

Comparaison de la lisibilité des graphes en représentation nœuds-liens et matricielle

Mohammad Ghoniem

Ecole des Mines de Nantes
4, rue Alfred Kastler. B.P.20722
44307 NANTES Cedex 3
Mohammad.Ghoniem@emn.fr

Jean-Daniel Fekete

INRIA Futurs/LRI
Bât 490, Université Paris-Sud
9105 Orsay Cedex
Jean-Daniel.Fekete@inria.fr

Philippe Castagliola

Ecole des Mines de Nantes/IRCyN
4, rue Alfred Kastler. B.P.20722
44307 NANTES Cedex 3
Philippe.Castagliola@emn.fr

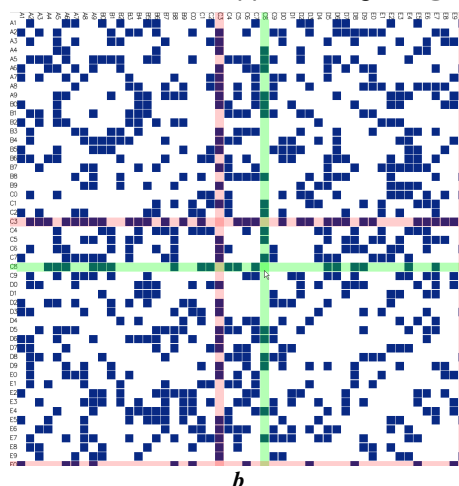
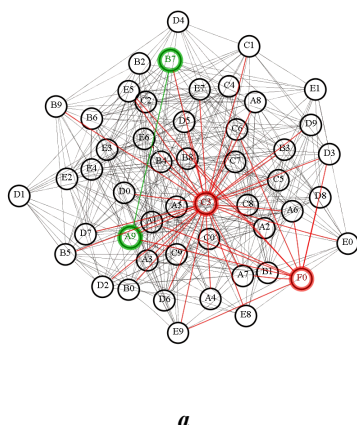


Figure 1 Deux représentations du meme graphe non orienté comportant 50 sommets et 400 arcs. Le diagramme nœuds-liens a) est construit à l'aide du programme "neato" et la representation matricielle b) est construite à l'aide de notre outil VisAdj.

RESUME

Cet article décrit une taxonomie de tâches sur les graphes et une expérience contrôlée visant à évaluer la lisibilité de deux représentations de graphes : les matrices d'adjacences et les diagrammes nœuds-liens. Cette expérience porte sur un sous-ensemble des tâches décrites et fournit des indications importantes sur l'utilisation des représentations en fonction de la taille et de la densité des graphes. Nous montrons par exemple que pour des tailles de graphes supérieures à une vingtaine de sommets, la visualisation matricielle permet d'accomplir la plupart des tâches évaluées plus efficacement que la représentation en nœuds-liens. Seule la recherche de chemin est à l'avantage de la représentation en nœuds-liens.

MOTS CLES : Visualisation de graphes, représentation matricielle, représentation nœud-lien, lisibilité, évaluation.

ABSTRACT

This article describes a taxonomy of tasks on graphs and a controlled experiment for assessing the readability of

two graph representations: matrices and node-link diagrams. The experiment only concerned a subset of the described tasks but provides important insights on the use of representations depending on the graph sizes and densities. It shows that, for graphs larger than 20 nodes or sufficiently dense, most tasks are performed more efficiently on the matrix-based representation. Only some path related tasks score better on node-link diagrams.

CATEGORIES AND SUBJECT DESCRIPTORS: H.5 [Information Interfaces and Presentation]: User Interfaces – Evaluation ; I.3 [Computer Graphics]: Picture/Image Generation – Display Algorithms.

GENERAL TERMS: Visualization of Graphs.

KEYWORDS: Visualization of graphs, adjacency matrices, node-link representation readability, evaluation.

INTRODUCTION

Les diagrammes nœuds-liens¹ ont depuis longtemps été utilisés pour représenter les graphes. De très nombreux travaux issus de la communauté du *graph drawing* portent sur diverses techniques de placement de nœuds tenant compte de règles d'esthétique telles que la minimisation du nombre d'intersections, la minimisation du rapport entre le lien le plus long et le lien le plus court et

Copyright © 2004 by the Association for Computing Machinery, Inc. permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Publications Dept., ACM, Inc., fax +1 (212) 869-0481, or permissions@acm.org. IHM 2004 Aug 30 – Sep 3, 2004, Namur, Belgium Copyright 2004 ACM 1-58113-926-8 \$5.00

¹ Dans cet article, nous employons le terme « lien » pour désigner la représentation d'un arc du graphe, et le terme « nœud » pour désigner la représentation d'un sommet.

la révélation des symétries [2]. La majorité de ces travaux se sont attachés à l'optimisation des algorithmes répondant à ces critères d'esthétique, tandis que des études moins nombreuses se sont consacrées à leur validation expérimentale sur le plan cognitif. On peut citer notamment les expériences contrôlées menées par Purchase et al. sur support papier [12], puis informatique [10, 11]. Ces études portent sur des graphes de petite taille fabriqués à la main (une vingtaine de sommets et vingt ou trente arcs), évaluant cinq critères d'esthétique et huit algorithmes de dessin de graphes. Elles montrent en substance que, si certains critères d'esthétique (e.g. minimiser le nombre d'intersections) pris isolément permettent d'améliorer la perception du graphe examiné, il est difficile néanmoins d'affirmer qu'un algorithme est meilleur qu'un autre de ce point de vue. De même, Ware et Purchase ont réalisé des études validant certaines propriétés esthétiques énoncées dans la communauté du *graph drawing*, comme l'influence de la continuité sur la perception des chemins [13]. Dans le domaine de la visualisation d'information, la représentation nœud-lien a été déclinée sous diverses formes tant en 2D qu'en 3D [5, 9]. Néanmoins, toutes ces techniques de visualisations se sont très vite confrontées aux problèmes d'occlusion dus à l'enchevêtrement des liens, dès que la taille du graphe ou la densité des liens augmentent un peu. Il devient difficile pour l'utilisateur d'explorer le graphe visuellement et d'interagir avec ses éléments constitutifs.

La visualisation matricielle des graphes en revanche évacue complètement les problèmes d'occlusion et représente donc un potentiel intéressant mais beaucoup moins populaire. Dans cet article, nous comparons les deux représentations pour mettre en évidence les avantages respectifs de chaque représentation en fonction de tâches d'exploration que nous avons choisies parmi des tâches génériques.

LA VISUALISATION MATRICIELLE DES GRAPHES

La visualisation matricielle des graphes repose d'un point de vue théorique sur la représentation d'un graphe par sa matrice d'adjacence. Il s'agit d'une matrice de booléens où les sommets du graphe sont disposés en lignes et en colonnes. Les colonnes représentent les sommets de départ et les lignes les sommets d'arrivée, bien que les conventions varient. Lorsque deux sommets sont connectés par un arc, la cellule intersection entre la ligne et la colonne contient la valeur « vrai ». Les autres cellules contiennent la valeur « faux ». La substitution de booléens par des valeurs numériques reflétant un attribut des liens permet d'enrichir la visualisation.

La représentation des graphes par matrices d'adjacence offre une alternative intéressante à la représentation nœuds-liens. Dans [4], Bertin montre qu'il est possible de révéler la structure sous-jacente des réseaux lorsque ceux-ci sont représentés sur une matrice moyennant un jeu de permutations des lignes et des colonnes. Dans [3], les auteurs suggèrent succinctement de représenter la

charge d'un réseau de télécommunication sur une matrice d'adjacence mais portent l'essentiel de leurs efforts sur l'amélioration des représentations en nœuds-liens comme l'affichage de demi liens ou l'affichage des liens importante de manière différée pour éviter qu'ils soient occultés. Plus récemment, dans [8], les auteurs mettent en œuvre une visualisation matricielle multi échelle pour représenter le graphe d'appels de composants d'un gros logiciel d'imagerie médicale. Dans [6], nous avons également montré que la représentation matricielle permettait de déceler clairement la structure sous-jacente d'un graphe de co-activité et d'apprécier l'activité en son sein au fil du temps alors que la représentation équivalente en nœuds-liens est inexploitable. Une application spécifique au domaine de la programmation par contraintes nous a servi d'illustration.

EVALUATION DES DEUX TECHNIQUES DE REPRESENTATION

La comparaison entre deux représentations ne peut se faire que sur une liste de tâches. La liste des tâches utiles ou importantes concernant l'exploration de graphe est sans fin. Il suffit pour s'en convaincre de prendre un exemple de graphe concret, le Web par exemple, et d'énumérer toutes les tâches que l'on voudrait réaliser à partir de ce graphe. Pour éviter de tomber dans ce puits infini, nous avons abordé le problème en partant des tâches les plus génériques de la visualisation d'information et nous les avons instanciées pour la visualisation de graphe. Nous pensons que la lisibilité d'une représentation se rapporte à la capacité de répondre à un certain nombre de questions à partir d'une vue d'ensemble (*overview*). Pour un graphe, certaines questions sont liées à sa topologie tandis que d'autres peuvent être liées à des attributs attachés à des éléments du graphe. Les questions les plus génériques – c'est-à-dire indépendantes de la sémantique des données représentées – liées à la topologie d'un graphe se rapportent à ses sommets, à ses arcs et chemins et à ses sous-graphes.

Caractéristiques élémentaires des sommets : on peut être intéressé par trouver le nombre de sommets (cardinalité), les sommets isolés, un sommet particulier (par son nom), les sommets les plus connectés ou les moins connectés.

Caractéristiques élémentaires des chemins : nombre de liens, existence d'un voisin commun, existence d'un chemin entre deux nœuds, chemin le plus court, nombre de voisins d'un sommet, cycles et chemins critiques.

Caractéristiques élémentaires des sous-graphes : on peut s'intéresser à un sous-graphe quelconque, tous les sommets accessibles à partir d'un ou plusieurs sommets (composante connexe) ou des groupes de connectivité forte (*clusters*).

Comparer la lisibilité des représentations de graphes devrait donc, en principe, prendre en compte toutes ces caractéristiques génériques pour déterminer les plus faciles à observer à partir d'une visualisation matricielle et celles pour lesquelles il est préférable ou raisonnable d'utiliser un diagramme nœuds-liens. Dans la suite de

cet article, nous présentons une étude visant à remplir cet objectif sur un sous-ensemble de ces caractéristiques, pour des raisons de temps.

Lisibilité d'une représentation de graphe

On peut raisonnablement définir la lisibilité d'une représentation graphique comme étant l'aisance relative avec laquelle l'utilisateur peut accéder à l'information qu'il cherche. Autrement dit, plus une représentation graphique est lisible pour une tâche donnée, plus vite on pourra réaliser cette tâche et moins on fera d'erreurs en l'effectuant. Si l'utilisateur répond rapidement et correctement, la représentation est très lisible pour la tâche considérée. S'il lui faut beaucoup de temps ou que la réponse formulée est erronée, la lisibilité de la représentation laisse à désirer.

Dans notre évaluation, nous avons sélectionné les tâches génériques suivantes :

- Tâche 1 : estimer approximativement le nombre de nœuds dans le graphe.
- Tâche 2 : estimer approximativement le nombre de liens dans le graphe.
- Tâche 3 : trouver le nœud ayant le plus de liens.
- Tâche 4 : trouver un nœud donné.
- Tâche 5 : déterminer si un lien relie deux nœuds donnés.
- Tâche 6 : déterminer si deux nœuds possèdent un voisin commun.
- Tâche 7 : déterminer un chemin entre deux nœuds.

Les tâches 1 et 2 portent sur la perception de la vue d'ensemble du graphe. Les tâches 4 et 5 portent sur la perception d'éléments donnés du graphe dans la vue d'ensemble. Les tâches 3, 6 et 7 portent sur des caractéristiques topologiques du graphes ou de certains de ses éléments : le nœud le plus connecté, un voisin commun à deux nœuds et un chemin entre deux nœuds.

Hypothèses préliminaires

Nous constatons que les représentations traditionnelles en nœuds-liens deviennent illisibles à cause de l'enchevêtrement des liens ou de leur longueur. De plus, certaines tâches nécessitant un parcours séquentiel des éléments du graphe, comme la recherche d'un nœud, sont très pénibles lorsque le nombre de nœuds devient trop important car les nœuds ne sont généralement pas ordonnés suivant l'ordre alphabétique par les algorithmes de placement. Nous partons donc de la supposition que la taille du graphe, i.e. le nombre de nœuds qu'il comporte, et la densité des liens en son sein sont deux facteurs prépondérants dans la lisibilité de sa représentation graphique. La densité d'un graphe désigne traditionnellement le rapport entre le nombre d'arcs et le nombre de sommets, mais cette quantité ne reflète pas une complexité visuelle.

Nous définissons la densité d des liens au sein du graphe par la formule : $d = \frac{l}{n^2}$, où l est le nombre liens et n le nombre de nœuds dans le graphe. Cela correspond

à la racine carrée du nombre de liens présents dans le graphe par rapport au nombre maximum de liens si le graphe était entièrement connecté. Cette définition de la densité reflète visuellement le taux de remplissage de la matrice d'adjacence.

La représentation matricielle présente deux avantages principaux à nos yeux susceptibles d'influencer la réalisation de certaines tâches. Premièrement, l'absence de chevauchements, contrairement à la représentation en nœuds-liens, devrait faciliter les tâches liées à la détection d'un lien (rechercher un lien entre deux nœuds, déterminer la présence d'un voisin commun). Deuxièmement, le caractère ordonnable des matrices ramène la recherche d'un nœud donné à trouver un élément dans une liste ordonnée, chose qui se fait aisément quelle que soit la taille du graphe.

En revanche, lorsqu'il s'agit de construire un chemin entre deux nœuds, la vue nœuds-liens semble plus performante ; la représentation matricielle est plus complexe du fait du dédoublement des nœuds sur les deux axes de la matrices, l'œil doit sans cesse passer de la ligne à la colonne représentant le même sommet, en l'absence d'une interaction appropriée. Enfin, pour les petits graphes, nous sommes persuadés que l'utilisation d'un diagramme nœuds-liens reste convenable et est donc préférable à une représentation matricielle moins intuitive.

Protocole expérimental

Les données. Pour tester nos hypothèses, nous avons choisi trois tailles de graphes (20, 50 et 100 nœuds) et trois densités de liens (0.2, 0.4 et 0.6) pour chaque taille, soit neuf graphes différents en tout (cf. le tableau 1). Pour éviter tout biais lié à des particularités des données choisies, nous avons opté pour des graphes aléatoires non orientés, générés par le serveur de graphes aléatoires (<http://www.ispt.waseda.ac.jp/rgs/>). De plus, pour rendre univoque la tâche portant sur le nœud le plus connecté, nous avons enrichi le nœud de plus haut degré dans ces graphes de 10% de liens supplémentaires distribués aléatoirement. En présence de plusieurs nœuds candidats, l'un d'entre eux a été choisi aléatoirement et enrichi.

Le générateur de graphes aléatoires que nous avons utilisé nomme les nœuds du graphe numériquement selon leur ordre de création si bien que, en l'état, la tâche consistant à compter le nombre de nœuds reviendrait à relever le label le plus élevé. Nous avons décidé de rendre la tâche plus pertinente en renommant les nœuds du graphe alphabétiquement (de A à T pour les graphes de 20 nœuds, de A1 à F0 pour les graphes de 50 nœuds, et de A1 à K0 pour les graphes de 100 nœuds).

taille\densité	0.2	0.4	0.6
20	graphe 1	graphe 2	graphe 3
50	graphe 4	graphe 5	graphe 6
100	graphe 7	graphe 8	graphe 9

Tableau 1 : les données expérimentales.

La population. Nous avons fait appel à une population composée d'étudiants et d'enseignants chercheurs en in-

formatique. Tous les sujets savaient ce qu'est un graphe. Aucune autre connaissance en théorie des graphes n'était requise. L'échantillon comprend 36 sujets, tous connaissant la représentation nœuds-liens et ayant participé à cette évaluation de manière volontaire.

Le programme d'évaluation. Nous avons développé un programme d'évaluation qui se charge de représenter les graphes sélectionnés selon les deux techniques de représentation, puis demande à l'utilisateur d'accomplir les tâches susmentionnées et enregistre les réponses fournies ainsi que le temps qu'il a fallu pour les formuler.

En termes d'interactions, notre programme fournit le survol et la sélection des éléments du graphe. Sur les deux visualisations, les nœuds survolés apparaissent en vert ainsi que leurs liens incidents ; les nœuds peuvent être sélectionnés par pointage direct, ils apparaissent alors en rouge ainsi que leurs liens incidents (Figure 1). De même, les liens survolés et leurs extrémités apparaissent en vert.

Une démonstration effectuée sur un jeu de deux graphes nous permet d'expliquer la manière de lire les deux représentations et d'exécuter les tâches demandées. D'abord, l'instructeur manipule le système et fournit des explications. Ensuite, c'est l'utilisateur qui manipule pour s'assurer que les représentations et les tâches sont bien assimilées et que l'interaction avec le système est bien comprise. À l'issue de la démonstration, une mise au point est faite pour s'assurer que l'utilisateur est prêt à entamer l'évaluation proprement dite. Si nécessaire, l'instructeur lui propose de refaire la démonstration une seconde fois. Enfin, l'utilisateur reçoit trois consignes :

1. Il doit répondre le plus vite possible.
2. Il doit répondre juste.
3. Il peut passer à la question suivante sans attendre la fin du temps, lorsqu'il ne réussit pas à trouver la réponse.

Des précautions indispensables. Pour éviter les phénomènes de mémorisation, le système choisit aléatoirement une technique de représentation – matricielle ou en nœuds-liens – et représente en série les neufs graphes retenus, demandant à l'utilisateur d'accomplir les sept tâches pour chacun des graphes. Ensuite, le système passe à la seconde technique et procède comme précédemment. En croisant les techniques de représentation de cette manière, nous faisons en sorte que nos sujets aient autant de chance de commencer l'évaluation par une série de diagrammes nœuds-liens, que par une série de matrices.

Chaque série est subdivisée en deux parties : la première comporte trois graphes relativement simples (les graphes 1, 2 et 4) afin que l'utilisateur se familiarise avec le système ; la seconde comprend les six graphes restants.

Par ailleurs, il est classique d'observer un effet d'apprentissage lorsqu'un utilisateur est confronté à une nouvelle représentation. Nous avons pu mesurer cet effet sur la représentation matricielle lors d'une série de tests préliminaires sur dix utilisateurs où nous faisons varier les tailles puis les densités dans l'ordre croissant. Pour

niveler cet effet indésirable, notre système d'évaluation effectue un tirage aléatoire des graphes au sein des deux parties de chaque série. De cette manière, tous les graphes ont la même probabilité d'apparaître au début, au milieu ou à la fin de leurs parties respectives.

Concernant les tâches 5, 6 et 7, mettant en jeu deux nœuds donnés du graphe, le système présélectionne les nœuds en question pour économiser le temps qu'il faut pour les localiser. Le temps mesuré correspond ainsi au temps net qu'il a fallu pour accomplir ces tâches.

Enfin, chaque séance d'évaluation comportant un total de 126 questions (9 graphes x 2 techniques de visualisation x 7 tâches), nous avons prévu trois pauses pour réduire les effets de fatigue des utilisateurs, une pause de dix minutes entre les deux séries et une pause de cinq minutes entre les deux parties de chaque série. De plus, l'évaluation étant plutôt longue (une heure de manipulation par utilisateur), nous avons fait le choix de limiter le temps de réponse à 45 secondes par question. Passé ce délai, le système passe automatiquement à la question suivante et émet un retour sonore pour attirer l'attention de l'utilisateur. Nous considérons alors que la représentation n'est pas assez efficace pour la tâche en question puisque la réponse ne peut être formulée dans le délai imparti. Le retour sonore sert aussi à inciter l'utilisateur à répondre rapidement aux questions suivantes.

Implantation et paramétrage. Les diagrammes en nœuds-liens sont dessinés à l'aide du programme *neato* et de la librairie *grappa*, tous deux développés par AT&T [1]. Après quelques tests préliminaires, nous avons retenu *neato* parmi trois programmes de dessin de graphes (*dot*, *neato* et *twopi*) pour sa flexibilité et parce qu'il met en œuvre un algorithme connu de placement par champ de forces. La représentation matricielle a été développée par nos soins, en OpenGL sous java, en utilisant la librairie *gl4java*. Nous nous sommes efforcés de paramétrer les deux représentations à leur avantage respectif et de proposer des interactions identiques sur les deux. Un soin particulier a été accordé à la taille des nœuds et à la lisibilité des labels sur les diagrammes nœuds-liens quels que soient le nombre de nœuds et la densité des liens. À cet effet, les labels des nœuds survolés ou sélectionnés étaient affichés par superposition sur un fond semi transparent, éliminant ainsi les problèmes d'occultation dus au chevauchement des liens sur ces nœuds. Étant donné que les graphes étudiés sont non-orientés, nous avons aussi supprimé les flèches aux deux extrémités des liens, ce qui allège sensiblement la représentation nœuds-liens des graphes denses et simplifie les tâches liées aux chemins dans les matrices. Les paramètres ainsi obtenus sont alors stockés dans des fichiers de données au format DOT, chargés à tour de rôle tout au long de l'évaluation. Nous obtenons ainsi la garantie que les différents utilisateurs sont confrontés à des représentations identiques pour un même graphe. De même, nous avons mis à profit le caractère ordonnable des matrices en triant leurs lignes et leurs colonnes dans l'ordre al-

phabétique. Le tri des matrices étant instantané, ces représentations n'ont requis aucun paramétrage préalable. Les tests ont été effectués sur une station Dell munie d'une carte graphique accélérée NVidia GeForce2, un biprocesseur Pentium III à 1Ghz, 512 Mo de mémoire vive, sous Windows 2000. Les visualisations sont affichées en plein écran sur un moniteur 21". Les sujets sont assis à soixante centimètres du moniteur et exécutent toutes les tâches à l'aide d'une souris.

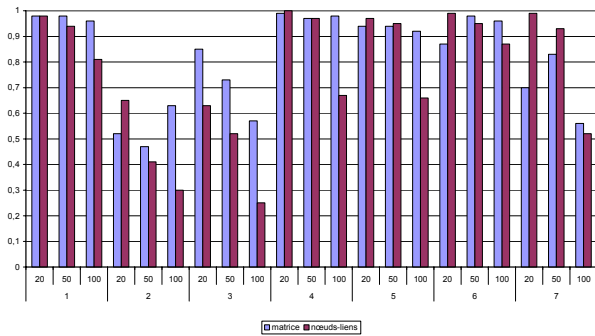


Figure 2 : Taux de bonnes réponses par tâche et par taille. En bleu, la représentation matricielle ; en mauve, les nœuds-liens.

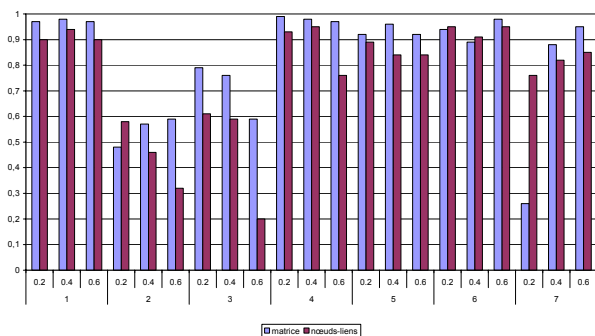


Figure 3 Taux de bonnes réponses par tâche et par densité. En bleu, la représentation matricielle ; en mauve, les nœuds-liens.

Résultats expérimentaux

Les données recueillies ont été analysées à l'aide d'une méthode qualitative graphique (Box-Plot) et à l'aide d'une méthode quantitative (test non paramétrique de Wilcoxon). Le test non paramétrique de Wilcoxon permet de comparer la position centrale de deux échantillons sans supposer de la distribution a priori des données (normalité par exemple). Ce test fournit une p-value qui est une probabilité. Si la p-value obtenue est inférieure à 5%, on déduit que les deux échantillons ont la même valeur centrale, si la p-value est supérieure à 5%, on déduit qu'ils ont des valeurs centrales différentes.

Sur les diagrammes suivants, les données associées aux abscisses impaires correspondent à la représentation matricielle et celles associées aux abscisses paires correspondent à la représentation nœuds-liens. L'axe des ordonnées correspond au temps de réponse. Pour chaque tâche, nous produisons un premier diagramme reflétant l'évolution du temps pour les trois tailles de graphes, et un second retraçant l'évolution du temps pour les trois densités choisies.

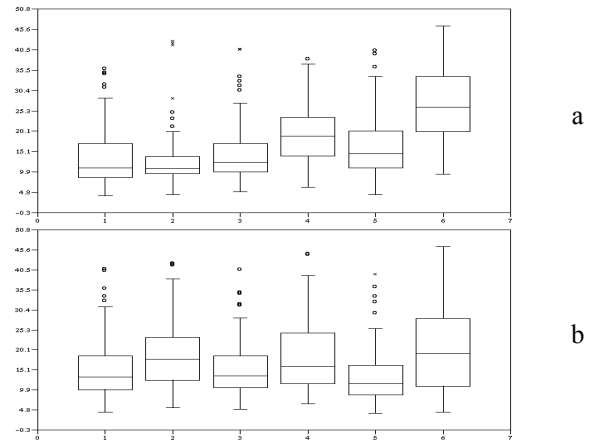


Figure 4

Tâche 1 : nombre de nœuds dans le graphe. Sur la Figure4a, le temps de réponse médian et sa distribution varient peu en fonction de la taille sur la représentation matricielle (abscisses 1, 3, 5) tandis qu'ils varient notablement sur la représentation en nœuds-liens (abscisses 2, 4, 6). On en déduit que, pour déterminer le nombre de nœuds, la lisibilité de la représentation en nœuds-liens se dégrade fortement alors que la représentation matricielle en est moins affectée. Sur la Figure4b, le temps de réponse médian et sa distribution varient peu en fonction de la densité sur la représentation matricielle (abscisse 1, 3, 5) tandis qu'ils croît légèrement, en valeur médiane et en dispersion, sur la représentation en nœuds-liens.

De plus, sur la Figure 2, nous constatons que pour les grands graphes, 96% des utilisateurs ont répondu correctement sur la représentation matricielle, contre 81% pour la représentation en nœuds-liens, soit un écart de 15%, jugé statistiquement significatif selon la méthode des *p-values*. La représentation matricielle affiche un taux de bonnes réponses stable en fonction de la densité (Figure 3) d'environ 97%, avec une avance statistiquement significative de 7% sur la représentation en nœuds-liens pour les graphes de petite et de grande densité.

La lisibilité de la représentation en nœuds-liens est fortement affectée par la taille du graphe et faiblement par la densité des liens, tandis que la lisibilité de la représentation matricielle est légèrement dépendante de la taille et est insensible à l'augmentation de la densité. De plus, la représentation matricielle permet de répondre plus rapidement à la tâche 1 pour les tailles et les densités moyennes et grandes.

Tâche 2 : nombre de liens dans le graphe.

L'estimation du nombre de liens dans le graphe semble relativement indépendante de sa taille et de sa densité à partir d'une taille et d'une densité moyennes. On constate une fracture dans le temps de réponse lorsqu'on passe de 20 à 50 nœuds sur la Figure5a (abscisses 2 et 4) et lorsqu'on passe d'une faible densité à une densité moyenne sur la Figure5b (abscisses 2 et 4). Cependant, aucune différence notable ne peut être relevée entre les deux techniques de représentation pour une taille ou une densité données. Sur la Figure 2, pour les grands gra-

phes, 57% des utilisateurs exécutent cette tâche correctement à l'aide de la représentation matricielle contre 25% seulement pour les nœuds-liens, soit un écart statistiquement significatif de 27 %. Pour les graphes de petite et moyenne tailles, les écarts enregistrés respectivement en faveur de la représentation en nœuds-liens, puis en faveur de la représentation matricielle, ne sont pas statistiquement significatifs. Des conclusions analogues peuvent être tirées lorsque la densité varie (Figure 3).

Tâche 3 : trouver le nœud le plus connecté. Concernant la recherche du nœud le plus connecté, on constate que le temps de réponse pour la représentation en nœuds-liens est sensible à l'augmentation de la taille (Figure 6a), alors qu'il varie peu en fonction de la densité (Figure 6b). Un constat identique peut être fait pour la représentation matricielle. Au regard du temps de réponse, aucune des deux représentations ne semble plus favorable pour la recherche du nœud le plus connecté.

Mais, d'après la Figure 2, 85% des utilisateurs exécutent cette tâche correctement sur les petits graphes à l'aide de la représentation matricielle contre 63% pour les nœuds-liens. Pour les graphes de taille moyenne, 73% répondent correctement à l'aide de la représentation matricielle contre 52% pour les nœuds-liens. Enfin, pour les grands graphes, 57% des utilisateurs répondent correctement à l'aide des matrices contre 25% seulement pour les nœuds-liens. Ces écarts sont statistiquement significatifs selon la technique des p-values. Des conclusions analogues peuvent être formulées au regard de la densité (Figure 3).

Tâche 4 : trouver un nœud par son nom. Au regard du temps de réponse, on constate que les nœuds-liens sont très sensibles à l'augmentation de la taille du graphe (Figure 7a) et sont modérément affectés par la densité des liens (Figure 7b), alors que les matrices n'accusent qu'une très faible dérive en fonction de la taille du graphe et ne semblent pas du tout affectées par la densité des liens. La dispersion du temps de réponse pour la vue matricielle est remarquablement faible dans les deux cas. Pour les petits graphes, les deux représentations se valent à peu près. Le taux de bonnes réponses (Figure 2) est très bon, de l'ordre de 98%, pour la représentation matricielle, quelle que soit la taille du graphe. Ce taux est aussi bon pour la représentation en nœuds-liens, sauf dans le cas des grands graphes où ce taux tombe à 67%, accusant 31% de déficit par rapport à la représentation matricielle. Des conclusions analogues peuvent être atteintes en faisant varier la densité des graphes (Figure 3).

Tâche 5 : Existence d'un lien. Sur la représentation en nœuds-liens, la recherche d'un lien nécessite un temps croissant avec la taille du graphe (Figure 8a) mais ne varie pas significativement en fonction de la densité (Figure 8b). Sur la représentation matricielle, cette tâche est insensible à la variation de la taille et de la densité. On constate un écart significatif entre les deux techniques en faveur de la représentation matricielle pour les graphes de grande taille et pour les graphes de densité moyenne ou grande. Un écart significatif est relevé éga-

lement en faveur de la représentation en nœuds-liens pour les graphes de petite taille.

Les deux représentations accusent d'excellents taux de réponses correctes, de l'ordre de 95%, pour les graphes de petite et moyenne tailles. Pour les graphes de grande taille, la représentation matricielle enregistre 92% de bonnes réponses contre 66% pour les nœuds-liens, soit un écart significatif de 26%.

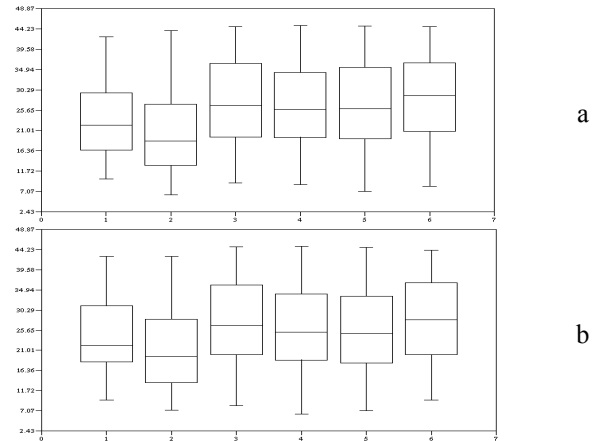


Figure 5

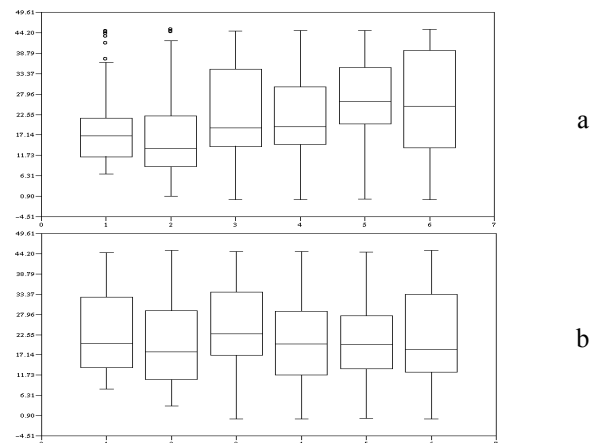


Figure 6

Tâche 6 : Existence d'un voisin commun. Pour déterminer l'existence d'un voisin commun, le temps de réponse est affecté sur la représentation en nœuds-liens lorsque la taille des graphes croît (Figure 9a) mais ne varie pas beaucoup en valeur médiane avec la densité (Figure 9b). Sur les matrices, le temps de réponse ne varie pas en fonction de la taille et s'améliore légèrement en valeur médiane et en dispersion lorsque la densité des liens est grande.

Pour les petits graphes, la représentation en nœuds-liens affiche 99% de bonnes réponses avec une avance significative de 12% sur la représentation matricielle. Cet écart s'inverse pour les grands graphes avec 96% de bonnes réponses pour la représentation matricielle. Les deux techniques affichent des taux de bonnes réponses similaires pour les graphes de taille moyenne (Figure 2). Lorsque la densité varie (Figure 3), les deux représenta-

tions connaissent des taux de bonnes réponses élevés et similaires, d'environ 90%.

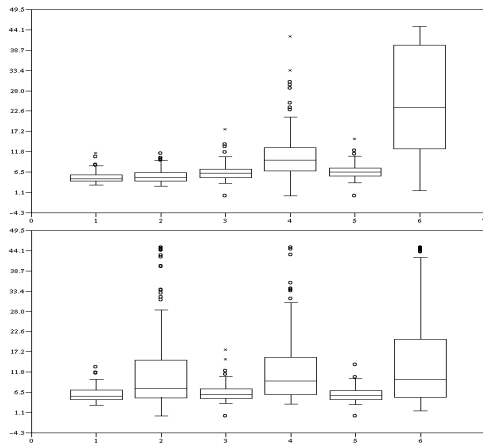


Figure 7

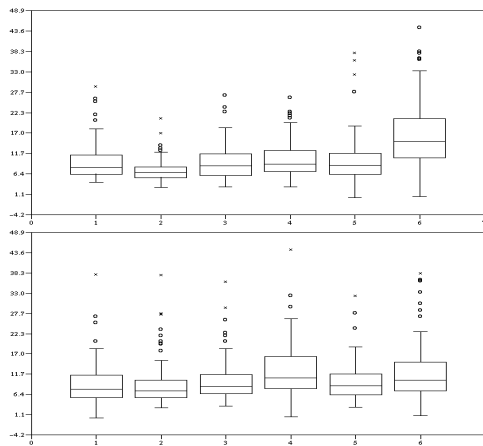


Figure 8

Tâche 7 : Existence d'un chemin. La détermination d'un chemin entre deux nœuds est de plus en plus difficile sur la représentation en nœuds-liens lorsque la taille du graphe augmente (Figure 10a), tandis que le temps de réponse médian augmente légèrement avec la densité (Figure 10b). La représentation matricielle est peu lisible pour cette tâche, indépendamment de la taille du graphe. On constate cependant que sa lisibilité s'améliore jusqu'à surpasser la représentation en nœuds-liens lorsque la densité des liens est grande (Figure 10b). Ce constat se traduit également sur le taux de bonnes réponses qui s'élève à 95% pour la vue matricielle contre 85% pour les nœuds-liens avec les graphes de forte densité (Figure 3). Pour les petits graphes, 99% répondent correctement à l'aide de la représentation en nœuds-liens contre 70% seulement sur la représentation matricielle. Pour les graphes de taille moyenne, la représentation en nœuds-liens connaît 93% de bonnes réponses avec 10% d'avance sur la représentation matricielle. Pour les grands graphes, environ un utilisateur sur deux répond correctement sur les deux représentations avec un écart non significatif statistiquement en faveur des matrices. Enfin, cette tâche

est très largement en faveur de la représentation en nœuds-liens lorsque les graphes sont peu denses.

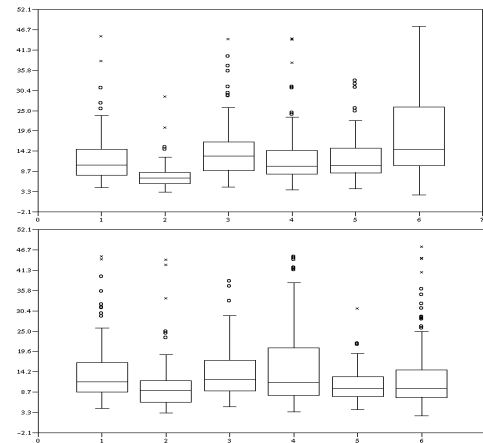


Figure 9

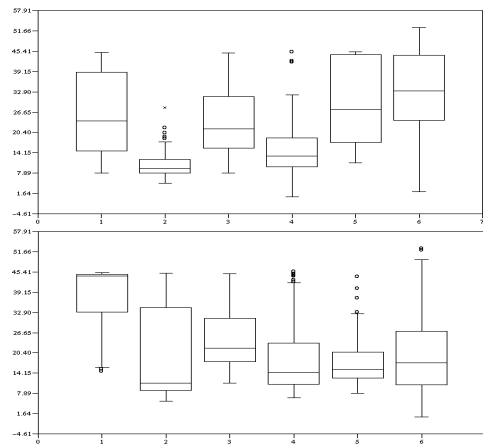


Figure 10

DISCUSSION

Nos hypothèses étaient que les diagrammes nœuds-liens devenaient moins lisibles lorsque leur taille et densité augmentaient. Cette hypothèse est confirmée pour toutes les tâches considérées. Seule la tâche liée aux chemins montre une supériorité des diagrammes nœuds-liens, malgré une dégradation pour les gros graphes. De ce point de vue, il convient de nuancer la conclusion car cette tâche est très difficile à mener à bien visuellement pour des distances supérieures à deux ou trois arcs [11]. Une seconde hypothèse concernait le caractère ordonnable des matrices dans la recherche de sommets et de liens. Les tâches 4 et 5 montrent que cette hypothèse est vérifiée pour les grands graphes et pour les graphes denses.

Pour la tâche 2, la proportion d'erreurs est très importante pour les deux visualisations. Notre interprétation – qui n'a pas encore été vérifiée expérimentalement – est que ce nombre est intrinsèquement difficile à voir sur les représentations nœuds-liens et que les utilisateurs se sont souvent trompés en essayant de les calculer à partir des matrices où les liens apparaissent deux fois pour des graphes non-orientés.

Il est également important de s'interroger sur l'influence du programme de dessin de graphes – neato – que nous avons utilisé dans cette évaluation quant à la possibilité de généraliser les résultats obtenus. Nous nous fondons sur [10] pour rappeler que, pour de petits graphes, l'influence du programme est minime et n'inverserait pas nos observations, ce qui serait vrai a fortiori pour les graphes plus complexes que nous avons étudiés. Mais cela reste à valider expérimentalement.

Dans cette évaluation, nous avons comparé deux représentations de graphes, les matrices d'adjacence et les diagrammes nœuds-liens placés par un algorithme de champ de forces, sur un jeu de neuf graphes aléatoires et un ensemble de sept tâches d'exploration. Dans les limites des paramètres de cette expérience contrôlées [7], nos préconisations sont les suivantes : pour de petits graphes (une vingtaine de sommets), les diagrammes nœuds-liens sont toujours les plus lisibles et les plus familiers. Pour des graphes de taille plus importante, les performances des diagrammes nœuds-liens se dégradent rapidement tandis que les matrices restent lisibles avec des écarts allant jusqu'à 30% de bonnes réponses en plus pour des temps comparables sinon meilleurs. En revanche, les matrices requièrent un apprentissage qui, une fois acquis, permettrait certainement d'améliorer encore les performances que nous avons mesurées. Pour des tâches plus compliquées comme la recherche de chemins, nous pensons qu'une interaction est toujours préférable, par exemple en sélectionnant un nœud et en affichant tous les chemins ou le meilleur chemin menant au nœud placé sous le pointeur. Dans la représentation matricielle, cela peut se faire à l'aide de courbes connectant les liens connexes, c'est-à-dire les cellules représentant les liens.

CONCLUSION

Dans cet article, nous avons donné une liste de tâches génériques pour la visualisation de graphes et nous avons comparé deux représentations de graphes sur un sous-ensemble de ces tâches. Nous avons pu démontrer que ces deux techniques se complètent pour les tâches considérées : nœuds-liens pour les petits graphes, matrices pour les graphes grands ou denses. Les tâches liées aux chemins restent difficiles à réaliser quelle que soit la représentation et nécessitent des interactions adaptées.

La représentation matricielle apparaît donc comme sous-exploitée actuellement, malgré sa grande vitesse de placement et sa meilleure lisibilité pour beaucoup de tâches. Nous pensons qu'une plus grande utilisation de cette représentation permettra un apprentissage qui en facilitera encore la lisibilité. Nous utilisons actuellement la représentation matricielle pour la supervision en temps réel de programmes par contraintes avec explications dans lesquels des graphes parfois très denses évoluent, tant en taille qu'en activité et nous obtenons des résultats encourageants. Nous comptons travailler sur le clustering et l'agrégation avec les matrices pour la visualisation de

très gros graphes, de plusieurs dizaines de milliers de sommets.

REMERCIEMENTS

Nos remerciements vont à Pierre Dragicevic, Véronique Libérati et Vanessa Tico pour leur disponibilité et leurs conseils. Nous remercions également l'ensemble des utilisateurs qui ont bien voulu prendre part à cette évaluation. Ce travail a été financé par le projet RNTL OADYMPAC (<http://contraintes.inria.fr/OADymPPaC/>) que nous remercions pour leur collaboration féconde.

BIBLIOGRAPHIE

1. AT&T Labs Research. Graphviz - open source graph drawing software, 2004.
2. Battista, G.D., Eades, P., Tamassia, R. and Tollis, I.G. *Graph Drawing*. Prentice Hall, 1999.
3. Becker, R.A., Eick, S.G. and Wills, G.J. Visualizing network data. *IEEE Transaction on Visualizations and Graphics*, 1 (1). 16-28.
4. Bertin, J. *Sémiologie graphique : Les diagrammes - Les réseaux - Les cartes*. Editions de l'Ecole des Hautes Etudes en Sciences, Paris, France, 1967.
5. Cohen, R.F., Eades, P., Lin, T. and Ruskey, F., Volume upper bounds for 3D graph drawing. in *Proceedings of the 1994 conference of the Centre for Advanced Studies on Collaborative research*, (Toronto, Ontario, Canada, 1994), IBM Press.
6. Ghoniem, M. and Fekete, J.-D., Visualisation de graphes de co-activité par matrices d'adjacence. in *Actes de la conférence IHM 2002*, (Poitiers, France, 2002), ACM Press, 279-284.
7. Gottsdanker, R. *Experimenting in Psychology*. Prentice-Hall, 1978.
8. Ham, F.v., Using Multilevel Call Matrices in Large Software Projects. in *Proc. IEEE Symp. Information Visualization 2003*, (Seattle, WA, USA, 2003), IEEE Press, 227-232.
9. Herman, I., Melançon, G. and Marshall, M.S. Graph Visualization and Navigation in Information Visualization: a Survey. *IEEE Transactions on Visualization and Computer Graphics*, 6 (1). 24-43.
10. Purchase, H.C., The Effects of Graph Layout. in *Proceedings of the Australasian Conference on Computer Human Interaction*, 1998, p. 80.
11. Purchase, H.C., Carrington, D.A. and Allder, J.-A. Empirical Evaluation of Aesthetics-based Graph Layout. *Empirical Software Engineering*, 7 (3). 233-255.
12. Purchase, H.C., Cohen, R.F. and James, M.I. An Experimental Study of the Basis for Graph Drawing Algorithms. *The ACM Journal of Experimental Algorithmic*, Volume 2, Article 4, 1997.
13. Ware, C., Purchase, H.C., Colpoys, L. and McGill, M. Cognitive measurements of graph aesthetics. *Information Visualization*, 1 (2). 103-110.