Résumé : Modélisation et solutions approchées d'un problème d'éclairage de graphe.

Mots clefs : graphe, arbre, complexité algorithmique, algorithmique géométrique, calcul propositionnel

► *Il est rappelé que le jury n'exige pas une compréhension exhaustive du texte. Vous êtes laissé(e) libre d'organiser votre discussion comme vous l'entendez. Des suggestions de développement, largement indépendantes les unes des autres, vous sont proposées en fin de texte. Vous n'êtes pas tenu(e) de les suivre. Il vous est conseillé de mettre en lumière vos connaissances à partir du fil conducteur constitué par le texte. Le jury appréciera que la discussion soit accompagnée d'exemples traités sur ordinateur.*

1. Problématique

Nous appellerons *ville* un certain nombre de *places* reliées par des *rues*. On se propose d'installer des lampadaires sur certaines places afin d'éclairer la ville. Chaque lampadaire éclaire toutes les rues qui partent de cette place. L'objectif est d'éclairer toutes les rues de la ville. Les opérateurs de téléphonie mobile, par exemple, sont confrontés à ce genre de problème lors de l'installation de leurs antennes relais, c'est également le cas lors du design de circuits électroniques etc.

La modélisation d'une ville par un graphe planaire non orienté non pondéré (place = sommet, rue = arête sans se soucier de la longueur des rues) s'impose naturellement. On considérera que tous les graphes à traiter sont *connexes*, en ce sens que l'on peut toujours se rendre d'une place à une autre en empruntant une ou plusieurs rues.

On appellera *éclairage* une solution au problème c'est-à-dire la donnée d'une liste de places à éclairer pour illuminer la ville tout entière. Un éclairage est alors dit *minimal* si le fait d'enlever un lampadaire plonge au moins une rue dans l'obscurité ; il est dit *optimal* s'il n'existe aucun éclairage nécessitant moins de lampadaires. Par soucis d'économie en lampadaires, on recherche bien sûr les éclairages optimaux (ou approchants).

Il n'est pas aisé de prédire le nombre minimal de lampadaires à utiliser pour un graphe donné connaissant son nombre de sommets et d'arêtes.

L'approche naïve consiste à tester toutes les configurations possibles (exponentiel en le nombre de sommets du graphe) pour retenir un éclairage optimal. L'éclairage systématique des sommets d'arité élevée n'est pas facile à mettre en œuvre également.

Il s'agit en effet d'un problème intrinsèquement exponentiel. Si l'on associe un booléen à chaque sommet (vrai = éclairé, faux = éteint), la contrainte correspondant à l'éclairage de la ville entière se traduit par la satisfiabilité minimale d'une conjonction de disjonctions de deux

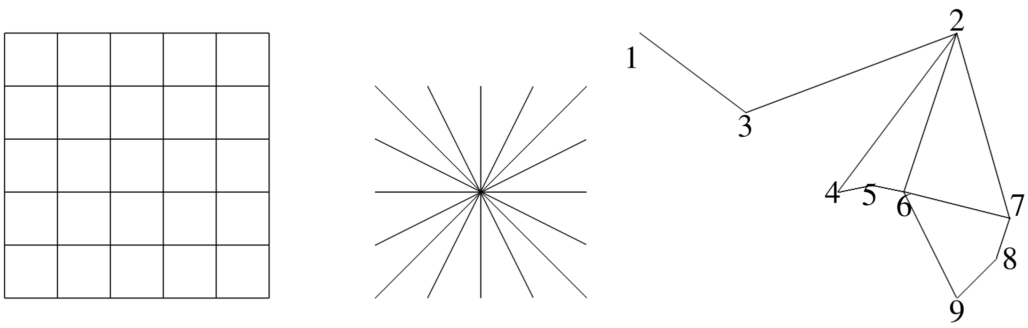

FIG. 1. trois exemples de ville

booléens (les deux extrémités de chaque arête). Par exemple, voici la proposition logique[1] correspondant au 3ème exemple de la figure 1 :

$$(S_1+S_3)(S_3+S_2)(S_2+S_4)(S_2+S_6)(S_2+S_7)(S_4+S_5)(S_5+S_6)(S_6+S_7)(S_6+S_9)(S_7+S_8)(S_8+S_9)$$

En développant cette dernière sous la forme d'une disjonction de conjonctions de booléens, la détermination d'un éclairage optimal correspond à la recherche d'une conjonction contenant un nombre minimal de booléens (on trouve par exemple ici $S_3S_2S_4S_6S_8$ ou $S_3S_4S_6S_7S_9$ ou ...). Ceci nécessite donc la lecture des 2^A conjonctions avec A le nombre d'arêtes du graphe.

Il va donc falloir renoncer à la détermination coûteuse d'un éclairage optimal ; nous allons présenter dans ce texte différentes approches conduisant plus rapidement à des solutions "convenables".

Mais tout d'abord :

Exercice de programmation :

► *Il vous est demandé de rédiger un programme conforme aux spécifications ci-dessous dans l'un des langages C, Caml ou Java à votre choix. Ce programme devra être accompagné d'un exemple d'exécution permettant d'en vérifier le bon fonctionnement. La clarté et la concision du programme seront des éléments importants d'appréciation pour le jury.*

Dans le langage de votre choix, implémenter un programme qui étant donnés un graphe et une proposition d'éclairage des lampadaires teste si celle-ci est correcte *i.e.* si toutes les rues sont bien éclairées. Le tester sur différents exemples bien choisis (on pourra justifier la/les structures de données utilisée).

[1] +=ou, ·=et

Présentons à présent différentes approches du problème.

2. Solution approchée par algorithme génétique

Les méthodes génétiques parodient les lois naturelles. Il s'agit d'algorithmes itératifs qui engendrent des générations d'individus (les solutions au problème) jusqu'à ce qu'un individu convenable soit créé. On table sur une forte masse d'individus et sur un grand nombre de générations. Il existe beaucoup de variantes dans ces méthodes, voici la plus basique[2] sur, disons, une population initiale de 100 individus.

Les individus constituant la génération $n+1$ sont obtenus de trois façons :

- par *sélection* : il s'agit des 48 meilleurs individus de la génération n,
- par *reproduction* : il s'agit des 48 "enfants" des 48 individus précédents,
- par *mutation* : 4 individus sont aléatoirement transformés à partir des meilleurs issus de la génération n.

L'effectif de la population reste ainsi constant d'une génération à l'autre et la «qualité» du meilleur individu d'une génération ne peut que s'améliorer (largement) au fil du temps.

La mise en œuvre de cette méthode suppose que l'on est capable :

- de créer une population initiale de solutions (il suffit, pour notre problème, les A rues étant représentées par A couples de places, de créer une liste de A booléens : vrai (resp. faux) = la première (resp. la seconde) place du couple est éclairée – si les deux places sont éclairées on choisit indifféremment vrai ou faux –),
- de disposer d'une fonction d'évaluation d'une solution (ici on parcourt une liste de booléens pour déterminer à combien de lampadaires elle correspond),
- de réaliser effectivement les trois procédés précédents. Ici, on peut singer le *crossing over* génétique pour faire se reproduire deux listes de booléens (on les coupe en deux et on les recolle en croix). On espère que les enfants héritent ainsi de certaines des qualités de leurs parents !

Vous noterez qu'il n'y a pas unicité dans la représentation d'un éclairage solution par une liste de booléens[3]...

3. Solution approchée par utilisation de quadtree

Les problèmes d'informatique géométrique font souvent appel à des sectorisation de l'espace : nous allons envisager ici le recours à des arbres quaternaires ou *quadtrees*.

On considère le graphe à n sommets ($n > 1$) comme inclus dans un carré de côté 2^p ; on décompose alors récursivement chaque carré en 4 blocs carrés de taille $2^{p-1} \times 2^{p-1}$ (Nord-Ouest, Nord-Est, Sud-Ouest, Sud-Est) puis l'on recommence sur chacun des carrés précédents qui

[2] effectivement employée par un opérateur de téléphonie mobile

[3] et que sélectionner les 48 meilleurs individus parmi 100 peut se faire par une méthode algorithmiquement moins coûteuse qu'un tri...

contient strictement plus d'un sommet. On itére cette décomposition tant qu'un carré contient strictement plus d'un sommet.

L'arbre quaternaire associé est alors constitué :

- de nœuds internes qui correspondent aux étapes précédentes de la décomposition ; l'arité de chacun de ces nœuds est 4 (avec par convention l'ordre NO NE SO SE pour les fils),
- et de feuilles désignant soit un carré vide (V) soit un sommet du graphe.

La figure 2 montre la sectorisation du plan obtenu pour l'exemple 3 et son quadtree.

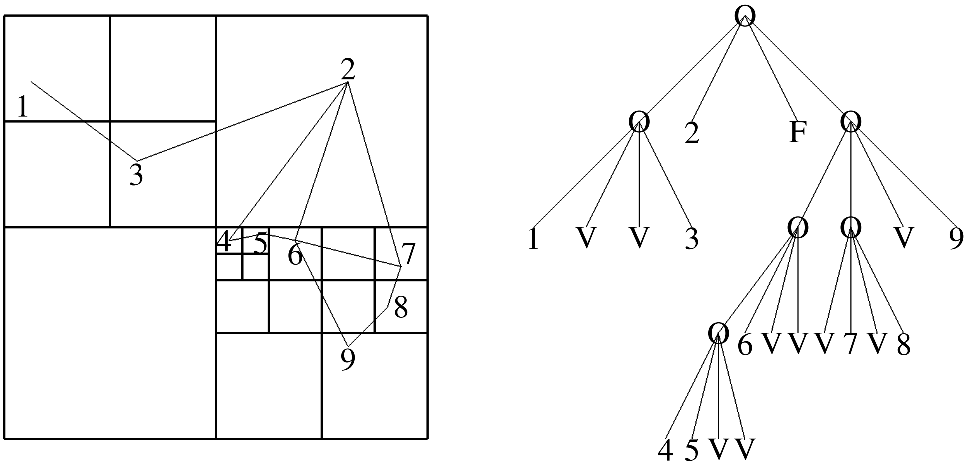

FIG. 2. Régionnement d'une ville et son quadtree

Vous noterez que chaque bloc comporte au maximum huit blocs adjacents (par un côté ou un angle) de taille égale ou supérieure au bloc considéré. La détermination des voisins d'un bloc utilise des fonctions récursives de parcours dans le quadtree.

L'obtention d'un éclairage se fait alors comme suit :

- initialement, on considère que tous les sommets sont éclairés,
- puis en remontant dans le quadtree (parcours bottom up) on considère les sommets appartenant aux blocs voisins d'un bloc B et les arêtes reliant ces sommets aux sommets de B (et seulement celles ci) : on éteint alors les sommets inutilement éclairés. On obtient ainsi un bloc de taille 4 fois supérieure convenablement éclairé.
- À chaque étape on dispose ainsi d'un éclairage convenable de toute la ville ; l'ascension dans le quadtree faisant diminuer le nombre de places éclairées en conservant le caractère convenable de l'éclairage.
- Arrivé à la racine on dispose d'une solution où seules les arêtes reliant des sommets issus de blocs non voisins peuvent éventuellement conduire à des éclairages de places superflus.

Pour une profondeur p fixée du quadtree, le nombre d'arêtes à considérer est borné : ceci induit une complexité satisfaisante pour cette méthode.

4. Solution approchée par recuit simulé

Cette méthode ne sera qu'abordée ici ; elle rappelle d'ailleurs par certains points l'approche génétique.

La recherche d'extrema locaux se fait souvent de façon itérative par des méthodes de plus grande pente (gradient) etc. La méthode du *recuit simulé* est issue du monde de la chimie : on cherche à fabriquer un polymère possédant de bonnes propriétés à partir de plusieurs composants. Si le mélange obtenu à la suite de la cuisson de ces derniers n'est pas satisfaisant, on le réchauffe en modifiant légèrement les proportions respectives des composants puis on laisse à nouveau refroidir et l'on teste le nouveau mélange figé. Tant que les augures de la chimie le permettent, on recommence l'opération.

La traduction de cette approche consiste ici à partir du graphe totalement éclairé puis à enlever des lampadaires de façon heuristique pour conserver un éclairage jusqu'à tomber sur un éclairage minimal. Si le nombre de lampadaires obtenu n'est pas convenable, on rajoute aléatoirement quelques lampadaires pour rechercher à nouveau un éclairage minimal etc.

Suggestions pour le développement

► *Soulignons qu'il s'agit d'un menu à la carte et que vous pouvez choisir d'étudier certains points, pas tous, pas nécessairement dans l'ordre, et de façon plus ou moins fouillée. Vous pouvez aussi vous poser d'autres questions que celles indiquées plus bas. Il est très vivement souhaité que vos investigations comportent une partie traitée sur ordinateur et, si possible, des représentations graphiques de vos résultats.*

– Nous invitons le lecteur à reprendre en détail une ou plusieurs des approches précédentes ; à se pencher plus avant sur l'étude de leur complexité, leur implémentation, leur adaptation à une approche dynamique du problème (que se passe-t-il si l'on rajoute ou enlève des rues, places ? etc) aux contraintes réelles (portée des lampadaires par exemple).
– Sans trop s'éloigner du texte, on pourra également rapprocher ce problème d'autres problèmes d'algorithmique géométrique (graphe, convexité etc) ou bien s'en servir pour illustrer différents aspects de la programmation (diviser pour régner, structure de données, problèmes NP complets etc).