Cet article est rédigé par des élèves. Il peut comporter des oublis ou des imperfections, autant que possible signalés par nos relecteurs dans les notes d'édition.

Robocal

2016-2017

Nom, prénom et niveaux des élèves : Mélanie CETTOUR-CAVE, Thomas MOREAU, Elodie STAMPONE, Yann VINCENT : élèves de Terminale S.

Établissement : Lycée de la Versoie, Thonon les Bains ; Institut Florimont, Petit Lancy (Suisse)

Enseignant(s) : Sabrina Biglione, Murièle Jacquier, Gilles Lamboley

Chercheur(s) : Catriona MacLean, Université Grenoble-Alpes ; Fathi Ben Aribi, Université de Genève

Table des matières

1 Présentation

1.1 prérequis

Dans le cas de notre sujet nous traitons des graphes, nous allons donc définir quelques termes propre aux graphes en plus de ceux propre à notre sujet pour que les personnes n'ayant aucune connaissance sur les graphes ne soient pas perdues.

Un graphe : c'est un ensemble de points appelés **sommets** reliés entre eux par des segments appelés **arêtes**.

Un sommet : ils sont représentés par des points, et sont pourvues de coordonnées données par des bornes.

Une arrête : elles sont représentées par un segment qui relie deux sommets.

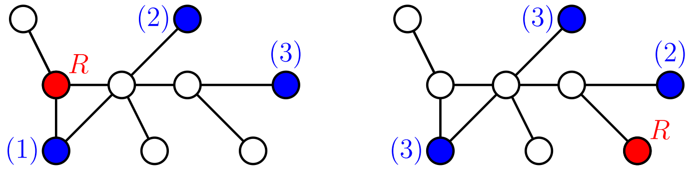
Une balise ou borne : elles sont représentées par des cercles bleus et enverront pour chaque sommets une coordonnée qui est la distance entre deux sommets.

La distance entre deux sommets : lorsque nous parlons de distance nous voulons plutôt désigner le plus petit nombre d'arêtes qui sépare un sommet d'une balise.

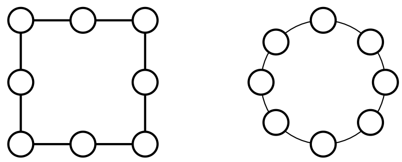
1.2 Description du sujet

On prend un graphe quelconque. Sur ce graphe un robot se déplace. Sur certains sommets du graphe, il y a des bornes que nous auront préalablement placées. Le robot possède un plan prédéfini du graphe avec la position des bornes qui indiquent la distance séparant le sommet où il se situe et le sommet où se situe la borne. Le but est de faire en sorte que le robot puisse se situer sur le graphe pour pouvoir se déplacer jusqu'aux bornes qui sont des également des bornes de réparations utiles au robot.

Illustration : On représente le robot par une balise rouge.



Remarque : la **disposition des sommets** dans **l'espace** et la taille des **arêtes** n'a aucune **importance**. Par exemple les deux graphes ci-dessous sont équivalents.



1.3 Problématique

Localisation : Avec seulement **la distance à chaque borne**, le robot peut-il savoir **avec certitude où il se trouve ?**

Optimisation : Comment **placer les bornes** pour que **le robot sache toujours où il se trouve** tout en mettant le moins possible ? Est-ce **toujours possible d'optimiser ?**

1.4 Plan

Ce sujet est un sujet de recherche actuel qui n'a pas été résolu pour n'importe quel graphe. Pour répondre à la problématique nous avons donc séparé notre sujet en deux parties. Tout d'abord nous verrons le cas général qui permet d'optimiser **partiellement** la résolution de tous les graphes puis quelques cas spéciaux qui sont optimisés au maximum.

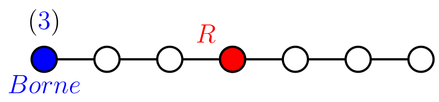
2 Des outils pour mieux comprendre

Avant de commencer à répondre à la problématique nous allons définir des outils que nous utiliserons tout le long de l'article pour faciliter la compréhension des différents cas.

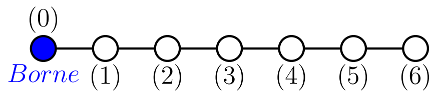
2.1 Les coordonnées

Comme vu précédemment, quand le robot se situe sur un sommet du graphe, les bornes indiquent une distance sous forme de nombre entier. Un sommet possède donc des coordonnées définies par les différentes bornes sur le graphe. Ces coordonnées sont définies avec un ordre correspondant à l'ordre des bornes, une borne est donc toujours associée à un rang compris entre : 1 et le nombre de bornes.

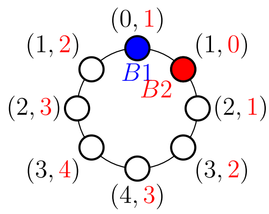
Exemple : Ce graphe, correspondant à une configuration possible



devient celui là, regroupant au sein du même graphe toutes les configurations possibles.



Exemple : Avec deux bornes :



Le robot sait se localiser car les coordonnées (a,b) et (b,a), où a et b sont des nombres entiers, ne sont pas identiques. L'écriture des coordonnées respecte l'ordre prédéfini des bornes.

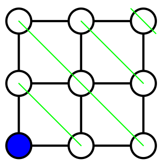
Théorème : Tous les sommets du graphe ont des coordonnées différentes si et seulement si le robot sait se localiser quel que soit le sommet où il se situe

Démonstration : Tous les sommets du graphe ont des coordonnées différentes si et seulement si chaque sommet possède une combinaison de distance à chaque borne propre à lui-même. Si chaque sommet possède une combinaison de distance à chaque borne propre à lui-même alors chaque sommet est identifiable juste en connaissant la combinaison qui lui est associée or le robot possède un plan du graphe et les distances à chaque borne donc il est capable de se localiser quelque soit le sommet où il se situe.
Réciproquement, si le le robot sait se localiser quelque soit le sommet où il se situe alors quelque soit le sommet où il se situe ce sommet aura une combinaison de distances aux bornes identiques à aucun autre sommet, donc une combinaison propre à lui-même. Et donc cela veut dire que tous les sommets du graphe ont des coordonnées différentes.

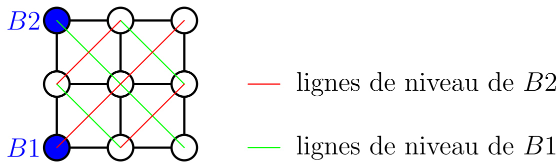
2.2 La ligne de niveau

Définition : On appelle ligne de niveau, une ligne où tous les sommets présents sur cette ligne sont à la même distance d'un point de référence (ex : une borne).

Exemple :



Exemple : Avec deux bornes :



Le robot sait se localiser car des mêmes lignes de niveau ne se croisent jamais plus d'une fois.

Remarques : Un sommet est traversé par autant de lignes de niveau qu'il y a de bornes sur le graphe.
Des lignes de niveau ne se coupent pas nécessairement.
Les lignes de niveau ne sont pas nécessairement des droites, des mêmes lignes de niveau peuvent se couper plusieurs fois.

Théorème : Chaque combinaison de lignes de niveau concourantes sont concourantes en un sommet unique si et seulement si le robot sait se localiser quel que soit le sommet où il se situe.

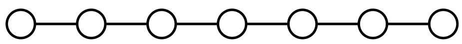
Démonstration : La ligne de niveau est une autre représentation des coordonnées à chaque sommet. Dire que chaque combinaison de lignes de niveau concourantes sont concourantes en un sommet unique équivaut à dire que chaque combinaison de coordonnées que l'on trouve sur un sommet, on ne peut la trouver que sur ce sommet là. Ce qui équivaut à dire que chaque sommet possède une combinaison de distance à chaque borne propre à lui-même et donc d'après la première démonstration, équivaut à dire que tous les sommets du graphe ont des coordonnées différentes et enfin cela équivaut à dire que le robot sait se localiser quel que soit le sommet où il se situe.

3 Cas général

Soit n un entier naturel, n représente le nombre de sommets du graphe (on gardera cette notation pour toute la suite de l'article)

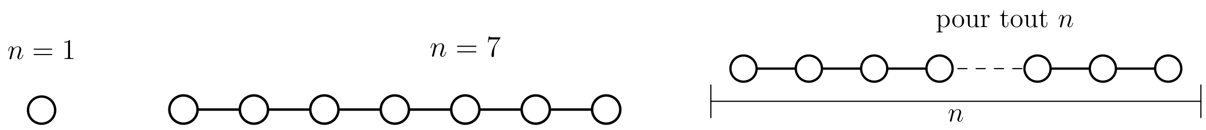
3.1 La Ligne

Définition : On appelle une ligne un graphe de la forme :



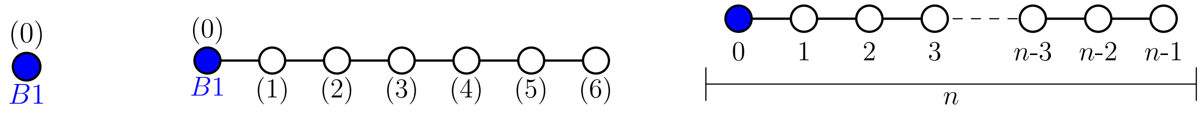
Une ligne est définie pour $n \geq 1$

Exemples :



Propriété : Il suffit d'une balise pour localiser le robot à condition de la placer à une extrémité.

Illustrations :

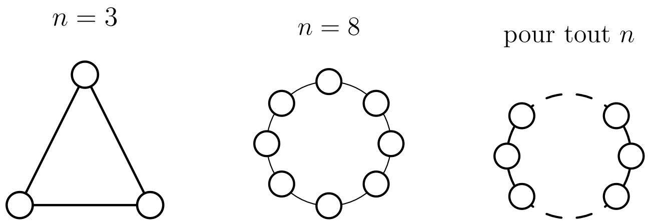


Démonstration : Les sommets sont numérotés de 1 à n .
Les distances possibles sont de 0 à $(n-1)$.
Dans un graphe à 1 sommet, le robot sait qu'il se situe à 1-1 arête de la borne ce qui est égale à 0. Donc sur un graphe à n sommets, du fait que la borne soit à l'extrémité, toutes les distances seront différentes, il pourra donc se repérer partout sur ce graphe. Si le graphe passe à $(n+1)$ sommets, comme le robot savait se repérer sur n sommets il le pourra également sur $(n+1)$ sommets grâce au fait que toutes les distances seront différentes 0, 1, 2, 3, 4, [...] ,$(n-1)$, n

3.2 La boucle

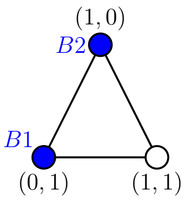
Définition : On appelle une boucle, une ligne dont les extrémités sont reliées par une arête.
Une boucle est définie pour $n \geq 3$.

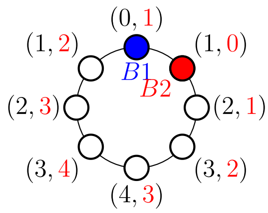
Exemples :

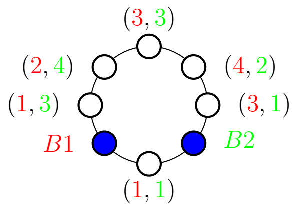


Propriété : Il suffit de deux balises placées n'importe où sauf dans un cas précis, dans une boucle avec un nombre pair de sommets il ne faut pas les placées en "opposition", c'est à dire qu'il ne faut pas placer la deuxième borne au sommet le plus éloigné du sommet où se trouve la première borne.

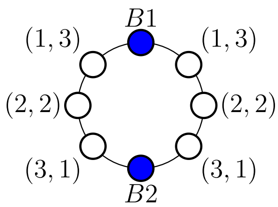
Illustrations :

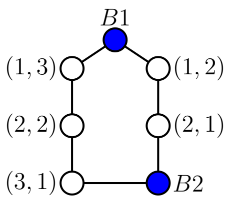




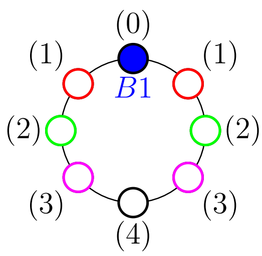


Nous pouvons remarquer que dans une boucle avec un nombre pair de sommets quand les deux bornes sont en opposition on ne peux localiser le robot. Ce problème est absent dans les boucles avec un nombre impair de sommets car il n'y a pas une extrémité mais deux.





Démonstration : Tout d'abord, quand nous plaçons une borne, l'issue est toujours la même :



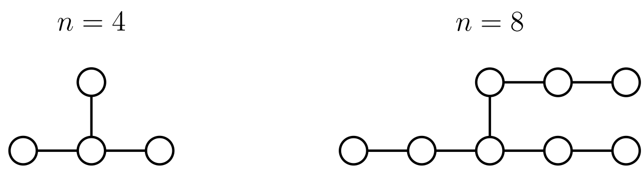
Le robot ne peut se localiser. Cela veut dire que nous pouvons placer la première borne où l'on veut et qu'il faudra au moins deux bornes pour localiser le robot. Nous pouvons remarquer également qu'après avoir placé une borne, nous obtenons deux lignes qui partent de la borne pour soit se relier par une arête dans les boucles impaires soit par un sommet dans les boucles paires. Pour localiser le robot, il suffit donc de savoir sur quelle ligne il se situe. Pour cela, nous pouvons placer une borne n'importe où sur une de ces lignes, c'est à dire sur un sommet autre que celui en opposition par rapport à la première borne. En effet, les sommets qui possèdent les mêmes coordonnées forment des "couples" et en plaçant une borne n'importe où, celle ci va donner une coordonnée supplémentaire à chaque sommet. Cette coordonnée va permettre de localiser le robot car tous les sommets de la ligne où cette deuxième borne se situe ont un nombre d'arêtes à cette borne plus petit d'au moins une arête que leur couple respectif sur l'autre ligne, du au fait justement que ces deux lignes sont séparées par au moins une arêtes, la deuxième coordonnée est donc différente pour chaque couple. Cela veut dire que tous les sommets qui avaient des coordonnées identiques ont maintenant des coordonnées différentes et donc que tous les sommets ont des coordonnées différentes et donc que le robot sait se localiser.

Remarque : La ligne et la boucle sont des cas spéciaux. Nous les avons placées là car elles permettent d'introduire le cas général que nous allons présenter après.

3.3 La ramification de ligne :

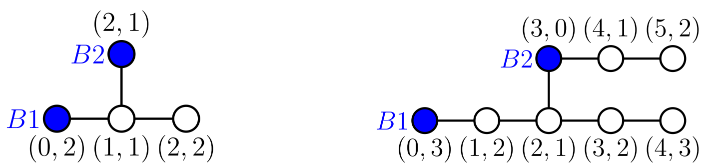
Définition : On appelle une ramification de ligne, une ligne dont une des extrémités est reliée par une arête à un sommet d'une autre ligne dite "ligne principale" ou "ligne de référence". La ramification est reliée à n'importe quel sommet de la principale à part une des extrémités car dans ce cas les deux lignes n'en forme qu'une seule. La ligne principale doit donc être composée d'au moins 3 sommets.
Un graphe avec une ramification de ligne est définie pour $n \geq 4$.

Exemples :



Propriété : Il suffit d'une borne sur la ramification en plus de celle sur la ligne principale. Ces bornes sont placées aux extrémités de leur ligne respective pour respecter la première propriété sur les lignes.

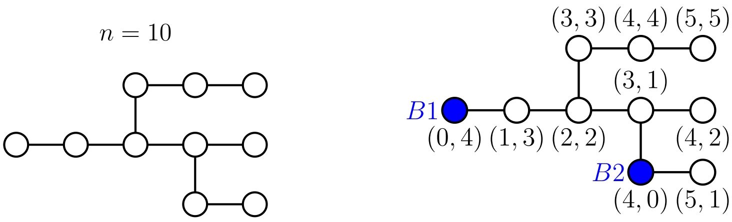
Illustrations :



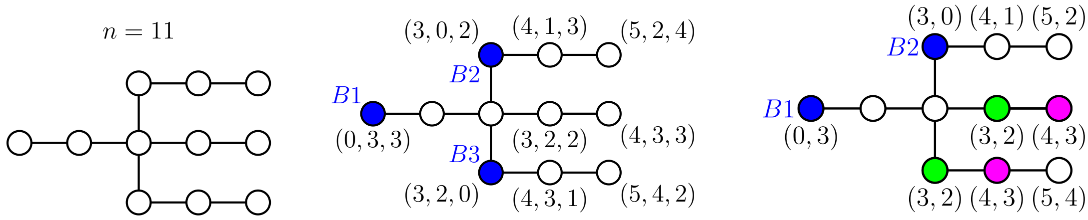
Démonstration : Tout d'abord, il faut une borne à une extrémité d'une ligne pour résoudre la ligne principale, conformément à la première propriété sur les lignes. A partir du sommet d'intersection, la ramification est comme une deuxième ligne qui fait apparaître des couples de sommets à partir du point d'intersection. Ensuite, le raisonnement pour résoudre cette situation est proche de celui sur la boucle, le but est de savoir sur quelle ligne se situe le robot. Pour cela, il suffit d'une borne en plus placée n'importe où sur une des lignes. La deuxième coordonnée des sommets sur la ligne où se trouve cette deuxième borne va être plus petite d'au moins une arête que celle sur l'autre ligne du au fait que la ramification est à une arête de la ligne principale [1]. Dans la ramification de ligne, les sommets qui relient les deux lignes sont ceux entre l'intersection est la première borne, c'est pourquoi il ne faut pas placer la deuxième borne sur ces sommets. Enfin, pour simplifier la résolution, nous considérons qu'il faut placer la borne sur une extrémité de la ramification mais cela n'a d'importance que dans les cas avec plusieurs ramifications comme ceux que nous allons présenter après.

Remarque : Nous allons étendre cette propriété à un graphe avec plusieurs ramifications. En effet, le robot se trouvera toujours sur une ramification avec une ligne de référence et donc pourra être localisé. Cependant l'optimisation est valide pour une seule ramification c'est pourquoi nous considérons qu'avec plusieurs ramifications la propriété est une approximation de l'optimisation totale. Dans certains cas nous pouvons encore plus optimiser, dans d'autres on ne peut pas.

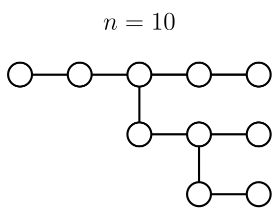
Exemples : Avec ce graphe deux bornes suffisent



Dans celui là, 3 bornes sont nécessaires



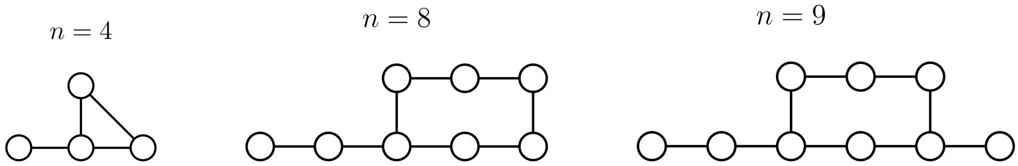
Remarque : Une ramification peut également servir de référence à une autre ramification. Dans ce cas, il est important de bien placer les bornes aux extrémités.



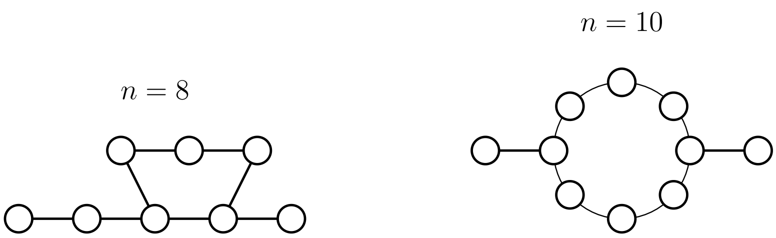
3.4 La ramification de ligne bouclée :

Définition : On appelle ramification de ligne bouclée, une ramification de ligne dont l'extrémité, qui n'est normalement pas reliée à ligne de référence, est ici reliée par une arrête à un sommet de cette ligne de référence autre que celui dont la première extrémité est déjà reliée. Cette extrémité de la ramification peut être reliée à une extrémité de ligne de référence.
Comme pour la ramification de ligne, la ramification de ligne bouclée est définie pour $n \geq 4$.

Exemples :

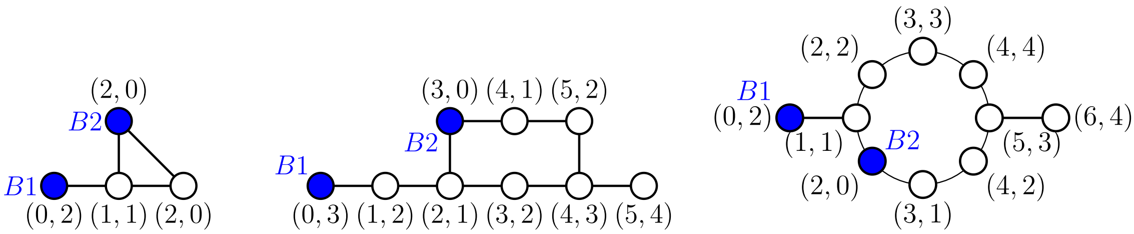


Autres configurations possibles :

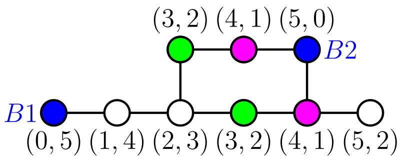


Propriété : Comme pour la ramification de ligne, il suffit d'une borne sur la ramification en plus de celle sur la ligne principale. Ces bornes sont placées aux extrémités de leur ligne respective pour respecter la première propriété sur les lignes avec une condition en plus, la balise sur la ramification ne doit pas être placée en opposition.

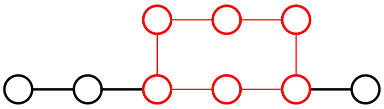
Illustrations :

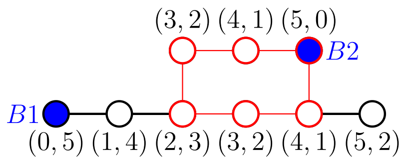


Dans ces cas nous pouvons localiser le robot car la dernière condition est respectée.
Dans ce cas suivant elle ne l'est pas, on ne peut donc pas localiser le robot.

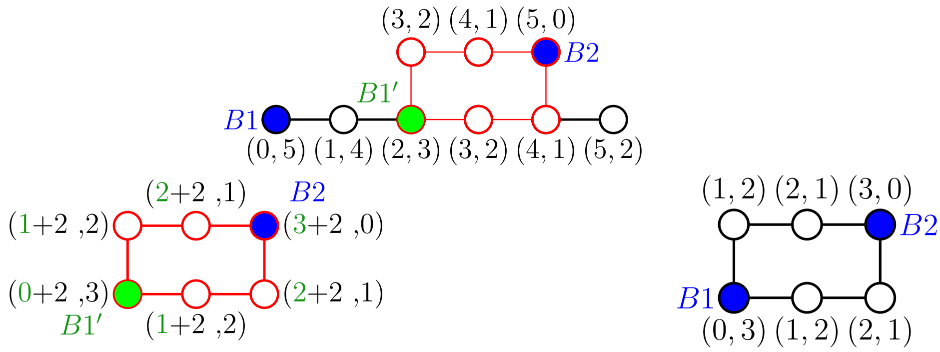


Explication : Nous allons appeler la boucle de ramification la partie du graphe suivante (représenter en rouge) :





Nous constatons que cette partie forme une boucle. L'information venant de la borne située sur la ligne principale est, pour la boucle de ramification, identique si on déplace la borne sur le sommet d'intersection puisqu'on ne fait qu'enlever la même distance à chaque sommet, ce qui n'empêche pas de comparer et de savoir quel sommet est à la même distance de quel sommet.

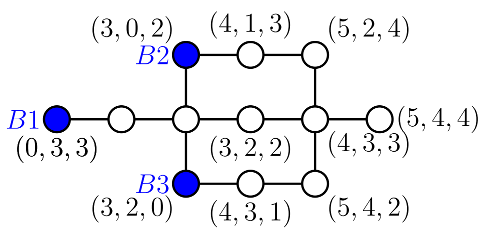


Pour localiser le robot sur cette boucle, il va donc falloir placer les bornes comme sur une boucle normale, et donc ne pas les placer en opposition.

Démonstration : Le raisonnement est proche de celui sur la ramification de ligne. Il faut une borne sur la ligne principale et une sur la ramification pour différencier les différents couples de coordonnées. Cependant, il faut faire attention à ne pas placer la borne en opposition du sommet d'intersection comme indiqué ci-dessus.

Remarque : Comme pour la ramification de ligne, nous allons étendre cette propriété à un graphe avec plusieurs ramifications en considérant l'optimisation comme partielle.

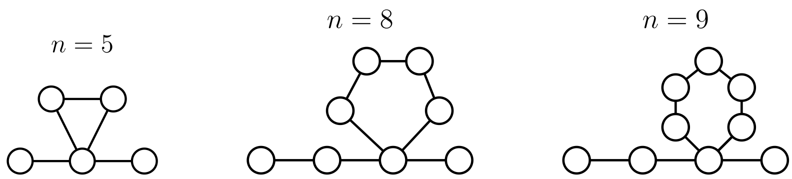
Exemple :



3.5 La ramification de boucle :

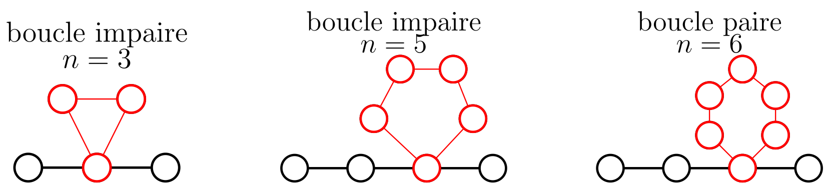
Définition : On appelle ramification de boucle, une ramification de ligne bouclée dont les deux extrémités de la ramification sont reliées au même sommet sur la ligne principale. La ramification ne doit pas être reliée à une extremité de la ligne principale car sinon on se retrouve dans la situation de ligne bouclée, ce qui veut dire que ligne principale doit être composée d'au moins 3 sommets . La ramification doit l'être d'au moins 2 sommets.
Un graphe avec une ramification de boucle est définie pour $n \geq 5$.

Exemples :



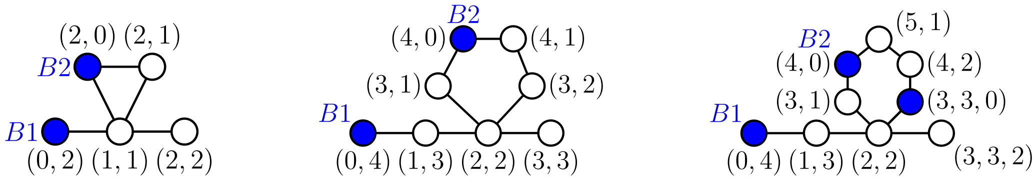
Comme pour la ramification de ligne bouclée, on appelle "boucle de ramification" la partie en rouge. Cette définition est importante car la propriété va dépendre du nombre de sommets de cette boucle de ramification.

Exemples :



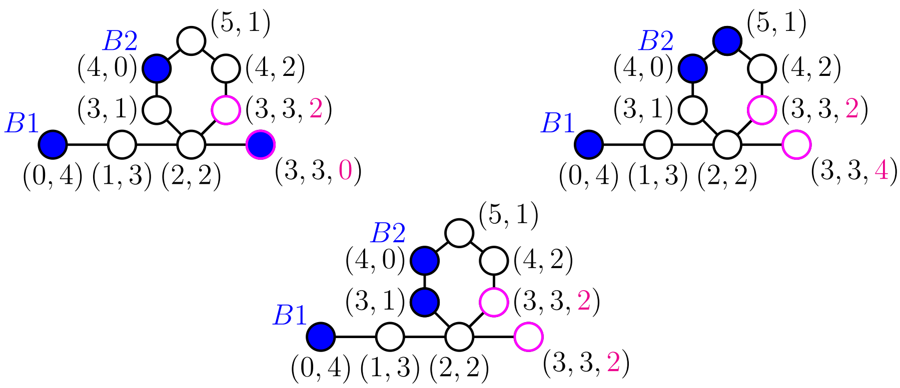
Propriété : Pour localiser le robot, il va falloir une borne à une extrémité de la ligne de référence, ensuite : si la boucle de ramification est impaire il suffira d'une borne placée à l'opposé du point d'intersection avec la ligne de référence, si la boucle de ramification est paire il suffira et sera nécessaire de placer deux bornes dont une sur la ramification et l'autre en opposition du sommet d'intersection ou sur un autre sommet comme indiqué après.

Illustrations :



Dans le troisième cas, la troisième balise sert juste à différencier deux sommets, c'est pourquoi pour alléger l'illustration nous n'avons placé la troisième coordonnée que juste pour ces deux sommets, sa localisation a une importance mais il y a plusieurs possibilités. Pour simplifier nous considérons qu'il faut la placer sur la ramification de boucle à l'opposé du point d'intersection ou directement sur un des sommets concernés alors que sur d'autres sommets ce n'est pas une bonne idée.

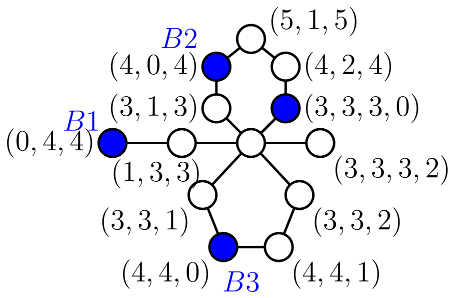
Exemples :



Démonstration : La ramification de boucle est différente des ramification précédentes car il part trois lignes du point d'intersection. Encore une fois, nous allons placer une borne à une extrémité de la ligne principale. Puisque la ramification est une boucle, on sait qu'avec deux bornes tous les sommets peuvent être différenciés, ce qui veut dire qu'il suffit d'une borne sur la ramification en plus de celle sur la ligne principale car celle ci ce comporte comme une borne sur le sommet d'intersection. Pour les bornes impaires, il n'y a pas de problèmes mais pour les boucles paires il ne faut placer la borne en opposition du sommet d'intersection, ce qui revient à respecter les propriétés sur les boucles. Le problème est celui de la suite des sommets de la ligne principale après le sommet d'intersection, qui ne sont pas différenciés si on place la borne sur la ramification n'importe où. Cela est du au fait que ci la borne sur la ramification est trop près du sommet d'intersection alors le nombre d'arêtes avec la suite de la ligne est aussi grand voire plus petit que certain sommets de la ramification elle même. Nous nous retrouvons donc avec encore plusieurs sommets indifférenciés. On sait que dans une boucle les sommets prennent une valeur extrême délimitée par le nombre de sommets de la boucle, dans une boucle paire l'extrémité est $n/2$ et dans les boucle impaires, les deux extrémités sont à $n/2$-1. Si un sommet est relié à une de ces extrémités, il va être directement différencié car il va prendre une valeur supérieures à toutes celles de la boucle. Dans notre ramification, ce sommet va être différencié si on place la borne de la ramification à une des extrémités par rapport au sommet d'intersection. Ce problème persiste dans la ramification de boucle paire puisque si on place la borne à l'extrémité, se sont les sommets de la ramification qui seront indifférenciés. Pour la boucle paire, il va donc falloir au moins deux bornes en plus de celle sur la ligne, une pour différencier les sommets de la ramification et une pour différencier les sommets avec la suite de la ligne.

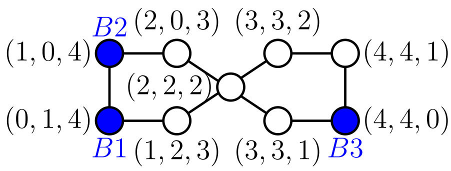
Remarque : Comme pour la ramification de ligne, nous allons étendre cette propriété à un graphe avec plusieurs ramifications en considérant l'optimisation comme partielle.

Exemple :



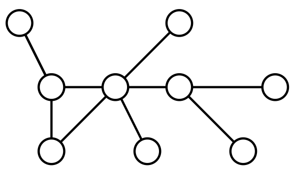
Remarque : Nous venons de voir les trois types de ramifications possible et à chaque fois nous avons pris une ligne comme figure de référence cependant, ces propriétés peuvent s'étendre à n'importe qu'elle figure de référence, que se soit un cas spécial comme une boucle ou n'importe quoi d'autre, du moment qu'elle est de base résolue. Le problème reste encore l'optimisation mais la localisation du robot n'en est pas moins compromise.

Exemple :



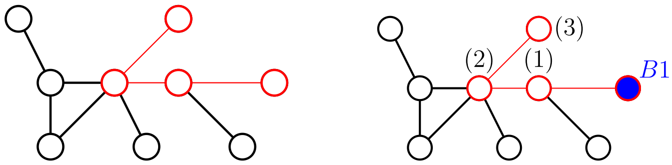
3.6 La méthode de résolution :

Nous considérons que chaque graphe, quelque soit sa forme, peut être décomposé de la façon suivante : une ligne ou une boucle, servant de figure de référence, suivit de ramifications selon les trois types que nous avons vu. La méthode de résolution de n'importe quel graphe va se diviser en trois étapes que nous détaillerons à l'aide d'un exemple :

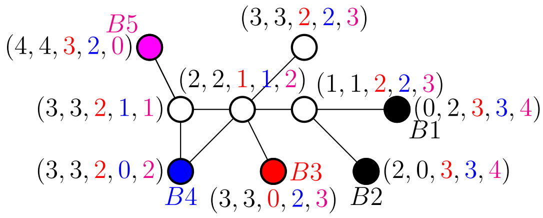


Ce graphe est le tout premier que nous avons présenté, pour que la boucle soit bouclée, nous allons le résoudre.

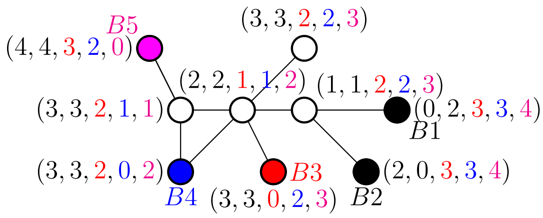
Étape 1 : Il faut choisir une figure de référence. Cette figure doit être une ligne ou une boucle mais il n'y a pas d'importance concernant sa position (il n'y a pas besoin de prendre la plus grand ligne par exemple). Ensuite il faut la résoudre. Nous avons choisi la ligne représenté en rouge.



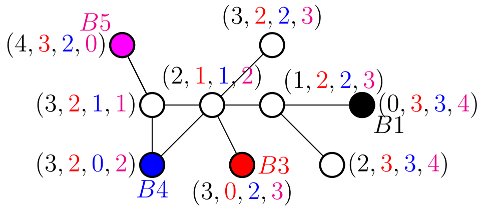
Étape 2 : Cette étape consiste à placer les bornes sur les ramifications en suivant les propriétés sur les différents types de ramifications sans se soucier de l'optimisation. Sur ce graphe nous avons deux ramifications de ligne à n=1 sommet reliées à la figure de référence, donc deux bornes en plus, puis une ramification de boucle impaire à n=3 sommets, donc une borne en plus, et enfin une ramification de ligne reliée à la boucle, et donc une borne en plus. Encore une fois, plusieurs interprétations sont possibles. Ce que l'on considère une boucle avec une ligne peut être une ligne avec une ligne bouclée, l'important est de choisir une figure et de bien suivre les propriétés concernant cette figure. Voilà le résultat : (nous avons pris des couleurs différentes pour les bornes pour la lisibilité)



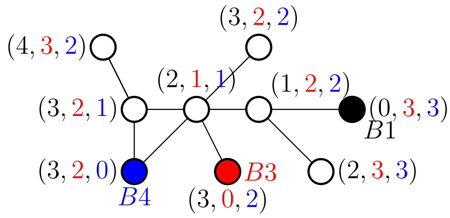
Étape 3 : La dernière étape consiste à optimiser un maximum. cette optimisation va être faite "à la main" et sera par conséquent imparfaite. C'est pour cela que nous considérons notre méthode comme une résolution partielle, qui permet cependant de traiter tous les graphes.
Pour faire cette optimisation, il faut regarder borne par borne si elle apporte une information utile à la localisation du robot. Attention, certaines bornes servent juste à différencier deux sommets mais elles restent nécessaires.



Pour commencer, on considère la borne 1 comme nécessaire puisqu'on nous avons vu au début qu'elle permettait déjà de différencier certains sommets. Quand on regarde la borne 2 on se rend compte qu'elle donne les mêmes informations que la borne 1 pour tous les sommets sauf pour deux, celui où se situe la B1 et celui ou se situe la B2. La borne 2 ne sert donc qu'à différencier ces deux sommets sauf que la borne 1 se situe déjà sur l'un de ces sommets et permet déjà de les différencier. La borne 2 est donc inutile.

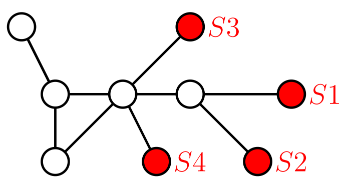


Ensuite, si on regarde la borne 5, elle n'apporte aucune information utile car avec juste les trois autres nous pouvons différencier tous les sommets. La borne 5 est donc inutile. Quand aux trois autres nous ne pouvons pas les enlever sinon il reste des sommets indifférenciés.

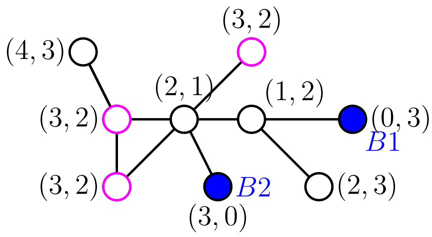


En conclusion, nous venons de démontrer que trois bornes suffisaient pour résoudre ce graphe et permettre au robot de se localiser. Cependant, nous n'avons pas démontré que trois bornes étaient nécessaires, peut-être qu'avec un autre placement des bornes nous pouvons en placer moins. c'est là, la limite de notre méthode. La réponse à notre problématique étant impossible, cette méthode est une approximation de la réponse.

Vérification : Pour des graphes simples comme celui de notre exemple, nous pouvons essayer de vérifier si le nombre de bornes trouvées est bien le minimum. Il n'y a pas de méthode pour la vérification [2] , celle que l'on va utiliser est propre à notre exemple mais peut être réutilisée dans des cas assez similaires.
Tout d'abord, donnons des noms aux sommets :



Puis, on constate que les sommets 1 et 2 sont à la même distance de tous les autres sommets du graphe. C'est à dire que n'importe quelle borne placée sur autre sommet que ces deux là ne pourra différencier ces deux sommets si le robot se trouve sur l'un d'eux, elle affichera le même nombre. Il faut donc une borne placée sur l'un de ces deux sommets pour pouvoir les différencier. Nous pouvons répéter ce raisonnement aux sommets 3 et 4. Nous savons donc que deux bornes minimum sont nécessaires.



Avec ces deux bornes placées, il reste des sommets indifférenciés. Nous pouvons en conclure qu'il faut au moins trois bornes pour localiser le robot. Or avec trois bornes nous pouvons localiser le robot. Donc trois bornes sont nécessaires et suffisantes pour localiser le robot. Nous venons de démontrer le résultat.

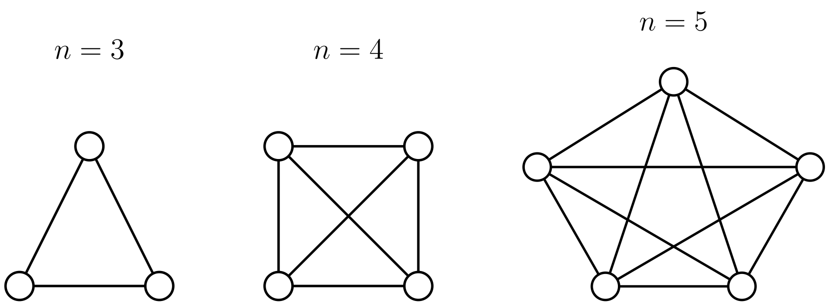
4 Cas spéciaux

Nous allons voir maintenant quelques graphes dont la méthode de résolution est totalement différente.

4.1 Le graphe complet

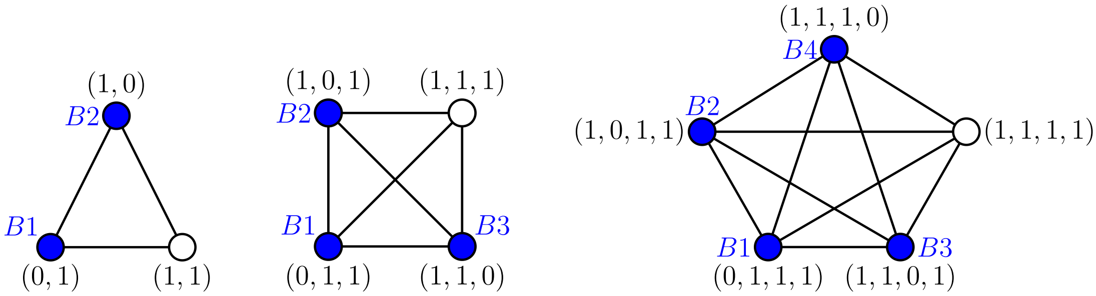
Définition : On appelle graphe complet, un graphe dont chaque sommet est relié par une arête à tous les autres sommets.
Nous choisissons de définir un graphe complet à partir de $n \geq 3$.

Exemples :



Propriété : Il faut $n-1$ bornes minimum pour localiser le robot. C'est à dire une borne sur chaque sommet sauf un.

Illustrations :

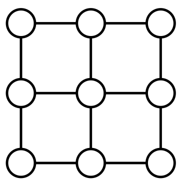


Démonstration : Dans un graphe complet, tous les sommets sont à 1 de tous les autres sommets. Quand on place une borne, elle affichera 0 si le robot se situe dessus ou 1 quel que soit l'autre sommet. Si on rajoute des bornes, on ne pourra pas différencier les sommets où il n'y en a pas car leurs coordonnées seront identiques, c'est à dire une succession de 1 correspondant au nombre de bornes. Donc, dans un graphe complet, une borne ne sert à différencier que le sommet où elle se situe. Pour différencier tous les sommets, il faut donc placer des bornes sur tous les sommets. Cependant, on peut en enlever une, car si toutes les bornes affichent 1 alors le robot se trouve forcément sur le seul sommet où il n'y a pas de bornes.

Remarque : Ce type de graphe est en quelque sorte le "pire" type de graphe, c'est à dire que c'est le type de graphe qui nécessite le nombre le plus important de bornes par rapport au nombre de sommets. Parallèlement, la ligne est le "meilleur" type de graphe, c'est le graphe qui nécessite le moins de bornes par rapport au nombre de sommets puisqu'une seule borne suffit pour un nombre n de sommets aussi grand que l'on veut.

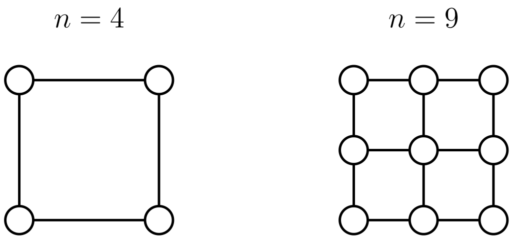
4.2 Le quadrillage

Définition : On appelle quadrillage, un graphe de la forme :

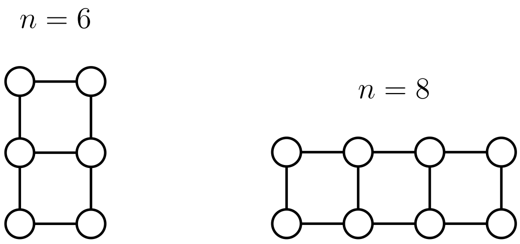


Un quadrillage est définie pour $n \geq 4$.

Exemples :

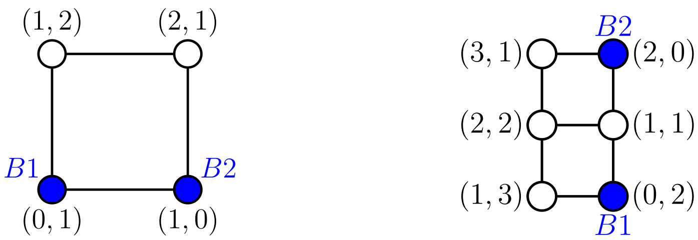


Les quadrillages peuvent avoir des formes rectangulaires :

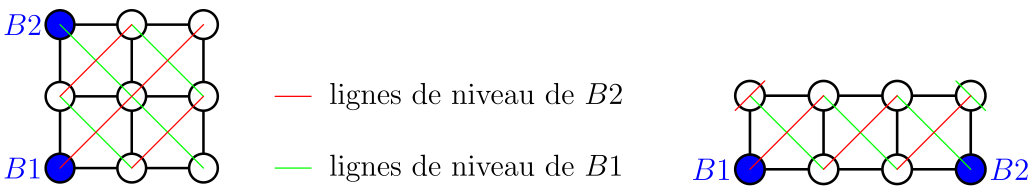


Propriété : Il suffit de deux bornes placées aux extrémité d'un même coté pour localiser le robot.

Illustrations :



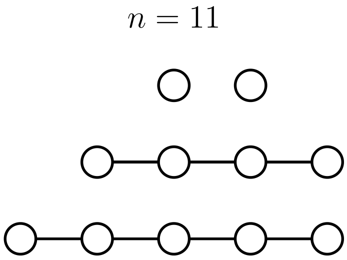
Dans ce cas, la représentation avec des lignes de niveaux est plus parlante :



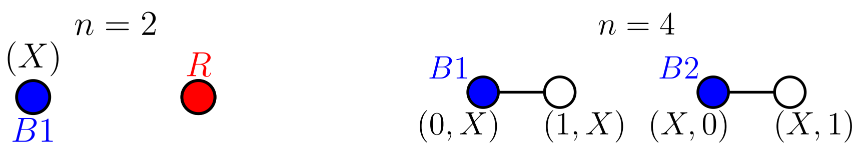
Démonstration : Dans un quadrillage, quand nous plaçons les bornes sur les extrémités d'une même ligne, toutes les lignes de niveau vont former des droites. Or on sait que deux droites ne peuvent être sécantes qu'en un point. Donc cela veut dire que deux lignes de niveau ne peuvent se rencontrer qu'en un seul sommet et donc que chaque combinaison de lignes de niveau concourantes sont concourantes en un sommet unique et donc d'après le théorème sur les lignes de niveau, le robot sait se localiser quel que soit le sommet où il se situe.

4.3 Le graphe non-connexe

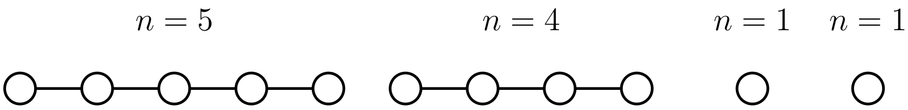
Définition d'un graphe connexe : On appelle un graphe connexe, un graphe dont deux sommets quelconques peuvent être reliées par une chaînes ou, dans notre sujet, une succession d'arêtes. Pour l'instant tous les graphes que nous avons présentés sont des graphes connexes mais il existe aussi des graphes non-connexes comme celui ci :



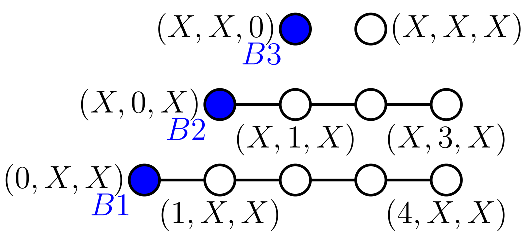
Dans l'optique de traiter tous les graphes, nous allons également traiter celui des graphes non-connexes.
Tout d'abord, nous allons définir une nouvelle notation. Une borne donne un nombre d'arêtes, celui qui la sépare d'un sommet donné. Entre deux sommets qui ne sont pas reliés par une succession d'arêtes, la borne n'est pas en mesure de donner un nombre. Dans notre sujet, nous considérons que la borne est alors brouillée et qu'elle envoie un message d'erreur. ce message d'erreur est modélisé par un "X".



Ensuite, nous allons décomposé notre graphe en parties correspondant à des graphes connexes.



L'information qu'envoie une borne a donc deux utilités : permettre de résoudre la partie où elle se situe, et indiquer si le robot se situe sur cette partie ou sur une autre partie. Une borne ne peux différencier deux sommets autre que la partie où elle se situe.
Enfin, pour résoudre un graphe un graphe non-connexe, il faut donc résoudre chaque partie de façon indépendante. Nous avons décidé de parler de ce type de graphe à la fin car il nécessite de revenir sur tout ce que nous venons de voir.



Remarque : Dans l'optique d'optimiser au maximum, nous pouvons remarquer une petite subtilité. Si une partie du graphe non-connexe est un graphe à un seul sommet, alors il n'y a pas besoin de placer de borne dessus. En effet, si toutes les bornes indiquent que le robot ne se situe pas sur leur partie, alors le robot ne peut se situer que sur la partie où il n'y a pas de bornes et puisqu'il n'y qu'un seul sommet, le robot se situe forcément sur ce sommet. Cela ne fonctionne bien sûr que pour un seul graphe à un sommet dans un graphe non-connexe.

5 Conclusion :

En conclusion, pour répondre à la problématique nous avons vraiment suivi deux voies différentes, l'une consistait à trouver une méthode générale mais approximative et l'autre consistait à trouver des cas particuliers intéressants. Nous avons beaucoup insisté sur la méthode générale et nous pensons que c'est sur la recherche de cas particuliers que nous pourrions continuer à chercher.

Pour finir, Math-en-Jeans est une aventure qui nous aura montré que le plus important dans le travail de groupe est la cohésion et l'entente et que sans ça, il est très difficile voire impossible d'avancer.

6 Notes d'édition

[1] Cette phrase est un peu obscure ...

[2] Aux auteurs : dire plutôt que vous n'en avez pas trouvée.