Modéliser l'Aléa - TP2 : Maximisation du PageRank.

Tong ZHAO & Yonatan DELORO

Pour le 11 juin 2017

PS : Pour des raisons pratiques, on a séparé le code des questions 1 à 7 et de la question 12 dans le fichier *"q_1_a_q7_et_q12"* du code des questions 8 à 11 propres à l'ergodicité dans le fichier *"ergodicite_q8_a_q11.sce"*.

1 Une chaine de Markov

Question 1

Rappelons que la matrice de transition de notre chaine (X_n) s'écrit :

$$P = \alpha P_1 + (1-\alpha)ez$$

où P_1 est obtenu en divisant chaque ligne de la matrice d'adjacence du graphe des pages par le degré du noeud de départ indicé par la ligne et en remplacant les lignes identiquement nulles par z vecteur ligne aléatoire à valeurs strictement positives. α est un coefficient strictement positif et e est le vecteur colonne avec que des 1.

- Une chaine de Markov de matrice P_1 permet d'aller de la page i à la page j avec une probabilité égale à l'inverse du nombre de noeuds partant de i, sauf si i pointe sur aucune page auquel cas elle permet d'aller de i vers j avec une probabilité $z_j > 0$.
- Aussi on ajoute pour notre chaîne X la possibilité de se téléporter d'une page à une autre (sans suivre des liens) avec une probabilité $1-\alpha$ (les probabilités de redirection d'une page vers une autre données par P_1 sont donc aussi multipliées par α). En effet, z permet d'atteindre depuis une page quelconque i toute page j avec une probabilité $(1-\alpha)z_j$. C'est pourquoi z est appelé **vecteur de téléportation.** Autrement dit, z rajoute des "shortcuts" entre toute paire de pages dans le graphe de départ correspondant à la matrice de transition P_1.

$(X_n)_{n\in\mathbb{N}}$ est une chaine de Markov **homogène** sur un espace d'états **fini**. Aussi elle est **irréductible** puisque tous les états communiquent : $1-\alpha > 0$ et z a ses composantes toutes strictement positives.

Par conséquent, (X_n) est une chaine **récurrente positive**.

2 Calcul du PageRank des états de la chaîne

Question 2

Puisque (X_n) est une chaine récurrente positive, le théorème ergodique nous dit qu'elle admet une **unique mesure de probabilité π invariante** par la matrice de transition P, qui correspond aux inverses des espérances des temps de retours en les différents états. Cette mesure invariante correspond au **PageRank** des états de la chaîne.

En figure 1, on représente la matrice d'adjacence Adj simulée pour $n = 10$ pages sous la forme d'un graphe. On construit également la matrice de transition P de la chaine de Markov (X_n) associée à cette matrice d'adjacence pour un vecteur de téléportation z choisi aléatoirement et une probabilité de se téléporter $\alpha = 0.85$. On donne les valeurs de Adj, z et de la matrice de transition P associée en figure 2.

Question 3

Même si la matrice de transition de la chaine P es pleine, on peut quand même profiter de la structure creuse de la matrice stochastique Pss sous-jacente (matrice d'adjacence Adj dont chaque ligne est divisé par le degré du noeud indicé par elle) pour calculer $P^T x$ où x est un vecteur de probabilités d'états.

En effet, on peut écrire par "définition" de P puis par "construction" de P_1 :

$$\begin{aligned} P^T x &= (\alpha P_1^T + (1-\alpha) z^T e^T) x \\ &= (\alpha (Pss^T + z^T d^T) + (1-\alpha) z^T e^T) x \\ &= \alpha Pss^T x + \alpha z^T (d^T - e^T) x + z^T e^T x \end{aligned}$$

où d est le vecteur colonne booléen (0 ou 1) qui indique si la somme des termes d'une ligne de Adj est nulle.

Ce calcul qui préserve le caractère creux des opérations est implémenté dans le code. On vérifie que l'on retrouve bien le même résultat qu'en effectuant le calcul direct de $P^T x$: on obtient en effet des différences terme à terme entre les deux vecteurs de l'ordre de 10^{-16}.

Question 4

Le PageRank π correspond à l'unique mesure invariante par la chaîne (cf. question 2). Ainsi π est l'unique loi de probabilité vérifiant $\pi = \pi P$, soit le vecteur propre sommant à 1 associé à la valeur propre 1 de P^T.

Grâce à la fonction *spec* de Scicoslab, on obtient le PageRank donné en figure 3. Ainsi, on peut redessiner en figure 4 le graphe associé à la matrice d'adjacence des pages en représentant cette fois chaque page par un cercle de diamètre proportionnel à son PageRank. On vérifie à vue qu'une page a un PageRank d'autant plus grand que la somme des PageRank des pages qui pointent vers elle est grande.

Question 5

On calcule maintenant le PageRank π comme point fixe de l'application "image par P^T" et ainsi comme limite de la suite $p_{k+1} = P^T p_k$.

On obtient bien convergence, puisqu'au bout d'un certain nombre d'itérations N de l'application P la différence entre p_N et $P^T p_N$ est inférieure à $\epsilon = 2^{-52}$.

Pour vérification, on donne le vecteur $\pi = p_N$ obtenu à ce niveau de tolérance en figure 5. On obtient bien le même PageRank que par la méthode du spectre.

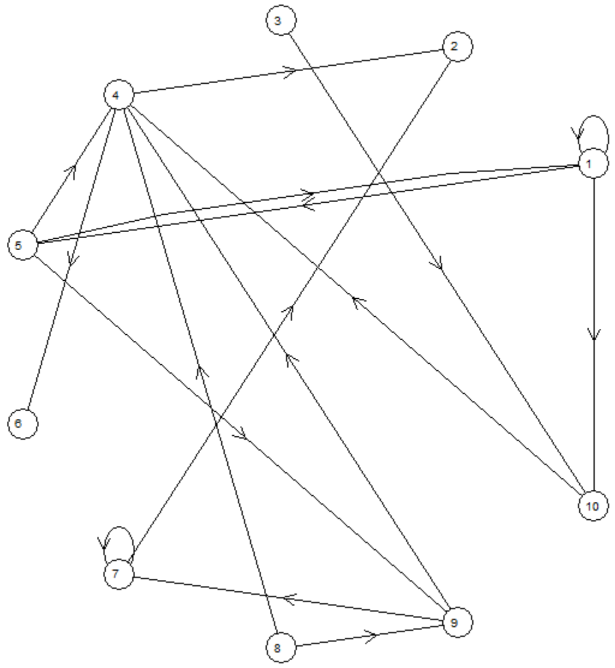

FIGURE 1 – Graphe associé à la matrice d'adjacence Adj des pages du graphe.

```
-->Adj
 Adj  =

    1.    0.    0.    0.    1.    0.    0.    0.    0.    1.
    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
    0.    0.    0.    0.    0.    0.    0.    0.    0.    1.
    0.    1.    0.    0.    0.    1.    0.    0.    0.    0.
    1.    0.    0.    1.    0.    0.    0.    0.    1.    0.
    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
    0.    1.    0.    0.    0.    0.    1.    0.    0.    0.
    0.    0.    0.    1.    0.    0.    0.    0.    1.    0.
    0.    0.    0.    1.    0.    0.    1.    0.    0.    0.
    0.    0.    0.    1.    0.    0.    0.    0.    0.    0.

-->z
 z  =

        column 1 to 6

    0.1446714    0.0733759    0.0769743    0.1277669    0.0349647    0.1566150

        column  7 to 10

    0.1297031    0.0510626    0.1125515    0.0923147

-->P
 P  =

        column 1 to 6

    0.3050340    0.0110064    0.0115461    0.0191650    0.2885780    0.0234922
    0.1446714    0.0733759    0.0769743    0.1277669    0.0349647    0.1566150
    0.0217007    0.0110064    0.0115461    0.0191650    0.0052447    0.0234922
    0.0217007    0.4360064    0.0115461    0.0191650    0.0052447    0.4484922
    0.3050340    0.0110064    0.0115461    0.3024984    0.0052447    0.0234922
    0.1446714    0.0733759    0.0769743    0.1277669    0.0349647    0.1566150
    0.0217007    0.4360064    0.0115461    0.0191650    0.0052447    0.0234922
    0.0217007    0.0110064    0.0115461    0.4441650    0.0052447    0.0234922
    0.0217007    0.0110064    0.0115461    0.4441650    0.0052447    0.0234922
    0.0217007    0.0110064    0.0115461    0.8691650    0.0052447    0.0234922

        column  7 to 10

    0.0194555    0.0076594    0.0168827    0.2971805
    0.1297031    0.0510626    0.1125515    0.0923147
    0.0194555    0.0076594    0.0168827    0.8638472
    0.0194555    0.0076594    0.0168827    0.0138472
    0.0194555    0.0076594    0.3002161    0.0138472
    0.1297031    0.0510626    0.1125515    0.0923147
    0.4444555    0.0076594    0.0168827    0.0138472
    0.0194555    0.0076594    0.4418827    0.0138472
    0.4444555    0.0076594    0.0168827    0.0138472
    0.0194555    0.0076594    0.0168827    0.0138472
```

FIGURE 2 – Matrice d'adjacence Adj des pages du graphes, vecteur de téléportation z, matrice de transition P de la chaine construite sur la base de Adj et de z.

```
 The value of pi calculated with spectrum of P
-->disp(pi);

        column 1 to 6
   0.1010135    0.1697080    0.0320111    0.1834010    0.0431611    0.1430764
        column  7 to 10
   0.1441128    0.0212352    0.0680604    0.0942205
```

FIGURE 3 – PageRank calculé par étude du spectre de P.

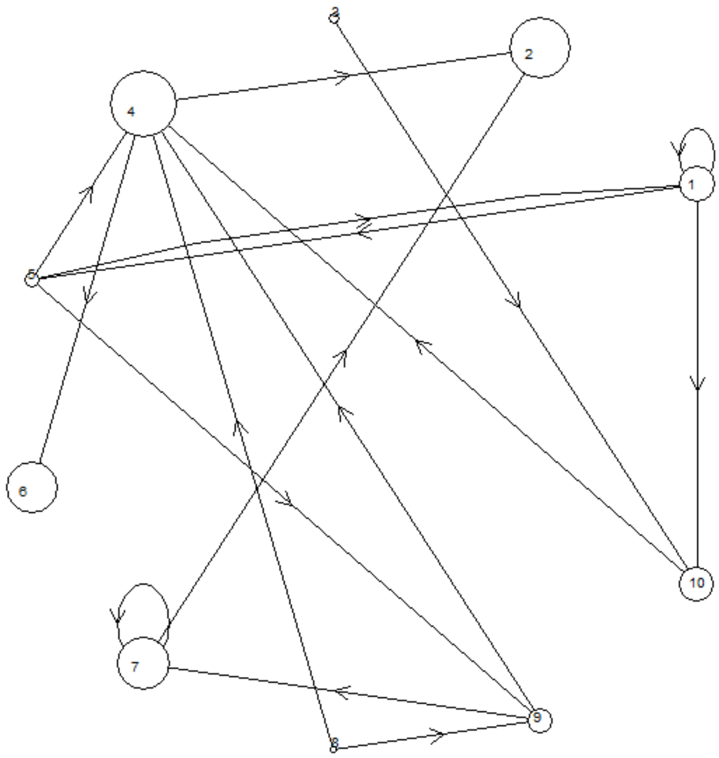

FIGURE 4 – Graphe associé à la matrice d'adjacence Adj des pages du graphe, où le diamètre de chaque page est proportionnel à son PageRank.

```
 The value of pi calculated by convergence of sequence
-->disp(pi);

        column 1 to 6
   0.1010135    0.1697080    0.0320111    0.1834010    0.0431611    0.1430764
        column  7 to 10
   0.1441128    0.0212352    0.0680604    0.0942205
```

FIGURE 5 – PageRank calculé par itérations successives de l'application P^T dont il est point fixe.

```
 The value of pi calculated by sparse iteration
-->disp(pi);

        column 1 to 6
   0.1010135    0.1697080    0.0320111    0.1834010    0.0431611    0.1430764
        column  7 to 10
   0.1441128    0.0212352    0.0680604    0.0942205
```

FIGURE 6 – PageRank calculé comme point fixe de l'application P^T, en utilisant le caractère creux de Pss pour calculer $P^T x$.

Question 6

Le but de cette question est d'obtenir le PageRank π par la même méthode itérative que celle employée en question 5, mais en calculant l'image de x par P^T en utilisant le caractère creux de Pss comme explicité en question 3.

Fort heureusement, on obtient le même vecteur de PageRank qu'en calculant $P^T x$ en "force brute" (voir figure 6).

3 Maximisation discrète du PageRank

Question 7

On cherche maintenant à résoudre le problème d'optimisation du PageRank qui consiste à maximiser la somme des PageRank des m-premières pages en modifiant le liens des p-premières pages ($p \leq m$) vers les pages m+1,...,n, soit en modifiant la sous-matrice d'adjacence $Adj(1:p, m+1:n)$.

Algorithme On va ainsi écrire un programme d'optimisation discret pour résoudre ce problème. Une manière naïve de procéder consiste à considérer chaque graphe d'adjacence possible et à conserver celui dont le PageRank des m-premières pages est le plus grand. On sait qu'il y a $2^{p-(n-m)}$ graphes possibles, puisque l'on décide de la présence ou de l'absence d'un lien entre chacune des p-premières pages et chacune des pages m+1,...,n ; ainsi une manière simple en terme d'implémentation pour créer chaque sous matrice d'adjacence possible $Adj(1:p, m+1:n)$ consiste à lui faire correspondre l'écriture binaire d'un entier compris entre 1 et $2^{p-(n-m)}$.

Résultats Si l'on souhaite résoudre le problème avec $p = 2$ et $m = n/2$ c'est-à-dire maximiser le PageRank des 5 premières pages en modifiant les liens des 2 premières vers les 5 autres pages, alors on obtient comme meilleur choix des liens la matrice d'adjacence donnée en figure 7, le PageRank donné en figure 8, et le graphe associé à la nouvelle matrice d'adjacence donné en figure 9.

En particulier, on note 2 changements de liens pour maximiser le PageRank des 5 premières pages. Premièrement, on observe que la page 1 pointe maintenant vers la page 6 et non plus vers la page 10. Aussi la page 2 pointe vers la page 5 alors qu'elle ne pointait sur personne. Ainsi la PageRank de la page 10 diminue considérablement, la page 6 voit certes son PageRank augmenter mais ne pointant sur personne elle permet de se téléporter avec un peu moins d'une chance sur 2 sur une des pages 1 à 5 (voir z) ce qui n'est finalement pas si mauvais. Enfin, le choix pour la page 2 de pointer sur 5 semble a priori un bon choix puisque 5 pointe sur 2 des 5 premières pages (pages 1 et 4) contre une seule des 5 dernières pages (page 9) qui de surcroit permet de se rendre rapidement et exclusivement à la page 2 via 7.

Par conséquent, la somme des PageRank des 5 premières pages est après optimisation de 0.60 contre 0.53 dans le graphe initial. Cela tient essentiellement à l'augmentation du PageRank de la page 5 (et la diminution du PageRank de la page 10) comme on l'observe en regardant l'évolution de π du graphe initial et du graphe optimisé.

```
-->[Adj_optim, cost]=pagerank(Adj, z,alpha,m, p, n);
-->[P_optim,Pss_optim,Pprim_optim,d_optim]=google2(Adj_optim, z, alpha);

-->pi_optim=pi_iterative_sparse(n, Pss_optim, d_optim, z, alpha);

-->Adj
 Adj  =

    1.    0.    0.    0.    1.    0.    0.    0.    0.    1.
    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
    0.    0.    0.    0.    0.    0.    0.    0.    0.    1.
    0.    1.    0.    0.    0.    1.    0.    0.    0.    0.
    1.    0.    0.    1.    0.    0.    0.    0.    1.    0.
    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
    0.    1.    0.    0.    0.    0.    1.    0.    0.    0.
    0.    0.    0.    1.    0.    0.    0.    0.    1.    0.
    0.    0.    0.    1.    0.    0.    1.    0.    0.    0.
    0.    0.    0.    1.    0.    0.    0.    0.    0.    0.

-->sum(pi(1:5))
 ans  =

    0.5292947

-->Adj_optim
 Adj_optim  =

    1.    0.    0.    0.    1.    1.    0.    0.    0.    0.
    0.    0.    0.    0.    1.    0.    0.    0.    0.    0.
    0.    0.    0.    0.    0.    0.    0.    0.    0.    1.
    0.    1.    0.    0.    0.    1.    0.    0.    0.    0.
    1.    0.    0.    1.    0.    0.    0.    0.    1.    0.
    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
    0.    1.    0.    0.    0.    0.    1.    0.    0.    0.
    0.    0.    0.    1.    0.    0.    0.    0.    1.    0.
    0.    0.    0.    1.    0.    0.    1.    0.    0.    0.
    0.    0.    0.    1.    0.    0.    0.    0.    0.    0.

-->sum(pi_optim(1:5))
 ans  =

    0.5963489
```

Figure 7 – Matrices d'adjacences initiale et optimisée. Sommes des PageRank des 5 premières pages dans le graphe initial (sum(pi(1 :n)) et dans le graphe optimisé.

```
-->pi
 pi  =

         column 1 to 6

    0.1010135    0.1697080    0.0320111    0.1834010    0.0431611    0.1430764

         column  7 to 10

    0.1441128    0.0212352    0.0680604    0.0942205

-->pi_optim
 pi_optim  =

         column 1 to 6

    0.1184312    0.1386520    0.0209027    0.1574585    0.1609045    0.1430048

         column  7 to 10

    0.1218977    0.0138663    0.0820466    0.0428357
```

Figure 8 – PageRank du graphe initial et du graphe optimisé.

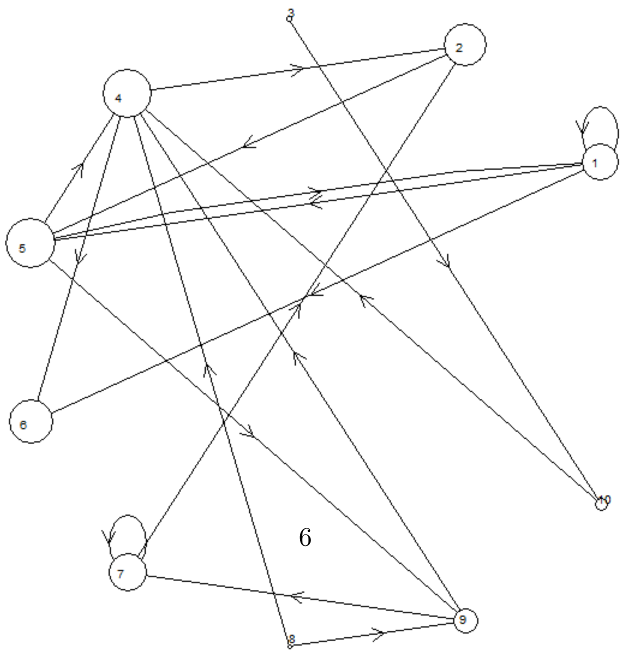

Figure 9 – Graphe associée à la matrice d'adjacence optimisée.

```
-->P
 P  =

    0.0729956    0.3508961    0.3442691    0.2318391
    0.2537432    0.2806736    0.1390090    0.3265742
    0.1972367    0.3816980    0.1452025    0.2758629
    0.0686794    0.2965317    0.2788189    0.3559699

-->T=100000; CT=ergodique_markov_T(T,P);

-->CT
 CT  =

    8.33555

-->[c,pi]=ergodique_markov(P);

-->c
 c  =

    8.3301531

-->c-CT
 ans  =

  - 0.0053969
```

FIGURE 10 – Une trajectoire pour la chaine de matrice de transition P. Calcul du cout obtenu sur cette trajectoire et comparaison avec le coût ergodique.

4 Problèmes ergodiques

Question 8

Dans cette question, on considère une matrice de transition P aléatoire irréductible (et chaque ligne étant toujours normalisée à 1 pour avoir des lois de probabilité partant de chacune des pages) On se donne une fonction r qui attribue un coût $r(x) = x^2$ au passage sur un état x donné.

On cherche à vérifier le théorème ergodique, à savoir que le coût moyen le long d'une trajectoire sur les pages du graphes suivant la matrice P tend vers $\langle \pi, r \rangle$, où π est toujours l'unique mesure invariante associée à P irréductible.

Ainsi, pour la chaîne de matrice de transition P donnée en figure 10, on obtient pour une trajectoire arrêtée à la $T = 100.000$ itération :

$$\frac{1}{T}\sum_{t=0}^{T-1} r(X_t) \simeq 8.336$$

.

Par ailleurs, en calculant la mesure invariante π par convergence vers un point fixe de P, on obtient :

$$\sum_x \pi(x) r(x) \simeq 8.330$$

Ainsi, la petite différence obtenue entre les valeurs, de l'ordre de 5.10^{-3}, est en accord avec le théorème ergodique.

Aussi, il faudrait répéter ceci sur un grand nombre de trajectoires, ou se donner un plus grand nombre d'itérations T, pour mieux se convaincre de la convergence de $\frac{1}{T}\mathbb{E}\left[\sum_{t=0}^{T-1} r(X_t)\right]$ vers $\sum_x \pi(x) r(x)$, T augmentant.

Question 9

Dans cette question, on vérifie numériquement que l'on peut également obtenir $c = \sum_x \pi(x) r(x)$ en résolvant le système linéaire suivant :

$$\left\{ \begin{array}{rcl} (P-I)w+R-c_1 & = & 0 \\ (P-I)c_1 & = & 0 \end{array} \right. \Leftrightarrow \left(\begin{array}{cc} P-I & -I \\ 0 & P-I \end{array} \right) \left(\begin{array}{c} w \\ c_1 \end{array} \right) + \left(\begin{array}{c} R \\ 0 \end{array} \right) = 0$$

où R est le vecteur des coûts en tous les états, et où (w, c_1) est l'inconnue. Alors $c_1^{\#}$ solution est en effet un vecteur constant dont les composantes valent c.

Posons

$$\left\{ \begin{array}{rcl} w^{\#} & = & [(Pr-(P-I))^{-1} - Pr]R \\ c_1^{\#} & = & PrR \end{array} \right.$$

où Pr est le projecteur spectral sur $E_1(P)$ l'espace propre de P associé à la valeur propre 1 : $Pr = (\pi\ \pi\ ...\ \pi)$.

Alors, par définition du projeté sur $E_1(P)$, on a bien sûr :

$$(P-I)c_1^{\#} = (P-I)PrR = 0$$

D'autre part,

$$\begin{aligned} (P-I)w^{\#} + R - c_1^{\#} &= (P-I)(Pr-(P-I))^{-1}R - (P-I)PrR + R - PrR \\ &= (P-I)(Pr-(P-I))^{-1}R - 0 + R - PrR \\ &= -(Pr-(P-I))(Pr-(P-I))^{-1}R + Pr(Pr-(P-I))^{-1}R + R - PrR \\ &= -R + Pr(Pr-(P-I))^{-1}R + R - PrR \\ &= Pr(Pr-(P-I))^{-1}R - PrR \end{aligned}$$

Or si $B = (Pr-(P-I))^{-1}$, alors $PrB - (P-I)B = I$ d'où $PrPrB - Pr(P-I)B = Pr$ soit encore $PrB + 0 = Pr$.

Finalement, on obtient :

$$(P-I)w^{\#} + R - c_1^{\#} = PrR - PrR = 0$$

.

Donc $(w^{\#}, c_1^{\#})$ est bien solution du système linéaire. En figure 11, on vérifie numériquement que $(w^{\#}, c_1^{\#})$ est solution du système.

Réciproquement, soit $(w^{\star}, c_1^{\star})$ solution du système linéaire. Alors en composant par Pr la première équation du système, on a :

$$Pr(P-I)w^{\star} + PrR - Prc_1^{\star} = 0 \Leftrightarrow PrR - Prc_1^{\star} = 0$$

.

En lui soustrayant la deuxième équation, on a :

$$PrR = (Pr-(P-I))c_1^{\star} \Leftrightarrow c_1^{\star} = (Pr-(P-I))^{-1}PrR = PrR$$

,

avec le même "jeu d'écriture" que précédemment.

Si l'on demande à Scicoslab de résoudre la fonction "linsolve" le système, on obtient bien $c_1^{\star} = PrR$ solution du système, comme on l'observe en figure 12.

```
-->A*w + R -c
 ans  =

   1.0D-14 *

  - 0.1776357
    0.
  - 0.1776357
    0.

-->A*c
 ans  =

   1.0D-15 *

    0.0274303
  - 0.2077331
    0.4816026
    0.2558717
```

FIGURE 11 – Vérification numérique que les expressions explicites de $(w^{\#}, c_1^{\#})$ sont solutions du système linéaire.

```
-->[x0,K]=linsolve([A,-eye(n,n);zeros(n,n),A],[R;zeros(n,1)]);

-->x0
 x0  =

  - 6.6482636
  - 3.8024927
    0.698828
    9.7519284
    8.3301531
    8.3301531
    8.3301531
    8.3301531

-->c
 c  =

    8.3301531
    8.3301531
    8.3301531
    8.3301531
```

FIGURE 12 – Vérification numérique que tout $c_1^\star$ solution du système linéaire est égal au vecteur PrR dont tous les termes sont égaux au coût ergodique.

Question 10

Maintenant, on se replace dans le cas où P la matrice de transition de la chaine est de la forme $P = \alpha P_1 + (1-\alpha)ez$. On choisit aussi une fonction de coût qui s'écrit pour un état i donné :

$$R(i) = \sum_j P_{i,j} Rm_{i,j}$$

avec R_m matrice donnée aussi de taille (n, n).

Supposons que $w^\#$ soit point fixe de l'opérateur $w \to \alpha P_1 w + b$, où l'on pose

$$b_i = R(i) = \sum_j (P \bullet R_m)_{ij}$$

. Alors, on a :

$$Pw^\# = \alpha P_1 w^\# + (1-\alpha)ezw^\# = w^\# - b + (1-\alpha)zw^\#$$

Soit, en posant $c^\# = (1-\alpha)zw^\#$:

$$Pw^\# + b = w^\# + c^\#$$

On vérifie cette égalité numériquement, capture d'écran donnée figure 13.

Question 11

L'opérateur $w \to \mathcal{L}(w) = \alpha P_1 w + b$ est contractant. En effet :

$$\forall\ w, w',\ ||\mathcal{L}(w) - \mathcal{L}(w')|| = ||\alpha P_1(w - w')|| \leq |\alpha|||P_1||||w - w'||$$

.

En choisissant $||P_1|| = \sup_i \sum_j |P_{ij}| = 1$, on a donc :

$$||\mathcal{L}(w) - \mathcal{L}(w')|| \leq \alpha||w||$$

Avec $\alpha < 1$, ce qui achève la preuve.

Par conséquent, pour obtenir le point fixe de l'opérateur $\mathcal{L}$, on peut l' itérer successivement ($w_{k+1} = \alpha P_1 w_k + b$) à partir d'un vecteur initial w_0, ce jusqu'à ce que $||w_{k+1} - w_k||$ soit suffisamment proche de 0.

On vérifie alors numériquement que $(w^\#, c^\# = (1-\alpha)zw^\#)$ est bien solution du système linéaire précédemment énoncé (voir figure 14).

5 Problèmes ergodiques et maximisation du PageRank

Question 12

L'idée est ici de résoudre le problème de maximisation du PageRank des m-premières pages décrit en question 7, en l'interprétant comme un problème ergodique.

En effet, en posant $Rm_{i,j} = 1_{i \in [1,m]}$, alors le cout ergodique correspond au PageRank des m-premières pages :

```
-->P
 P  =

    0.1530135    0.344231     0.2628937    0.2398618
    0.2580854    0.3692400    0.3126534    0.0600213
    0.1987008    0.1336505    0.2688250    0.3988237
    0.4139922    0.1289593    0.1613387    0.2957098

-->Rm
 Rm  =

    0.9940685    0.1251828    0.4905890    0.1258966
    0.8234578    0.3170995    0.0344461    0.3815585
    0.8219033    0.7637500    0.6636055    0.2102091
    0.6948286    0.9502220    0.4387444    0.7655168

-->w = linsolve(alpha * P1 - eye(n, n), sum(P.*Rm, 'c'));
-->

-->// calcul de c

-->c = (1-alpha)*z*w
 c  =

    0.4882653

-->

-->// (w,c) solution du pb ergodique ?

-->w + c - (P*w + sum(P.*Rm, 'c'))
 ans  =

    0.
    8.882D-16
    4.441D-16
    1.332D-15
```

FIGURE 13 – Vérification numérique que si $w^{\#}$ est point fixe de $\mathcal{L}$ (calcul par résolution d'un système linéaire),alors $(w^{\#}, c^{\#} = (1-\alpha)zw^{\#})$ solution du problème ergodique $Pw^{\#} + b = w^{\#} + c^{\#}$.

```
-->w=iterative_c(10*%eps);
-->// calcul de c

-->c = (1-alpha)*z*w
 c  =

    0.4882653

-->

-->// (w,c) solution du pb ergodique ?

-->w + c - (P*w + sum(P.*Rm,'c'))
 ans  =

   1.0D-14 *

  - 0.1332268
  - 0.1776357
  - 0.1776357
  - 0.1776357
```

FIGURE 14 – Calcul $w^{\#}$ par itérations successives de $\mathcal{L}$ dont il est point fixe. Vérification que $(w^{\#}, c^{\#} = (1-\alpha)zw^{\#})$ est toujours solution du problème ergodique.

$$c = \langle \pi, R \rangle = \sum_i \pi(i)R(i) = \sum_{i,j} \pi(i)P_{i,j}Rm_{i,j} = \sum_{i \in [1,m],j} \pi(i)P_{i,j} = \sum_{i \in [1,m]} \pi(i)$$

Aussi on sait par la question 10 que si $w^{\#} = \alpha P_1 w + b$ alors $c^{\#} = (1-\alpha)zw^{\#}$ correspond au cout ergodique, où on a noté pour rappel $b_i = R(i) = \sum_j (P \bullet R_m)_{ij}$.

Par conséquent, il s'agit finalement de trouver la sous-matrice d'adjacence $Adj(1:p, m+1:n)$ qui maximise w solution de $w = \mathcal{L}(w) = \alpha P_1 w + b$.

Ainsi, partant d'un w_0 initial, l'algorithme implémenté consiste à alterner contrôle optimal et calcul du "coût ergodique" associé (de w en pratique) jusqu'à stabilisation de w :

— choisir chaque ligne $i \in [1,p]$ de la matrice de "décision" $Adj^k(1:p, m+1:n)$ qui maximise :
$$\mathcal{L}_k(w^k)_i = \alpha(P_1^k w^k)_i + b_i^k$$

— calculer w_{k+1} point fixe de $\mathcal{L}_k$ et dont dérive le coût ergodique $c^{k+1} = (1-\alpha)zw^{k+1}$:
$$w^{k+1} = \mathcal{L}_k(w^{k+1}) = \alpha P_1^k w_{k+1} + b^k$$

.

Comme on l'observe en figures 15 et 16, l'algorithme d'optimisation semble terminer (stabilisation de w), il augmente bien le PageRank des 5 premières pages en supprimant le lien du noeud 1 vers le noeud 10, la somme des PageRank des 5 premières pages passant de 0.53 à plus de 0.56. Toutefois, le PageRank des 5 premières pages en fin d'algorithme reste curieusement inférieur à celui obtenu avec l'algorithme d'optimisation discret.

```
-->[Adj_optim_2, cost_2]=pagerank_ergodic(Adj, z,alpha,m, p, n);

-->[P_optim_2,Pss_optim_2,Pprim_optim_2,d_optim_2]=google2(Adj_optim_2, z, alpha);

-->pi_optim_2=pi_iterative_sparse(n, Pss_optim_2, d_optim_2, z, alpha);

-->Adj_optim_2
 Adj_optim_2  =

    1.    0.    0.    0.    1.    0.    0.    0.    0.    0.
    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
    0.    0.    0.    0.    0.    0.    0.    0.    0.    1.
    0.    1.    0.    0.    0.    1.    0.    0.    0.    0.
    1.    0.    0.    1.    0.    0.    0.    0.    1.    0.
    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
    0.    1.    0.    0.    0.    0.    1.    0.    0.    0.
    0.    0.    0.    1.    0.    0.    0.    0.    1.    0.
    0.    0.    0.    1.    0.    0.    1.    0.    0.    0.
    0.    0.    0.    1.    0.    0.    0.    0.    0.    0.

-->pi_optim_2
 pi_optim_2  =


         column 1 to 6

    0.1359633    0.1611149    0.0307775    0.1651639    0.0717647    0.1328157

         column  7 to 10

    0.1448978    0.0204169    0.0740132    0.0630721

-->sum(pi_optim_2(1:5))
 ans  =

    0.5647842
```

Figure 15 – Matrice d'adjacence en fin d'optimisation. PageRank et somme des PageRank des 5 premières pages dans nouveau le graphe.

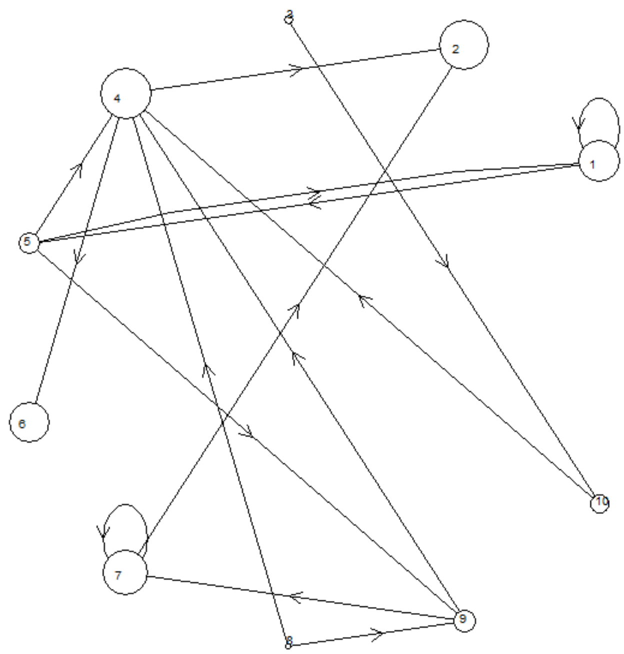

Figure 16 – Graphe associée à la matrice d'adjacence en fin d'optimisation.