ağdır. Sonuç ise aynı büyüklükte ve düğüm etiketlerinde kodlanmış sayılarla tanımlanan (örneğin: 3, 9, ... 1 nolu sınıfa diğerleri 0'a) bir sınıflamadır. Eğer ağ çeşitli yollarla indirgenmişse bu işlem sınıflamalar, vektörler, vb. ağla uyumlu diğer bazı nesnelere üretmek için de kullanılabilir.
- **p -Cliques** (p klikler) Ağ p kliklere göre sınıflandır (sınıflamada düğümler en azından p (0-1 arasında bir sayı) kadar komşusuyla aynı sınıfa paylaşır.
 - * **Strong** (Güçlü) ... yönlü ağlar için.
 - * **Weak** (zayıf) ... yönsüz ağlar için.
- **Create Vector** (Vektör Oluştur) – Ağdan bir vektör çıkar.
 - **Centrality** (Merkezilik) – Sonuçta her bir düğüm için seçilen merkezilik ölçüsünü içeren bir vektör ve bütün ağın merkezileşme indeksi çıkar. [65, ss. 169-219].
 - * **Degree** (Derece)
 - **Input** (İç) – Düğümlere giren bağlantı sayısı.
 - **Output** (Dış) – Düğümlerden çıkan bağlantı sayısı.
 - **All** (Tüm) – Düğümlerin bağlantı sayıları.
 - * **Weighted Degree** (Ağırlıklı Derece) – Ağırlıklı derece bir düğüme giren, çıkan ve bütün bağlantılarının değerlerinin toplamıdır.
 - * **Closeness centrality** (Sabidussi yakınlık merkeziliği).
 1. **Input** (İç) – her bir düğümün diğer düğümlerden kendisine olan mesafeye göre merkeziliği



Şekil 7: ABD Patentleri Ağı – Ana ada ‘liquid-crystal display (LCD ekran)’



Şekil 8: Papyon – Web sayfalarının link ağı [26]

2. **Output** (Dış) – her bir düğümün diğer düğümlere olan mesafesine göre merkeziliği
 3. **All** (Tümü) – bağlantıların yönünü yok sayar ve böylece ağ yönsüz olarak alınıp yakınlık merkeziliği hesaplanır
- * **Betweenness** (Aradalık) merkeziliği (Freeman).
 - * **Hubs-Authorities** (Dağıtıcı ve toplayıcı düğümler) – Yönlü ağlarda genelde iki çeşit önemli düğüm belirlenebilir: *hubs* (dağıtıcı düğümler) and *authorities* (toplayıcı düğümler) [47]. Bir düğümden toplayıcılığı yüksek birçok düğüme doğru bağlantı varsa o düğüm iyi bir dağıtıcıdır. Bir düğüme dağıtıcılığı yüksek birçok düğümden bağlantı varsa o düğüm iyi bir toplayıcıdır. Sonuçta oluşan sınıflamada 1 değeri düğümün iyi bir toplayıcı, 2 değeri hem iyi bir toplayıcı hem de iyi bir dağıtıcı, 3 değeri ise iyi bir dağıtıcı olduğunu gösterir.
 - * **Proximity Prestige** (Yakınlık Prestiji) – Her bir düğümün nüfuzunu (domain) aşağıdakilere göre hesapla: *input* (iç), *output* (dış) veya *all* (tüm) komşularına göre. Sonuçta oluşan nesnelere:
 - Nüfuzun değerini (erişilebilen düğüm sayısı) içeren bir sınıflama
 - Normalize edilmiş nüfuz değerlerini içeren bir vektör – normalizasyon toplam düğüm sayısının bir eksiği ile hesaplanır.

- Nüfuzdan veya nüfuza olan ortalama mesafeleri içeren vektör.
- Yakınlık prestijini (proximity prestige) içeren bir vektör
- * **Laplace** (Laplace merkeziliği) – Laplace merkeziliği. Detaylar için:
www.scirp.org/journal/PaperDownload.aspx?paperID=27402
- * **Line Values** (Bağlantı Değerleri)
 - **Min** (Minimum) – Seçili düğümün iç, dış ve tüm bağlantılarının minimum değerini bul.
 - **Max** (Maksimum) – Seçili düğümün iç, dış ve tüm bağlantılarının maksimum değerini bul.
- * **Centers** (Merkezler) – ‘robbery’ (soygun) algoritmasına göre ağdaki merkezleri bulur. Bu algorithmada komşularından daha yüksek derece (ya da daha yüksek ağırlıklı dereceye sahip) düğümler onlardan “çalarlar”:
 - başlangıçta düğümlere derecelerine göre ağırlıklı derece değerleri verilir ya da hepsi 1 değerini alır
 - ‘zayıf’ bir düğüm bulunduğunda komşuları ondan ağırlıklı dereceleri oranında çalarlar ya da hepsi eşit miktarda çalar
- **Clustering Coefficients** (Kümelenme Katsayısı) – Yönsüz bir ağ için farklı kümelenme katsayılarını hesaplar:
deg(v) düğüm v’nin derecesi olsun, |E(G₁(v))| düğüm v’nin 1-komşuluğunda (1 mesafede) bulunan düğümler arasındaki bağlantı sayısıdır, MaxDeg ağdaki en büyük derece değeridir, ve |E(G₂(v))|, düğüm v’nin 1 ve 2-komşuluğunda bulunan tüm düğümler arasındaki bağlantı sayısıdır.
 - * CC₁ – 1-komşuluk kümelenme katsayısı:
$$CC_1(v) = \frac{2|E(G_1(v))|}{\deg(v) \cdot (\deg(v) - 1)} \quad CC'_1(v) = \frac{\deg(v)}{\text{MaxDeg}} CC_1(v)$$
 - * CC₂ – 2-komşuluk kümelenme katsayısı
$$CC_2(v) = \frac{|E(G_1(v))|}{|E(G_2(v))|} \quad CC'_2(v) = \frac{\deg(v)}{\text{MaxDeg}} CC_2(v)$$

Eğer düğüm v için deg(v) ≤ 1 ise kümelenme katsayıları eksik değer (999999998) olur. Hem Watts-Strogatz kümelenme katsayısı (geçişlilik) hem de ağ kümelenme katsayısı raporlanır.
- **Structural Holes (Yapısal Boşluklar)** – Burt tarafından ilk defa tanımlanan ve bir kısıtlılık ölçüsü olan yapısal boşlukları hesaplar [27, page 54-55]. Sonuçlar:

- * ağ p_{ij} : i 'nin j ile bağlantılarının değerinin i 'nin bütün bağlantılarının toplam değerine oranı. Burada a_{ij} i 'den j 'ye olan bağlantının değeri (ağırlığı)

$$p_{ij} = \frac{a_{ij} + a_{ji}}{\sum_k (a_{ik} + a_{ki})}$$

- * ikili kısıtlılıkları (c_{ij}) içeren ağ – j 'nin etrafındaki eksik birincil boşlukların i üzerindeki kısıtlılığı: Açıklama: Komşu j i 'nin girişimci fırsatlarını aşağıdaki şekilde kısıtlayacaktır:
 - j 'ye ulaşmak için büyük bir zaman ve enerji yatırımı yaptınız ve
 - j yatırımınıza iyi bir dönüş almak için pazarlık yapabileceğiniz birkaç yapısal boşlukla sarılmıştır.

$$c_{ij} = (p_{ij} + \sum_{k, k \neq i, k \neq j} p_{ik} p_{kj})^2$$

- * toplu kısıtlılıklar C_i : $C_i = \sum_j c_{ij}$ 'yi içeren bir vektör, $C_i = 1$ (izole düğümler için)

– **Generalized Core** (Genelleştirilmiş Çekirdek) – k genelleştirilmiş çekirdek.

- * **Degree** (derece) – sıradan çekirdek.
- * **Sum** (Toplam) – bağlantıların değerlerini hesaba katarak (pcore içindeki bağlantıların değerleri toplamı).
- * **Max** (Maksimum) – bağlantıların değerlerini hesaba katarak (pcore içindeki bağlantıların maksimum değeri)

– **Distribution of Distances*** (Mesafelerin Dağılımı*) – Ağdaki en kısa yolların uzunluğunun bir dağılımını ve ulaşılabilir bütün düğümler arasındaki ortalama yol uzunluğunu hesapla. Bütün düğümler başlangıç noktası olarak kullanılır.

– **Get Loops** (İlmekleri Al) – İlmeklerin değerlerini bir vektöre kaydet.

– **Get Coordinate** (Koordinatları Al) – ağın x, y, veya z koordinatları. Bütün koordinatlar aynı anda da alınabilir. Koordinatlar 3'den fazla olabilir ve ondalık nokta (.) içermelidir.

• **Create Permutation** (Permütasyon Oluştur) – verilen ağın permütasyonlarını hesaplar.

– **Depth First** (Derinliğine Arama) – Seçilen ağın derinliğine arama algoritması ile numaralandırılması...

- * **Strong** (Güçlü) ... yayların yönünü dikkate alarak.

- * **Weak** (Zayıf) ... yayların yönlerini dikkate almadan (veya yönsüz ağ).
- **Breadth First** (Genişliğine Arama) – Seçilen ağın genişliğine arama algoritması ile numaralandırılması...
- * **Strong** (Güçlü) ... yayların yönünü dikkate alarak.
- * **Weak** (Zayıf) ... yayların yönlerini dikkate almadan (veya yönsüz ağ).
- **Reverse Cuthill-McKee** (Ters Cuthill-McKee) – Seçilen ağın Ters Cuthill-McKee (TCM) algoritması ile numaralandırılması. [İlgili makale](#).
- **Core + Degree** (Çekirdek + Degree) – Tüm çekirdek sınıflamasına göre azalan numaralandırma. Aynı çekirdek numarasında düğümler aynı sayıda veya daha fazla çekirdek numaralı komşu sayılarına göre azalan bir sırada dizilir.
- **Create Hierarchy** (Hiyerarşi Oluştur) – Verilen ağın hiyerarşik ayrıştırılması (decompositon).
 - **Clustering*** (Kümelenme*) – Hiyerarşik kümelenme işlemi. Girdi olarak benzemezlik ağını (matrisini) kullanır. Bu matris aşağıdaki gibi elde edilir:
Operations / Cluster / Dissimilarity / Network based (Operasyonlar / Küme / Benzemezlik / Ağ tabanlı) veya girdi dosyasından okuyarak.
 - * **Run** (Çalıştır) – Hiyerarşik kümelenme işlemi. Sonuç EPS formatında iç içe kümeler ve bir dendrogramdan oluşan bir hiyerarşidir.
 - * **Options** (Seçenekler) – Hiyerarşik kümelenme işlemi yöntemini seçin: general (genel), minimum (minimum), maximum (maksimum), average (ortalama), Ward (Ward yöntemi), squared Ward (kareli Ward yöntemi).
 - **Symmetric-Acyclic** (Simetrik-Çevrimsiz) – Ağın simetrik ve çevrimsiz bir ayrıştırılması. Sonuç iç içe kümelerden oluşan bir hiyerarşidir [33].
 - **Clustering with Relational Constraint** (İlişkisel Kısıtlılıklı Kümelenme) – İlişkisel kısıtlılık işlemine göre hiyerarşik kümelenme. Bkz.: Ferligoj A., Batagelj V. (1983): Some types of clustering with relational constraints. *Psychometrika*, **48(4)**, 541-552.

Yalnızca bağlantılı düğümler arası benzemezlik hesaba katıldığı için kümelenmelerin bulunması büyük ağlarda da çok hızlıdır. Girdi benzemezlikleri içeren bir ağdır ve *Operations / Dissimilarity / Network*

(İşlemler / Benzemezlik / Ağ), *Vector based* (vektör tabanlı) ile elde edilebilir veya girdi dosyasından okunabilir.

* **Run** (Çalıştır) – Sonuçta ağacı (boğumların babaları) temsil eden bir sınıflama, düğümlerin yüksekliğini ve alt ağaçlardaki düğüm sayısını veren iki vektör elde edilir. Ağın büyüklüğü (düğüm sayısı) n ise elde edilen sınıflama ve vektörlerin boyutu $2*n-1$ olur. Bu oluşan nesnelere orijinal ağla uyumlu olmadığı unutulmamalıdır. Uyumlu nesnelere oluşturmak için *Make Partition* (sınıflama oluştur) çalıştırılmalıdır.

Options (Seçenekler) – İlişkisel kısıtlılıklı hiyerarşik kümelene için yöntemi seçin: *minimum* (minimum), *maximum* (maksimum), veya *average* (ortalama) ve strateji olarak: *strict* (sıkı), *leader* (lider), or *tolerant* (toleranslı).

* **Make Partition** (Sınıflama Oluştur) – Ağacı temsil eden sınıflamadan orijinal ağla uyumlu sınıflamayı oluştur.

- **using Threshold determined by Vector** (Vektör tarafından belirlenen eşiği kullanarak) – Bir eşik değeri vererek ağacı temsil eden sınıflamadan iki vektörden birinden (hepsinin de boyutu $2*n-1$ 'dir) orijinal ağla uyumlu sınıflamayı oluştur.
- **with selected Size of Clusters** (Kümelere seçilen boyutu ile) – Ağacı temsil eden sınıflamayı ve kümelere verilen düğüm sayısını kullanarak orijinal ağla uyumlu bir sınıflama oluştur.

* **Extract Subtree as Hierarchy** (Alt Ağacı Hiyerarşi Olarak Çıkar) – Elde edilen sınıflamadan kökü vererek bir Pa jek hiyerarşisi alt ağacı çıkar.

• **2-Mode Network** (İki Modlu Ağlar) – İki modlu ağlara özel işlemler.

– **Partition into 2 Modes** (İki moda ayır) – İki modlu bir ağın düğümlerini iki alt kümeye ayır.

– **Transpose 2-Mode** (İki Modlu Ağı Transpoze Et) – Verilen iki modlu bir ağda satır ve kolonları yer değiştir.

– **2-Mode to 1-Mode** (İki modluyu bir modluya çevir) – İki modlu bir (afiliasyon) ağından basit bir modlu bir ağ oluştur. Sonuç ağırlıklı bir ağdır. İki modlu bir ağı girdi dosyasında tutmak için **Pa jek** veya Ucinet formatı kullanabilirsiniz. Ucinet verisi için **Davis . dat**'a bakabilirsiniz.

* **Rows** (Satırlar) – Sonuç satır öğeleri (aktörler) arasındaki bağlantılardan oluşan bir ağdır. Bağlantının değeri iki aktörün ortak katıldığı etkinlik sayısını gösterir.

- * **Columns** (Kolonlar) – Sonuç kolon öğeleri (etkinlikler) arasındaki bağlantılardan oluşan bir ağdır. Bağlantının değeri iki etkinliğe de katılan aktör sayısıdır.
- * **Special** (Özel) – Çarpma yerine minimum (veya) maksimum değer kullanılır.
- * **Include Loops** (İlmekleri de İşleme Al) – Bu seçenek işaretlenirse her bir aktör için toplam etkinlik sayısını (her bir etkinlik için toplam aktör sayısını) içeren ilmekler eklenir.
- * **Multiple Lines** (Çoklu Bağlantılar) – Düğümler arasında çoklu bağlantıların olabileceği ağırlıksız bir modlu ağ oluşturur. Üretilen bağlantının etiketi oluşturulduğu aktivite/aktör etiketine karşılık gelir. Eşit boyutta bir sınıflama varsa çoklu ilişkili bir ağ oluşturulabilir.
- * **Normalize 1-Mode** (Bir modlu ağı normalize et) – Elde edilen bir modlu ağı normalize et. Bir modlu a **include loops** (ilmekleri de işleme al) seçeneğinin seçilmesi, **multiple lines** (çoklu bağlantılar) seçeneğinin ise seçilmemesi gerekiyor:

$$\text{Geo}_{ij} = \frac{a_{ij}}{\sqrt{a_{ii}a_{jj}}}$$

$$\text{Input}_{ij} = \frac{a_{ij}}{a_{jj}}$$

$$\text{Output}_{ij} = \frac{a_{ij}}{a_{ii}}$$

$$\text{Min}_{ij} = \frac{a_{ij}}{\min(a_{ii}, a_{jj})}$$

$$\text{Max}_{ij} = \frac{a_{ij}}{\max(a_{ii}, a_{jj})}$$

$$\text{MinDir}_{ij} = \begin{cases} \frac{a_{ij}}{a_{ii}} & a_{ii} \leq a_{jj} \\ 0 & \text{otherwise} \end{cases}$$

$$\text{MaxDir}_{ij} = \begin{cases} \frac{a_{ij}}{a_{jj}} & a_{ii} \leq a_{jj} \\ 0 & \text{otherwise} \end{cases}$$

Elde edilen ağ genelde seyrek bir ağ değildir. Daha seyrek bir hale getirmek için **Network / Transform / Remove / lines with value / lower than** (Ağ / Transforme et / Sil / belli bir değerden düşük olan bağlantıları) komutu kullanılarak değeri düşük bazı bağlantılar silinebilir.

- * **Rows=Cols** (Satırlar=Kolonlar) – Aynı düğüm kümesine sahip iki

modlu bir ağı bir modlu ağa çevir.

- * **Cols=0** (Kolonlar=0) – Kolon sayısını sıfır alarak iki modlu bir ağı bir modlu ağa çevir. Sonuçta yapılan iş ağ dosyasındaki *Vertices 32 18 komutunu *Vertices 32 komutuna çevirmektir.

– **Core** (Çekirdek) – İki modlu bir ağın çekirdeğini bul.

- * **2-Mode** (İki modlu)– İki modlu bir ağın çekirdek sınıflaması. İlk (k_1) ve ikinci (k_2) alt kümede verilen minimum derece ile yeni bir sınıflama oluşturulur. Bu sınıflamada 0 değeri düğümün k_1 ve k_2 ile belirlenen çekirdeğe ait olmadığını, 1 değeri ise düğümün çekirdeğe ait olduğunu gösterir.

- * **2-Mode Review** (İki modlu gözden geçirme) – k_1 ve k_2 'nin verilen başlangıç değerleriyle aşağıdaki liste oluşturur:

k_1 k_2 Satırlar Kolonlar bileşen

burada k_1 birinci alt kümedeki minimum derece, k_2 ikinci alt kümedeki minimum derece, *Satırlar* ve *Kolonlar* sırasıyla birinci ve ikinci alt kümedeki düğüm sayısı ve *bileşen* ise ağdaki k_1 ve k_2 ile indüklenmiş bağlantılı bileşen sayısıdır. k_1 ve k_2 sonuçta oluşan ağ boş oluncaya kadar artırılır.

- * **2-Mode Border** (İki modlu sınır) – Verilen bir iki modlu ağ için k_1 ve k_2 'nin yalnızca sınır değerlerini bul.

- * **Valued 2-Mode Core** (Ağırlıklı 2 modlu çekirdek) – İki modlu ağın bağlantı değerlerine göre ağırlıklı çekirdek sınıflaması. İlk (k_1) ve ikinci (k_2) alt kümede verilen minimum derece ile yeni bir sınıflama oluşturulur. Bu sınıflamada 0 değeri düğümün k_1 ve k_2 ile belirlenen çekirdeğe ait olmadığını, 1 değeri ise düğümün çekirdeğe ait olduğunu gösterir.

– **Important Vertices** (Önemli Düğümler) – Dağıtıcı ve toplayıcı düğümler algoritmasının iki modlu ağlar için genelleştirilmiş hali – bir ve ikinci alt kümedeki önemli düğümleri bul.

– **Info** (Bilgi) – İki modlu ağ hakkındaki temel bilgiler, örneğin iki modlu ağ yoğunluğu.

- **Multiple Relations Network** (Çoklu İlişkili Ağ) – Çoklu ilişkili ağlara özel işlemler. Bu ağlarda bağlantıların ilişkisi denilen farklı türleri vardır.

– **Select Relation(s) into One Network** (Seçili İlişkileri Bir Ağ Dönüştür)

– Verilen bir ağın sadece seçili ilişkilerini tut ve diğer bütün ilişkileri sil. Sonuç seçili ilişkilere sahip bir ağdır.

- **Extract Relation(s) into Separate Network(s)** (İlişkileri Farklı Ağlara Ayır) – Çoklu ilişkili ağda bir veya daha fazla ilişkiyi seç ve her bir seçilen ilişki için yeni bir ağ oluştur.
- **Canonical Numbering** (Kanonik Numaralandırma) – İlişkileri sırasıyla 1, 2,... şeklinde numaralandır.
- **Generate 3-Mode Network** (3 modlu Ağ Oluştur) – bir modlu bir ağdan veya iki modlu çoklu ilişkili bir ağdan 3 modlu bir ağ oluştur. Çoklu ilişkili ağdaki her bir bağlantı için $r: i j v$ (i 'den j 'ye değeri v ve ilişki numarası r olan bağlantı) aşağıdaki üç satırı oluşturur (üçgen):
 - * bir modlu ağlar:
 - $i \ N+j \ v$
 - $i \ 2N+r \ v$
 - $N+j \ 2N+r \ v$
 - * iki modlu ağlar:
 - $i \ j \ v$
 - $i \ N+M+r \ v$
 - $j \ N+M+r \ v$

burada N birinci moddaki, M ise ikinci moddaki düğüm sayısıdır.
- **Line Values – > Relation Numbers** (Bağlantı Değerleri – > İlişki Numaraları) – Bağlantı değerlerini ilişki numarası olarak kaydet (ondalık kısımları atılmış ve mutlak değeri alınmış olarak).
- **Relation Numbers – > Line Values** (İlişki Numaraları – > Bağlantı Değerleri) – İlişki numaralarını bağlantı değeri olarak kaydet.
- **Change Relation Number - Label** (İlişki Numarasını Değiştir - Etiket) – Seçili ilişkinin numarasını ilgili etiketli yeni ilişki numarasına çevir.
- **Info** (Bilgi) – Çoklu ilişkili ağ konusunda bilgi
 - * ilişkilerin sayısı
 - * yay, kenar ve toplam bağlantı sayıları
- **Acyclic Network** (Çevrimsiz Ağ) – Çevrimsiz ağlara özel işlemler.
 - **Create Partition** (Sınıflama Oluştur) – Çevrimsiz ağdan bir sınıflama oluştur.
 - * **Depth Partition** (Derinlik Sınıflaması)
 - **Acyclic** (Çevrimsiz) – Ağda düğümlerin derinliğine göre bir sınıflama oluştur

- **Genealogical** (Soyağacı) – Düğümlerin tabakalarına göre soyağacı temsil eden ağdan bir sınıflama oluştur.
 - **Generational** (Kuşaksal) – Düğümlerin tabakalarına göre soyağacı temsil eden ağdan bir sınıflama oluştur. Soyağacı sınıflamasına benzer fakat daha az tabakası vardır.
 - * **Select First Vertices** (İlk Düğümleri Seç) – Ağın ilk düğümlerini seç ve onlara farklı küme numaraları ver (1'den başlayarak) ve bir sınıflamaya koy. Bu sınıflama daha sonra ağın genetik yapısını (Genetic Structure) oluşturmak için girdi olarak kullanılabilir.
 - * **Select Last Vertices** (Son Düğümleri Seç) – Ağın son düğümlerini seç ve onlara farklı küme numaraları ver (1'den başlayarak) ve bir sınıflamaya koy.
- **Create Weighted Network + Vector** (Ağırlıklı bir ağ ve vektör oluştur)
– Bir çevrimsiz ağdaki düğüm (makaleler) ve yayların (atıflar) ağırlıklarını (önemini) hesapla.
- * **Traversal Weights** (Gezinme Ağırlıkları) – Eğer ağ bir atıf ağıysa yayların (atıfların) ve düğümlerin (makaleler) gezinme ağırlıkları hesaplanabilir. Sonuçlar:
 - Yaylarının ağırlığı atıfların önemini temsil eden bir ağ
 - Farklı iç yolların sayısını veren bir vektör
 - Farklı dış yolların sayısını veren bir vektör
 - Gezinme ağırlıklarını içeren bir vektör (düğümlerin ve makalelerin önemleri)
- Gezinme ağırlıklarını belirleyen farklı yöntemler [43]:
- **Search Path Count (SPC)** (Arama Yolu Sayısı) – Kaynak düğümden çıkış düğümüne hesaplanır.
 - **Search Path Link Count (SPLC)** (Arama Yolu Link Sayısı) – Her bir düğüm bir kaynak düğüm olarak alınır.
 - **Search Path Node Pair (SPNP)** (Arama Yolu Düğüm Çifti).
- Büyük ağlarda gezinme ağırlıkları çok büyük birer tamsayı olabilirler. Bu durumda ağırlıklar akışı veya maksimum değer kullanılarak normalize edilebilir ya da kaydedilebilir.
- * **Probabilistic Flow** (Olasılıklı Akış) – Sonuçta bir vektör ve yayları ağırlıklı yeni bir ağ ortaya çıkar. Vektör değerleri kaynak düğümlerden başlayan rassal bir yürüyüşün seçilen düğümlerden geçme olasılığını belirtir. Yayların değeri ise kaynak düğümlerden başlayan rassal bir yürüyüşün seçilen yaylardan geçme olasılığını verir.

– **Create (Sub)Network** ((Alt)ağ Oluştur) – Ağırlıklı çevrimsiz bir ağdan önemli yolları çeker.

* **Main paths** (Ana yollar) – Gezinme ağırlıkları olan bir atıf ağında (veya yayları üzerinde ağırlık olan herhangi bir çevrimsiz ağda) ana yolları arar. Ayrıntılar için [49].

· **Local Search** (Lokal arama) – aramanın her bir adımında mevcut yaya bağlı en yüksek ağırlıklı yay(ları)ı seçer. Bir tolerans parametresi kullanılabilir (0-1 arasında). Toleransın 0 olması her bir adımda yalnızca en yüksek ağırlıklı yayların hesaba katılmasını sağlar. Daha yüksek tolerans değerlerinde ise daha küçük ağırlıklı yaylar da alınır.

1. **Forward** (İleri) – Kaynak düğümden çıkış düğümüne ana yolları ara.

2. **Backward** (Geri)– Aksi yönde ana yolları ara (çıkış düğümünden kaynak düğüme).

3. **Key-Route** (Anahtar Rota)– Anahtar rota olarak adlandırılan bazı en önemli yaylar seçilir ve anahtar rotaların uçları arasında ana yolları arar

4. **Through Vertices in Cluster** (Kümedeki Düğümler Üzerinden) – Seçili kümedeki düğümlerden geçen lokal ana yolları arar.

· **Global Search** (Global Arama) – en yüksek genel ağırlık toplamı ana yolları ara.

1. **Standard** (Standart) – Kaynak düğümden çıkış düğümüne en yüksek genel ağırlık toplamı ana yolları ara. Bu yöntem Critical Path Method (CPM, kritik yol yöntemi) olarak da bilinir.

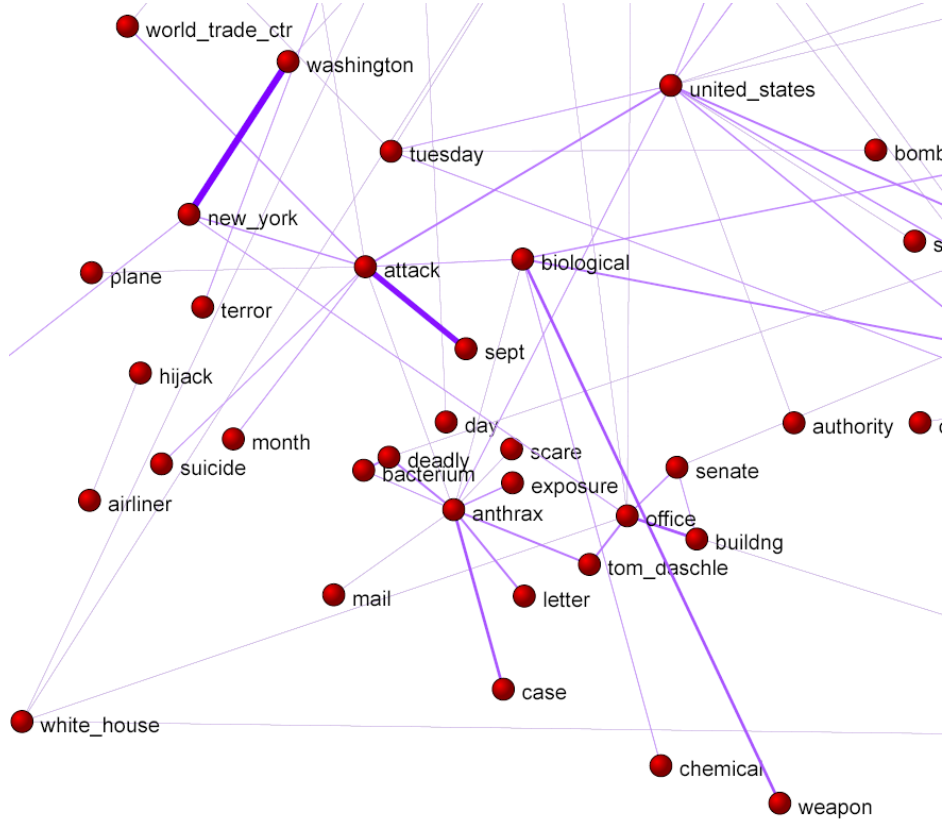
2. **Key-Route** (Anahtar Rota) – Seçilen anahtar rotaları içeren en yüksek genel ağırlık toplamı tüm ana yolları bul.

3. **Through Vertices in Cluster** (Kümedeki Düğümler Üzerinden) – Seçili kümedeki düğümlerden geçen en yüksek genel ağırlık toplamı ana yolları ara.

· **Mark Main Paths as Multirelational Network** (Ana Yolları Çoklu İlişkili Ağı Olarak İşaretle) – Gezinme ağırlıklı orijinal ağı bir çoklu ilişkili ağa çevir: ana yola ait olan yayların ilişki numarası 2 olur, diğer yayların ilişki numarası 1 olur.

Ayrıca küme 1'deki bütün ana yollarının üzerinde bulunan düğümler ve küme 0'da bulunan bütün düğümleri içeren bir ikili sınıflama oluşur.

- * **Critical Path Method (CPM)** (Kritik Yol Yöntemi) – Çevrimsiz ağdaki kritik yolu bul – Sonuçta kritik yolu içeren yeni bir ağ oluşturulur. Bu algoritma proje planlaması alanında da kullanılabilir gibi çevrimsiz ağların analizinde de kullanılır – Ayrıca aktivitele-
rin toplam ve serbest gecikme zamanlarını içeren ağlar da oluşturulur. Bu iki ağ çoklu ilişkilidir: kritik yola ait olan yayların ilişki numarası 2 olur. İki adet de vektör oluşturulur: Birincisi verilen durumlara gelmenin en erken mümkün zamanını ve ikincisi ise verilen durumlara gelmenin en geç uygulanabilir zamanını içerir.
- **Create Permutation** (Permütasyon Oluştur) – (Yaklaşık) çevrimsiz bir ağın permütasyonunu oluştur.
 - * **Topological Ordering** (Topolojik Sıralama) – Çevrimsiz ağın topolojik sıralaması.
 - * **Becker’s Heuristic** (Becker Keşifseli) – Becker’in yaklaşık çevrimsiz bir ağın doğrusal sıralamasını oluşturmak için geliştirdiği yöntem. Çevrimsiz ağların topolojik sıralamasına benzer fakat çevrimsiz olmayan ağlara da uygulanabilir. Küçük güçlü bileşenlere sahip büyük yönlü ağlarda kullanılması gerekmektedir.
- **Transform** (Transforme et) – Yaklaşık çevrimsiz ağı transforme et.
 - * **Add Source and Sink** (Kaynak ve Çıkış Düğümü Ekle) – Çevrimsiz ağa iki ekstra düğüm ekle: birinci ve sonuncu düğüm (yeni ağ iki ekstra yapay düğüme sahip olur).
 - * **Preprint Transformation** (Önbaskı Transformasyonu) – ‘Yaklaşık’ çevrimsiz ağı aşağıdaki şekilde ‘gerçek’ bir çevrimsiz ağa dönüştür: her bir güçlü bileşene ait her makale u kopyalanır ve bu makalenin ‘önbaskı’ versiyonu u' ağa eklenir. Güçlü bileşende bulunan makaleler önbaskı makalelere atıfta bulunur, önbaskı olanlarsa dışarıdaki makalelere. Her bir güçlü bileşen karşılık gelen tam ve yönlü bir iki modlu alt ağ (makaleleri \times önbaskılar) ile değiştirilir.
- **+ Partition** (+ Sınıflama) – Bu işlemler için bir çevrimsiz ağ ve bir sınıflamaya ihtiyaç vardır.
 - * **Create Vectors with Genetic Structure** (Genetik Yapılı Vektörler Oluştur) – Verilen çevrimsiz ağın verilen (birinci düğümlerden oluşan) sınıflamaya göre genetik yapısını hesapla. Sınıflama **Network / Acyclic Network / Create Partition / Select First Vertices** (Ağ / Çevrimsiz Ağ / Sınıflama Oluştur / Birinci Düğümleri Seç) komutu ile elde edilebilir. Sonuç olarak sınıflamadaki her bir küme numarası için bir vektör elde ederiz. Dikkat edilmelidir



Şekil 9: Reuters terör haberleri ağının bir parçası (36. gün)

ki küme sayısı yüksekse hesaplanması çok uzun sürebilir. Ayrıca baskın gen sınıflaması ve baskın gen vektörü de elde edilir.

- **Info** (Bilgi) – İlk ve son düğümlerin sayısı, minimum ve maksimum derece, bileşen sayısı, en büyük bileşenin büyüklüğü ve p çizgesi, varsa ayrıca tekrar bağlama indeksi (Relinking Index) rapor edilir.

- **Temporal Network** (Zamansal Ağ) – Zamansal ağlara özel işlemler.

- **Generate in Time** (Zamanda Oluştur) – Belirlenmiş zaman(lar) veya aralıklarda ağ oluştur. İlk zaman, son zaman ve basamak tam sayı olarak girilir.

Bu işlemi gerçekleştirmek için düğüm ve bağlantıların ne zaman aktif olduğu konusunda ekstra parametreler verilmelidir. Bu parametreler [ve] işaretleri arasında verilmelidir.

- işareti aralığın alt ve üst değerlerini ayırmak için kullanılır,
- , işareti aralıkları ayırmak için kullanılır,

* işareti ise sonsuz demektir. Örneğin:

```
*Vertices 3
1 "a" [5-10, 12-14]
2 "b" [1-3, 7]
3 "e" [4-*]
*Edges
1 2 1 [7]
1 3 1 [6-8]
```

'a' düğümü zaman noktası 5'den 10'a ve 12'den 14'e kadar aktiftir. 'b' düğümü ise 1'den 3'e kadar ve 7'de aktiftir. 'e' düğümü zaman 4'den itibaren aktiftir. Düğüm 1 ('a') ve 2 ('b') arasındaki bağlantı yalnızca zaman noktası 7'de ve düğüm 1 ve 3 arasındaki bağlantı ise 6 ve 8 arasında aktiftir.

Zamansal bir ağdaki bağlantı ve düğümler *tutarlılık* şartını sağlamalıdır: bir bağlantı zaman noktası t 'de aktifse bağlı olduğu düğümler de t 'de aktif olmalıdır. Verilen bir zamansal ağda zaman dilimleri oluştururken yalnızca 'tutarlı' bağlantılar oluşturulur.

Zaman kayıtları her zaman düğüm ve bağlantıların tanımlandığı satırın sonunda yazılmalıdır.

Bir diğer zamansal ağ oluşturma imkanı da zamansal ağın **time events** (zamansal olaylar) kullanarak oluşturulmasıdır.

- * **All** (Tümü) – Büyün ağları belirlenmiş zamanlarda oluştur.
- * **Only Different** (Yalnızca Farklı Olanlar) – Yalnızca yeni ağ oluşturulan sonuncu ağdan en az bir düğüm veya bir bağlantı açısından farklıysa belirlenmiş zaman noktasında ağ oluştur.
- * **Interval** (Aralık) – Seçilen aralıkta olan düğüm ve bağlantılarla bir ağ oluştur.

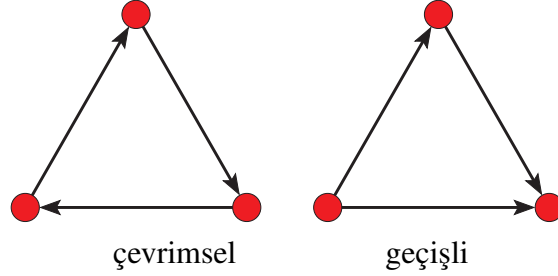
– **Info** (Bilgi) – Zamansal bir ağda her bir zaman noktasındaki düğüm ve bağlantı sayısını raporla

- **Signed Network** (İşaretli Ağ) – İşaretli ağlara özel işlemler. Bu ağlarda pozitif ve negatif değerli bağlantılar bulunur. Örneğin bir işaretli ağda pozitif bağlantılar dostluğu negatif bağlantılar düşmanlığı gösterebilir.

– **Set All Positive Line Values to 1, Negative to -1** – Bütün pozitif bağlantıları 1, negatif bağlantıları -1 yap.

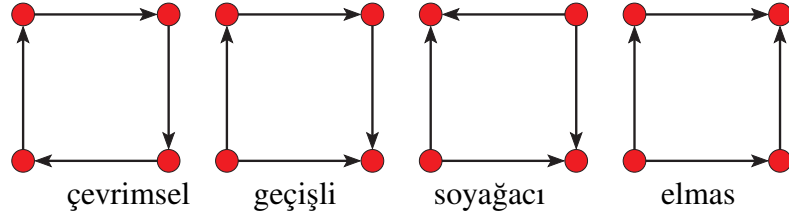
– **Create Partition** (Sınıflama Oluştur) – Ağdan bir sınıflama oluştur (ve ilk rassal sınıflama).

- * **Doreian-Mrvar method*** (Doreian-Mrvar yöntemi*) – İşaretli ağları sınıflamak için yerdeğiştirme algoritması. Verilen bir sınıflama mümkün olduğu kadar pozitif bağlantıları sınıf içinde tutarak ve yine mümkün olduğu kadar negatif bağlantıları da sınıflar arasında tutarak optimize edilir. Bir başka algoritma köşegensel ve köşegensel olmayan blokları birbirinden ayırmaz: her bir blok pozitif, negatif veya sıfır olabilir. Tekrar sayısı 1’den fazlaysa verilen sınıf numaralı ilk sınıflamalar her bir tekrarda ayrı ayrı rastgele seçilir. Program birçok optimum çözümler bulursa hepsi de rapor edilir. Algoritma hakkında daha fazla bilgi için bkz. Doreian ve Mrvar [32]. Bu seçenek iki modlu işaretli ağlar için de kullanılabilir: girdi olarak iki modlu sınıflama kullanılır. Bu durumda algoritma mümkün olduğu kadar belirgin pozitif, negatif ve sıfır bloklar bulmaya çalışır. Eğer *Prespecification* (ön belirleme) seçeneği seçilmişse kullanıcı önceden belirlenmiş bir model kullanılabilir. Bunun için hücelere **P**, **N** veya **0** girilir (sırasıyla pozitif, negatif ve sınıf bloklar için) veya hücreler boş bırakılabilir. Bu durumda blok herhangi bir tipte olabilir ve her bir düğüm için seçilen küme önceden belirlenmiş veya yasaklanmış olabilir. Her bir kümenin içermesi gereken minimum ve maksimum düğüm sayısı da önceden belirlenebilir. Küçük sıfır bloklar için ceza puanını sıfır olmayan bir değere eşitleyerek sıfır blokları mümkün olduğu kadar büyük yapmaya çalışırız.
- * **Zeleny Approach*** (Zeleny Yaklaşımı*) – Morale skorlarını optimize ederek işaretli ağları sınıflama yöntemi [69].
- * **Louvain method** (Louvain Yöntemi) – Standart Louvain yönteminin işaretli ağlar için geliştirilmiş hali. Ayrıntılar için bkz. **Louvain yöntemi**.
- * **Label Propagation** (Etiket Yayılması) – Label propagation (etiket yayılması) algoritmasının işaretli ağlarda kümeleri aramak için geliştirilmiş hali. İki versiyonu bulunmaktadır: *very fast* (çok hızlı) and *standard* (standart).
 - **Random Start** (Rastgele Başlangıç) – Düğümlere ilk etiketlerini rastgele vererek (hızlı) etiket yayılması algoritmasının çalıştırılması
 - **Optimize Partition** (Sınıflamayı Optimize Et) – Verilen bir sınıflama ile başla ve onu (hızlı) etiket yayılması algoritması ile optimize et. İlk sınıflama belirli etiketlere (kümelere) atanmış sadece birkaç düğümlü ve bu nedenle neredeyse boş kabul edilen bir sınıflama da olabilir.



Şekil 10: Çevrimsel (cyclic) ve geçişli (transitive) üçlü döngülerde bağlantılar

- **Info** (Bilgi) – Verilen işaretli ağın sınıflamaya göre Yapısal Denge Uyumsuzluğu (Structural Balance Inconsistency) ve İşaretli Modülaritesini hesapla. Ayrıca işaretli E-I indeksi (pozitif ve negatif) ve Morale indeksi de [69] hesaplanır.
- **Info** (Bilgi) – Ağ hakkında bilgi
 - **General** (Genel) – Ağ hakkındaki genel bilgiler
 - * düğüm sayısı
 - * yay, kenar ve ilmek sayısı
 - * bağlantı yoğunluğu
 - * ortalama derece
 - * En önemli (önemsiz) bağlantıları bulmak için bağlantıları değerlerine göre büyükten küçüğe (küçükten büyüğe) sırala
 - **Line Values** (Bağlantı Değerleri) – Bağlantı değerlerinin frekans dağılımı.
 - **Triadic Census** (Üçlü Sayım) – Ağdaki farklı üçlülerin sayısı. Bkz. Faust ve Wasserman [65] kitabı ve sayfa 74 Şekil 15.
 - **Degree Assortativity** (Derece Sınıflandırması) – Verilen bir ağın derece sınıflandırması.
 - **Vertex Label -> Vertex Number** (Düğüm Etiketi -> Düğüm Numarası) – Etiketinin bir kısmını vererek düğüm numarasını bulmak veya verilen düğüm numarası ile düğümün etiketini bul.
 - **Line -> Rank of its Value** (Bağlantı -> Değerinin Sırası) – Seçilen bağlantının değerine göre sırasını bul.
 - **Indices** (İndeksler) – Ağın farklı indisleri (çoğunlukla kimyasal).



Şekil 11: Yayılar üzerindeki yönlü dörtlü döngü çeşitleri

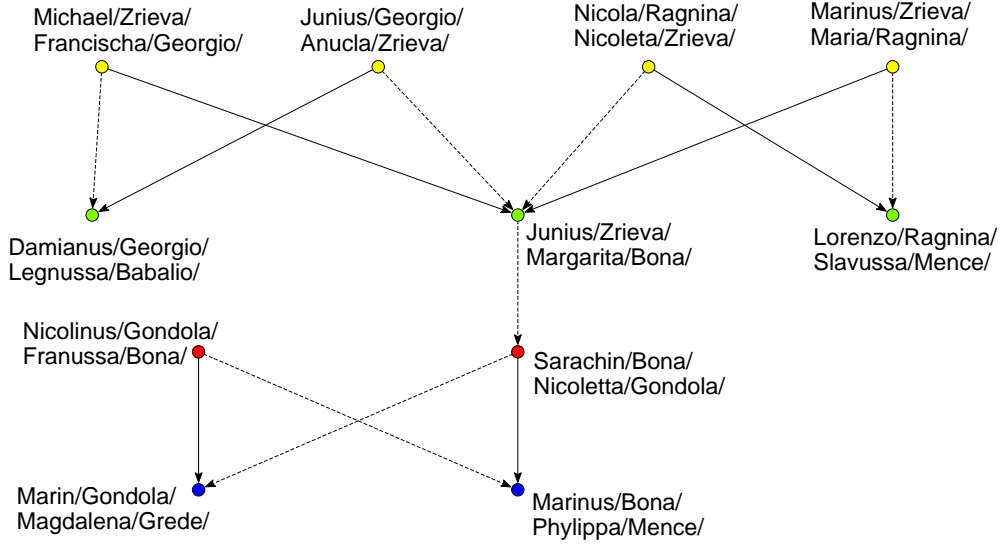
3.3 Networks (Ağlar)

İki ağ üzerinde işlemler.

- **Union of Lines** (Bağlantıların Birleşimi) – Seçili ağları kaynaştır. Sonuç çoklu ilişkili bir ağdır. Ağların birleşimini elde etmek istiyorsanız yine de çoklu bağlantılar silinmelidir. Ağlar boyut olarak aynı olmalıdır veya bir ağ m değeri n düğüme sahipse ve $m < n$ ise bu durumda n düğüme sahip ağda ilk m düğüm m düğümlü ağdaki düğümlerle eşleşmelidir.
- **Union of Vertices** (Düğümün Birleşimi) – İkinci ağı birincinin sonuna ekle.
- **Cross-Intersection** (Kesişim) – Seçilen ağların kesişimi. Ağlar boyut olarak aynı olmalıdır veya bir ağ m değeri n düğüme sahipse ve $m < n$ ise bu durumda n düğüme sahip ağda ilk m düğüm m düğümlü ağdaki düğümlerle eşleşmelidir. Kesişimdeki bağlantıların değeri karşılık gelen bağlantıların *sum* (toplamı), *difference* (farkı), *product* (çarpımı), *quotient* (kalanı), *min* (minimum)u, veya *max* (maksimumu) olabilir. Ayrıca değer olarak birinci ağdaki (*the first*) veya ikinci ağdaki (*the second*) değeri kullanabiliriz.
- **Intersection of Multiple Relations Networks** (Çoklu İlişkili Ağların Kesişimi) – İlişki numaraları dikkate alınarak seçilen ağların kesişimi bulunur.
- **Cross-Difference** (Fark) – Seçilen ağların farkı bulunur.
- **Difference of Multiple Relations Networks** (Çoklu İlişkili Ağların Farkı) – İlişki numaraları dikkate alınarak seçilen ağların farkı bulunur.
- **Multiply Networks** (Ağları Çarp) – Seçilen bir modlu veya iki modlu ağların çarpımını bulur (ağlar çarpma kriterlerine uymalıdır). Ağların birisi bir modlu

diğeri iki modlu ise çarpma yapmadan önce **Network / Create New Network / 1-Mode to 2-Mode** (Ağ / Yeni Ağ Oluştur / Bir modludan iki modluya) yoluyla bir modlu olan iki modluya çevrilmelidir. Yoksa ağlar uyumlu değildir (non-compatible) uyarısıyla karşılaşılır.

- **Multiply Networks - Get Loops only** (Ağları Çarp - Sadece İlmekleri Al) – Seçilen bir modlu ağların (ya da iki modlu da olabilir fakat bu durumda sonuç bir modlu ağ olur) çarpımıyla elde edilen matrisin sadece köşegenine döner.
- **Fragment (First in Second)** (Fragman (Birinci İkincide)) – Birinci ağla tanımlanan bütün fragmanları ikinci ağda bulur.
 - **Induced** (İndirgenmiş) – eşleşen fragman örneğindeki düğümler arasında hiç bir ekstra bağlantı olmamalıdır (güçlü koşul), yoksa ekstra bağlantılar olabilir (zayıf koşul).
 - **Negative lines inside fragment not allowed** (Fragman İçerisinde negatif bağlantılar olmamalıdır) – eşleşen fragman örneğinde düğümler arasında negatif değerli bağlantı olmamalıdır. İki düğüm arasındaki negatif değerli bir bağlantı o düğümler aynı fragmana ait olamaz demektir. Bu seçenek yalnızca *Induced* (indirgenmiş) seçeneği işaretli değilse görünür olur.
 - **Labeled** (Etiketli) – etiketler eşleşmelidir (örneğin bir moleküldeki atomlar). Etiketler sınıflamadaki sınıflar tarafından belirlenir (renk olarak) – etiketli fragmanlar aranmadan önce birinci ve ikinci sınıflama seçilmelidir. Birinci sınıflama birinci ağın (fragman) ‘etiketlerini’, ikinci sınıflama ikinci ağın (orijinal) ‘etiketlerini’ belirler.
 - **Check values of lines** (Bağlantı değerlerini kontrol et) – bağlantıların değerleri aynı olmalı (örneğin soyağacında değerler cinsiyeti temsil eder: 1 – erkek, 2 – kadın).
 - **Check relation number** (İlişki numarasını control et) – ilişki numaraları aynı olmalı.
 - **Check only cluster** (Sadece kümeyi control et) – yalnızca ilk düğümün kümedeki düğümlerden biri olduğu fragmanlar aranır.
 - **Extract subnetwork (Alt ağ çıkar)** – ekstradan bir sonuç üretir: fragmanlara ve karşılık gelen bağlantılara ait olan düğümleri içeren alt ağ çıkar.
 - * **Retain all vertices after extracting** (Çıkarımdan Sonra Bütün Düğümleri Tut) – Çıkarılmış ağda orijinal ağdaki düğümler olmalı, yalnızca hiç bir fragmana ait olmayan bağlantılar silinir.



Şekil 12: Fragmanlar – Ragusa’da akrabalar arası evlilikler

- **Same vertices determine one fragment at most** (Aynı Düğümler En Fazla bir Fragmanı Belirler) – aynı düğüm grubunda olan fragmanlar için ne yapılmalı: *if not checked* (işaretili değilse) – aynı düğümlere sahip fragmanlara izin verilir.
 - * **Create Hierarchy with fragments** (Fragmanlı Hiyerarşi Oluştur) – fragman aramasının sonucu aynı zamanda fragmanlardaki düğümlerin bir hiyerarşisidir (yalnızca **Same vertices determine one fragment at most** (Aynı Düğümler En Fazla bir Fragmanı Belirler) seçeneği işaretili değilse yapılabilir)
- **Repeating vertices in fragment allowed** (İzin Verilen Fragmanda Tekrar Eden Düğümler) – fragmanda aynı düğümler birden fazla görünebilir (örneğin döngülerde).
 - işaretili değilse*: bulunan fragmanlar her zaman orijinal fragmanla aynı sayıda düğüme sahiptir.
 - işaretliyse*: bulunan bazı fragmanlar orijinal fragmandan daha az düğüme sahip olabilir.
- **Match Vertex Labels** (Düğüm Etiketlerini Eşleştir) – Verilen iki ağ için iki sınıflama oluştur: birinci ağdaki düğümlerin etiketlerinin pozisyonu ikinci ağda ve ikinci ağdaki düğümlerin etiketlerinin birinci ağda (0 karşılık gelen etiketlerin diğer ağda var olmadığını gösterir).
Bu işlem bazı (hepsi değil) düğüm etiketlerinin aynı olduğu iki ağı birleştirmek

için faydalıdır.

Sınıflamalar ve vektörler bu işlemi kullanarak elde edilen çıkarılmış alt ağlarla uyumlu yapılabilir.

- **Shrink coordinates (First to Second)** (Koordinatları Küçült (Birinciden İkinciye)) – Bu işlemle bir ağ küçülttüğünüzde küçültülmüş ağ ayrı çizilir ve orijinal ağdaki düğümlere bütün koordinatları uygulayabilirsiniz (aynı sınıftaki düğümler aynı koordinatlarda olurlar). İkinci ağdaki koordinatları küçültülmüş birinci ağın koordinatları ile değiştirin. Küçültme aşağıdakilerden biriyle belirlenebilir:
 - **Sınıflama** veya
 - **Hiyerarşi**

3.4 VertexID (DüğümKimlik)

Girdi olarak düğüm kimliklerinin (DüğümKimlik) bir listesini alan işlemler

- **Default VertexIDs (Varsayılan DüğümKimlikler)** – İstenen büyüklükte ve ilk değerlere sahip varsayılan kimlikleri oluştur
- **Copy VertexIDs to Partition (DüğümKimlikleri Sınıflamaya Kopyala)** – Kimlikleri bir sınıflamaya kopyala.
- **Expand VertexIDs to Partition (DüğümKimlikleri bir Sınıflamaya Genişlet)** – Seçili boyutta bir sınıflama oluşturulur. Bu sınıflamada DüğümKimlik'ten kimliklerle tanımlanan düğümlerin sınıfı 1 diğerlerinin sınıfı 0 olur.
- **Info (Bilgi)** – Kimlikler hakkında genel bilgiler: boyutu, en büyük ve en küçük değerleri.

3.5 Operations (İşlemler)

Girdi olarak bir ağa veya başka bir şeye ihtiyaç olan işlemler

- **Network + VertexID (Ağ + DüğümKimlik)** – Ağ ve DüğümKimlik üzerindeki işlemler
 - **Shrink Network (Ağı Küçült)** – Seçili DüğümKimlik'e göre ağ küçült.

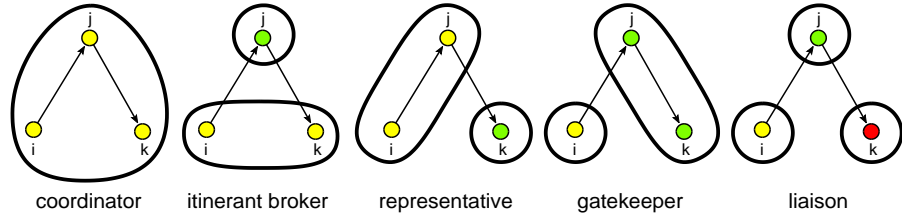
- **Network + Partition (Ağ + Sınıflama)** – Bir ağ ve bir sınıflama üzerinde yapılabilen işlemler. Count Neighbor Clusters (Komşu sınıfların sayılması) durumunda sınıflama yerine sınıf, coloring (renklendirme) durumunda ise sınıflama yerine permütasyon kullanılır).

– **Extract (Çıkar)** – Seçili sınıflamaya göre alt ağları oluşturur:

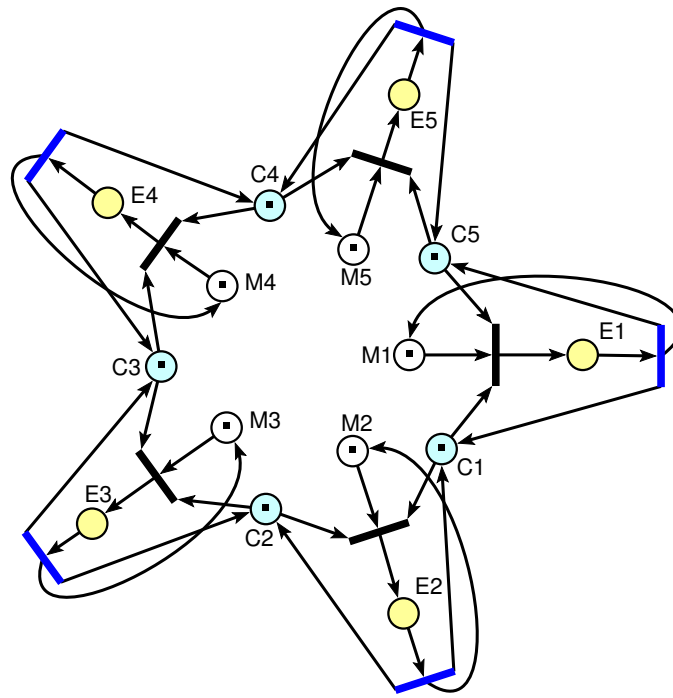
- * **SubNetwork Induced by Union of Selected Clusters** (Seçili Kümelerin Birleşimi Tarafından İndüklenmiş Alt Ağ) – Seçili sınıflamaya göre bir alt ağ oluşturur (Bir sınıflamadan belirli bir aralıktaki kümeleri çeker). Sonuçta bir alt ağ oluşturulur. Çekilen sınıflama ekstra bir sonuç olarak oluşturulur.
- * **SubNetworks Induced by Each Selected Cluster Separately** (Seçili Her bir Küme için Ayrı Ayrı İndüklenmiş Alt Ağlar) – Bir sınıflamadaki seçili kümeler tarafından indüklenmiş alt ağları çeker. Tek bir komut ile bir çok alt ağ çekilir. Her bir seçilen küme bir tane indüklenmiş alt ağ oluşturur.
- * **2-Mode Network** (İki modlu ağ) – Bir modlu ağdan iki modlu bir ağ çeker: birinci ve ikinci mod sınıflamanın verilen kümeleri tarafından belirlenir
- * **to GEDCOM** (GEDCOM'a aktar) – Seçili sınıflamaya göre bir alt soyağacı çeker (bu zayıf bağlantılı bir bileşen oluşturur) ve bir GEDCOM dosyasına aktarır (soyağacı bir Ore çizgesi olarak okunmalıdır).

– **Shrink Network** (Ağı Küçült) – Seçili sınıflamaya göre ağı küçült. Kümesi (sınıfı) 0 olan düğümler (varsayılan) değişikliğe uğramaz fakat diğerleri küçültülür. Sonuçta küçültülmüş ağ ve küçültülmüş sınıflama oluşturulur. Küçültmeye başlamadan önce Seçenekler (Options) menüsünden uygun blokmodeli seçin. Varsayılan: küçültülmüş ve orijinal ağda olan düğümler arasındaki bağlantı sayısı yeni ağda bir bağlantı haline gelir.

– **Brokerage Roles** (Aracı Roller) – Her bir düğüm i için verilen sınıflamaya göre beş aracılık rolü sayar: *coordinator* (koordinatör), *itinerant broker* (seyyar aracı), *representative* (temsilci), *gatekeeper* (kapıcı) ve *liaison* (irtibat).



- **Internal/External Cluster** (İç/Dış Küme) – Verilen bir ağ ve sınıflama için iki vektör oluşturulur: $Iç$ and $Dış$ vektörler. Vektörlerin boyu sınıflamadaki en büyük değere (küme) eşittir. Sınıflamadaki her bir küme i için iki vektör aşağıdakileri verir:
 - * **Min Line Value** (Minimum bağlantı değeri) – $In[i]$: her iki ucundaki düğümü de aynı küme i 'de olan bağlantıların en küçük değeri. $Out[i]$: yalnızca bir ucundaki düğümü küme i 'de olan bağlantıların en küçük değeri.
 - * **Max Line Value** (Maksimum bağlantı değeri) – $In[i]$: her iki ucundaki düğümü de aynı küme i 'de olan bağlantıların en büyük değeri. $Out[i]$: yalnızca bir ucundaki düğümü küme i 'de olan bağlantıların en büyük değeri.
- **Expand Partition** (Sınıflamayı Genişlet)
 - * **Greedy Partition** (Obur Sınıflama) – Sınıfı bilinmeyen (0) düğümleri seçili düğümler olarak aynı sınıfa koyar eğer
 - **Input** (İç) ... seçili düğümlere en fazla k adımda ulaşabiliriz.
 - **Output** (Dış) ...seçili düğümlerden diğer düğümlere en fazla k adımda gidebiliriz.
 - **All** (Tüm) ...İç + Dış (bağlantıların yönlerini yok say)
 Eğer bir düğüm birden fazla sınıfa ait olursa sınıflar birleştirilir.
 - * **Influence Partition** (Nüfuz Sınıflaması) – Sınıfı bilinmeyen (0) düğümleri en yakın düğüm olarak verilen sınıflamadaki aynı sınıfa koy. Eğer bilinen sınıf numaralı bir çok düğüm aynı mesafede ise en yüksek değer kullanılır.
 - * **Make Multiple Relations Network** (Çoklu İlişkili Ağ Oluştur) – Sınıflamayı kullanarak ağı çoklu ilişkili bir ağa çevir: eğer bağlantının uç noktaları (uç düğümleri) sınıflamada aynı sınıfa aitse çoklu ilişki etiketi olarak uç noktaların sınıf numarası kullanılır. Değilse 0, bağlantının düğümün sınıf numarası veya son düğümünün sınıf numarası kullanılır.
- **Identify** (Belirle) – Belirle (yeniden sırala ve/veya bazı üniteler ekle).
- **Refine Partition** (Sınıflamayı Rafine Et) Seçili ağa göre sınıflamayı rafine et (reachability).
 - * **Strong** (Güçlü) ... yönlü ağlar için.
 - * **Weak** (Zayıf) ... yönsüz ağlar için.
- **Leader Partition** (Lider Sınıflaması) – tabakalar içindeki düğüm kümelerini bul.



Şekil 13: Petri ağı

- **Petri** (Petri) – Sınıflama ile belirlenen yerleri işaretlemeyle başlayarak Petri ağı çalıştır. Bir ağıdaki yer sayısı sınıflamanın büyüklüğüne eşittir. Yerler önce $(1..m)$ olarak belirlenir sonra da $(m + 1..n)$ 'e geçiş yapar. Birden fazla geçiş olursa yapılacak iki şey var:
 - * **Random** (Rastgele) – Geçiş rastgele seçilir.
 - * **Complete** (Tam) – Bütün mümkün geçişlerin ağacı oluşturulur - Sonuç bir hiyerarşidir. Burada ağacın maksimum derinliği seçilebilir veya mümkünse Petri Ağı çalıştırılır.

Peterson'ın kitabından [57, page 21] `petri2` ya da `petri52`'yi deneyebilirsiniz `petri52` (see Figure 13). (Figure 13 incelenebilir) [Veri burada](#).

- **Transform** (Transform) – Sınıflamaya göre ağ transformasyonları.
 - * **Remove Lines** (Bağlantıları sil) – Sınıflamaya göre bağlantıların silinmesi.
 - **Inside Clusters** (Sınıf İçinde) – Seçili aynı sınıf(lar) içindeki bütün düğümlerin bağlantılarını siler.
 - **Between Clusters** (Sınıflar Arasında) – Farklı sınıflar içindeki düğümleri birleştiren bütün bağlantıları sil.
 - **Between Two Clusters** (İki Sınıf Arasında)
 1. **Arcs** (Yaylar) – Seçili iki sınıf arasındaki bütün yayları sil.
 2. **Edges** (Kenarlar) – Seçili iki sınıf arasındaki bütün kenarları sil.
 - **Inside Clusters with value** (Sınıf İçinde Değere Göre)
 1. **lower than Vector value** (vektör değerinden düşük olanlar) – Sınıflamayla belirlenen bir sınıf içindeki bir vektör tarafından belirlenen bir değerden düşük bütün bağlantıları sil.
 2. **higher than Vector value** (vektör değerinden yüksek olanlar) – Sınıflamayla belirlenen bir sınıf içindeki bir vektör tarafından belirlenen bir değerden yüksek bütün bağlantıları sil.

Vektörün büyüklüğü sınıflamadaki en yüksek sınıf değerine eşit olmalıdır.
 - * **Add Time Intervals determined by Partitions** (Sınıflama Tarafından Belirlenen Zaman Aralıklarını Ekle) – iki sınıflama kullanılarak ağı bir zamansal ağa çevirir: birinci sınıflama ilk zaman noktasını ikincisi ise son noktasını belirler

- * **Direction** (Yön) – Bütün yayların aşağıdaki gibi yönlendirildiği bir yönlü ağ oluştur
 - **Lower->Higher** (Düşükten Yüksekçe) Düşükten yüksek sınıf numarasına.
 - **Higher->Lower** (Yüksekten Düşüğe) Yüksekten düşük sınıf numarasına.

Sınıf içindeki bağlantılar silinebilir de silinmeyebilir de.
- **+ Cluster** (+ Küme) – Bir ağ, bir sınıflama ve bir küme gerektiren işlemler.
 - * **Count Neighbour Clusters** (Komşu Kümeleri Say) – Seçili ağ ve sınıflamadaki her bir küme numarası için bir vektör oluşturulur. Elde edilen her vektörde her bir düğümün komşuluğundaki verilen kümenin düğümlerinin sayısı bulunur. Vektör olarak sonuçlarını istediğimiz küme numaraları kümede verilmelidir. İstisna: Küme bir düğüm kümesi yerine bir grup küme numaralarını temsil etmesi durumu hariç.
- **+ Permutation** (+ Permütasyon) – Bir ağ, bir sınıflama ve bir permütasyon gerektiren işlemler.
 - * **Coloring** (Renklendirme)
 - **Create New** (Yenisini Oluştur) – Permütasyon tarafından belirlenen sırada düğümlerin boyanması. Sonuç seçilen permütasyona fazlasıyla bağlı olacaktır.
 - **Complete Old** (Eskisini Tamamla) – Permütasyon tarafından belirlenen sırada düğümlerin kısmi boyanması. Örneğin bazı düğümler elle boyanabilir fakat düğümlerin çoğu boyanmamış (0 sınıfında) olabilir. Bu durumda daha iyi bir boyama için programa yardım edebilirsiniz.
- **Info** (Bilgi) – Verilen bir ağa ait sınıflama hakkında bilgi.
 - * **Modularity** (Modülarite) – Sınıflamaya göre network modülaritesi.
 - * **VOS Quality** (VOS Kalitesi) – Sınıflamaya göre VOS Kalite skoru.
 - * **E-I Index** (E-I endeksi) – Sınıflamaya göre E-I endeksi (ağırlıklı veya ağırlıksız).
- **Network + Vector** (Ağ + Vektör) – Bir ağ ve vektör üzerindeki işlemler. (Difüzyon Sınıflaması durumunda küme de eklenebilir. Harmonik Fonksiyon durumunda sınıflama ve permütasyon da eklenebilir.).

- **Network * Vector** (Ağ + Vektör) – Vektörün matrisle (ağ) çarpılması. Sonuç yeni bir vektördür.
- **Vector # Network (Vektör # Ağ)** – Sonuç yeni bir ağdır:
 - * **Input (İç)** – Bir ağdaki iç yayları karşılık gelen vektör değerleriyle çarparak – matrisin *i*. kolonunu vektörün *i*. bileşeni ile çarparak.
 - * **Output (Dış)** – Bir ağdaki dış yayları karşılık gelen vektör değerleriyle çarparak – matrisin *i*. satırını vektörün *i*. bileşeni ile çarparak.
- **Neighbours** (Komşular)
 - * **Sum** (Toplam) – Her bir düğüm için komşularının vektör değerlerini topla. Üç durum var:
 - **Input (İç)** – komşular
 - **Output (Dış)** – komşular
 - **All** (Tüm) – komşular
 - * **Min (minimum)** – Her bir düğüm için komşularının vektör değerlerinin en küçüğünü hesapla. Üç durum var:
 - **Input (İç)** – komşular
 - **Output (Dış)** – komşular
 - **All** (Tüm) – komşular
 - * **Max** (Maksimum) – Her bir düğüm için komşularının vektör değerlerinin en büyüğünü hesapla. Üç durum var:
 - **Input (İç)** – komşular
 - **Output (Dış)** – komşular
 - **All** (Tüm) – komşular
- **Islands** (Adalar) – Düğümleri vektörle belirlenen ağırlıklarına göre yapışkan (cohesive) kümelere sınıfla.
 - * **Vertex Weights** (Düğüm Ağırlıkları) – Düğüm adası içindeki düğümlerin ağırlıklarının komşuluktaki düğümlerin ağırlıklarından fazla olduğu bir düğüm kümesidir. Burada ağırlıklar bazen yükseklik olarak da adlandırılır.
 - * **Vertex Weights [Simple]** (Düğüm Ağırlıkları [Basit]) – Basit düğüm adası sadece bir tepe değeri olan düğüm adasıdır.
- **Transform** (Transform) – Ağ vektöre göre transforme et.
 - * **Put Loops** (İlmekler Ekle) – vektör değerlerini mevcut ağa ilmek olarak ekle (yay veya kenar olarak)
 - * **Put Coordinate** (Koordinat ekle) – vektöre *x*, *y* veya *z* koordinatı olarak ekle veya düğümlerin polar yarıçap ve polar açısı olarak ağ yerleşimine ekle.

- * **Vector(s) -> Line Values** (Vektör(ler) -> Bağlantı Değerleri) – Bağlantı değerlerini iki ucundaki düğüme karşılık gelen vektör değerleri üzerinde bazı işlemlerin sonucuyla değiştir (toplam, fark, çarpma, bölme, minimum, maksimum). Bağlantı değeri ayrıca iki ucundaki düğümün vektör değerleri ile de değiştirilebilir.
- + **Cluster** (+ Sınıf) – Bir ağ, bir vektör ve bir küme üzerindeki işlemler.
- * **Diffusion Partition** (Difüzyon Sınıflaması) – Bir vektörle verilen eşik değerlere göre difüzyon sınıflamasını hesapla. Seçili küme-
deki düğümlerin zaman noktası 1’de adapte olduğu kabul edilir.
- + **Partition + Permutation** (+ Sınıflama + Permütasyon) – Bir ağ, bir vektör, bir sınıflama ve bir permütasyonla yapılan işlemler.
- * **Harmonic Function** (Harmonik Fonksiyon) – Bollobas’a bakabilirsiniz [25, page 328]. (G, a) bağlantılı ve $a(x, y)$ ağırlık fonksiyonu ile ağırlıklı bir ağ olsun. Ayrıca S de bu ağın düğüm kümesi $V(G)$ ’nin bir alt kümesi olsun. Bir fonksiyon $f : V(G) \rightarrow \mathbb{R}$ olarak tanımlansın. Bu fonksiyonun S sınır olmak üzere (G, a) ’da harmonik olması için aşağıdaki şartın sağlanması gerekir.

$$f(x) = \frac{1}{A(x)} \sum_y (a(x, y)f(y)), \quad \forall x \in V(G) \setminus S$$

$$A(x) = \sum_y a(x, y)$$

Pa jek’te nasıl gerçekleştirilir:

- f fonksiyonu vektör tarafından belirlenir
- ağırlık fonksiyonu $a(x, y)$ (ağırlıklı) tarafından verilir
- alt küme S sınıflama tarafından belirlenir – 1 sınıfındaki (küme-
sindeki) düğümler S alt kümesindedir (sabit düğümler), diğer
düğümler ise $V(G) \setminus S$ ’dedir.
- ek olarak, permütasyon düğümlerin hesaplamadaki sırasını belirler

Pa jek’de harmonik fonksiyon bir kereliğine veya ard arda işlemlerde fonksiyon değerleri arasında fark yeterince küçükse yinelemeli şekilde hesaplanabilir. Fonksiyon f ’i temsil eden vektörün bileşenleri hesaplandıkları anda veya sadece her bir yinelemenin sonunda değiştirilebilir (bütün bileşenler hesaplandıktan sonra). Prosedür bağlantıların yönüne göre gerçekleştirilebilir:

- **Input** (İç) – komşular
- **Output** (Dış) – komşular

- **All** (Tümü) – komşular
- **Info** (Bilgi) – bir veya iki vektör kullanılarak ağın indisleri elde edilebilir (birisi bağlantının başlangıç düğümü ve diğeri ise son düğümü için)
 1. **Assortativity** (Asortativite) – bir veya iki vektör kullanılarak ağın indisi elde edilebilir. Eğer iki vektör kullanılırsa birincisi bağlantının başlangıç düğümünün sayısal değerini tanımlar diğeri ise son düğümün sayısal değerini. Bir vektör kullanılırsa iki değeri de tanımlar.
- **Network + Scalar** (Ağ + Skaler) – Bir ağ ve bir skaler üzerindeki işlemler. Skaler boyutu 1 olan vektördür.
 - **Line Values** (Bağlantı Değerleri) – Bağlantı değerleri verilen skaler değeri ile işleme tabi tutulur: Ekle, çıkar, çarp, böl, minimum veya maksimum.
- **Network + Permutation** (Ağ + Permütasyon) – Bir ağ ve bir permütasyon işlemleri.
 - **Reorder Network** (Ağı Tekrar Düzenle) – Seçilen permütasyona göre ağın düğümlerini yeniden sıralar.
 - **Numbering*** (Numaralandırma*) – Ağa göre verilen permütasyonu numaralandır.
 - * **Travelling Salesman** (Gezgin Satıcı) – Benzemezlik matrisine veya ağı temsil eden değiştirilmiş matrise uygulanabilir (köşegeni doldur ve matristeki sıfırları büyük değerlerle değiştir).
 - **Run** (Çalıştır) – Gezgin satıcı problemini çözmek için 3-OPT algoritmasını çalıştır.
 - **Options** (Seçenekler) – Seçilen değeri köşegene koy, birkaç yapay düğüm ve yüksek ağırlıklı bağlantılar ekle, 0 değerini seçilen büyük bir değerle değiştir.
 - * **Seriation** (Seri Dizilim) – Ağla ve rastgele bir permütasyonla başlayıp permütasyonu Murtagh’ın seri dizilim algoritması ile iyileştir (Bkz. [54, page 11-16]).
 - **1-Mode** (Bir modlu) – sıradan bir modlu ağlar
 - **2-Mode** (İki modlu) – iki modlu ağlar
 - * **Clumping** (Topaklaşma) – Ağla ve rastgele bir permütasyonla başlayıp permütasyonu Murtagh’ın topaklaşma algoritması ile iyileştir (Bkz. [54, page 11-16]).
 - **1-Mode** (Bir modlu) – bir modlu normal ağlar için

- **2-Mode** (İki modlu) – iki modlu ağlar için
 - * **R-Enumeration** (E numaralandırması) – Ağla ve rastgele bir permütasyonla başlayarak öyle bir permütasyon bul ki komşu düğümlerin numaralandırılması birbirine mümkün olduğu kadar yakın olsun.
 - **Network + Cluster** (Ağ + Küme) – Bir ağ ve küme üzerindeki işlemler.
 - **Extract SubNetwork** (Alt ağ çek) – Seçilen kümeye göre alt ağ çek.
 - **Dissimilarity*** (Benzemezlik*)
 - * **Network based** (Ağ Tabanlı) – Sınıf içindeki düğümler arasında ortak komşu sayısına bağlı olarak seçili benzemezlik matrisini oluştur (d_1, d_2, d_3 veya d_4) *Düzeltilmiş Öklit benzeri* d_5 ve *Manhattan benzeri* d_6 benzemezlik skorları da hesaplanabilir. [13]. Elde edilen matris daha sonra hiyerarşik kümelendirme prosedürü için kullanılabilir.
 - Komşuluk listesine düğüm v 'nin kendisi de dahil edilebilir ve raporlama penceresinde sadece üst üçgen (yönsüz ağlarda) veya tam matris (yönlü ağlarda) gösterilebilir (eğer düğüm sayısı düşükse).
- N_v : v düğümünün iç, dış veya bütün komşuları, +: simetrik toplam, \cup : küme birleşimi, \setminus : küme farkı, $|\cdot|$: kümenin büyüklüğü (kardinalite), 1. maks. derece: büyük ağdaki derece değeri ve 2. maks. derece: ağdaki en büyük ikinci derece değeri.

$$d_1(u, v) = \frac{|N_u + N_v|}{1. \text{ maks. derece} + 2. \text{ maks. derece}}$$

$$d_2(u, v) = \frac{|N_u + N_v|}{|N_u \cup N_v|}$$

$$d_3(u, v) = \frac{|N_u + N_v|}{|N_u| + |N_v|}$$

$$d_4(u, v) = \frac{\max(|N_u \setminus N_v|, |N_v \setminus N_u|)}{\max(|N_u|, |N_v|)}$$

$$d_5(u, v) = \sqrt{\sum_{\substack{s=1 \\ s \neq u, v}}^n ((q_{us} - q_{vs})^2 + (q_{su} - q_{sv})^2) + p \cdot ((q_{uu} - q_{vv})^2 + (q_{uv} - q_{vu})^2)}$$

$$d_6(u, v) = \sum_{\substack{s=1 \\ s \neq u, v}}^n (|q_{us} - q_{vs}| + |q_{su} - q_{sv}|) + p \cdot (|q_{uu} - q_{vv}| + |q_{uv} - q_{vu}|)$$

d_5 ve d_6 benzemezlikleri düğümler üzerinde tanımlı $Q = [q_{uv}]$ matrisine dayanır. Örneğin komşuluk matrisi veya mesafe matrisi. p matrisi genelde 1 veya 2 seçilir. $N_u = N_v = 0$ olması durumunda d_1 'den d_4 'e bütün benzemezlik skorları 1'e eşitlenir.

Among all linked Vertices only (Sadece bütün bağlantılı düğümler arasında) seçeneği işaretliyse benzemezlik skorları verilen ağın bağlantı değerleri olarak hesaplanır.

* **Vector based (Vektör Tabanlı)** – Sınıf tarafından belirlenen vektörler arasındaki **Öklit, Manhattan, Canberra, veya (1-Cosine)/2** benzemezlik skorları verilen ağın bağlantı değerleri olarak hesaplanır.

– **k -Neighbors from Vertices in Cluster** (Kümedeki düğümlerden k komşuları) – Verilen kümedeki her bir düğüm için aşağıdaki mesafeler hesaplanır

* **Input (İç)** ...bütün diğer düğümlerden.

* **Output (Dış)** ...bütün diğer düğümlere.

* **All (Tümü)** ...bağlantıların yönünü dikkate almadan bütün mesafeler

Seçilen düğümden ulaşılamayan düğümler için mesafe 999999999 olarak alınır. Sonuçta sınıftaki her bir düğüm için bir vektör elde edilir.

– **Distribution of Distances from Vertices in Cluster*** (Kümedeki Düğümlerden Mesafelerin Dağılımı*) – En kısa mesafelerin dağılımını ve ağdaki ulaşılabilir bütün düğüm çiftleri arasındaki ortalama mesafeyi hesapla. Sadece küme olarak seçilen düğümlerden mesafeler hesaplanır.

– **Transform** (Transform) – Sınıfa göre ağı transforme et.

* **Add Arcs from Vertex to Cluster** (Düğümden Kümeye Yay Ekle) – seçili düğümden kümedeki diğer bütün düğümlere yay ekle.

* **Add Arcs from Cluster to Vertex** (Kümeden Düğüme Yay Ekle) – kümedeki bütün düğümlerden seçilen düğüme yay ekle.

• **Network + Hierarchy** (Ağ + Hiyerarşi) – Ağ ve hiyerarşi üzerindeki İşlemler.

– **Shrink Network** (Ağı Küçült) – Seçilen hiyerarşiye göre ağı küçült. Hiyerarşideki *Kapalı (closed)* olan düğümler yeni bir düğüm haline getirilir. *Cut (Kesik)* olan düğümler bir sanal düğüme küçültülür. *Sınır (Border)* olan düğümler küçültülmez fakat görünebilir değildir. Diğer düğümler değişmeden kalır. Küçültme yöntemi (blok model) Options (Seçenekler) menüsünden seçilebilir.

- **Expand Reduction** (İndirgemeyi Genişlet) – İndirgenmiş ağ (hiyerarşik indirgeme) ve ilgili hiyerarşi original ağa genişletilir. Sonuç her zaman bir yönlü ağdır.
- **VertexID + Partition** (DüğümKimlik + Sınıflama) – DüğümKümlük ve sınıflama işlemleri.
 - **Extract SubVertexIDs** (DüğümKümlükleri Çıkar) – Yalnızca sınıflama tarafından belirlenen verilen kritere (verilen bir aralığa) uyan kimlikleri çıkar
- **Partition + Permutation** (Sınıflama + Permütasyon) – Sınıflama ve permütasyon işlemleri.
 - **Reorder Partition** (Sınıflamayı Tekrar Düzenle) – Permütasyona göre. Aynı sonuç fonksiyonel kompozisyon **Permutation*Partition** (Permütasyon*Sınıflama) kullanarak da elde edilebilir: f bir permütasyon ve g ise bir sınıflama olsun. Sonuç aşağıdaki gibi tanımlanan yeni bir sınıflamadır (r): $r[v] = (f * g)[v] = g[f[v]]$.
 - **Functional Composition Partition*Permutation** (Fonksiyonel Kompozisyon Sınıflama*Permütasyon) – f bir sınıflama ve g ise bir permütasyon olsun. Sonuç aşağıdaki gibi tanımlanan yeni bir sınıflamadır (r): $r[v] = (f * g)[v] = g[f[v]]$.
- **Partition + Cluster** (Sınıflama + Küme) – Bir sınıflama ve küme üzerinde tanımlı işlemler.
 - **Binarize Partition** (Sınıflamayı İkileştir) – Sınıflamayı kümeye göre ikileştir – verilen sınıflamanın boyutunda bir ikili sınıflama oluştur, küme tarafından belirlenen küme numaralarına sahip düğümler sınıf 1'e diğerleri sınıf 0'a gider. İstisna: bu durumda küme sınıf numaralarının bir kümesini temsil eder, düğüm numaralarının bir kümesini değil.
- **Vector + Partition** (Vektör + Sınıflama) – Vektör and Sınıflama üzerindeki işlemler.
 - **Extract Subvector** (Alt Vektörü Çıkar) – Verilen bir vektörden bir alt vektörü çıkar – kriter seçili sınıflamadaki sınıftır.
 - **Shrink Vector** (Vektörü Küçült) – Sınıflamanın küme numaralarına göre vektör değerlerinden yeni bir vektör oluştur – vektör küçültülmüş ağa göre ayarlanır. Birçok değeri bir değere küçültmede değerlerin toplamı, ortalaması, minimumu, maksimumu veya ortancası kullanılabilir.

- **Functional Composition Partition*Vector** (Fonksiyonel Kompozisyon Sınıflama*Vektör) – f bir sınıflama ve g ise bir vektör olsun. Sonuç aşağıdaki gibi tanımlanan yeni bir vektör r 'dir: $r[v] = (f * g)[v] = g[f[v]]$.
- **Vector + Permutation** (Vektör + Permütasyon) – Vektör ve Permütasyon üzerindeki işlemler.
 - **Reorder Vector** (Vektörü Yeniden Düzenle) – permütasyona göre. Aynı sonuç fonksiyonel dekompozisyon **Permutation*Vector** (Permütasyon*Vektör) kullanılarak da elde edilebilir: f bir permütasyon ve g ise bir vektör olsun. Sonuç aşağıdaki şekilde tanımlanan yeni bir vektör r 'dir: $r[v] = (f * g)[v] = g[f[v]]$.

3.6 Partition (Sınıflama)

Girdi olarak yalnızca sınıflama kullanılır.

- **Create Constant Partition (Sabit Sınıflama oluştur)** – Seçili büyüklükte sabit değerli bir sınıflama oluştur. Varsayılan büyüklük seçili ağdaki düğüm sayısıdır (bellekte bir ağ varsa).
- **Create Identity Partition (Kimlik Sınıflaması Oluştur)** – Kimlik sınıflaması oluşturur ($C[i] := i$, her bir node kendine ait sınıfta olacaktır).
- **Create Random Partition (Rastgele Sınıflama Oluştur)** – Rastgele değerlere sahip bir veya iki modlu sınıflama oluştur.
- **Binarize Partition (Sınıflamayı İkilileştir)** – Seçili sınıflamayı 0-1 şeklinde ikilileştir.
- **Fuse Clusters (Kümelere Kaynaştır)** – Seçili sınıflama numaralarını yeni bir sınıflamaya kaynaştır.
- **Canonical Partition (Kanonik Sınıflama)**
 - **with Vertex 1 in Cluster 1** (Küme 1'deki Düğüm 1'le) – Sınıflamayı kanonik (tekil) formuna transforme et (düğüm 1 her zaman küme 1'de, düğüm 1'le aynı kümeden olmayan en küçük numaralı sonraki düğüm küme 2'de, ...)
 - **with Decreasing Frequencies** (Azalan Frekanslarla) – Sınıflamayı kanonik (tekil) formuna transforme et (küme 1'de en yüksek frekanslı eski küme olur, küme 2'de en yüksek ikinci frekanslı eski küme, ...)

- **Make Network** (Ağ Oluştur) – Sınıflamadan bir ağ oluştur
 - **Random Network** (Rassal Network) – Düğümlerin dereceleri sınıflamadan gelecek şekilde bir rassal ağ oluştur..
 - * **Undirected** (Yönsüz) – sınıflama yönsüz ağdaki düğümlerin derecelerini verir.
 - * **Input** (İç derece) – sınıflama düğümlerin iç derecelerini verir.
 - * **Output** (Dış derece) – sınıflama düğümlerin dış derecelerini verir.
 - **2-Mode Network** (2-modlu Ağ) – İki modlu bir ağ oluştur: İlk küme ($v_1 \dots v_n$) düğümlerini içerir, ikinci küme ise ($c_0 \dots c_m$) sınıflarını. Eğer düğüm i sınıf j 'de ise v_i 'den to c_j 'ye bir bağlantı oluşturulur. *Existing Clusters only* (yalnızca mevcut sınıflar) seçeneği seçilmişse ikinci ağda yalnızca en az bir düğüm içeren sınıflar oluşturulur.
- **Copy to VertexID** (DüğümKimliğe Kopyala) – Sınıflamanın kümelerini düğüm kimliklerine çevir.
- **Make Permutation** (Permütasyon Oluştur) – Seçili sınıflamadan permütasyon oluşturur (önce en düşük sınıf numaralı bütün düğümler, sonra ...)
- **Make Cluster** (Küme Oluştur) – Sınıflamayı kümeye dönüştür.
 - **Vertices from Selected Clusters** (Seçili Kümelerden Düğümler) – Sınıflamadaki seçilen kümelerden düğümleri kümeye kopyala.
 - **Random Representatives of each Cluster** (Her bir Kümenin Rastgele Temsilcisini Seç) – Sınıflamadaki her bir kümeden rastgele bir düğüm (temslci) seç. Temsilcileri bir kümeye kopyala. Bu temsilciler daha sonra Pivot MDS algoritmasında pivot olarak kullanılabilir.
- **Make Hierarchy** (Hiyerarşi Oluştur) – Sınıflamayı hiyerarşiye dönüştür (iç içe veya değil).
- **Copy to Vector** (Vektöre Kopyala) – Sınıflamayı vektöre kopyala ($V[i] := C[i]$).
- **Count, Min-Max Vector** (Say, Min-Maks Vektör) – Küme frekansları hakkında bilgi, verilen bir sınıflamaya göre minimum ve maksimum vektör değerleri vektör değerleri.
- **Info** – Sınıflama hakkında genel bilgiler. En önemli düğümleri görmek için düğümler sınıf numarasına göre (azalan ya da artan) şekilde sıraya dizilir. Sınıf numaralarının frekans dağılımı, ortalaması, ortancası ve standart sapması gibi bilgiler de verilir.

3.7 Partitions (Sınıflamalar)

İki sınıflama ile yapılan işlemler. İşlemleri gerçekleştirmek için iki sınıflamanın seçilmesi gerekmektedir.

- **Extract SubPartition (Second from First)** (Alt Sınıflamayı Çıkar (Çek) (Birinciden İkinciye) – Birinci sınıflamadan yalnızca ikinci sınıflama tarafından belirlenen gerekli kriteri sağlayan (aralık olarak belirlenen) düğümleri al. Bu işlem özellikle düğümler hakkında bilgi içeren (örneğin cinsiyet) sınıflamalar olduğunda faydalıdır. Ağın daha küçük bir kısmını aldığımızda (çıkardığımızda) (örneğin seçilen düğümden 3 adım veya daha az mesafede olan düğümler) aynı çıkarma işlemini cinsiyet sınıflaması üzerinde de yapmazsanız cinsiyet bilgileri kaybedersiniz.
- **Add (First+Second) (Birinci ve İkinci Sınıflamayı Birleştir)** – İki sınıflamayı birleştir. Bu işlem örneğin çevrimsiz ağlarda iç ve dış komşuları birleştirirken faydalı olabilir.
- **Min (First, Second)** (Minimum) – İki sınıflamanın minimumu.
- **Max (First, Second)** (Maksimum) – İki sınıflamanın maksimumu.
- **Fuse Partitions** (Sınıflamaları Kaynaştır) – İki sınıflamayı kaynaştır – ikinciyi birincinin sonuna ekle (iki modlu ağlar için faydalıdır).
- **Expand Partition** (Sınıflamayı Genişlet) – Sınıflamanın boyutunu (orijinal haline) genişlet.
 - **First according to Second – Shrink** (İkinciye Göre Birinci – Küçült) – Birinci sınıflamayı ikinci sınıflama tarafından belirlenen küçültmeye göre genişlet.
 - **Insert First into Second according to Third - Extract** (Üçüncüye Göre Birinciyi İkincinin Arasına Ekle - Çek) – Mevcut sınıflama ikinci sınıflama tarafından tanımlanan seçili sınıfları birinci sınıflamadan çekilmesiyle elde edilir. Bu alt sınıflama modifiye edilmiştir. Bu işlemi kullanarak modifiye edilmiş alt sınıflamayı birinci sınıflamaya geri sokabiliriz.
- **Intersection of Partitions** (Sınıflamaların Kesişimi) – Seçilen sınıflamaların kesişimi.
- **Cover with** (İle Ört) – p bir sınıflama, b bir ikili sınıflama, c ise seçilen küme numarası olsun. Sonuç aşağıdaki gibi tanımlanan yeni bir sınıflama olan q 'dur:
Eğer $b(v) = 0$ ise $q(v) = p(v)$ yoksa $q(v) = c$.

- **Merge Partitions** (Sınıflamaları Kaynaştır (İç içe sok)) – p ve q sınıflamalar ve b ise ikili bir sınıflama olsun. Sonuç aşağıdaki gibi belirlenen yeni bir sınıflama s olur:
Eğer $b(v) = 0$ ise $s(v) = p(v)$ yoksa $s(v) = q(v)$.
- **Make Random Network** (Rassal Ağ Oluştur) – İç dereceleri birinci sınıflamadan dış dereceleri ikinci sınıflamadan gelen rassal bir ağ oluştur.
- **Functional Composition First*Second** (Fonksiyonel Kompozisyon Birinci*İkinci)– f ve g iki sınıflama olsun. Sonuç aşağıdaki gibi tanımlanan yeni bir sınıflama r olur: $r[v] = (f * g)[v] = g[f[v]]$.
- **Info** (Bilgi) – Seçilen iki sınıflama arasındaki bazı istatistiksel analizler:
 - **Cramer’s V, Rajsiki, Adjusted Rand Index** – Gerçeklik tablosu yanında Cramer V, Rajsiki katsayıları and düzeltilmiş Rand indeksi hesaplanır.
 - **Spearman Rank** – Spearman sıra korelasyonu katsayısı.

3.8 Vector (Vektör)

Vektör işlemleri.

- **Create Constant Vector** (Sabit Vektör Oluştur) – Seçilen büyüklükte ve bütün değerleri aynı olan bir vektör oluştur. Varsayılan boyut bellekte seçilen ağın boyutudur.
- **Create Scalar** (Skaler Oluştur) – Verilen vektörden bir skaler oluştur. Skal-ler boyutu bir olan bir vektördür.
- **Make Partition** (Sınıflama Oluştur) – Vektörü bir sınıflamaya dönüştür. :
 - **by Intervals** (Aralıklarla) – vektördeki seçili bölünme numaralarına göre uygun sınıf numaralarını bul. Aralıklar aşağıdaki gibi verilebilir:
 - * **First Threshold and Step** (İlk Eşik ve Adım) – İlk eşiği ve artırma miktarını (adım) seç.
 - * **Selected Thresholds** (Seçilen Eşiklerle) – Bütün eşikleri seç ya da sınıf sayısını (#) baştan seç.
 - **Copy to Partition by Truncating (Abs)** (Kırparak Sınıflamaya Kopyala (mutlak değerle)) – vektör değerlerinin mutlak değeri alınıp ondalık kısımları kırılır ve bir sınıflamaya kopyalanır.
- **Make Permutation** (Permütasyon Oluştur) – Vektörü permütasyona çevir – permütasyonu sıraya dizerek.

- **Make Cluster** (Küme Oluştur) – Vektörü bir kümeye çevir – seçili değerden düşük ya da yüksek olan vektör değerlerine sahip düğümleri seç.
- **Make 2-Mode Network** (İki Modlu Ağ Oluştur) – Vektörü iki modlu ağa çevir (satır veya kolon).
- **Transform** (Transform) – Verilen vektörü transforme et:
 - **Multiply by** Bir sabitle çarp.
 - **Divide by** Bir sabite böl.
 - **Add Constant** Vektör değerlerine bir sabit ekle.
 - **Absolute** mutlak değerini al.
 - **Absolute + Sqrt** – önce mutlak değerini sonra kare kökünü al.
 - **Truncate** – ondalık kısımlarını kırp.
 - **Exp** – eksponansiyelini al.
 - **Ln** – tabii logaritmasını al.
 - **Power** – üssünü al.
 - **Normalize** (normalize et)
 - * **Sum** (Toplam) – öğelerin toplamı 1 olacak şekilde normalize et.
 - * **Max** (Maksimum) – öğelerin en büyüğü 1 olacak şekilde normalize et.
 - * **Standardize** (Standardize Et) – öğelerin aritmetik ortalaması 0 ve standard sapması 1 olacak şekilde standardize et.
 - * **Number of Vertices** (Düğüm Sayısı) – düğüm sayısına (vektörün boyutu) böl.
 - * **Number of Vertices - 1** (Düğüm Sayısı - 1) – düğüm sayısının bir eksiğine (vektörün boyutu-1) böl.
 - **Invert** (Tersini Al) – Vektördeki değerlerin tersini al.
 - **Cumulatives** (Birikimli) – aşağıdaki şekilde vektör v 'nin birikimli hali ile yeni bir u oluşturulur: $u_1 = v_1, u_i = u_{i-1} + v_i, i > 1$.
- **Missing Values** (Eksik Değerler) – Vektördeki 999.999.997'den büyük değerler işlemlerde nasıl kullanılmalı: geçerli değerler olarak mı yoksa eksik değerler olarak mı?
- **Info** (Bilgi) – Vektör hakkında genel bilgiler: düğümler değerlerine göre sıraya dizilir ve ortalama, medyan veya standard sapma ve frekans dağılımı verilen sınıf numarasına göre ayrılır (sınıf sayısı veya seçilen bölünme değerleri verilebilir).

3.9 Vectors (Vektörler)

İki vektörlü işlemler. İşlemleri yapmadan önce iki vektör seçilmelidir. Vektörler aynı boyutta olmalıdır veya bir vektör skaler olabilir.

- **Add (First+Second)** – Birinci ve ikinci vektörün toplamı.
- **Subtract (First-Second)** (Birinciden İkinciye Çıkar) – seçilen vektörlerin farkı.
- **Multiply (First*Second)** – seçilen vektörlerin çarpımı.
- **Divide (First/Second)** – seçilen vektörlerin bölümü.
- **Min (First, Second)** – seçilen vektörlerdeki daha küçük öğeler.
- **Max (First, Second)** – seçilen vektörlerdeki daha büyük öğeler.
- **Fuse Vectors (Vektörleri Kaynaştır)** – vektörleri kaynaştır.
- **Linear Regression** (Doğrusal Regresyon) – iki vektörü doğrusal regresyon kullanarak fit et . Sonuçta regresyon doğrusu denklemi, ikinci vektörün doğrusal tahminleri ve karşılık gelen hatalar elde edilir.
- **Transform (Transforme et)** – İki vektörü başka iki vektöre transforme et:
 - **Cartesian → Polar** (Kartezyenden Polar Koordinatlara) – Birinci vektör x ikincisi y koordinatlarını içermelidir. Sonuçta: polar yarıçaplar ve derece olarak açılar iki ayrı vektörde depolanır.
 - **Polar → Cartesian** (Polardan Kartezyen Koordinatlara) – Birinci vektör polar yarı çapları, ikincisi ise derece olarak açılarını içermelidir. Sonuçta c ve y koordinatlarını içeren iki vektör elde edilir.

Sonuçların çizim (draw) penceresinde doğrudan kullanılabilmesi için (de)normalize edebilirsiniz.

- **Missing Values (Eksik Değerler)** – 999.999.997’den büyük değerlerin vektör işlemlerinde nasıl alınacağını belirleyin: geçerli mi yoksa eksik değer olarak mı?
- **Info (Bilgi)** – Seçili vektörler arasındaki Pearson korelasyon katsayısı.

3.10 Permutation (Permütasyon)

Girdi olarak yalnızca permütasyon kullanır.

- **Create Identity Permutation** (Kimlik Permütasyonu Oluştur) – Seçilen boyuta sahip bir kimlik permütasyonu oluştur. Varsayılan boyut bellekte yüklü ve seçili olan ağı boyutudur.
- **Create Random Permutation** (Rassal Permütasyon Oluştur) – Seçilen boyutta bir veya iki modlu permütasyon oluştur. Varsayılan boyut bellekte yüklü ve seçili olan ağı boyutudur.
- **Inverse Permutation** (Permütasyonun Tersini Al) – Seçilen permütasyonun ters permütasyonunu oluştur.
- **Mirror Permutation** (Permütasyonun Ayna Görüntüsü) – Seçilen permütasyonun ayna görüntüsünü oluştur (ters yönde sırala).
- **Make Partition** (Sınıflama Oluştur) – Permütasyondan bir sınıflama oluştur.
 - **Into Given Number of Clusters** (Verilen Sayıda Kümeye) – Verilen küme sayısı ile sınıflama oluştur.
 - **Orbits** (Yörüngeler) – Permütasyondaki yörüngelere göre sınıflama oluştur.
- **Copy to Vector** (Vektöre Kopyala) – Permütasyonu vektöre kopyala.
- **Info** (Bilgi) – Önce permütasyonun geçerli olup olmadığını control et, sonra yörünge sayısına dön.

3.11 Permutations (Permütasyonlar)

İki permütasyon üzerinde yapılan işlemler

- **Fuse Permuations** (Permütasyonları Kaynaştır) – iki permütasyonu kaynaştır – ikinciyi birincinin sonuna ekle (iki modlu ağlar için faydalıdır)
- **Functional Composition First*Second** (Fonksiyonel Kompozisyon Birini*İkinci) – f ve g birer permütasyon olsun. Sonuç aşağıdaki gibi tanımlanan yeni bir permütasyon r 'dir: $r[v] = (f * g)[v] = g[f[v]]$. Permütasyonların boyutları farklıysa sonuç bir sınıflama olur.

3.12 Cluster (Küme)

Yalnızca bir küme üzerinde yapılan işlemler.

- **Create Empty Cluster** (Boş Küme Oluştur) – Hiç bir düğümü olmayan boş bir küme oluştur.
- **Create Complete Cluster** (Tam bir Küme Oluştur) – 1...n arası bütün düğümlerin içinde olduğu bir küme oluştur.
- **Create Random Cluster** (Rassal Bir Küme Oluştur) – Seçilen maksimum değere ve boyuta sahip rassal bir küme oluştur.
- **Make Partition** (Sınıflama Oluştur) – Sınıflamayı kümeye transforme et.
- **Info** (Bilgi) – Küme hakkında bilgiler: boyutu, en küçük ve en büyük değeri

3.13 Hierarchy (Hiyerarşi)

Yalnızca bir hiyerarşi üzerinde yapılan işlemler

- **Extract Cluster** (Küme Çek) – Hiyerarşiden bir küme oluştur – küme hiyerarşideki seçilen boğumun tam alt ağacıdır.
- **Make Network** (Ağ Oluştur) – Hiyerarşiyi bir ağa çevir (Örneğin bu hiyerarşiyi çizmek için kullanılabilir – tabakalı çizim). Kapalı boğumlar da hesaba katılır.
- **Make Partition** (Sınıflama Oluştur) – Hiyerarşiyi sınıflamaya çevir (kapalı boğumlara göre).
- **Make Permutation** (Permütasyon Oluştur) – Hiyerarşiyi permütasyona çevir.
- **Info** (Bilgi) – Hiyerarşi hakkında genel bilgiler. Bu işlem yalnızca düğüm numaraları tam sayı olduğunda mümkündür ve hiyerarşinin ilk seviyesindeki boğumlarıdaki düğüm sayısını verir.

3.14 Options (Seçenekler)

- **Read - Write** (Oku - Yaz)
 - **Threshold (Eşik)** – İki düğüm arasında bir bağlantı oluşturmak için bağlantının değeri verilen eşik değerden mutlak değer olarak yüksek olmalıdır.

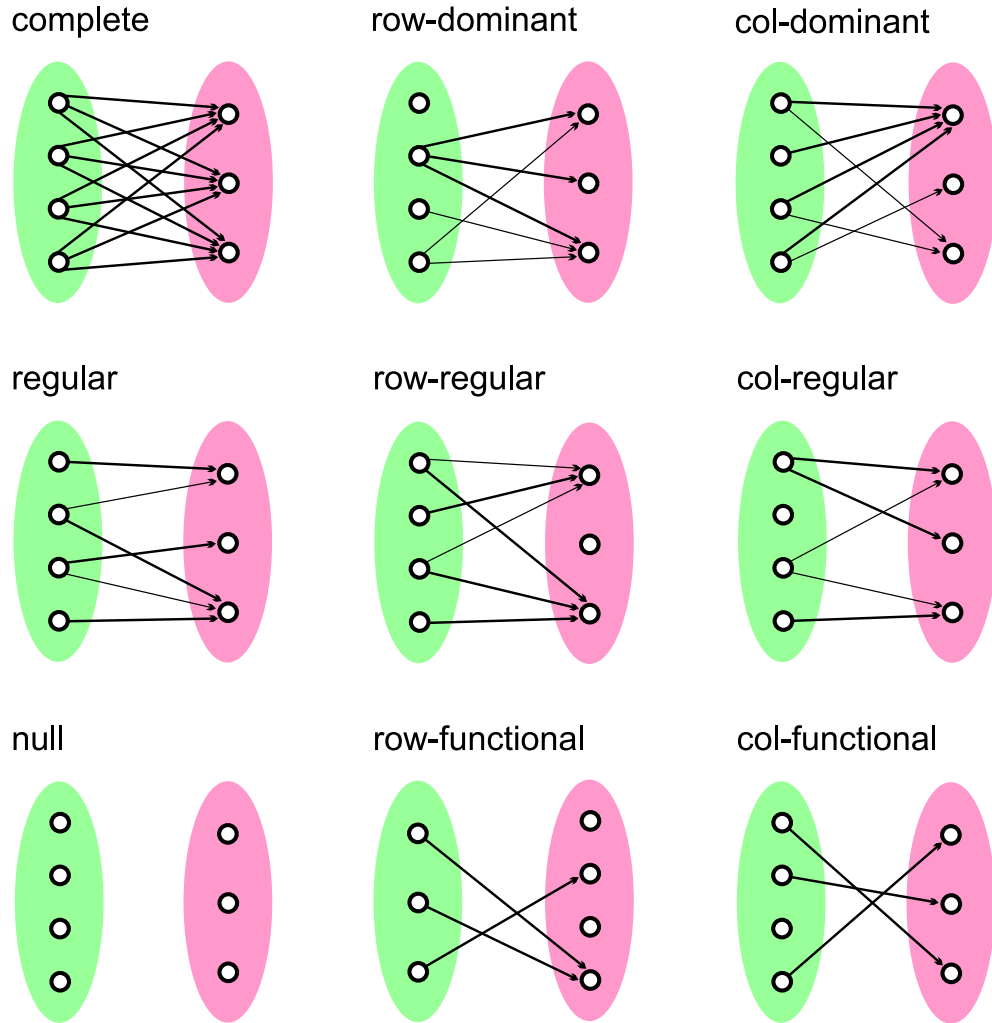
- **Large Network (Vertices)** (Büyük Ağ (Düğüm)) – Bir ağı büyük ağ yapan düğüm sayısının eşik değerini seçin. Böyle ağlar için Pajek bazı işlemlerde eski ağın silinip silinmeyeceğini sorar ve eski ağı elde edilen yenisiyle değiştirir. Böylece bellekten kazanılabilir. Bu eşik değeri aynı zamanda **Vector/Info** (Vektör / Bilgi) menüsüyle çeyreklikler hesaplanırken kullanılır (Birçok değerin eşit olabileceği büyük ağlarda bu işlem yavaş olabilir).
- **x / 0** = Sıfır olmayan bir sayı sıfıra bölündüğünde elde edilecek değer.
- **0 / 0** = Sıfır sıfıra bölündüğünde elde edilecek değer.
- **Ignore Missing Values in menu Vector and Vectors** (Vektör ve Vektörler menülerindeki eksik değerleri görmezlikten gel) – Vektör ve Vektörler menülerinde işlem yaparken veya Vector/Info (Vektör / Bilgi) komutu ile tanımlayıcı istatistikleri hesaplanırken eksik değerleri (99999997'den büyük değerleri) geçerli sayı olarak alır.
- **Use Scientific format for small/large real numbers** (Küçük veya büyük sayılar için bilimsel notasyonu kullan) – sayılar çok küçük veya çok büyük olduğunda bilimsel notasyonla yazılır, örneğin 2.4E+015 veya 2.4E-015 (2.4×10^{-15}).
- **Save Files as Unicode UTF8** (Dosyaları Unicode UTF8 Formatında Kaydet) – Pajek nesnelerini Ascii formatı yerine UTF8 formatında metin dosyası olarak kaydet, böylece özel karakterler de kaydedilebilir.
 - * **With BOM** (BOM'lu) – UTF8 dosyalarını BOM'lu veya BOM'suz kaydet.

Özel bazı dosyaları UTF8 formatında kaydetmede önemli bilgiler:

- * Pajek'in UTF8 dosyalarını okuyabilmesi için BOM'a ihtiyaç vardır. İngilizcede olmayan karakterlerden oluşan etiketlerin kullanıldığı ağlarda dosyaların UTF8 + BOM şeklinde kaydedilmesi tavsiye edilir.
- * SVG ve X3D çıktıları her zaman BOM'suz UTF8 formatında oluşturulur. (bu formatlar **xml** tabanlıdır ve format dosyadaki ilk satırda açık bir şekilde belirtilir.).
- * Raporlar her zaman UTF8 (BOM'lu veya BOM'suz) formatında kaydedilir.
- * *Ayrıla ayrılmış* dosyalar için BOM'un eklenip eklenmemesi **Tools / Delimited File Properties (Araçlar / Ayrılaçlı Dosya Özellikleri)** menüsü ile ayarlanabilir.

- **Auto Report** (Oto Raporlama) – Bütün metinsel sonuçları otomatik olarak rep1.rep dosyasına kaydet.
- **Read - Save vertices labels** (Düğüm etiketlerini Oku-Kaydet) – Oku / Kaydet ile etiketlerin yanında koordinatlar v.b. diğer düğüm özellikleri de kaydedilebilir. Düğüm etiketleri okunmazsa (ağ çok büyükse ve etiketler çok uzunsa bu zaten tavsiye edilmez) sonradan bir girdi dosyasından okunabilir. Bunun için
Network / Transform / Add / Vertex Labels / from File
(Ağ / Transforme Et / Ekle / Düğüm Etiketleri / Dosyadan) kullanılır.
- **Save coordinates of vertices** (düğümlerin koordinatlarını kaydet) – Düğümlerin koordinatlarını ağ dosyasına kaydeder veya etmez.
- **Save complete vertex description** (Düğümle ilgili bütün bilgileri kaydet) – Ağ bir dosyaya çıktı olarak kaydedilirken her bir düğüm için ilgili bütün bilgiler de kaydedilir (düğüm şekilleri, zaman aralıkları v.b. bu bilgiler aynı olsa da).
- **Check equality of vertex descriptions by reading** (Düğüm bilgilerinin eşitliğini okuyarak kontrol et) – Düğüm bilgileri açısından büyük ağ dosyalarının okunmasında kullanıcılara zaman kazandırır: Düğümlerin aynı özellikleri çok sık tekrar tekrar okunuyorsa bu seçenek bellek açısından da avantaj sağlar (örneğin düğümlerin şekilleri). Bir dosyada zamansal ağların zaman noktaları gibi farklı düğüm özelliklerinden çok fazla varsa zamandan kazanmak için bu seçeneği kullanabilirsiniz.
- **Check equality of line descriptions by reading** (Bağlantı bilgilerinin eşitliğini okuyarak kontrol et) – Bağlantı bilgileri açısından büyük ağ dosyalarının okunmasında kullanıcılara zaman kazandırır: Bağlantıların aynı özellikleri çok sık tekrar tekrar okunuyorsa bu seçenek bellek açısından da avantaj sağlar (örneğin noktalı/sürekli çizgi gibi bağlantı özellikleri). Bir dosyada zamansal ağların zaman noktaları gibi farklı bağlantı özelliklerinden çok fazla varsa zamandan kazanmak için bu seçeneği kullanabilirsiniz.
- **Max. vertices to draw** (Çizilebilecek maksimum düğüm sayısı) – Uzun süre beklemeyi önlemek için çizilebilecek maksimum düğüm sayısı burada belirlenebilir.
- **Ore: Different relations for male and female links** (Ore: Kadın ve erkek bağlantılar için farklı ilişkiler) – Bir soyağacını Ore çizgesi olarak okurken iki çeşit yay tipi oluştur: ilişki tipi 1 olan yaylar (manevi)

- baba->oğul ilişkisini (değeri de 1 olur), ilişki tipi 2 olanlar ise (manevi) anne->çocuk ilişkisini gösterir (değeri de 2 olur).
- **Ore: Generate Godparent relation** (Ore: Manevi baba ilişkisi oluştur)
 - Bir soyağacını Ore çizgesi olarak okurken manevi ebeveyn (ilişki numarası: 4) veya manevi baba (ilişki numarası: 4) ve manevi anne (relation number 5) ilişkisini de oluştur.
 - **GEDCOM - Pgraph** – pgraph formatı kullan. Soyağaçları (D. R. White) okunurken düğümler çiftler veya bireylerdir, diğer durumlarda bireylerdir).
 - **Bipartite Pgraph** (İki modlu Pgraph) – Evlilikler için kareleri, bireyler için üçgen ve çemberleri olan iki modlu bir ağ oluştur.
- **Background Colors** (Arkaplan Renkleri) – Farklı arkaplanlar ve paneller için renkleri seç.
 - **Font (Typeface, Style, Size, Color)** (Font (Tipi, Stili, Büyüklüğü ve Rengi))
 - **Select Proportional Font** (Orantılı Fontu Seç) – Unicode karakterleri göstermek için (örneğin Draw penceresinde) orantılı bir font seç.
 - **Select Monospaced Font** – Unicode karakterleri göstermek için (örneğin Report penceresinde) orantılı olmayan font bir seç.
 - **Default Fonts** (Varsayılan Fontlar) – Orantılı font olarak: *Arial Unicode MS (kalın, Maroon rengi)*. Orantılı olmayan font (eş aralıklı) olarak: *Courier New (Kalın, Maroon rengi)*.
 - **Blockmodel** (Blokmodel) – Küçültme için blokmodeli seçin. Mümkün olan modeller:
 - 0..Min Number of Links (minimum bağlantı sayısı)
 - 1..Null (boş)
 - 2..Complete (tam)
 - 3..Row-Dominant (sıra baskın)
 - 4..Col-Dominant (kolon baskın)
 - 5..Row-Regular (sıra düzenli)
 - 6..Col-Regular (kolon düzenli)
 - 7..Regular (düzenli)
 - 8..Row-Functional (sıra fonksiyonel)
 - 9..Col-Functional (kolon fonksiyonel)



Şekil 14: Genelleştirilmiş blok tipleri

- 10..Degree Density (derece yoğunluğu)

Ayrıntılı bilgi için Batagelj [7] ve Doreian, Batagelj, Ferligoj [35] makalelerine bakılabilir.

- **Menu without Icons** (İkonsuz Menü) – Menüdeki bütün ikonları kaldırır.
- **Hide Drawing Icons** (Çizim İkonlarını Gizle) – Pajek ana penceresindeki çizim ikonlarını gizler.
- **Use Old Style Dialogs (Eski Tarz Diyalog Pencereleeri Kullan)** – Windows 7 kullanıyorsanız ve dosya açma veya kaydetmede sorun yaşıyorsanız bu seçeneği işaretleyin.
- **Refresh Objects** (Nesneleri Yenile) – O anda bellekte yüklü bütün nesneleri yenile.

3.15 Info (Bilgi)

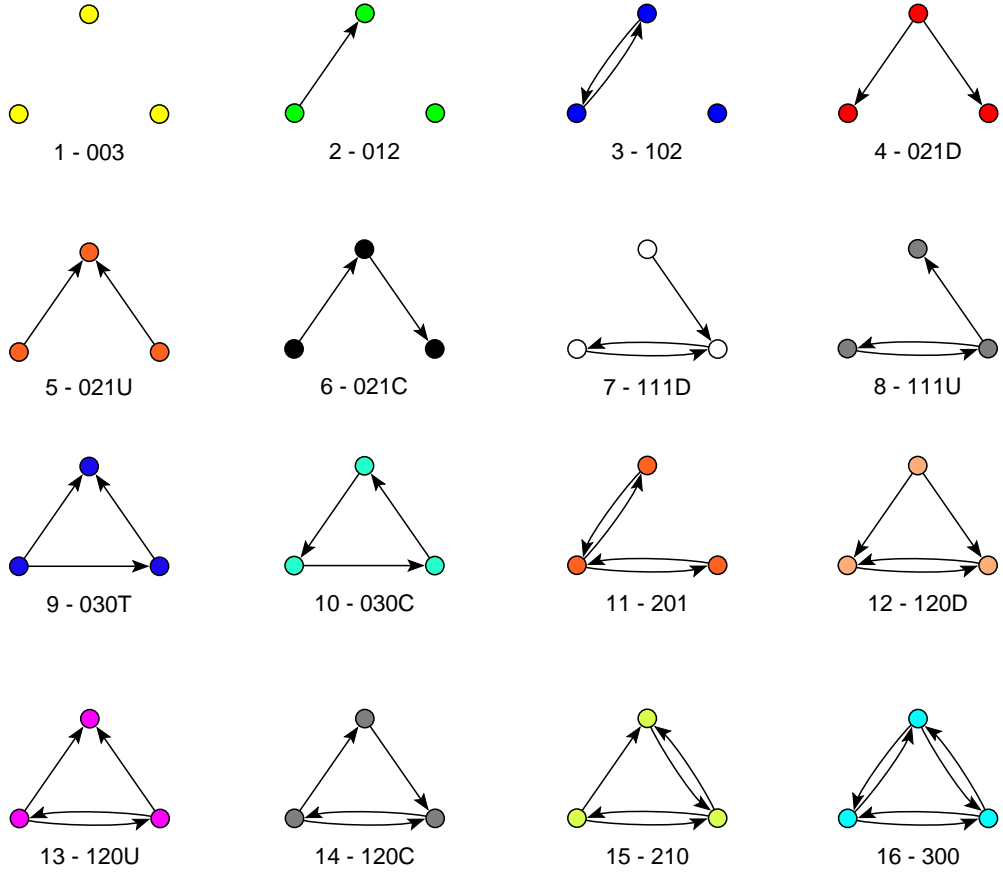
- **Child Windows** (Alt pencereler) – Pajek bilgi pencerelerini (Ağ, sınıflama, vektör, permütasyon, grup, hiyerarşi, çizim) açmak, kapatmak ve taşımak için.
- **Memory** (Bellek) – Bilgisayarda kullanılan ve kalan fiziksel bellek miktarı hakkında bilgi
- **About** (Hakkında) – Pajek versiyonu, yazarları, telif hakları vs. hakkında bilgi.

3.16 Tools (Araçlar)

- **Pajek** – PajekXXL kullanarak daha küçük bir alt ağ oluşturduktan sonra normal Pajek'i kullanmak için. Ayrıca düğüm etiketleri ve diğer düğüm bilgileri (örneğin şekiller, koordinatlar) normal Pajek'e gönderilen ağ dosyasına eklenebilir. Çıkarılan alt ağa ek olarak, karşılık gelen sınıflama ve/veya vektörler de PajekXXL'den doğrudan Pajek'e gönderilebilir).
- **R**
 - **Send to R** (R'a gönder) – Grup olarak seçilmiş bir vektör/ağ, vektörler/ağları veya bütün mevcut vektörleri ve/veya ağları istatistiksel analiz yazılımı R'da aç [58]
 - **Locate R** (R'ı Bul) – İstatistiksel analiz yazılımı R'ın yerini (Rgui.exe veya Rterm.exe dosyalarını) hard diskte bul.

- **SPSS**
 - **Send to SPSS** (SPSS'e gönder) – Grup olarak seçilmiş bir sınıflama, vektör, ağ veya sınıflamalar/vektörleri veya bütün mevcut sınıflama ve vektörleri istatistiksel analiz yazılımını SPSS'de aç.
 - **Locate SPSS** (SPSS'i Bul) – İstatistiksel analiz yazılımını SPSS'in yerini (runsyntx.exe dosyasını) hard diskte bul.
- **Excel**
 - **Send to Excel** (Excel'e Gönder) – Ağ, sınıflama ve/veya vektörleri Excel yazılımında aç.
 - **Locate Excel** (Excel'i Bul) – Excel'i (Excel.exe dosyasını) hard diskte bul.
- **Export to Delimited File** (Ayıraçlı Dosyaya Aktar) – Ağ, sınıflama ve vektörleri ayıraçlı dosyaya aktar. Oluşturulacak dosya istatistiksel analiz veya Excel gibi yazılımlarda kullanılabilir.
- **Delimited File Properties** (Ayıraçlı Dosya Özellikleri) – Oluşturulacak ayıraçlı dosyanın özelliklerini belirle (bu özellikler Excel'e aktarılırken de kullanılır).
 - **Delimiter** (Ayıraç) – Ayıraçlı veya Excel dosyasına aktarmak için gerekli ayırıcı belirle. Ayıraç *Tab* (sekme), *Semicolon* (noktalı virgül), *Comma* (virgül) or *Space* (boşluk) olabilir. *ASCII* ve *UTF8* dosyaları için farklı ayıraçlar da kullanılabilir.
 - * **ASCII – Options / Read - Write** ((Seçenekler / Oku-Yaz)'da **Save Files as Unicode UTF8 with BOM** (Dosyaları BOM'lu UTF formatında yaz) seçeneği seçili olmadığında ayırıcı seç. Bu tür dosyalar için varsayılan ayıraç *Tab* (sekme) Default delimiter for these kind of files is *Tab*.
 - * **UTF8** – Select delimiter to be used when **Save Files as Unicode UTF8 with BOM** is checked in **Options / Read - Write** (Seçenekler / Oku-Yaz). Default delimiter for these kind of files is *Semicolon*.
 - **Add BOM to UTF8 files** (UTF8 dosyasına BOM ekle) – BOM (Byte Order Mark) karakterini oluşturulan UTF8 dosyasının başına ekle
- **Web Browser** (Web tarayıcısı) – Draw (çizim) penceresinde bir düğüme *Shift* ve *sağ fare butonu* ile tıkladığında hangi Web tarayıcısının açılacağını seçebilirsiniz.

- **Inkscape** (Inkscape)– **Inkscape** yazılımının nerede kurulu olduğunu seçin (matris ve dendrogramların SVG ve PDF'e aktarılması kısmen Inkscape tarafından yapılır).
Inkscape'in hangi versiyonunun kurulu olduğu da (1.0'dan düşük, 1.0 veya 1.0'dan yüksek) bildirilmelidir.
- **Add Program** (Program ekle) – Tools (Araçlar) menüsüne çalıştırılabilir yeni bir dış programı parametreleriyle birlikte ekle.
- **Edit Parameters (Parametreleri değiştir)** – Seçili dış programın parametrelerini değiştir.
- **Remove Program** (Programı sil) – Seçili dış programı Tools (Araçlar) menüsünden çıkar



Şekil 15: Oluşturulabilecek bütün üçlü gruplar

4 Çizim Penceresi Araçları

4.1 Ana Pencere Çizim (Draw) Menüsü

- **Network (Ağ)** – Ağ çiz. Yeni bir menüsü olan yeni bir pencere açılır. Ağ elle değiştirebilirsiniz: sol fare butonu ile düğümleri hareket ettirebilirsiniz; sağ fare butonu ile bir alan çizerek resmin bir kısmını seçebilirsiniz; fare sağ butonu ile bir düğümün üstüne tıklayarak bağlantılarını değiştirebilirsiniz; X, Y, Z, S, x, y, z, s tuşları ile resmi döndürebilirsiniz.
- **Network + First Partition (Ağ + birinci sınıflama) – Draw / Network (Çiz / Ağ)**'a benzerdir. Düğümlerin renkleri seçilen sınıflamadaki sınıfları temsil eder. Ayrıca seçilen düğüm veya düğümleri farenin orta butonuna basarak (veya Shift + Sol buton, sınıfı artır) veya Alt + sol buton ile sınıf numarasını azaltarak sınıflama içindeki verilen sınıfa koyabilirsiniz (sınıflar farklı renklerle gösterilir). Sayfa 114 Şekil 20 hangi rengin hangi seçilen sınıfı temsil ettiğini gösteriyor.

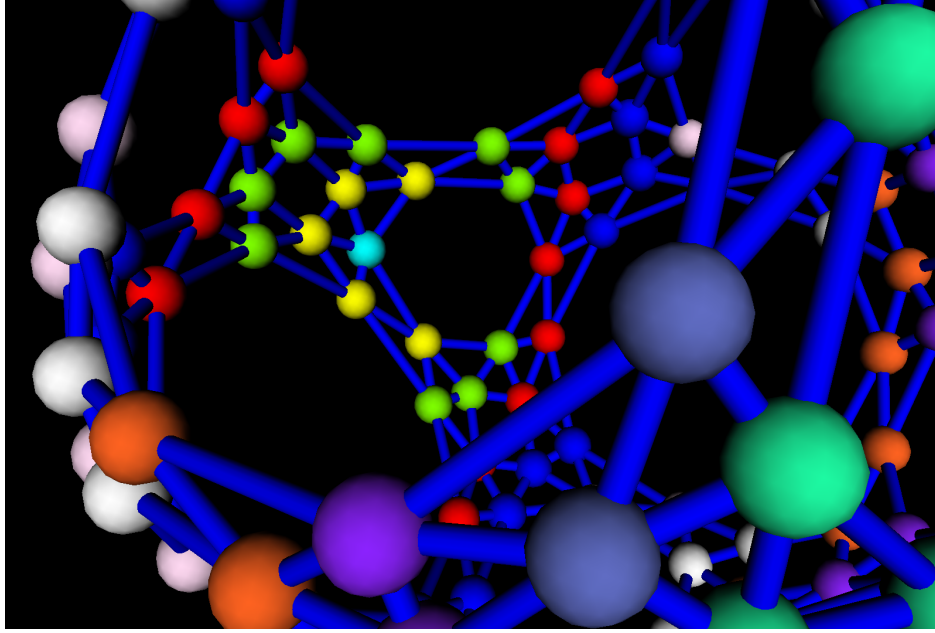
Daha önce bahsedilen bazı ekstra menü öğeleri görünür (sınıflamadaki tabakalara göre ağ çizebilirsiniz ve sınıflama ile yeri sabitlenmiş düğümleri kullanarak enerjiyi optimize edebilirsiniz). Ayrıca o sınıflamadaki bir düğümün yakınına tıklayarak seçilen sınıfı hareket ettirmeniz de mümkündür.
- **Network + First Vector (Ağ ve birinci vektör)** – Düğümlerin büyüklüğü seçilen vektörle belirlenir.
- **Network + First Vector + Second Vector (Ağ + birinci vektör + ikinci vektör)** – Düğümlerin büyüklüğü seçilen iki vektörle belirlenir (birincisi genişlik ikincisi yükseklik için).
- **Network + First Partition + First Vector (Ağ + birinci sınıflama + birinci vektör)** – Düğümlerin rengi seçilen sınıflama, büyüklükleri ise seçilen vektör tarafından belirlenir.
- **Network + First Partition + First Vector + Second Vector (Ağ + birinci sınıflama + birinci vektör + ikinci vektör)** – Düğümlerin rengi seçilen sınıflama, büyüklükleri ise seçilen iki vektörle belirlenir (birincisi genişlik ikincisi yükseklik).
- **Network + Create Null Partition (Ağ + Yeni boş sınıflama oluştur)** – Boş bir sınıflama oluştur ve bunu kullanarak ağ çiz.

4.2 Layout (Yerleşim)

Ağın yerleşimini oluştur.

- **Circular** (Çembersel) – düğümler bir çember üzerine yerleştirilir
 1. **Original** (Orijinal) – düğümler ağ dosyasında belirtildiği gibi yerleştirilir.
 2. **using Permutation** (Permütasyon kullanarak) – düğümler mevcut permütasyon tarafından belirlendiği gibi yerleştirilir.
 3. **using Partition** (Sınıflama kullanarak) – Düğümler seçili sınıflamada her bir küme için farklı bir çember üzerine yerleştirilir.
 - (a) **Different Centers** (Farklı merkezler) – Çemberin merkezi kümedeki düğümlerin koordinatlarının aritmetik ortalaması ile belirlenir
 - (b) **Concentric** (Eşmerkezli) – Bütün çemberler eş merkezlidir ve bu merkez çizim penceresinin ortasına yerleştirilir.
 4. **Random** (Rastgele) – rastgele bir sırada.
- **Energy** (Enerji) – Düğümler arasında fiziksel yaylar varmış gibi otomatik belirlenen yerleşimler.
 1. **Kamada-Kawai** (Kamada-Kawai)– Düzlemde otomatik yerleşim oluşturan bir algoritma.
 - (a) **Free (Serbest)** – Düzlemdeki her bir pozisyon kullanılabilir.
 - (b) **Separate Components** (Bileşenleri Ayır) – Her bir bileşeni ayrı bir şekilde optimize eder ve bu bileşenleri pencereye dizer.
 - (c) **Optimize inside Clusters only** (Yalnızca Kümeleri İçten Optimze Et) – Kümeler tarafından oluşturulan alt ağları optimize et (yani topluluklar)
 - (d) **Fix first and last** (İlk ve Son Düğümü Sabitle) – İlk ve son düğüm karşılıklı köşelere yerleştirilir.
 - (e) **Fix One vertex in the middle** (Ortada Bir Düğümü Sabitle) - Seçilen düğüm pencerenin ortasına yerleştirilir.
 - (f) **Selected group only** (Yalnızca Seçili Grubu) – Optimizasyon sırasında çizimin yalnızca seçilen bölümü alınır.
 - (g) **Fix selected vertices** (Seçili Düğümleri Sabitle) - Sınıflamadaki seçili düğümler verilen pozisyonlara yerleştirilir. Bu seçenek yalnızca Draw partition (Sınıflamayı Çiz) aktif olduğunda seçilebilir.

2. **Fruchterman-Reingold** – (Fruchterman-Reingold) Kamada-Kawai algoritmasına benzer fakat daha hızlı bir başka yerleşim algoritması.
 - (a) **2D** (2B) – düzlemde optimizasyon.
 - (b) **3D** (3B) – uzayda optimizasyon.
 - (c) **Factor** (Faktör) – Fruchterman-Reingold algoritmasının kullanılması durumunda düğümler arasındaki en uygun mesafenin faktörü.
 3. **Starting positions** (Başlama Pozisyonları) – enerji yerleşimleri için (rastgele, çembersel, verilen xy düzlemsel koordinatları, verilen z koordinatları).
- **Pivot MDS** (Pivot MDS)– Otomatik yerleşim algoritması (Brandes ve Pich). Bu algoritma iki ya da üç boyutlu yerleşimler oluşturabilir. Pivotlar (dayanak noktaları) rastgele seçilebilir ya da küme kullanılarak önceden belirlenebilir. Bu fiziksel yay bazlı algoritmalarından daha hızlıdır ve 100 bin seviyesinde büyük ağlar için de kullanılabilir.
 - **VOS Mapping** (VOS haritalaması) – Van Eck ve Waltman tarafından bulunan Visualization of Similarities (VOS, Benzerliklerin Görselleştirilmesi) Mapping algoritması kullanılarak yapılan yerleşimler. Bu algoritma özellikle bağlantıların değerinin benzerliği temsil ettiği durumda yoğun ağlarda bile iyi sonuçlar vermektedir. Bellek ihtiyacı düğüm sayısı 20 bini aştığı durumlarda çok yükselmekte ve ayrıca algoritma yavaşlamaktadır. Optimizasyonu kontrol etmek için iki parametre kullanılmaktadır: Optimizasyonun *tekrar başlama sayısı* (genelde bir defa tekrar yeterlidir) ve *Her Bir Yeniden Başlamada Maksimum İterasyon Sayısı* (genelde 100 iterasyon yeterli olmaktadır).
 - **EigenValues** (Özdeğerler) – Özdeğer ve özvektörler kullanılarak yapılan yerleşimler (Lanczos algoritması). Bağlantıların değerleri de istenirse dikkate alınabilir.
 1. **1 1 1** – algoritma 2 ya da 3 özdeğer seçer ve karşılık gelen özvektörleri hesaplar. Özdeğerler çoklu olabilir ve bundan dolayı olasılıklar çoktur. Bazı örnekler:
 - (a) **1 1 1** – birinci özdeğere karşılık gelen 3 özvektör bul
 - (b) **1 1 2** – birinci özdeğere karşılık gelen 2 özvektör ve ikinci özdeğere karşılık gelen bir özvektör
 - (c) **1 2 2** – birinci özdeğere karşılık gelen bir özvektör ve ikinciye karşılık gelen iki özvektör
 - (d) **1 2 3** – birinci özdeğere karşılık gelen bir özvektör, ikinciye karşılık gelen bir özvektör ve üçüncüye karşılık gelen bir özvektör



Şekil 16: Özvektörler tarafından belirlenmiş yerleşimin VRML görüntüsü

(e) $1 \ 1$ – birinci özdeğere karşılık gelen iki özvektör (iki boyutlu yerleşim için)

- **Tile Components** (Bileşenleri Döşe) – Zayıf bağlanmış bileşenleri bir düzlemde yan yana döşe

4.3 Layers (Tabakalar)

Draw Partition modu aktifken (bir ağ ve sınıflaması beraber çizildiğinde) görünür olur. Sıralamaya göre tabakaları çizer.

- **Type of Layout** (Yerleşim Türü) – Resmin tipini seç (İki boyutlu – tabakalar y yönünde, veya üç boyutlu – tabakalar z yönünde). Buna göre uygun menü görünür olur.
- **In y direction** (y yönünde) – Dğümleri tabakalar (y koordinatı) halinde çiz, tabakaların içinde dğümleri merkeze eşit uzaklıkla (x koordinatı) çiz, z koordinatı bütün dğümler için 0.5'dir.
- **In y direction+random in x** (y yönünde + x yönünde rastgele) – Birinci seçenikle aynı, yalnız dğümler tabakalara rastgele bir şekilde konur (vertex numaralarına göre değil).

- **In z direction** (z yönünde) – Tabakaları z yönünde çiz, x ve y koordinatları olduğu gibi kalsın.
- **In z direction + random in xy** (z yönünde + xy yönünde rastgele) – Tabakaları z yönünde çiz, x ve y koordinatları rastgele değer alır.
- **Averaging x coordinate** (x koordinatını ortalama) – Düğümler iki boyutlu tabakalara konduktan sonra kullanın. Özyinelemeli bir şekilde bütün komşuların ortalama x koordinatını hesaplar ve normalize eder. Global resmin iyi bir yaklaşık görüntüsünü verir fakat düğümler birbirine çok yakın konur. Bütün düğümlerde kullanın veya sadece bir tane seçili olan üzerinde.
- **Averaging x and y coordinates** (x ve y koordinatlarında ortalama) – Düğümler üç boyutlu tabakalara konduktan sonra kullanın. Özyinelemeli bir şekilde bütün komşuların ortalama x ve y koordinatlarını hesaplar ve normalize eder. Global resmin iyi bir yaklaşık görüntüsünü verir fakat düğümler birbirine çok yakın konur. Bütün düğümlerde kullanın veya sadece bir tane seçili olan üzerinde.
- **Tile in x direction** (x yönünde döşe) – x koordinatı ortalamasından sonra düğümler birbirine çok yakında bulunur, bundan dolayı bu seçenek kullanılarak düğümler resolution (çözünürlük) parametresi ile tanımlanan minimum mesafeye yeniden konumlandırılır.
- **Tile in xy plane** (xy yönünde döşe) – Bir önceki seçenekle aynıdır fakat üç boyutlu resim durumunda kullanılır.
- **Optimize layers in x direction** (x yönünde tabakaları optimize et) – Bağlantıların toplam uzunluğunu minimize ederek tabakalı yerleşimi optimize et.
 1. Forward (İleri) – İlk tabakadan son tabakaya doğru git. Mevcut tabakada mevcut tabaka numarasına veya bir eksikliğine sahip tabakaları optimize et.
 2. Backward (geri) – Son tabakadan ilk tabakaya doğru git. Mevcut tabakada mevcut tabaka numarasına veya bir fazlasına sahip tabakaları optimize et.
 3. Complete (tam) – İlk tabakadan son tabakaya doğru git. Mevcut tabakada mevcut tabaka numarasına, bir fazlasına veya bir eksikliğine sahip tabakaları optimize et.
- **Optimize layers in xy plane** (xy düzleminde tabakaları optimize et) – Bir önceki seçenek ile aynıdır fakat üç boyutlu resim için kullanılır.

- **Resolution** (Çözünürlük) – Tabakalarda kaç tane ekstra pozisyon uygundur. Yalnızca Pgrah için kullanılır. Çözünürlük ne kadar yüksekse optimizasyon sonucu o kadar iyi olur fakat uzun sürer.

4.4 GraphOnly (Sadece Ağ)

Mevcut ağı etiket ve oklar olmadan tam olarak göster.

4.5 Default (Varsayılan)

Varsayılan ayarlarla mevcut ağı tam olarak tekrar çiz: düğümler, bağlantılar ve oklar varsayılan büyüklükte; düğüm etiketleri görünür; bağlantıların etiket ve değerleri görünmez; bağlantıların farklı kalınlıkta ve gri olarak çizilmesi pasif.

4.6 Previous (Önceki)

Pa jek’de yüklü bir önceki ağı ve/veya sınıflamayı ve/veya vektörü çiz (Options/PreviousNext/Apply to (Seçenekler/ÖncekiSonraki/Uygula) seçeneğine bağlı olarak)

4.7 Redraw (Tekrar Çiz)

Ağı tekrar çiz.

4.8 Next (Sonraki)

Pa jek’de yüklü olan sonraki ağı, ve/veya sınıflamayı ve/veya vektörü çiz (Options/ PreviousNext/Apply to seçeneğine bağlı olarak)

4.9 ZoomOut (Küçült)

Çizimi küçült (yerleşim üzerinde büyütme yaparken aktiftir).

4.10 Options (Seçenekler)

Yerleşimin görüntüsü için ekstra parametreler.

- **Transform** (Transforme et) – Görselin farklı transformasyonları için.

1. **Fit area** (alana sığdır)

- (a) **max(x), max(y), max(z)** – düğüm koordinatlarını bağımsız bir şekilde artırarak ağ görselinin penceredeki bütün alanı kullanmasını sağlar.
 - (b) **max(x,y,z)** – Orantıyı koruyarak koordinatları değiştirir ve görselin bütün çizim alanını kapsamasını sağlar (üç koordinattaki mesafelerin en büyüğüyle orantılı büyütme yapılır).
2. **Resize** (Yeniden boyutlandır) – Ağ görselinin tamamında veya seçili bir kısımdaki düğümlerin koordinatlarını verilen bir faktörle çarpar.
 3. **Translate** (Çevir) – Görseli veya seçili kısmını uzayda çevirir.
 4. **Reflect y axis** (y eksenine göre yansıma) – görselin veya seçili kısmının y eksenini etrafında simetriğini al.
 5. **Set z-coordinate to 0.5** (z koordinatını 0,5 yap) – yerleşimi iki boyuta (düzleme) çevir.
 6. **Rotate 2D** (2B’de Döndür) – Görseli veya seçili kısmını xy düzleminde döndür.
 7. **FishEye** (Balık Gözü) – Balık gözü transformasyonu (kartezyen veya polar). Eğer bir düğüm seçilmemişse pencerenin orta noktası odak noktası olarak kullanılarak düğüm koordinatları balık gözü transformasyonuna tabi tutulur. Yoksa seçilen düğümün pozisyonu odak noktası olarak alınır.
 8. **Resize Clusters Area** (Küme Bölgesini Yeniden Boyutlandır) – Kümeleri kendi merkezleri etrafında verilen faktörle yeniden boyutlandır.
- **Values of lines** (Bağlantıların değeri) – Enerji bazlı veya özvektörlü görselleştirmelerde bağlantıların değerlerinin anlamı: no meaning (anlamı yok), similarities (benzerlik), dissimilarities (benzemezlik)
 - **Mark Vertices Using** (Düğümleri İşaretle) – İşaretleme aşağıdakileri kullanarak yapılabilir:
 1. **Labels** (Etiketler)
 2. **Numbers** (Numaralar)
 3. **No Labels** (Etiketsiz)
 4. **No Labels no Arrows** (Etiketsiz ve Oksuz)
 5. **Mark Cluster Only** (Yalnızca kümeyi işaretle) – Yalnızca mevcut kümeye ait düğümler görselde etiketlenir.
 6. **Vector Values** (Vektör Değerleri) – aynı boyuta sahip bir veya iki vektör seçilmişse düğümler ekstra olarak bir veya iki vektörle işaretlenir.

7. **Clusters of Second Partition** (İkinci Sınıflamanın Kümeleri) – Aynı boyuta sahip ikinci bir sınıflama seçilmişse düğümler ekstra olarak küme numaraları ile işaretlenir.
 8. **Cluster Symbols of Second Partition** (İkinci Sınıflamanın Küme Sembolleri) – Aynı boyuta sahip ikinci bir sınıflama seçilmişse düğümler ekstra olarak küme sembolü ile işaretlenir. Yerleşim SVG veya EPS'e aktarıldığında semboller düğümün ortasına konur (EPS'de sadece ASCII semboller kullanılabilir).
 9. **Labels as Tooltips** (İpucu olarak Etiketler) – Düğüm etiketleri (ve ekstra özellikler) fare ile düğüm üzerine gelindiğinde gösterilir (düğüme yakın bir yazı olarak değil).
 10. **Labels Centered** (Etiketler Ortalanmış) – Düğüm etiketleri düğüm sembollerinin üzerine ortalanmış.
- **Lines** (Bağlantılar) – Bağlantıların nasıl çizileceğini seçin:
 1. **Draw Lines** (Bağlantıları Çiz)
 - (a) **Edges** (Kenarlar) – Kenarları çiz veya çizme
 - (b) **Arcs** (Yaylar) – Yayları çiz veya çizme
 - (c) **Relations** (İlişkiler) – bütün bağlantıları (boş karakter dizisi bırak) veya sadece seçili ilişkilere sahip bağlantıları çiz, örneğin 1-3,6,10-15.
 2. **Mark Lines** (Bağlantıları İşaretle)
 - (a) **No** (Hayır) – Bağlantıları işaretleme
 - (b) **with Labels** (Etiketlerle)
 - (c) **with Values** (Değerleriyle)
 3. **Different Widths** (Farklı Kalınlıkta) – İşaretlenirse bağlantıların kalınlığı değeri tarafından belirlenir.
 4. **GreyScale** (Gri Tonlar) – İşaretlenirse bağlantıların rengi gri tonlarda değeri tarafından belirlenir.
 - **Size** (Boyut) – Düğümlerin büyüklüğünü, düğüm çizgilerinin kalınlığını, bağlantıların kalınlığını, okların büyüklüğünü, sembollerin büyüklüğünü ve fontunu belirle veya varsayılan değerlerine çevir.
Düğümlerin büyüklüğü otomatik olarak da belirlenebilir (0-ortalama). Ayrıca bir girdi dosyasından da (x_fact ve y_fact) şeklinde okunabilir. Bir diğer yol ise seçili bir veya iki vektör tarafından belirlenmesidir. FontSize (font büyüklüğü) üçüncü sınıflamadaki değerlerle orantılı olarak belirlenebilir.

- **Colors (Renkler)** – Arkaplanın, düğümlerin, düğümlerin çizgilerinin, kenarların, yayların ve fontun (düğüm ve bağlantı etiketlerinde kullanılan) rengini belirle veya varsayılan değerlerine çevir. Sembollerin rengi üçüncü sınıflama tarafından belirlenebilir. Kenar ve yayların rengi bağlantının ilişki numarasını temsil edebilir. Verilen sınıfı hangi rengin temsil edeceği de **Relation Colors** (İlişki renkleri) ile seçilebilir. Ayrıca girdi dosyasında düğümlerin iç rengi (örneğin **ic Red**) ve çizgisinin rengi (örneğin **bc Blue**), yay ve kenarların rengi (örneğin **c Green**) belirlenebilir (Sayfa 113, Şekil 19). Font-Color (font rengi) ikinci sınıflamadaki değerlerle belirlenebilir. Önceden tanımlanmış renklere ek olarak RGB (örneğin RGBFF0000 veya RGB(1,0,0)) veya CMYK formatını (e.g. CMYK00FF0000, or CMYK(0,1,0,0)) kullanan özel renkler de elde edilebilir. Renk tanımlarında boşluk olmamasına dikkat edilmelidir.

Örnek NET dosyası:

```
*Vertices          9
 1 "a" ic Pink          bc Black
 2 "b" ic CMYK(0,0,1,0.0) bc CMYKFF000000
 3 "c" ic Cyan          bc Yellow
 4 "d" ic Purple        bc Orange
 5 "e" ic Orange        bc Brown
 6 "f" ic Magenta       bc Green
 7 "g" ic Brown         bc Magenta
 8 "h" ic RGB(1,0,0)    bc Blue
 9 "i" ic Green         bc Magenta

*Arcs
 1          2          1 c RGB0000FF
 2          3          1 c Red
 3          4          1 c Black
 4          5          1 c Yellow
 5          6          1 c Gray
 6          7          1 c Cyan
 7          8          1 c Magenta
 8          9          1 c Purple
 9          1          1 c Brown
```

- **Symbols for Partition Clusters** (Sınıflama Kümeleri için Semboller) – Sınıflama kümelerine sembol ataması yap. Semboller Unicode karakterleri olabilir.
- **Layout (Yerleşim)** – Yerleşim seçenekleri

1. **Redraw** (Tekrar Çiz) – Mevcut ağı yeniden çiz:

- **during Vertex Moving** (Düğümler Hareket Ederken)
- **after Vertex Moving** (Düğümler Hareket Ettikten Sonra)
- **on Layout Paint** (Yerleşim Tekrar Yapıldığında)

- **on Layout Resized** (Yerleşimin Büyüklüğü Değiştirildiğinde) – Draw (çizim) penceresinin büyüklüğü değiştirildiğinde.
- **Rendering Refresh Period (Tekrar Çizme Yineleme Süresi)** – (büyük) bir ağ çizilirken çizimin yenilenme süresi. Bu durumda ağ çizilirken ayrıca **Wait** (bekle) bildirim sağ üst köşede görünür. Fakat 0 seçilirse çizim yenilenmez ve bildirim de görünmez. Bazı seçeneklerin yalnızca yenilemenin seçilmemesi durumunda etkin olacağı akıldan çıkarılmamalıdır (örneğin yerleşim 'x', 'y', 'z', or 's' tuşlarına basarak döndürülürken veya kaydırma çubuğu kullanılırken).
- 2. **Real xy proportions** (Gerçek xy oranları) – Çizim penceresinin şekli hep kare şeklindedir veya değildir.
- 3. **Arrows in the Middle** (Oklar Ortada) – Okları bitiş düğümlerine değil bağlantı çizgilerinin ortasına çiz.
- 4. **Size of vertex 0** (0 Boyutlu Düğümün Büyüklüğü) – Büyüklüğü 0 olan düğümler nasıl gösterilecek? (düğümün büyüklüğü girdi dosyasında veya bir vektör tarafından belirlenebilir)
 - (a) **Hide vertex (Düğümü Gizle)** – büyüklüğü 0 olan düğümler gösterilsin mi gösterilmesin mi?
 - (b) **Hide attached lines (Bağlantıları Gizle)** – bir ucu sıfır büyüklükte düğüm olan bağlantıları göster veya gösterme.
- 5. **Decimal Places** (Ondalık Basamak Sayısı) – Düğümlerin vektörlerdeki değerlerle işaretlenmesi durumunda ondalık basamak sayısı.
- 6. **Show SubLabel** (Alt Etiketleri Göster) – Ağ yerleşiminde gösterilen düğüm alt etiketlerinin pozisyonunu seç.
- 7. **Labels with Transparent Background** (Transparan Arkaplanlı Etiketler) – Transparan arkaplanlı etiketlerin gösterilmesini seç.
- **ScrollBar On/Off** (Kaydırma Çubuğu Açık/Kapalı) – Çizim penceresinin üst sol köşesindeki kaydırma çubuklarını göster/gizle. Çizimin (resmin) bir kısmı seçildiğinde kaydırma çubuğu çizimi hareket ettirmek için kullanılır. Bütün çizim seçildiğinde kaydırma çubuğu döndürmek için kullanılır (X, Y, Z, x, y, z tuşlarına basmak gibi – tuş tarafından belirlenen eksenin etrafında döndürmek için veya S – seçili normal etrafında döndürme).
- **Interrupt** (Kesme) – Optimizasyon sırasında kesme süresi (her ? saniyede dur veya durma).

- **Previous/Next** (Önceki/Sonraki) – Çizim penceresindeki ağ dizisinin çizmek için Previous (Önceki) ve Next (sonraki) komutlarının kullanımı için parametreleri seç:
 1. **Max. number** (Maksimum Sayı) – Dizide kaç adet ağ gösterilecek. Bu numara mevcut ağ sayısından yüksek ise dizi erken biter.
 2. **Seconds to wait** (Saniye Olarak Bekleme Süresi) – İki yerleşim arasında saniye olarak bekleme süresi.
 3. **Optimize Layouts** (Yerleşimleri Optimize Et) – Mevcut yerleşimi optimize et veya etme. Ağ dizisi aynı ağdan elde edildiyse Energy / Starting Positions / Given xy (Enerji / Başlama Pozisyonları / Verilen xy) seçeneği ile optimizasyonu mevcut koordinatlardan başlatmak uygun olur.
 - (a) **Kamada-Kawai** (Kamada-Kawai) – Mevcut yerleşimi Kamada-Kawai algoritması ile optimize et.
 - (b) **2D Frucht. Rein.** (İki Boyutlu Fruchterman-Reingold) – Mevcut yerleşimi iki boyutlu Fruchterman-Reingold algoritması ile optimize et.
 - (c) **3D Frucht. Rein.** (Üç Boyutlu Fruchterman-Reingold) – Mevcut yerleşimi üç boyutlu Fruchterman-Reingold algoritması ile optimize et
 - (d) **No** (Hayır) – Yerleşimi optimize etme sadece resmi göster.
 4. **Apply to** (Uygula) – Önceki veya Sonraki seçildiğinde hangi nesnenin (ağ, sınıflama, vektör) de değişeceğini belirleme:
 - (a) **Network** (Ağ) – Bellekteki önceki / sonraki ağ çizilir. Eğer Draw / Network+Partition seçilmişse ve yeni ağ seçili sınıflama ile aynı boyuta sahipse bu sınıflama ağdaki düğümlerin rengini belirler. Eğer Draw/Network+Vector seçilmişse ve yeni ağ seçili vektörle aynı boyuta sahipse bu vektör ağdaki düğümlerin büyüklüğünü belirler.
Aynı boyutlu birçok ağı aynı sınıflama/vektörle gösterme seçeneği kullanılabilir.
 - (b) **Partition (Sınıflama)** – Bellekteki önceki / sonraki sınıflama kullanılır. Eğer Draw / Network+-Partition seçilmişse aynı ağ önceki / sonraki sınıflama ile çizilir (ağ ve sınıflama aynı boyuta sahip olmalıdır).
Seçili ağın birçok sınıflamasını gösterme seçeneği kullanılabilir.
 - (c) **Vector** (Vektör) – Bellekteki önceki / sonraki vektör seçilir. Eğer Draw / Network+Vector seçilmişse aynı ağ önceki / sonraki vektör

kullanılarak çizilir (ağ ve vektör aynı boyuta sahip olmalıdır).
Seçili ağın birçok vektörünü gösterme seçeneği kullanılabilir.

Birçok nesneyi (ağ, sınıflama ve vektör) aynı anda seçerek önceki / sonraki ağlar önceki / sonraki sınıflama veya vektörlerle aynı anda çizilecektir. Bütün seçili nesleler aynı boyuta sahip olmalıdır.

4.11 Export (Aktar)

Ağın yerleşimini aşağıdaki iki veya üç boyutlu formatlardan birine aktar:

- **2D** – iki boyutlu aktarmalar:
 - **EPS/PS (EPS/PS)**– Klipli veya klipsiz, WYSIWYG (What you see is what you get, neyi görüyorsan onu elde edersin) özellikli EPS (encapsulated postscript) formatına aktar – aktarılmış EPS resmi çizim ekranındaki benzer olacaktır – yalnız renkler siyah-beyaz olur ve renk sınıflama tarafından belirlenir. **PS (PS)** – PS (postscript) formatına aktar (EPS’e benzerdir fakat sınırı yoktur).
 - **SVG (SVG)**– SVG (Scalable Vector Graphics, ölçeklenebilir vektör grafiği) formatına aktar. SVG ve HTML’e aktarmada ekstra bazı yerleşim kontrol parametreleri de kullanılabilir. Doğrusal veya radyal gradyant (renğin bir renkten başka bir renge değişirken sürekli bir şekilde değişmesi) gibi özellikler de seçilebilir – Export / Options (Aktar / Seçenekler) penceresinden üçe kadar renk seçilebilir.
 1. **General (Genel)** – Resmin parçalarını seçme şansı olmadan SVG formatına aktar.
 2. **Labels/Arcs/Edges (Etiketler/Yaylar/Kenarlar)** – Etiketleri, yay ve kenarları değiştirme imkanı olsun/olmasın.
 3. **Partition (Sınıflama)** – Bir veya iki sınıflamayı kullanarak SVG’ye aktar. Bir ağ bir sınıflama kullanılarak ve ikinci bir sınıflama seçilmeden çizilirse ilk sınıflama kümeleri ve renkleri belirler. Fakat eğer iki sınıflama seçili ve ikinci sınıflama sembol ve etiket renkleri için kullanılmamışsa o zaman birinci sınıflama renkleri ikinci sınıflama kümeleri belirler.
 - (a) **Classes (Sınıflar)** – Kullanıcı seçili sınıfları ve aralarındaki bağlantıları açıp kapatabilir.
 - (b) **Classes with semi-lines (Yarı bağlantılı sınıflar)** – Kullanıcı sınıfları açıp kapatabilir. Sınıflar arasındaki bağlantılar yarı çizgi olarak çizilir.

- (c) **Nested Classes** (İç İç Sınıflar) – Üst sınıflar alt sınıflarda iç içedir – ne zaman bir sınıf açılırsa daha yüksek bütün sınıflar da açılır ve alt sınıflar kapatılır (bu örneğin çekirdekleri göstermek için uygundur).
4. **Line Values** (Bağlantı Değerleri) – Bağlantı değerlerini de kullanılarak SVG'ye aktar. Eşik değeri ve sınıf sayısı verilmelidir. #n girilirse n tane eşit boyutta sınıf oluşturulur. Elde edilen eşik değerlere göre bağlantıların alt kümesi (ve bağlı olduğu düğümler) tanımlanır ve web uygulamasında açılıp kapanabilir.
- (a) **Classes** (Sınıflar) – Kullanıcı seçilen değerde bağlantıları ve bağlı olan düğümleri açıp kapatabilir.
- (b) **Nested classes** (İç İç Sınıflar) – Kullanıcı seçili değerdeki veya daha yüksek değerli bağlantıları ve bağlı olan düğümleri açıp kapatabilir.
- (c) **Options** (Seçenekler) – Bazı görsel özellikleri kullanarak bağlantıların değerini vurgulamak için bazı ekstra seçenekler. **Different Colors** (Farklı Renkler) – Düğümleri sınıflamak için kullanılan renkleri kullanarak bağlantıların alt kümesi çizilebilir. **Using GreyScale** (Gri Tonları Kullan) – Bir bağlantı çizgisinin koyuluğu değerini gösterir (orantılıdır). **Different Widths** (Farklı Kalınlıklar) – Bir bağlantı çizgisinin kalınlığı değerini gösterir (orantılıdır).
5. **Multiple Relations Network** (Çoklu İlişkili Ağ) – Çoklu ilişki ağını SVG'ye aktar. Kullanıcı yerleşimde bir veya daha fazla ilişkiyi gösterip gizleyebilir.
6. **Current and all Subsequent** (Mevcut ve Bütün Sonrakiler) – İşaretliyse mevcut ağ ve sonraki diğer bütün ağlar SVG'ye aktarılır. Her bir ağ için ayrı bir html dosyası oluşturulur. Dosyaları aşağıdaki gibi adlandırılır: file0001.htm, file0002.htm, ..., file9999.htm. Oluşturulan html dosyalarına ekstradan ağlar arasında geçiş yapmak için Önceki / Sonraki linkleri de konur. Sonraki sınıflama ve vektörler boyut açısından ağlara uyumluysa sınıflamalar düğümlerin rengini, vektörler ise büyüklüğünü belirler. *Subsequent* (sonra gelen) özelliği çizim penceresindeki *Options / Previous / Next / Apply to* (Seçenekler / Önceki / Sonraki / Uygula) seçeneğine göre ağ, sınıflama ve vektör kombinasyonlarına uygulanabilir (üçünden biri, ikisi veya üçü de).
7. **Create PDF also** (PDF de Oluştur) – Ne zaman bir yerleşim SVG'ye aktarılırsa aynı zamandan bir PDF dosyasına da aktarılır.

- **JPEG** (JPEG) – JPEG/JPG formatına aktar. Sıkıştırma kalitesi (0-100) ve gri tonlar seçilebilir.
 - **Bitmap** (Bitmap) – Windows bitmap (bmp) formatına aktar.
 - **VOSviewer** (VOSviewer)– VOSviewer yazılımını (1.5.2 versiyonu veya daha üstü) bir Pajek ağı, sınıflaması ve/veya vektörü ile çalıştır.
 - **Copy to Clipboard** (Panoya Kopyala) – Draw penceresindeki çizimi panoya (clipboard) aktar.
- **3D** (3B) – üç boyutlu aktarmalar:
 - **X3D** (X3D) – X3D formatına aktar. X3D XML tabanlı bir üç boyutlu bilgisayar grafiği formatıdır ve VRML teknolojisinin devamıdır. X3D formatı üç boyutlu yazdırma için <https://www.shapeways.com/> adresine gönderilebilir.
 - **Kinemages** (Kinemages) – Toplar ve etiketlerle beraber Kinemages formatına aktar. Çıktıyı seyretmek için Mage veya King görüntüleme yazılımını kullanabilirsiniz. Mage yazılımı ücretsiz olarak indirilebilir: [59].
 1. **Current Network Only** (Sadece Mevcut Ağ) – Sadece mevcut ağ Kinemages’e aktar. Partitions (sınıflamalar) menüsü ile tanımlanmış iki sınıflama kullanılabilir. Birisi nesiller diğeri renkler için.
 2. **Current and all Subsequent** (Mevcut ve Bütün Sonrakiler) – Mevcut ağ ve sonraki bütün ağları aktar (KINEMAGE/Next veya Mage’de Ctrl N komutunu kullanın). Sonraki sınıflamalar ve vektörler boyut olarak ağlara uygunsuz sınıflamalar düğümlerin rengini, vektör ise büyüklüğünü belirlemede kullanılır. *Subsequent* (sonra gelen) özelliği çizim penceresindeki *Options / Previous / Next / Apply to* (Seçenekler / Önceki / Sonraki / Uygula) seçeneğine göre ağ, sınıflama ve vektör kombinasyonlarına uygulanabilir (üçünden biri, ikisi veya üçü de).
 3. **Multiple Relations Network** (Çoklu İlişkili Ağ) – Seçili ilişkileri gösterme veya gizleme seçeneği ile aktar.
 - **VRML** (VRML) – VRML formatına aktar. Virtual Reality (sanal gerçeklik) formatı. Çıkan sonucu incelemek için VRML göstericisi yazılıma ihtiyacınız olacaktır: Cortona [29] veya (daha eski) Cosmo player [30].
 - **MDL MOL file** (MDL MOL Dosyası) – MDL MOL dosyası formatına aktar. Çıktıyı görüntülemek için Netscape’de Chime eklentisine (Chemscape Chime) ihtiyacınız var [52].

- **Options** (Seçenekler) – EPS, SVG, X3D ve VRML'e aktarma için var-sayılan seçenekler (bkz. **Exports to EPS / SVG / X3D / VRML**).
- **Append to Pajek Project File** (Pajek Proje Dosyasına Ekle) – Mevcut ağı seçili Pajek proje dosyasının sonuna ekle (proje dosyası **PajekToSvAnim** programına aktarılmaya uygun bir şekilde formatlanır).
 - **Select file** (Dosyayı Seç) – Proje dosyasını seç.
 - **Append** (Ekle) – Seçilmiş proje dosyasına ekle.

Önemli not: **Macro / Repeat Last Command** (Makro / Son Komutu Tekrar Et) komutu ağ dizisini seçilen proje dosyasına göndermek için kullanılabilir (ağları elle tek tek eklemek yerine)

4.12 Spin (Döndür)

- **Spin around** (Döndür) – Seçili normal etrafında ağı döndür.
- **Perspective** (Perpspektif) – Uzak düğümler daha küçük (büyük) çizilir..
- **Normal** (Normal) – Etrafında döndürülecek normal vektörünü belirle.
- **Step in degrees** (Derece Olarak Adım) – Döndürmenin gösterilmesinde adım olarak kullanılacak açının derece olarak değeri.

4.13 Move (Kaydır)

Elle düğüm hareket ettirilmesinde ekstra kısıtlılıklar.

- **Fix** (Sabitle) – x ve y yönünde sabitle ve hareketi engelle veya merkezden (dolaşan) mesafenin değiştirilmesine izin verme.
- **Grid** (Izgara) – Izgara üzerinde (x, y) pozisyonları tanımla. Elle düğümlerin hareket ettirilmesinde bu noktalar rehber görevi görürler.
- **Circle** (Çember) – Eşmerkezli çemberler üzerinde (x, y) pziyonları tanımla. Elle düğümlerin hareket ettirilmesinde bu noktalar rehber görevi görürler.
- **Grasp** (Tut) – Verilen bir sınıfta bir düğüme yakın bir noktaya sol fare tuşuyla tıklanıldığında hangi ekstra düğümlerin hareket edeceğini belirle. Hareket edecek düğümler:
 1. **Closest Class Only** (Yalnızca en yakın sınıfta)
 2. **Closest Class and Higher** (En yakın sınıf ve üstünde)
 3. **Closest Class and Lower** (En yakın sınıf ve altında) olabilir.

4.14 Info (Bilgi)

Mevcut yerleşimin hesaplanacak estetik özelliklerini seç:

- Closest vertices (En yakın düğümler)
- Smallest angle (En küçük açı)
- Shortest/Longest line (En kıs/en uzun bağlantı)
- Number of crossings of lines (kesişen bağlantı sayısı)
- Vertex closest to line (bağlantıya en yakın düğüm)
- All properties (bütün özellikler)
- Çizim penceresindeki düğümlerin arasındaki mesafeler ile jeodezik mesafeleri arasındaki korelasyon

4.15 FishEye (Balık Gözü)

Çizim penceresini balık gözü moduna alır veya bu moddan çıkarır. Pencerede fareni gezdirdiğinizde belirli bölgeleri büyütebilirsiniz. Elde edilen yerleşimi korumak isterseniz Draw penceresinde başka herhangi bir yere tıklayabilirsiniz.

- **Cartesian** (Kartezyen) – Kartezyen balık gözü transformasyonu kullan.
- **Polar** (Polar) – polar balık gözü transformasyonunu kullan.
- **Factor** (Faktör) – kaydırma miktarını seçer.
- **Exit** (Çıkış) – balık gözü modundan çıkar ve eski koordinatlara döner.

5 EPS/SVG/X3D/VRML'e Aktarma

5.1 Varsayılan parametreler

Çizim (Draw) penceresinde etiketler veya sayılar görünmüyorsa EPS/SVG'ye aktarıldığında da görünmeyecektir. Çizim/Sınıflama (Draw/Partition) görseline bakılıyorsa renkler EPS/SVG resmine de aynı şekilde otomatik olarak aktarılacaktır.

5.2 EPS, SVG, X3D ve VRML için Varsayılan Parametreler Penceresi

Bu pencere ikisi solda üçü sağda olmak üzere 5 çerçeveye bölünmüştür. Pajek ağ dosyasında parametreler tanımlanmaışsa bu penceredeki parametrelere önceliği olacaktır.

Sol Üst Çerçeve – EPS/SVG Vertex Default (EPS/SVG düğüm varsayılanları)

Bu çerçeve yerleşimin EPS veya SVG formatına aktarımında düğümlerin çizilmesiyle ilgili varsayılan parametreleri içerir.

- **Interior Color** (İç renk) – Düğümlerin iç rengi. Renk seçenekleri için bkz.: sayfa 113), Şekil 19. Sınıflama kullanılarak ağ çizilmişse burada verilen renkler yerine sınıflama renkleri kullanılır.
- **x/y Ratio** (x/y oranı) – düğümün x ve y yönlerindeki boyutlarının oranı (örneğin bu değer 1 ise şekil çember, daha büyük/daha küçük olmasında düğümler elips şeklinde olurlar).
- **Symbol Size** (Sembol Büyüklüğü) – düğümün ortasında gösterilen sembolün göreceli veya mutlak büyüklüğü. Sembol büyüklüğü pozitif bir sayı olarak girilirse (örneğin 0.75) bu sayı düğüm büyüklüğüne göredir: Sembolün son büyüklüğü düğüm büyüklüğü ve varsayılan sembol büyüklüğü kullanılarak hesaplanır. Böylece farklı büyüklükteki düğümlerin farklı büyüklükte sembolleri olur. Sembol büyüklüğü negatif bir sayı olarak girilirse (örneğin -25) bu mutlak bir değer olarak alınır ve sembol büyüklüğü de bu olur. Böylece farklı büyüklükteki düğümler aynı büyüklüğe sahip olurlar.
- **Border Color** (Sınır Rengi) – düğümlerde kullanılan çizgilerin rengi.
- **Border Width** (Sınır Genişliği) – düğümlerde kullanılan çizgilerin kalınlığı.
- **Shape Angle** (Şekil Açısı) – şeklin derece olarak dönme miktarı.

- **Label Color** (Etiket Rengi) – düğümlerin etiketinin rengi.
- **Font Size** (Etiket büyüklüğü) – düğümlerin etiketlerinin font büyüklüğü.
- **Label Angle** (Etiket Açısı) – düğüm etiketlerinin gösteriminde kullanılacak açı: Açı 360 dereceden küçükse yatay çizgiye görecedir (0 – yatay); değilse yerleşimin merkezine görecedir (360 – ortak merkezli). Son seçenek bütün düğümler **Layout/Circular** (Yerleşim/Çembersel) kullanılarak eşmerkezli çemberler üzerinde çizildiğinde faydalıdır.
- **Label Position: Radius /Angle** (Etiket Pozisyonu: Yarıçap/Açı) – etiketin gösterileceği pozisyon:
 - **Radius** (Yarıçap) – düğüm etiketinin başlangıcının düğüm merkezine mesafesi – birinci polar parametre.
 - **Angle** (Açı) – Düğüm etiketinin derece olarak pozisyonu – ikinci polar parametre (0..360).

Eğer yarıçap 0 ise etiket ortalanır yoksa belirlenen pozisyonda sola dayalı olarak yazılır.

- **Shape** (Şekil) – varsayılan düğüm şekli (*ellipse* (elips), *box* (kutu), *diamond* (elmas), *triangle* (üçgen), *house* (ev), *man* (erkek), or *woman* (kadın)).
- **Shapes file** (Şekil dosyası) – varsayılan şekil dosyası. Değiştirmek için çift tıklayın.
- **Export options overwrite shapes file** (Aktarma seçenekleri şekil dosyasına göre önceliklidir) – seçili ise bu penceredeki seçenekler şekil dosyasındaki seçeneklere göre önceliklidir yoksa seçilen şekil dosyasında tanımlı her bir şekil için varsayılan değerler kullanılır.

Sol alt çerçeve – EPS/SVG Line Default (EPS/SVG Bağlantı Varsayılanları)

Bu çerçeve EPS veya SVG formatına aktarmada bağlantıların çizilmesi ile ilgili varsayılan parametreleri içerir.

- **Edge Color** (Kenar Rengi) – kenarların rengi.
- **Edge Width** (Kenar Genişliği) – kenarların kalınlığı.
- **Arc Color** (Yay Rengi) – yayların rengi.
- **Arc Width** (Yay Genişliği) – yayların kalınlığı.

- **Pattern** (desen) – bağlantı çizgisi deseni: *Solid* (düz) or *Dots* (noktalı).
- **Arrow Size** (Ok büyüklüğü) – okların büyüklüğü.
- **Arrow Position** (Ok Pozisyonu) – okun son düğümden uzaklığı: eğer bu mesafe 0 ve 1 arasındaysa yay uzunluğuna göre orantılı mesafe demektir. Örneğin 0 – ok son düğüme dokunur, 0.5 ok yayın ortasındadır. Eğer mesafe 1'den büyükse son düğümden mutlak mesafeyi ifade eder (uzunluğu ne olursa olsun bütün yayların son düğümden eşit mesafede olmasını istiyorsanız bu kullanışlı bir seçenektir)
- **Label Color** (Etiket Rengi) – bağlantı etiketlerinin rengi.
- **Label Angle** (Etiket Açısı) – bağlantı etiketinin gösterileceği açı: bu açı 360 dereceden küçükse bağlantının yönüne görecelidir (0 – bağlantıya paralel); aksi durumda yatay doğruya görecelidir (360 – yatay).
- **Label Position** (Etiket Pozisyonu) – bağlantı etiketinin merkezinin pozisyonu – pozisyon bağlantı üzerinde bir noktadır – bağlantı etiketinin merkezinden son düğüme olan mesafe (ayrıca bkz. **Arrow Position** (Ok Pozisyonu))
- **FontSize** (Etiket Büyüklüğü) – bağlantı etiketleri için font büyüklüğü.
- **Label Position: Radius /Angle** (Etiket Pozisyonu: Yarıçap/Açı) – bağlantı etiketinin merkezinin **Label Position** (etiket pozisyonuna) görece gösterilme pozisyonu:
 - **Radius** (yarıçap) – bağlantı etiketinin merkezinden **Label Position** (etiket pozisyonu) olarak tanımlanan noktaya olan mesafeye – birinci polar parametre
 - **Angle** (açı) – etiketin derece olarak pozisyonu – ikinci polar parametre (0..360)
- **Bezier Curves** (Bezir Eğrileri) – İlk ve son düğümlerdeki *angle* (açı) and *velocity* (hız) parametrelerini girin (Bezier parametreleri). Her iki açı da 0 ise bütün bağlantılar doğru olarak çizilir yoksa seçilen Bezier eğrileri bağlantıları çizmek için kullanılır.
- **Straight Lines** (Düz Çizgiler) – bütün bağlantılar (iki yönlü ve çoklu bağlantılar da dahil) doğru çizgiler olarak çizilir

Sağ üst çerçeve

Bu çerçeve EPS, SVG, X3D ve VRML aktarmaları için ekstra varsayılan parametreleri tanımlar.

- **EPS, SVG, X3D, VRML Size of Vertices** (Düğüm Büyüklüğü) – X3D/VRML'e aktarmada varsayılan düğüm büyüklüğü (EPS ve SVG formatları için geçerlidir).
- **X3D/VRML Size of Lines** (Bağlantı Kalınlığı) – X3D/VRML formatlarına aktarmada bağlantı kalınlığının varsayılan değeri .
- **SVG: Opacity of Vertices** (SVG: Düğümlerin Bulanıklığı) – 0.00 (düğümler tamamen şeffaf) ve 1.00 (düğümler tamamen opak (arkasını göstermeyen), varsayılan) arasında bir değer.
- **SVG: Opacity of Lines** (SVG: Bağlantıların Bulanıklığı) – 0.00 (bağlantılar tamamen şeffaf) ve 1.00 (bağlantılar tamamen opak (arkasını göstermeyen), varsayılan) arasında bir değer.
- **SVG: Vertices 3D Effect** (SVG: Düğümler için 3B efekti) – seçildiğinde düğümlere üç boyutlu bir görüntü kazandırmak için (doğrusal veya radyal gradyant kullanılır).
- **SVG Tooltips for Labels of Vertices/Lines/Clusters** (Düğüm/bağlantı/küme-ler için SVG ipuçları) – seçildiğinde düğüm ve bağlantılar için etiket gösterilmez. Onun yerine fare ile nesne üzerinde gelindiğinde bir ipucu görünür. Seçilen kümenin başlığı kümedeki bütün düğümlerin bir listesidir)
- **EPS, SVG: Lines finished at Vertex Border** (EPS, SVG: Bağlantılar Düğüm Sınırında Biter) – Düğümler şeffaf bir biçimde çizilirken bağlantıları düğüm merkezine kadar çizmek yerine düğüm sınırında bitirmek daha iyi görüntü sağlayacaktır.
- **EPS: Use RGB colors instead of CMYK** (EPS: CMYK yerine RGB renklerini kullan) – EPS dosyalarında varsayılan CMYK renkleri yerine RGB renklerini kullan.

Sağ orta çerçeve – *Background Colors* (Arkaplan Renkleri)

Bu çerçeve EPS/SVG/X3D/VRML formatlarına aktarmada kullanılacak arkaplan renklerini ve SVG/X3D formatları için gradyantların (bir renkten diğerine yumuşak geçiş) belirlemek için kullanılır.

- **Bckg. Color 1** (Arkaplan rengi 1) – EPS/SVG/X3D/VRML düzeni için arkaplan rengi

- **Bckg. Color 2** (Arkaplan rengi 2) – SVG/X3D çıktısında ikinci renk. *No (Hayır)* – ikinci renk olmadan demektir yoksa seçili gradyant kullanılır.
- **Bckg. Color 3** (Arkaplan rengi 3) – SVG/X3D çıktısında üçüncü renk – gradyant.
- **Opacity** (Bulanıklık) – Arkaplanın bulanıklığı.
- **Gradients** (Gradyantlar) – SVG'de kullanılan gradyant tipi (*No (Hayır)*, *Linear (Doğrusal)*, veya *Radial (Radyal)*). X3D'de yalnızca *Radial (Radyal)* gradyant kullanılabilir.

Sağ Alt Çerçeve

EPS/SVG formatlarına aktarmada kullanılan diğer varsayılan parametreler:

- **Left, Right, Top, Bottom** (Sol, Sağ, Üst, Alt) – EPS'e aktarmada yerleşimin etrafında ekstradan sınırlar çizer (yalnızca *EPS Clip* (EPS klip) formatı seçildiğinde)
- **Color** (Renk) – yerleşim sınırının rengi. *No (Hayır)* – sınır çizilmesin demektir.
- **Radius** (Yarıçap) – yerleşim sınırının yarıçapı (0'dan büyükse oval değilse dikdörtgen demektir).
- **Width** (Kalınlık) – yerleşimin sınırının kalınlığı.



Şekil 17: Örümcek / Krizevec; Fotoğraf: Stana.

5.3 Resimlerin EPS/SVG'ye aktarılması – parametrelerin girdi dosyasında tanıtılması

Bir Metin Girdi Dosyasında Ağın ve Resminin Tanımı

Her bir düğüm ve bağlantının nasıl çizileceği konusunda detayları belirleyebiliriz (renkler, şekiller, büyüklükler, örüntüler, döndürmeler, kalınlıklar..).

Ağların tanımlanmasında bir çeşit standardize dil kullanılır. Aşağıdaki anahtar kelimeler bu dile dahildir:

1. ***Vertices** n – düğümler tanımlanır. Burada n düğüm sayısıdır. Her bir düğüm aşağıdaki tanımlama satırı kullanılarak tanımlanır.

düğüm_numarası etiket [x y z] [şekil] [varsayılan parametrelerde değişiklikler]

Açıklama:

- düğüm_numarası – düğüm numarası (1, 2, 3 ... n)
- etiket – eğer etiket A..Z veya 0..9 gibi bir karakterle başlarsa ilk boşluk etiketin sonunu belirler (örneğin düğüm1). Birden fazla kelime içeren etiketler özel karakterler arasına alınmalıdır (örneğin "düğüm 1").
- x, y, z – düğüm koordinatları (0 ve 1 arasında)

- **şekil** – düğümün şekli. Şekiller SHAPES.CFG dosyasında tanımlıdır (elips, kutu, elmas, üçgen, haç, boş, ev, erkek, kadın). Çizim penceresinde ev şekli üçgenle, erkek kutu ile, kadın ise elips ile değiştirilir. Şeklin bir çok parametresi vardır:
 - **sh** – sh ifadesini elips, kutu, elmas, üçgen, haç, boş, ev, erkek, kadın'dan biri ile değiştirin. Bu aslında nesnelere çizen PostScript prosedürünün adıdır (drawnet.pro dosyasında tanımlanmıştır).
 - **s_size** – varsayılan büyüklük
 - **x_fact** – x yönünde büyütme çarpanı
 - **y_fact** – y yönünde büyütme çarpanı
 - **phi** – Artı yönünde döndürme parametresi (0..360)
 - **r** – Dikdörtgen ve elmas çiziminde köşelerin yarıçap parametresi ($r = 0$ – dikdörtgen, $r > 0$ – yuvarlak köşeler)
 - **q** – elmas şekli parametresi – elmas şeklinin üst ve orta kenarının oranı ($q = 0.01, q = 0.5, q = 2, \dots$ vb. deneyebilirsiniz)
 - **ic** – düğümün içinin rengi. Renk seçenekleri için bkz.: Sayfa 113 Şekil 19.
 - **bc** – düğümün sınır rengi
 - **bw** – düğümün sınır kalınlığı
 - **lc** – etiket rengi
 - **la** – derece olarak etiket açısı (0..360)
 - **lr** – düğüm etiketinin başlangıcının düğüm merkezine uzaklığı (yarıçap – birinci polar parametre)
 - **lphi** – derece olarak etiketin pozisyonu (0..360) (phi açısı – ikinci polar parametre)
 - **fos** – font büyüklüğü
 - **font** – etiketlerin yazılmasında kullanılan PostScript fontu (Helvetica, Courier, ...)
 - **HOOKS** – kenarların seçilen şekille birleşme pozisyonları - s_size parametresine göre. Bu pozisyonlar üç farklı şekilde belirlenebilir:
 - (a) **CART** – x y – Kartezyen koordinatlar (x,y)
 - (b) **POLAR** – r phi – polar koordinatlar – yarıçap r ve pozitif açı phi (0..360)
 - (c) **CIRC** – r phi1 – pozisyonların polar koordinatlarda iterasyonu r – yarıçap, $phi = k * phi1, k = 1, 2, \dots; k * phi1 \leq 360$

Tanımlama satırında her bir düğüm için varsayılan değerler değiştirilebilir.

Örneğin

1 "düğüm bir" 0.3456 0.1234 0.5 box ic White fos 20

Açıklama: 1 nolu düğümü beyaz bir kutu temsil ediyor, etiketi (düğüm bir) için font büyüklüğü 20'dir.

2. ***Arcs** (or ***Edges**) – yay (arc) ve kenarların (edge) tanımlanması. Formatı:

v1 v2 değer [ekstra parametreler]

Açıklama:

- v1 – ilk düğüm numarası
- v2 – son düğüm numarası
- değer – v1 ve v2 arasındaki yay veya kenarın değeri

Yukarıdaki üç parametre tanımlamada her zaman olmak zorundadır. Başka bir parametre verilmezse varsayılan yay/kenar özellikleri siyah renkli ve düz çizgili olur. Bunun istisnaları:

- eğer değer negatifse düz çizgi yerine noktalı çizgi kullanılır
- eğer yay bir ilmekse (düğümünden çıkıp düğüme giren yay) Bezier eğrisi kullanılır,
- iki yönlü yay varsa iki Bezier eğrisi kullanılır

Ok yayın son ucuna (son düğümünün olduğu tarafında) çizilir.

Daha önce değinildiği gibi bağlantıların düğümlere birleştiği noktaları belirlemek için kancalar kullanılır.

İlave parametreler:

- **w** – çizgi kalınlığı
- **c** – çizgi rengi
- **p** – çizgi deseni: solid (düz), dots (noktalı)
- **s** – okun büyüklüğü
- **a** – okun şekli (A or B)
- **ap** – okun pozisyonu
 - $ap = 0$ – ok son düğümde
 - $0 < ap \leq 1$ – son düğümünden bağlantı uzunluğuna göre oranlanmış mesafe

- $ap > 1$ – mutlak mesafe
- **l** – bağlantı etiketi (örneğin "line 1 2")
- **lp** – etiket pozisyonu (ap maddesine bakınız)
- **lr** – etiket yarıçapı (position of center text from point on edge)
- **lphi** – etiket yarıçapı (angle of center text according to point on edge) (lr ve lphi polar koordinatlarıdır)
- **lc** – etiket rengi
- **la** – etiket açısı ($0 < la < 360$ – kenara göre mutlak açı, $la \geq 360$ – x eksenine göre mutlak açı)
- **fos** – etiketin font büyüklüğü
- **font** – Etiketlerin yazımında kullanılacak PostScript fontu (Helvetica, Courier, ...)
- **h1** – ilk düğümdeki kanca (0 – merkez, -1 en yakın, 1, 2.. kullanıcının tanımladığı)
- **h2** – son düğümdeki kanca
- **a1** – ilk düğümdeki açı (Bezier)
- **k1** – ilk düğümdeki hız (Bezier)
- **a2** – son düğümdeki açı (Bezier)
- **k2** – son düğümdeki hız (Bezier)

Bağlantılar için özel şekiller $a1$ ($\alpha1$), $k1$, $a2$ ($\alpha2$) ve $k2$ parametrelerinin değişik kombinasyonları kullanılarak elde edilebilir.

- $\alpha1 = \alpha2 = 0, k1 \geq 0, k2 \geq 0$ – düz çizgi (varsayılan)
- $\alpha1 = \alpha2 = 0, k1 = -1, k2 > 0$ – $k2$ yarıçaplı oval kenar (yarıçap yukarıda ifade edildiği gibi mutlak olarak ölçülür)
- $\alpha1 = \alpha2 = 0, k1 = -1, k2 < 0$ – $-k2$ yarıçaplı ikinci mümkün oval kenar
- $\alpha1 = \alpha2 = 0, k1 = -2, k2 > 0$ – $k2$ yarıçaplı pozitif yönde çembersel yay
- $\alpha1 = \alpha2 = 0, k1 = -2, k2 < 0$ – $-k2$ yarıçaplı pozitif yönde ikinci mümkün yay
- $\alpha1 = \alpha2 = 0, k1 = -3, k2 > 0$ – $k2$ yarıçaplı negatif yönde çembersel yay

- $alpha1 = alpha2 = 0, k1 = -3, k2 < 0 - -k2$ yarıçaplı negatif yönde çembersel yay
 - $alpha1 = alpha2 = 0, k1 = -4 -$ çift kenar
 - $alpha1$ or $alpha2 \neq 0, k1 > 0, k2 > 0 -$ Bezier eğrisi (eğer $alpha1$ ve $alpha2$ farklı işarete sahipse bağlantı düğümleri birleştiren düz çizginin bir tarafından diğerine gider, eğer $alpha1$ ve $alpha2$ aynı işarete sahipse – bağlantı düğümleri birleştiren doğrunun aynı tarafında kalır)
3. ***Edges** (Kenarlar) – kenarların tanımı. Yaylar için kullanılan parametrelerin aynısı kenarlar için de kullanılır yalnız tip (a), büyüklük (s) ve pozisyon (ap) parametrelerinin kenarlar için bir anlamı yoktur.

PS ve EPS

- PS – Postscript (.PS) dosyasına aktar (üstbilgi drawnet.pro olmadan). Eğer birçok resminiz varsa ve kelime işlemci yazılımınız üstbilgiyi ayrıca tanımlamanıza izin veriyorsa (örneğin L^AT_EX) boş alanı korumak için bu seçeneği kullanın.
- EPS – Encapsulated PostScript dosyasına (.EPS) aktar . Drawnet.pro üstbilgi olarak dosyanın başına eklenir. Sonuçta çıkan resim tam bir resimdir. Bundan dolayı bir metinde kullanabilir veya kalitesini kaybetmeden büyütüp küçültebilir, döndürebilir, Postscript yazıcıda yazdırabilir, GhostScript göstericisinde bakabilir ve PDF veya JPG'e çevirebilirsiniz.

Örnek

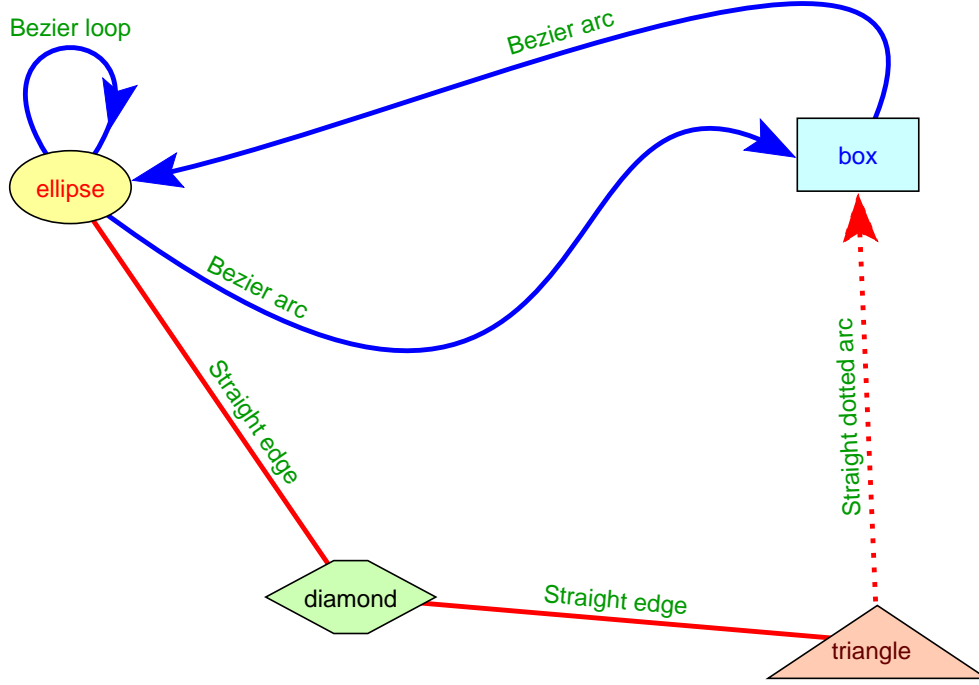
```
*Vertices 4
1 "ellipse" 0.120 0.285 0.5 ellipse x_fact 5 y_fact 3 fos 20 ic LightYellow lc Red
2 "box" 0.818 0.246 0.5 box x_fact 5 y_fact 3 fos 20 ic LightCyan lc Blue
3 "diamond" 0.368 0.779 0.5 diamond x_fact 7 y_fact 3 fos 20 ic LightGreen lc Black
4 "triangle" 0.835 0.833 0.5 triangle x_fact 9 y_fact 3 fos 20 ic LightOrange lc Brown lr 30 lphi 200
*Arcs
1 1 1 h2 0 w 3 c Blue s 2 a1 -130 k1 0.6 a2 -130 k2 0.6 ap 0.25 l "Bezier loop" lc OliveGreen fos 20 lr 13 lp 0.5 la 360
2 1 1 h2 1 a1 120 k1 1 a2 10 k2 0.8 ap 0 l "Bezier arc" lphi 270 la 180 lr 13 lp 0.5
1 2 1 h2 -1 a1 40 k1 2 a2 -30 k2 0.8 ap 0 l "Bezier arc" lphi 90 la 0 lp 0.75
4 2 -1 l "Straight dotted arc" p Dots c Red
*Edges
1 3 1 l "Straight edge" lp 0.4
3 4 1 l "Straight edge"
```

Pajek'in çizim penceresinde bazı seçenekleri belirlemeniz gerekir. Örneğin bağlantı etiketlerini aktifleştirmek için:

Options / Lines / Mark Lines / with labels (Seçenekler / Bağlantılar / Bağlantıları İşaretle / Etiketle)

5.4 Pajek Resimlerinde Unicode Kullanımı

Bazen bir Pajek resminde farklı bir alfabeye ait olağandışı karakterler kullanmak isteyebiliriz, örneğin özel semboller, Kril, Arapça, Çince, ...). Bu alfabe kodları



Şekil 18: Örnek resim

Unicode adresinde bulunabilir. Pajek versiyon 2.00'dan itibaren BOM'lu Unicode UTF8 dosyalarını desteklemektedir.

Pajek'den Unicode kullanımına dair bazı ipuçları:

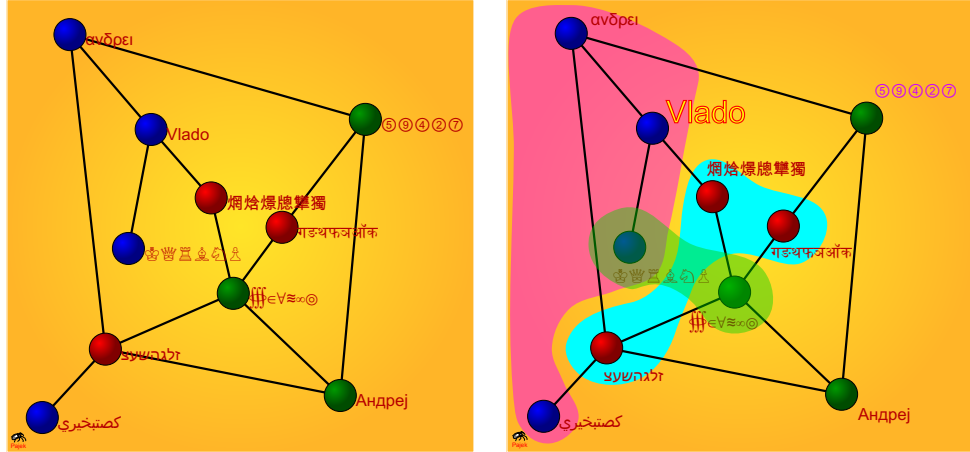
- Ücretsiz bir Unicode editörünü kullanarak **BabelPad** vb. ağ dosyasının (*.NET) Unicode versiyonu hazırlanır. Unicode karakterleri girmek için U aracı (Unicode tablosu) kullanılabilir.
- **BabelPad** seçeneğini kullanarak `File/Save as/UTF8` ve `Byte Order Mark` seçeneğini işaretleyerek bir ağ Unicode dosyası olarak kaydedebilirsiniz. Ayrıca `File/Save as/ASCII plus decimal NCR`'yi (`Dosya/Kaydet / ASCII` artı ondalık NCR) de kullanabilirsiniz. Böylece Unicode dosya Unicode karakterlerinin numerik kod `&#dxxx;` olarak gösterildiği ASCII dosyasına dönüştürülebilir.

Üretilen iki dosyadan herhangi biri Pajek'de açılabilir ve uygun Unicode karakterleri (uygun orantılı ve eş aralıklı fontlar `Options/Select Font`

ile veya ana pencerede seçilirse) çizim, rapor ve diğer pencerelerde görünecektir. Tavsiyemiz orantılı font olarak Arial Unicode MS veya Lucida Sans Unicode fontlarını (genelde birisi Windows'da kullanıma hazırdır), orantılı olmayan yani eş aralıklı font olarak da **GNU Unifont**'un kullanılmasıdır. Courier New de kullanılabilir.

Options/Read-Write/Save Files as Unicode UTF8 with BOM (Seçenekler / Oku-Yaz / Dosyaları BOM'lu Unicode UTF8 formatında kaydet) seçeneği seçilmezse ağlar kaydedildiğinde genellikle ASCII dosyası olarak kaydedilir. Bu varsayılan durumdur ve ASCII olmayan karakterler &#dddd; olarak kaydedilir.

Aşağıdaki görsel **Pajek 2.00** (ya da daha yükseği) kullanılarak elde edilmiş ve daha sonra SVG formatına aktarılmıştır.



- SVG dosyası ücretsiz bir yazılım olan **InkScape** kullanılarak açılabilir. Burada resim üzerinde birkaç iyileştirme yapabiliriz.
- InkScape'de **File/Save as/pdf** veya **eps** seçeneklerini kullanarak resmi Unicode karakterli etiketleriyle beraber PDF veya EPS vs formatlarda kaydedebiliriz. PDF (versiyon 1.4 ve sonrası) şeffaflık parametresini de destekler. EPS dosyaları ayrıca Word dokümanına da eklenebilir.

Benzer bir yaklaşım 3XD'de üç boyutlu modeller için de kullanılabilir. Bunun için **Instant Player**'ı kullanmanızı tavsiye ederiz.

6 Pa jek'de Makrolar

6.1 Makro nedir?

Makro ile temel Pa jek komutlarından oluşan bir dizi bir dosyaya kaydedilebilir. Daha sonra bu dosyayı kullanarak içindeki komutlar dizisini sırasıyla çalıştırabilirsiniz.

6.2 Makro nasıl kaydedilir?

1. Öncelikle makro tarafından kullanılacak bütün nesnelere (ağlar, sınıflamalar, ...) yüklenir.
2. Makroda kullanılacak nesnelere seçilir. Örneğin bir ağ kullanılacaksa ağ kutucuğunda görünür olmalıdır.
3. Macro/Record komutunu çalıştırın ve makro dosyanız için bir isim verin (varsayılan makro dosyası uzantısı: `.mcr`).
4. Pa jek'i her zamanki gibi kullanın ve böylece bir komut dizisi oluşacaktır. Makronun daha iyi anlaşılması için dosya içine açıklamalar koymanız yararlı olacaktır. Makro çalıştırıldığında bu açıklamalar rapor penceresinde görünecek ve böylece komutları takip etmeniz kolaylaşacaktır.
5. Sonunda Macro/Record komutunu tekrar çalıştırarak macro kaydını durdurun.

6.3 Makro nasıl çalıştırılır?

1. Öncelikle makronun çalıştırılmasında girdi olarak kullanılacak nesne(ler)i yükleyin
2. Macro/Play komutu ile çalıştırılacak makro dosyasını seçin.

6.4 Örnek

Aşağıdaki makro çevrimsiz bir ağ üzerinde topolojik sıralama işlemini yapar.

```
NETBEGIN 2
CLUBEGIN 1
PERBEGIN 1
CLSBEGIN 1
HIEBEGIN 1
VECBEGIN 1
```

NETPARAM 1

```
Msg Depth Partition
C 1 DEP 1 (10)
Msg Make Permutation
P 1 MPER 1 (10)
Msg Reordering network
N 2 REOR 1 1 (10)
```

Buradaki ilk yedi komut ana pencerede görünen kutucukların mevcut durumlarını kaydeder. Makroyu çalıştırmadan önce topolojik sıralama yapmak istediğimiz çevrimsiz ağ ağ kutucuğundaki ilk ağ olmalıdır.

6.5 Kurulum dosyasında mevcut makrolar

6.5.1 Soyağacı ve diğer çevrimsiz ağlar için hazır olarak gelen makrolar

- **Path** – İki düğüm arasındaki en kısa zinciri (yönsüz yolu) bul
- **Descendants** – Bir p çizgesindeki seçili bir düğümde 'sonra' gelen bütün düğümleri bul.
- **Ancestors** – Bir p çizgesindeki seçili bir düğümde 'önce' gelen bütün düğümleri bul.
- **Cognatic3** – Bir p çizgesindeki seçili bir düğümde 'önce' gelen üç nesildeki ve 'sonra' gelen üç nesildeki bütün düğümleri bul.
- **Cognatic** – Bir g çizgesindeki seçili bir düğümde 'önce' ve 'sonra' gelen ulaşılabilir bütün düğümleri bul
- **Layers, Layers1, Layers2** – Çevrimsiz bir ağı tabakalar halinde çiz (farklı algoritmalar).
- **zLayers** – Çevrimsiz bir ağı tabakalar halinde z yönünde çiz.
- **LongestPatrilineage** – Ore soyağacında baba tarafından en uzun silsileyi bul. Soyağacı Ore çizgesi: 1-Erkek, 2-Kadın bağlantıları işaretli şekilde okunmalı.
- **LongestMatrilineage** – Ore soyağacında anne tarafından en uzun silsileyi bul. Soyağacı Ore çizgesi: 1-Erkek, 2-Kadın bağlantıları işaretli şekilde okunmalı.

- **NumDescendants** – her bir düğüm için dış etki alanının büyüklüğünü hesapla (Ore çizgesinde alt nesillerdeki düğüm sayısı).
- **NumAncestors** – her bir düğüm için iç etki alanının büyüklüğünü hesapla (Ore çizgesinde üst nesillerdeki düğüm sayısı).

6.5.2 Akrabalık ilişkilerini hesaplamak için hazırlanmış makrolar

Pajek akrabalık ilişkilerini bir Ore çizgesi olarak okumak için üç adet sınıflama oluşturur: Seçenekler: Ore çizgesi: 1-Erkek, 2-Kadın bağlantıları işaretlenmiş:

- 1: *babası*
- 2: *annesi*
- 3: *eşi*

Diğer yakınlık ilişkileri makro çalıştırarak bu sınıflamalardan elde edilebilir. Cinsiyet sınıflaması da akrabalık ilişkilerinin okunmasıyla ortaya çıkan bir sınıflamadır ve aşağıdaki makrolar için girdi olarak kullanılır.

- 4: *ebeveyni*
- 5: *çocuğu*
- 6: *oğlu*
- 7: *kızı*
- 8: *kocas*
- 9: *karısı*
- 10: *kardeşi*
- 11: *erkek kardeşi*
- 12: *kız kardeşi*
- 13: *amcası/dayısı*
- 14: *teyzesi/halası*
- 15: *yarı kardeşi*

`add.all.relations` makrosunu kullanarak yukarıdaki bütün ilişkileri bir defada ekleyebilirsiniz.

Bir test/örnek olarak `*/pajek/data/family.ged`'i kullanabilirsiniz.

6.6 Oturumu Tekrar Etmek

Yazılımı kullanırken çalıştırılan bütün komutlar bir kayıt dosyasına yazılır (`*.log`). Böylece *Macro/Repeat Last Command* menüsünü kullanarak bu log dosyasını seçip çalıştırabilir ve bütün işlemleri tekrar edebilirsiniz. Kayıt dosyasını adını değiştirirseniz yazılım bir dahaki çalıştırmanızda size dosya adı soracaktır. Böylece komut serisini farklı bir veri için tekrar edebilirsiniz. Başlangıç kayıt dosyası olan `Pajek.log` dosyası `Pajek.exe` dosyasının olduğu klasörde mevcutsa `Pajek` çalıştırıldığında bu kayıt dosyasındaki komutlar da tekrar çalıştırılır.

6.7 Komut çalıştır

'Run Command' (komut çalıştır) daha önce çalıştırılmış komutlardan birinin yeni parametrelerle yeniden çalıştırılmasını sağlar. Bunun için listede önceden çalıştırılmış komutlardan birini seçin, parametrelerini değiştirin ve sonra da güncellenmiş parametrelerle bu komutu çalıştırın.

6.8 Son komutun tekrar edilmesi

Makro alt menüsü Pajek tarafından çalıştırılan son komutun sırayla farklı nesnelere defalarca uygulanmasına imkan verir.

Örneğin: 100 adet ağ Pajek'e yükledikten sonra birinci ağda derece sınıflamasını yaptıktan sonra Repeat Last Command (Son Komutu Tekrar Çalıştır) menüsünü kullanıp girdi olarak 99 girebilir ve kalan bütün ağlarda derece sınıflamasını oluşturabilirsiniz.

Birçok nesneyi içeren komutlar da bu şekilde çalıştırılabilir, örneğin seçili sınıflamalara göre alt ağlar çıkarmak. Bir grup ağdan çıkarmak için aşağıdaki üç farklı yol kullanılabilir (ilgili nesnelere sabitlenerek bu durumlar ayarlanabilir).

- bütün ağlar için aynı sınıflamanın çıkarmak için kullanılması (artış ağa uygulanır, sınıflamaya değil yani sınıflama sabit)
- her bir ağ için farklı bir sınıflamanın kullanılması (artış hem ağa hem de sınıflamaya uygulanır yani hiç bir şey sabit değil)
- bir ağdan için farklı sınıflamaların kullanılması (artış sınıflamaya uygulanır, ağa değil, yani ağ sabit)

Önemli: Her zaman önce komutu yüklenen ilk nesne(ler) üzerinde çalıştırın. Yineleme, komutun çalıştırıldığı nesne(ler)in numarasından bir fazla olan nesne(ler) ile başlayacaktır.

Komutun sonucu da bir sabit ise, tüm sabitler bir vektörde tutulur. Pajek'te aşağıdaki sabitler bulunur: düğüm sayısı, yaylar, kenarlar, ağ yoğunlukları, merkezileştirme indisleri, çap, yeniden bağlama indisleri, korelasyon ve bağımlılık katsayıları, parçaların sayısı, ana çekirdek sayısı, bileşenlerin sayısı, kritik yolun uzunluğu, maksimum akış, düğümler arası mesafe, adaların sayısı, sınıflama/hiyerarşideki minimum ve maksimum değer, vektörde minimum, maksimum, aritmetik ortalama, medyan ve standart sapma.

6.9 Son komutla oluşturulan sabitlerin skalerlere kaydedilmesi

Bazı komutların çalıştırılması sonucunda bir veya daha fazla sabit rapor penceresine eklenebilir. Örneğin bütün bilgi komutu sonuçları bileşenler (sayısı, en

büyüğünün büyüklüğü), ağ merkezileşmeleri, ağ kümelenme katsayısı, ağ çapı, ... Bu sabitleri skalerlere (bir boyutlu vektörler) daha sonra kullanmak üzere kaydedebiliriz. Örneğin bu skalerleri ağların veya vektörlerin normalize edilmesi için kullanabiliriz.

7 Pajek'de Blok Modelleme

Genelleştirilmiş blok modellemesine optimizasyon yaklaşımları olan MODEL2 ve TwoMODEL modelleri STRAN (STRucture ANalysis, [9]) paketinden Pajek'e aktarılmıştır.

Blok modelleme komutu verilen bir ağın verilen blok tiplerini sağlayan en uygun sınıflamayı bulmaya çalışır ve genelleştirilmiş blok modellemesi olarak adlandırılır. [7, 35]

En uygun sınıflamanın bulunmasına verilen sayıda küme için rassal bir sınıflamadan başlanır. Sonrasında blok modelleme algoritması düğümleri bir kümeden diğerine aktararak veya iki farklı kümeye ait iki düğümü karşılıklı olarak değiştirerek (transpoze ederek) sınıflamayı optimize etmeye çalışır. Blok modeli Pajek içinde oluşturulabilir (*User Defined* seçeneği seçili ise) ve/veya tanımlaması bir MDL dosyasına kaydedilerek oluşturulabilir. Her bir bloktaki hataların etkisi ceza ağırlıkları kullanılarak kontrol edilebilir.

Bu seçenek iki modlu ağların genelleştirilmiş blok modellemesini de destekler [34].

Komutun uygulanabileceği ağın maksimum boyutu 1000 düğümdür. Ancak gerçek sınırlama zamanla ilgilidir – 100 düğümde bile optimizasyon birkaç saat sürebilir.

Komutun sonuçları sınıflamalar halinde tutulur. Bu sonuçlar bir resim olarak görüntülenebilir.

```
Draw / Network+Partition
ve
Layout / Energy / Kamada-Kawai / Free
veya
Layout / Circular / using Partition
```

Sonuç ayrıca matris formatında da görüntülenebilir. Bunun için iki adım gereklidir:

```
Partition / Make Permutation
File / Network / Export as Matrix to EPS /
Using Permutation + Partition.
```

Sonra dosya adını girin ve yes'e basın.

7.1 MDL dosyaları

Bir MDL dosyasının yapısı aşağıdaki örnekten anlaşılabilir.

```
*MODEL Tina
9
0 3 100 0 1 2 3 4
```

```
*CONSTRAINTS
1 100 2 1
4 100 1 3
*EOM
```

Her bir satırın ilk karakteri bir yıldız (*) veya boşluk olmalıdır. Bir satırdaki son karakter boşluk olmamalıdır.

İkinci satırdaki sayı küme sayısıdır. **iki modlu ağ** durumunda bu satırda iki numara verilmelidir (satır küme sayısı ve kolon küme sayısı). Diğer veriler her iki tip ağda da aynıdır. Aşağıdaki satırlar yapıyı içerir:

$$i \quad j \quad penalty \quad t_1 \quad t_2 \dots t_k$$

$i, j > 0$ olduğunda bu satır (i, j) bloğunun $t_1, t_2 \dots t_k$ tiplerine sahip olabileceğini gösterir. Bu tipler aşağıdaki gibi kodlanmıştır.

```
0 - - null (boş)
1 com - complete (tam)
2 rdo - row-dominant (satır baskın)
3 cdo - col-dominant (kolon baskın)
4 reg - regular (düzenli)
5 rre - row-regular (satır düzenli)
6 cre - col-regular (kolon düzenli)
7 rfn - row-function (satır fonksiyonu)
8 cfn - col-function (kolon fonksiyonu)
9 den - density (yoğunluk)
10 dnc - do not care (farketmez)
11 one - non-null (boş olmayan)
12 sym - symmetric (simetrik)
```

$i = 0$ olan satırlar model matrisinin parçalarının tiplerini tanımlar.

- $j = 0$: diagonal (köşegensel, yalnızca bir modlu ağlarda geçerli);
- $j = 1$: upper triangle (üst üçgen, yalnızca bir modlu ağlarda geçerli);
- $j = 2$: lower triangle (alt üçgen, yalnızca bir modlu ağlarda geçerli);
- $j = 3$: complete matrix (tam matris).

Aynı bloğu tanımlayan bir çok satır olması durumunda son satır dikkate alınır.

*CONSTRAINTS satırından sonraki satırlar blok modellemesindeki diğer kısıtlılıkların tanımlamasını yapar. Kısıtlılıklar aşağıdaki formata sahiptir:

$$k \text{ penaltı } i \ j$$

burada

- $k = 1$: $i \in C_j - i$ düğümü C_j kümesine aittir;
- $k = 2$: $i \notin C_j - i$ düğümü C_j kümesine ait değildir;
- $k = 3$: $C(i) = C(j) - i$ ve j düğümleri aynı kümeye aittir;
- $k = 4$: $C(i) \neq C(j) - i$ ve j düğümleri farklı kümeye aittir;
- $k = 5$: $i \leq |C(j)| - C_j$ kümesinde en az i düğüm vardır;
- $k = 6$: $i \geq |C(j)| - C_j$ kümesinde en fazla i düğüm vardır.

$k = 5$ ve $k = 6$ tipi kısıtlılıklar (seçili kümede minimum/maksimum düğüm sayısı kısıtlılıkları) için iki modlu ağla çalışılması durumunda dikkatli olunmalıdır: Bir blok modelde r satır ve c kolon kümesi varsa satır kümesi kısıtlılığı tanımlamak için $1 \dots r$ aralığında sayıları, kolon kümesi için ise $r + 1 \dots r + c$ aralığındaki sayıları kullanın.

Kısıtlılıkların ihlali kriter fonksiyonuna aşağıdaki terimle katkıda bulunur:

$$+ \text{ ihlal sayısı } \times \text{ penaltı}$$

Penaltı değerleri 0-1000 aralığında olmalıdır.

7.2 MDL dosyası örnekleri

7.2.1 Düzenli bloklar

```
*MODEL Regular
10
0 3 1 0 1 4
*EOM
```

7.2.2 Köşegen bloklar (kümelenme)

```
*MODEL Diagonal
10
0 3 100 0
0 0 1 0 1 4
*EOM
```


7.2.3 Çevrimsel olmayan model (yukarı)

```
*MODEL Hierarchy
9
0 1 1 0 5 6
0 0 10 0 1 4 12
0 2 100 0
*EOM
```

7.2.4 Simetrik kümeli çevrimsel olmayan model (aşağı)

```
*MODEL SymHiera
9
0 0 10 0 1 12
0 1 100 0
0 2 1 0 11
*EOM
```

7.2.5 Merkez-çevre

```
*MODEL Center-Periphery
2
0 3 1 0 11
2 2 10 0
1 1 100 0 1 4
*EOM
```

7.2.6 Düzenli yol

```
*MODEL Regular Path
9
0 0 10 0 1 4
1 2 10 0 1 4
2 3 10 0 1 4
3 4 10 0 1 4
4 5 10 0 1 4
5 6 10 0 1 4
6 7 10 0 1 4
7 8 10 0 1 4
8 9 10 0 1 4
*EOM
```





















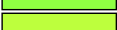























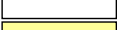


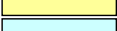




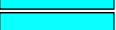


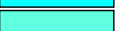






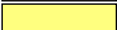


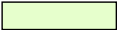




























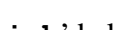

7.2.7 Düzenli zincir

```
*MODEL Regular Chain
9
0 0 10 0 1 4
1 2 10 0 1 4
2 3 10 0 1 4
3 4 10 0 1 4
4 5 10 0 1 4
5 6 10 0 1 4
6 7 10 0 1 4
7 8 10 0 1 4
8 9 10 0 1 4
2 1 10 0 1 4
3 2 10 0 1 4
4 3 10 0 1 4
5 4 10 0 1 4
6 5 10 0 1 4
7 6 10 0 1 4
8 7 10 0 1 4
9 8 10 0 1 4
*EOM
```

7.2.8 Davis.net dosyası için iki modlu 'standart model'

```
*MODEL UserDefined
2 3
1 1 1 1
1 2 1 1
1 3 100 0
2 1 100 0
2 2 1 1
2 3 1 1
*EOM
```

8 Pa jek'de renkler

	GreenYellow		Fuchsia		JungleGreen
	Yellow		Lavender		SeaGreen
	Goldenrod		Thistle		Green
	Dandelion		Orchid		ForestGreen
	Apricot		DarkOrchid		PineGreen
	Peach		Purple		LimeGreen
	Melon		Plum		YellowGreen
	YellowOrange		Violet		SpringGreen
	Orange		RoyalPurple		OliveGreen
	BurntOrange		BlueViolet		RawSienna
	Bittersweet		Periwinkle		Sepia
	RedOrange		CadetBlue		Brown
	Mahogany		CornflowerBlue		Tan
	Maroon		MidnightBlue		Gray
	BrickRed		NavyBlue		Black
	Red		RoyalBlue		White
	OrangeRed		Blue		LightYellow
	RubineRed		Cerulean		LightCyan
	WildStrawberry		Cyan		LightMagenta
	Salmon		ProcessBlue		LightPurple
	CarnationPink		SkyBlue		LightGreen
	Magenta		Turquoise		LightOrange
	VioletRed		TealBlue		Canary
	Rhodamine		Aquamarine		LFadedGreen
	Mulberry		BlueGreen		Pink
	RedViolet		Emerald		LSkyBlue
	Gray05		Gray10		Gray15
	Gray20		Gray25		Gray30
	Gray35		Gray40		Gray45
	Gray55		Gray60		Gray65
	Gray70		Gray75		Gray80
	Gray85		Gray90		Gray95

Şekil 19: Pa jek'de kullanılabilir renkler.

	0 - Cyan		20 - WildStrawberry
	1 - Yellow		21 - ForestGreen
	2 - LimeGreen		22 - Salmon
	3 - Red		23 - LSkyBlue
	4 - Blue		24 - GreenYellow
	5 - Pink		25 - Lavender
	6 - White		26 - LFadedGreen
	7 - Orange		27 - LightPurple
	8 - Purple		28 - CornflowerBlue
	9 - CadetBlue		29 - LightOrange
	10 - TealBlue		30 - Tan
	11 - OliveGreen		31 - LightCyan
	12 - Gray		32 - Gray20
	13 - Black		33 - Gray60
	14 - Maroon		34 - Gray40
	15 - LightGreen		35 - Gray75
	16 - LightYellow		36 - Gray10
	17 - Magenta		37 - Gray85
	18 - MidnightBlue		38 - Gray30
	19 - Dandelion		39 - Gray70

Şekil 20: Sınıflama renkleri

9 Pajek'e nasıl atıfta bulunulur?

Pajek'e atıfta bulunmak için aşağıdaki referans bilgilerini kullanabilirsiniz:

- W. de Nooy, A. Mrvar, V. Batagelj: *Exploratory Social Network Analysis with Pajek: Revised and Expanded Edition for Updated Software. Third Edition*. Structural Analysis in the Social Sciences 46, Cambridge University Press, 2018. ISBN:9781108462273. [CUP](#), [Amazon](#).
- A. Mrvar, V. Batagelj: *Pajek – Program for Large Network Analysis*. Home page: <http://mrvar.fdv.uni-lj.si/pajek/>
- Mrvar, A. and Batagelj, V.: *Analysis and visualization of large networks with program package Pajek*. Complex Adaptive Systems Modeling, 4:6. 2016. [SpringerOpen](#).
- V. Batagelj, A. Mrvar: *Pajek – Analysis and Visualization of Large Networks*. In Jünger, M., Mutzel, P. (Eds.): Graph Drawing Software. Springer (series Mathematics and Visualization), Berlin 2003. 77-103. ISBN 3-540-00881-0. [Springer](#), [Amazon](#), [preprint](#)
- V. Batagelj, A. Mrvar: *Pajek – Program for Large Network Analysis*. Connections, **21**(1998)2, 47-57. [preprint](#)



Şekil 21: *Gartenkreuzspinne / Araneus diadematus*; **Fotoğraf:** Stefan Ernst

Kaynaklar

- [1] Adobe SVG viewer. <http://www.adobe.com/svg/>
<http://www.adobe.com/svg/viewer/install>
- [2] Ahmed, A., Batagelj, V., Fu, X., Hong, S.-H., Merrick, D., Mrvar, A. (2007): Visualisation and analysis of the Internet movie database. Asia-Pacific Symposium on Visualisation 2007 (IEEE Cat. No. 07EX1615), 17-24.
- [3] Albert R., Barabasi A.L.: Topology of evolving networks: local events and universality.
<http://xxx.lanl.gov/abs/cond-mat/0005085>
- [4] Batagelj V.: Papers on network analysis.
<http://vlado.fmf.uni-lj.si/pub/networks/doc/>
- [5] Batagelj V.: Workshop on Network Analysis, Sydney, Australia: 14th to 17th June 2005; at Nicta (National ICT Australia).
<http://vlado.fmf.uni-lj.si/pub/networks/doc/#NICTA>
- [6] Batagelj V.: Some new procedures in Pajek. Dagstuhl seminar 05361, Dagstuhl, Germany, Sept 5-9, 2005.
<http://vlado.fmf.uni-lj.si/pub/networks/doc/dagstuhl/NewProcs.pdf>
- [7] Batagelj, V. (1997) Notes on blockmodeling. *Social Networks* **19**, 143-155.
- [8] Batagelj V.: Efficient Algorithms for Citation Network Analysis.
<http://arxiv.org/abs/cs.DL/0309023>
- [9] Batagelj V.: **MODEL 2**. <http://vlado.fmf.uni-lj.si/pub/networks/>
- [10] Batagelj V. (2009): Social Network Analysis, Large-Scale. R.A. Meyers, ed., Encyclopedia of Complexity and Systems Science, Springer 2009: 8245-8265. <http://www.springerlink.com/content/tp3w7237m4624462/>
- [11] Batagelj V. (2009): Complex Networks, Visualization of. R.A. Meyers, ed., Encyclopedia of Complexity and Systems Science, Springer 2009: 1253-1268. <http://www.springerlink.com/content/m472707688618h17/>
- [12] Batagelj V., Brandes U. (2005): Efficient Generation of Large Random Networks. *Physical Review E* **71**, 036113, 1-5.
- [13] Batagelj, V., Ferligoj, A., and Doreian, P. (1992), Direct and Indirect Methods for Structural Equivalence, *Social Networks*, **14**, 63–90.

- [14] Batagelj, V., Doreian, P., and Ferligoj, A. (1992) An Optimizational Approach to Regular Equivalence. *Social Networks* **14**, 121-135.
- [15] Batagelj, V, Ferligoj, A, Doreian, P (2007): Indirect blockmodeling of 3-way networks. Selected contributions in data analysis and classification, Springer, Berlin, 151-159.
- [16] Batagelj, V., Kejžar, N., Korenjak-Černe, S. (2008): Analysis of the Customers? Choice Networks: An Application on Amazon Books and CDs Data. *Metodološki zvezki/Advances in Methodology and Statistics* **4** (2): 191-204.
- [17] Batagelj V., Mrvar A.: Pajek.
<http://vlado.fmf.uni-lj.si/pub/networks/pajek/>
- [18] Batagelj V., Mrvar A. (2000) Some Analyses of Erdős Collaboration Graph. *Social Networks*, **22**, 173-186
- [19] Batagelj V., Mrvar A. (2001) A Subquadratic Triad Census Algorithm for Large Sparse Networks with Small Maximum Degree. *Social Networks*, **23**, 237-243
- [20] Batagelj V., Mrvar A. (2008) Analysis of Kinship Relations With Pajek. *Social Science Computer Review* **26**(2), 224-246, 2008.
- [21] Batagelj V., Mrvar A., Zaveršnik M. (1999) Partitioning Approach to Visualization of Large Graphs. In: Kratochvil J. (Ed.) GD'99, Štířin Castle, Czech Republic. LNCS **1731**. Springer-Verlag, 90-97.
- [22] Batagelj V., Mrvar A., Zaveršnik M. (2002) Network analysis of texts. *Language Technologies*, Ljubljana, p. 143-148.
<http://nl.ijs.si/isjt02/zbornik/sdjt02-24bbatagelj.pdf>
- [23] Batagelj V., Zaveršnik M. (2011): Fast algorithms for determining (generalized) core groups in social networks. *Advances in Data Analysis and Classification*. Springer
- [24] Batagelj, V. and Zaveršnik, M. (2007): Short Cycles Connectivity. *Discrete Math* 307 (3-5): 310-318.
<http://arxiv.org/abs/cs.DS/0308011>
- [25] Bollobas B.: *Random Walks on Graphs*,
- [26] Broder A. et al. (2000): Graph structure in the web.
<http://www.almaden.ibm.com/cs/k53/www9.final/>

- [27] Burt R.S. (1992): *Structural Holes*. The Social Structure of Competition. Cambridge MA: Harvard University Press.
- [28] Butts, C.T. (2002) sna: Tools for Social Network Analysis.
<http://cran.at.r-project.org/src/contrib/PACKAGES.html#sna>
- [29] Cortona Player (2006)
<http://www.parallelgraphics.com/products/cortona/>
- [30] Cosmo Player (2002) <http://ca.com/cosmo/>
- [31] de Nooy W., Mrvar A., Batagelj V. (2018) *Exploratory Social Network Analysis With Pajek*. Revised and Expanded Edition for Updated Software. Third Edition. Structural Analysis in the Social Sciences 46, Cambridge University Press, 2018. ISBN:9781108462273. CUP, 2018, Chinese - Amazon, 2014, Japanese - Amazon, 2009.
- [32] Doreian P., Mrvar A. (1996) A Partitioning Approach to Structural Balance. *Social Networks*, **18**. 149-168
- [33] Doreian, P., Batagelj, V., Ferligoj, A. (2000) Symmetric-acyclic decompositions of networks. *J. classif.*, **17**(1), 3-28.
- [34] Doreian P., Batagelj V., Ferligoj A. (2004) Generalized blockmodeling of two-mode network data. *Social Networks* **26**, 29-53.
- [35] Doreian P., Batagelj V., Ferligoj A.: *Generalized Blockmodeling*, Structural Analysis in the Social Sciences 25, Cambridge University Press, 2005. ISBN:0521840856. CUP, Amazon.
- [36] Dremelj P., Mrvar A., Batagelj V. (2002) Analiza rodoslova dubrovačkog vlasteoskog kruga pomoću programa **Pajek**. Anali Dubrovnik XL, HAZU, Zagreb, Dubrovnik, 105-126 (in Croat).
- [37] GEDCOM 5.5.
<http://homepages.rootsweb.com/~pmcbride/gedcom/55gctoc.htm>
- [38] Ghostscript, Ghostview and GSview. <http://www.cs.wisc.edu/~ghost/>
- [39] Gibbons A. (1985) *Algorithmic Graph Theory*. Cambridge University Press.
- [40] Grossman J. (2002) The Erdős Number Project.
<http://www.oakland.edu/~grossman/erdoshp.html>

- [41] Hall, B.H., Jaffe, A.B. and Tratjenberg M.: The NBER U.S. Patent Citations Data File. NBER Working Paper 8498 (2001).
<http://www.nber.org/patents/>
- [42] Hamberger, K., Houseman, M., Daillant, I., White, D.R., Barry, L. (2004) *Matrimonial Ring Structures*. Math. & Sci. hum. / Mathematics and Social Sciences, **42(4)**, 83-119.
<http://www.ehess.fr/revue-msh/pdf/N168R965.pdf>
- [43] Hummon, N.P., Doreian, P. (1989) Connectivity in a citation network: The development of DNA theory. *Social Networks*, **11**, 39–63.
- [44] Jones B. (2002). Computational geometry database.
<http://compgeom.cs.uiuc.edu/~jeffe/compgeom/biblios.html>
- [45] Kejžar, N., Korenjak Černe, S., Batagelj, V. (2010) Network Analysis of Works on Clustering and Classification from Web of Science. Classification as a Tool for Research. Hermann Locarek-Junge, Claus Weihs eds. Proceedings of IFCS 2009. Studies in Classification, Data Analysis, and Knowledge Organization, Part 3, 525-536, Springer, Berlin, 2010. **Springer**
- [46] Kejžar, N., Nikoloski, Z., Batagelj, V. (2008): Probabilistic Inductive Classes of Graphs. *Journal of Mathematical Sociology* **32**: 85-109.
- [47] Kleinberg J. (1998) Authoritative sources in a hyperlinked environment. In Proc 9th ACM/SIAM Symposium on Discrete Algorithms, p. 668-677.
<http://www.cs.cornell.edu/home/kleinber/auth.ps>
- [48] Knuth, D. E. (1993) *The Stanford GraphBase*. Stanford University, ACM Press, New York. <ftp://labrea.stanford.edu/pub/sgb/>
- [49] Liu, J.S. and L.Y.Y. Lu (2012) An Integrated Approach for Main Path Analysis: Development of the Hirsch Index as an Example. *Journal of the American Society for Information Science and Technology*, 63(3), p. 528-542
<http://onlinelibrary.wiley.com/doi/10.1002/asi.21692/abstract>
- [50] LOCKS: CRA Analyses of News Stories on the Terrorist Attack. Arizona State University. <http://locks.asu.edu/terror/>
- [51] McCabe, T. Computer Science Approaches: Visualization Tools and Software Metrics. in Survey Automation. NAP, 2003, p. 116-136.
<http://books.nap.edu/books/0309089301/html/116.html>
- [52] MDL Information Systems, Inc. (2002) <http://www.mdli.com/>

- [53] James Moody home page (2002)
<http://www.soc.sbs.ohio-state.edu/jwm/>
- [54] Murtagh, F. (1985) Multidimensional Clustering Algorithms, *Compstat lectures*, **4**, Vienna: Physica-Verlag.
- [55] Pajek's datasets:
<http://vlado.fmf.uni-lj.si/pub/networks/data/>
- [56] D.M. Pennock et al. (2002) Winners don't take all, *PNAS*, **99**/8, 5207-5211.
- [57] Peterson J. L.: *Petri Net Theory and the Modeling of Systems*.
- [58] The R Project for Statistical Computing. <http://www.r-project.org/>
- [59] Richardson D.C., Richardson J.S. (2002) The Mage Page.
<http://kinemage.biochem.duke.edu/index.html>
- [60] Scott, J. (2000) *Social Network Analysis: A Handbook*, 2nd edition. London: Sage Publications.
- [61] Seidman S. B. (1983) Network structure and minimum degree, *Social Networks*, **5**, 269–287.
- [62] Tarjan, R. E. (1983) *Data Structures and Network Algorithms*. Society for Industrial and Applied Mathematics Philadelphia, Pennsylvania.
- [63] UCINET (2002) <http://www.analytictech.com/>
- [64] The United States Patent and Trademark Office.
<http://patft.uspto.gov/netahtml/srchnum.htm>
- [65] Wasserman S., Faust K. (1994). *Social Network Analysis: Methods and Applications*. Cambridge University Press, Cambridge.
- [66] White D.R., Batagelj V., Mrvar A. (1999) Analyzing Large Kinship and Marriage Networks with Pgraph and **Pajek**. *Social Science Computer Review*, **17** (3), 245-274
- [67] Wilson, R.J., Watkins, J.J. (1990) *Graphs: An Introductory Approach*. New York: John Wiley and Sons.
- [68] W3C SVG page. <http://www.w3.org/Graphics/SVG>
- [69] Zeleny, L.D. (1947): Selection of compatible flying partners. *American Journal of Sociology*, **52**, 424-431.