Mini-projet novembre 2013

Pour cela, vous utiliserez R. Vous publierez vos observations sur `http://rpubs.com` à l'aide de Rstudio et enverrez l'url à `arnaud.legrand@imag.fr` avant le 23 décembre à minuit en indiquant la chaîne [RICM4-PS :miniprojet] dans le sujet du message.

Afin de s'assurer que nous puissions reproduire vos résultats, vous prendrez soin de fixer la valeur de la graine du générateur aléatoire à l'aide de la fonction `set.seed`.

Sujet 1 : Complexité algorithmique

Pour anonymiser des routages, des algorithmes construisent des arbres aléatoires sur un ensemble de n nœuds. Pour être difficiles à "casser", l'idée est de générer des arbres aléatoires uniformément parmi tous les arbres possibles couvrant les n nœuds.

L'objectif de ce sujet est d'évaluer la complexité du routage sur une telle construction aléatoire.

Question 1.1 : Algorithme de génération

Écrire un algorithme de génération uniforme d'arbre de taille n. On démontrera que l'algorithme génère bien un arbre selon la loi uniforme.

Question 1.2 : Diamètre de l'arbre

Étudier en fonction de n la distribution de probabilité associée au diamètre (plus long chemin élémentaire d'un arbre de taille n généré uniformément. On prendra $n = 10, 100, 1000$ et on justifiera la taille des échantillons générés pour estimer cette distribution.

Question 1.3 : Diamètre moyen d'un arbre uniformément généré

Tracer en fonction de n le diamètre moyen d'un arbre uniformément généré ainsi que l'intervalle de confiance associé. Qu'en concluez-vous ? Peut-on comparer ce résultat à celui de la hauteur moyenne d'un arbre binaire généré uniformément ?

Sujet 2 : Réseau de capteurs

L'objectif de ce sujet est d'évaluer la couverture d'un espace sur lequel on a déposé des capteurs de manière aléatoire. Pour simplifier l'approche on considère que l'espace est représenté par un carré de coté 1 et que les n capteurs sont répartis selon une loi uniforme sur ce carré. Les capteurs ont une portée de communication θ, c'est à dire que 2 capteurs à une portée inférieure à θ peuvent échanger des informations.

Question 2.1 : Algorithme de génération

Écrire un algorithme qui génère les n points dans le CARRÉ. Calculer la matrice des distances entre les différents capteurs et par suite le graphe de connexion.

Question 2.2 : Connectivité

Étudier en fonction de n et de θ la probabilité $p_c(n, \theta)$ que l'ensemble des capteurs soient connectés (il existe un chemin entre 2 capteurs quelconques). Pour cela tracer $p_c(n, \theta)$ en fonction de θ pour $n = 10, 20, 100$, on estimera l'erreur à l'aide d'intervalles de confiance.

Question 2.3 : Portée critique

Montrer que le phénomène est critique, c'est à dire que $p_c(n, \theta)$ varie brutalement à partir d'une certaine valeur $\theta_c(n)$ que l'on estimera graphiquement.

Question 2.4 : Couverture

Calculer, pour n et θ fixé la probabilité qu'un point à distance r du centre soit couvert. Par simulation calculer le taux de couverture du carré, c'est à dire la proportion de surface moyenne couverte par le réseau de capteurs.

Sujet 3 : La puce dans le labyrinthe

Un labyrinthe de taille $m \times n$ est constitué par des obstacles placés aléatoirement sur une grille, on note p la probabilité d'avoir un obstacle en un point de la grille. Une puce part du point A (en bas à gauche de la grille) pour rejoindre le point B (en haut à droite de la grille), on supposera qu'il existe un chemin de A à B. Son déplacement est aléatoire, c'est à dire qu'elle choisit uniformément la case suivante parmi ses voisines.

L'objectif du travail demandé est l'étude du temps mis par la puce pour atteindre B en partant de A

Question 3.1 : Algorithme de génération de labyrinthe

Écrire un algorithme qui génère, en fonction de p, un terrain acceptable, c'est-à-dire tel qu'il existe un chemin entre A et B.

Question 3.2 : Déplacement de la puce

Écrire un algorithme de déplacement de la puce dans le labyrinthe.

Question 3.3 : Étude expérimentale (référence)

Pour $p = 0$ et plusieurs valeurs de (m, n), estimer le temps moyen d'atteinte de B partant de A.

Question 3.4 : Cas particulier $m = n = 10$

Étudier la distribution du temps d'atteinte de B partant de A pour $p = 0$ et $m = n = 10$.

Question 3.5 : Étude de l'impact de p

Évaluer l'impact de p sur le temps moyen d'atteinte de B partant de A.

Sujet 3 : circuit élastique

Une application distribuée est constituée d'un ensemble de n sites totalement connectés, le réseau est supposé fiable. On note par $d_{i,j} = d_{j,i}$ la durée d'envoi d'un message du site i au site j. Pour implanter un algorithme distribué d'exclusion mutuelle on utilise sur un jeton qui parcourt tous les sites. Le "Round Trip Time" (RTT) est la durée du tour du jeton par tous les sites. L'objectif est de construire le tour du jeton de durée minimale pour une matrice de durée D donnée.

Une configuration C du système consiste en un circuit passant une et une seule fois par chacun des sites. L'opération d'échange consiste à modifier le circuit en échangeant 2 sites. Par exemple si

$$C = 1 \to 2 \to 3 \to 4 \to 5 \to 6 \to 7 \to 1,$$

l'échange de 2 et de 5 donne le circuit

$$C' = 1 \to 5 \to 4 \to 3 \to 2 \to 6 \to 7 \to 1.$$

Un premier algorithme consiste à partir d'un circuit initial et de l'améliorer par échanges successifs jusqu'à ce que cela ne soit plus possible. On peut montrer que cet algorithme donne une approximation mais ne donne pas forcemment de valeur optimale. On améliore cet algorithme en utilisant un tirage aléatoire permettant de faire des échanges même si on ne gagne pas sur le RTT.

```
Optimise()
  C ← genere_circuit()
  T ← valeur − initiale
  repeat
  |  i ← random_site
  |  j ← random_site
  |  C' ← echange(C, i, j)
  |  Δ = RTT(C') − RTT(C)
  |  if Random < exp(−Δ/T)
  |  |  C ← C'
  |  Actualiser T
  until Condition sur T ou C
  return C
```

Algorithm 1: Algorithme de recuit simulé

Le paramètre T est appellé paramètre de température et décroit au fur et à mesure de l'exécution de l'algorithme.

Question 4.1 : Explications

Que se passe-t-il pour $T = +\infty$, pour $T = 0$? Donner une intuition expliquant que T doit décroitre "pas trop vite" ni "trop lentement".

Question 4.2 : Algorithme de génération

Écrire un algorithme qui génère une matrice symétrique D dont les $d_{i,j}$, pour $i < j$ sont indépendants uniformément répartis sur $[0, 1[$, avec $d_{i,i} = 0$. Écrire un algorithme de génération uniforme d'un circuit. Implanter l'algorithme de recuit simulé.

Question 4.3 : Etude expérimentale

Pour $n = 200$ proposer plusieurs schémas de décroissance de température et les comparer sur une même matrice et circuit initial.

Question 4.4 : Loi du RTT "optimal"

Etudier, pour $n = 200$ la loi empirique du RTT optimal.