



NARZĘDZIA INFORMATYCZNE W BADANIACH NAUKOWYCH

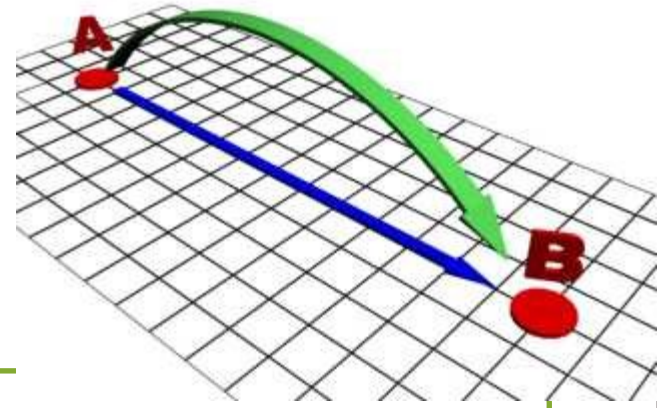
Biblioteka igraph

Prowadzący zajęcia:
dr hab. inż. Krystian Pyka, prof.
AGH

Student:
mgr inż. Beata Szafrńska



IGRAPH – TO PAKIET OPROGRAMOWANIA DLA KOMPLEKSOWYCH BADAŃ SIECI



Zdolność tworzenia dużych wykresów z milionami

wierzchołków i krawędzi

Realizacja
Algorytmów
w
wykresu

Nie ma innego
pakietu spełniającego
wszystkie nast.
wymagania:

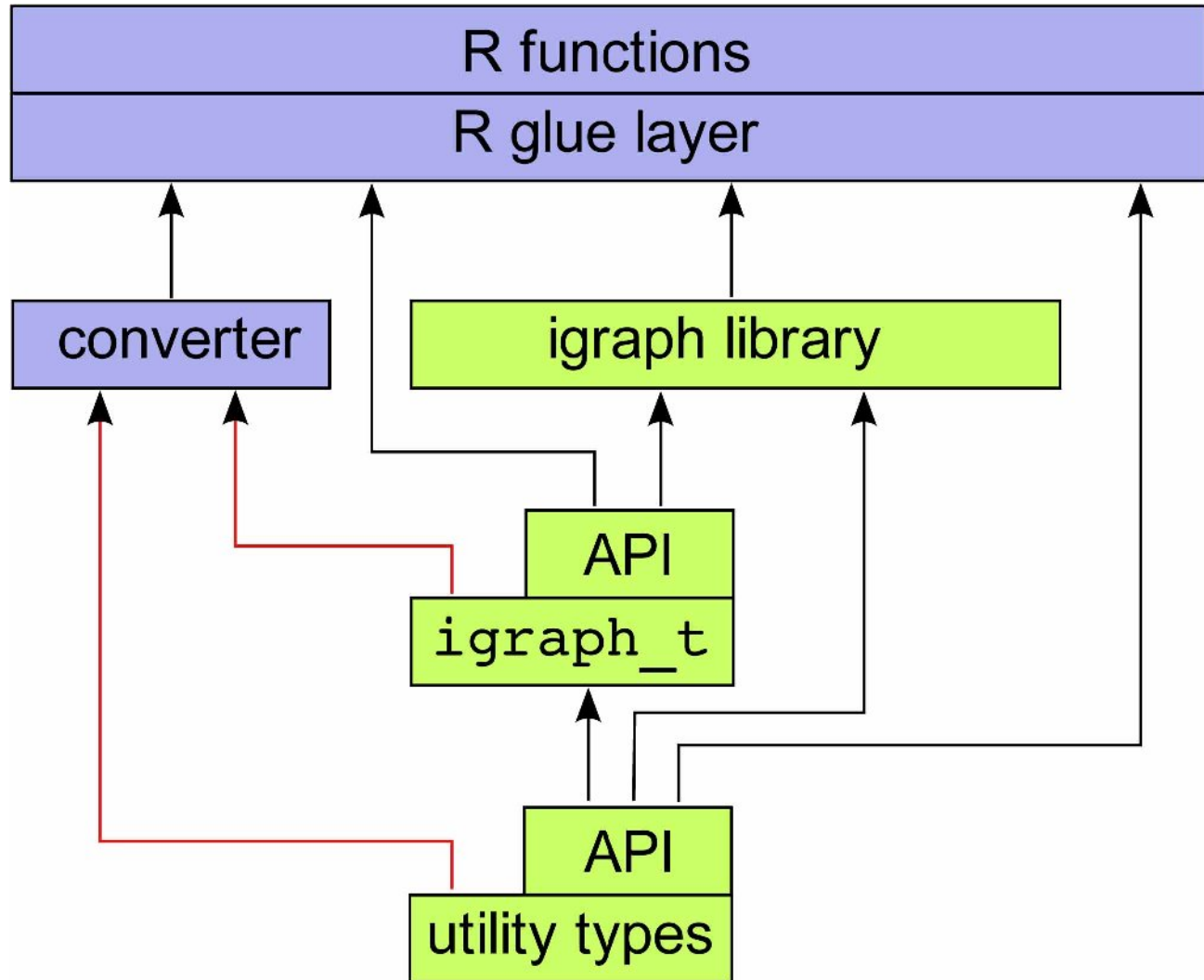
Umiejętność
do
szybkiego
prototypowa
nia

Wykorzystywany głównie w teorii grafów

(dział w matematyce i informatyce zajmujący się badaniem własności grafów. Informatyka rozwija również algorytmy wyznaczające pewne właściwości grafów. Algorytmy te stosuje się do rozwiązywania wielu zadań praktycznych, często w dziedzinach na pozór nie związanych z grafami.

Architektura

igraph

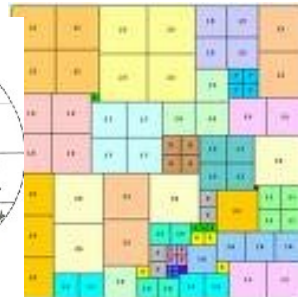
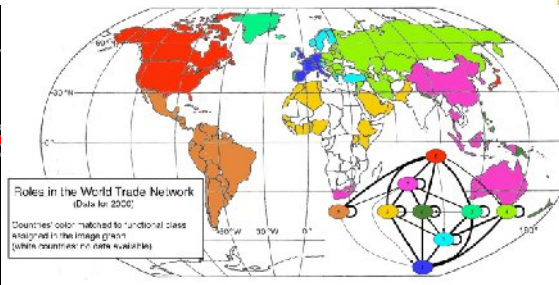
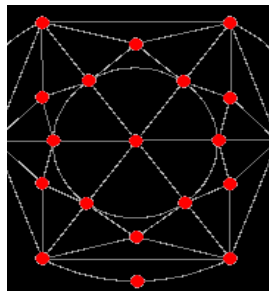
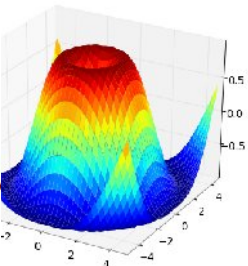
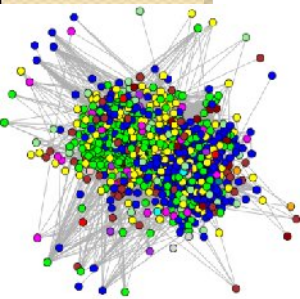


Instalacja

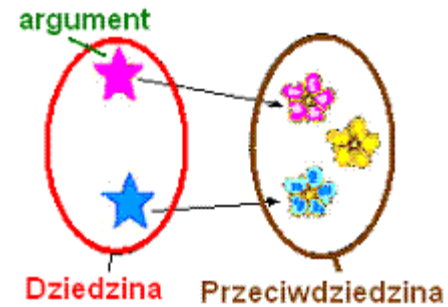
a

Pakiet R `igraph` może być instalowany podobnie jak pakiet R.

- Najprostszym sposobem jest użycie polecenia `install.packages()`. To pliki do pobrania ze źródła pakietu binarnego właściwe dla systemu począwszy od CRAN i instaluje się domyślnie do katalogów systemowych.
- Pakiet może być również załadowany przez wpisanie *biblioteki (igraph) po instalacji*.



Właściwości igraph



source. Zawsze można pobrać kod źródłowy i kompilować nowe funkcje. Rozprowadzany na licencji GPL. Wdrożenie igraph obejmuje najbardziej efektywne wykorzystanie algorytmów

portowalność. Ponieważ napisany niemal wyłącznie w ANSI C, działa na różnych platformach, w tym również odmianach Linuksa, FreeBSD, Mac OS X, Sun Solaris, a nawet Minix!

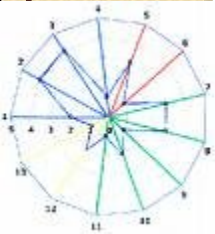
zawiera interfejsy do języków wyższego poziomu (GNU R, Python). Nowe interfejsy mogą być dodawane z łatwością.

zawiera procedury do manipulacji wykresami, dodawanie i usuwanie krawędzi i wierzchołków.

zawiera procedury do wizualizacji wykresów w pakiecie R oraz module Pythona na wiele sposobów, w 2D i 3D, interaktywnie lub nie interaktywnie.

zawiera procedury do czytania i pisania wielu formatów plików, np. GraphML, CSV, etc.

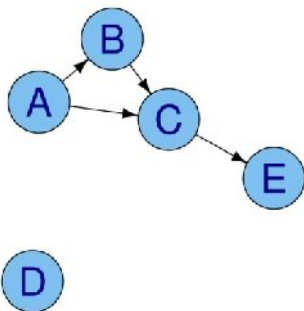
zawiera procedury do udokumentowania zarówno dla użytkowników jak



Co to jest graf?

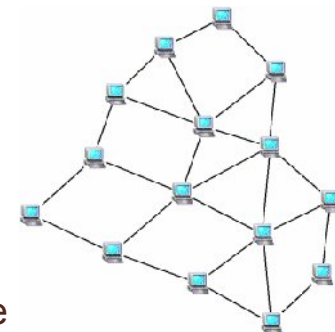
Graf jest zborem wierzchołków, które mogą być połączone krawędziami, w taki sposób, że każda krawędź kończy się i zaczyna w którymś z wierzchołków.

Wierzchołki grafu są numerowane i czasem stanowią reprezentację jakichś obiektów, natomiast krawędzie mogą wówczas obrazować relacje między takimi obiektami.



Krawędzie mogą mieć wyznaczony kierunek, a graf zawierający takie krawędzie jest **grafem skierowanym**.

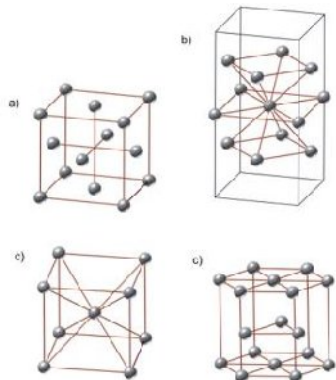
Krawędź może posiadać także wagę, tzn. przypisaną liczbę, która określa np. odległość między wierzchołkami (jeśli np. graf jest reprezentacją połączeń między miastami).



Sieć IP-MPLS Telekomunikacji Kolejowej Sp. z o.o.

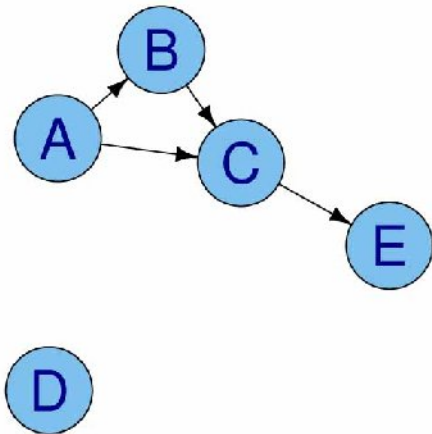
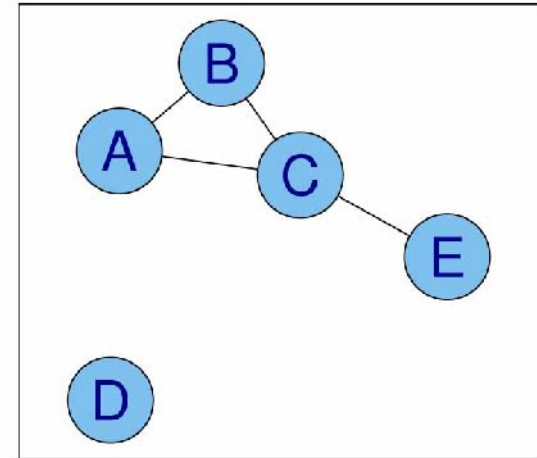


W grafie skierowanym wagi mogą być zależne od kierunku przechodzenia przez krawędź (np. jeśli graf reprezentuje trud poruszania się po jakimś terenie, to droga pod górkę będzie miała przypisaną większą wagę niż z górki).



Relacja pomiędzy wierzchołkiem a krawędzią.

vertices = {A,B,C,D,E}
edges = ({A,B}, {A,C}, {B,C},
{C,E}).



vertices = {A,B,C,D,E}
edges = ((A,B), (A,C), (B,C),
(C,E)).

Wykres

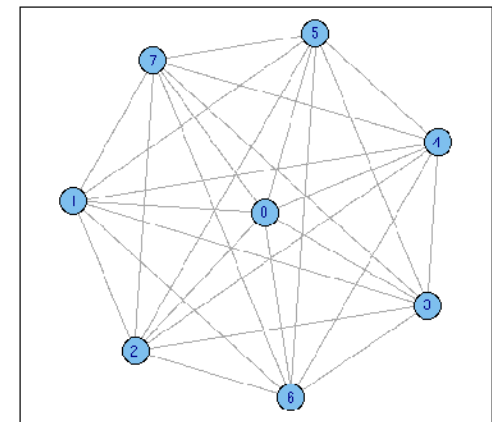
y

Puste wykresy – składają się jedynie z pustych wierzchołków, nie zawierają żadnych krawędzi.

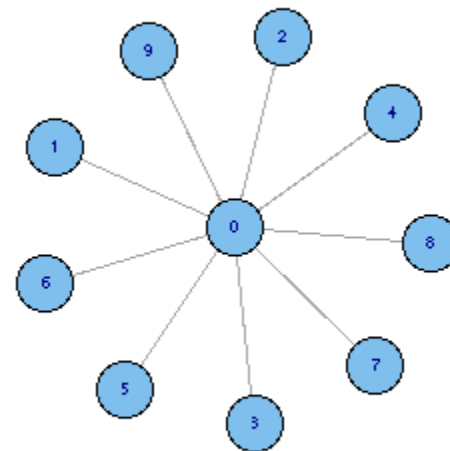
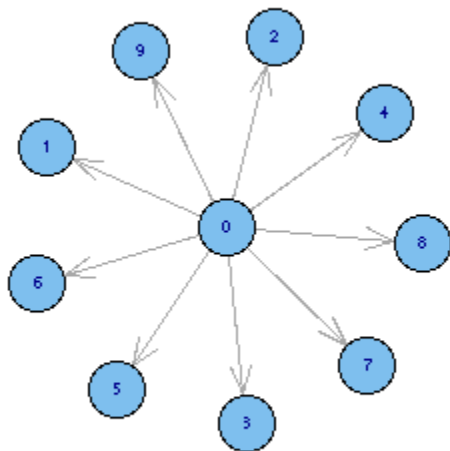
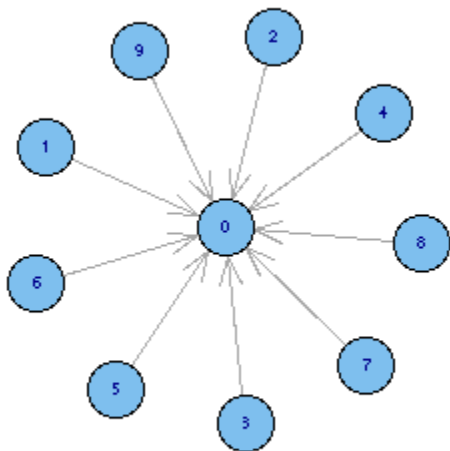
Jeśli stworzymy puste wykresy i przechowujemy je w zmiennej o nazwie `g`, to jeśli wpisujemy nazwę zmiennej zawierającej wykres `R` drukujemy informacje o wykresie i listę krawędzi.

`graph.empty()` – to polecenie tworzy pusty wykres.

Pełne wykresy – to takie, w których każde dwa wierzchołki są połączone krawędzią. Mogą być tworzone za pomocą funkcji `graph`.



wiazdy i pierścienie – ten wykres charakteryzuje się tym, że można odróżnić jeden wierzchołek który jest połączony z każdym innym wierzchołkiem



Funkcje:

mode decyduje który z wierzchołków generuje

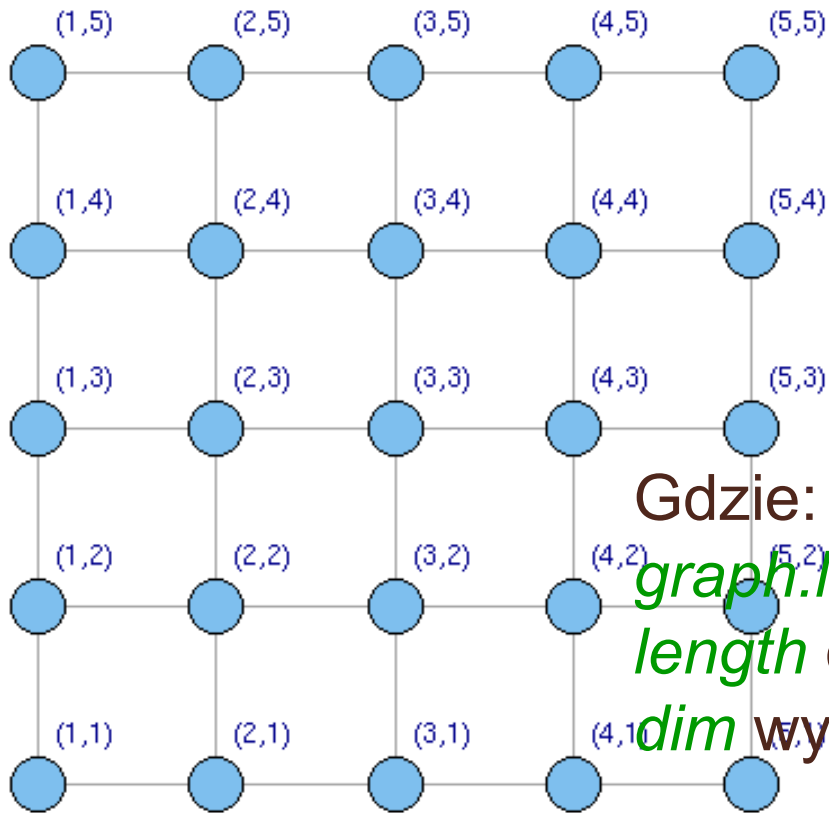
in tworzy wykres nadając kierunek rys.1

out tworzy wykres na zewnątrz rys.2

undirected nie nadaje kierunku

graph_ring() służy do tworzenia wykresów pierścieni

- sieć – n-wymiarowe wykresy, w której wierzchołki łączą się z sąsiednimi, oddalonymi o jedna jednostkę. Tzn. jeśli przestrzeń jest dwuwymiarowa np. 5x3 to ma 15 wierzchołków i są one umieszczone na współrzędnych (1,1),(1,2),(1,3),(2,1),...(5,3)



Np.

- L1 <- graph.lattice (length= 5, dim = 2)
- > l2 <- graph.lattice(c(5,5))
- > L2 <- graph.lattice (c (5,5))

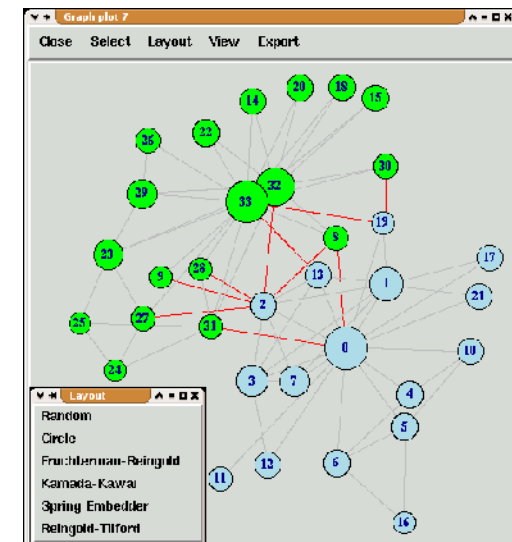
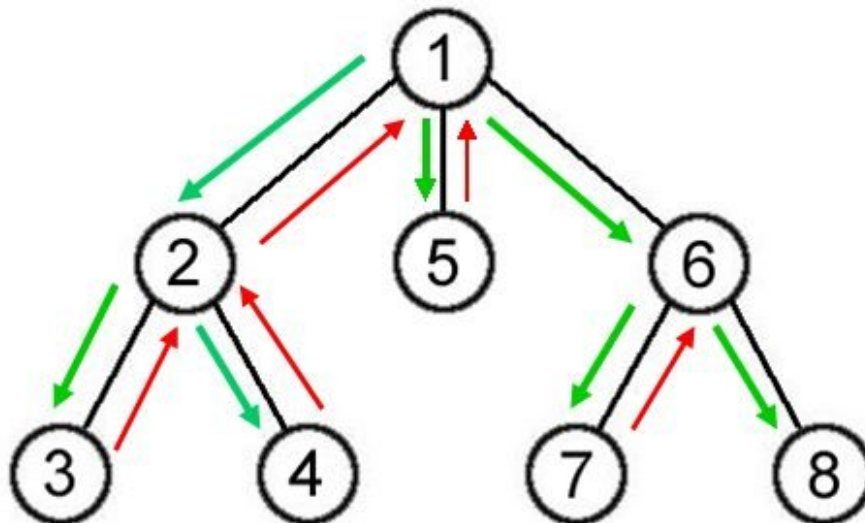
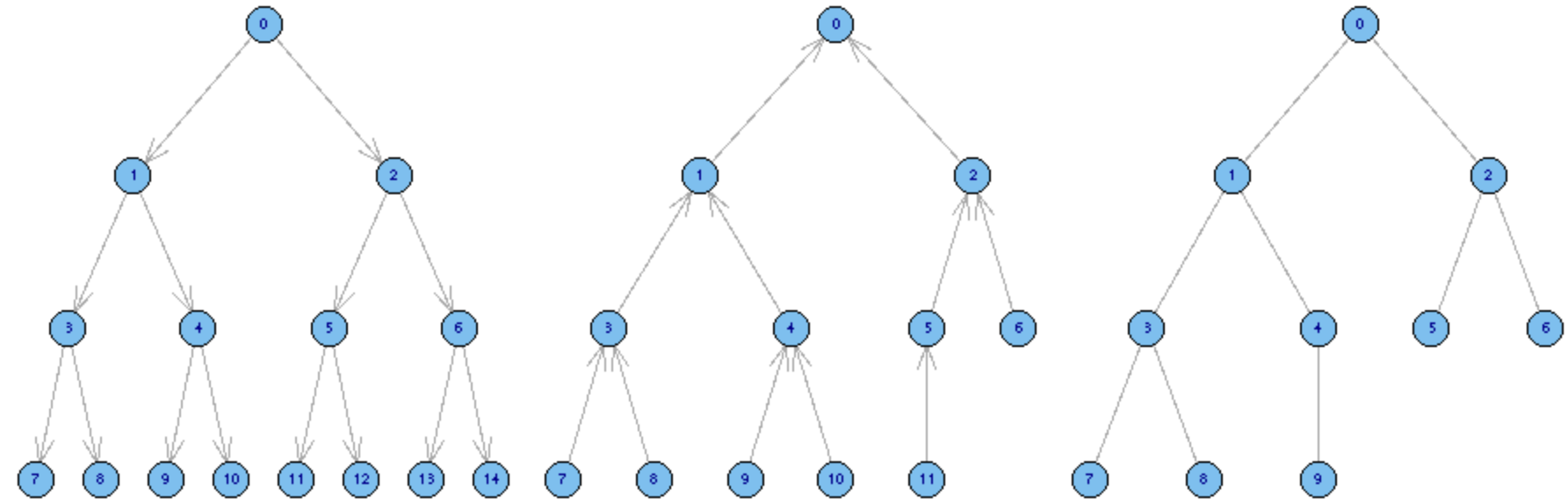
Gdzie:

graph.lattice() generowanie ogrodzenia

length odległość między wierzchołkami

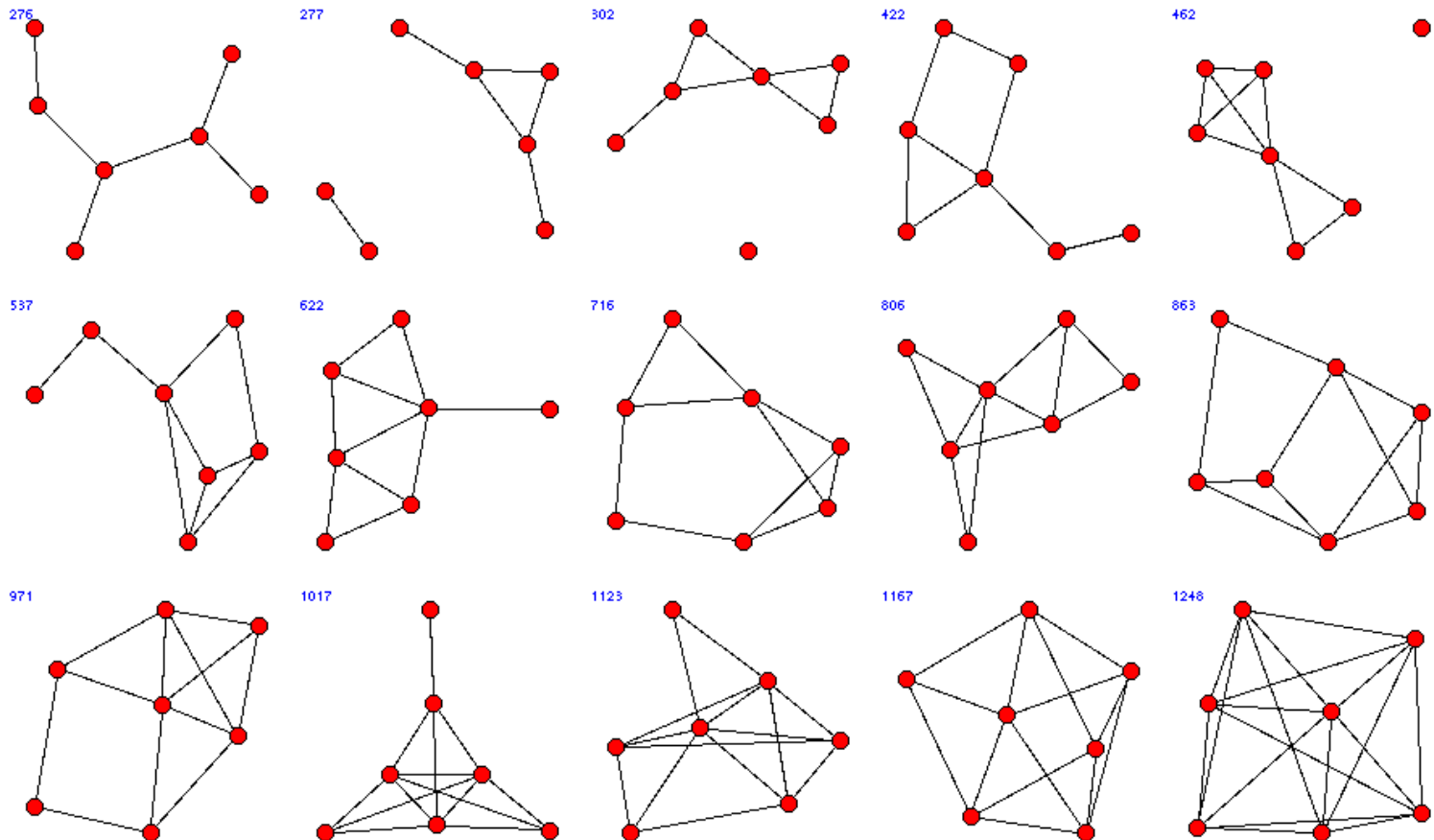
dim wymiar parametrów

□ **drzewa** – igragh może służyć do tworzenia drzew za pomocą funkcji *graph.tree()*. Z każdego wierzchołka drzewa można dotrzeć do każdego innego wierzchołka (spójność).



□ wykres Atlas – są tworzone w celu:

- Zwiększenia liczby węzłów
- Dla określonej liczby węzłów, w celu zwiększenia liczby krawędzi
- Dla stałych numerów wierzchołków i krawędzi, w porządku rosnącym np. $111223 < 112222$
- Dla stałych sekwencji



Tworzenie wykresów

grap

Wierzchołki są zawsze numerowane od zera!!!!
Numeracja jest ciągła od 0 do n-1

h

```
> g <- graph( c(0,1, 1,2, 2,3, 3,4), n=6, directed=TRUE )
```

```
> g
```

Vertices: 6

Edges: 4

Directed: TRUE

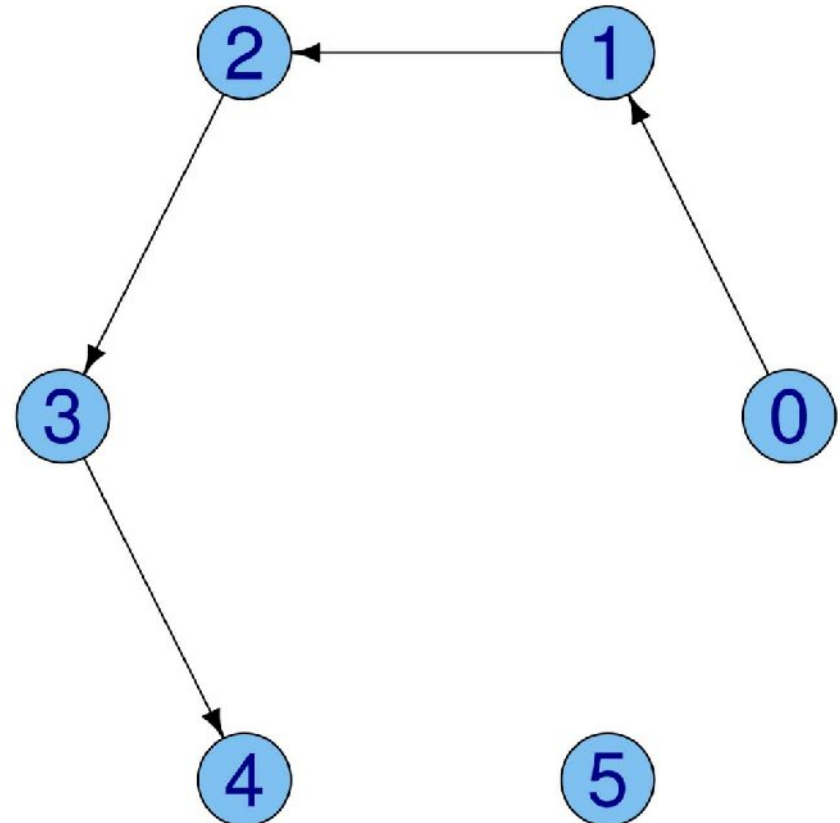
Edges:

[0] 0 -> 1

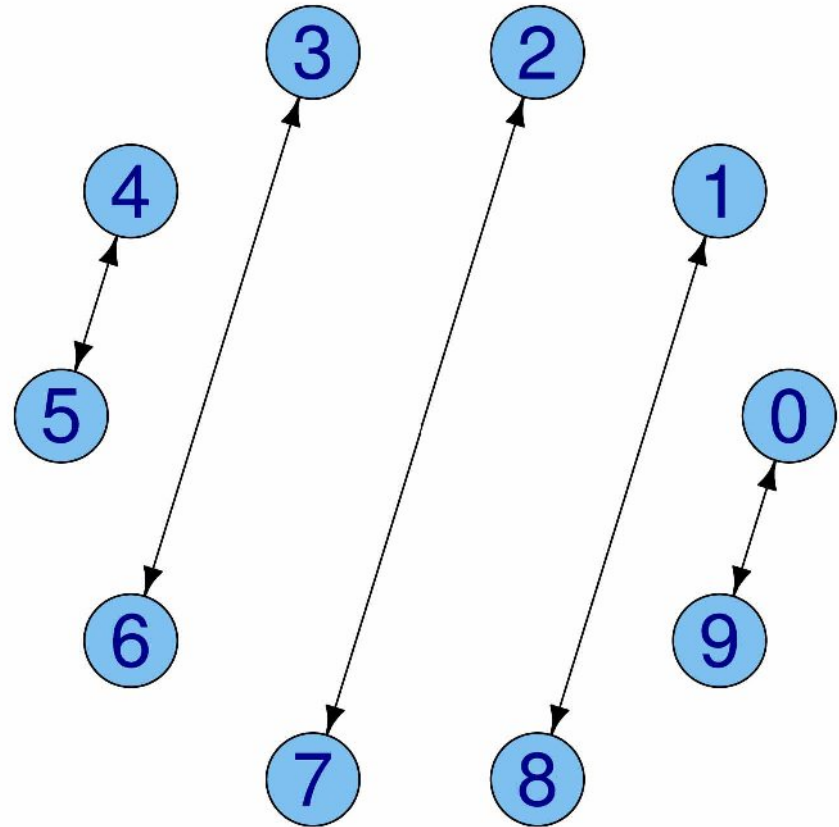
[1] 1 -> 2

[2] 2 -> 3

[3] 3 -> 4

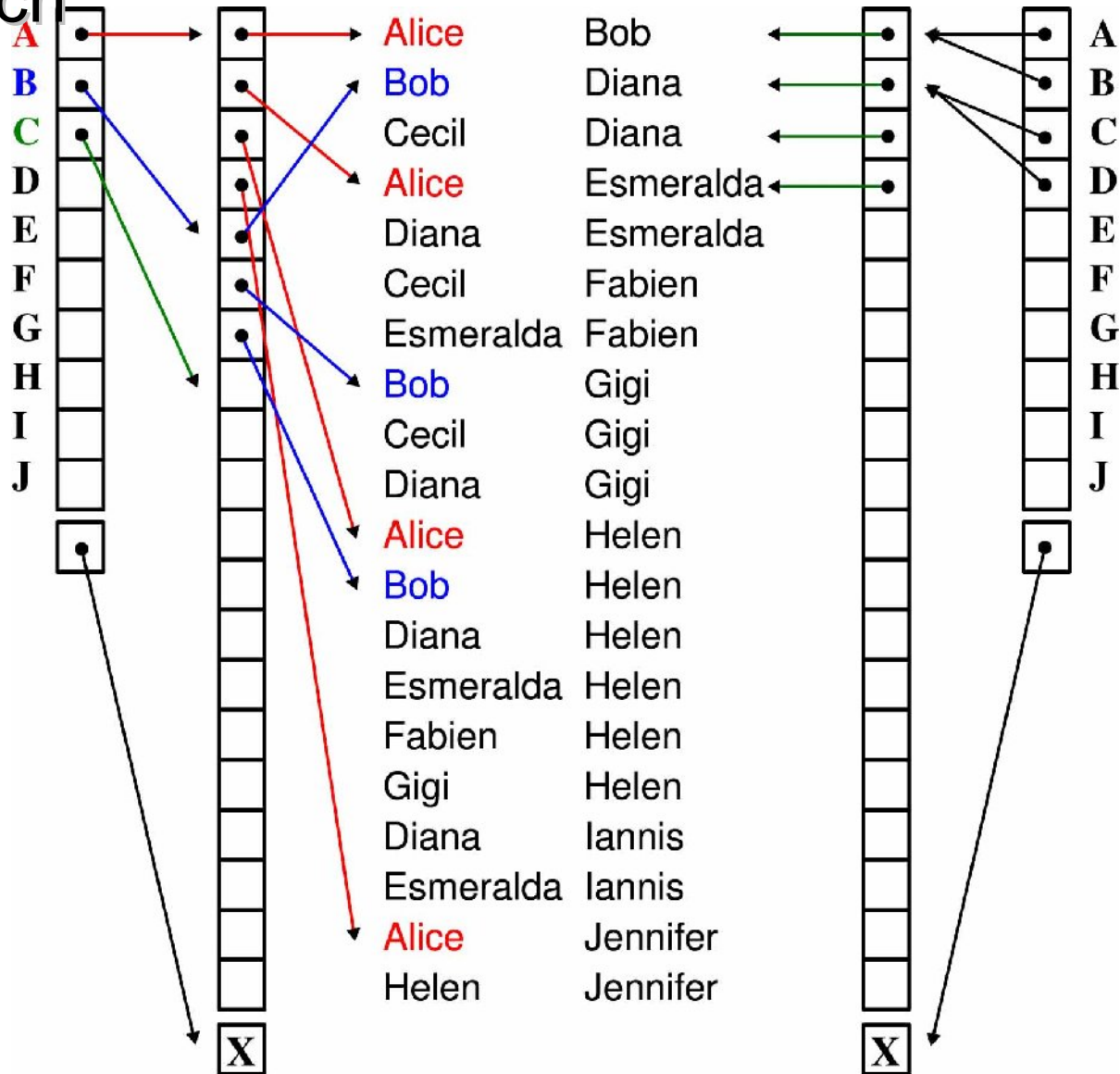


```
el <- cbind(0:9, 9:0)
> g <- graph( t(el), directed=TRUE)
> g
Vertices: 10
Edges: 10
Directed: TRUE
Edges:
[0] 0 -> 9
[1] 1 -> 8
[2] 2 -> 7
[3] 3 -> 6
[4] 4 -> 5
[5] 5 -> 4
[6] 6 -> 3
[7] 7 -> 2
[8] 8 -> 1
[9] 9 -> 0
```



Płaskie struktury

danych



Graph.formul

^a
Funkcja ta jest przydatna do szybkiego tworzenia małych wykresów

```
g <- graph.formula( Alice-Bob-Cecil-Alice,  
Daniel-Cecil-Eugene, Cecil-Gordon )
```

```
> g
```

```
Vertices: 6
```

```
Edges: 6
```

```
Directed: FALSE
```

```
Edges:
```

```
[0] Alice -- Bob
```

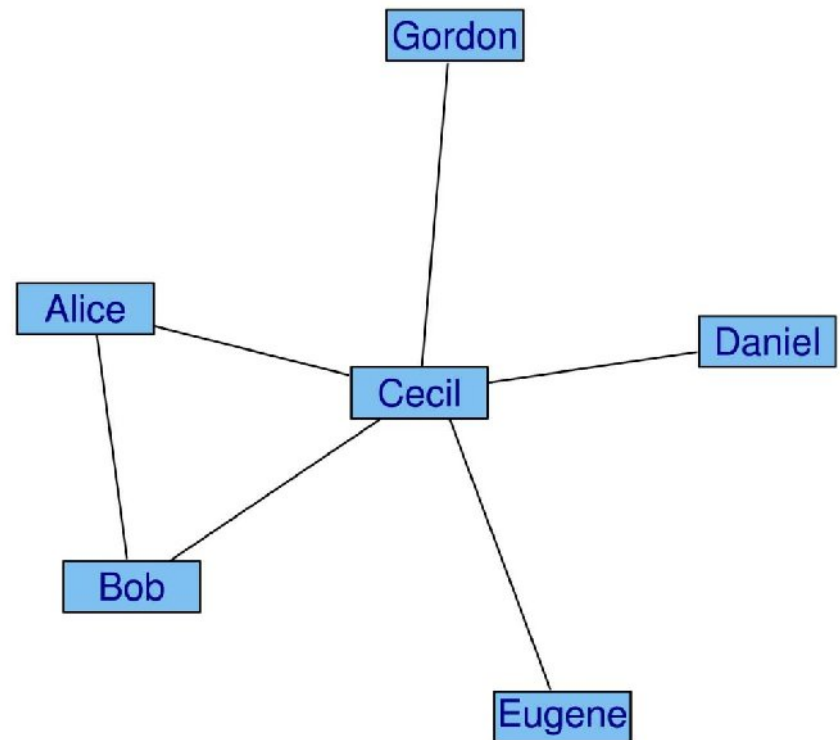
```
[1] Bob -- Cecil
```

```
[2] Alice -- Cecil
```

```
[3] Cecil -- Daniel
```

```
[4] Cecil -- Eugene
```

```
[5] Cecil -- Gordon
```



Graph.famou

S

graph.famous("Cubical")

Vertices: 8

Edges: 12

Directed: FALSE

Edges:

[0] 0 -- 1

[1] 1 -- 2

[2] 2 -- 3

[3] 0 -- 3

[4] 4 -- 5

[5] 5 -- 6

[6] 6 -- 7

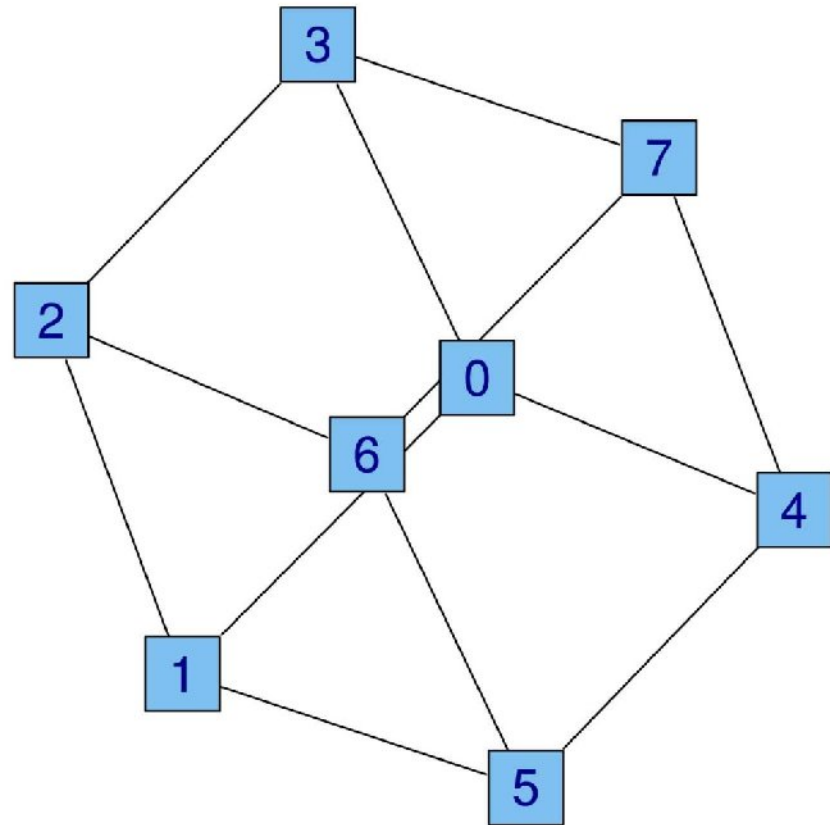
[7] 4 -- 7

[8] 0 -- 4

[9] 1 -- 5

[10] 2 -- 6

[11] 3 -- 7



graph.data.frame

Tworzenie wykresów igraph z ramkami danych
Funkcja ta tworzy wykres igraph z jednego lub dwóch ramek danych zawierających atrybuty wierzchołków.

```
> traits <- read.csv("traits.csv", head=F)
```

Traits

	V1	V2	V3
1	Alice Anderson	48	F
2	Bob Bradford	33	M
3	Cecil Connor	45	F
4	David Daugher	34	M
5	Esmeralda Escobar	21	F
6	Frank Finley	36	M
7	Gabi Garbo	44	F
8	Helen Hunt	40	F
9	Iris Irving	25	F
10	James Jones	47	M

```
> colnames(traits) <- c("name", "age", "gender")
```

```
> traits[,1] <- sapply(strsplit(as.character(traits[,1]), " "), "[", 1)
```

```
> relations <- read.csv("relations.csv", head=F)
```

```
> relations
```

	V1	V2	V3	V4	V5
1	Bob	Alice	N	4	4
2	Cecil	Bob	N	5	5
3	Cecil	Alice	Y	5	5
4	David	Alice	N	3	4
5	David	Bob	N	4	2
6	Esmeralda	Alice	Y	4	3
7	Frank	Alice	N	3	2
8	Frank	Esmeralda	N	4	4
9	Gabi	Bob	Y	5	5
10	Gabi	Alice	N	3	0
11	Helen	Alice	N	4	1
12	Iris	Cecil	N	0	1

```
...
```

```
> colnames(relations) <- c("from", "to", "same.room",  
"friendship", "advice")
```

```
> orgnet <- graph.data.frame(relations, vertices=traits)
```

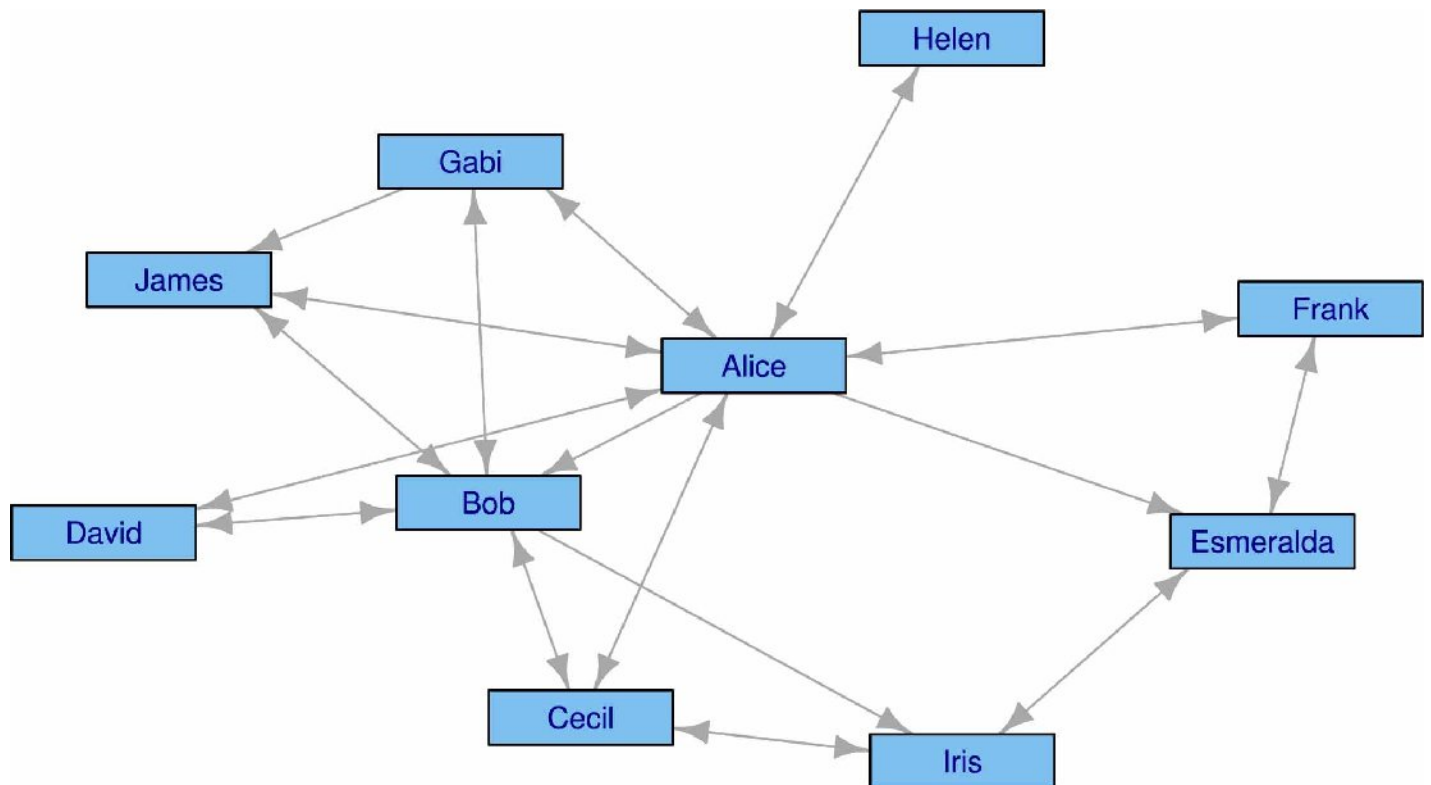
```
> summary(orgnet)
```

Vertices: 10

Edges: 34

Directed: TRUE

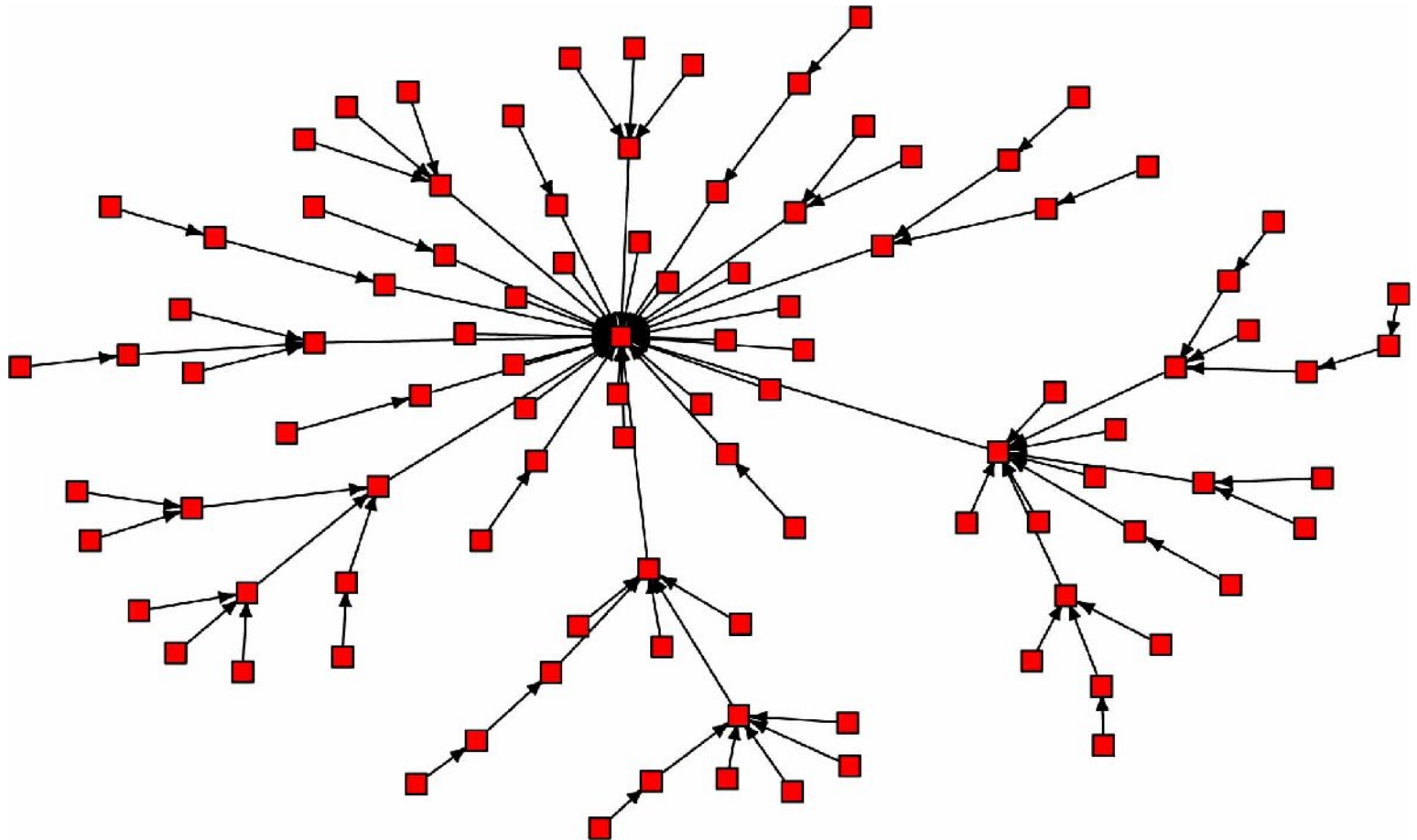
```
> plot(orgnet, layout=layout.kamada.kawai, vertex.label=V(orgnet)$name,  
vertex.shape="rectangle", vertex.size=20, asp=FALSE)
```



Tworzenie wykresów, wykresy losowe.

```
> ba <- ba.game(100, power=1, m=1)
```

```
> plot(ba, vertex.size=3, vertex.label=NA, asp=FALSE, vertex.shape="square",  
layout=layout.fruchterman.reingold, edge.color="black",edge.arrow.size=0.5)
```



Praca z nieco większym wykresem.

```
> is.connected(jg) # Is it connected?
```

```
[1] FALSE
```

```
> no.clusters(jg) # How many components?
```

```
[1] 4881
```

```
> table(clusters(jg)$csize) # How big are these?
```

```
  1    3    4 25389
4871  8    1    1
```

```
> max(degree(jg, mode="in")) # Vertex degree
```

```
[1] 248
```

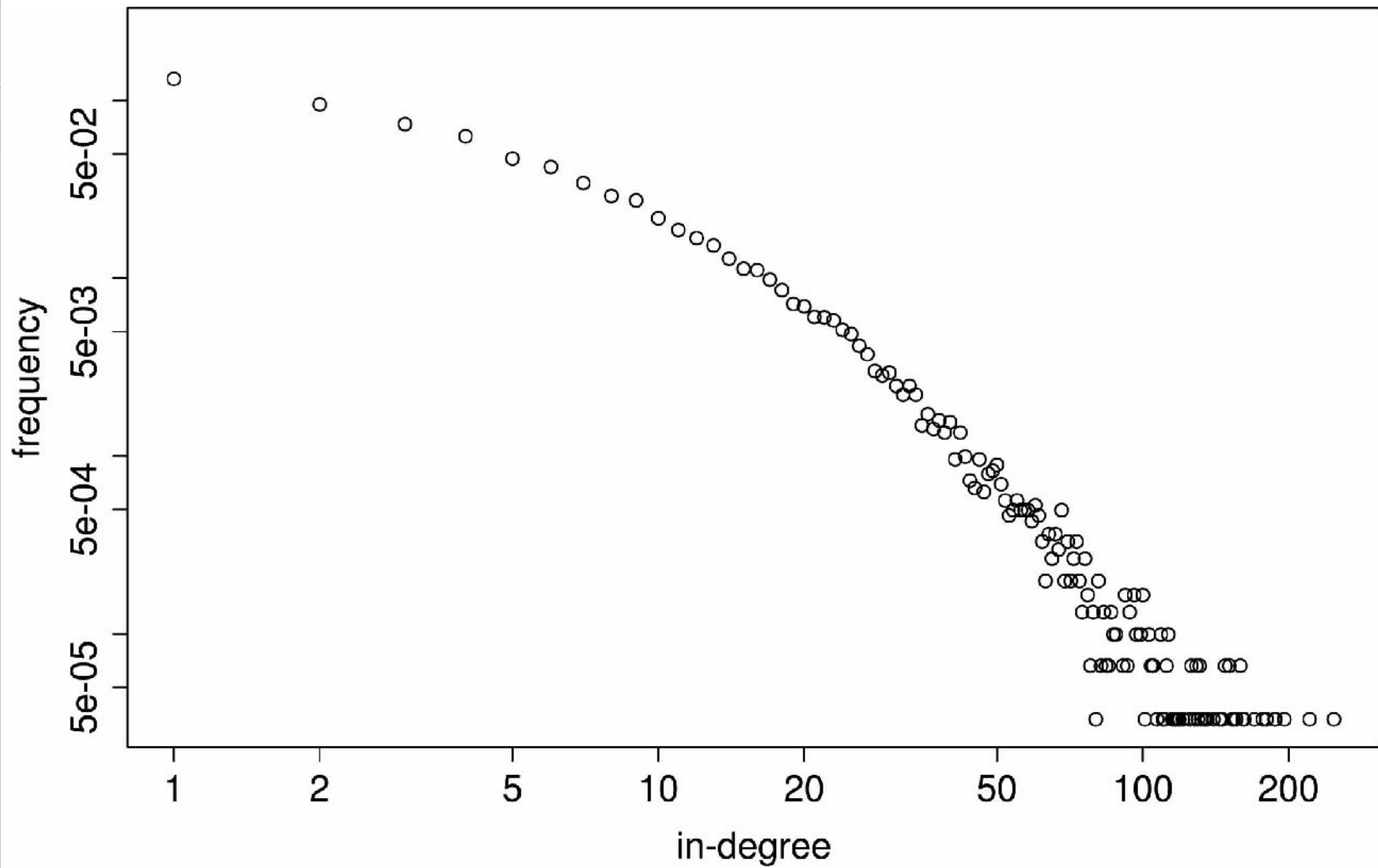
```
> max(degree(jg, mode="out"))
```

```
[1] 195
```

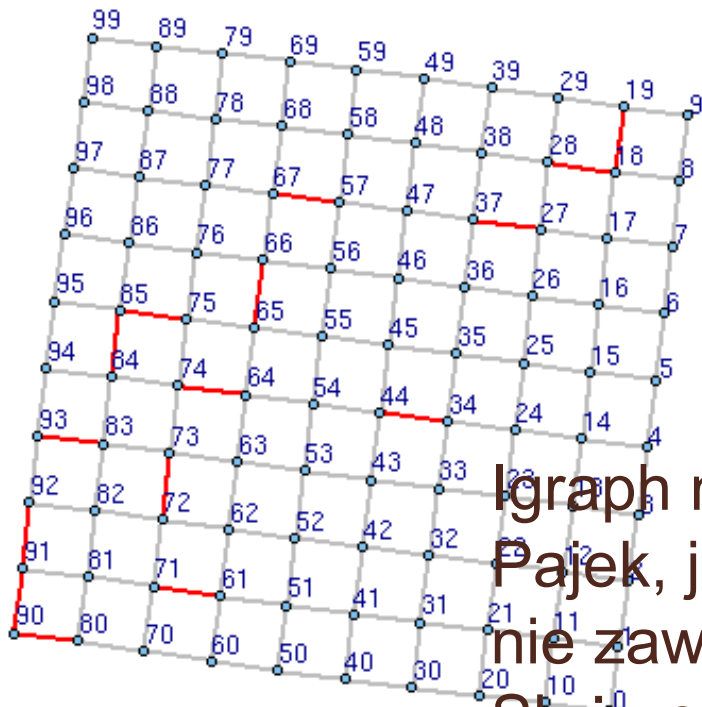
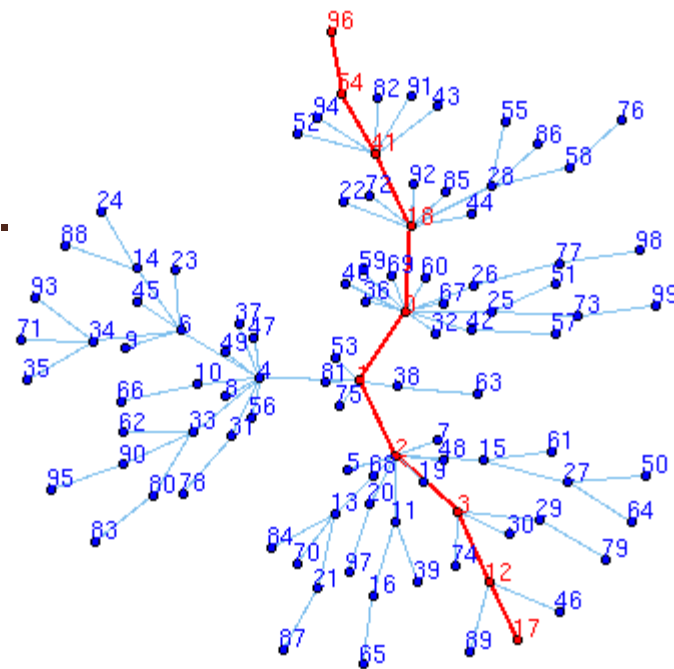
```
> max(degree(jg, mode="all"))
```

```
[1] 313
```

```
> plot(degree.distribution(jg, mode="in"), log="xy")
```

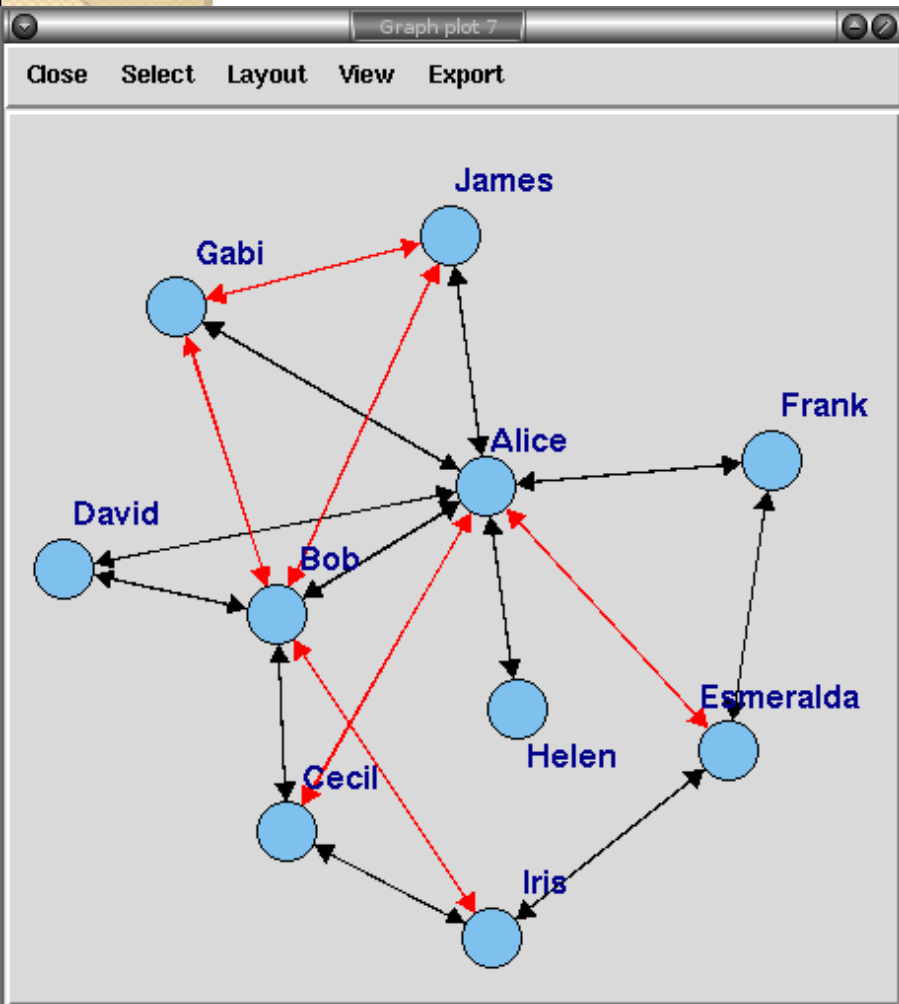


Igraph może być wykorzystany do obliczania najkrótszej trasy, oraz maksymalnego przepływu sieci.



Igraph mogą odczytywać i zapisywać pliki Pajek, jednak nazwy atrybutów nie zawsze są takie same jak w pliku Pajek. Służy on do analizy sieci w systemie Windows.

Macierz sąsiedztwa (matryce przylegania)



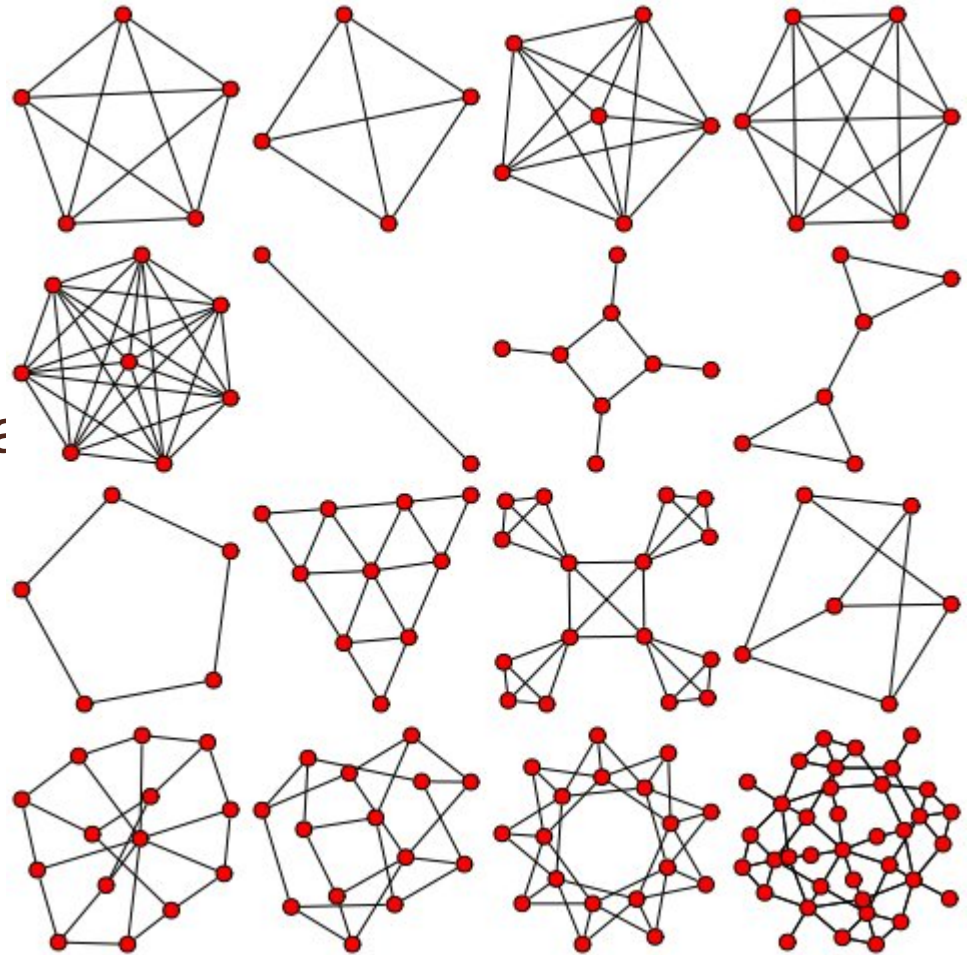
ymiarowa macierz, reprezentowana
licę M o wymiarach $[0...V]$ na $[0...V]$,
to liczba wierzchołków grafu.

między wierzchołkami v_i i v_j jest
to $M[i][j]=1$, w przeciwnym wypadku

graph.adjacency () służy do tworzenia
kresów z matrycami przylegania

Zastosowanie igraph

Generowanie wyglądu
algorytmu
Fruchterman'a-Reingold'a
nawet dla kilku tysięcy
wierzchołków.
W tym przykładzie
używamy
pakietu R i pakietu Kair.



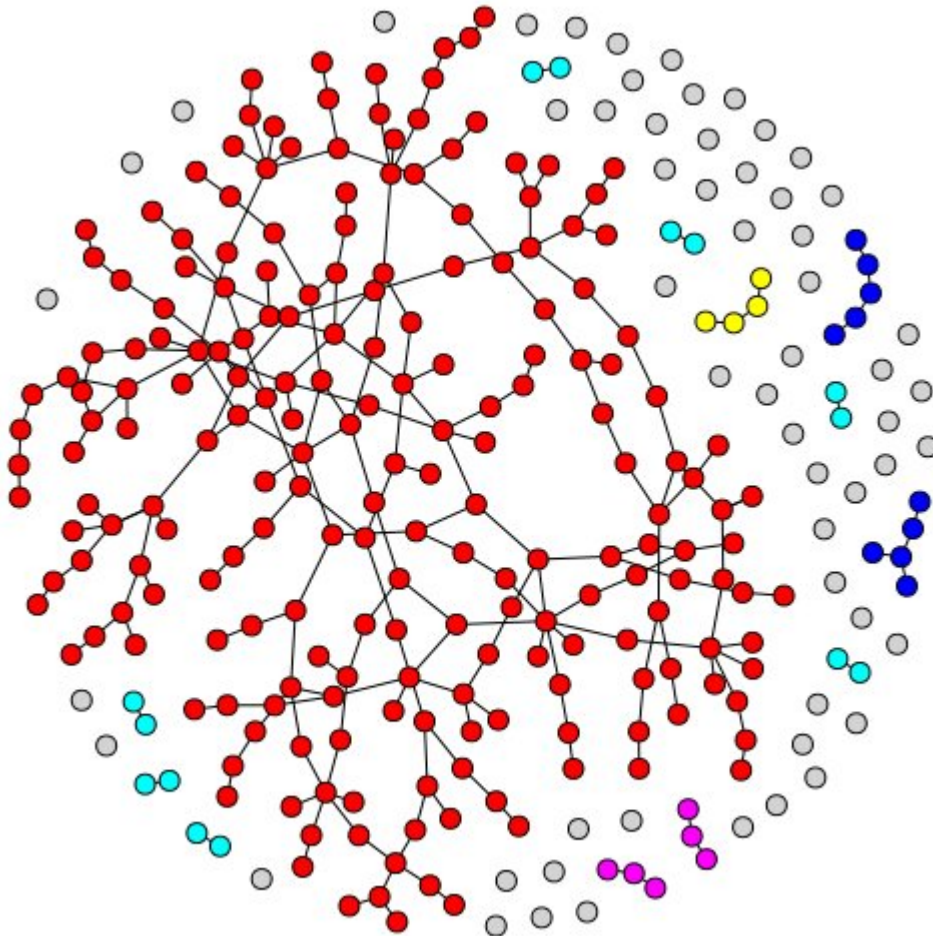
Zastosowanie

igraph

grafów losowych mających stałą liczbę wierzchołków (podane przez parametr n) oraz stałą liczbę krawędzi (podane przez parametr m), które utworzone są pomiędzy parami wierzchołków z jednakowym

obieństwem.

W momencie stopniowego zwiększania parametru m wierzchołki są odłączane.

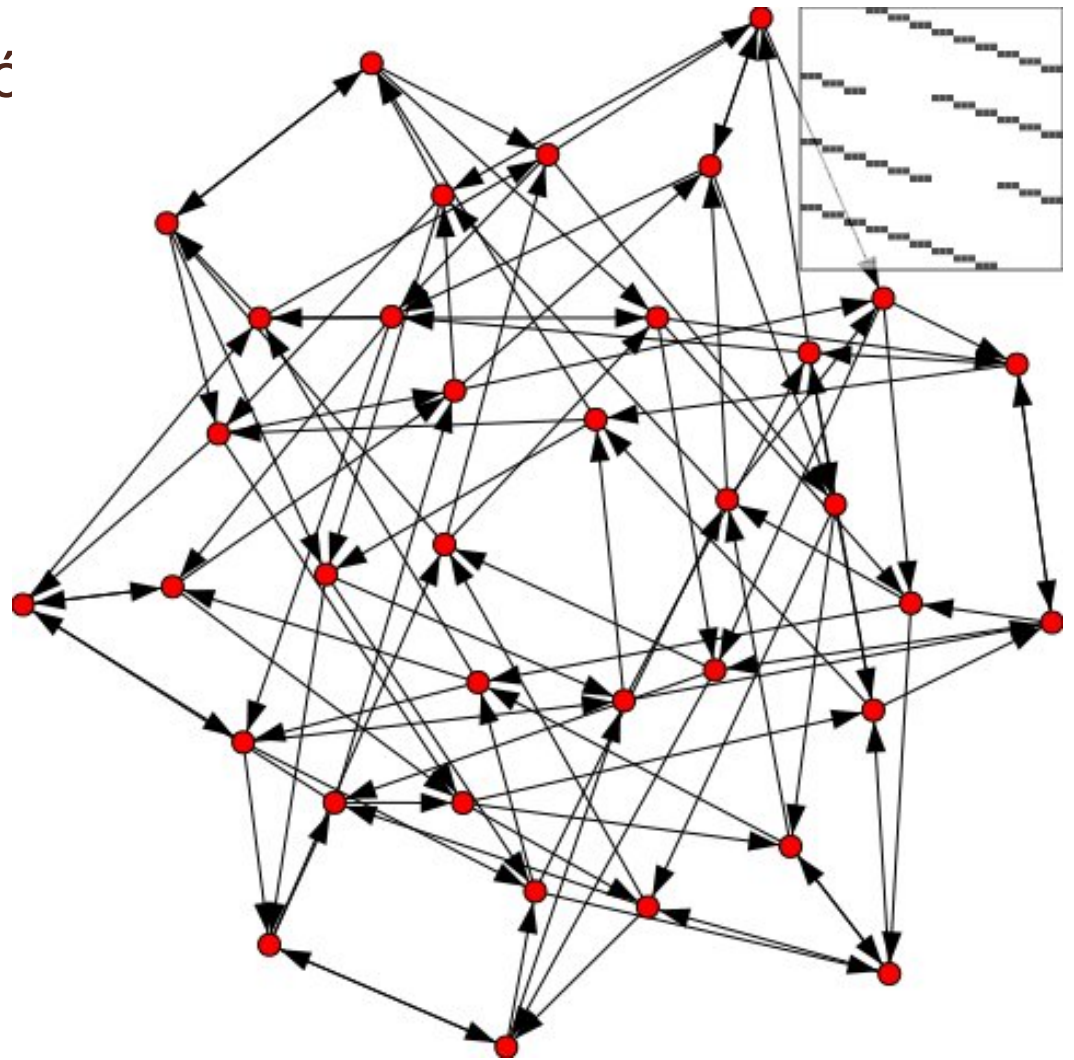


Zastosowanie

igraph

Wykres Kautz'a i macierz przylegania

Generowanie wykresów Kautz'a wraz z macierzą przylegania do rozszyfrowania wewnętrznej struktury wykresu.



Wykres

Zastosowanie

igraph

wierzchołkach.

Dwa wierzchołki są połączone ze sobą wtedy i tylko wtedy, jeśli ich odległość jest mniejsza niż 0,2.

Szerokość krawędzi

jest

proporcjonalna

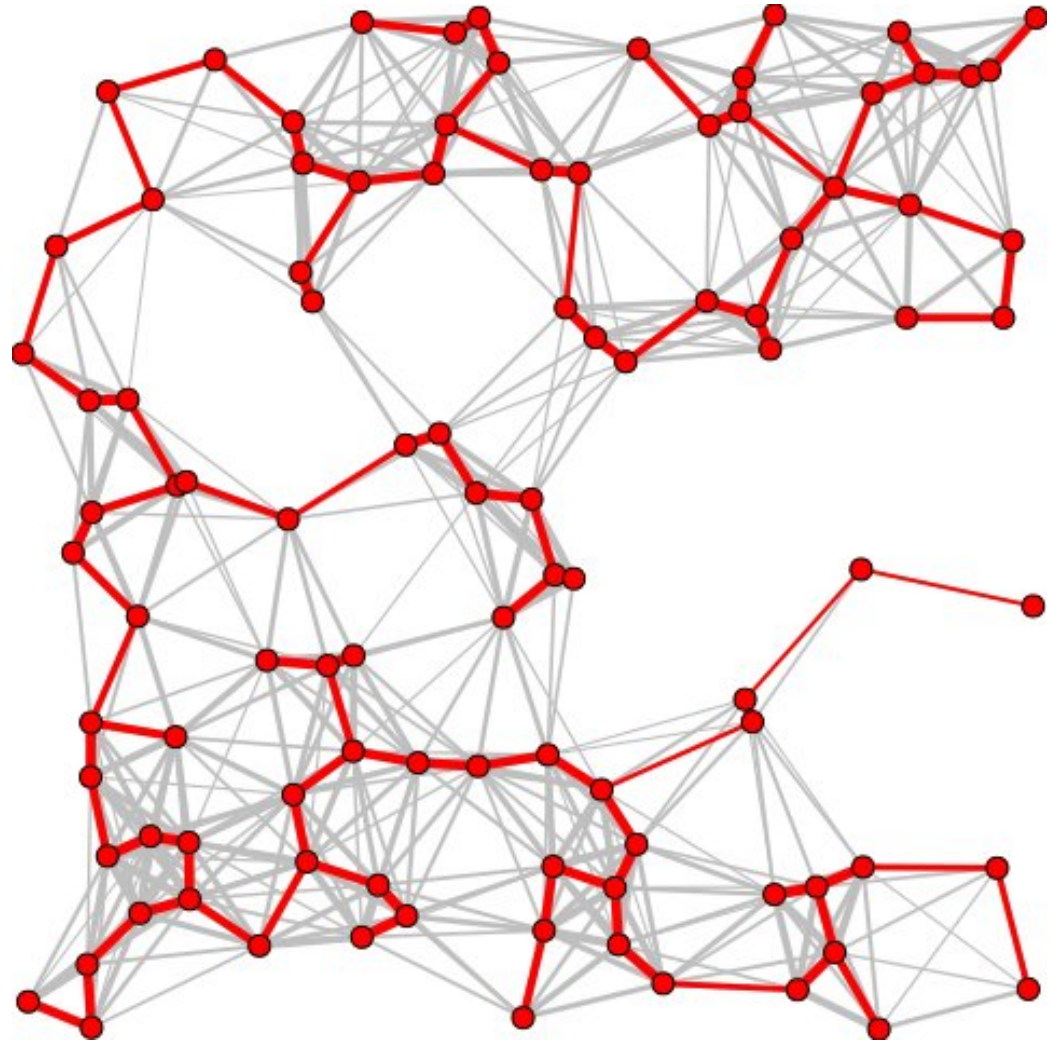
do bliskości

punktów

końcowych

(bliższe są

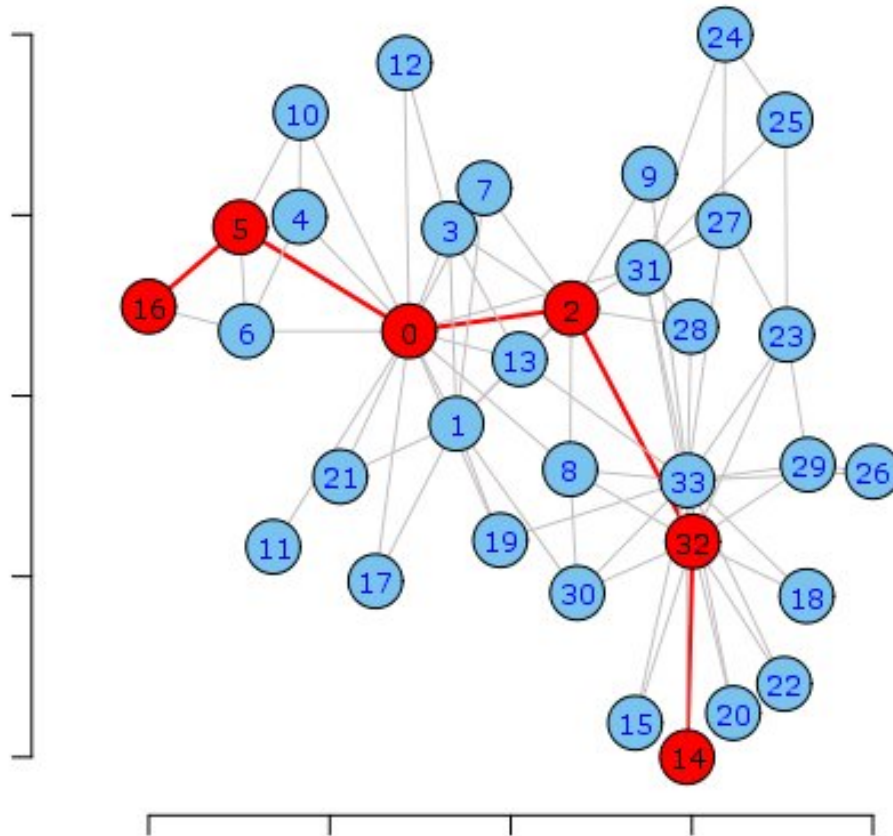
grubsze).



Zastosowanie

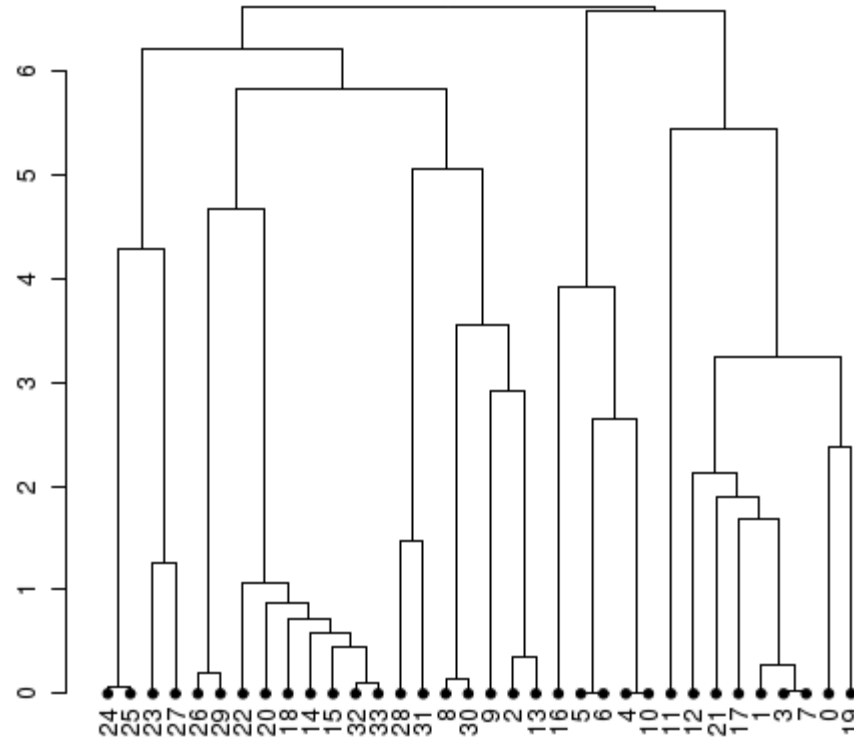
inrah

Diameter of the Zachary Karate Club network



Ten przykład pokazuje, jak można wybrać „ścieżkę” wśród wierzchołków.

Zastosowanie igraph



ie dendrogramu, tj. diagramu w kształcie drzewa ukazującego
pomiędzy wybranymi elementami na podstawie przyjętego kryterium
stywany do przedstawienia zależności i podobieństw.
za się je na podstawie przekształceń macierzy danych.

Funkcj

Igraph posiada 117 funkcji

e

alpha centrality oblicza centralne alfa niektórych (lub wszystkich) wierzchołków w grafie

arpack obliczanie wektorów ARPACK

atrybuty określa atrybuty należące do wykresu, wierzchołków lub krawędzi.

ba.game model do prostego stochastycznego algorytmu budowania wykresu

cohesive.block oblicza spójne boki dla obiektów klasy igraph

decompose.graph tworzy osobny wykres dla każdego elementu wykresu

degree.sequence.game funkcja szczególnie przydatna do tworzenia wykresu danego stopnia wierzchołków

graph.adjacency funkcje do tworzenia „igraph” z matrycami przylegania

graph.constructors tworzenie (głównie regularnych) wykresów: wykresy puste, wykresy z danej krawędzi, wykresy z matrycami przylegania, wykresy star, ogrodzenia, pierścienie, drzewa.

graph.data.frame funkcja do tworzenia wykresów igraph z jednej lub dwóch ramek danych zawierających atrybuty wierzchołków

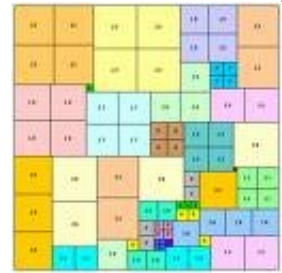
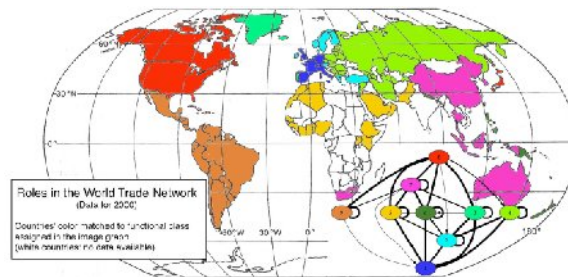
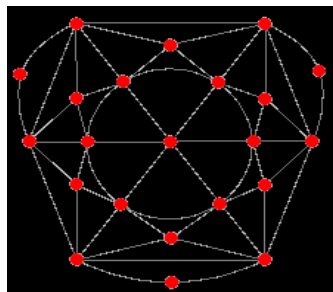
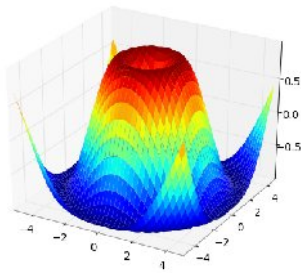
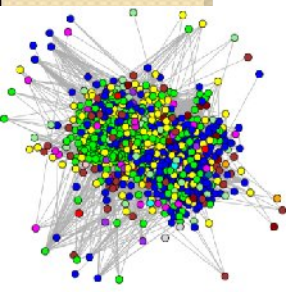
graph.de.bruijn wykresy reprezentujące powielanie łańcuchów

graph.density wykres gęstości jako stosunek liczby krawędzi do liczby możliwych krawędzi

graph.kautz wykresy kautz jako wykresy reprezentujące powielanie łańcuchów

graph.laplacian wykres laplacian

graph.structure dodawanie i usuwanie krawędzi i wierzchołków na wykresie



Funkcj

e
graphnel konwersja pomiędzy igraph i obiektów graphnel

grg.game geometria losowych grafów

igraph.par ustalanie parametrów dla pakietu igraph

igraph.vertex.shapes tworzenie różnych kształtów wierzchołków podczas kreślenia wykresów

independent.sets funkcja szuka niezależne zestawy wierzchołków w nieskierowanych wykresach, zestawy wierzchołków nazywamy niezależnymi, jeśli nie mają krawędzi między dwoma wierzchołkami.

is.graph zapytanie czy obiekt jest wykresem

layuot.merge połączenie kilku wykresów w jednym układzie

preference.game generowanie losowych wykresów w oparciu o różne rodzaje wierzchołków

read.graph odczytywanie wykresów

rglplot kreślenie wykresów w 3d z opengl

shortes.paths obliczanie najkrótszej drogi między wierzchołkami

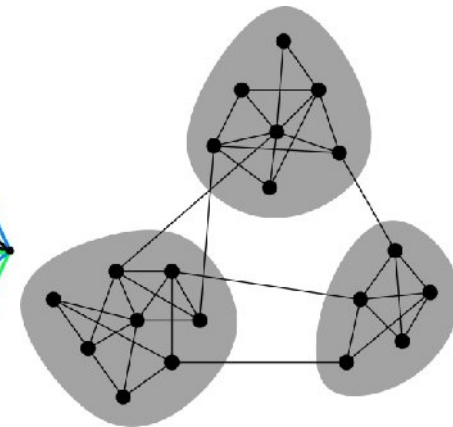
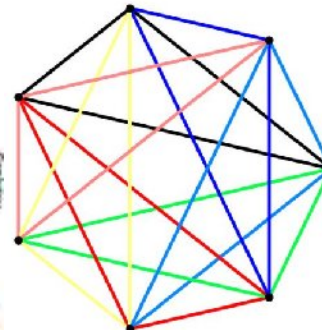
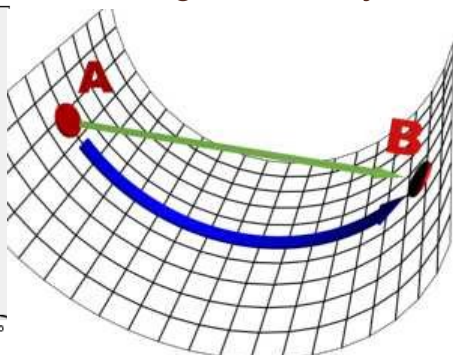
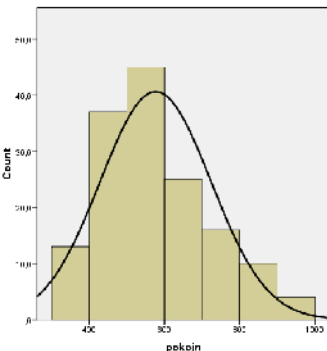
struktura.info określenie struktury sieci: liczba wierzchołków i krawędzi, sprawdzanie sąsiedztwa węzłów, czy dwa wierzchołki są połączone krawędzią.

tkplot tworzenie interaktywnych wykresów

topological.sort sortowanie topologiczne wierzchołków w grafie

vertex.connectivity łączenie wierzchołków wykresu

write.graph ogólna funkcja eksportowania wykresów



doktorska

**yczne zastosowanie teorii grafów Berge'a do komputerowego
ania i rozwiązywania zagadnień z zakresu ochrony środowiska'**

ż. Sławomir Leszczyński, AGH, 2000r.

ne katasralne jako baza do rozbudowy modelu sieci drogowej”.

ż. hab. Elżbieta Lewandowicz. Katedra Geodezji Szczegółowej,

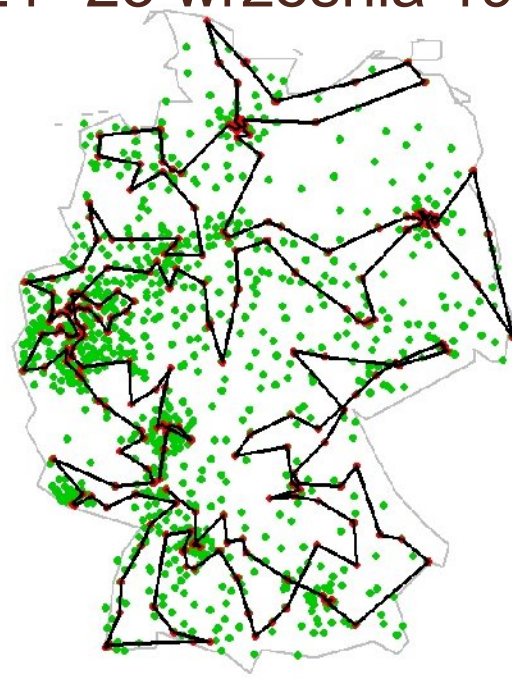
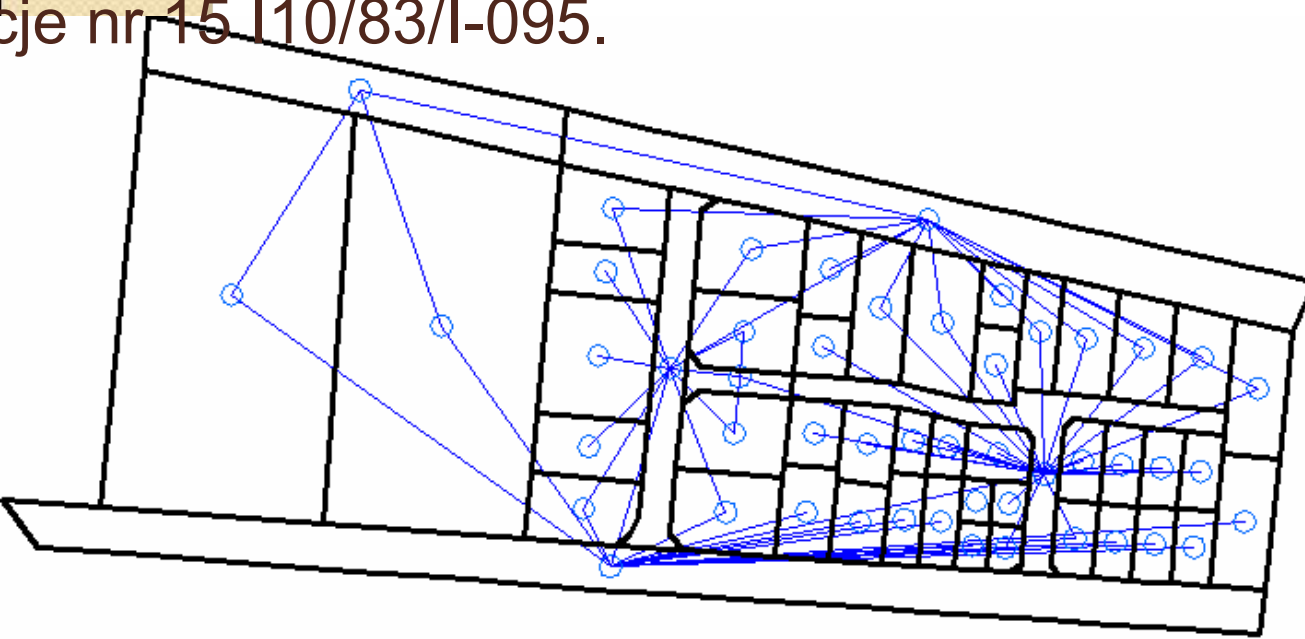
M w Olsztynie

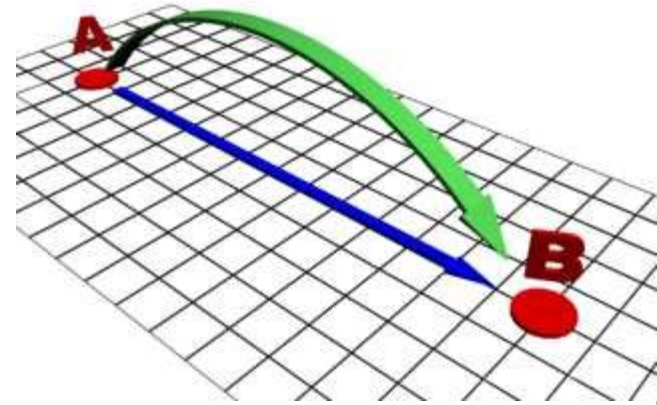
wanie teorii grafów w sieciach niwelacyjnych”, Józef Woźniak.

teotechniki Politechniki Wrocławskiej, Geodezja w pracach

kich. VIII Jesienna Szkoła Geodezji. Wrocław, 21- 23 września 1982r.

cje nr 15 I10/83/I-095.





DZIĘKUJĘ ZA UWAGĘ

mgr inż. Beata Szafrńska

