Analysis of weighted multiway networks

Cores in multi-relational networks

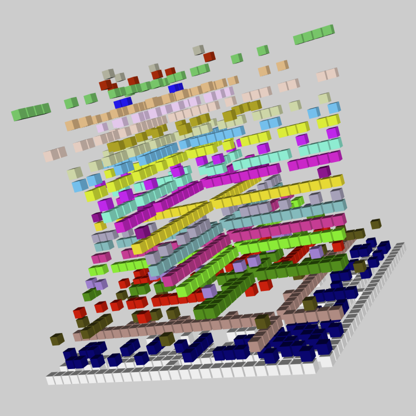
Vladimir Batagelj

IMFM Ljubljana

1333. sredin seminar

Ljubljana, January 25, 2023

Outline



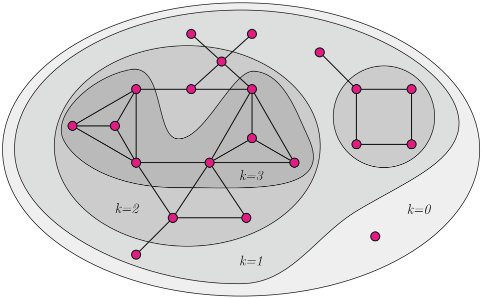
Vladimir Batagelj: vladimir.batagelj@fmf.uni-lj.si
Current version of slides (January 25, 2023 at 16 : 52): slides PDF
https://github.com/bavla/ibm3m/

Networks

A network $\mathcal{N} = (\mathcal{V}, \mathcal{L}, \mathcal{P}, \mathcal{W})$; $n = |\mathcal{V}|, m = |\mathcal{L}|$ is based on four sets: the set of nodes $\mathcal{V}$, the set of links $\mathcal{L}$, the set of node properties $\mathcal{P}$, and the set of weights $\mathcal{W}$. Each link is linking two nodes – its end-nodes. A link is either directed – an arc, or undirected – an edge. A pair of sets $(\mathcal{V}.\mathcal{L})$ forms a graph.

The notion of a k-core was introduced by Seidman in 1983 [11].

Cores



Let $\mathcal{G} = (\mathcal{V}, \mathcal{L})$ be a graph with set of nodes $\mathcal{V}$ and set of links $\mathcal{L}$. The degree of node $v \in \mathcal{V}$ is denoted by $\deg(v)$. For a given integer k, a subgraph $\mathcal{H}_k = (\mathcal{C}_k, \mathcal{L}|\mathcal{C}_k)$ induced by the subset of nodes $\mathcal{C}_k \subseteq \mathcal{V}$ is called a k-core or a core of order k iff $\deg_{\mathcal{H}_k}(v) \geq k$, for all $v \in \mathcal{C}_k$, and $\mathcal{H}_k$ is the maximum such subgraph. Usually, also $\mathcal{C}_k$ is called a k-core.

. . . Cores

The core of maximum order is called the *main core*. The *core number*, $\mathrm{core}(v)$, of node v is the highest order of a core that contains this node.

A k-core graph $\mathcal{H}_k$ is not always connected. They are nested

$$k_1 < k_2 \Rightarrow \mathcal{H}_{k_2} \subseteq \mathcal{H}_{k_1}$$

Let $N(v)$ denote the set of neighbors of a node v in a network $\mathcal{N}$, and $N(v, \mathcal{C}) = N(v) \cap \mathcal{C}$ is the set of neighbors of a node v within the subset of nodes $\mathcal{C} \subseteq \mathcal{V}$.

The procedure for determining the k-core is simple – remove from the network $\mathcal{N}$ all nodes with current degree less than k.

k-core algorithm

function $\textsc{Core}(\mathcal{N}, k)$
    $C \leftarrow \mathcal{V}$
    while $\exists u \in C : deg_C(u) < k$ **do** $C \leftarrow C \setminus \{u\}$
    return C
end function

Core decomposition

A very efficient algorithm exists for determining the *core decomposition* described with the *core partition* – a vector, $core[v]$, $v \in \mathcal{V}$, of core numbers of all nodes.

All nodes have a degree at least 0. They belong to the core $\mathcal{H}_0$ of order 0, $\mathcal{C}_0 = \mathcal{V}$. Removing all nodes of degree 0 we obtain the set $\mathcal{C}_1$ in which each node has degree at least 1 – the core $\mathcal{H}_1$ of order 1.

Removing all nodes of (current) degree 1 we obtain the set $\mathcal{C}_2$ in which each node has degree at least 2 – the core $\mathcal{H}_2$ of order 2. ...

This leads to an algorithm for determining each node's core number presented in the next slide. An efficient, $O(|\mathcal{L}|)$, implementation of this algorithm is given in [3].

Core decomposition algorithm

function $\textsc{CoreDecomposition}(\mathcal{N})$
 $deg \leftarrow [\mathrm{deg}(u) \text{ for } u \in \mathcal{V}]$
 $C \leftarrow \mathcal{V};\ k \leftarrow 1;\ core[1:n] \leftarrow 0$
 while $\neg empty(C)$ **do**
  while $\exists u \in C : deg[u] < k$ **do**
   for $v \in N(u, C)$ **do** $deg[v] \leftarrow deg[v] - 1$
   $C \leftarrow C \setminus \{u\}$
   $core[u] \leftarrow k - 1$
  end while
  $k \leftarrow k + 1$
 end while
 return *core*
end function

Generalized cores

Assume that in a network $\mathcal{N} = (\mathcal{V}, \mathcal{L}, \mathcal{P}, \mathcal{W})$ a *node property function* $p(v, \mathcal{C})$ is defined, $p : \mathcal{V} \times 2^{\mathcal{V}} \to \mathbb{R}$, where $\mathcal{C} \subseteq \mathcal{V}$ is a cluster – a subset of nodes, and $v \in \mathcal{C}$ is a node.
Some examples of such node properties were proposed in [3]:

1. $p_1(v, \mathcal{C}) = \deg_{\mathcal{C}}(v)$: node degree within the induced subgraph $\mathcal{N}(v, \mathcal{C}) = (\mathcal{C}, \mathcal{L}(\mathcal{C}))$;
2. $p_2(v, \mathcal{C}) = \text{indeg}_{\mathcal{C}}(v) + \text{outdeg}_{\mathcal{C}}(v)$: if all links are directed it holds $p_2 = p_1$;
3. $p_3(v, \mathcal{C}) = \mathrm{wdeg}(v, \mathcal{C}) = \sum_{u \in N(v,\mathcal{C})} w(v, u)$ for $w : \mathcal{L} \to \mathbb{R}_0^+$: *weighted degree*, the sum of weights of incident links within $\mathcal{N}(v, \mathcal{C})$;
4. $p_4(v, \mathcal{C}) = \max_{u \in N(v,\mathcal{C})} w(v, u)$ for $w : \mathcal{L} \to \mathbb{R}$: the maximum weight of incident links within $\mathcal{N}(v, \mathcal{C})$;
5. $p_5(v, \mathcal{C}) = \frac{\deg_{\mathcal{C}}(v)}{\deg(v)}$ if $\deg(v) > 0$ else $f_5(v, \mathcal{C}) = 0$: the fraction of neighbors within $\mathcal{N}(v, \mathcal{C})$;
6. $p_6(v, \mathcal{C}) = \frac{\mathrm{wdeg}(v,\mathcal{C})}{\mathrm{wdeg}(v,\mathcal{V})}$ for $w : \mathcal{L} \to \mathbb{R}_0^+$: the fraction of the weighted degree of incident links within $\mathcal{N}(v, \mathcal{C})$.

Generalized cores

The node property functions are used in the definition of generalized cores [3].

The subgraph $\mathcal{H} = (\mathcal{C}, \mathcal{L}(\mathcal{C}))$ induced by the set $\mathcal{C} \subseteq \mathcal{V}$ is a p-core at level $t \in \mathbb{R}$ iff $\forall v \in \mathcal{C} : t \leq p(v, \mathcal{C})$ and $\mathcal{C}$ is the maximum such set.

We say that the node property function $p(v, \mathcal{C})$:

- is *local* iff: $p(v, \mathcal{C}) = p(v, N(v, \mathcal{C}))\ \forall v \in \mathcal{V}$.
- is *monotonic* iff: $\mathcal{C}_1 \subset \mathcal{C}_2 \Rightarrow \forall v \in \mathcal{V} : p(v, \mathcal{C}_1) \leq p(v, \mathcal{C}_2)$..

For a local and monotone property function p the corresponding p-core numbers can be efficiently determined using a generalized version of cores algorithm [3].

p-core at level t algorithm

1: **function** $\textsc{Core}(\mathcal{N}, p, t)$
2: $\quad C \leftarrow \mathcal{V}$
3: $\quad$ **while** $\exists u \in C : p(u, C) < t$ **do** $C \leftarrow C \setminus \{u\}$
4: $\quad$ **return** C
5: **end function**

For a monotonic property function p, the result of this algorithm is independent on the nodes elimination order.

Generalized core at level t algorithm

```
1:  function GenCore(𝒩, f, t)
2:      C ← 𝒱
3:      for v ∈ 𝒱 do p[v] ← p(v, N(v, C))
4:      BUILD_MIN_HEAP(𝒱, p)
5:      while top.value < t do
6:          C ← C \ {top}
7:          for v ∈ N(top, C) do
8:              pv ← p(v, N(v, C))
9:              UPDATE_HEAP(v, pv)
10:         end for
11:     end while
12:     return C
13: end function
```

The step 8 can often be speeded up by updating the $p[v]$ – considering the change of $p[v]$ in each step.

Generalized core decomposition algorithm

function $\textsc{GenCoreDecomposition}(\mathcal{N}, f)$
  $C \leftarrow \mathcal{V}$; $core[1:n] \leftarrow 0$
  for $v \in \mathcal{V}$ **do** $p[v] \leftarrow p(v, N(v, C))$
  $\textsc{build\_min\_heap}(\mathcal{V}, p)$
  while $sizeof(heap) > 0$ **do**
    $top \leftarrow pop(heap)$; $C \leftarrow C \setminus \{top.node\}$
    $core[top.node] \leftarrow top.p$
    for $v \in N(top.node, C)$ **do**
      $p[v] \leftarrow \max\{top.p, p(v, N(v, C))\}$
      $\textsc{update\_heap}(v, p)$
    end for
  end while
  return $core$
end function

If $p(v, N(v, C))$ can be computed in $O(deg(v, C))$ time then for a monotone and local node property function p this algorithm determines the p-core hierarchy in $O(m \max(\Delta, \log n))$ time.

Two-mode cores

The notion of (p, q)-cores was introduced in [1] in 2007.

A subset $\mathcal{C} \subseteq \mathcal{V}$ determines a (p, q)-core in a two-mode network $\mathcal{N} = ((\mathcal{V}_1, \mathcal{V}_2), \mathcal{L})$, $\mathcal{V}_1 \cup \mathcal{V}_2 = \emptyset$ and $\mathcal{V}_1 \cup \mathcal{V}_2 = \mathcal{V}$ iff in the sub-network $\mathcal{K} = ((\mathcal{C}_1, \mathcal{C}_2), \mathcal{L}|\mathcal{C})$, $\mathcal{C}_1 = \mathcal{C} \cap \mathcal{V}_1$, $\mathcal{C}_2 = \mathcal{C} \cap \mathcal{V}_2$ induced by $\mathcal{C}$ it holds that for all $v \in \mathcal{C}_1 : \deg_K(v) \geq p$ and for all $v \in \mathcal{C}_2 : \deg_K(v) \geq q$, and $\mathcal{C}$ is the maximal such subset in $\mathcal{V}$.

Generalized two-mode cores

Let $\mathcal{N} = ((\mathcal{V}_1, \mathcal{V}_2), \mathcal{L}, (f, g), w)$, $\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2$ be a finite two-mode network — the sets $\mathcal{V}$ and $\mathcal{L}$ are finite. Let $P(\mathcal{V})$ be a power set of the set $\mathcal{V}$. Let functions f and g be node property functions defined on the network $\mathcal{N}$: $f, g : \mathcal{V} \times P(\mathcal{V}) \to \mathbb{R}_0^+$.

A subset of nodes $\mathcal{C} \subseteq \mathcal{V}$ in a two-mode network $\mathcal{N}$ is a generalized two-mode core $\mathcal{C} = Core(p, q; f, g)$, $p, q \in \mathbb{R}_0^+$ [4] if and only if in the subnetwork $\mathcal{K} = ((\mathcal{C}_1, \mathcal{C}_2), \mathcal{L}|\mathcal{C})$, $\mathcal{C}_1 = \mathcal{C} \cap \mathcal{V}_1$, $\mathcal{C}_2 = \mathcal{C} \cap \mathcal{V}_2$ induced by $\mathcal{C}$ it holds that for all $v \in \mathcal{C}_1 : f(v, \mathcal{C}) \geq p$ and for all $v \in \mathcal{C}_2 : g(v, \mathcal{C}) \geq q$, and $\mathcal{C}$ is the maximal such subset in $\mathcal{V}$.

Generalized two-mode core for node properties f and g at levels p and q

function $\textsc{remove?}(v, C)$
    if $v \in \mathcal{V}_1$ **then return** $f(v, C) < p$
    else return $g(v, C) < q$
end function
function $\textsc{Core}(\mathcal{N}, f, g, p, q)$
    $C \leftarrow \mathcal{V}$
    while $\exists u \in C : \textsc{remove?}(u, C)$ **do** $C \leftarrow C \setminus \{u\}$
    return C
end function

. . . more detailed

function $\text{REMOVE}(f_n, value, C_t, C_n, H_t, H_n)$
    while $H_t(top) < value$ **do**
        $top \leftarrow pop(H_t);\ C_t \leftarrow C_t \setminus \{top.node\}$
        for $v \in N(top.node, C_n)$ **do**
            $f_n[v] \leftarrow f_n(v, C_t)$
            $\text{UPDATE}(H_n, v)$
        end for
    end while
end function
function $\text{TWOMODECORE}(\mathcal{N})$
    $C_1 \leftarrow \mathcal{V}_1;\ C_2 \leftarrow \mathcal{V}_2$
    for $v \in V_1 : f_1[v] \leftarrow f_1(v, C_2)$; **for** $v \in V_2 : f_2[v] \leftarrow f_2(v, C_1)$
    $\text{BUILD}(H_1, f_1, C_1);\ \text{BUILD}(H_2, f_2, C_2)$
    repeat
        $\text{REMOVE}(f_2, p, C_1, C_2, H_1, H_2)$
        $\text{REMOVE}(f_1, q, C_2, C_1, H_2, H_1)$
    until no vertex was removed
    return C
end function

Multiway networks

A *weighted multiway network* $\mathcal{N} = (\mathcal{V}, \mathcal{L}, w)$ is based on *nodes* from k sets (ways or dimensions)

$$\mathcal{V} = \{\mathcal{V}_1, \mathcal{V}_2, \dots, \mathcal{V}_k\}$$

The set of *links* $\mathcal{L} \subseteq \mathcal{V}_1 \times \mathcal{V}_2 \times \cdots \times \mathcal{V}_k$. The weight $w : \mathcal{L} \to \mathbb{R}$. It can be represented by a k-dimensional array W

$$W[v_1, v_2, \dots, v_k] = w(v_1, v_2, \dots, v_k) \quad \text{for} \quad (v_1, v_2, \dots, v_k) \in \mathcal{L}$$

otherwise $W[v_1, v_2, \dots, v_k] = 0$.

In a general multiway network, different additional data can be known for nodes and/or links.

If for $i \neq j$, $\mathcal{V}_i = \mathcal{V}_j$, we say that $\mathcal{V}_i$ and $\mathcal{V}_j$ are of the same *mode*.

Cores and multiway networks

Selecting a pair of ways $\mathcal{V}_i$ and $\mathcal{V}_j$ in a multiway network $\mathcal{N}$ we essentially selected an ordinary network (with parallel links). If both ways are of the same mode it is a one-mode network; otherwise, it is a two-mode network.

For an undirected one-mode network the corresponding ways are of the same mode and "symmetric" (wiki).

The notion of a core is easily extended in this way to multiway networks. Additional ways can be considered in the selected node property functions.

Marmello77 links

```
> MA$links
   an pl R  w    an pl R  w    an pl R  w    an pl R  w
1   5  1 1  2 19  7 13 1  2 37  6 28 1  1 55  7 11 2 12
2   7  2 1 20 20  4 14 1  1 38  7 28 1 18 56  7 12 2  2
3   9  4 1  1 21  4 15 1 50 39  3 29 1  1 57  4 15 2  5
4   2  5 1 13 22  2 16 1  2 40  2 31 1 11 58  2 16 2  1
5   4  5 1  6 23  3 17 1  3 41  7 32 1  2 59  3 17 2  1
6   9  5 1  1 24  3 18 1  5 42  3 33 1  1 60  3 18 2  2
7   2  3 1 12 25  4 19 1  1 43  4 34 1  3 61  4 23 2 36
8   4  3 1 11 26  4 20 1  2 44  7  2 2  1 62  8 23 2  1
9   7  6 1  3 27  9 21 1  1 45  2  5 2  9 63  6 25 2  2
10  1  7 1  6 28  4 22 1  1 46  4  5 2 12 64  7 25 2 62
11  2  7 1 19 29  4 23 1 19 47  2  3 2  3 65  4 27 2  7
12  4  8 1  1 30  2 24 1 72 48  4  3 2  4 66  7 27 2  1
13  4  9 1  3 31  7 24 1 10 49  7  6 2  1 67  2 28 2  2
14  4 10 1  2 32  2 25 1  9 50  1  7 2  1 68  6 28 2  1
15  2 11 1  3 33  7 25 1 58 51  2  7 2  2 69  7 28 2 24
16  7 11 1 15 34  4 26 1  2 52  4  7 2  1 70  3 29 2  3
17  7 12 1  4 35  4 27 1  2 53  4  8 2  6 71  9 30 2  1
18  2 13 1  1 36  7 27 1  9 54  4  9 2  8 72  7 32 2  5
```

GitHub

R library MWnets

I included in the R library MWnets two functions implementing generalized cores algorithms

- `GenCoresDec(MN,way1,way2,way3=,weight=,p=)` – Generalized cores decomposition of the multiway network MN for the node property p on the ways way_1 and way_2.
- `Gen2modeCore(MN,way1,way2,f1,f2,t1,t2)` – Generalized two-mode cores of the multiway network MN for the node properties f_1 and f_2 on the ways way_1 and way_2 at levels t_1 and t_2.

I first programmed two core functions `relCore` and `relCore2` that are functionally special cases of the above functions. For example
`relCore(MN,way1,way2,way3)` $\equiv$
`GenCoresDec(MN,way1,way2,way3=way3,p=pRel)`

...included node property functions

A node property function `p(MN,u,C,way1,way2,...)` returns a node property value for the node u for the cluster C and selected ways way_1 and way_2. The function p can be selected among

- `pDeg(MN,u,C,way1,way2)` – Degree of the node u.
- `pSum(MN,u,C,way1,way2,weight=)` – Sum of weights of links in the node u.
- `pMax(MN,u,C,way1,way2,weight=)` – Maximum of the weights of links in the node u.
- `pRel(MN,u,C,way1,way2,way3=)` – The number of different relations of type way_3 in the node u.

Examples

1. The network from the cores figure [3] as a multiway network (wiki).
2. Biological multiway network Marmello77 (wiki).
3. European airlines – relational cores (wiki).
4. Italian students mobility – weight sum core (wiki).

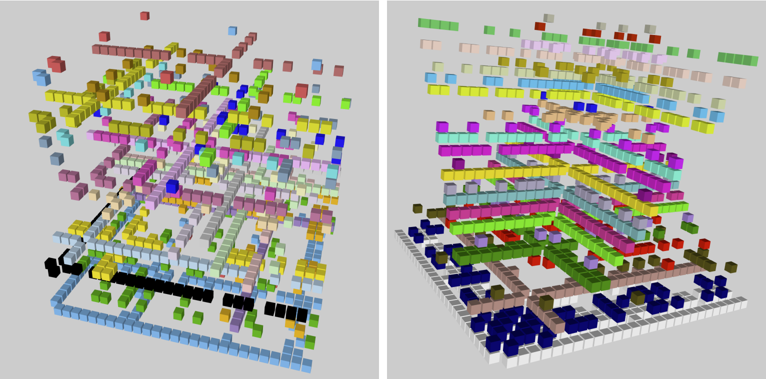
EUair core13 3D layout

To get some insight into the structure of the obtained core we created its 3D layout in X3D format – see the left side figure on the next slide (Wiki). A cube in the layout represents a link/triple (ap_1, ap_2, co) - the airport ap_1 is linked to the airport ap_2 with a line provided by the company co. Lines provided by the same company are of the same color.

In the X3D layout description we surrounded each link with the tag `<Anchor>` with the attribute `description`, for example
`<Anchor description="link 217 :  Milan-Malpensa A , Malaga Airport , Easyjet , 1 ">`,
to enable the user to get the info about the selected link during the layout inspection.

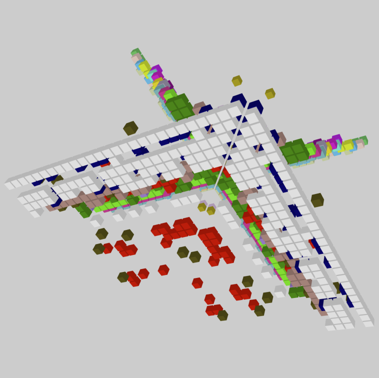
The layout contains many "crosses" – lines from a company's base airport. As the first attempt, we reordered companies by the number of lines and afterward the airports by the main airport for each company.

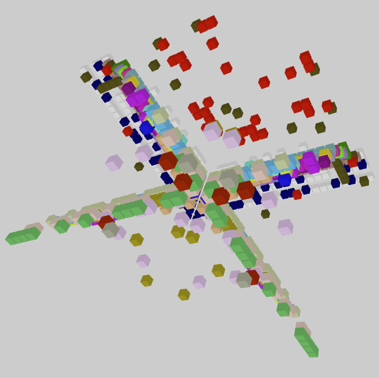
EUair core13 3D layout



The obtained layout confirmed our observation – most companies are providing lines from their base airport. It can be further improved by reordering (making closer) companies having the same base airport or airports served by the same company – see the right side figure (Wiki).

EUair core13 layout / Improvements





The bottom (left) and top (right) side views show that most of the links lie on crosses, but also that there are some companies with "dispersed" lines linking different airports.

EUair core13 layout / Detailed observations

- Lufthansa and Air Berlin are serving lines from German airports Frankfurt, Berlin, Munich, Dusseldorf, Hamburg, and Milan in Italy
- Swiss IAL and Netjets are serving lines from Swiss airports Zurich and Geneva
- SAS and Norwegian AS are serving lines from Copenhagen and Stockholm
- Spanish airports Madrid, Barcelona, and Malaga are served by companies Iberia, Air Nostrum, Vueling, and Ryanair
- companies serving lines from their base airport: (British A, Heathrow), (LOT, Warsaw), (Austrian A, Niki; Vienna), (KLM, Transavia; Amsterdam), (Air France, CDG), (Czech A, Prague), (Alitalia, Fiumicino), (Brussels A, European AT; Brussels), (Malev, Budapest), (Aegean, Olympic Air; Athens); the only “irregular” link is by Iberia between Barcelona and Budapest

. . . Detailed observations

- Companies with dispersed services are Easyjet, Netjets, Wizz Air, European AT, and TNT Airways
  - Easyjet is serving lines from Milan and also from CDG, Amsterdam, Fiumicino, Madrid
  - Netjets is serving lines also from Venice, Nice, Vienna, and Barcelona
  - Wizz Air is serving lines from Budapest and Fiumicino
  - European AT has also some lines from Heathrow, Milan, Venice, Barcelona, and Madrid
  - TNT Airways is linking airports CDG and Athens, and Sofia and Henri Coanda

EUair core13 / Airports


```
 [1] "Frankfurt Airport"
 [2] "Berlin Tegel Airport"
 [3] "Munich Airport"
 [4] "Dusseldorf Airport"
 [5] "Hamburg Airport"
 [6] "Zurich Airport"
 [7] "Geneva Airport"
 [8] "Milan-Malpensa A"
 [9] "Copenhagen Airport"
[10] "Stockholm Arlanda Airport"
[11] "Heathrow Airport"
[12] "Warsaw Chopin Airport"
[13] "Vienna International Airport"
[14] "Amsterdam Airport Schiphol"
[15] "Charles de Gaulle Airport (Roissy Airport)"
[16] "Adolfo Suarez Madrid-Barajas Airport"
[17] "Barcelona El Prat Airport"
[18] "Malaga Airport"
[19] "Vaclav Havel Airport Prague"
[20] "Leonardo da Vinci-Fiumicino Airport"
[21] "Brussels Airport (Zaventem Airport)"
[22] "Budapest Ferenc Liszt International Airport"
[23] "Athens International Airport (Eleftherios Venizelos Airport)"
[24] "Ben Gurion Airport"
[25] "Henri Coanda International Airport"
[26] "Venice Marco Polo Airport"
[27] "Sofia Airport"
[28] "Nice Cote d'Azur Airport"
```

EUair core13 / Companies


```
 [1] "Lufthansa"    "Air Berlin"   "Swiss IAL"    "Netjets"      "Easyjet"
 [6] "SAS"          "Norwegian AS" "British A"    "LOT Polish A" "Austrian A"
[11] "Niki"         "KLM"          "Transavia H"  "Air France"   "Iberia"
[16] "Air Nostrum"  "Vueling A"    "Ryanair"      "Czech A"      "Alitalia"
[21] "Brussels A"   "European AT"  "Malev HA"     "Wizz Air"     "Aegean A"
[26] "Olympic Air"  "TNT Airways"
```

Conclusions

1. Work in progress
   1. more testing,
   2. improving some code (efficiency, programming solutions),
   3. unification,
   4. robustness.
2. Apply the methods to different data sets (test, experience, ideas)
3. Normalizations
4. Additional data sets. New problems.
5. Julia and/or Python version
6. Additional Javascript support for X3DOM visualization (slicing, ...)

Acknowledgments

The computational work reported in this presentation was performed using R library MWnets. The code and data are available at GitHub Bavla/ibm3m.

We would like to thank Andreas Plesch for extending the library X3DOM with support for `Anchor`'s attribute `description`.

This work is supported in part by the Slovenian Research Agency (research program P1-0294 and research projects J5-2557, J1-2481 and J5-4596).

References I

- Ahmed, A., Batagelj, V., Fu, X., Hong, S.-H., Merrick, D., Mrvar, A., 2007. Visualisation and analysis of the Internet movie database. In: 6th International Asia-Pacific Symposium on Visualization, IEEE, pp. 17–24.
- Batagelj, V, Cerinšek, M.: On bibliographic networks. Scientometrics 96 (2013), pages 845–864. Springer
- Batagelj, V., Zaveršnik, M.: Fast algorithms for determining (generalized) core groups in social networks. Advances in Data Analysis and Classification, 2011. Volume 5, Number 2, 129-145.
- Cerinšek, M., Batagelj, V.: Generalized two-mode cores. Social Networks 42 (2015), 80–87.
- Batagelj, V.: On fractional approach to the analysis of linked networks. Scientometrics 123 (2020), pages621–633. Springer
- Batagelj, V, Ferligoj, A, Doreian, P: Indirect Blockmodeling of 3-Way Networks. In: Brito, P, Cucumel, G, Bertrand, P, de Carvalho, F (eds) Selected Contributions in Data Analysis and Classification pp 151–159 Springer, 2007; preprint.

References II

Borgatti, SP, Everett, MG: Regular blockmodels of multiway, multimode matrices. Social Networks 14(1992)1-2, 91–120.

Everett, MG, Borgatti, SP: Chapter 9, Partitioning Multimode Networks. In: Doreian, P, Batagelj, V, Ferligoj, A (Eds) Advances in Network Clustering and Blockmodeling , Wiley, 2019.

Kroonenberg, PM: Applied multi-way data analysis. Wiley, 2008.

Schoonees, PC, Groenen, PJF, van de Velden, M: Least-squares bilinear clustering of three-way data. Advances in Data Analysis and Classification (2022) 16:1001–1037. paper.

S. B. Seidman.
Network structure and minimum degree.
Social Networks, 5:269–287, 1983.