



**HAL**  
open science

## Properties of B2B invoice graphs and detection of structures

Joannès Guichon, Nazim A. Fatès, Sylvain Contassot-Vivier, Massimo Amato

► **To cite this version:**

Joannès Guichon, Nazim A. Fatès, Sylvain Contassot-Vivier, Massimo Amato. Properties of B2B invoice graphs and detection of structures. Complex Networks 2023, Nov 2023, Menton, France. hal-04230839v2

**HAL Id: hal-04230839**

**<https://inria.hal.science/hal-04230839v2>**

Submitted on 12 Oct 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Properties of B2B invoice graphs and detection of structures

Joannès Guichon, Nazim Fatès, Sylvain Contassot-Vivier<sup>1</sup>, and  
Massimo Amato<sup>2</sup>

<sup>1</sup> Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France  
joannes.guichon@inria.fr,

WWW home page: <https://members.loria.fr/JGuichon/>

<sup>2</sup> Bocconi University, Milan, Italy

**Abstract.** In economy, a major issue is the potential lack of liquidity for settling the debts generated by payment delays among companies. Since this lack may trigger cascading failures, we analyse the interconnection of debts. Settling debts means lowering the systemic risks. We analyse the data of a large economic network from an Italian invoice operator on a one-year span. We compare different methods to detect structures or communities that could be helpful for debt netting algorithms. The structure of such networks is not currently well known. We give hints on how to sort and identify the type of B2B invoice graphs. In particular, we address the possibility to identify relevant communities in such networks.

**Keywords:** Graph analysis, community detection, B2B invoice networks.

## 1 Introduction

The intricate nature of debt arises from financial obligations, among companies triggered by the payment system, fostering a web of interconnected relationships. As companies generally pay their *invoices* with a delay, debts accumulate and intertwine. A potential chain reaction of defaults emerges, posing a significant threat to the stability of the entire financial system. Debt settlement emerges as a vital mechanism to uphold commitments, maintain credibility, and bolster trust in the corporate world.

To address the challenges posed by trade debts, the concept of *netting* may come into play [9]. Such treatment consists in reducing the global amount of debts between the companies by using mutual compensations, thus lowering the need for immediate liquidity. This liquidity saving mechanism could streamline the debt settlement process, minimize credit risks, and enhance the efficiency and resilience of the corporate ecosystem to ensure a smoother and more secure financial landscape. Different types of debt netting techniques exist, mainly focused on partial [12] or complete [2] settlement. Partial netting is based on the possibility to split invoices in order to partially settle the debt. On the opposite way, complete netting is restricted to cancel entire invoices. Today, research on

this type of netting is not well developed and this technique needs a finer knowledge about the graphs structure. Globally, all those kinds of processes mainly exist between banks [14,15] and our aim is to extend them to B2B exchanges.

However, in order to implement such a method in algorithmic form, it would be useful to identify specific properties of invoice graphs between companies. These networks are quite specific as they correspond to weighted, directed, multi-edge and time-varying graphs. In Section 2, we determine the main characteristics of exchange networks. Then, in Section 3, we try to identify particular structures or communities. The aim is to split up our graphs in order to facilitate their processing by netting algorithms. This work should be considered a preparatory work before more research on netting algorithms. As a first step, we focus on three methods to understand the structure of these graphs.

## 2 B2B invoice graphs : Definition and properties

This study is based on a set of  $27,445,353$  invoices emitted by companies of North Italy over the span of the 2019 year. This dataset was provided by Infocert, an Italian electronic invoices operator. Each invoice is composed of the following data:

- unique identifier to register the invoice;
- unique identifier of the debtor company;
- unique identifier of the creditor company;
- due debt in euros;
- date of the invoice emission.

It is worth noticing that this dataset is anonymized and obviously it is partial as it includes only the invoices given to the operator by subscribers of its service. However, although we have non-exhaustive information, we think that the size of the dataset is in itself significant and could provide some hints on the structure of the national activity.

We filtered the original dataset (removal of incomplete data) and divided it into subsets of different sizes, according to the span of time considered (day, week, month). In particular, studying different time granularities is useful to determine which time span is best suited to the netting procedures [2].

We use **weighted directed multi-edges graphs**, often referred to as a **weighted directed multigraphs**. This type of graphs is represented by a tuple  $G = (V, E, w)$ , where  $V$  is the set of nodes,  $E$  is the set of directed edges and  $w$  is the weight function over the edges. The edge with index  $i$  is defined as  $e_i = (v_j, v_k)$ , where  $v_j$  and  $v_k$  are distinct vertices in  $V$ . The weight function returns a positive value for a given edge index.

The particularity of multigraphs is that the same couple of distinct nodes may be in different edges. However, they are distinguished by the edge index.

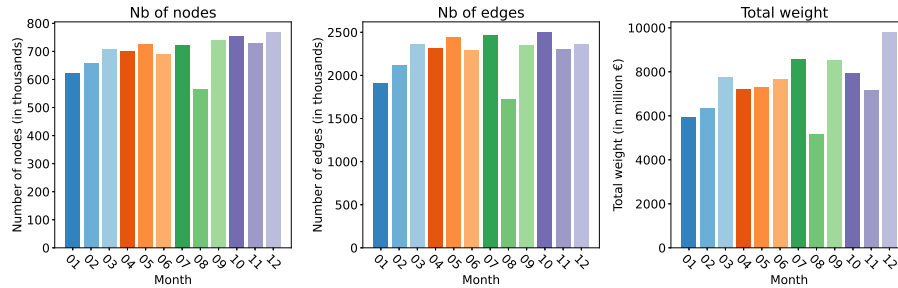
In the economic context, we call our multigraphs *invoice graphs*. They are oriented as follows: the source of an arc is the receiver (debtor) while the destination is the emitter (creditor).

### 2.1 Comparative study of our datasets

As mentioned above, we analyse our data on a monthly, weekly and daily basis in order to identify specific properties according to time granularity.

**Aggregation per month.** The monthly decomposition offers several advantages. It is commonly used for financial reporting purposes, providing a concise overview of companies’ revenue and expenses over time. Furthermore, it facilitates the evaluation of overall performance, offering a holistic view of revenues and expenses by comparing the trends from month to month. This method enables long-term decision-making, including activities like setting budgets or identifying long-term growth trends when comparing year-on-year performance. Particularly, if one possesses data spanning more than a year, these trends become crucial for prioritization within a netting process.

When we compare each month, Figure 1 shows that the start of year and the month of August are lighter in terms of transactions. It seems quite logical as the month of January is a month of slowdown for companies that allocate their budget, plan their decision-making strategy and have a decline in sales after the end of year rush. On the other hand, August registers a lower number of invoices because it is the preferred month for companies’ Summer holiday in Italy. We observe similar behaviours for nodes, edges and weights variations.

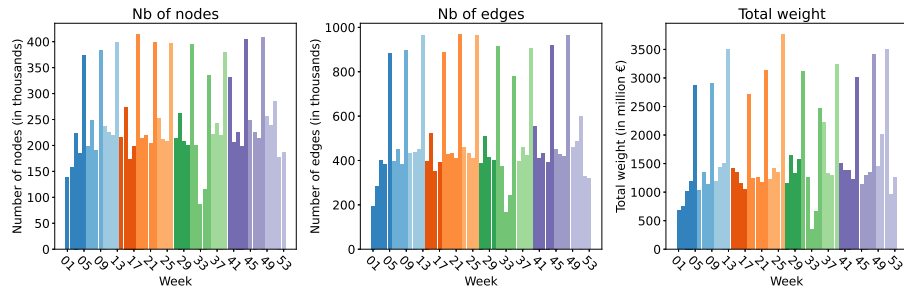


**Fig. 1.** Variations of nodes, edges and weights on monthly graphs. Colour tones depend on the trimester.

**Aggregation per week.** The advantages of making weekly cuts include the ability to detect short-term trends. This approach is beneficial for reporting and mid-term planning as it offers more detailed insights than broader monthly cuts. Additionally, this type of cut smoothens the daily fluctuations and highlights cyclical activity patterns. Indeed, weekly cuts allow for a closer view on the working capital turnover.

Looking at the invoice distribution on a weekly basis, we can see in Figure 2, that companies operate mostly on a monthly billing cycle, sending their invoices

grouped on the last week of each month. This highlights the need in terms of efficiency for companies to send invoices in bulk. Also, it allows for payment alignment with their clients' schedule. From our perspective, it could be interesting to tackle invoices by performing the netting algorithm twice a month. According to the edge distribution over a month, half of the transactions happen on the first three weeks and the second half occurs during the last week. So, it appears reasonable to apply the netting on the first three weeks separately from the last one. As for monthly cuts, variations on each graph are similar.



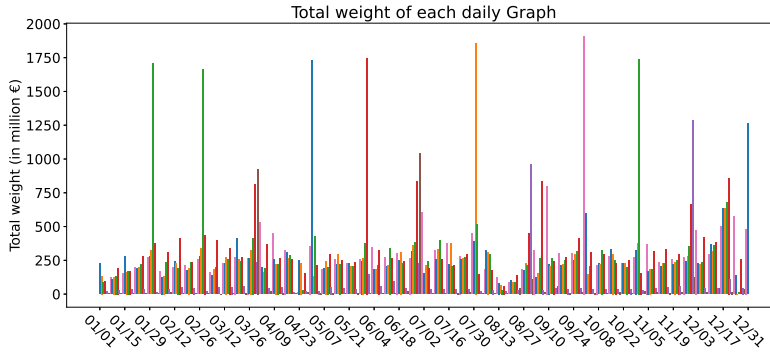
**Fig. 2.** Variations of nodes, edges and weights on weekly graphs. Colour depends on the trimester and the month of the first day of the week.

**Aggregation per day.** From an economic point of view, cutting our data by day provides a high level of granularity for a detailed view of transactions and cash flow patterns. It is useful to identify short-term trends or irregularities and for real-time monitoring, in order to promptly respond to issues of delayed payments for example. Moreover, it gives an insight for operational decision-making, improving the management of inventories and resources allocation of the different companies.

From the graph analysis point of view, this aggregation gives us a fine-grain decomposition into small sub-graphs that may be easier to process by a netting algorithm. Figure 3 gives the total weights of incomes for each daily sub-graphs. General periodic patterns can be observed for weeks and months with only one exception in June with an extremely high concentration of volume on the last day of each month. As before, the total number of nodes and edges are strongly correlated to the weights, so we do not include them here.

## 2.2 Statistical study on monthly graphs

We now focus on monthly cuts as they provide graphs that are less subject to variations and they are more densely connected than weekly and daily cuts.



**Fig. 3.** Total weights of daily graphs. Colours depend on the day of the week.

Table 1 gathers a set of statistics exhibiting particular behaviours. Concerning the diameters, they are obtained after removing all nodes of degree one (about a third of the nodes). Average path lengths are obtained by computing shortest paths between ten thousand random pairs of nodes.

Looking at our data from a correlation point of view, we see that the diameter and the clustering coefficient do not seem to be especially correlated with the size of our network and are quite stable from one month to another. The low values of clustering coefficient indicate that we are far from having a Small-World network in which the clustering coefficient is much higher [10].

Also, we observe that the diameter and the average path length are quite small compared to the number of nodes and strongly correlated.

For the month of January, only thirty nodes have a degree higher than a thousand. These nodes can be considered as mega hubs in the network since they interact with more than 0.1% of the rest of the nodes. Coupled with the low diameter and average path length, it means that no node is far from a hub, linking it to the rest of the graph.

### 2.3 Comparison with well-known families of graphs

We now compare B2B invoice networks with Small-World and Scale-Free networks as they are the most commonly used to analyse real world networks. Such a comparison is relevant as in the case of similarities with a given family, we could take advantage of the identified structure to design efficient netting algorithms.

Small-world graphs were famously described by social psychologists Milgram and Travers in their “six degrees of separation” experiment, in which they found that individuals are, on average, separated by only a few acquaintances. This type of networks was formalized by Watts and Strogatz [19] and exhibits a high clustering coefficient and a short average path length.

Scale-free networks, characterized by hubs with many connections, resemble many real-world systems like social networks and the internet, as highlighted by

**Table 1.** Statistical values describing our twelve monthly and the annual graphs. GWCC and GSCC respectively stand for Giant Weakly and Giant Strongly Connected Component.

Month	01	02	03	04	05	06	07	08	09	10	11	12	Annual
Nodes ( $10^3$ )	623	657	707	699	723	691	721	566	740	754	729	767	2057
Edges ( $10^3$ )	1910	2120	2362	2311	2443	2296	2462	1723	2348	2496	2301	2364	27138
Mean Degree	3.07	3.23	3.34	3.31	3.38	3.22	3.42	3.05	3.17	3.31	3.16	3.08	13.19
Clustering Coefficient	0.010	0.007	0.010	0.007	0.007	0.013	0.007	0.008	0.011	0.011	0.007	0.016	0.028
Diameter	29	29	28	28	31	25	27	32	28	33	31	27	34
Average Path Length	9.57	9.14	9.26	9.52	9.34	9.51	9.24	10.25	9.64	9.31	9.20	9.42	9.37
Nodes GWCC (%)	99.19	99.20	99.34	99.32	99.36	99.29	99.40	99.07	99.40	99.42	99.40	99.45	99.95
Edges GWCC (%)	99.80	99.20	99.84	99.84	99.85	99.74	99.85	99.75	99.84	99.85	99.84	99.85	99.97
Nodes GSCC (%)	7.70	8.58	9.09	8.88	9.35	9.15	9.50	7.01	8.69	9.26	8.77	8.99	11.83
Edges GSCC (%)	34.87	34.75	36.86	36.42	38.08	36.53	37.87	32.07	33.64	37.14	36.18	36.1	47.34

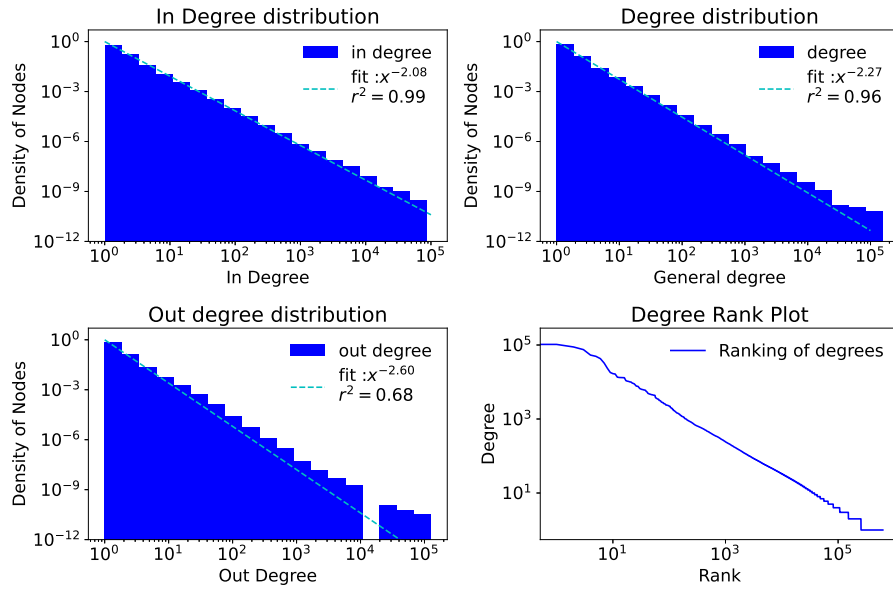
Faloutsos and al. [6]. Their structure affects how information spreads. In these graphs, edges' growth is generally faster than nodes' growth due to the property of preferential attachment. The presence of hubs enhances communication efficiency but also makes networks vulnerable to targeted attacks. Barabási et al. showed the signature power-law degree distribution of these networks [3]. Their main feature is the existence of power-law distributions with the conservation of properties at different scales.

## 2.4 Distribution of degrees and weights

The link between nodes and edges suggests a power-law relationship (Table 1). This suggests that our graphs follow the usual scale-free property of economic networks which is the typical behaviour of scale-invariant graphs. The degree distribution resembles a power-law but presents a large tail. Indeed, the plot of degree distribution on log-log graphs shows good agreement with straight lines. These lines are the mark of a power-law distribution but other distributions such as the negative binomial distribution can be considered, as raised by Fricke and Lux in their article on interbank networks [8].

In order to find the power-law coefficients that fit our graphs, the method of the Maximum Likelihood Estimator (MLE) has been used on our entire dataset. The different gamma values obtained with MLE are provided in Table 2. As strong differences have been observed between In and Out degrees in our graphs, it is better suited to distinguish their power-law analysis. Figure 4 provides the different histograms as well as the overall degree rank plot. These plots are made using density in order to make the comparison easier visually and to have normalized data.

To judge the fitting quality of those power-law, the generalized  $r^2$  proposed by Cox and Snell [5] was used for each month on the whole data interval. For the In degree fitting, the minimal value found was  $r^2 = 0.95$  in December which means that a power-law fits quite well with the In degree distribution. The results for Out degree laws are lower with the best being  $r^2 = 0.72$ , meaning that this distribution does not precisely correspond to a power-law. Concerning the rank plot, its global linear aspect tends to empower the idea of a construction following power-laws. It is an open problem to understand this law asymmetry between In and Out degrees.



**Fig. 4.** Distribution of degrees for the month of January

**Table 2.** Values of our different gammas found by MLE. SD is the Standard Deviation between months.

Month	01	02	03	04	05	06	07	08	09	10	11	12	Mean	SD
$\gamma_{\text{General}}$	2.27	2.25	2.21	2.22	2.23	2.22	2.22	2.19	2.23	2.21	2.23	2.23	2.23	0.02
$\gamma_{\text{Out}}$	2.60	2.62	2.61	2.59	2.54	2.55	2.52	2.51	2.58	2.53	2.55	2.61	2.57	0.04
$\gamma_{\text{In}}$	2.08	2.06	2.08	2.05	2.05	2.04	2.05	2.01	2.04	2.04	2.05	2.05	2.05	0.02



We also compare the gamma obtained for different intervals on the month of January since the tail of our data can be not representative because of variations. We also cut the nodes of degree one in our intervals since they may be strongly biased by our dataset that is limited to Infocert clients’ invoices. The results in Table 3 show the variability of the gamma depending on the data considered. The interval that fits best all type of degree is the  $[2-50]$  range with a  $r^2$  of 0.91 for In Degree distribution, 0.87 for general degree and 0.86 for Out degree. When taking all the data into account, a reason we don’t observe a high value of  $r^2$  for the Out degree distribution probably comes from the fact that statistically speaking, the Out degree is smaller than the In degree, which generates a narrower distribution of Out degree and less reliable statistical results.

**Table 3.** Values of our different gammas found by MLE for the month of January depending on the range for data.

Range	Global	[2-50]	[2-100]	[2-500]	[2-1000]	[2-10000]	[2-∞[
$\gamma_{\text{General}}$	2.27	2.60	2.51	2.43	2.42	2.41	2.41
$\gamma_{\text{Out}}$	2.60	2.80	2.71	2.67	2.66	2.66	2.66
$\gamma_{\text{In}}$	2.08	2.40	2.31	2.22	2.20	2.19	2.18

As stated before, our aim in finding a classic graph family close to monthly cut graphs is to use their specific properties to improve netting algorithms. As seen above, our networks present potentially two distinct laws on the In and Out degree distributions. These laws allow us to consider the possibility of efficient graphs partitioning to apply netting algorithms.

### 3 Communities detection in B2B invoice graphs

In this section we tackle the problem of detecting communities. As mentioned before, this would be useful to apply a decomposition of the netting problem. Indeed, since the netting problem we are interested in is NP-complete [2], this divide-and-conquer approach allows us to apply our algorithms on large graphs in reasonable time.

The main obstacle in community detection is the multiplicity of definitions of what a community is. In particular, in economy there are many ways to define communities, it could be actors of the same sector, of the same geographical perimeter, etc. In our case it is more relevant to define communities as groups of highly connected actors.

#### 3.1 Two direct methods for communities detection

As a first step, we use two different approaches that are not the most popular ones for communities detection nowadays. However, they are still used due to their simplicity and acceptable results quality.

**Matrix reordering methods.** In some cases, adjacency matrices can allow one to visualize the structure of graphs. However, the difficulty is to find a reordering of the nodes that spacially separates the communities inside the matrix. Different methods exist and we applied some of those exposed by Behrish et al. in their survey [4]. We were surprised to notice that none of these approaches could reveal the presence of communities. After analysing the problem, we found out that it is not really surprising as even with strongly structured graphs such as Caveman graphs, this method does not produce good results as soon as some noise is added to the structuration of the graph (random links between communities).

**$k$ -core method.** A simple idea to detect communities is described by Fortunato [7] (see Section 4). The idea is to get rid of nodes that have less than a certain degree, to update the degree of the other nodes and to repeat until there is no change in the graph anymore. This method named  $k$ -cores ( $k$  is the degree threshold) was first used by Seidman when working on social network graphs [17]. After trying this method, we found out that it often generates either one giant component or a myriad of small communities with high degree nodes. As shown in Table 4, each decomposition of the January graph presents a main weakly connected component (WCC) that concentrates most of the nodes and edges.

The mean degree is not linearly correlated to the core number. This comes from the fact that invoice graphs present some nodes of extremely high degree heavily influencing the mean degree when the size of the graph decreases. Concerning the diameter, we observe an expected decrease according to the increase of  $k$ . However, it is interesting to see that the decrease is not strictly monotonous. Finally, we can see that the strongly connected components are smaller than the weakly connected ones and their evolution is not monotonous. But, the higher the  $k$ , the higher the concentration inside the GWCC, highlighting our hyper-connected parts of the main graph. Indeed, suppressing small degree nodes tends to remove small connected components that are not highly connected making the SCC ratios increase. However, the removal of higher degree nodes tend to divide the SCCs so that their ratio decreases (which is not what we apply here).

These decompositions help us to find groups of extremely highly connected components we could use as a starting point for netting algorithms.

### 3.2 Communities detection using modularity

The most usual method in community detection on big networks is the graph partitioning according to the measure of modularity. This parameter measures the degree of segregation or clustering of nodes within a network compared to what would be expected by chance (random network of the same size). It quantifies the extent to which a network can be divided into distinct, densely connected groups or communities. Readers that want more details about modularity can refer to the survey of Newman [16]. The classic way to create communities using modularity to is the Louvain method, described by Lambiotte et al. in their article on communities for large networks [13].

**Table 4.** Statistical values describing different  $k$ -cores for the month of January

$k$ -core	Complete	3-core	5-core	10-core	20-core	50-core	100-core	200-core
Nodes	622 941	147 521	74 386	28 066	9 434	2 180	876	355
Edges ( $10^3$ )	1 910	1 334	1 092	803	567	367	282	217
Mean Degree	3.07	9.04	14.30	29.01	60.00	169.20	323.42	612.00
Diameter	24	21	20	17	14	10	9	9
Nodes GWCC (%)	99.19	99.73	99.75	99.78	99.84	99.82	100.00	99.44
Edges GWCC (%)	99.80	99.93	99.93	99.95	99.95	99.96	100.00	99.89
Nodes GSCC (%)	7.70	27.32	37.25	47.09	57.01	73.30	78.88	82.25
Edges GSCC (%)	34.87	48.55	54.98	62.95	72.24	81.72	85.56	87.04

This method takes a resolution parameter that guides the size of extracted communities. It allows one either to create numerous small groups of weakly connected nodes while providing good inter-connections between groups, or to produce a set of larger groups with less inter-connections. Having little groups is useful for computing algorithms but it may miss a part of the information contained in the inter-connection. Moreover, although the number of communities does not change much with the resolution, there is an increase of the number of large communities (see Table 5). This means that the largest communities merge and that the smaller ones stay on their own.

**Table 5.** Repartition of the nodes according to modularity

Resolution	0.5	1.0	1.5
Modularity	0.747	0.749	0.741
Number of communities	2170	2082	2058
Number of community with more than 5% of the nodes	4	6	9
Percentage of nodes in the biggest community	6.95	7.43	14.74
Percentage of edges in the biggest community	3.32	4.92	10.70
Percentage of inter-community edges	21.5	19.65	17.93

This method allows for the control of the sizes of the largest communities. It is efficient to find well-connected nodes but it does not affect smaller groups. These small groups are either in different connected components or not connected enough to bigger ones.

Another limitation of this modularity approach as well as previous methods is that they are based on grouping nodes and not on grouping edges. Contrary to classic approaches, the netting problem requires to group invoices by similar amounts instead of grouping them with a node-based approach. Hence, an open problem is the research of new algorithms that would group edges by similarities rather than nodes.

## 4 Conclusion

We have proposed a two-fold study on B2B invoice graphs from Infocert. On the one hand, we focused on identifying distinctive characteristics of such graphs. On the other hand, we compared different methods to extract communities from these graphs.

Concerning the first aspect, we observe that B2B graphs do not correspond to small-world graphs, displaying lower clustering coefficients. They closely align with scale-free graphs, demonstrated by power-law degree distributions. These findings emphasize the presence of influential nodes in the network and global robustness to random failures. This also tends to align with the study of other real networks in economy that present a tendency to preferential attachment and so power-laws. Although not a perfect fit, this study advances our understanding of invoice graphs and their properties, crucial for our application on B2B networks.

Concerning the communities aspect, the different approaches for communities successfully identify densely connected node groups. Indeed, we observe the presence of a giant strongly connected component in our network. However, the obtained results show that further refinement is necessary to be fully suited for edge-centric applications such as netting.

This research not only advances our understanding of invoice graphs but also serves as a stepping stone for future work on B2B netting procedures. The imbalance between In and Out degree may have to be taken into account when working on debt netting algorithms.

## Acknowledgement

The authors are grateful to Infocert for their trust and support.

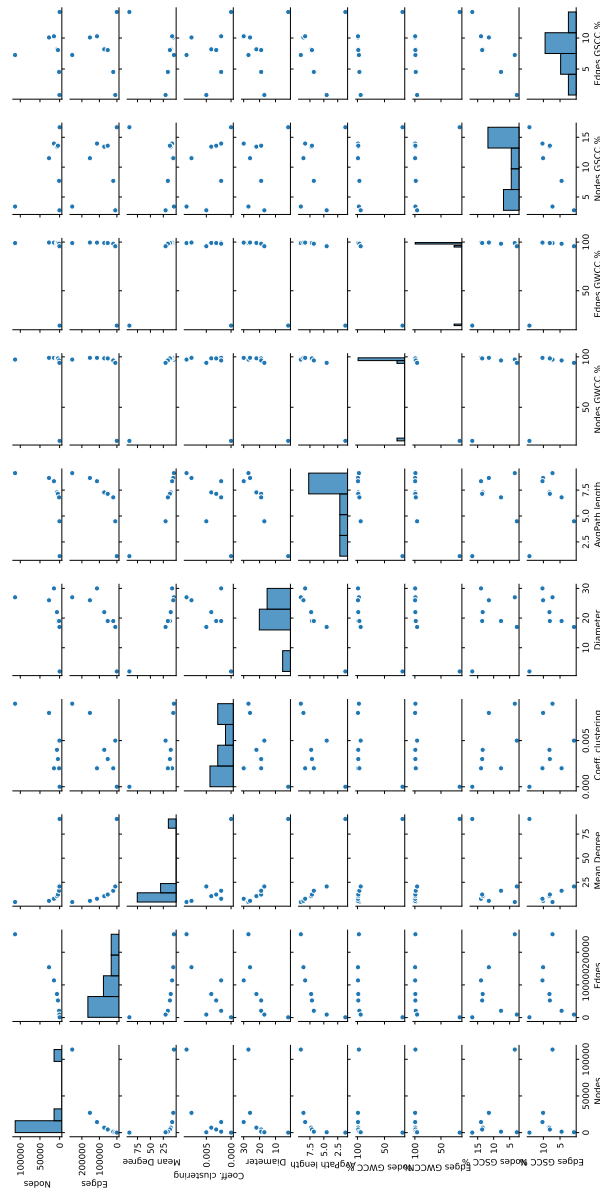
## References

1. Akella A, Chawla S, Kannan A, Seshan S. Scaling properties of the Internet graph. Association for Computing Machinery, New York, NY, USA, 337–346. 2003. <https://doi.org/10.1145/872035.872087>
2. Amato M, Fatès N, Gobbi L: The economics and algorithmics of an integral settlement procedure on B2B networks September 1, 2021 <http://dx.doi.org/10.2139/ssrn.3915380>

3. Barabasi A-L., Bonabeau E: Scale-Free Networks. 2003, *Scientific American*, 288(5), 60–69. <http://www.jstor.org/stable/26060284>
4. Behrisch M, Bach B, Henry Riche N, Schreck T, Fekete JD.: Matrix Reordering Methods for Table and Network Visualization. *Computer Graphics Forum*, Wiley, 2016, 35, pp.24. <https://inria.hal.science/hal-01326759/document>
5. Cox, D, Snell, E :Special logistic analyses. *Analysis of binary data* 2nd ed London: Chapman and Hall, 26-105. (1989).
6. Faloutsos M, Faloutsos P, Faloutsos C : On power-law relationships of the Internet topology. *SIGCOMM Comput. Commun. Rev.* 29, 4 (Oct. 1999), 251–262. <https://doi.org/10.1145/316194.316229>
7. Fortunato S: Community detection in graphs. *Physics Reports*. Volume 486, Issues 3–5, 2010, Pages 75-174, ISSN 0370-1573, <https://doi.org/10.1016/j.physrep.2009.11.002>.
8. Fricke D and Lux T: On the Distribution of Links in the Interbank Network: Evidence from the e-Mid Overnight Money Market, January, 2013 <https://EconPapers.repec.org/RePEc:zbw:ifwkwp:1819>
9. Gaffeo E, Gobbi L, Molinari M: The economics of netting in financial networks. *J Econ Interact Coord* 14, 595–622 (2019). doi:10.1007/s11403-018-0229-4
10. Gu L, Huang H.L., Zhang X.D.: The clustering coefficient and the diameter of small-world networks. *Acta. Math. Sin.-English Ser.* 29, 199–208 (2013). <https://doi.org/10.1007/s10114-012-0387-6>
11. Iosifidis G, Charette Y, Airoidi EM, Littera G, Tassioulas L, Christakis NA.: Cyclic motifs in the Sardex monetary network. *Nat Hum Behav* 2, 822–829 (2018) doi: 10.1038/s41562-018-0450-0
12. Klein M: A primal method for minimal cost flows with applications to the assignment and transportation problems. *Management Science*, 14(3):205–220, 1967. <https://www.jstor.org/stable/2627433>
13. Lambiotte R, Lefebvre E, Blondel V, Guillaume JL.: Fast unfolding of communities in large networks. July, 2008 <https://doi.org/10.1088/1742-5468/2008/10/P10008>
14. Leinonen H, Soramäki K: Optimizing liquidity usage and settlement speed in payment systems. Discussion Papers 16/1999, Bank of Finland, Helsinki, 1999. <https://dx.doi.org/10.2139/ssrn.228249>
15. Matthias Leclerc Michael M. Güntzer, Dieter Jungnickel. Efficient algorithms for the clearing of interbank payments. pages 212–219. 1998. [https://doi.org/10.1016/S0377-2217\(97\)00265-8](https://doi.org/10.1016/S0377-2217(97)00265-8)
16. Newman M: Communities, modules and large-scale structure in networks, December, 2011, *Nature Physics*, <https://doi.org/10.1038/nphys2162>
17. Seidman S: Network structure and minimum degree, *Social Networks*, Volume 5, Issue 3, 1983, Pages 269-287, ISSN 0378-8733, [https://doi.org/10.1016/0378-8733\(83\)90028-X](https://doi.org/10.1016/0378-8733(83)90028-X)
18. Travers J, Milgram S: An Experimental Study of the Small World Problem, *Sociometry*, Vol. 32, No. 4 (Dec., 1969), pp. 425-443 <http://www.jstor.org/stable/2786545>
19. Watts D, Strogatz S: Collective dynamics of ‘small-world’ networks. *Nature* 393, 440–442 (1998). <https://doi.org/10.1038/30918>

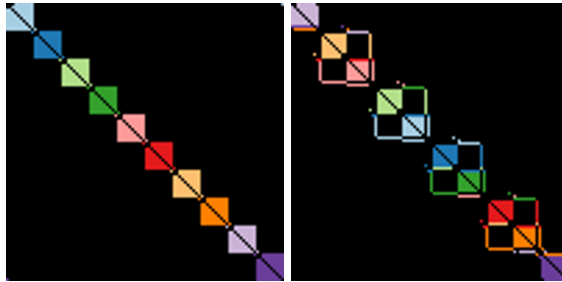
## Annexes

### Correlation of statistical data

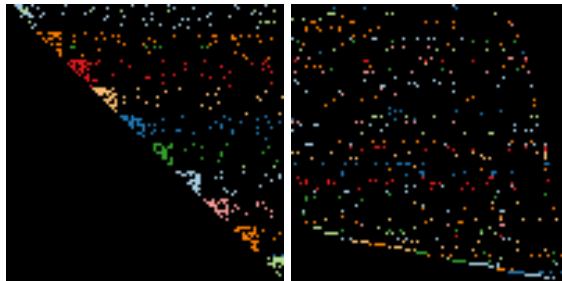


**Fig. 5.** Correlogram for the statistical data of monthly graphs

### Matrix reordering for caveman and noisy caveman



**Fig. 6.** Comparison of adjacency matrices between a standard caveman graph (left) and its bandwidth minimisation reordering (right). The colouration of nodes depends on the community the node is part of when constructing our graph.



**Fig. 7.** Comparison of adjacency matrices between a noisy caveman graph (left) and its bandwidth minimisation reordering (right). The colouration of nodes depends on the community the node is part of when constructing our graph.