

Chapitre 3

Algorithmes d'approximation.

Dans ce chapitre, nous reprenons certains de nos résultats relatifs à la construction d'algorithmes d'approximation ou la preuve de non-approximabilité de problèmes.

Dans la première section, nous rappelons les bases de la théorie de l'approximation. Nous présentons ensuite plusieurs de nos résultats relatifs à la numérotation de sommets sur les graphes d'intervalle (essentiellement [27]). Nous présentons ensuite nos travaux sur l'ordonnancement pour la redistribution de données (essentiellement [32], [14]) puis sur l'ordonnancement de messages dans les réseaux optiques (essentiellement [7], [20], [21], [33]).

Chacun de ces travaux est présenté via une discussion de l'état de l'art des problèmes correspondants.

3.1 Problèmes d'optimisation

Dans cette section, nous introduisons la notion d'approximation d'un problème d'optimisation sous l'angle de la complexité. Face à un problème d'optimisation (par exemple calculer le nombre chromatique d'un graphe, la distance ou le chemin entre deux sommets du graphe), avoir un algorithme le plus *efficace* possible est important. La notion d'efficacité est souvent cruciale. En particulier lorsque les problèmes de décision sont NP-difficiles, calculer la solution exacte peut prendre beaucoup de temps de façon pratique (des mois, voir des années). Dans de tels cas, il paraît judicieux d'avoir des algorithmes calculant une solution proche de l'optimum en un temps polynomial.

A partir de cette constatation, la théorie de l'approximation permet d'évaluer la qualité d'une solution calculée par un algorithme et de classifier les différents problèmes.

Voici une définition plus formelle d'un problème d'optimisation :

Définition 6 (Problème d'optimisation) *Un problème d'optimisation Π est un quadruplet $(I_\Pi, SOL_\Pi, m_\Pi, goal_\Pi)$ où*

- I_Π est l'ensemble des instances ;
- $SOL_\Pi(x)$ est l'ensemble des solutions possibles pour chaque instance $x \in I_\Pi$;
- m_Π est une fonction (mesure) qui associe une valeur à chaque solution possible d'une instance ;
- $goal_\Pi$ détermine s'il s'agit de maximiser ($goal_\Pi = max$) ou de minimiser ($goal_\Pi = min$).

Bien sûr, l'objectif d'un problème d'optimisation Π est de construire une solution y^* de l'instance

$$m_{\Pi}(x, y^*) = \text{goal}_{\Pi} \{m_{\Pi}(x, y) : y \in \text{SOL}_{\Pi}(x)\}.$$

Par exemple, le problème de calculer le nombre chromatique d'un graphe correspond à déterminer le nombre de couleurs pour colorier ce graphe sans que deux sommets adjacents aient la même couleur.

Le problème de décision associé est le suivant :

Instance : un graphe non-orienté G et un entier K

Question : existe-t-il une coloration propre du graphe ayant un nombre de couleurs inférieur à K ?

Ce problème est NP-complet.

Sa formulation en tant que problème d'optimisation avec la définition précédente est la suivante :

- I_{Π} : est un ensemble des graphes non-orientés G ;
- $\text{SOL}_{\Pi}(G)$: est l'ensemble des colorations propres de G ;
- m_{Π} : est une fonction retournant le nombre de couleurs dans une coloration ;
- c'est un problème de minimisation ($\text{goal}_{\Pi} = \min$).

Nous introduisons les notations suivantes pour une instance x d'un problème d'optimisation Π :

- $|x|$ est la taille de l'instance, c'est-à-dire la longueur de sa représentation binaire.
- $m_{\Pi}^*(x)$ est la valeur de la fonction m_{Π} d'une solution optimale du problème pour l'instance x .

Maintenant nous pouvons définir :

Définition 7 (La classe NPO) *Un problème d'optimisation Π est dans la classe NPO Π si*

- l'ensemble des instances est reconnaissable en temps polynomial ;
- il existe un polynôme q tel que pour chaque instance x et pour toute solution $y \in \text{SOL}_{\Pi}(x)$, on a $|y| \leq q(|x|)$;
- pour chaque instance x et pour tout y tel que $|y| \leq q(|x|)$, le problème de décider si $y \in \text{SOL}_{\Pi}(x)$ peut être résolu en temps polynomial ;
- La fonction m_{Π} est calculable en temps polynomial.

Remarquons que les problèmes de décision dans NP correspondent à des problèmes d'optimisation dans la classe NPO. Maintenant, nous allons introduire la notion d'évaluation de la qualité des solutions.

Définition 8 (Rapport à l'optimum) *Étant donnée une instance x d'un problème d'optimisation Π , une solution y possible de x a le rapport à l'optimum*

$$r_{\Pi}(x, y) = \max \left\{ \frac{m_{\Pi}(x, y)}{m_{\Pi}^*(x)}, \frac{m_{\Pi}^*(x)}{m_{\Pi}(x, y)} \right\}$$

Définition 9 (Algorithme d'approximation) *Un algorithme \mathcal{A} donne une approximation à un facteur $f(n)$ près en temps polynomial pour un problème d'optimisation Π si pour chacune de ses instances x de taille n , il calcule en temps polynomial une solution y possible pour x telle que $r_{\Pi}(x, y) \leq f(n)$.*

De plus, si un tel algorithme existe pour le problème d'optimisation Π , Π est dit approximable à un facteur $f(n)$ près.

Voici, une classe intéressante de problèmes ayant des algorithmes d'approximation de bonne qualité :

Définition 10 (La classe APX) La classe APX est la classe de tous les problèmes de NPO approximables à un facteur constant près (c'est-à-dire tels qu'il existe une approximation à un facteur r près en temps polynomial pour une certaine constante $r \geq 1$).

Malheureusement, il existe des problèmes d'optimisation qui semblent être durs à approximer. Une technique classique (utilisée dans [34] par exemple) pour prouver la non approximation d'un problème est celle de la *réduction de préservation d'intervalle* (en anglais *gap technique*) : on applique le théorème ci-dessous (voir [Ausiello et al., 1999] pour la preuve).

Théorème 7 Soit \mathcal{P} un problème de décision NP-complet, et soit Π un problème d'optimisation de minimisation dans NPO. Supposons qu'il existe des fonctions calculables en temps polynomial $f : I_{\mathcal{P}} \mapsto I_{\Pi}$ et $c : I_{\Pi} \mapsto \mathbb{N}$ et une constante $r > 0$ telles que pour toute instance x de \mathcal{P}

$$m_{\Pi}^*(f(x)) = \begin{cases} c(x) & \text{si } x \text{ est une instance positive} \\ (1+r)c(x) & \text{sinon} \end{cases}$$

Alors, à moins que $P = NP$, il n'existe pas d'algorithme d'approximation à un facteur r près de Π en temps polynomial.

Bien sûr, la technique reste la même pour un problème de maximisation.

3.2 Numérotation de sommets sur les graphes d'intervalle

Certains problèmes d'optimisation liés aux numérotations de sommets restent ouverts. Plus précisément, une *numérotation de sommets* d'un graphe $G = (V, E)$ est une numérotation des sommets $L : V \rightarrow [1, \dots, n]$ avec n correspondant au nombre de sommets de V . Le *poids* d'une arête $e = (u, v)$ d'une numérotation de sommets L dans G est $w_L(e) = |L(u) - L(v)|$. A partir de cette définition, plusieurs problèmes d'optimisation ont été introduits comme par exemple :

- Minimiser la *largeur de bande* (*bandwidth* en anglais). La largeur de bande d'une numérotation L d'un graphe G que l'on notera $bw_L(G)$ correspond à

$$bw_L(G) = \max \{w_L(e) : e \in E\}$$

- Minimiser l'*arrangement linéaire* (*linear arrangement* en anglais). L'arrangement linéaire d'une numérotation L d'un graphe G que l'on notera $ola_L(G)$ correspond à

$$ola_L(G) = \sum_{e \in E} w_L(e)$$

- Minimiser la *largeur de coupe* (*cutwidth* en anglais). La largeur de coupe d'une numérotation L d'un graphe G que l'on notera $cw_L(G)$ correspond à

$$cw_L(G) = \max_{1 \leq i \leq n} |\{u, v\} \in E \mid L(u) \leq i \leq L(v)\}|$$

Les problèmes de décision (minimisation) associés à ces trois problèmes sont NP-difficiles dans les graphes quelconques [Garey and Johnson, 1979].

Nous allons discuter de ces paramètres sur des types de graphes classiques comme par exemple les graphes d'intervalle.

Définition 11 (graphe d'intervalle) *Un graphe est un graphe d'intervalle s'il existe une bijection entre les sommets du graphe et un ensemble de segments d'une droite telle que deux sommets u et v sont reliés par une arête si et seulement si les segments correspondant à u et v se superposent.*

Beaucoup de problèmes classiquement NP-complets deviennent polynomiaux pour les graphes d'intervalle. En fait, la majorité des algorithmes polynomiaux se basent sur leur représentation graphique ou sur leur numérotation de sommets *reo* ou *leo*.

Définition 12 *La numérotation sommet *reo* (resp. *leo*) du graphe G par rapport à une représentation graphique est telle que l'ordre des sommets est défini par l'extrémité droite (resp. gauche) des intervalles.*

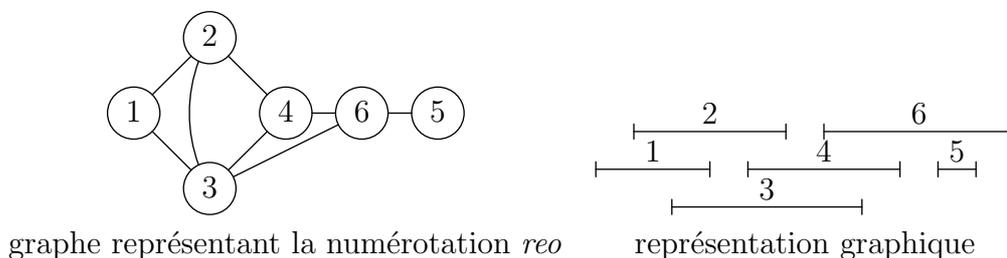


FIG. 3.1 – Graphe d'intervalle et sa représentation graphique

A propos de la largeur de bande. Le problème de la minimisation de la largeur de bande est NP-difficile dans les graphes quelconques. Il reste aussi dur si les graphes sont des arbres [Monien, 1986]. De plus, en général, la largeur de bande ne peut pas s'approcher à un facteur constant multiplicatif en temps polynomial [Unger, 1998] mais il peut s'approcher en un temps polynomial avec un facteur multiplicatif $O(\log^{9/2} n)$ [Feige, 2000].

Pour les graphes d'intervalle, un algorithme décrit dans [Kleitman and Vohra, 1990], résout un problème de décision 'A-t-on $bw(G) \leq k$?' en $O(nk)$ opérations et construit une numérotation de sommets ayant une largeur de bande minimum en $O(n^2 \log n)$ opérations. De plus, dans [Sprague, 1994] une amélioration de cet algorithme a été proposée en répondant à la question de décision en $O(n \log n)$ opérations et en construisant effectivement une numérotation en $O(n \log^2 n)$ opérations. Par contre, la complexité de ce problème pour les graphes de permutation est encore une question ouverte.

Pour les graphes d'intervalle, par la remarque faite dans [Kratsch and Stewart, 2002], la numérotation sommet (*reo*) satisfait la propriété suivante : pour chaque paire de sommets adjacents u et v tel que $reo(u) < reo(w)$, chaque sommet v tel que $reo(u) < reo(v) < reo(w)$ est adjacent à v . Ce qui implique que

Théorème 8 ([Kratsch and Stewart, 2002]) *Les numérotations reo et leo donnent une 2-approximation pour le problème de la largeur de bande.*

A propos de l'arrangement linéaire Calculer un arrangement linéaire optimal est NP-complet dans un graphe biparti [Even and Shiloach, 1975]. Par contre, il est possible de le calculer en temps polynomial pour des arbres [Goldberg and Klipker, 1976], [Chung, 1988], [Shiloach, 1979], mais aussi pour des classes de graphes comme les grilles, les hypercubes [Díaz et al., 2002].

Le meilleur algorithme d'approximation connu est à un facteur multiplicatif $O(\log n)$ [Uriel Feige and Lee, 2007]. Pour les classes classiques comme les graphes d'intervalles, les graphes de permutation et les co-graphes, ce problème est NP-complet :

Théorème 9 ([27]) *Déterminer si $ola(G) \leq k$ est NP-complet pour les graphes d'intervalle, les graphes de permutation et les graphes co-comparables.*

Nous nous sommes intéressé à ce problème lié à la modélisation des interactions de protéines de [Farach-Colton et al., 2004]. Farach-Colton et al. ont utilisé les graphes d'intervalle pour étudier la formation d'un type de ribosomes via des interactions des différentes protéines.

Si l'on considère que le graphe est construit par une succession d'insertions de nouveaux sommets en fonction de la numérotation *reo*, alors à chaque insertion, une étoile est rajoutée de centre le nouveau sommet. Ce qui implique que le coût supplémentaire des ces ajouts d'arêtes correspond à celui d'une étoile. Par une simple analyse, pour une étoile de n sommets, il est facile de voir que la meilleure numérotation correspond à celle numérotant le centre de l'étoile à $\lfloor n/2 \rfloor$ et que la pire correspond à celle numérotant le centre de l'étoile à n ou à 1. Le rapport de ce paramètre pour ces deux numérotations est de 2.

Cela signifie que l'algorithme qui met sous forme d'intervalles le graphe et qui trie les intervalles en les numérotant selon l'ordre *reo* est une 2-approximation.

Théorème 10 ([27]) *L'ordre reo est une approximation à un facteur multiplicatif 2 près.*

Cette borne est atteinte par le graphe correspondant à l'étoile (voir figure 3.2)

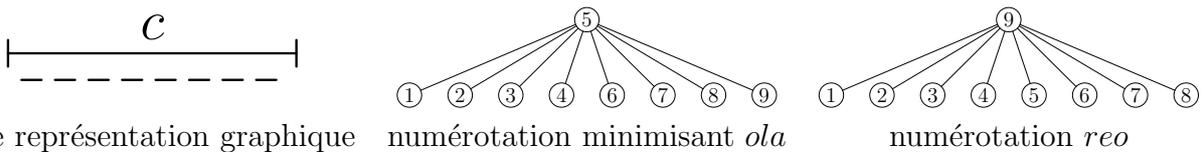


FIG. 3.2 – Etoile ayant huit feuilles et ses deux numérotations de sommets la meilleure et la pire.

De plus, pour les graphes co-comparables,

Théorème 11 ([27]) *Il existe un algorithme polynomial d'approximation pour les graphes co-comparables (basé sur leur ordre) à un facteur multiplicatif huit près.*

Pour l'instant, aucun exemple de graphe a été construit permettant d'affirmer que cette borne supérieure de huit est atteinte. Les cas pathologiques identifiés sont à six de l'optimal.

A propos de la largeur de coupe Ce problème est aussi NP-complet même si le graphe est planaire ayant un degré maximum de trois [Monien and Sudborough, 1986], grilles incomplètes [Díaz et al., 2001]. Par contre, pour les arbres, le problème devient polynomial [Yannakakis, 1985]. Il existe un polynomial algorithme d'approximation pour un graphe général avec un facteur multiplicatif $O(\log^2 n)$ [Leighton and Rao, 1988].

Par contre, déterminer si $cw(G) \leq k$ est NP-complet pour les graphes d'intervalles est une question ouverte. Récemment, un algorithme polynomial calcule ce paramètre dans des graphes d'intervalle particuliers [Heggernes et al., 2008] (graphes d'intervalle propres et graphes seuillés (*threshold* en anglais)).

Tous ces travaux tentent de trouver la frontière entre les problèmes difficiles et ceux polynomiaux à résoudre pour des graphes classiques.

3.3 Ordonnancement pour la redistribution de données

Nous avons travaillé sur le problème de la redistribution de données. Ce travail s'inscrivait dans l'Action de Recherche Incitative de l'INRIA *RedGrid*. La redistribution de données a été étudiée dans le contexte du calcul parallèle dans [Afrati et al., 2005], [Afrati et al., 2002], [Bongiovanni et al., 1981], [Crescenzi et al., 2001], ainsi que dans [Desprez et al., 1998]. Elle survient lorsque des données réparties sur un ensemble de processeurs doivent être transmises (éventuellement vers un autre ensemble de processeurs) le plus efficacement possible. Il s'agit d'ordonner les communications de manière à minimiser la durée totale de celles-ci.

Nous nous sommes penchées sur la résolution de ce problème dans un scénario particulier : des ordinateurs de deux grappes s'échangent des données via un réseau haut débit. Par exemple, supposons que le premier ensemble d'ordinateurs soit un ensemble de 50 PCs avec des cartes réseaux à 1 Gbit/s et que le second soit un autre ensemble de 200 PCs avec des cartes à 100 Mbit/s. Supposons que le réseau les connectant soit un réseau à 10 Gbit/s. Alors le premier ensemble ne pourra pas émettre plus de 10 communications à 1 Gbit tandis que le second ne pourra pas recevoir plus de 100 communications à 100 Mbits. Le nombre de communications simultanées maximum est le minimum des PCs dans les deux grappes soit $k = 10$.

En fonction des contraintes physiques, un modèle de communications peut se dessiner. Par exemple, chaque ordinateur ne peut gérer qu'une seule communication à la fois (mode 1-port). De plus, le réseau haut débit est une ressource critique, et le nombre de communications maximales que peut supporter le réseau inter-connectant les deux grappes d'ordinateurs au même instant est k .

Les communications sont uniquement entre les ordinateurs des différentes grappes. Elles sont représentées sous forme de matrices où chaque élément correspond à la quantité d'information qui doit être échangée. La modélisation naturelle est de représenter

ce problème sous forme de graphe biparti $G = (V_1, V_2, E)$ pondéré par une fonction $w : E \rightarrow \mathbb{Q}^+$ [Crescenzi et al., 2001].

Chaque ordinateur de la première grappe correspond à un sommet de V_1 . Il en est de même pour la seconde grappe. Une arête représente un échange d'information et son poids correspond à la durée du transfert total entre deux ordinateurs.

Pour exécuter les communications nous décomposons la redistribution en phases. Chaque phase représente des communications entre un ensemble de processeurs des deux ensembles d'ordinateurs en respectant le modèle 1-port, et en autorisant la préemption : un transfert de données peut être effectué en plusieurs étapes. A chaque phase, plusieurs communications sont réalisées simultanément et chaque phase a un coût d'initialisation dénoté par β et un coût de transfert de données dépendant de la quantité d'information à traiter.

De plus, des travaux comme [Afrati et al., 2005], [Crescenzi et al., 2001] supposent qu'il n'existe pas de goulot d'étranglement, c'est-à-dire que le réseau haut débit peut supporter autant de communications simultanées. Ce n'est pas la ressource critique. Dans [32], nous avons pris en compte cette contrainte. En généralisant le problème de la redistribution dans le cas où le nombre de communications simultanées est borné par un paramètre k entre les deux grappes de machines. Ce problème survient en particulier lorsque l'on veut redistribuer des données entre des processeurs de deux grappes de PCs reliées par un réseau à haut débit dans le cas d'applications de couplage de code ou de visualisation interactive à distance d'images calculées en parallèle.

Ce problème a été partiellement étudié dans le contexte d'accès au médium de communication pour les satellites [Gopal et al., 1982], [Gopal and Wong, 1985]. En particulier, dans [Bongiovanni et al., 1981], un algorithme polynomial a été étudié pour le cas où la phase d'initialisation est négligée ($\beta = 0$). Dans ce même contexte, lorsque le découpage des messages (préemption) n'est pas autorisé, le problème devient NP-complet et des heuristiques ont été proposées dans le même article.

Dans le cas où la phase d'initialisation de chaque étape possède un coût, lorsque le réseau haut débit n'est pas un goulot d'étranglement, il a été montré que minimiser le temps de communication est NP-complet [Even et al., 1976], [Gopal and Wong, 1985] et qu'il n'est pas approximable à $\frac{7}{6}$ [Crescenzi et al., 2001] à moins que $P = NP$. Ce problème reste NP-complet même si deux communications simultanées entre les deux grappes sont autorisées [32].

En fait, résoudre le problème revient à décomposer le graphe biparti en s couplages pondérés où chaque couplage correspond à une phase de communications. Tous les algorithmes d'approximation se basent sur la construction d'une décomposition de couplages dans un graphe biparti inspirée des techniques de coloration des arêtes pour les graphes bipartis [Berge, 1987].

Une approximation de ce problème à un facteur multiplicatif 2 près a été proposée dans [Crescenzi et al., 2001]. De plus, une modification de cette approximation a été apportée et permet de diminuer le rapport d'approximation $2 - \frac{1}{\beta+1}$ dans [Afrati et al., 2005]. Dans [14], nous proposons des algorithmes d'approximation à un facteur $8/3$ de l'optimal. Pour modéliser le goulot d'étranglement, des sommets et des arêtes ont été rajoutés dans le graphe biparti pour que chaque couplage dans ce nouveau graphe ait au plus k arêtes modélisant des communications.

Dans [14], nous avons validé par simulation et expérimentalement avec deux grappes de processeurs que nos algorithmes étaient plus efficaces que plusieurs autres approches

et aussi évalué l'impact du coût de la fragmentation des messages.

Ce travail fut un point de départ de la thèse de Frédéric Wagner qui a étendu ce problème à des scénarios plus complexes de redistributions de données [Wagner, 2005].

Aussi, le problème d'ordonnancement de messages apparaît dans le contexte de la redistribution des données mais aussi dans le contexte de la commutation par paquets pour multiplexage en longueur d'onde (WDM) des réseaux optiques [Choi et al., 1996], [Pieris and G.H., 1994], [Rouskas and Sivaraman, 1996]. Ces problèmes ont été traités comme des problèmes de colorations de chemins dans des graphes où les chemins représentent les communications avec la contrainte qu'une arête peut être traversée que par un nombre limité de chemins de même couleur. Dans les arbres, des algorithmes polynomiaux d'approximation à un facteur multiplicatif 4 ont été décrits dans [Chekuri et al., 2007] et dans [Erlebach et al., 2003].

3.4 Ordonnancement de messages dans les réseaux optiques

Ce travail s'inscrivait dans le projet européen DAVID [project of the 5th PCRD, 2005] du 5ième PCRD (*Data and Voice Integration on DWDM*). Ce projet travaillait autour du développement de méthodes pour la résolution du problème du routage dans des réseaux optiques sans possibilité de stockage. Dans ce projet, l'architecture de réseaux de télécommunications envisagée consiste à connecter des réseaux MAN, ayant la technologie WDM (Wave Division Multiplexing), dont la topologie correspond à une union d'anneaux, par un réseau européen WAN.

Ce travail a été fait en collaboration avec Stéphane Rousseau dont la thèse portait sur l'algorithmique du routage dans de tel réseaux [Rousseau, 2006].

En fait, nous nous sommes penchées sur une topologie bien particulière : une union d'anneaux. De tels réseaux transportent des cellules consécutives de taille identique fixée, tournant continûment sur l'anneau, vides ou contenant une information sous forme d'un paquet de données (on parle d'anneaux *slottés*). Le réseau est synchrone et les paquets optiques circulant sur les réseaux sont de taille fixe. Le temps peut-être discrétisé en étapes (appelées *slots*). Nous nous plaçons dans un réseau de n nœuds. Chaque lien contient $k - 1$ slots et sur chaque nœud est présent un slot. Cela implique qu'au plus $n \times k$ paquets optiques peuvent être en même temps sur le réseau. Un paquet optique envoyé au nœud i à la date t arrivera au nœud $i + 1$ à la date $t + k$. Le paquet optique est caractérisé par son nœud d'origine, son nœud de destination, et sa date d'envoi.

Le fonctionnement de ce réseau ressemble au problème d'ordonnancement que l'on appelle *Pinwheel* [Chen and Mok, 2004] pour l'accès au médium de communication pour les satellites dans le contexte du temps réel avec des émissions des messages de façon périodique.

À chaque étape de communication, une fois envoyés sur le réseau, les paquets ne peuvent pas être stockés par un nœud intermédiaire et ils sont acheminés jusqu'à atteindre leur destination. Cette contrainte caractérise les réseaux tout-optiques.

Un message est donc constitué de plusieurs paquets. La difficulté est d'assurer une contiguïté des paquets d'un même message sur le réseau pour garantir certaines qualités de service (on parlera de *communications contiguës*).

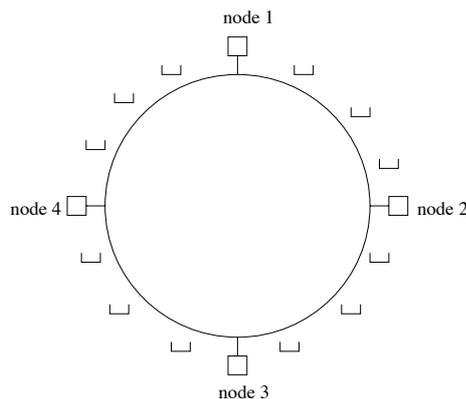


FIG. 3.3 – Réseau composé d’un seul anneau slotté

Une des questions abordées dans le projet DAVID est de comprendre l’impact de cette contrainte dans le réseau. Une première question est de comprendre comment insérer des messages en respectant ces messages. Nous avons partiellement répondu à cette question [7].

Une seconde question porte sur la gestion et l’utilisation des ressources liées à des problèmes d’optimisation. En fonction d’un ensemble de messages (caractérisés par leur source, leur destination, et leur date de création), la question est de construire un algorithme d’accès au médium qui permette d’assurer une certaine qualité de service (minimisation du temps d’attente, du délai moyen, du temps d’arrivée de tous les messages, maximisation du taux d’occupation). Les problèmes de décision associés sont tous NP-complets et des heuristiques ont été développées [20, 21].

Nous sommes concentrées plus particulièrement sur la date de fin d’acheminement de tous les messages.

Lorsque la taille des messages est identique, minimiser le temps de fin d’arrivée de tous les paquets est NP-complet lorsque la topologie est un anneau ou une chaîne. Dans tous ces cas il existe une approximation glouton à un facteur multiplicatif 2 près : dès que un noeud peut insérer un message, il l’insère. Cette approximation a l’avantage qu’elle est répartie puisque toutes les décisions sont locales et que ses performances ne se dégradent pas si les paquets arrivent à la volée au fur et à mesure [21]. Il est à noter que lorsque la taille des messages varie, cet algorithme devient une approximation à un facteur $\ell + 1$ près avec ℓ la taille maximale des messages.

Par contre, le problème devient polynomial lorsque tous les messages dans le réseau arrivent en même temps et que la topologie est une chaîne. Il suffit simplement de considérer que le problème équivaut à une coloration de sommets d’un graphe d’intervalle en considérant les sommets dans l’ordre *leo* [21].

Ceci se rapproche des problèmes d’ordonnancement que l’on appelle *Flowshop* (voir [Brucker, 2001]) où toutes les tâches doivent être exécutées sur plusieurs machines dans le même ordre. Mais, dans le problème Flowshop, il n’existe en général aucune contrainte sur le temps d’attente entre les machines. Ici, dans notre contexte, il y a de telles contraintes.

Dans la continuité de ce travail, en collaboration avec Henry Amet, Freddy Deffner, Marie-Claude Portmann, Stéphane Rousseau, nous avons travaillé sur la création d’heuristiques centralisées de ce problème [33].