

Recherche décentralisée de connexité pour réseaux de capteurs mobiles

Merwan Achibet

1 Introduction

On imagine le problème suivant : un groupe de n capteurs mobiles est réparti aléatoirement dans un espace aérien. On part de l'hypothèse qu'un capteur connaît uniquement le nombre total de capteurs du système et qu'il est assez sophistiqué pour pouvoir déterminer précisément sa position absolue. Les capteurs sont dotés de matériel de communication sans fil et peuvent s'envoyer des messages à condition que la distance les séparant soit inférieure à leur rayon d'émission R_c .

Le réseau constitué par cet essaim d'appareils volants forme un graphe dynamique dont les nœuds sont les capteurs. Deux nœuds sont reliés par un arc si les capteurs qui leur sont associés sont en mesure de communiquer, c'est à dire s'ils sont assez proches. La figure 1 décrit un exemple de scénario impliquant trois capteurs.

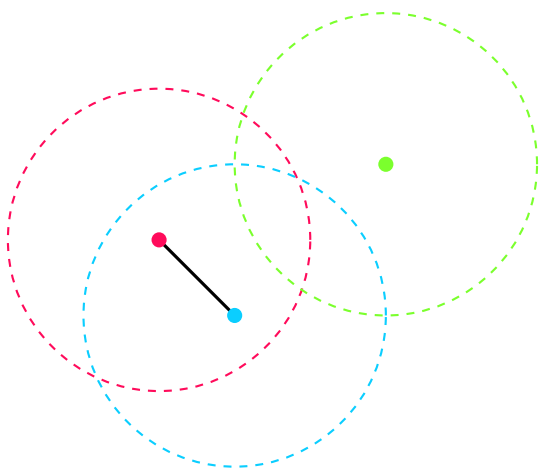


FIGURE 1 – Les capteurs rose et bleu peuvent communiquer et sont connectés tandis que le capteur vert est isolé.

On considère un capteur comme un agent autonome capable de se mouvoir dans l'espace. Afin de ne pas se préoccuper de considérations géométriques superflues, il est supposé qu'un capteur conserve toujours la même altitude et donc on limite son déplacement à deux dimensions. La contrainte principale de cet exercice est que l'on refuse toute forme de contrôle global sur l'essaim de capteurs. Toutes les actions entreprises par un appareil seront uniquement dû à ses décisions propres et dépendront de la vue réduite du système dont il dispose.

La possibilité pour un capteur de communiquer avec ses semblables est au centre de nos préoccupations car on considère qu'un capteur isolé est inutile puisqu'incapable de partager des données. Dans le contexte de l'étude décentralisée d'un graphe dynamique, deux questions se posent :

1. Comment déterminer si le réseau de capteurs est connexe ?
2. Comment déplacer les capteurs de façon à ce qu'il le devienne ?

Pour répondre à la première question, on se concentre sur les communications de capteur à capteur en proposant une méthode pour laquelle chaque agent diffuse sa vision de la connexité du réseau, étiquetée en fonction de l'origine de l'information et de sa date. Un mécanisme de filtrage autoréglant sera mis en place pour répandre rapidement les informations critiques telles que l'ajout d'un nouveau nœud à une composante connexe.

La réponse à la seconde question s'attache à l'aspect mobile d'un capteur. On propose une méthode de guidage décentralisé se basant sur l'imitation de plusieurs lois physiques de la nature afin de former un maillage à la fois connexe, équilibré et étalé dans l'espace. Finalement, on envisage une extension de ce système pour permettre aux capteurs d'éviter

naturellement les obstacles.

2 Déterminer si le réseau est connexe

Dans cette section, on se place du point de vue d'un capteur c et l'on considère que le réseau est connexe lorsque c le pense connexe.

Les capteurs étant mobiles, le réseau de communication qu'ils forment est hautement dynamique et des événements imprévisibles, comme le déplacement d'un capteur ou une panne, peuvent à tout moment faire varier la connexité du graphe résultant. Comme énoncé précédemment, c a connaissance du nombre de capteurs n du système. Déterminer la connexité du réseau revient donc à un problème de comptage décentralisé, où \tilde{n} est le nombre de capteurs que c sait connectés à sa propre composante connexe et le graphe est connexe si $\tilde{n} = n$.

La méthode que l'on présente repose sur un partage régulier, entre les capteurs, des informations qu'ils entretiennent à propos de leur voisinage direct. Si l'on pouvait traduire en langue humaine un message de c à c' , on entendrait :

Je suis c et je sais de façon certaine que x et y sont dans mon voisinage. Par contre, je n'ai pas eu de nouvelles de z depuis un certain temps, je doute qu'ils soit toujours présent dans le réseau

L'objectif de notre méthode est d'autoriser la diffusion de tels messages tout en gérant les conflits potentiels d'information et en répercutant sur le système les informations valides uniquement. Pour ce faire, on introduit trois mécanismes : la surveillance, le balisage et le filtrage.

Surveillance

Dans notre système, on suppose que c dispose d'une mémoire interne suffisante pour stocker plusieurs listes. La première représente le voisinage de c et contient uniquement les identifiants des nœuds en liaison directe avec ce dernier. Un nœud est ajouté à cette liste si c en reçoit un message et qu'il n'était pas déjà dans la liste. Un nœud est retiré de cette liste si c n'en a pas reçu de message depuis un certain temps, c'est le mécanisme de surveillance.

On choisit ici de rendre les capteurs proactifs : leurs messages sont envoyés à intervalle régulier afin d'informer de leur présence et l'on considère qu'un capteur a disparu s'il n'envoie plus de message.

Balisage

La seconde liste entretenue par un capteur représente sa vision de la connexité du réseau, c'est à dire tous les nœuds qui y sont connectés de son point de vue. Chacune de ses entrées est un triplet (x, t, s) avec x l'identifiant du nœud que l'on sait connecté, t le dernier temps auquel x a été détecté et s le nœud à l'origine de cette information.

Par exemple, si c a dans sa liste de connexité l'entrée $(y, 52, z)$, cela signifie qu'il sait que le capteur y fait partie de la même composante connexe car il a été détecté au temps 52 et l'information a été relayée par z . Le nœud à l'origine de l'information, ici z , apparaît nécessairement dans la liste de voisins de c puisque seuls les voisins peuvent échanger des messages. Il est important de différencier ce nœud du nœud distant qui a détecté y au temps 52 (et dont on ne se préoccupe pas de l'identité).

La troisième liste contient des données à propos des capteurs dont c n'est plus certain de la connexité. Un nœud c' y est ajouté lorsqu'il est présent dans la liste de connexité et que, par surveillance, on se rend compte qu'il n'a pas communiqué depuis un certain temps. À l'inverse, c' migre de la liste de doutes vers la liste de connexité si une nouvelle communication est plus tard détectée. Il est évident qu'un nœud ne peut être présent que dans une des deux listes à la fois et ce constat nous permet d'évaluer l'espace mémoire nécessaire à $2n$ informations (n informations dans la liste de voisins si le graphe est complet, n informations réparties dans la liste de connexité et la liste de doutes au maximum).

Le balisage est l'action, pour un capteur, de régulièrement partager sa liste de connexité et sa liste de doutes avec ses voisins.

Évaluation des rumeurs

3 Rendre le réseau connexe

La résolution de ce problème passe par la satisfaction de deux besoins a priori antagonistes. D'une part, il est naturellement nécessaire de réunir les

capteurs dans un espace réduit, de façon à ce qu'ils puissent communiquer et échanger des données en continu. D'autre part, et dans l'intérêt de leurs utilisateurs, ils doivent s'étaler dans l'espace afin d'effectuer des mesures sur la plus grande superficie possible. Nous sommes donc dans une situation de compromis, dans laquelle une contrainte technique (la portée de communication) force à agglomérer les capteurs en une même zone, tandis que leur but premier est de ramasser des données en masse et donc, rationnellement, de s'éloigner les uns des autres pour couvrir le plus de terrain possible. La seule issue favorable à ce problème est donc d'aboutir à une situation d'équilibre satisfaisant ces deux contraintes diamétralement opposées.

Le cadre de cet exercice s'accorde parfaitement avec la problématique de la prise de décision dans un réseau décentralisé puisque chaque capteur peut être assimilé à un agent autonome, capable d'agir sur la configuration de son environnement en se déplaçant dans l'espace. Quel que soit l'état d'un capteur, la vision du système dont il dispose est locale et doit servir seule à déterminer quelles actions il entreprend. L'objectif est donc ici de proposer une méthode de guidage que chaque capteur peut adopter et qui, par émergence d'une dynamique globale, résoudra notre problème en répartissant les capteurs dans l'espace de façon satisfaisante.

Les boîds de Craig Reynolds [Rey87] sont reconnus pour simuler un comportement de groupe a priori complexe et synchronisé à partir d'un jeu de règles simples. On s'en inspire, ainsi que d'un modèle de mouvement particulière [CSJ11] se basant sur la répulsion entre molécules de gaz, pour concevoir les trois règles qui gouverneront notre système. Chacune de ces lois, que l'on présente séparément par la suite, va engendrer une influence sur le déplacement de nos capteurs ; influence dont la composition sera détaillée en dernière partie.

Attraction

À la lecture de l'énoncé de ce problème, la nécessité de rapprocher les capteurs les uns des autres, afin qu'un réseau de communication ininterrompu se forme, vient naturellement à l'esprit. En effet, la condition *sine qua non* au bon fonctionnement du réseau est la communication. Un capteur isolé est inutile puisque son information n'est pas partagée.

Par sa loi de la gravitation universelle, Isaac Newton décrit la force qui attire toute paire de corps comme étant proportionnelle à leur masse respective et à la distance les séparant. Nous nous permettons de simplifier quelque peu son équation et d'en retirer l'aspect massique pour obtenir une règle qui a tout couple de corps associe une attraction proportionnelle à leur distance. S'ils obéissaient à cette loi, nos capteurs auraient naturellement tendance à se grouper, et donc à se mettre à portée de communication les uns des autres.

À l'échelle de notre système, le cadre est différent de celui de la loi de Newton, et cette influence n'est pas universelle. Nous ne pouvons malheureusement pas réécrire les règles de ce monde et inventer une attirance magique entre toute paire de capteurs. On peut néanmoins la simuler. Si deux capteurs sont à portée et peuvent échanger leur positions respectives alors la détermination de la distance les séparant est aisée. À partir de là, on peut imiter un phénomène d'attraction.

Concrètement, on associe à tout capteur c un rayon d'attraction R_a (voir figure 2). Si un capteur voisin se trouve à la fois dans le rayon de communication R_c de c et à l'extérieur de R_a , la force d'attraction que ce capteur devra subir est fournie par la formule suivante :

$$\vec{a} = \frac{\vec{p}_c - \vec{p}_v}{|\vec{p}_c - \vec{p}_v|^2}$$

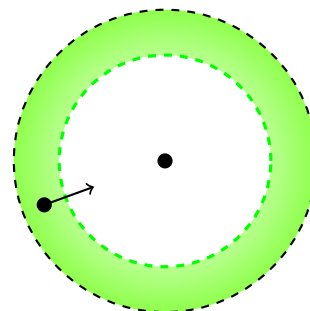


FIGURE 2 – En vert, le rayon d'attraction, dont l'intensité est décroissante de l'extérieur vers l'intérieur.

On note que l'attraction n'a pas pour but d'approcher des capteurs trop éloignés pour communiquer puisque l'existence d'une connexion entre deux

capteurs est un prérequis à l'application de la force d'attraction. ???

Dans le phénomène d'attraction réel, l'action est réciproque et instantanée mais dans notre cas elle est unidirectionnelle. Un capteur est attiré par tous ses voisins mais le processus décrit ne modifie pas directement la position du voisin et il n'en subit aucune influence. Il est néanmoins à prévoir que lorsque ce voisin calculera les influences qui s'appliquent à lui, il en subira la réciproque.

Répulsion

L'attraction a pour effet d'agglomérer en groupes serrés tous les capteurs démarrant la simulation dans une même zone. Mais même si de cette façon la communication est assurée, les capteurs finissent par tous être superposés sur la même position au bout d'un certain nombre d'itérations et la contrainte de couverture n'est absolument pas satisfaite.

Pour pallier cette déconvenue, on introduit une nouvelle force opposée à l'attraction, la répulsion. Le rayon R_r définit une nouvelle zone radiale qui, contrairement à la loi précédente, expulsera les capteurs envahissants vers l'extérieur. Il est bien sûr nécessaire que $R_r < R_c$.

$$\vec{r} = (\vec{p} - \vec{p}_v) ???$$

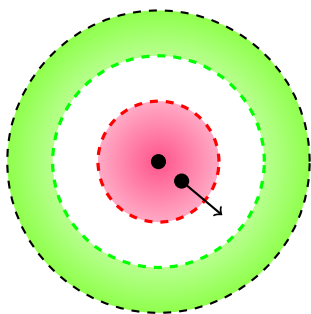


FIGURE 3 – En rouge, le rayon de répulsion, dont l'intensité est décroissance du centre vers les bords.

On prend $R_r = R_a - \varepsilon$, où ε correspond à la largeur d'une bande neutre entourant chaque capteur (voir figure 3). Les capteurs auront naturellement tendance à s'installer dans ces zones libre de toute influence et à se verrouiller ???. Nous verrons par la pratique que la conséquence principale

de ce positionnement forcé est un maillage régulier et géométrique. La régularité spatiale du réseau de capteur est ici un bonus appréciable. ???

Gravité

Les deux règles présentées permettant aux capteurs proches de s'agglomérer en composantes connexes équilibrées mais notre objectif est d'obtenir une unique composante connexe. Dans cette optique, on introduit une dernière influence inspirée du physique, la gravité.

Comme spécifié en introduction, on part de l'hypothèse que les capteurs sont munis de moyens de localisation dans l'espace et nous en avons allègrement profiter pour les calculs de distances des formules précédentes. Toujours dans cette ???, on peut supposer que tous les capteurs connaissent le centre de l'environnement dans lequel ils sont largués. Cette position peut être la position de laquelle ils ont été largués, une position pré-programmée ou un point calculée de façon commune par moyenne de toutes leur position(sujet intéressant!). L'important est que cette position soit connu de tous et servent d'origine dans leur repère.

La gravité est la force qui va attirer tous les capteurs vers le centre de l'environnement. À première vue, une telle règle semble dangereuse car on imagine qu'elle forcera les capteurs à se déplacer vers le centre de leur environnement au risque de sacrifier la couverture dont nous fait disposer l'effet répulsif inter-particulaire. Néanmoins, on compte sur l'effet combiné des différentes forces pour résister à cet effet.

Son intérêt est d'amener vers le centre les composantes connexes et les capteurs isolés pour tous les assembler en un unique réseau.

$$\vec{g} = -\vec{p} ???$$

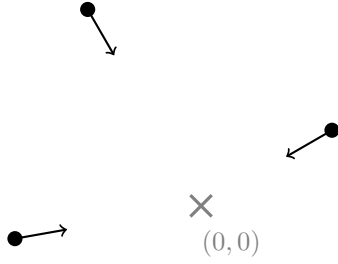


FIGURE 4 – La gravité attire tous les capteurs vers le centre de leur environnement.

Composition d'une force nette

Une fois ces trois forces calculées, on ne peut pas les appliquer naïvement sur le capteur concerné car il s'agit d'un objet physique soumis à des limitations quant à sa vitesse de déplacement. On associe donc à chaque capteur une vitesse de déplacement maximale qui servira de borne supérieure à la magnitude des trois forces combinées.

Puisque l'on dispose de trois influences différentes, chacune tenant un rôle particulier dans la résolution du problème, l'idée la plus intuitive est d'en déduire une influence moyenne.

$$\vec{f} = \frac{\vec{a} + \vec{r} + \vec{g}}{3}$$

En pratique, on observe que ce choix montre vite ses limites. Dans certains cas, deux forces seront opposées et par ce type de combinaison, elles s'annuleront mutuellement et on obtiendra au final un mouvement nul. Et l'immobilité n'est pas une solution envisageable à ce problème. Dans d'autres cas encore plus fréquents, les trois forces semblent adaptées : on observe des capteurs éparses flotter lentement vers le centre de leur environnement et s'assembler en composante connexe de plus en plus grande au fur et à mesure de leur traversée de l'espace. Mais une fois que le maillage est complet et qu'une situation d'équilibre semble avoir été atteinte, le centre du réseau commence à trembler,

les capteurs se rapprochent dangereusement et force le passage vers le centre en ignorant les rayons de répulsions de leur voisin. Ce problème est dû au nombre élevé de tensions s'opérant au centre du réseau, résultant en une pression intense qui fait s'effondrer le réseau jusqu'à ce que tous les capteurs occupent la même position. Moyenner les forces est donc une fausse bonne idée.

MEILLEURE SOLUTION
DEMO EN TROIS IMAGES

Obstacles

On propose d'étendre ce modèle de guidage afin de permettre aux capteurs d'éviter automatiquement les obstacles par l'ajout d'une nouvelle force virtuelle. Ici, un obstacle est défini comme étant une zone non traversable. En réalité, il peut s'agir d'un bâtiment, d'une zone dont les relevés ne nous intéressent pas, ou d'une zone interdite aux appareils volants.

REPULSION

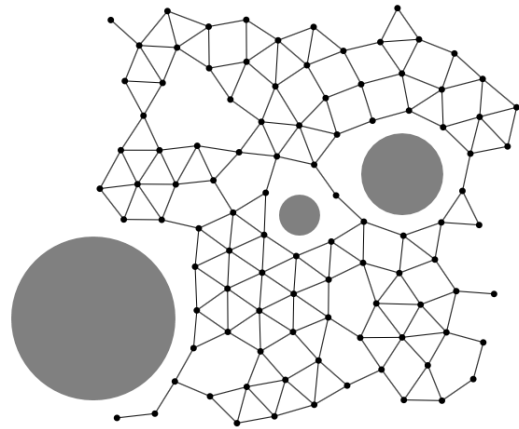


FIGURE 5 –

4 Conclusion

Références

- [CSJ11] Teddy M. Cheng, Andrey V. Savkin, and Faizan Javed. Decentralized control of a group of mobile robots for deployment in sweep coverage. *Robotics and Autonomous Systems*, 59(7–8) :497 – 507, 2011.

- [Rey87] Craig W. Reynolds. Flocks, herds, and schools : A distributed behavioral model. In *SIGGRAPH '87 : Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 25–34. ACM, 1987.