



Master2 Informatique

UE: ALGORITHMIQUE DISTRIBUÉE – 4TIN913U

Responsable : C. Gavaille

Date : 7 janvier 2026

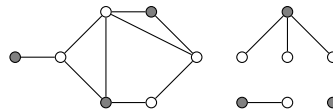
Durée : 1h30

Document: une feuille A4 recto-verso autorisée.

La langue pour vos réponses peut-être le français ou l'anglais, mais pas les deux!

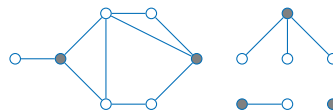
Petit Ensemble Dominant

On considère le problème consistant à construire dans un graphe G un ensemble dominant S de taille $|S| \leq |V(G)|/2$. On notera SDS un tel ensemble. On rappelle qu'un ensemble S est dominant pour G , si tout sommet de qui n'est pas dans S possède un voisin dans S . Un ensemble dominant S est *minimum* si sa taille $|S|$ est la plus petite possible parmi tous les ensembles dominants pour le graphe. On notera MDS un tel ensemble. Ci-dessous figure un exemple de SDS pour un graphe non connexe à 14 sommets.



Question 1. Déterminez un MDS pour le graphe ci-dessus en précisant sa taille. Un dessin avec le MDS distingué suffit. Justifiez le fait que votre solution est de taille minimum.

SOLUTION. [3 pts | Σ 3.0] Voici une solution de MDS.



Pour montrer qu'elle est minimum, il faut montrer qu'elle est minimum pour chacune de ses composantes. Il faut au moins 1 sommet par composant, ce qui règle le cas des trois plus petites. Pour la plus grosse, on remarque qu'un sommet couvre au plus 5 sommets (4 voisins et lui-même). Or la composante comprend > 5 sommets. Donc il faut au moins deux sommets pour cette composante, ce qui montre que la solution proposée est bien minimum.

Question 2. Quelle est, en fonction de n , la taille d'un MDS pour C_n , le cycle à n sommets. Justifiez votre réponse.

SOLUTION. [3 pts | Σ 6.0] Soit S un MDS pour C_n . Chaque bloc de trois sommets consécutifs doit comporter au moins un sommet dans S , ce qui implique $|S| \geq n/3$, et donc $|S| \geq \lceil n/3 \rceil$. D'un autre côté, en découpant C_n en $\lceil n/3 \rceil$ blocs de trois sommets et en choisissant le sommet central (ou n'importe lequel si le bloc à 1 ou 2 sommets) pour être dans S on obtient que $|S| \leq \lceil n/3 \rceil$. Donc $|S| = \lceil n/3 \rceil$.

On souhaite calculer un SDS pour C_n à l'aide d'un algorithme distribué dans le modèle LOCAL. Pour cela, on va supposer que C_n possède une 3-coloration arbitraire c , et on notera $c(u) \in \{0, 1, 2\}$ la couleur du sommet u . Donc au démarrage, chaque sommet u dispose de $c(u)$ ainsi que son identifiant $ID(u) \in \{0, \dots, n-1\}$. Notez bien que sans communiquer, u ne peut ni déterminer n ni la couleur la plus fréquente par exemple. Vous ne pouvez faire aucune hypothèse particulière sur c . Selon le cours, elle a pu, par exemple, être calculée en $\log^*(n) + O(1)$ rondes.

Question 3. Rappelez la définition de la fonction $\log^*(n)$? Que vaut $\log^*(16)$?

SOLUTION. [3 pts | $\sum 9.0$] Par définition $\log^*(n) = \min \{i \in \mathbb{N} : \log(\log(\dots \log(n))) \leq 1\}$ où la fonction \log (le logarithme en base 2) est itérée i fois. En particulier, $\log^*(16) = \log^*(2^{2^2}) = 3$ puisque $\log \log \log(2^{2^2}) = 1$.

Question 4. Montrez que pour tout entiers n, b tels que $1 \leq b \leq n/2$, il existe une 3-coloration de C_n où l'une des couleurs apparaît exactement b fois.

SOLUTION. [3 pts | $\sum 12.0$] On place les $b \leq n/2$ couleurs 0 par exemple, sans en mettre deux à côté, puis on complète avec les deux autres couleurs en alternant.

Question 5. Pourquoi une telle 3-coloration ne peut pas exister pour C_n si $b = 0$ ou $b > n/2$?

SOLUTION. [3 pts | $\sum 15.0$] Pour $b = 0$, il manque une couleur ce qui rend impossible le 2-coloriage de C_n pour n impair. Pour $b > n/2$, alors deux sommets de C_n seront voisins, et donc de la même couleur ce qui n'est pas valide.

On propose un premier algorithme distribué $A1$ calculant un SDS pour C_n . On supposera que la solution S est codée par une variable $s(u) \in \{\text{TRUE}, \text{FALSE}\}$ indiquant si $u \in S$ ou pas.

L'algorithme $A1$, sans ronde de communication, sélectionne u si et seulement si $c(u) = 0$. Autrement dit :

A1 : if $c(u) = 0$, then $s(u) := \text{TRUE}$ else $s(u) := \text{FALSE}$

Question 6. En vous appuyant sur les questions précédentes, démontrez que $A1$ n'est pas correct.

SOLUTION. [3 pts | $\sum 18.0$] En utilisant la question 4, il peut arriver que la couleur 0 apparaisse $b < n/3$ fois. Or, nous avons vu dans la question 2 que la taille du MDS est $\geq n/3$. L'algorithme $A1$ ne peut donc pas renvoyer un ensemble dominant pour C_n .

On se propose alors d'analyser un deuxième algorithme $A2$, probabiliste cette fois-ci, qui utilise la fonction $\text{RANDOM}(3)$ renvoyant un entier aléatoire uniforme pris dans $\{0, 1, 2\}$. On supposera que la fonction RANDOM est implémentée de la même manière sur tous les sommets, notamment que la graine initiale au démarrage est la même. Donc le premier appel à $\text{RANDOM}(3)$ donnera la même valeur pour tous les sommets qui l'exécute.

L'algorithme $A2$, également de zéro ronde, consiste alors à faire :

A2 : if $c(u) = \text{RANDOM}(3)$, then $s(u) := \text{TRUE}$ else $s(u) := \text{FALSE}$

Question 7. D'après vous, est-ce que la probabilité que la solution S calculée par $A2$ soit un SDS pour C_n est $\geq 1/2$? Justifiez.

SOLUTION. [3 pts | $\sum 21.0$] C'est a priori 0 car, quelque soit le choix RANDOM de la couleur, disons i , il est possible qu'aucune couleur donne un ensemble dominant valide. Malheureusement, un équilibre global (des fréquences de couleur) n'implique pas un équilibre local. Pour le voir, imaginons que chacune des trois couleurs ait la même fréquence d'apparition (équilibre global), soit $n/3$ (en supposant n multiple de 3).

Même avec cette fréquence on peut imaginer une « mauvaise » répartition locale pour l'algorithme $A2$. Par exemple, celle-ci :

$$\dots 1] - [0 - 1 - 0] - [2 - 0 - 2] - [1 - 2 - 1] - [0 \dots$$

où le patrone central se répéterait $n/9$ fois (en supposant n multiple aussi de 9). Il s'agit d'une 3-coloration et chaque couleur apparaît avec la même fréquence : 3 fois chacune pour chaque patrone de 9 sommets. Et pourtant, pour tout choix de couleur, un des trois blocs ne contient pas cette couleur, et donc le sommet central de ce bloc ne sera pas dominé. Le résultat ne sera pas un ensemble dominant.

Notons que cette mauvaise répartition implique aussi la question 6.

Pour vos algorithmes à venir, vous pourrez utiliser les deux primitives de communication suivantes :

- $\text{SENDALL}(m)$: envoie le message m à tous les voisins de u .
- $\text{RECEIVEALL}()$: renvoie la liste des messages reçus par u et émis à la ronde courante.

Vous pourrez utiliser NEWROUND pour d'indiquer le démarrage d'une nouvelle ronde.

Question 8. Proposez un algorithme (déterministe), évitant les écueils d'A1 et A2, permettant de calculer un SDS pour C_n supposé 3-colorié. Précisez le nombre de rondes. [Aide : Il en existe un qui s'exécute en seulement $O(1)$ rondes.]

SOLUTION. [4 pts | $\sum 25.0$] L'idée est d'ajouter chaque sommet s'il n'est pas dominé, et ceci en parallèle pour tous les sommets d'une même couleur, de manière à limiter le nombre de sommets dans l'ensemble.

Donc pour chaque couleur $i = 0, 1, 2$, le sommet u est ajouté à S si $c(u) = i$ et u n'est pas déjà dominé, c'est-à-dire si u et ses deux voisins ne sont pas déjà dans S .

À la fin, S est un ensemble indépendant (car à chaque ronde, pas deux sommets voisins n'ont pu être ajoutés à S puisqu'ils sont de la même couleur) et maximal (plus aucun sommet ne peut être ajouté à S puisque toutes les couleurs, et donc tous les sommets, ont été considérées). Il suit que S est un ensemble dominant (par maximalité de l'ensemble indépendant). Nous avons $|S| \leq n/2$, puisque sinon deux sommets seraient voisins : impossible, S est un ensemble indépendant. Donc S est un SDS pour C_n .

Plus précisément, l'algorithme de trois rondes s'écrit :

Algorithme Algorithm SDS

(code du sommet u)

1. $s := \text{FALSE}$ // au départ, aucun sommet dans S
 2. Pour chaque $i \in \{0, 1, 2\}$ // successivement pour chaque couleur
 - (a) NEWROUND
 - (b) SENDALL(s)
 - (c) $(s_v, s_w) := \text{RECEIVEALL}()$ // valeurs des 2 voisins de u
 - (d) si $c = i$ et $(s \vee s_v \vee s_w) = \text{FALSE}$, alors $s := \text{TRUE}$ // ajout si couleur i et qu'aucun n'est dans S
-

On se propose maintenant de faire le contraire : calculer une 3-coloration pour C_n à partir d'un ensemble dominant supposé minimum.

Question 9. Proposez un algorithme permettant de calculer une 3-coloration pour C_n en supposant connu un MDS pour C_n . Précisez bien le nombre de rondes.

SOLUTION. [4 pts | $\sum 29.0$] Soit S un MDS pour C_n . On va s'appuyer sur le fait que les configurations possibles des sommets qui sont dans S ou pas sont assez limitées. Si l'on considère trois sommets consécutifs, il y en aura un ou deux dans S : il n'est pas possible d'en avoir aucun ni que les trois soient dans S . En fait ceci est vrai non seulement si S est minimum, mais aussi si S est *minimal*, c'est-à-dire pour lequel la suppression d'un seul sommet aboutit à un ensemble qui n'est plus dominant.

On pourrait choisir de poser $c(u) := 1$ pour tous les sommets de S par exemple, mais deux voisins peuvent être dans S . Cependant on peut départager deux voisins dans S grâce à leurs identifiants. Cela suggère l'algorithme suivant qui prend 3 rondes. Dans la suite, v, w sont les deux voisins de u .

Initialement, on pose $c(u) := 0$ pour tous les sommets.

Ronde 1. Poser $c(u) := 1$, si $s(u) = 1$ et que $\text{ID}(u) < \text{ID}(v)$ pour tout voisin $v \in S$ de u .

On remarque qu'après cette ronde, il n'y a pas deux sommets de couleur 1 consécutifs car sinon c'est qu'il y avait eu trois sommets de S consécutifs. De plus, entre deux sommets de couleur 1, il n'y a qu'au plus 4 sommets de couleur 0 : initialement au plus deux sommets consécutifs n'étaient pas dans S et au pire on avait $\dots - 0 - S - S - 0 - 0 - S - S - 0 - \dots$ ce qui aurait pu donner $\dots - 0 - 1 - 0 - 0 - 0 - 1 - 0 - \dots$ après cette première ronde. Il est clair maintenant que le problème de 3-colorier les blocs de couleur 0 entre sommet de couleur 1 est de nature locale puisque la longueur de ces blocs vaut 1, 2, 3 ou 4. Afin de réduire la longueur des blocs de couleur 0, on va colorier par la couleur 2 tout sommet voisin d'un sommet de couleur 1.

Ronde 2. Poser $c(u) := 2$, si $c(u) = 0$ et $c(v) = 1$. (On pourrait imposer $c(w) = 0$ pour laisser inchangées les configurations $\dots - 1 - 0 - 1 - \dots$, mais cela n'a pas d'importance.)

Après cette étape, les blocs initiaux de longueur 1 ou 3 de couleur 0 sont correctement recoloriés en $\dots - 1 - 2 - 1 - \dots$ ou $\dots - 1 - 2 - 0 - 2 - 1 - \dots$. Pose problème uniquement les blocs qui étaient de longueur 2 ou 4, car ils se sont recoloriés en $\dots - 1 - 2 - 2 - 1 - \dots$ et $\dots - 1 - 2 - 0 - 0 - 2 - 1 - \dots$ respectivement. Il reste donc à gérer les cas où deux sommets voisins sont tous deux de couleur 0 ou 2. On peut facile les gérer en cassant la symétrie avec les identifiants.

Ronde 3. : Poser $c(u) := c(u) + 1 \pmod 2$, si $c(u) = c(v)$, et $ID(u) < ID(v)$.

En effet, les seuls cas possibles où un des voisins v de u est de même couleur, sont $c(u) = c(v) = 0$ ou $c(u) = c(v) = 2$. Dans le premier cas, il faut que celui de plus petit identifiant passe à 1. Dans le second cas, il faut qu'il passe à 0.

Question 10. En déduire qu'il n'est pas possible de calculer un MDS pour C_n en $o(\log^*(n))$ rondes.

SOLUTION. [3 pts | $\sum 32.0$] Si MDS pour C_n peut être résolu en $t = o(\log^*(n))$ rondes, alors, par la question 9, la 3-coloration de C_n peut être calculée en $t + t'$, où $t' = O(1)$ est temps pour le calcul d'une 3-coloration à partir d'un MDS pour C_n . Cela fait donc $o(\log^*(n))$ rondes pour une 3-coloration de C_n , en contradiction avec la borne inférieure de $\Omega(\log^*(n))$ vue en cours.

En utilisant la borne inférieure plus fine du cours pour t et la borne supérieure de $t' = 3$ de la question 9, on aurait pu même dire que $t + 3 \geq \frac{1}{2} \log^*(n) - 1$, soit $t \geq \frac{1}{2} \log^*(n) - 4$ pour le nombre de ronde pour calculer un MDS pour C_n .

Complexité de communication

Question 11. Expliquez ce qu'est un protocole de communication qui calcule une fonction booléenne f ? Quelles en sont les principales règles? Quel est l'objectif recherché?

SOLUTION. [3 pts | $\sum 35.0$] C'est un algorithme qui permet à deux participants, Alice et Bob, d'échanger des messages jusqu'à obtenir la valeur $f(x, y)$ où x, y sont respectivement les entrées d'Alice et Bob. Ils s'échangent des messages chacun à leur tour, Alice commençant, et le dernier bit doit être la réponse. Bien sûr, chacun ne connaît que son entrée et le protocole. L'objectif est de minimiser le nombre total de bits échangés.

Question 12. Pour démontrer quel résultat d'algorithmique distribuée vu en cours la complexité de communication nous a-t-elle servi?

SOLUTION. [3 pts | $\sum 38.0$] On s'est servi de la complexité de communication pour démontrer que dans le modèle b -CONGEST, il fallait $\Omega(\sqrt{n}/b)$ rondes dans le pire des cas pour détecter un C_4 dans un graphe à n sommets.

Question 13. Rappelez la définition d'un ensemble discriminant F_v pour une fonction booléenne f et une valeur $v \in \{0, 1\}$.

SOLUTION. [3 pts | $\sum 41.0$] Il s'agit d'un ensemble F_v d'entrées (x, y) pour f tel que, pour tout $(x, y) \neq (x', y') \in F_v$, $f(x, y) = f(x', y') = v$ et $\bar{v} \in \{f(x, y'), f(x', y)\}$, où $\bar{v} = 1 - v$ est la valeur complémentaire de v .

On notera $CC(f)$ la complexité de communication de f .

Question 14. Démontrez que si f possède un ensemble discriminant F_v tel que $|F_v| \geq 2^k$, pour un certain entier $k \geq 1$ et une valeur $v \in \{0, 1\}$, alors $CC(f) \geq k + 1$. [Aide : Attention au « +1 »! Utilisez les liens établis dans le cours entre $CC(f)$, le nombre de scripts et la taille des ensembles discriminants.]

SOLUTION. [3 pts | $\sum 44.0$] Par définition de F_v , la matrice de f doit contenir au moins une entrée (x_0, y_0) avec $f(x_0, y_0) = \bar{v}$, sinon F_v ne serait pas un ensemble discriminant de cardinalité $2^k \geq 2$ (car $k \geq 1$). En effet, $|F_v| \geq 2$ doit contenir deux entrées (x, y) et (x', y') distinctes telles que $f(x, y) = f(x', y') = v$ et donc soit $(x_0, y_0) = (x, y')$ ou $(x_0, y_0) = (x', y)$ est telle que $f(x_0, y_0) = \bar{v}$. Cette entrée (x_0, y_0) forme un ensemble

discriminant $\{(x_0, y_0)\}$ pour \bar{v} . D'après le cours, $CC(f) \geq \log_2 \sigma(f)$, le nombre de script minimum pour f , et $\sigma(f) \geq |F_v| + |\{(x_0, y_0)\}| = 2^k + 1$. Et donc $CC(f) \geq \log_2(2^k + 1) > \log_2(2^k) = k$. Comme $CC(f)$ et k sont des entiers, cela implique que $CC(f) \geq k + 1$.

Soit Π un protocole de communication pour une certaine fonction f tels que Π ne génère que deux scripts possibles sur toutes les entrées $(x, y) \in \{0, 1\}^{2n}$ de f .

Question 15. Montrez qu'Alice et Bob peuvent, en interne, utiliser Π pour calculer $f(x, y)$ en échangeant au plus 2 bits au total, quelle que soit l'entrée (x, y) . Ce nouveau protocole devra être sans préfixe.

SOLUTION. [3 pts | $\sum 47.0$] Soient $s_1 = (m_0, m_1, \dots, m_i)$ et $s_2 = (n_0, n_1, \dots, n_i)$ les deux seuls scripts générés par Π pour f . Notons qu'Alice et Bob peuvent calculer s_1 et s_2 sans communication, puisqu'ils connaissent f , Π et n .

Soit i le plus grand indice tel que $(m_0, m_1, \dots, m_i) = (n_0, n_1, \dots, n_i)$, indice qu'Alice et Bob peuvent calculer sans communication. Comme les $i + 1$ premiers messages des deux scripts sont communs, Alice et Bob pourraient démarrer le protocole directement en envoyant le message m_{i+1} de s_1 ou n_{i+1} de s_2 .

Si $i + 1$ est pair, alors c'est au tour d'Alice de communiquer m_{i+1} ou n_{i+1} en fonction de son entrée. C'est donc Alice qui va déterminer si c'est s_1 ou s_2 qui doit être utilisé. Elle peut donc directement envoyer la réponse, le dernier message du bon script, qui vaut m_i (si c'est s_1) ou n_i (si c'est s_2). Dans ce cas, 1 bit suffit.

Si $i + 1$ est impair, alors c'est Bob qui va déterminer si c'est s_1 ou s_2 qui doit être utilisé. Et cela ne peut dépendre d'Alice. Bob peut donc transmettre comme dernier bit la réponse m_i ou n_i , et Alice pourra envoyer 0 comme premier message par exemple. Dans ce cas, 2 bits suffisent.

On peut modifier légèrement le protocole pour le rendre sans-préfixe dans le cas $i + 1$ impair. Pour cela Alice, ajoute un « 1 » et envoie « $1m_i$ » ou « $1n_i$ ». Il y a toujours au plus deux bits transmis en tout et ce nouveau protocole Π' est sans préfixe. En effet, l'ensembles des messages (concaténation des scripts) jusqu'à la ronde $j \in \{0, 1\}$ sont : $\mathcal{M}_{\Pi'}(0) = \{10, 11\}$ et $\mathcal{M}_{\Pi'}(1) = \{10, 11, 00, 01\}$, soient deux ensembles sans-préfixes.

On considère la fonction $lt_n(x, y) \in \{0, 1\}$ (pour *less-than*) qui étant donnés deux mots binaires x, y , chacun de n bits, renvoie 1 si et seulement si « $x < y$ », c'est-à-dire si les entiers représentés en binaires par les mots x et y vérifient $x < y$. Par exemple, $lt_2(11, 10) = 0$. En revanche, $lt_2(01, 10) = 1$. Bien évidemment, $lt_n(x, x) = 0$ pour tout $x \in \{0, 1\}^n$.

Question 16. Calculez $CC(lt_n)$. Justifiez. [Aide : Notez que $CC(lt_n) \leq n + 1$.]

SOLUTION. [3 pts | $\sum 50.0$] Considérons la matrice de lt_n . Clairement, elle est triangulaire supérieure par définition de $lt_n(x, y)$, avec des 0 pour les valeurs diagonales $lt_n(x, x)$. Par exemple, pour $n = 3$ nous obtenons.

	y	000	001	010	011	100	101	110	111
x	000	0	1	1	1	1	1	1	1
001	0	0	1	1	1	1	1	1	1
010	0	0	0	1	1	1	1	1	1
011	0	0	0	0	1	1	1	1	1
100	0	0	0	0	0	1	1	1	1
101	0	0	0	0	0	0	1	1	1
110	0	0	0	0	0	0	0	0	1
111	0	0	0	0	0	0	0	0	0

L'ensemble $F_0 = \{(x, x) : x \in \{0, 1\}^n\}$ est discriminant pour la valeur 0 puisqu'il constitue la diagonale de 0 et que toutes les valeurs au-dessus sont des 1.

Comme $|F_0| = 2^n$, on peut appliquer la question 14 pour obtenir $CC(lt_n) \geq n + 1$. Comme $CC(f) \leq n + 1$ pour tout f , on en déduit que $CC(lt_n) = n + 1$.

FIN.