

Autograph: Editeur pour les marches aléatoires sur les graphes*

M. MOSBAH P. BOGOSSIAN

LaBRI, Université Bordeaux-I

351, cours de la Libération, 33405 Talence, France.

mosbah@labri.u-bordeaux.fr

Abstract

Autograph is a software for simulating random walks on graphs. A random walk on a weighted graph is defined as a sequence of vertices visited by a particle that starts at a given vertex, and visits vertices using the following rule: if the particle is at a vertex, then it moves to a neighbor with a probability proportional to the weight of the traversed edge. Many interesting statistical parameters are difficult to compute, such as the cover time (the mean time to visit all vertices of the graph), or the probability for a vertex to be the last visited. Autograph is a tool for editing graphs, defining random walks, and computing numerical values of several statistical parameters, including the cover time. Several other options are provided by this software. It is written in Java and hence, can be used by just using through a web browser. It does not require any local install.

Résumé

Autograph est un outil pour simuler les marches aléatoires sur les graphes. Une marche aléatoire sur un graphe est une suite de sommets visités par une particule, placée initialement sur un sommet donné, et se déplace dans le graphe selon la règle suivante: si la particule se trouve sur un sommet, alors elle se déplace vers son voisin avec une probabilité proportionnelle au poids de l'arête traversée. Des paramètres statistiques liés aux marches aléatoires, comme le temps de parcours, sont difficiles à calculer. Autograph est un outil pour éditer des graphes, définir des marches aléatoires, et calculer plusieurs valeurs numériques de paramètres statistiques. D'autres fonctionnalités offertes par ce logiciel sont aussi disponibles. Le logiciel est écrit en Java, et peut donc être utilisé à l'aide d'un simple navigateur web. Aucune installation locale n'est nécessaire.

*This is supported by the EC TMR Network GETGRATS.

1 Présentation du logiciel

Autograph est un logiciel écrit en *Java 1.1.5* qui permet de simuler des marches aléatoire sur un graphe, et de fournir divers calculs statistiques. C'est un éditeur de graphes, il permet de saisir des graphes à la souris puis d'exécuter différents tests ou traitements sur les graphes ainsi obtenus. Il comporte un module qui permet d'éditer des graphes non-orientés, pondérés et étiquetés et qui propose de nouveaux traitements dédiés à l'étude des marches aléatoires. L'utilisation du logiciel peut se faire à partir d'un navigateur et ne nécessite pas d'installer l'application sur un disque local.

2 Fonctionnalités

Autograph offre les fonctionnalités suivantes:

- éditer des graphes non-orientés, pondérés et étiquetés. Un module spécifique pour les arbres est aussi disponible.
- Simuler des marches aléatoires sur le graphe dessiné. Un marche aléatoire est une particule qui se déplace d'un sommet à son voisin avec une probabilité proportionnelle au poids de l'arête traversée [2,3,4]. Avant de lancer une simulation, il faut sélectionner un unique sommet qui sera le sommet de départ de la (ou des) simulation(s). Trois types différents de simulations sont proposés:
 - visuelle: on lance un seul tour couvrant, la position courante de la particule est visualisée à l'écran (le sommet en question est colorié en gris foncé).
 - par lots: on lance n tours couvrants successifs (n choisi par l'utilisateur), tous issus du sommet de départ sélectionné, la position courante de la particule n'est pas visualisée à l'écran.
 - marche infinie: cette fois, on ne simule pas un tour couvrant, la particule se déplace un nombre de fois fixé par l'utilisateur, la position courante de la particule n'est pas visualisée à l'écran.

Des données sont collectées durant les simulations, remplaçant les données précédemment stockées en mémoire.

- Recherche des extrema

Lance un nombre donné (fixé par l'utilisateur) de tours couvrants pour chaque sommet (i.e. chaque sommet est alors le sommet de départ). Ce type de simulation permet de déterminer "expérimentalement" quels sont les sommets du graphe qui maximisent ou minimisent le temps de parcours moyen.

- Simulation Hitting
L'utilisateur sélectionne deux sommets du graphe qui seront le sommet de départ et le sommet de destination. On lance alors un nombre donné (fixé par l'utilisateur) de simulations de type "rencontre" (i.e. on part du sommet de départ et on s'arrête dès que le sommet de destination est visité).
- Calculer les arbres couvrants
On lance un nombre donné (fixé par l'utilisateur) de tours couvrants et on mémorise pour chaque tour couvrant l'arbre recouvrant généré. Quand toutes les simulations sont terminées, la liste des arbres trouvés est affichée.
- Voir arbres couvrants
Réaffiche la liste des arbres recouvrants trouvés lors de la dernière simulation de ce type.
- Resultats
Affiche les résultats mémorisés lors de la dernière simulation. Parmi les données disponibles, on peut citer: le temps de parcours moyen, l'attente moyenne de chaque sommet entre deux visites successives de la particule, le nombre moyen de parcours pour chaque sommet, la probabilité pour un sommet donné d'être le dernier sommet visité.
- Exporter résultats
Sauvegarde les résultats mémorisés lors de la dernière simulation dans un fichier texte.

Le calcul exact du temps de parcours (moyen) peut se faire à l'aide d'un automate[2]. Le logiciel peut construire l'automate, qui est souvent de taille très grande. Plusieurs opérations sur l'automate sont offertes. En particulier, sa construction, sa destruction, le nombre d'états, nombre de transitions, etc.

3 Architecture du logiciel

Le programme comporte une architecture en couches. En effet, comme nous avons implémenté en *Java*, l'ensemble du programme a été conçu dans une optique objet. Ainsi, en exploitant des propriétés d'héritage et d'encapsulation, l'organisation du logiciel est la suivante:

1. Niveau graphe

C'est la couche de plus bas niveau du programme. Il s'agit en fait du cœur de l'application. On y trouve l'ensemble des structures de données de bas niveau représentant la description mathématique du comportement et de l'organisation d'un graphe. Ainsi, y sont définis des objets tels que les

sommets, les arêtes, les graphes, et toutes les notions telles que le degré, l'incidence, l'adjacence...

2. Niveau "visuel"

C'est la couche où les objets mathématiques peuvent être dessinés à l'écran et manipulés à la souris. A ce niveau, on dispose déjà d'un éditeur de graphe, certes minimaliste, mais néanmoins opérationnel. Ici, on gère des graphes les plus génériques qu'ils soient: graphes orientés, sans étiquettes, non pondérés... Il est en effet très simple de rajouter par la suite des données supplémentaires telles que des étiquettes ou des poids, alors qu'ajouter "une notion d'orientation" impliquerait de profondes modifications.

On dispose également de toutes les fonctionnalités de base de l'éditeur: création du graphe, modification, duplication d'une partie sélectionnée, sauvegarde, restauration...

3. Niveau utilisateur

C'est à ce niveau que l'on programme les spécificités propres à un type de graphe et les algorithmes utiles pour résoudre un certain type de problème. Dans notre cas, c'est ici que nous avons défini les contraintes propres à notre graphe (non orienté, sans boucle...), que nous avons ajouté les propriétés supplémentaires (étiquettes, poids...) et que nous avons implémenté les algorithmes du problème (simulation, automate...).

4 Utilisation

Il faut tout d'abord savoir qu'il y a deux manières différentes d'exécuter **AutoGraph**. soit en tant qu'applet, soit en standalone.

Pour exécuter le programme en tant qu'applet, utilisez un navigateur web supportant *Java 1.1.5* (comme par exemple Netscape Navigator 3.x ou 4.0, HotJava 1.0 ou bien Microsoft Explorer 3.x) et accédez à l'une des deux URL suivantes:

<http://www.enserb.u-bordeaux.fr/~mosbah/AutoGraphe/>

ou

<http://dept-info.labri.u-bordeaux.fr/~mosbah/AutoGraphe/>

La figure 1 montre un exemple d'utilisation de ce logiciel.

References

- [1] P. Bogossian, S. Detruiseux. Autograph, Un éditeur de graphes en Java, Marches aléatoires sur des graphes. ENSERB, 1997.

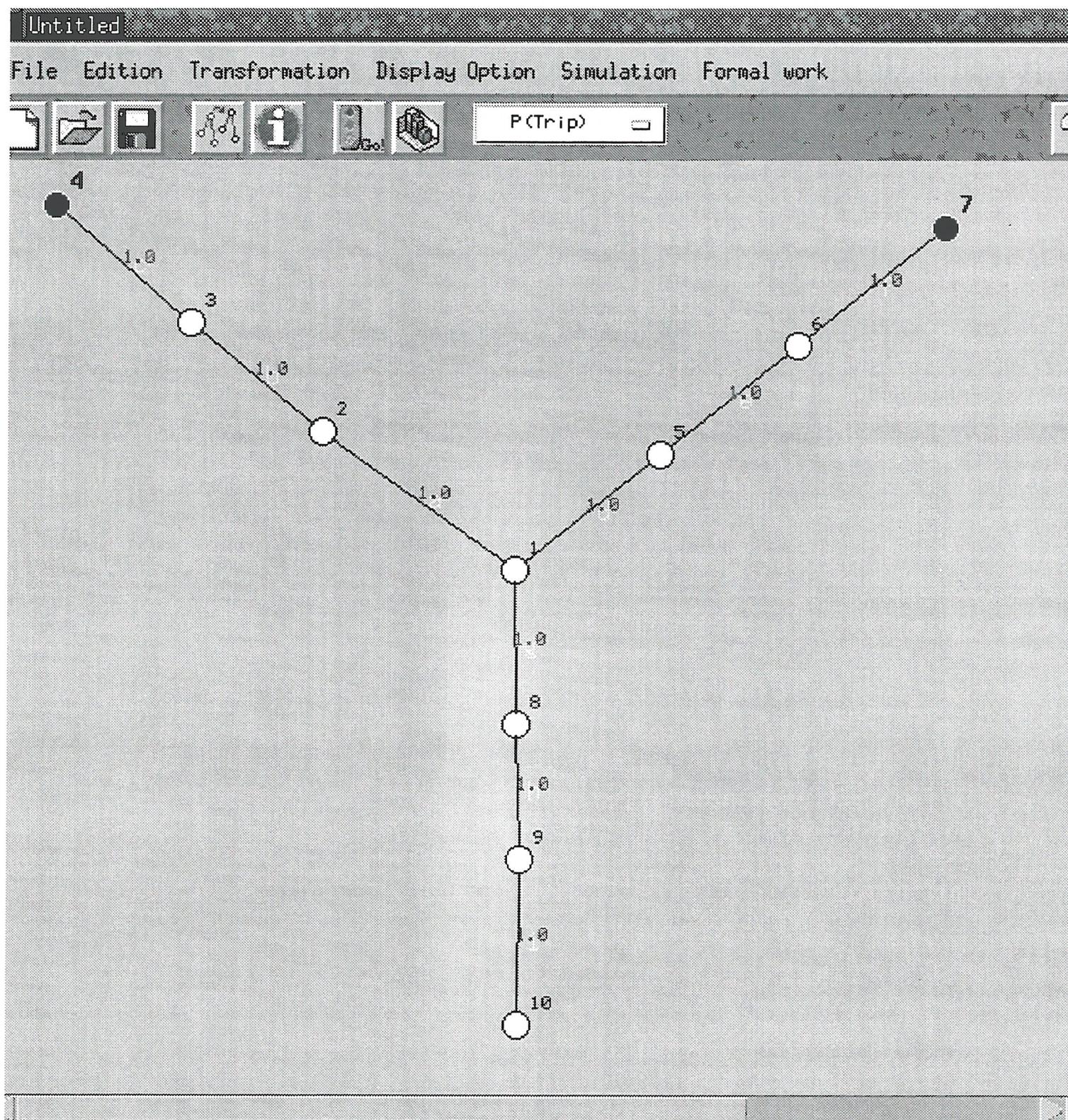


Figure 1: Exemple d'utilisation d'Autograph.

- [2] M. Mosbah and N. Saheb. A Syntactic Approach to Random Walks on Graphs. 23rd International Workshop, WG'97. Lect. Notes. Comput. Sci.1335. Pages 258-272.
- [3] M. Mosbah and N. Saheb. Formal rational fractions and random walks on cycle graphs. Présenté à FPSAC 97 - Vienne 14-18 juillet 97. A paraître dans *Discret. Math.*, 1999.
- [4] M. Mosbah and N. Saheb. Non uniform random spanning trees on weighted graphs. A paraître dans *Theoret. Comput. Science* 1999.