Partitionnement de Graphe pour l'Identification de Goulots d'Étranglement Partagés

Paul GRISLAIN[2,3] Emmanuel LOCHIN[3] Jean-Yves TOURNERET[1,2]

[1]Université de Toulouse, Institut de Recherche en Informatique de Toulouse (IRIT), CNRS, Toulouse INP-ENSEEIHT, Toulouse

[2] Laboratoire TéSA, 7 boulevard de la gare, 31500 Toulouse, France

[3]École Nationale de l'Aviation Civile, 7 avenue Edouard Belin, 31400 Toulouse, France

Résumé – Un réseau IP peut être représenté sous la forme d'un graphe où les nœuds sont des routeurs et les arêtes des liens de communication IP. Ce graphe aide à analyser les interactions et les flux d'information au sein du réseau. Chaque routeur, agissant comme une file d'attente, gère le trafic avec une capacité de mémoire tampon et un débit de sortie. Lorsque le trafic entrant dépasse cette capacité, une congestion se produit, dégradant le service. Identifier ces goulots d'étranglement est crucial pour évaluer la performance du réseau. Cet article explore une méthode de partitionnement de graphe permettant de regrouper les flux partageant un goulot commun à l'aide d'un modèle probabiliste plus général que ceux de la littérature.

Abstract – An IP network can be modeled as a graph whose nodes represent routers and edges represent IP communication links. This model helps the interactions and information flows within the network to be analyzed. Each router, acting as a queue, manages the traffic with a buffer and an output rate. When incoming traffic exceeds this capacity, congestion occurs at a so-called bottleneck, degrading the service. Identifying these bottlenecks is crucial for evaluating the network performance. This article explores a graph partitioning method to group flows sharing a common bottleneck, using a probabilistic model more general than the existing models.

1 Introduction

Les communications sur internet prennent la forme de flux de paquets acheminés à travers des réseaux de routeurs. Chaque routeur est représenté comme une file d'attente composée d'une mémoire tampon et d'un service correspondant à son débit de sortie. La capacité de cette file et le volume de trafic qui la traverse influencent directement l'écoulement du trafic. Lorsque le trafic entrant dépasse la vitesse de service de la file, il y a congestion et les paquets excédentaires sont mémorisés. Le goulot d'étranglement est la file limitante du chemin emprunté par les paquets d'un même flux. Elle bride et dégrade la communication ; son service marque particulièrement le flux et son impact est mesuré à l'aide d'une métrique. Un test de présence de goulot est déduit de cette métrique afin de classer les flux observés dans un même routeur en fonction de leur goulot commun. Dans [1], les classes sont initialisées avec des flux "référents", éloignés au-delà d'un seuil prédéfini, puis les autres flux sont agrégés un à un. Dans [2], les auteurs classent progressivement les flux en minimisant l'entropie globale. Chaque nouveau flux classé contribue à améliorer la fiabilité du test mais alourdit les calculs. La première méthode présente l'inconvénient d'un seuil à définir tandis que la seconde est sensible à l'initialisation.

Une méthode alternative est de modéliser le problème par un graphe dont les nœuds sont les flux et les arêtes des tests de goulot commun, puis de classer les flux par goulot avec un algorithme de partitionnement bien optimisé issu de la littérature. La première partie de l'article résume les principaux algorithmes et sélectionne celui qui offre le meilleur compromis vitesse et précision. La seconde partie introduit un nouveau modèle probabiliste de graphe. Ce modèle est ensuite utilisé pour simuler des scénarios permettant d'évaluer la performance des algorithmes. Les paramètres du modèle sont finalement adaptés à des données émulées afin de compléter l'étude sur un cas réaliste. Nous concluons sur l'utilité d'un tel modèle pour l'étude de performance de partitionnement.

2 Partitionnement de graphes

Le partitionnement d'un graphe est un problème complexe (NP difficile) qui consiste à retrouver l'association des N nœuds d'un graphe en K classes $C_1, ..., C_K$. Nous considérons des graphes non orientés et non pondérés associés à une matrice d'adjacence symétrique $\boldsymbol{V} = (v_{ij})$, avec $(i, j) \in \{1, ..., N\}^2$, où $a_{ij} \in \{0, 1\}$ indique si les nœuds i et j appartiennent à la même classe, i.e., $v_{ij} = 1$ si ces nœuds appartiennent à la même classe et $v_{ij} = 0$ sinon. Seule une observation bruitée de cette matrice nous parvient, obtenue par exemple obtenue par des tests statistiques réalisés sur chaque paire de flux de paquets. De nombreuses méthodes ont été proposées dans la littérature pour répartir les nœuds d'un graphe en différentes classes et déterminer le nombre K de ces classes. La résolution du problème met en œuvre un algorithme de parcours des partitions possibles du graphe et une fonction objectif qui évalue ces propositions.

2.1 Recherche d'une solution

Une évaluation exhaustive de toutes les partitions des nœuds d'un graphe est combinatoire, donc l'exploration de ces partitions est en général limitée à un sous-ensemble de partitions d'intérêt. Les algorithmes de partitionnement recherchent donc une solution mais sans preuve d'optimalité. On peut distinguer plusieurs familles d'algorithmes résumées dans cette partie. Les méthodes de partitionnement hiérarchique recherchent la

meilleure partition dans un dendrogramme issu de fusions ou divisions de clusters. L'approche est soit ascendante (algorithmes par agglomération, e.g. [3]), soit descendante (algorithmes par bissections, e.g. [4]). Une fonction objectif permet d'évaluer la meilleure partition, qui est la meilleure coupe du dendrogramme. Les méthodes itératives telles que le recuit simulé ou l'échantillonneur de Gibbs affinent progressivement une partition en déplaçant les nœuds entre les classes [5]. Notons $\boldsymbol{V}$ la vérité terrain, $\boldsymbol{A}$ une estimation de cette matrice obtenue à partir d'observations du réseau et $\hat{\boldsymbol{V}}$ l'estimation de $\boldsymbol{V}$ à partir de $\boldsymbol{A}$. La solution peut s'écrire sous la forme d'une matrice indicatrice $\hat{\boldsymbol{Z}} = (\hat{z}_{kn})$ de taille $K \times N$ qui associe à chaque nœud n une unique classe k, avec $\hat{z}_{kn} = 1$ si $n \in C_k$ et $\hat{z}_{kn} = 0$ sinon. La qualité d'une matrice $\hat{\boldsymbol{Z}}$ est évaluée au moyen d'une fonction objectif souvent définie à l'aide d'une métrique entre la matrice d'adjacence $\boldsymbol{A}$ du graphe observé et la matrice $\hat{\boldsymbol{V}} = \hat{\boldsymbol{Z}}^T \hat{\boldsymbol{Z}}$, où $\hat{\boldsymbol{V}} = (\hat{v}_{ij}) \in \{0,1\}^{N \times N}$ est la matrice d'adjacence du graphe partitionné qui indique si deux nœuds (i,j) sont supposés connectés $(\hat{v}_{ij} = 1)$ ou non $(\hat{v}_{ij} = 0)$, i.e., s'ils appartiennent à la même classe ou pas.

2.2 Fonctions objectifs

Les premiers algorithmes de partitionnement de graphes utilisaient une fonction objectif définie à l'aide d'une métrique de coupe minimale (théorème max-flow min-cut) [6]. D'autres métriques ont été proposées, notamment la modularité Q qui compare la proportion d'arêtes à l'intérieur d'une classe avec la proportion d'arêtes attendue dans le cas d'une répartition aléatoire [4]. La définition de la modularité fait intervenir les termes suivants : 1) $m = \sum_{i,j} a_{ij}$ qui est la somme des degrés des nœuds, 2) $m_k = \sum_{i,j} a_{ij} \mathbb{1}(i \in C_k)$ qui est la somme des degrés des nœuds de la classe k et 3) $e_k = \sum_{i,j} a_{ij} \mathbb{1}((i,j) \in C_k^2)$ qui est le nombre d'arêtes internes à la classe k, où $\mathbb{1}(i \in C_k)$ est la fonction indicatrice de la classe C_k. La modularité est alors définie par :

$$Q = \sum_{k=1}^{K} \left[\frac{e_k}{m} - \left(\frac{m_k}{m} \right)^2 \right]. \quad (1)$$

Lorsque la matrice d'adjacence suit un modèle probabiliste défini, une fonction objectif de référence est obtenue par maximum de vraisemblance. Malheureusement, ces modèles font intervenir des paramètres qui ne sont pas toujours connus a priori et qu'il est souvent difficile d'estimer conjointement avec le partitionnement. Notons que la modularité peut être obtenue à partir de la vraisemblance d'un modèle probabiliste appelé "Planted Partition Model" [7].

2.3 Partitionnement spectral

Les méthodes de partitionnement spectral (spectral clustering) simplifient le problème en projetant les nœuds d'un graphe dans un espace de dimension inférieure au nombre de nœuds N et le résolvent de façon géométrique. Les coordonnées projetées des nœuds sont obtenues en déterminant les vecteurs propres de la matrice d'adjacence, ou d'une forme dérivée de cette matrice, par exemple le Laplacien. Un algorithme de partitionnement spectral suppose de connaître la dimension de l'espace de projection, qui est reliée au nombre recherché de classes. Ce nombre peut être estimé a priori (e.g., avec les valeurs propres de la matrice d'adjacence) ou a posteriori avec la fonction objectif. L'article [8] donne plus de détails sur la manière d'estimer K ainsi que l'importance de calculer les vecteurs propres du Laplacien au lieu de la matrice d'adjacence. Le partitionnement est ensuite effectué à l'aide de l'algorithme k-means. L'algorithme 1 illustre les étapes d'une méthode de partitionnement spectral. Dans [5], l'auteur propose l'algorithme 2 dans lequel une fonction objectif f_o (à définir) sélectionne la meilleure partition obtenue avec l'algorithme spectral pour différents nombres de classes.

Algorithme 1 : Partitionnement spectral [9]

Data : matrice d'adjacence du graphe $\boldsymbol{A}$, nombre de clusters K
Calculer le Laplacien normalisé $\boldsymbol{L} = \boldsymbol{I} - \boldsymbol{D}^{-1}\boldsymbol{A}$ où $\boldsymbol{D}$ est la matrice diagonale des degrés des nœuds et $\boldsymbol{I}$ la matrice identité
Calculer $\boldsymbol{U}$ la matrice des K premiers vecteurs propres $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_K$ associés aux plus petites valeurs propres du Laplacien
Chaque ligne i de $\boldsymbol{U}$ correspond aux coordonnées projetées du nœud i dans un espace de dimension K.
Partitionner les nœuds en K clusters avec l'algorithme k-means.

Algorithme 2 : Algorithme mixte

Data : matrice d'adjacence du graphe $\boldsymbol{A}$, Nombre maximal de clusters $N_{\max}$, vecteur paramètre $\boldsymbol{\theta}$
For $k = 1, ..., N_{\max}$
Initialiser $\mathcal{P}$ par le résultat du partitionnement spectral de $\boldsymbol{A}$ avec k classes
Évaluer $S = f_o(\mathcal{P}; \boldsymbol{A}, \boldsymbol{\theta})$ le score de la partition $\mathcal{P}$.
Améliorer le score de la partition par échantillonnage de Gibbs
Renvoyer la partition la plus vraisemblable.

3 Modèles probabilistes pour graphes

3.1 Planted Partition Model

Le "Planted Partition Model" est une version simplifiée du "Stochastic Block Model". Contrairement à ce dernier, son partitionnement optimal est toujours strict et sa vraisemblance peut être utilisée comme fonction objectif sans hypothèse sur le nombre de clusters. Elle s'écrit de la façon suivante en définissant les paramètres du modèle $p = P(a_{ij} = 1 | v_{ij} = 1)$ et $q = P(a_{ij} = 1 | v_{ij} = 0)$:

$$L(\hat{\boldsymbol{V}}; p, q, \boldsymbol{A}) = \prod_{i,j} p^{a_{ij}\hat{v}_{ij}} (1-p)^{(1-a_{ij})\hat{v}_{ij}} \times \prod_{i,j} q^{a_{ij}(1-\hat{v}_{ij})} (1-q)^{(1-a_{ij})(1-\hat{v}_{ij})}. \quad (2)$$

L'estimateur de $\boldsymbol{V}$ s'obtient en trouvant $\hat{\boldsymbol{V}}$ qui maximise la vraisemblance 2 sous contrainte que $\hat{\boldsymbol{V}}$ est de la forme $\hat{\boldsymbol{Z}}^T \hat{\boldsymbol{Z}}$. Pour cela il faut connaître les probabilités p et q, comme dans [10]. Dans [11] des valeurs de p et q sont fixée et un terme pénalisant le nombre de classes ajouté à la vraisemblance.

3.2 Nouveau modèle de partitionnement

Le "Planted Partition Model" a l'avantage de modéliser systématiquement un partitionnement strict mais ne convient pas aux données réseaux. En effet, les probabilités de détection de goulots et de fausse alarme dépendent de paramètres propres aux nœuds (débit, variabilité des temps d'inter-arrivée des flux de paquets,...). Aussi, nous proposons un modèle probabiliste plus général, appelé "Modèle par Nœuds" dans lequel chaque nœud n du graphe est associé à des attributs p_n, q_n telles que $1 \geq p_n > q_n \geq 0, \forall n$. On définit la probabilité de présence d'une arête entre deux nœuds (i,j) par une moyenne de leurs attributs :

$$P(a_{ij}=1|v_{ij}=1) \quad = \frac{p_i+p_j}{2} \triangleq p_{ij},$$
$$P(a_{ij}=1|v_{ij}=0) \quad = \frac{q_i+q_j}{2} \triangleq q_{ij}.$$

Ce modèle permet de générer des partitionnements plus proches de données émulées (voir fig 4). La vraisemblance associée à ce modèle a la forme suivante :

$$L(\hat{\boldsymbol{V}}; \boldsymbol{A}, \boldsymbol{p}, \boldsymbol{q}) = \prod_{i,j} p_{ij}^{a_{ij}\hat{v}_{ij}} (1-p_{ij})^{(1-a_{ij})\hat{v}_{ij}} \times \prod_{i,j} q_{ij}^{a_{ij}(1-\hat{v}_{ij})} (1-q_{ij})^{(1-a_{ij})(1-\hat{v}_{ij})}, \quad (3)$$

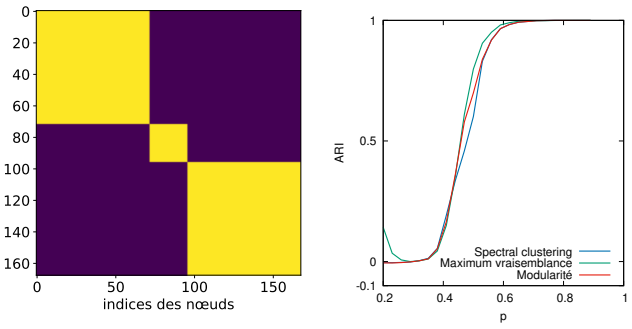
avec $\boldsymbol{p} = (p_1, ..., p_N)^T$ et $\boldsymbol{q} = (q_1, ..., q_N)^T$. Un autre modèle plus fin qualifié d'« extended Planted Partition Model » a été étudié dans [12]. Ce modèle ajoute un unique paramètre à chaque nœud, ce qui implique de conserver un même rapport $\frac{p_n}{q_n}$ pour tous les nœuds. L'hypothèse ne semble pas réaliste.

4 Performance de la modularité

Cette section étudie la performance de l'algorithme 2 maximisant la modularité. Nous comparons ses résultats avec ceux obtenus par l'algorithme 1 de partitionnement spectral (sachant le nombre de classes) et la vraisemblance lorsque les paramètres associés au partitionnement sont connus [1]. L'indice de performance utilisé pour l'analyse des algorithmes de partitionnement est l'Adjusted Rand Index (ARI) [13] compris entre -1 (partition complètement différente) et 1 (partition conforme à la vérité terrain), sachant que ARI $= 0$ indique un partitionnement aléatoire. La matrice d'adjacence de référence associée au graphe idéal est présentée dans la figure 1a. Cette matrice est utilisée dans toutes les expériences de cet article. Les matrices d'adjacence estimées seront toujours présentées avec le même ordre des nœuds pour mettre en évidence les classes correspondant à des blocs sur la diagonale.

4.1 Données synthétiques

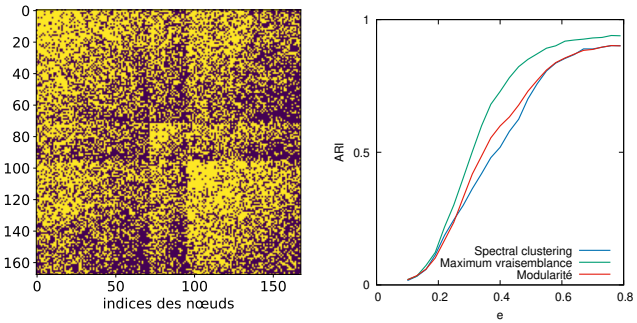
Les algorithmes sont d'abord testés avec le Planted Partition Model. Des résultats similaires étant obtenus pour différentes valeurs de q, cet article se limite à $q = 0.3$ avec p variable. Pour chaque valeur de p, 60 matrices différentes ont été simulées sur la base de la matrice théorique 1a. La figure 1b montre que les solutions obtenues par partitionnement spectral, maximum de vraisemblance (calculé par échantillonneur de Gibbs) et modularité s'améliorent lorsque p augmente. Le maximum de vraisemblance et la modularité améliorent effectivement les partitions proposées par l'algorithme spectral. Ces résultats confirment que la modularité est une bonne approximation de la vraisemblance du Planted Model.



(a) Vraie matrice d'adjacence. (b) ARIs moyens.

FIGURE 1 : Matrice d'adjacence et ARI pour le Planted Model.

Les expériences suivantes s'intéressent au Modèle par Nœuds. Les paramètres du Modèle par Nœuds sont trop nombreux pour effectuer une analyse de performance exhaustive à l'aide de tous les vecteurs $\boldsymbol{p}$ et $\boldsymbol{q}$ du modèle. Nous proposons d'évaluer les différentes méthodes de partitionnement en générant des vecteurs $\boldsymbol{p}$ et $\boldsymbol{q}$ dans leur domaine de définition et en faisant varier un paramètre $e = \sup_n\{p_n - q_n\}$, qui est le maximum des écarts entre les vecteurs $\boldsymbol{p}$ et $\boldsymbol{q}$ et contrôle la difficulté du partitionnement. Pour chaque valeur de e, 60 matrices d'adjacence sont simulées [1]. Les résultats obtenus avec le Modèle par Nœuds sont représentés dans la figure 2b. La performance du partitionnement s'améliore lorsque e augmente (comme dans la figure 1b), bien qu'ici certains nœuds dont l'écart $p_n - q_n$ est trop faible soient mal classés. La vraisemblance permet d'atteindre les meilleurs résultats, mais la modularité reste une bonne alternative. La figure 2a montre une matrice d'adjacence obtenue avec le Modèle par Nœuds avec $e = 0.4$. Les résultats de partitionnement sont présentés dans la figure 3. Les résultats de l'algorithme spectral et de la modularité sont identiques et un peu moins bons que ceux obtenus avec la vraisemblance.



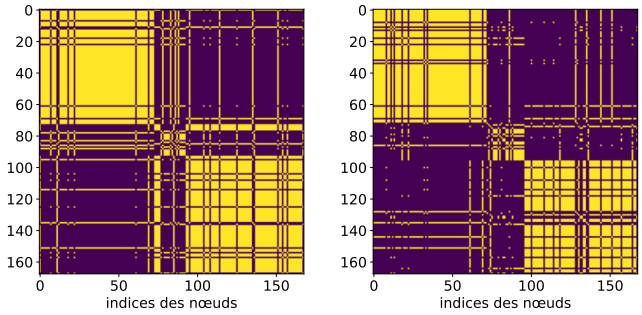
(a) Matrice d'adjacence simulée, $e = 0.4$. (b) ARIs moyens.

FIGURE 2 : Matrice et ARI pour le Modèle par Nœuds.

4.2 Données émulées

Cette partie considère des données issues d'une expérience effectuée dans un réseau émulé. Nous devons trouver 3 goulots,

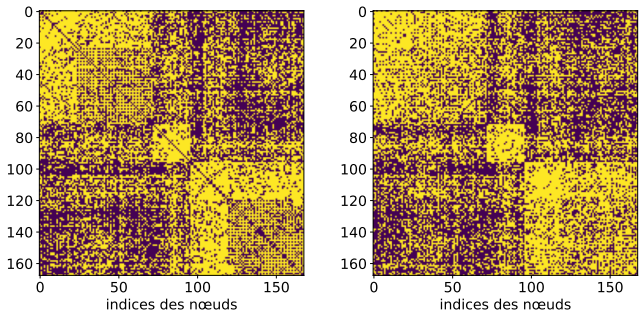
[1] Les détails d'implémentation sont disponibles ici lien code. On notera que le nombre maximal de classes a été fixé à 10, en cohérence avec le nombre de goulots à identifier dans les réseaux internet que nous étudions.



(a) Modularité - Spectral. (b) Maximum de vraisemblance.

FIGURE 3 : Matrices solutions pour la matrice simulée 2a.

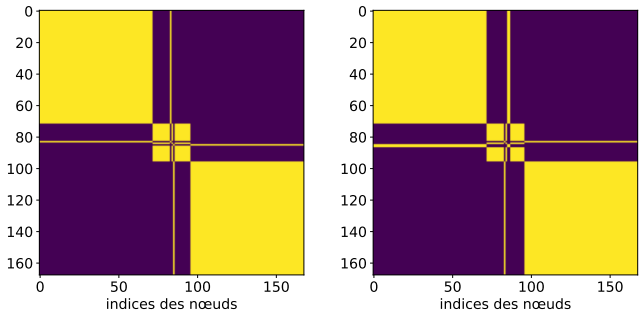
i.e., 3 classes. Des files d'attente en amont régulent le trafic et conduisent à former des sous-classes diagonales moins connectées. La figure 4a illustre la matrice des tests de présence de goulots communs pour les données émulées. En comparaison, la figure 4b montre une matrice simulée à partir du Modèle par Nœuds avec des paramètres $\boldsymbol{p}$ et $\boldsymbol{q}$ optimisés. Les sous-classes ont été éliminées, mais les matrices des deux figures 4a et 4b sont très similaires, confirmant l'intérêt du Modèle par Nœuds.



(a) Données émulées. (b) Modèle par Nœuds.

FIGURE 4 : Matrices d'adjacences.

Les résultats obtenus avec la modularité et spectral clustering sont très bons et identiques : 2 nœuds seulement sont mal classés dans les données émulées et 3 sur les données simulées. Les matrices d'adjacence obtenues sont montrées sur les figures 5a et 5b. En comparaison, le maximum de vraisemblance commet une erreur de classification sur la matrice simulée.



(a) Données émulées. (b) Modèle par Nœuds.

FIGURE 5 : Partitionnements des matrices d'adjacence.

5 Conclusion

Après avoir sélectionné un algorithme de partitionnement de la littérature, nous avons proposé un nouveau modèle probabiliste de graphe partitionné appelé "Modèle par Nœuds" qui est une extension du "Planted Model". L'algorithme permet de résoudre des problèmes simulés par les modèles puis émulés dans un réseau, validant ainsi l'approche par graphe pour la recherche de goulots communs. La métrique de modularité donne des résultats similaires à l'estimateur du maximum de vraisemblance pour le modèle générateur des données. L'intérêt du "Modèle par Nœuds" est de diviser le problème initial de détection de goulot en deux parties : partitionnement et test de présence de goulot. D'un côté le modèle garantit l'existence d'une partition optimale du graphe, de l'autre il s'adapte bien aux données émulées grâce à ses nombreux paramètres. Des axes de progression existent encore pour ces deux parties. Le premier est l'amélioration des performances de l'algorithme de clustering en exploitant les relations entre les conditions du réseau internet et les paramètres du modèle. Ces paramètres semblent dépendre avant tout des débits des flux, mais nos efforts n'ont pas abouti plus loin à présent. Enfin le modèle pourrait être amélioré pour le rendre robuste à la présence d'éléments aberrants (outliers).

6 Remerciements

Les auteurs de cet article aimeraient remercier leur collègue Herwig Wendt pour d'intéressantes discussions relatives à cet article et Thalès Alenia Space pour son support financier.

Références

[1] D. A. Hayes, M. Welzl, S. Ferlin, D. Ros, and S. Islam, "Online identification of groups of flows sharing a network bottleneck," *IEEE/ACM Transactions on Networking*, vol. 28, no. 5, pp. 2229–2242, 2020.

[2] D. Katabi, I. Bazzi, and X. Yang, "A passive approach for detecting shared bottlenecks," in *Proc. Int. Conf. Computer Communications and Networks*, (Scottsdale, AZ, USA), pp. 174–181, 2001.

[3] T. Bonald, B. Charpentier, A. Galland, and A. Hollocou, "Hierarchical graph clustering using node pair sampling," *arXiv preprint arXiv :1806.01664*, 2018.

[4] M. E. Newman, "Modularity and community structure in networks," *Proc. of the national academy of sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.

[5] S. Fortunato, "Community detection in graphs," *Physics reports*, vol. 486, no. 3-5, pp. 75–174, 2010.

[6] B. W. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *The Bell system technical journal*, vol. 49, no. 2, pp. 291–307, 1970.

[7] M. E. Newman, "Equivalence between modularity optimization and maximum likelihood methods for community detection," *Physical Review E*, vol. 94, no. 5, p. 052315, 2016.

[8] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, pp. 395–416, 2007.

[9] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, 2000.

[10] M. Cucuringu, X. Dong, and N. Zhang, "Maximum likelihood estimation on stochastic blockmodels for directed graph clustering," *arXiv preprint arXiv :2403.19516*, 2024.

[11] Y. Chen, S. Sanghavi, and H. Xu, "Improved graph clustering," *IEEE Trans. Inf. Theory*, vol. 60, no. 10, pp. 6440–6455, 2014.

[12] K. Chaudhuri, F. Chung, and A. Tsiatas, "Spectral clustering of graphs with general degrees in the extended planted partition model," *Journal Machine Learning Research*, pp. 35–1, 2012.

[13] A. J. Gates and Y.-Y. Ahn, "The impact of random models on clustering similarity," *Journal of Machine Learning Research*, vol. 18, no. 87, pp. 1–28, 2017.