# Overlapping community detection in labeled graphs

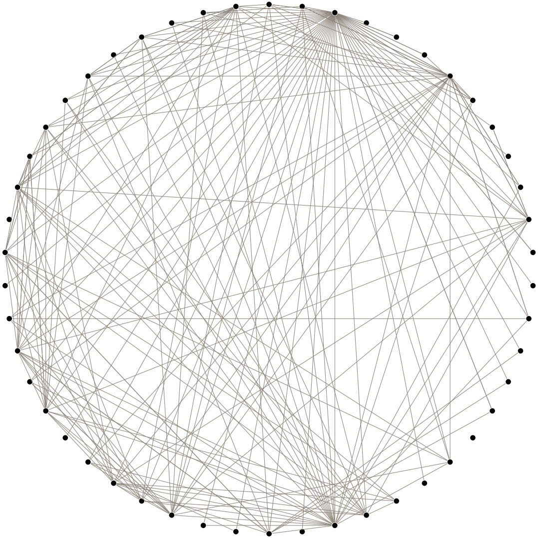
Esther Galbrun, Aristides Gionis and Nikolaj Tatti

BU
Boston University College of Arts & Sciences

Aalto University
School of Science

September 2014

# Example



# Example


{job-title:label, job-title:record…}
{gender:1, job-title:hous, institution:freelance…}
{place:milano, job-title:label, place:italy…}
{job-title:music, job-title:hous…}
{job-title:produc, job-title:artist, place:london, job-title:label…}
{job-title:label, gender:1, place:london…}
{job-title:produc, job-title:hous…}
{gender:1, university:london, place:london…}
{gender:1, job-title:analyst, job-title:hous…}
{gender:1, place:london…}
{place:london, job-title:comput…}
{job-title:develop, gender:1…}
{job-title:internat, job-title:music, job-title:media, job-title:man…}

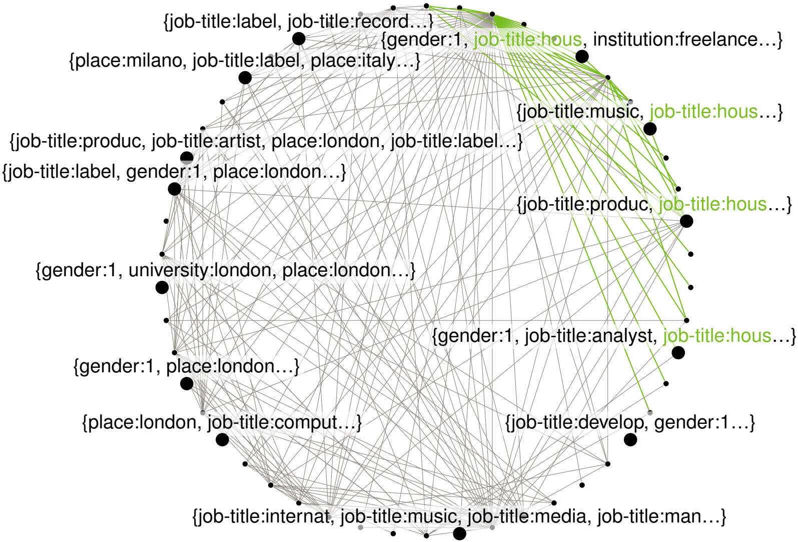
# Example


{job-title:label, job-title:record…}
{gender:1, job-title:hous, institution:freelance…}
{place:milano, job-title:label, place:italy…}
{job-title:music, job-title:hous…}
{job-title:produc, job-title:artist, place:london, job-title:label…}
{job-title:label, gender:1, place:london…}
{job-title:produc, job-title:hous…}
{gender:1, university:london, place:london…}
{gender:1, job-title:analyst, job-title:hous…}
{gender:1, place:london…}
{place:london, job-title:comput…}
{job-title:develop, gender:1…}
{job-title:internat, job-title:music, job-title:media, job-title:man…}

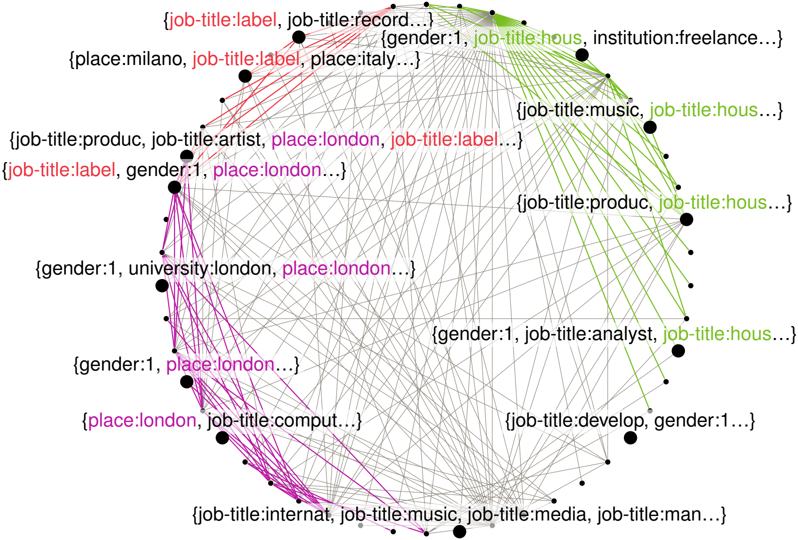
# Example


{job-title:label, job-title:record…}
{gender:1, job-title:hous, institution:freelance…}
{place:milano, job-title:label, place:italy…}
{job-title:music, job-title:hous…}
{job-title:produc, job-title:artist, place:london, job-title:label…}
{job-title:label, gender:1, place:london…}
{job-title:produc, job-title:hous…}
{gender:1, university:london, place:london…}
{gender:1, job-title:analyst, job-title:hous…}
{gender:1, place:london…}
{place:london, job-title:comput…}
{job-title:develop, gender:1…}
{job-title:internat, job-title:music, job-title:media, job-title:man…}

# Example


{job-title:label, job-title:record…}
{gender:1, job-title:hous, institution:freelance…}
{place:milano, job-title:label, place:italy…}
{job-title:music, job-title:hous…}
{job-title:produc, job-title:artist, place:london, job-title:label…}
{job-title:label, gender:1, place:london…}
{job-title:produc, job-title:hous…}
{gender:1, university:london, place:london…}
{gender:1, job-title:analyst, job-title:hous…}
{gender:1, place:london…}
{place:london, job-title:comput…}
{job-title:develop, gender:1…}
{job-title:internat, job-title:music, job-title:media, job-title:man…}

# Example


{job-title:label, job-title:record…}
{gender:1, job-title:hous, institution:freelance…}
{place:milano, job-title:label, place:italy…}
{job-title:music, job-title:hous…}
{job-title:produc, job-title:artist, place:london, job-title:label…}
{job-title:label, gender:1, place:london…}
{job-title:produc, job-title:hous…}
{gender:1, university:london, place:london…}
{gender:1, job-title:analyst, job-title:hous…}
{gender:1, place:london…}
{place:london, job-title:comput…}
{job-title:develop, gender:1…}
{job-title:internat, job-title:music, job-title:media, job-title:man…}

# Goal

Considering a multi-labeled graph $G = (V, E, \ell)$,
we look for communities that are not only **dense subgraphs**,
but that also admit **compact descriptions** in terms of labels.

# Dense subgraphs

- We consider subgraphs $H = (U, F)$ with $F \subseteq E(U)$, where $E(U)$ is the set of edges induced by vertices $U$.
- Our reward function is the density of $H$

$$d(H) = \frac{2|F|}{|U|}.$$

- Each edge is assigned to *at most one* subgraph.

# Compact descriptions

- *Conjunctive* predicate over labels and vertices

$$p(S) = \{v \in V \mid S \subseteq \ell(v)\}.$$

# Interlude: non-labeled graphs

# GMC & DS

Our algorithm couples two approximation algorithms:

**GMC** the Generalized Maximum-Coverage problem, by Cohen and Katzir (2008).

*A variant of the max k -cover problem where elements have different rewards for each bin.*

**DS** the Densest-Subgraph problem, by Charikar (2000).

*Finding a subgraph maximizing the density.*

# Dense: **Greedy on vertices**

For $i = 1 \dots k$

- Remove iteratively the vertex with smallest degree.
- Pick among obtained subgraphs the one having the highest degree.
- Take out edges assigned to the selected subgraph.

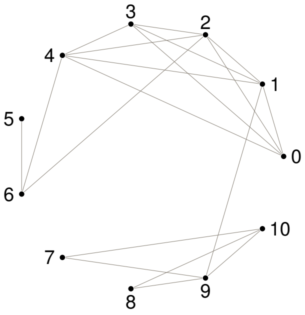
# Dense: **Greedy on vertices**

For $i = 1 \ldots k$

- Remove iteratively the vertex with smallest degree, *reintroduce edges if the global score improves.*
- Pick among obtained subgraphs the one having the highest degree.
- *Tentatively* take out edges assigned to the selected subgraph.

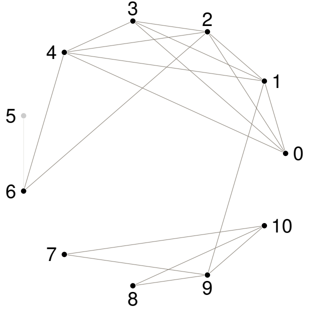
# Dense: **Greedy on vertices**

| Vertex | $d(H)$ |
|---|---|
| | 3.45 |



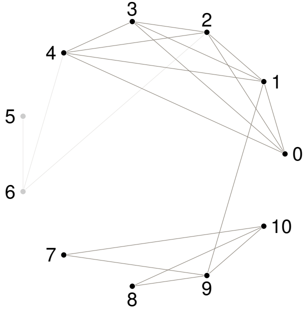
# Dense: **Greedy on vertices**

| Vertex | $d(H)$ |
|---|---|
| | 3.45 |
| 5 | 3.60 |



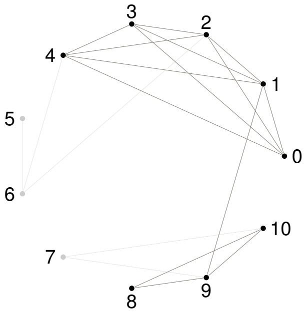
# Dense: Greedy on vertices

| Vertex | $d(H)$ |
|---|---|
| | 3.45 |
| 5 | 3.60 |
| 6 | 3.56 |



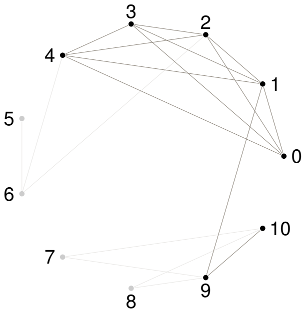
# Dense: **Greedy on vertices**

| Vertex | $d(H)$ |
|---|---|
| | 3.45 |
| 5 | 3.60 |
| 6 | 3.56 |
| 7 | 3.50 |



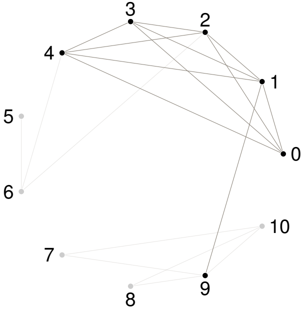
# Dense: **Greedy on vertices**

| Vertex | $d(H)$ |
|---|---|
| | 3.45 |
| 5 | 3.60 |
| 6 | 3.56 |
| 7 | 3.50 |
| 8 | 3.43 |



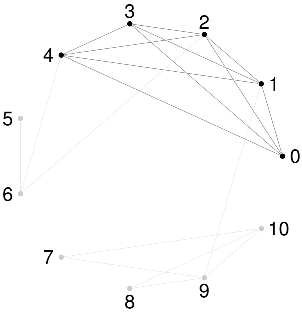
# Dense: **Greedy on vertices**

| Vertex | $d(H)$ |
|---|---|
| | 3.45 |
| 5 | 3.60 |
| 6 | 3.56 |
| 7 | 3.50 |
| 8 | 3.43 |
| 10 | 3.67 |



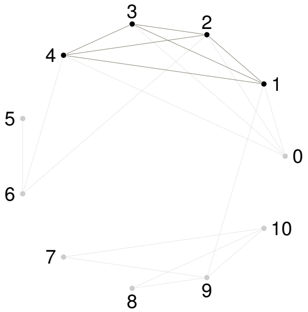
# Dense: **Greedy on vertices**

| Vertex | $d(H)$ |
|---|---|
| | 3.45 |
| 5 | 3.60 |
| 6 | 3.56 |
| 7 | 3.50 |
| 8 | 3.43 |
| 10 | 3.67 |
| 9 | 4.00 |



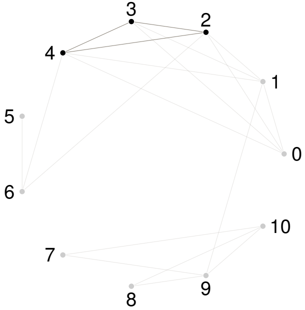
# Dense: **Greedy on vertices**

| Vertex | $d(H)$ |
|---|---|
| | 3.45 |
| 5 | 3.60 |
| 6 | 3.56 |
| 7 | 3.50 |
| 8 | 3.43 |
| 10 | 3.67 |
| 9 | 4.00 |
| 0 | 3.00 |



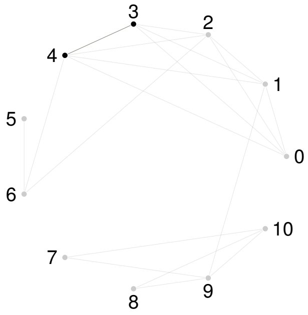
# Dense: **Greedy on vertices**

| Vertex | $d(H)$ |
|---|---|
| | 3.45 |
| 5 | 3.60 |
| 6 | 3.56 |
| 7 | 3.50 |
| 8 | 3.43 |
| 10 | 3.67 |
| 9 | 4.00 |
| 0 | 3.00 |
| 1 | 2.00 |



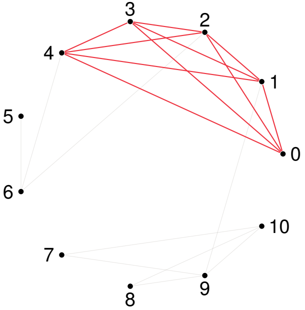
# Dense: Greedy on vertices

| Vertex | $d(H)$ |
|---|---|
| | 3.45 |
| 5 | 3.60 |
| 6 | 3.56 |
| 7 | 3.50 |
| 8 | 3.43 |
| 10 | 3.67 |
| 9 | 4.00 |
| 0 | 3.00 |
| 1 | 2.00 |
| 2 | 1.00 |



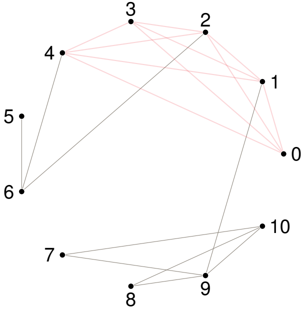
# Dense: **Greedy on vertices**

| Vertex | $d(H)$ |
|---|---|
| | 3.45 |
| 5 | 3.60 |
| 6 | 3.56 |
| 7 | 3.50 |
| 8 | 3.43 |
| 10 | 3.67 |
| **9** | **4.00** |
| 0 | 3.00 |
| 1 | 2.00 |
| 2 | 1.00 |



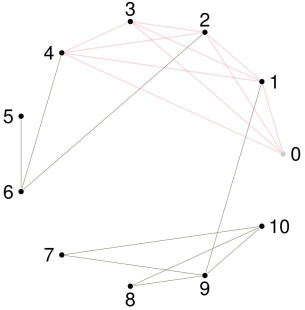
# Dense: Greedy on vertices

| Vertex | $d(H)$ |
|---|---|
| | 1.64 |



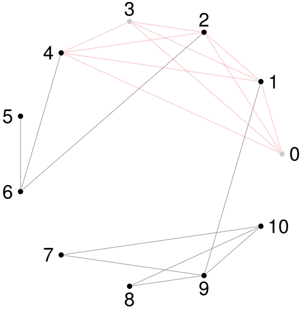
# Dense: **Greedy on vertices**

| Vertex | $d(H)$ |
|---|---|
| | 1.64 |
| 0 | 1.80 |



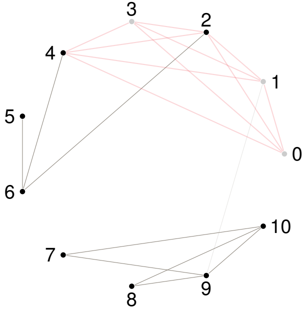
# Dense: **Greedy on vertices**

| Vertex | $d(H)$ |
|---|---|
| | 1.64 |
| 0 | 1.80 |
| 3 | 2.00 |



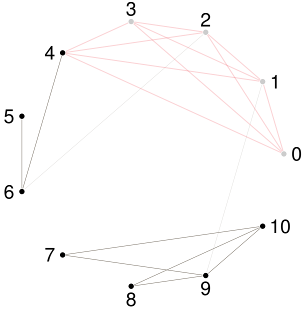
# Dense: **Greedy on vertices**

| Vertex | $d(H)$ |
|---|---|
| | 1.64 |
| 0 | 1.80 |
| 3 | 2.00 |
| 1 | 2.00 |



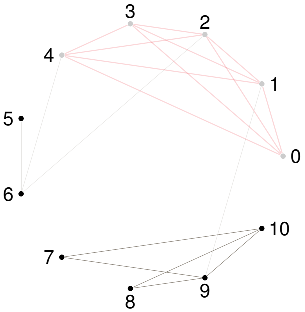
# Dense: **Greedy on vertices**

| Vertex | $d(H)$ |
|---|---|
| | 1.64 |
| 0 | 1.80 |
| 3 | 2.00 |
| 1 | 2.00 |
| 2 | 2.00 |



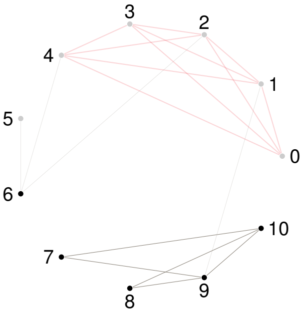
# Dense: **Greedy on vertices**

| Vertex | $d(H)$ |
|---|---|
| | 1.64 |
| 0 | 1.80 |
| 3 | 2.00 |
| 1 | 2.00 |
| 2 | 2.00 |
| 4 | 2.00 |



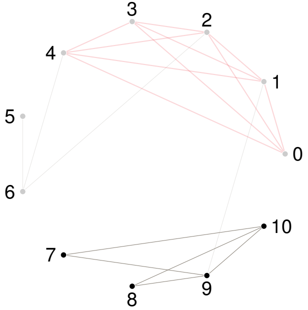
# Dense: **Greedy on vertices**

| Vertex | $d(H)$ |
|---|---|
| | 1.64 |
| 0 | 1.80 |
| 3 | 2.00 |
| 1 | 2.00 |
| 2 | 2.00 |
| 4 | 2.00 |
| 5 | 2.00 |



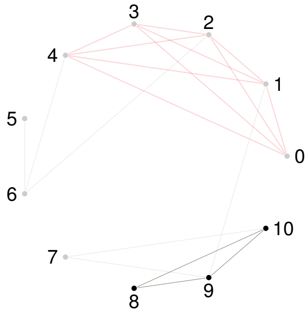
# Dense: **Greedy on vertices**

| Vertex | $d(H)$ |
|---|---|
| | 1.64 |
| 0 | 1.80 |
| 3 | 2.00 |
| 1 | 2.00 |
| 2 | 2.00 |
| 4 | 2.00 |
| 5 | 2.00 |
| 6 | 2.50 |



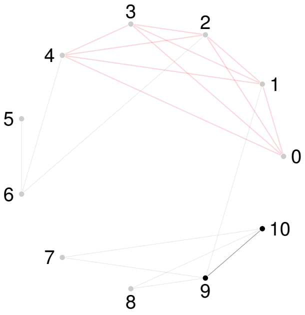
# Dense: **Greedy on vertices**

| Vertex | $d(H)$ |
|---|---|
| | 1.64 |
| 0 | 1.80 |
| 3 | 2.00 |
| 1 | 2.00 |
| 2 | 2.00 |
| 4 | 2.00 |
| 5 | 2.00 |
| 6 | 2.50 |
| 7 | 2.00 |



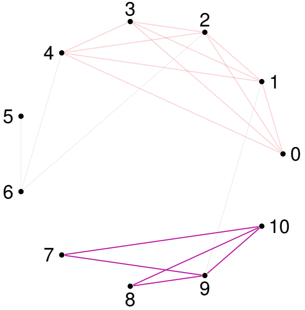
# Dense: **Greedy on vertices**

| Vertex | $d(H)$ |
|---|---|
| | 1.64 |
| 0 | 1.80 |
| 3 | 2.00 |
| 1 | 2.00 |
| 2 | 2.00 |
| 4 | 2.00 |
| 5 | 2.00 |
| 6 | 2.50 |
| 7 | 2.00 |
| 8 | 1.00 |



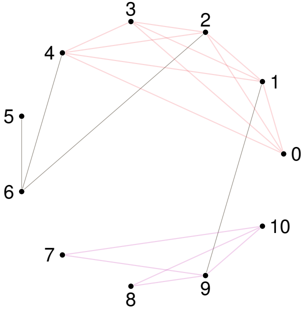
# Dense: **Greedy on vertices**

| Vertex | $d(H)$ |
|---|---|
| | 1.64 |
| 0 | 1.80 |
| 3 | 2.00 |
| 1 | 2.00 |
| 2 | 2.00 |
| 4 | 2.00 |
| 5 | 2.00 |
| **6** | **2.50** |
| 7 | 2.00 |
| 8 | 1.00 |



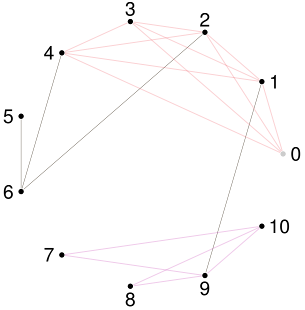
# Dense: **Greedy on vertices**

| Vertex | $d(H)$ |
|---|---|
| | 0.73 |



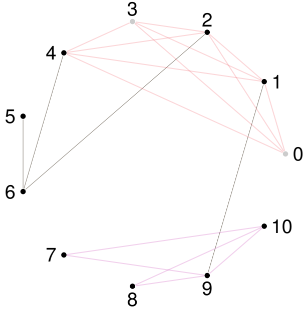
# Dense: **Greedy on vertices**

| Vertex | $d(H)$ |
|---|---|
| | 0.73 |
| 0 | 0.80 |



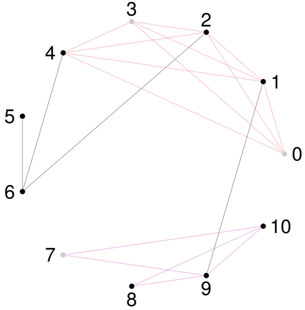
# Dense: **Greedy on vertices**

| Vertex | $d(H)$ |
|---|---|
| | 0.73 |
| 0 | 0.80 |
| 3 | 0.89 |



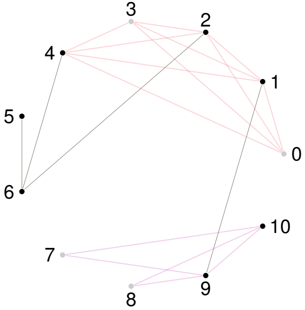
# Dense: **Greedy on vertices**

| Vertex | $d(H)$ |
|---|---|
| | 0.73 |
| 0 | 0.80 |
| 3 | 0.89 |
| 7 | 1.00 |



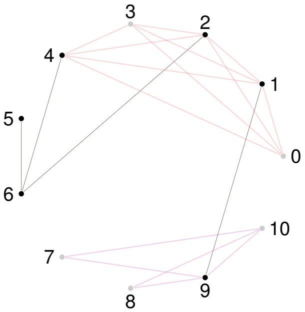
# Dense: **Greedy on vertices**

| Vertex | $d(H)$ |
|---|---|
| | 0.73 |
| 0 | 0.80 |
| 3 | 0.89 |
| 7 | 1.00 |
| 8 | 1.14 |



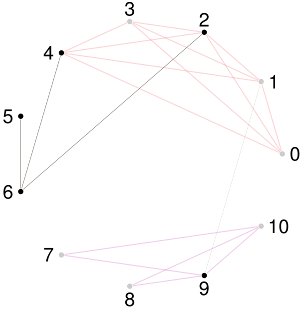
# Dense: **Greedy on vertices**

| Vertex | $d(H)$ |
|---|---|
| | 0.73 |
| 0 | 0.80 |
| 3 | 0.89 |
| 7 | 1.00 |
| 8 | 1.14 |
| 10 | 1.33 |



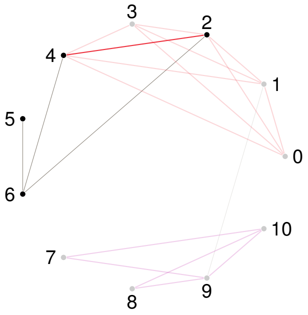
# Dense: **Greedy on vertices**

| Vertex | $d(H)$ |
|---|---|
| | 0.73 |
| 0 | 0.80 |
| 3 | 0.89 |
| 7 | 1.00 |
| 8 | 1.14 |
| 10 | 1.33 |
| 1 | 1.20 |



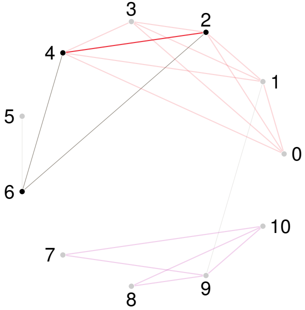
# Dense: **Greedy on vertices**

| Vertex | $d(H)$ |
|---|---|
| | 0.73 |
| 0 | 0.80 |
| 3 | 0.89 |
| 7 | 1.00 |
| 8 | 1.14 |
| 10 | 1.33 |
| 1 | 1.20 |
| 9 | 2.00 |



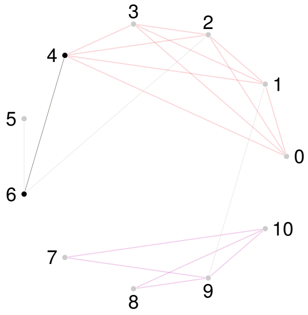
# Dense: **Greedy on vertices**

| Vertex | $d(H)$ |
|---|---|
| | 0.73 |
| 0 | 0.80 |
| 3 | 0.89 |
| 7 | 1.00 |
| 8 | 1.14 |
| 10 | 1.33 |
| 1 | 1.20 |
| 9 | 2.00 |
| 5 | 2.00 |



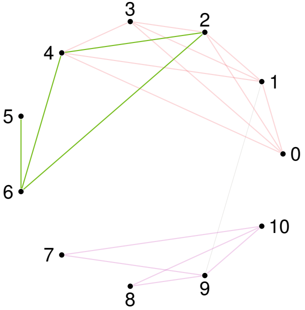
# Dense: **Greedy on vertices**

| Vertex | $d(H)$ |
|---|---|
| | 0.73 |
| 0 | 0.80 |
| 3 | 0.89 |
| 7 | 1.00 |
| 8 | 1.14 |
| 10 | 1.33 |
| 1 | 1.20 |
| 9 | 2.00 |
| 5 | 2.00 |
| 2 | 1.00 |



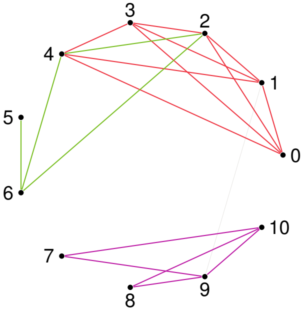
# Dense: **Greedy on vertices**

| Vertex | $d(H)$ |
|---|---|
| | 0.73 |
| 0 | 0.80 |
| 3 | 0.89 |
| 7 | 1.00 |
| 8 | 1.14 |
| 10 | 1.33 |
| 1 | 1.20 |
| **9** | **2.00** |
| 5 | 2.00 |
| 2 | 1.00 |



# Dense: **Greedy on vertices**

$d(H_1) = \ 3.60$
$d(H_2) = \ 2.50$
$d(H_3) = \ 2.00$



# Back to labeled graphs

For $i = 1 \ldots k$

- Select a dense subgraph with a compact label description

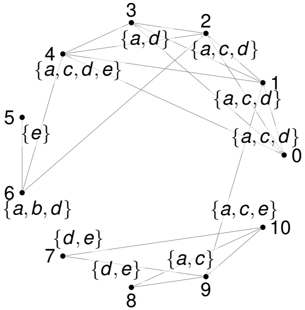
*3 algorithms for finding subgraphs*

# LDense: Greedy on labels

- Instead of peeling-off individual vertices, use labels to guide the process.

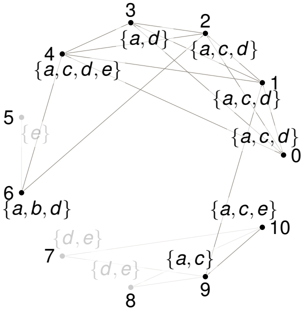
# LDense: **Greedy on labels**

| Label | $d(H)$ |
| --- | --- |
| | 3.45 |



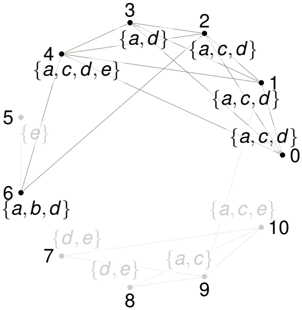
# LDense: **Greedy on labels**

| Label | $d(H)$ |
|---|---|
| | 3.45 |
| a | 3.50 |



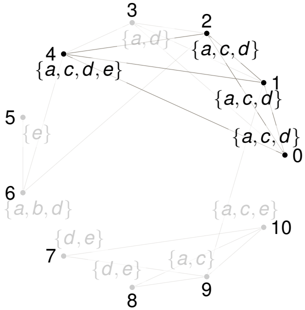
# LDense: **Greedy on labels**

| Label | $d(H)$ |
|---|---|
| | 3.45 |
| a | 3.50 |
| d | 4.00 |



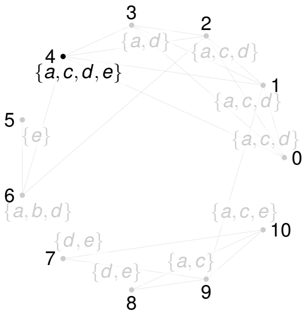
# LDense: Greedy on labels

| Label | $d(H)$ |
|---|---|
| | 3.45 |
| a | 3.50 |
| d | 4.00 |
| c | 3.00 |



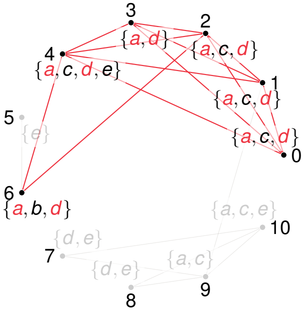
# LDense: **Greedy on labels**

| Label | $d(H)$ |
|---|---|
| | 3.45 |
| a | 3.50 |
| d | 4.00 |
| c | 3.00 |
| e | 0.00 |



# LDense: **Greedy on labels**

| Label | $d(H)$ |
|---|---|
| | 3.45 |
| a | 3.50 |
| **d** | **4.00** |
| c | 3.00 |
| e | 0.00 |



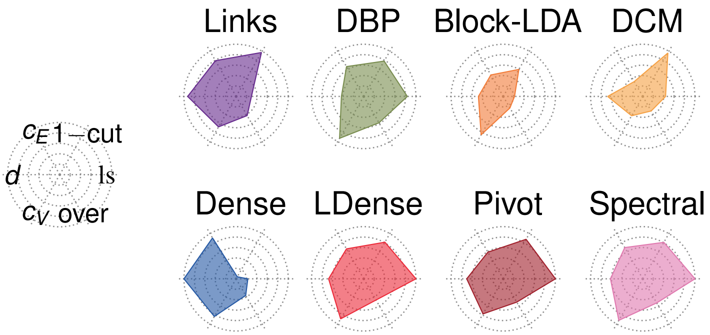
# Spectral

- Order the labels according to the Fielder vector of the similarity matrix.
- Consider only contiguous sets of labels in this ordering.

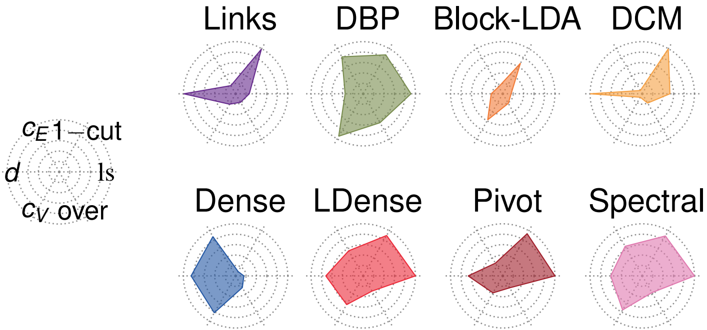
# Pivot

- Consider the local neighborhood of each vertex, “pivoted subgraphs”, as initial candidates.
- Refine by enforcing labels predicate.

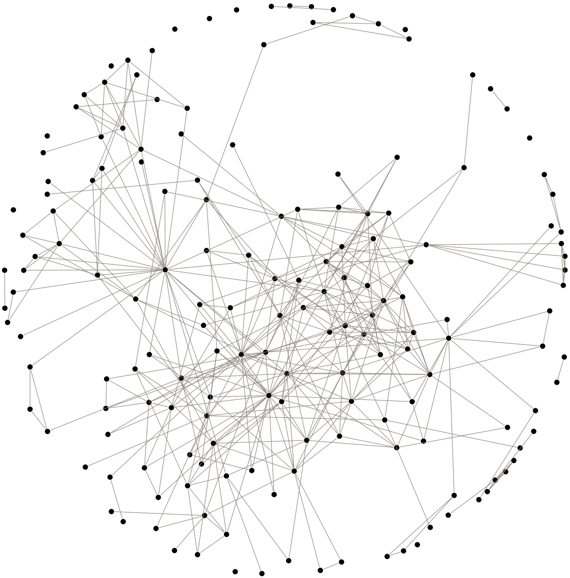
# Comparative evaluation: FB, k=20



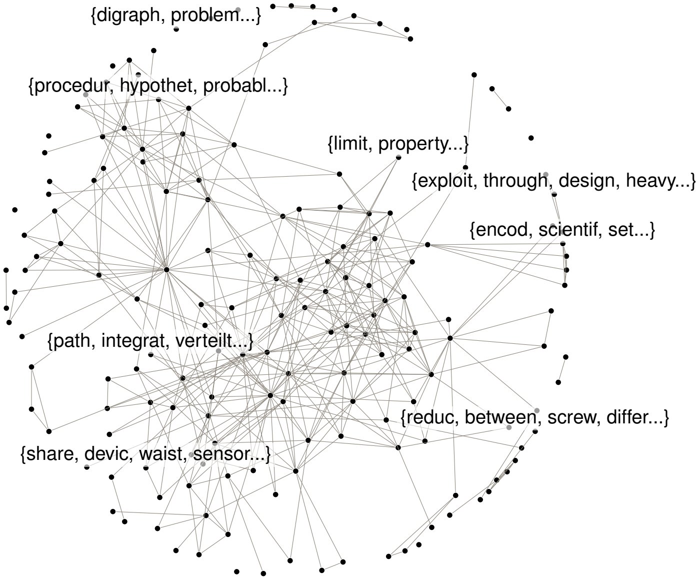
# Comparative evaluation: DBLP.E2, k=20



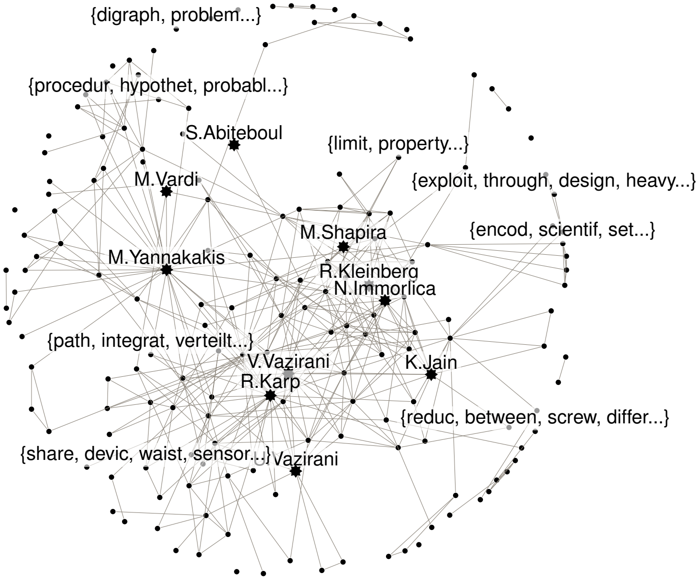
# Co-authors of C. H. Papadimitriou



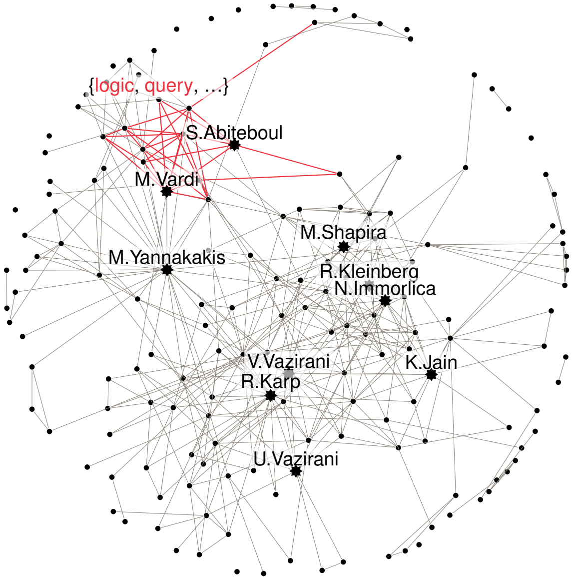
# Co-authors of C. H. Papadimitriou



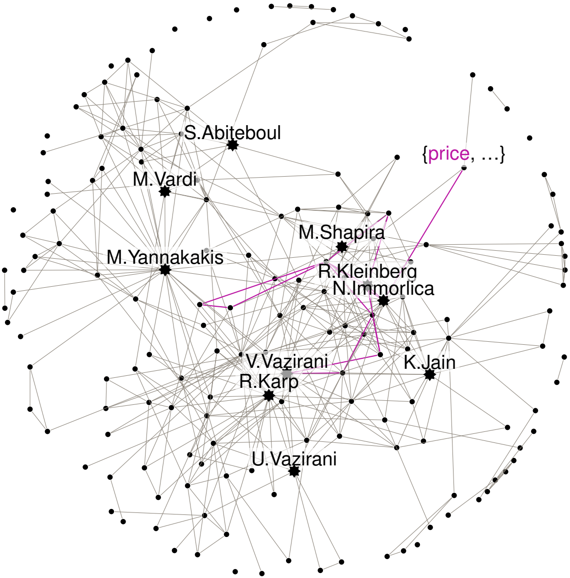
# Co-authors of C. H. Papadimitriou



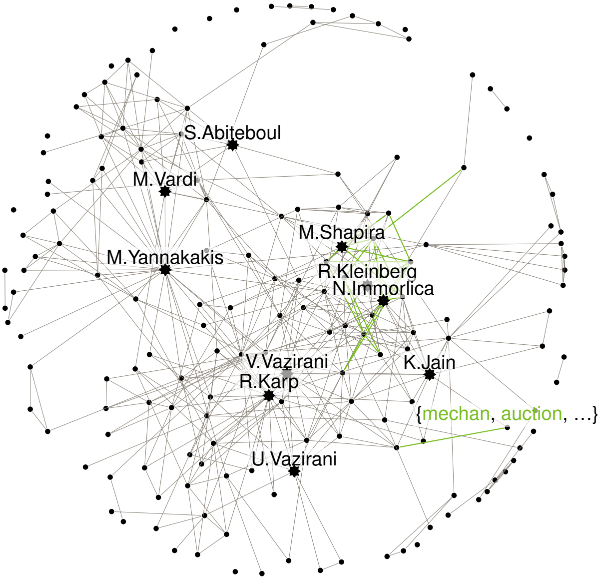
# Co-authors of C. H. Papadimitriou



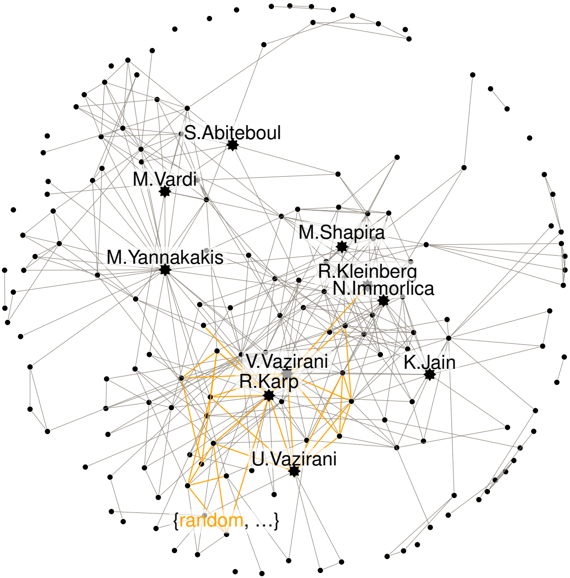
# Co-authors of C. H. Papadimitriou



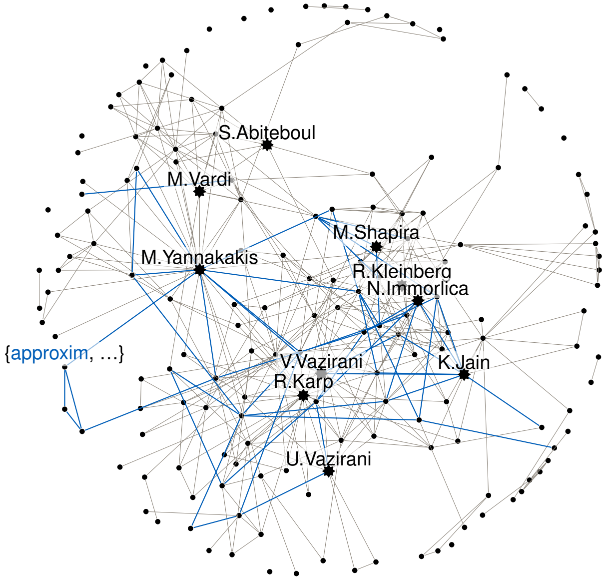
# Co-authors of C. H. Papadimitriou



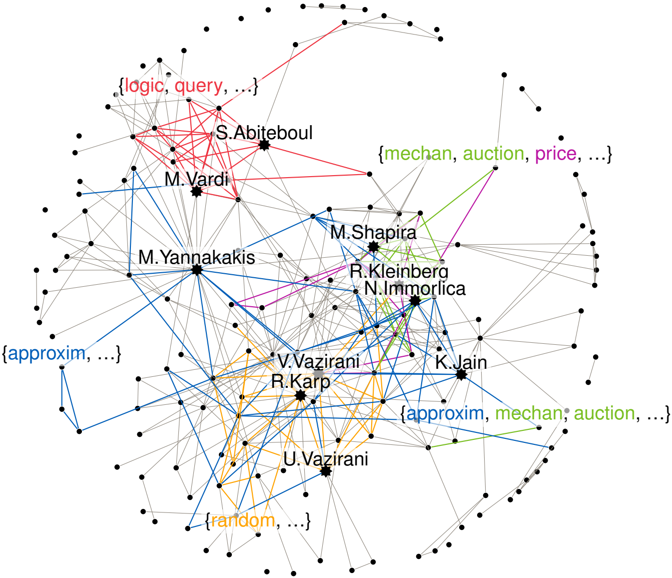
# Co-authors of C. H. Papadimitriou



# Co-authors of C. H. Papadimitriou



# Co-authors of C. H. Papadimitriou



# Co-authors of C. H. Papadimitriou

| Labels | Authors |
|---|---|
| logic, query | M. Vardi, G. Kuper, S. Abiteboul, Y. Sagiv, F. Afrati, ... |
| price | M. Babaioff, R. Kleinberg, N. Immorlica, ... |
| mechan, auction | S. Dobzinski, K. Talwar, M. Schapira, R. Sami, ... |
| random | R. Karp, U. Vazirani, V. Vazirani, A. Saberi, ... |
| approxim | M. Yannakakis, K. Jain, X. Deng, C. Daskalakis, ... |

# Conclusions

- Formalized a problem of overlapping community detection in labeled graphs.
- Proposed a greedy scheme with three variants inspired from approximation algorithms for the GMC and DS problems.
- Compared the proposed algorithms to existing algorithms.

# Conclusions

- Formalized a problem of overlapping community detection in labeled graphs.
- Proposed a greedy scheme with three variants inspired from approximation algorithms for the GMC and DS problems.
- Compared the proposed algorithms to existing algorithms.

**Thanks!**