Pairwise gossip in CAT(k) metric spaces

Anass Bellachehab

▶ **To cite this version:**

Anass Bellachehab. Pairwise gossip in CAT(k) metric spaces. Networking and Internet Architecture [cs.NI]. Institut National des Télécommunications, 2017. English. NNT : 2017TELE0017 . tel-01892392

HAL Id: tel-01892392
https://theses.hal.science/tel-01892392

Submitted on 10 Oct 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Gossip pair-à-pair dans les espaces CAT(κ)

THESE DE DOCTORAT
Ecole doctorale: Informatique, Télécommunications et Electronique de Paris

Spécialité

Informatique et réseaux

Présentée par

M.Bellachehab Anass

Pour obtenir le grade de

DOCTEUR de Télécom SudParis

Soutenue le 10/11/2017, à Télécom ParisTech devant le jury composé de :

Rapporteurs :
Cédric Richard – Professeur – Université Nice
Silvère Bonnabel – Professeur – Mines ParisTech

Examinateurs :
Anne Bouillard – Chercheuse – Nokia Bell Labs
Romain Couillet – Professeur – Centrale-Supelec
Olivier Rioul – Professeur – Télécom ParisTech

Encadrant :
Jérémie Jakubowicz – Maître de conférence – Télécom SudParis

Directeur de thèse :
Pascal Bianchi – Professeur – Télécom ParisTech

N° NNT : 2017TELE0017

Remerciements

Je voudrais tout d'abord remercier mon encadrant Jérémie Jakubowicz et mon directeur de thèse Pascal Bianchi qui m'ont accompagné durant toute ma thèse. Leur exigence et leur rigueur m'ont permis de m'améliorer et leur soutien m'a été d'une grande importance. Je tiens aussi à remercier Tijani Chahed pour ses remarques précieuses sur mon manuscrit.

Je tiens particulièrement à remercier Cédric Richard et Silvère Bonnabel pour avoir accepté de lire et de rapporter ma thèse. Je remercie également Anne Bouillard, Olivier Rioul, et Romain Couillet, examinateurs lors de ma soutenance.

Finalement, je tiens à remercier ma famille pour m'avoir poussé et supporté durant ce projet.

Pairwise gossip in $CAT(\kappa)$ metric spaces

Bellachehab Anass

September 23, 2018

Contents

Chapter 1

Introduction

Cette thèse s'est déroulée au sein du laboratoire SAMOVAR (Services répartis, Architectures, Modélisation, Validation, Administration des Réseaux) qui est une unité mixte de recherche Télécom SudParis-CNRS (UMR 5157), sous la direction de Pascal Bianchi et Jérémie Jakubowicz. L'objectif de cette thèse est de généraliser l'algorithme du "gossip" pair-à-pair ([BGPS06]). L'algorithme du "gossip" pair-à-pair consiste à calculer la moyenne arithmétique, de façon répétée, de vecteurs stockés dans deux noeuds adjacents d'un graphe, et d'affecter la moyenne ainsi calculée aux deux noeuds en question. Nous reviendrons en détail sur cet algorithme dans la suite de ce manuscrit. La généralisation proposée dans cette thèse s'intéresse au cas où les "vecteurs" sont de simples points dans un espace métrique et les calculs de moyennes deviennent des calculs de milieux de géodésiques. Dans les espaces métriques quelconques, il n'y a aucune raison pour que deux points admettent nécessairement un milieu, ni une géodésique, ni-même, si ce milieu existe, qu'il soit unique. Et quand bien même on ne considérerait que des espaces métriques pour lesquels le milieu de deux points existe toujours et est unique, il n'y a pas de raison pour que l'algorithme du "gossip" pair-à-pair décrit converge. Cette thèse développe un bon cadre pour étudier cette généralisation : celui des espaces métriques dit $CAT(\kappa)$ [Gro07], suivant Cartan, Alexandrov et Toponogov et où κ est un majorant scalaire de la "courbure" de l'espace. Là encore, nous reviendrons en détail sur les aspects techniques dans la suite de ce manuscrit.

Dans un premier chapitre, on passe en revue un certain nombre d'approches des algorithmes de gossip pour le consensus. En particulier, on présente les travaux précurseurs de DeGroot [DeG74] sur le consensus par moyennes itérées, puis l'approche de Tsitsiklis [Tsi84] sur les calculs distribués où le consensus est abordé sous l'angle de l'optimisation distribuée. Sont présentés également le gossip pair-à-pair aléatoire [XB04] ainsi que les algorithmes de consensus en temps fini [SJJ12, HSJ15]. La dernière partie du chapitre 1 introduit l'angle sous lequel l'ouverture vers les espaces métriques sera la plus aisée : la minimisation d'une

fonction uniquement basée sur la métrique de l'espace sous-jacent.

Le deuxième chapitre est la clé de voûte de cette thèse. Il développe le cadre $CAT(\kappa)$ et montre les principaux résultats théoriques. Loin d'être évidente, la généralisation aux espaces métriques $CAT(\kappa)$ fait jouer un rôle important à la courbure. Plus précisément, nous montrons que lorsque κ est négatif, tout se passe comme dans le cas euclidien: il y a convergence linéaire quelle que soit la configuration initiale des points. En revanche, lorsque κ est strictement positif, la configuration initiale joue, et on ne peut pas garantir la convergence dans toutes les configurations initiales.

Le troisième chapitre porte sur une application du cadre $CAT(\kappa)$ aux systèmes métamorphes. Outre leur intérêt applicatif en robotique, ils sont aussi remarquables dans la mesure où ils ne s'inscrivent ni dans le cadre des espaces vectoriels, ni dans celui des variétés riemanniennes de courbure négative. Autrement dit, cette application est originale dans la mesure où elle rentre dans le cadre de l'algorithme de consensus étudié dans cette thèse, mais pas dans les cadres qui préexistaient.

Le quatrième chapitre de cette thèse généralise l'approche développée aux cas des graphes variant dans le temps et montre que, pourvu que la communication entre toutes les parties du graphe soit préservée, les résultats du cas "statique" restent valides. Un sens précis est donné à la "préservation de la communication entre toutes les parties du graphe".

1.1 Motivations

Un des plus anciens problèmes dans le domaine des algorithmes distribués, et probablement un des plus fondamentaux aussi, est le problème dit du *consensus* [FLP85]. Un résultat marquant concernant les algorithmes de consensus est le théorème d'impossibilité de Fischer, Lynch et Paterson (FLP) [FLP85]. Ce résultat concerne le modèle dit "panne-arrêt" ("fail-stop" en anglais) pour les agents déterministes asynchrones. Dans ce modèle, les agents font des calculs et s'envoient des messages mais peuvent également tomber en panne. Un agent qui tombe en panne cesse de calculer et n'envoie plus aucun message. Les pannes peuvent se produire de façon imprévisible, ou ne pas se produire du tout. Les pannes sont irréversibles, ce qui signifie qu'une fois qu'un agent est en panne, il le reste. Le théorème FLP traite des communications asynchrones : les messages prennent un temps non borné pour être transmis d'un agent à un autre, même si on est sûr qu'ils finiront par être délivrés. De sorte qu'un agent qui attend un message d'un autre agent, et ne l'a pas encore reçu, ne peut pas dire si cet autre agent est en panne, ou que le message prend simplement du temps à être délivré. En revanche, il est sûr que si un message lui a été envoyé, il finira par le recevoir, sauf

à lui-même tomber en panne. Chaque agent part avec une valeur binaire d'entrée et devra, à terme, fournir une valeur binaire de sortie, à moins de tomber en panne entre temps. Une fois qu'un agent décide d'une valeur de sortie, il ne peut plus en changer mais peut continuer à communiquer avec les autres agents. Chercher le consensus signifie que toutes les valeurs de sortie doivent coïncider, sachant que les agents en panne n'affichent aucune valeur de sortie. De plus, pour éviter une situation inintéressante où les agents s'entendraient au préalable sur une valeur de consensus, indépendamment de leur valeur initiale, on demande que si tous les agents sont déjà dans une situation de consensus initial, disons "vrai" pour fixer les idées, le consensus atteint en sortie soit aussi "vrai". Le théorème FLP stipule qu'en présence de pannes, même s'il s'agit d'une seule panne, il n'y a pas moyen d'établir un protocole qui garantisse ce consensus.

Le cadre du théorème FLP est très éloigné du cadre dans lequel on va se placer, mais c'est une motivation essentielle dans la mesure où c'est probablement l'un des résultats phare du domaine des algorithmes de consensus. Le cadre étudié dans cette thèse est aussi appelé "asynchrone" mais dans un sens différent. C'est un cadre beaucoup moins "asynchrone" que celui du théorème FLP. En effet, dans le cadre qui va nous occuper, chaque agent possède une horloge distincte qui bat suivant un processus de Poisson d'intensité fixe. Ce processus de Poisson modélise une horloge régulière mais bruitée. Cette régularité sous-jacente est totalement absente du cadre FLP. Néanmoins, dans les deux cas, le mot "asynchrone" correspond à la terminologie standard. C'est le fait d'avoir des conditions de synchronisation beaucoup plus régulières qui fait que dans le cadre qui nous intéresse, on arrive non seulement à montrer qu'il existe un protocole menant au consensus, mais nous allons aussi en exhiber un et étudier sa vitesse de convergence.

Un autre travail remarquable, cette fois ci beaucoup plus proche du nôtre est la thèse de J. Tsitsiklis [Tsi84] qui date de 1984. Dans le travail de Tsitsiklis, les agents communiquent dans un temps borné et aucune panne ne peut se produire, tout au plus des retards bornés. Tsitsiklis montre que non seulement on peut atteindre un consensus, mais qu'on peut utiliser l'algorithme qui permet de l'atteindre comme une brique élémentaire pour des systèmes plus sophistiqués, tels que les algorithmes d'optimisation distribuée.

Citons également le travail de Boyd *et al.* [BGPS06] qui est le premier à proposer le cadre suivant : chaque agent possède une valeur initiale vectorielle et les agents communiquent de façon pair-à-pair dans un ordre aléatoire, en calculant deux-à-deux la moyenne de leurs deux valeurs. Boyd *et al.* montrent que cet algorithme converge presque sûrement vers un consensus.

Dans la mesure où calculer la moyenne arithmétique de deux vecteurs consiste à calculer le milieu du segment formé par leurs extrémités, nous nous sommes posé la question suivante: est-il possible de proposer une variante géométrique

de l'algorithme de Boyd *et al.* ? Une variante où chaque agent part d'un point dans un espace géométrique et calcule le milieu du segment formé par son point à une extrémité, et celui d'un de ses voisins tiré au hasard parmi l'ensemble de ses voisins, à l'autre extrémité.

1.2 Algorithmes de consensus et espaces métriques $CAT(\kappa)$

Dans cette thèse, on étudie donc le problème du consensus dans un réseau de N agents, modélisé par un graphe non orienté $G = (V, E)$, où l'ensemble des noeuds $V = \{1, \ldots, N\}$ désigne les agents et l'ensemble des arêtes E désigne les liens de communications entre ces agents. Chaque agent i stocke une donnée dont la valeur x_i appartient à un espace métrique $\mathcal{M}$.

Le cadre de communication du gossip pair-à-pair n'est pas synchrone. Chaque noeud du réseau possède une horloge interne qui suit un processus de Poisson de paramètre $\lambda > 0$, commun à tous les agents. A chaque coup d'horloge, un agent peut faire des calculs et communiquer avec d'autres agents voisins. Les horloges des agents sont supposées indépendantes. Ce modèle est équivalent à celui d'une horloge commune qui suit un processus de Poisson de paramètre λN, et qui à chaque coup, réveille de manière aléatoire et uniforme un agent du réseau. Cette horloge commune est une vue de l'esprit : il n'y a pas besoin de l'implémenter en pratique, il suffit que toutes les horloges des agents soient indépendantes et de même intensité.

À chaque réveil de l'horloge commune, un noeud v est choisi aléatoirement et uniformément dans le graphe. On note $v(k) \in V$ le noeud choisi au k-ème réveil de l'horloge commune. Le noeud v réveille à son tour un noeud w voisin de lui dans le graphe G. On note $w(k)$ le voisin choisi par $v(k)$. À chaque instant k, on note $X(k) = (x_1(k), \ldots, x_N(k))$ l'état global du système.

Par rapport au travail de Boyd *et al.* [BGPS06], notre travail consiste a résoudre trois problèmes:

1. **Problème 1:** identifier un cadre de généralisation et une version adaptée de l'algorithme du gossip au cas des données non-vectorielles.

2. **Problème 2:** établir les conditions suivant lesquelles l'algorithme du gossip généralisé converge et estimer, le cas échéant, sa vitesse de convergence.

3. **Problème 3:** étudier sous quelles conditions l'algorithme du gossip généralisé peut être appliqué aux réseaux de topologie variable.

Dans le cas du gossip pair-à-pair classique, les calculs consistent à faire des moyennes arithmétiques. Pour généraliser l'algorithme à des espaces de données non-euclidiens (dans notre cas des espaces métriques), on a besoin de remplacer la notion de moyenne arithmétique par celle du point milieu. Cependant, les espaces métriques généraux sont trop irréguliers pour que la notion de point milieu y soit bien définie: il peut y avoir entre deux points donnés plusieurs points milieux ou aucun.

Les espaces métriques avec plusieurs points milieux sont nombreux. Un exemple est la sphère: chaque couple de points antipodaux a plusieurs points milieux qui forme "l'équateur". Un exemple d'espace métrique où on peut trouver des couples de points sans point milieu est le plan $\mathbb{R}^2$ sans l'origine. Les points $(-1, 0)$ et $(1, 0)$ par exemple n'ont pas de point milieu dans $\mathbb{R}^2 - (0, 0)$.

1.3 Résultats obtenus

Notre approche pour généraliser l'algorithme du gossip aux espaces de données non-Euclidiens utilise la notion d'espace métrique $CAT(\kappa)$. Les espaces $CAT(\kappa)$ permettent d'étendre la notion de courbure, qui vient des variétés riemanniennes, aux espaces métriques. Ceci permet ensuite, via des inégalités, dites de comparaison, d'étudier les propriétés géométriques d'un espace métrique en le comparant à des espaces de référence de courbure constante. Les espaces de courbure constante ont des propriétés géométriques bien connues qui permettent d'établir des éléments de comparaison.

Dans les espaces $CAT(\kappa)$, la notion de point milieu entre chaque couple de points est bien définie, pourvu que les points soient assez proches. On peut donc généraliser sans peine l'algorithme du gossip pair-à-pair "classique", pourvu que les points initiaux soient assez proches pour qu'on puisse considérer leur milieu. En terme algorithmique, on utilise le procédé décrit à la table RPM.

Ce qui n'est pas évident, et qui sera démontré dans cette thèse, est que si les points sont initialisés de façon à être suffisamment proches alors ils le resteront au fur et à mesure des itérations. On peut alors démontrer que l'algorithme converge vers un état de consensus avec une vitesse linéaire si l'espace de données est de type $CAT(\kappa)$.

Pour énoncer rigoureusement le résultat obtenu, il nous faut un moyen de mesurer l'écart au consensus. Pour des raisons techniques qui apparaîtront plus tard dans le manuscrit, on utilise la fonction suivante, dite *fonction de désaccord*.

Définition 1. *Étant donnée une configuration $x = (x_1, \ldots, x_N) \in \mathcal{M}^N$, la fonc-*

Algorithm RPM

[BJ16]

Input: un graphe $G = (V, E)$ et une configuration de points initiaux $X_v(0), v \in V$

for all $k > 0$ **do**

On choisit au hasard uniformément aléatoirement un noeud V_k de V, puis un noeud W_k uniformément aléatoirement parmi les voisins de V_k

Faire:

$x_{V_k}(k) = \text{Milieu}\,(x_{V_k}(k-1), x_{W_k}(k-1))$

$x_{W_k}(k) = \text{Milieu}\,(x_{V_k}(k-1), x_{W_k}(k-1))$

$x_v(k) = x_v(k-1)$ for $v \notin \{V_k, W_k\}$

end for

tion de désaccord $\Delta(x)$ *est donnée pour* $\kappa = 0$ *par:*

$$\Delta(x) = \sum_{\substack{v \sim w \\ \{v,w\} \in \mathcal{P}_2(V)}} \frac{1}{N}(\deg(v)^{-1} + \deg(w)^{-1}) d^2(x_v, x_w)$$

Et pour $\kappa > 0$ *par:*

$$\Delta_\kappa(x) = \sum_{\substack{v \sim w \\ \{v,w\} \in \mathcal{P}_2(V)}} \frac{1}{N}(\deg(v)^{-1} + \deg(w)^{-1}) \chi_\kappa\left(d(x_v, x_w)\right)$$

Avec $\chi_\kappa(x) = 1 - \cos(\sqrt{\kappa}x)$*. Et où on note* deg*, la fonction degré d'un nœud.*

A noter que lorsque le réseau est connexe, alors $\Delta_\kappa = 0$ si et seulement si le réseau est dans un état de consensus.

On l'a évoqué plus haut, dans le cas où l'espace de données est de type $CAT(\kappa)$ avec $\kappa > 0$, on doit ajouter une condition sur l'état initial du système: les points doivent être suffisamment proches. Précisément, on exige que le diamètre du nuage de points initial (définit comme étant la borne supérieure sur les distances entre les points) soit inférieur à la grandeur $\frac{D_\kappa}{2}$ où:

$$D_\kappa = \begin{cases} +\infty & \text{si } \kappa \leq 0 \\ \frac{\pi}{\sqrt{\kappa}} & \text{si } \kappa > 0 \end{cases}$$

On obtient alors le résultat suivant, qui est le résultat central de cette thèse.

Théorème 1. *Si le graphe* G *est connexe, et que* $X(k) = (X_1(k), \ldots, X_N(k))$ *désigne l'état des agents au temps* k*; que le diamètre de l'ensemble* $\{X_1(0), \ldots, X_N(0)\}$ *est inférieur à* $\frac{D_\kappa}{2}$*, alors l'algorithme décrit à la table RPM converge et il existe* $L \in (-1, 0)$ *tel que* $\forall k$*:*

$$\mathbb{E}\Delta_\kappa(X(k)) \leq \exp(Lk)$$

1.4 Applications

Une première application de l'algorithme RPM concerne les matrices définies positives $n \times n$ noté $\mathrm{Pos}(n)$. L'ensemble des matrices définies positives équipé de la métrique: $d(M,N) = \|\log(MN^{-1})\|_2$ – où $\log(M)$ désigne la seule matrice symmétrique positive telle que $\exp(\log(M)) = M$ – est un espace métrique de type $CAT(0)$, c'est-à-dire, à courbure négative. Le point milieu entre deux éléments de cet espace est donné par:

$$\frac{M+N}{2} = M^{1/2}(M^{-1/2}NM^{-1/2})^{1/2}M^{1/2} \ .$$

L'algorithme RPM fournit alors un outil pour obtenir un consensus dans l'espace des matrices définies positives. Lorsqu'on trace la courbe $n \mapsto \log \Delta_\kappa(X_n)$ on peut bien vérifier dans la Figure 1.1 qu'on obtient une droite décroissante bruitée, ce qui confirme les résultats théoriques obtenus dans cette thèse. Le bruit est dû à la nature stochastique de l'algorithme qui choisit à chaque itération une paire d'agents aléatoire.

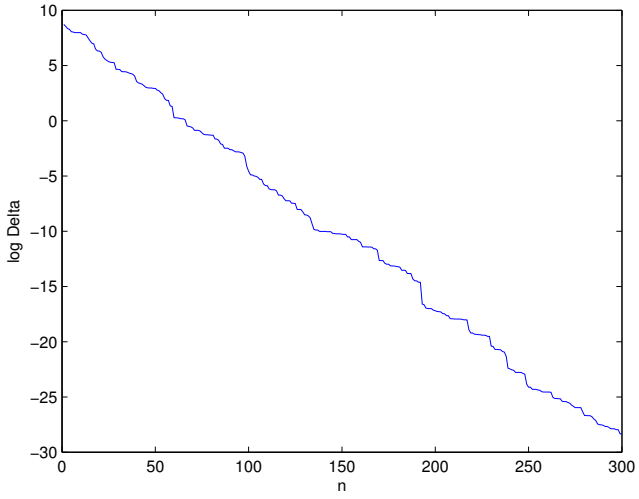


Figure 1.1 – Courbe $n \mapsto \log \Delta_\kappa(X_n)$. On voit ce qui s'apparente à une droite de pente négative – ce qui est en accord avec le théorème de convergence mentionné plus haut. Le bruit est dû à la nature stochastique de l'algorithme.

Une autre application est celle de la classification distribuée. Ici, chaque noeud v du réseau doit choisir une parmi $p < \infty$ possibilités. Ce choix est modélisé par une loi de probabilité notée $(x_v^i)_{i\in[1,p]}$. Ainsi $x_v^i \geq 0$ représente la croyance du nœud v à la possibilité i et $\sum_{i=1}^p x_v^i = 1$. Le réseau doit s'entendre sur une loi de probabilité commune, c'est-à-dire un consensus. Notons qu'ici le consensus porte sur la

loi de probabilité et non sur les possibilités elles-mêmes. Formellement, l'espace $\mathcal{M}$ considéré est le *simplexe* de dimension $p-1$ qu'on munit de la métrique de Fisher, qui est la métrique la plus naturelle pour cet usage. En utilisant l'application $T : (x_1, \ldots, x_p) \to (2\sqrt{x_1}, \ldots, 2\sqrt{x_p})$, on peut montrer que le simplOnexe peut être identifié à l'hypersphère de dimension $p-1$ et de rayon 2. Cette dernière est un espace métrique $CAT(\frac{1}{2})$. En initialisant les points suffisamment proches et en traçant la courbe $n \mapsto \log \Delta(X_n)$ on obtient la Figure 1.2

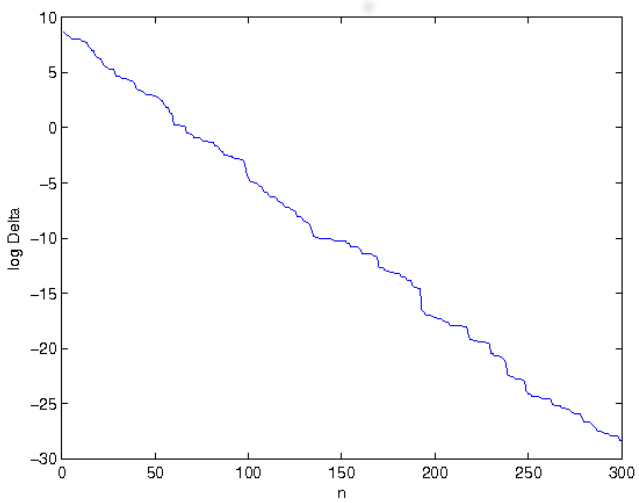


Figure 1.2 – Courbe $n \mapsto \log \Delta_\kappa(X_n)$. Il s'agit d'une droite de pente négative – ce qui est en accord avec le théorème de convergence mentionné plus haut. On décèle la présence de bruit, dû à la nature stochastique de l'algorithme.

Les deux exemples cités plus haut ont la particularité d'être analysables dans un cadre moins général que celui des espaces métriques. Les espaces de matrices définies positives et le simplexe sont tous les deux des exemples de variétés riemanniennes. Il existe cependant des cas où les espaces de données sont purement métriques auxquels aucune structure différentielle ne peut être appliquée. Un exemple de tels espaces est celui des systèmes métamorphes.

Beaucoup de problèmes en robotique, en informatique, ou en biologie impliquent des systèmes dits *reconfigurables* ou *métamorphes*. Ces systèmes changent leur état suivant un ensemble de *règles locales*, pour passer d'un état A vers un autre état B. Le système doit déterminer la séquence de mouvements locaux qu'il doit exécuter [KHJ+03], les arbres phylogénétiques [BHV01] et les robots métamorphes [ABY12] en sont des exemples.

Dans cette thèse nous avons cherché à appliquer l'algorithme RPM au cas des robots métamorphes. Un robot métamorphe est une collection de modules

individuels qui sont capables de se connecter/déconnecter les uns aux autres et de former une structure rigide, qu'on appelle une *configuration* ou *état*. Les modules individuels peuvent changer leur position relativement à leurs voisins suivant une série de règles locales, pourvu que le système reste connexe. Ceci permet au système de changer sa position et sa configuration de manière dynamique.

Certains robots métamorphes sont basés sur des grilles, c'est à dire que les modules de base occupent un ensemble discret de positions sur une grille. La grille peut être hexagonale, carrée, etc. Ces systèmes peuvent être décrits mathématiquement comme une série de points dans un graphe (voir l'appendice A). Par exemple, le bras robotique qui consiste en des liens attachés à l'intérieur d'une grille et dont l'une des extrémités est attachée au point $(0, 0)$ de la grille (cf figure 1.3) est un robot métamorphe.

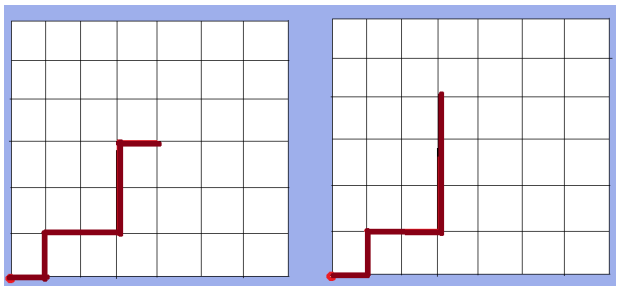


Figure 1.3 – Un exemple de robot métamorphe: le bras robotique. Les arêtes en couleur représentent la présence d'un module, les arêtes en noir indiquent l'absence de module. Le bras est attaché à sa base au point $(0, 0)$. Ici un mouvement élémentaire a été executé par le module au bout du bras, ce qui change la configuration du système.

Dans [AG04, GP07] les auteurs introduisent une description mathématique des systèmes métamorphes en injectant l'espace des états du système dans un espace géométrique continu. Cette injection est appelée le *complexe d'états*. Muni d'une métrique adéquate le complexe d'état est un espace $CAT(0)$ (voir l'appendice B).

On suppose ici qu'on dispose de plusieurs robots métamorphes connectés grâce à un réseau de communication. On cherche à amener tous les robots vers un même état commun sans avoir recours à un agent central, c'est-à-dire, en ayant uniquement recours au réseau de communications utilisé par les robots eux-mêmes.

Le point milieu de deux points quelconques est calculé via modification d'un algorithme d'estimation du plus court chemin [Bac14] entre deux points du complexe d'états du système. Contrairement aux cas mentionnés plus haut, le point milieu est ici relativement plus complexe à calculer. Là encore, on pourrait montrer que les prédictions du théorème de convergence mentionné plus haut se retrouvent en pratique.

1.5 Graphe variable

Dans un dernier travail, nous avons cherché à mesurer quel pourrait être l'impact d'un graphe qui change dans le temps. En particulier, on cherche à savoir si cela affecte les propriétés de convergence de l'algorithme (RPM).

Le réseau de N agents est représenté par un graphe où les noeuds forment un ensemble fixe mais les arêtes sont susceptibles de varier: $G(t) = (V, E(t))$ où $E(t)$ est l'ensemble des arêtes à l'instant t. Si t_1 et t_2 sont deux instants distincts, on peut définir l'union de deux graphes $G(t_1) = (V, E(t_1))$ et $G(t_2) = (V, E(t_2))$ par $G(t_1) \cup G(t_2) = (V, E(t_1) \cup E(t_2))$.

L'algorithme étudié est rigoureusement le même. À chaque instant k il n'y a que deux agents qui communiquent dans tout le réseau: V_k et W_k. On suppose que les (V_k, W_k) sont indépendants même s'il ne peuvent bien évidemment plus être identiquement distribués.

Il faut également adapter la fonction de désaccord en prenant en compte le fait que les degrés dépendent maintenant du temps:

Définition 2.

$$\Delta(x,k) = \sum_{\{v,w\}\in E(k)} (\deg_v(k)^{-1} + \deg_w(k)^{-1})\chi_\kappa\left(d(x_v, x_w)\right)$$

On obtient alors le résultat suivant:

Théorème 2. *On suppose qu'il existe $T > 0$ tel que pour tout $k \geq k_0$, $G(k) \cup \cdots \cup G(k+T)$ soit un graphe connexe, où k_0 désigne un entier. On suppose aussi que le diamètre de l'ensemble $\{X_1(0), \ldots, X_N(0)\}$ est inférieur à $\frac{D_\kappa}{2}$. Alors l'algorithme décrit à la table RPM converge et il existe $L \in (-1, 0)$ tel que:*

$$\mathbb{E}\mathcal{D}_\kappa(k) \leq \exp(Lk)$$

Avec:

$$\mathcal{D}_\kappa(k) = \sum_{n=k}^{k+T} \Delta_\kappa(X(n))$$

1.6 Organisation du mémoire de thèse

Le manuscrit se divise en quatre chapitres dont les relations sont résumées par le schéma représenté à la figure 1.4.

1.7 Production scientifique

Les travaux de recherches présentés dans ce manuscrit ont fait l'objet des publications suivantes

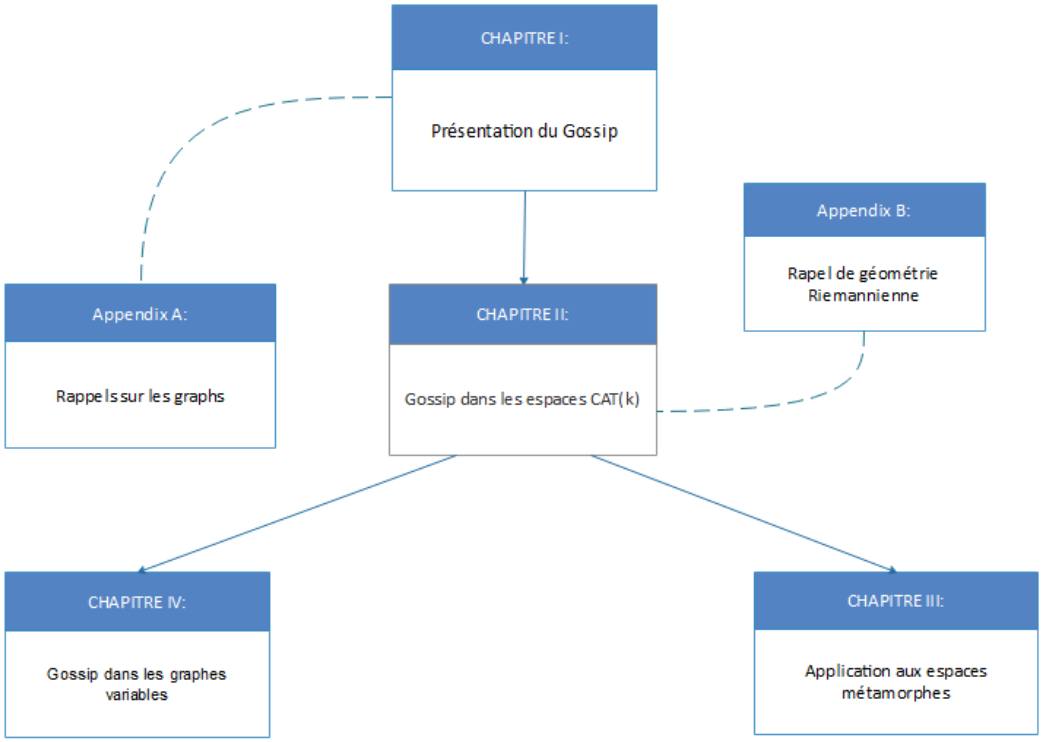


Figure 1.4

1.7.1 Articles dans des revues internationales à comité de lecture

J1 A.Bellachehab, "Distributed consensus for metamorphic systems using a gossip algorithm for $CAT(0)$ metric spaces", Entropy 2015, 17(3), 1165-1180

J2 A.Bellachehab, J.Jakubowicz, "Random Pairwise Gossip on $CAT(\kappa)$ metric spaces" soumis à IEEE Transactions on Automatic control, Volume: 61, Issue: 12, Dec. 2016

1.7.2 Articles de conférences internationales

C1 A.Bellachehab, J.Jakubowicz, "Random pairwise gossip on Hadamard manifolds", Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2013 IEEE 5th International Workshop on

C2 A.Bellachehab, J.Jakubowicz, "Random pairwise gossip on CAT(0) metric spaces", Decision and Control (CDC), 2014 IEEE 53rd Annual Conference

on

C3 A.Bellachehab, J.Jakubowicz, "Distibuted consensus for metamorphic systems using a gossip algorithm for CAT(0) metric spaces" AIP proceedings on bayesian inference and maximum entropy methods in science and engineering (MaxEnt 2014)

1.7.3 Articles de conférences nationales

CN1 A.Bellachehab, P.Bianchi, J.Jakubowicz, "Consensus pair-a-pair asynchrone dans les variétés", 15-èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel), May 2013, Pornic, France

Chapter 2

Gossip in distributed networks

Gossip algorithms are iterative algorithms. Each agent starts with its own data (temperature, polluting substance concentration, *etc.*) and progressively exchanges with its neighbors, such that, in the end, all agents end up sharing the same data. The data space, defined as the space where data belong, plays an important role. For instance, if there is a pre-identified special point in this space, such as 0, agents can agree to set their data straight to that point. However, it can occur that such a special point does not exist, and more importantly, we require the consensus configuration to be dependent on the initial state.

There is another very important distinction related to the data space, that we will denote from now on by $\mathcal{M}$. Namely, in the case where $\mathcal{M}$ is a vector space, the existence of a scalar multiplication and addition greatly simplifies the algorithms. It is therefore quite natural that the first gossip algorithms took place in a Euclidean setting.

On the other hand, there is an invariant in both Euclidean and non-Euclidean cases, which is the communication framework. This framework encompasses the communication network topology, the degree of synchronization between agents and the way agents can exchange information. Let us describe this communication first and make the distinction according to the data space afterwards.

2.1 Communication Framework

2.1.1 Notations

Assume V is some finite set. We denote by $\mathcal{P}_2(V)$ the set of *pairs* of elements in V: $\mathcal{P}_2(V) = \{\{v, w\} : v \neq w\}$. Notice that, by definition, for $v \neq w$, $\{v, w\} = \{w, v\}$ whereas $(v, w) \neq (w, v)$. In what follows, random variables are assumed to be functions from a probability space Ω equipped with its σ-field $\mathcal{F}$ and probability

measure $\mathbb{P}$; $x = X(\omega)$ denotes the realization associated to $\omega \in \Omega$ for random variable X. For any set S and any subset A of S, $\delta\{A\}$ denotes the indicator function defined over S that takes value 1 on A and 0 otherwise. If $x = (x_1, \ldots, x_N) \in \mathbb{R}^N$ is an N-dimensional real vector then define : $\bar{x} := \frac{1}{N}\sum_{i=1}^{N} x_i$ which is the average value of the components of x.

2.1.2 Graph model

The notion of gossip protocol is usually tied to that of a distributed network i.e. a network where there does not exist a central fusion node that can communicate with all the others or where all the information could be routed and processed. This network is represented using a graph, with the agents being represented by the vertices of the graph and communication links by its edges.

A distributed network in our working definition must satisfy the following constraints:

- The communications and computational capacity of each node is very limited and each agent can communicate only with neighboring nodes in the communication graph.
- The network has no central fusion node.
- The topology of the network is not necessarily known to the agents.

Mathematically, we model a distributed network of N agents as a graph $G = (V, E)$ where V is the set of vertices representing the nodes, ($|V| = N$) and $E \subset \mathcal{P}_2(V)$ is the set of edges representing communication links between agents. A link $e \in E$ is given by a pair $\{v, w\} \in \mathcal{P}_2(V)$ where v and w are two distinct agents in the network that are able to communicate directly. We have implicitly assumed that the graph is undirected, meaning that if an agent $v \in V$ is able to communicate with an agent $w \in V$ then the reverse communication is also possible. This assumption makes sense when all nodes are supposed identical and communication is fast compared to agents movements. When a communication link $e = \{v, w\}$ exists between two agents, both agents are said to be *neighbors* and the link is denoted $v \sim w$.

How nodes communicate between each other and coordinate their actions with respect to time is the subject of the next subsection.

2.1.3 Time model

In this work we consider only discrete-time models, meaning that the time parameter will be a discrete variable. This implies that the different communications,

computations and update events will be occurring at distinct separate points in time.

There are two types of discrete-time synchronization models: synchronous and asynchronous.

In the synchronous model, multiple agents can communicate at the same time. The nodes perform calculations simultaneously and update their results at the same time. Various methods to enforce clock synchronization have been proposed in [SG07, LR06].

In the asynchronous model (see [BGPS06]), it is assumed that each agent possesses its own internal clock that ticks according to a Poisson process of intensity λ (the clocks are identically made), each clock is assumed to be independent of the others. When an agent's clock ticks, the agent is able to perform some computations and communicate with neighboring agents. This model is equivalent to that of a single global Poisson clock that ticks at a rate of $N\lambda$ at times Z_k, where Z_k is a Poisson process *i.e.* $\{Z_{k+1} - Z_k\}$ are iid exponentials of rate $N\lambda$. Time is discretized according to the global clock ticks, the interval $[Z_k, Z_{k+1})$ denoting the k-th time slot.

To this picture we add the capability for any given agent $v \in V$ awakened to awake a neighboring agent $w \in \mathcal{N}_v$ in order to communicate with it within the same slot. This is a strong requirement that implies some degree of synchronization between the nodes. This requirement is still less strong than the completely synchronous network where time would be slotted commonly between nodes.

2.1.4 Data space

Each node $v \in V$ stores data represented as an element x_v belonging to some data space $\mathcal{M}$ (more restrictive assumptions on $\mathcal{M}$ will follow). Initially, each node v has a value $x_v(0)$ and $x(0) = (x_1(0), \ldots, x_N(0))$ is the N-tuple of initial values of the network. We focus on iterative algorithms that tend to drive the network to a *consensus state*; meaning a state of the form $x_\infty = (x_\infty, \ldots, x_\infty)$ with: $x_\infty \in \mathcal{M}$. We denote by $x_v(k)$ the value stored by the agent $v \in V$ at the k-th ticking of the global clock and $x(k) = (x_1(k), \ldots, x_N(k))$ the global state of the network at instant k.

2.2 Euclidean gossip

Let us thence assume that the data space $\mathcal{M}$ is a Euclidean vector space. This case has been extensively studied, and its solution will serve as a template for other cases of more general data spaces. The Euclidean space encompasses many data spaces of practical interest (temperature, concentration of a polluting agent, etc).

In what follows, it is going to turn out that we can safely assume, *i.e.* without loss of generality, that $\mathcal{M}$ consists of scalars only: gossip algorithms are processed the same way when $\mathcal{M} = \mathbb{R}$ or $\mathcal{M} = \mathbb{R}^p$ with $p > 1$.

2.3 Previous approaches

We now review some well known approaches to tackle the problem of consensus in the Euclidean setting. One of the first attempts is due to M.H. DeGroot in the early 70's, even though his main goal was more related to social behavior than consensus in general networks. The idea is quite simple: each agent in the network averages its own value with the values of its neighbors. M.H. DeGroot proved that the scheme indeed converges to consensus. Later, Chaterjee and Seneta [CS77] refined DeGroot's model, allowing the averaging weights to change over time. They provided sufficient conditions under which consensus can still be guaranteed. Another landmark was the work of Krause [Kra00] that slightly modified the algorithm taking averages only among the agents that have close data values. Krause was able to show that consensus can no more be guaranteed with this modification. Instead the network has a tendency to form clusters.

One of the studies that investigated consensus in a more general context than social behavior was the work of Tsitsiklis [Tsi84]. The consensus problem is no longer restricted to the case of opinion dynamics, its scope has been extended now to include distributed computation between agents. Each agent of the network now stores a data value and can potentially –in the case of distributed optimization– have a local cost function that it seeks to minimize. The communication between agents is no longer assumed to be instantaneous and synchronization is no longer perfect. The consensus protocol in this case must take into account the synchronization delays and the fact that not every two agents can directly communicate with one another. A modifed version of the gossip protocol has been proposed by the author and has been proven to converge to consensus under certain communications assumptions.

Another interesting question is whether it is possible to devise a modified version of the distributed gossip protocol so that consensus is achieved in a finite series of steps rather than being asymptotically approached. In [HSJ15] the authors have shown that in some specific network and communication situations, there exists a sequence of communication and averaging steps that lead to a consensus in a finite number of iterations. This sequence of steps can be characterized using matrix notation, where each iteration is seen as the application of a linear operator to the vector of all data points stored by the agents. If the underlying communications graph of the network is undirected and connected, then there exists a finite series of matrices that drive the system towards a consensus state. If the underlying

communications graph is directed, further assumptions need to be made in order to guarantee finite time consensus.

2.3.1 Consensus in opinion dynamics

In [DeG74] the author studies consensus in the context of opinion dynamics. Consider a group of N agents that must estimate some parameter θ. Each agent $i \in \{1, \ldots, N\}$ has its own probability distribution F_i that describes its opinion of what θ is. In what follows we can assume without loss of generality that the F_i's are scalars. Indeed, since all the following operations are pointwise operations, the mathematical treatment of the problem with scalars, instead of functions, will remain the same. Our goal will be getting all the agents to agree on a common scalar F_c. The form of which is $F_C = \sum_{i=1}^{N} p_i F_i$ with: $\sum_{i=1}^{N} p_i = 1$.

The solution proposed by De Groot is a discrete time iterative process where at each step, one agent updates its value by averaging it with the values of the other agents. In mathematical terms, if at step $k-1$ the value of agent i is $F_{i,k-1}$, then the update at step k is:

$$F_{i,k} = \sum_j p_{i,j} F_{j,k-1} \tag{2.1}$$

The numbers $p_{i,j} \geq 0$ are such that: $\sum_{j=1}^{N} p_{i,j} = 1$, they represent the different weights an agent i gives to other agents opinions. These numbers form an $N \times N$ matrix: $P = (p_{i,j})_{1 \leq i,j \leq N}$. The global state of the system at iteration k is denoted by: $F_k = (F_{1,k}, \ldots, F_{N,k})$.

The update relation in equation 2.1 can be rewritten in matrix form as:

$$F_k = P F_{k-1} = P^k F_0$$

The components of the matrix P^k are denoted: $(p^{(k)})_{ij}$. We say that the system converges towards a *consensus state* if and only if for any initial state F_0, F_k converges towards the same limit as $k \to \infty$. The author shows that this condition is equivalent to saying that there exists a unique vector $\pi = (\pi_1, \ldots, \pi_N)$ such that:

$$\lim_{k \to \infty} p_{ij}^{(k)} = \pi_j$$

for every i and j. The scalars $\pi_1, \ldots, \pi_N$ are such that:

1. $\pi_j \geq 0 \ \forall j$
2. $\sum_{j=1}^{N} \pi_j = 1$

The consensus F_C will be of the form: $F_C = \sum_{i=1}^{N} \pi_i F_i$.

De Groot proved, using Markov chain theory, that if there exists an integer k such that all of the elements of at least one column of P^k has strictly positive elements, then a consensus will be reached. The asymptotic probability vector π will be the solution to the equation: $\pi P = \pi$.

Subsequent work by Chatterjee and Seneta [CS77] extended the result to the case where the coefficients p_{ij} vary with time. In this case the update rule becomes:

$$F_k = P(k)F_{k-1} = P(k)P(k-1)\dots P(1)F_0$$

where $(P(k))_{k\geq 0} = ((p_{i,j}(k))_{1\leq i,j\leq N})_{k\geq 0}$ is a sequence of row stochastic matrices.

Denote by $U_{r,k}$ the sequence of matrices $U_{r,k} = P(r+k)\dots P(r+2)P(r+1)$. The sequence is said to be *strongly ergodic* if and only if for any $r \geq 0$ there exists a probability vector d_r such that:

$$\lim_{k\to\infty} U_{r,k} = \mathbf{1}d_r^T$$

Strong ergodicity indicates that opinions converge towards a same fixed value.

The authors prove the following theorem that establishes sufficient conditions for consensus:

Theorem 1. *[CS77] Let G_1 be the class of regular stochastic matrices (see appendix A, definition 32). If $\forall r \geq 0$, $\forall k \geq 1$, $U_{r,k} \in G_1$ and $\exists \delta > 0$ such that:*

$$\forall k \geq 1, \quad \min_{i,j}\{p_{i,j}(k) | p_{i,j}(k) \neq 0\} \geq \delta > 0\,,$$

then strong ergodicity is obtained.

The opinion dynamics models seen up until now have common properties: They are linear, and they seek to establish how consensus is reached among all agents.

There are many observed situations in opinion dynamics where instead of reaching a single point of consensus, the agents tend to form clusters of opinions, causing a state of "*polarization*" instead of consensus. One of the first works to explain this phenomena using a nonlinear model was that of Krause in [Kra00]. The author proposed a model where each agent only takes into account the opinions of agents whose opinions are close enough to its opinion. Let $x_1(k), \dots, x_N(k)$ be the current state of the system at time k, for each $i \in \{1, \dots, N\}$ and $\epsilon_i > 0$, define $I(i,k) = \{1 \leq j \leq N | \|x_i(k) - x_j(k)\| < \epsilon_i\}$, the update rule is:

$$x_i(k+1) = \|I(i,k)\|^{-1} \sum_{j\in I(i,k)} x_j(k) \tag{2.2}$$

This update rule imposes that weights not only change with time but that they also depend on the current state of the system. Numerical experiments have

shown that this update rule might drive the system towards a state where many clusters of nodes sharing the same opinion are formed. These clusters remain unchanged afterwards. A mathematical characterization of this phenomena has been given by Blondel et al. in [BHT09], where the authors have established that when $\epsilon_1 = \epsilon_2 = \cdots = \epsilon_N = \epsilon$ we have the following:

Theorem 2. *[BHT09] If the system evolves according to equation 2.2 then for any $i \in \{1, \ldots, N\}$, $x_i(k)$ converges to a limit x_i^* in finite time. Moreover, for any i, j we have either $x_i^* = x_j^*$ or $\|x_i^* - x_j^*\| > \epsilon$*

The theorem states that over a finite amount of time, the opinions converge into clusters whose distance is at least ϵ. The number of clusters depends on the interaction constant ϵ. Clearly if ϵ is very small, there will be very few interactions between agents and thus, there will be more polarization of opinions, conversely if ϵ is large enough, the interaction would become sufficiently strong to pull the system towards a single consensus state.

The works of DeGroot, Chatterjee, Seneta, and Krause were among the pioneering works of distributed gossip algorithms and were aimed towards opinion dynamics. Furthermore, the agents were assumed to be perfectly synchronous, delays due to communications were neglected. A much more general framework has been presented in the work of Tsitsiklis in his Phd report [Tsi84], the next subsection will describe this work and its contributions.

2.3.2 Consensus in decentralized computational systems

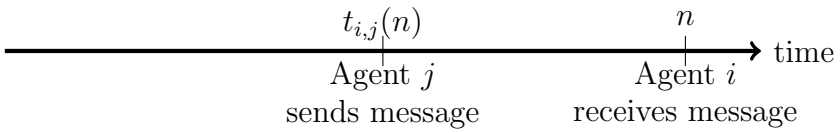
In his PhD report [Tsi84], Tsitsiklis mentions the problem of distributed optimization in a network of agents that are referred to as processors. In addition to having data storage, each agent has a local cost function. The objective for these agents is to agree on a value that minimizes the sum of their local cost functions.

This situation is a generalization of the simple distributed averaging case, in that the local cost functions in the latter are quadratic functions.

Each agent is able to perform computations on its stored value or transmit it to another agent. Time is discretized according to a hypothetical global clock that ticks during each update event (computation performed by an agent). Such a global clock is needed only for analysis purposes, it will serve to describe the scheme used to achieve consensus. This time model differs only slightly from the one seen in section 2.1.3.

At the n-th tick of the global clock, each agent $i \in \{1, \ldots, N\}$ has a pending data point $x_i(n) \in \mathcal{M}$. When agent j sends a message to agent i that arrives at the n-th clock tick, the time the message takes to be transmitted is taken into account by introducing the variable $t_{i,j}(n)$, which is the time when the message was sent. The content of the received message is: $x_j(t_{i,j}(n))$. It is therefore assumed

that $t_{i,j} \leq n$ and $t_{i,i} = n$, and denote by $T^{i,j}$ the set of all times agent i receives a message from agent j. Moreover, Tsitsiklis makes the assumption that $T^{i,j}$ is either empty or infinite. This assumption means that two agents either communicate an infinite amount of time or don't communicate at all, which greatly simplifies the framework.



The entire network of agents is modeled as a directed graph $G = (V, E)$ whose vertices $V = \{1, \ldots, N\}$ correspond to the set of agents. An edge (j, i) belongs to E if and only if $T^{i,j}$ is infinite, i.e. if agent j sends messages to agent i an infinite amount of time. Since we assumed that agents either communicate an infinite number of time or don't communicate at all, this condition is equivalent to having at least one message sent from agent j to agent i.

Since we are interested in this work in the case of distributed consensus, we will restrict ourselves to the following scheme:

1. At iteration n, each agents i updates its current value once it has received all messages to $x_i(n+1)$ through the formula:

$$x_i(n+1) = \sum_{j=1}^{N} a_{ij}(n) x_j(t_{ij}(n))$$

Where the scalars $a_{ij}(n)$ verify:

- For every i, j, n: $a_{ij}(n) \geq 0$
- For every i, n: $\sum_{j=1}^{N} a_{ij}(n) = 1$
- For every i, j, n such that: $i \neq j$, and $n \notin T^{ij}$ we have $a_{ij}(n) = 0$.

This is essentially a scheme similar to that of DeGroot with the addition of communication delays. The coefficients $(a_{i,j}(n))_{i,j,n}$ are averaging weights that change over time. In order to have convergence of the procedure towards a consensus state, further assumptions must be made on the weights:

Assumption 1. *[Tsi84]*

- *G is strongly connected*
- *There exists an $\alpha > 0$ such that:*

- *If agent i receives a message from agent j at time n (i.e. if $n \in T^{ij}$), then $a_{ij} \geq \alpha$.*
- *For every computing agent i, $a_{ii}(n) \geq \alpha$*

These conditions make it so that no message received by an agent receives a weight less than a fixed value α in its value update, it also ensures the past states of that agent have a lasting effect on its future ones.

Assumption 1 ensures that all links of the network of agents are active a sufficient number of times and that data is exchanged with enough regularity so that information is effectively propagated throughout the network.

Finally, the issue of communication delays is dealt with by introducing a uniform bound on all communication delays in the following assumption:

Assumption 2. *[Tsi84] Communication delays are bounded by some $B_0 \geq 0$, i.e. for all i, j and $n \in T^{ij}$ we have $n - t_{ij}(n) \leq B_0$.*

Given assumptions 1 and 2, Tsitsiklis proves the following consensus result:

Theorem 3. *Under all the previous assumptions, the distributed averaging protocol with communication delays converges towards a consensus state.*

The process of attaining consensus does not have to be asymptotic. When the objective is simply to force the agents to agree on a common value, there are protocols that can drive a distributed network of agents to a consensus state in a finite number of iterations. An example of one of these protocols will be the focus of the next subsection.

2.3.3 A word on finite time consensus

The gossip methods seen so far lead to consensus in an asymptotic way, the convergence speed of the algorithm gives an estimate of the number of iterations necessary to get to a state as close as desired to a consensus state. However, if we want to achieve consensus in a finite number of iterations, we need a different approach than the ones seen so far.

In [HSJ15] the authors show that it is possible to achieve consensus in a finite number of iterations for any undirected connected graph, and for directed graphs under certain conditions.

The authors begin by restating the consensus problem in matrix terms. Given a graph $G = (V, E)$ and a stochastic matrix A, we say that A is *compatible* with G if $A_{i,j} > 0$ for $i \neq j$ only if $(j, i) \in E$. We say that A has positive diagonal if $A_{i,i} > 0$ for every i.

Definition 1. *[HSJ15] The sequence of matrices $(A_1, A_2, \ldots, A_T)$ with positive diagonal achieves finite-time consensus on a graph G if A_t is compatible with G for $t = 1, \ldots, T$ and $A_T A_{T-1} \ldots A_2 A_1 = \frac{1}{n}\mathbf{1}\mathbf{1}^T$*

The authors prove the following theorem for undirected graphs:

Theorem 4. *[HSJ15] If G contains a bidirectional spanning tree, then there exists a sequence of at most $\frac{n(n-1)}{2}$ stochastic matrices with positive diagonal that achieves average consensus on G. In particular, finite-time average consensus can be achieved on every undirected graph*

For directed graphs, the authors add another condition to strong connectedness which is that the graphs contain a simple directed cycle with even length.

All the historical approaches seen so far deal with data belonging to a linear vector space. We will see later that this treatment is not sufficient in covering many cases where agents store data that do not belong to a linear space. Subsequent historical approaches have addressed these non linear cases using descriptions and tools from Riemannian geometry. Most of the generalizations to Riemannian manifolds have been built on top of a specific gossip framework which we will present in the next section.

2.4 A generalizable Euclidean gossip framework

In this section we describe a framework for Euclidean gossip in both synchronous and asynchronous cases. We will use this framework to extend gossip to Riemannian data spaces and later in chapter 3 to a special class of general metric spaces, which is our main contribution. We will also explore how the model can be made to accommodate changes in network topology.

2.4.1 Synchronous Gossip

We now go back to the network model defined in the beginning of the chapter. In this section, we study how the gossip protocol relates to the concept of gradient descent, and how we can use that to achieve consensus in both the synchronous case and the asynchronous case.

Consider a network of N identical sensors, and assume that each agent v measures a scalar quantity $x_v \in \mathbb{R}$, set $x(0) = (x_1(0), \ldots, x_N(0))$ and define the cost function:

$$\phi(x(k)) = \frac{1}{2}\sum_{i \sim j}^{N} \|x_i(k) - x_j(k)\|^2. \tag{2.3}$$

The index notation $i \sim j$ means that we are summing over pairs $\{i, j\} \in E$, where E is the set of edges of the communications graph, that is only assumed to be connected. Function ϕ measures how much disagreement is left in the network.

One approach for achieving consensus consists of using the gradient descent on the function ϕ.

$$x_i(k+1) = x_i(k) - \epsilon \operatorname*{grad}_{x_i} \phi(x(k)), \tag{2.4}$$

with $0 < \epsilon < \frac{1}{\deg(G)}$ being the step-size.

This is equivalent to:

$$x_i(k+1) = x_i(k) - \epsilon \sum_{j \in \mathcal{N}(i)} (x_i(k) - x_j(k)) \tag{2.5}$$

Function ϕ is convex, and its global minimum is 0. If the graph is connected, then the minimum is only attained at a consensus state i.e. when: $x_1 = x_2 = \cdots = x_N$.

The update stated above requires that all nodes be synchronized, and that they need to wake up and update simultaneously. At each time step k, each agent $i \in V$ broadcasts its current values to his graph neighbors and receives their values. It then performs the update rule shown in equation (2.5).

The following theorem establishes the convergence of synchronous gossip to a consensus state:

Theorem 5. *[OSFM07] If the process $x(k)$ is generated according to equation 2.5 then:*

$$\lim_n x(n) = \overline{x}(0)\,\mathbf{1}$$

2.4.2 Asynchronous pairwise gossip

In order to relax the strict synchronization requirement, we introduce a stochastic version of gradient descent optimization: The pairwise gossip protocol, which was proposed in [BGPS06]. It is sometimes also referred to as the *epidemic algorithm* since it propagates information in a way similar to how a virus propagates in a biological community [DGH$^+$87, GKG06].

The difference with the previous approach lies in the time and communication models. This setting uses the *asynchronous model* based on independent, identical, Poisson clocks described in section 2.1.3. At a given time slot k, we denote by V_k the agent whose clock ticks. This agent will randomly and uniformly activate a neighbor node W_k to communicate and share data with. Therefore, at time k, the only communicating agents in the whole network are V_k and W_k. A single link is then active at each time, and the other links are not used. The above described time model implies that (V_k, W_k)'s are independent and identically distributed.

We assume that the distribution of V_k is uniform over the network while the distribution of W_k is uniform in the neighborhood of V_k. More precisely, the probability distribution of (V_k, W_k) is given by:

$$\mathbb{P}[V_k = v, W_k = w] = \begin{cases} \frac{1}{N}\frac{1}{\deg(v)} & \text{if } v \sim w \\ 0 & \text{otherwise} \end{cases} \tag{2.6}$$

Notice that this probability is not symmetric in (v, w). It will turn out to be convenient to also consider directly the link $\{V_k, W_k\}$, forgetting which node was the first to wake up and which node was second. In this case $\mathbb{P}[\{V_k, W_k\} = \{v, w\}]$ is of course symmetric in (v, w). One has:

$$\mathbb{P}[\{V_k, W_k\} = \{v, w\}] = \begin{cases} \frac{1}{N}\left(\frac{1}{\deg(v)} + \frac{1}{\deg(w)}\right) & \text{if } v \sim w \\ 0 & \text{otherwise} \end{cases} \tag{2.7}$$

Thus, at each count of the virtual global clock one node v is selected uniformly randomly from the set of agents V; node v then randomly selects a node w from $\mathcal{N}(v)$. Both nodes v and w then compute and update their value to $\frac{x_v + x_w}{2}$.

Denote, $\mathbf{e}_i = [0, \ldots, 0, 1, 0, \ldots, 0]^T \in \mathbb{R}^N$, with 1 being at the i-th position. We can rewrite the update rule in a more concise form:

$$x(k) = M(k)x(k-1)$$

For each time slot k, $M(k)$ is a sequence of iid random matrices such that:

$$M(k) = I_N - \frac{(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^T}{2}$$

The following theorem establishes the convergence of asynchronous gossip to a consensus state at a linear rate:

Theorem 6. *[BGPS06] If $x(k) = (x_{v_1}(k), ..., x_{v_N}(k))$ is the sequence generated by the gossip algorithm. Then we have:*

- *Almost sure convergence*

$$\lim_n x(n) = \overline{x}(0)\mathbf{1} \ a.s. \tag{2.8}$$

- *$\exists L < 0$ such that:*

$$\forall n : \frac{1}{n}\log \mathbb{E}[\|x(n) - \bar{x}(0)\mathbf{1}\|^2] \leq L \tag{2.9}$$

2.4.3 Case of directed time-varying graphs

Before moving to the generalization of gossip to Riemannian manifolds, it is of practical importance to address the issue of what happens when the underlying graph topology changes with time. Taking time variability into account is important to study dynamics of mobile autonomous agents [OSM02, JL+03, BT+07], where new links can form and existing ones disappear.

Given a network with a *time changing* graph $G(t) = (V(t), E(t))$, what conditions on G can still guarantee convergence? and at which rate?

Let $(M_{i,j})$ be a doubly stochastic time-varying matrix, i.e a matrix with nonnegative coefficients each of whose rows and columns sum to 1. We consider a gossip protocol where the update rule can be written as:

$$x_i(t+1) = M_{i,i}(t)x_i(t) + \sum_{j \in \mathcal{N}_i(t)} M_{i,j}(t)x_j(t)$$

the update can also be written more concisely as: $x(t+1) = M(t)x(t)$.

Assumption 3. *[BT+07] Assume that the coefficients $M_{i,j}(t)$ are required to satisfy:*

- $M_{i,i}(t) \geq 0$, $\forall i, t$
- $M_{i,j} = 0$, *if* $i \notin \mathcal{N}_j(t)$ *where* $\mathcal{N}_j(t)$ *is the set of neighbors of* $j \in V$ *at time* t.
- *There exists some* $\alpha > 0$ *such that if* $\{i, j\} \in E(t)$ *then:* $M_{i,j}(t) \geq \alpha$
- *There exists some* B *such that for every* t: $E(t+1) \cup \cdots \cup E(t+B)$ *is strongly connected*

The following theorem from [BT+07] ensures convergence towards a consensus state.

Theorem 7. *[BT+07] There are nonnegative coefficients $\phi_1, \ldots, \phi_N$ such that:*

$$\lim_{t \to \infty} x_i(t) = \sum_{j=1}^{N} \phi_j x_j(0), \forall i$$

This theorem is important since it establishes the robustness of the protocol to changing graph topology and to non bidirectional communications, which are essential constraints of real world distributed systems.

2.4.4 Limits of the Euclidean framework

The gossip algorithm in Euclidean setting is interesting when data can be seen as elements of a vector space, for example temperature or coordinate estimates. It is however inadequate in treating cases where data is non-linear. Take for example a network of cameras, each one having its own orientation. We would like the cameras to agree on a same direction. The data in this case belongs to a space of non Euclidean structure, for example, the 3-dimensional rotations group $SO(3)$. Indeed the elements of $SO(3)$ are rotation matrices, and if we consider two rotations R_1 and R_2 from $SO(3)$ with two different axis, their average $\frac{R_1+R_2}{2}$ is not a rotation, hence does not belong to $SO(3)$. And since different rotations have different rotation axis trying to do a consensus on the rotation angles wont work either. Thus, the arithmetic averaging technique cannot be applied in this setting. The same problem arises if data belongs to a projective space $\mathbb{P}_n$, since two projective lines cannot be added.

In general, when treating non linear data spaces, we need to find a substitute for the notions of addition and averaging. In the next section, we shall deal with a special class of data spaces that were thoroughly studied, and on which we have a series of results: Riemannian manifolds.

2.5 Gossip in Riemannian manifolds

The study of gossip in the context of Riemannian manifolds (see appendix, section B.1 for a definition) has recently received attention. The work of Sarlette and Sepulchre [SS09] use an embedding framework to extend regular gossip first into the unit circle and then into Riemannian spaces. This embedding approach alternates between adding data values in the Euclidean embedding space and re-projecting them in the data space $\mathcal{M}$ again. The notion of centroid of points is also derived from re-projecting the Euclidean centroid onto the Riemannian data space. The authors prove that asymptotic convergence towards a consensus state is achieved provided we have a certain set of conditions on the embedding space. The work of Tron et al [TAV11] proposes an intrinsic approach to consensus optimization in the synchronous case. Their approach relies on the notion of Riemannian gradient and the exponential function (see appendix B.1), that make it possible to define a Riemannian version of the gradient descent protocol. Finally, we shall look at an intrinsic asynchronous case that was first studied by Bonnabel in [Bon13] which uses similar notions as Tron et al but with stochastic gradient descent, in order to accommodate for agents asynchronism.

2.5.1 Consensus in circular data

One of the first extensions of gossip to non-Euclidean data spaces is that of circular data by Sarlette *et al.* [STBS08].

Consider N agents whose stored data belong to the circle S^1. The unit circle is 2π periodic which makes it fundamentally different from vector spaces. Each element of the circle can be uniquely represented by an angle $\theta \in [0, 2\pi[$. A candidate for consensus protocol can be obtain by replacing the Euclidean update rule:

$$x_i(k+1) = x_i(k) + \sum_{j \in \mathcal{N}_i} a_{ij}(x_i(k) - x_j(k))$$

With the following update rule:

$$\theta_i(k+1) = \theta_i(k) + \sum_{j \in \mathcal{N}_i} a_{ij} \sin(\theta_i(k) - \theta_j(k))$$

Defining $z_u(k) = \exp(i\theta_u(k))$ the update rule is equivalent to:

$$z_u(k+1) = z_u(k) + \mathrm{Proj}_{z_u(k)} \left(\sum_{v \in \mathcal{N}_u} a_{uv}(z_u(k) - z_v(k)) \right)$$

where $\mathrm{Proj}_{z_u(k)}(x)$ is the projection of $x \in \mathbb{C}$ onto the tangent to the unit circle at $z_u(k)$: $\mathrm{Proj}_{z_u(k)}(x) = iz_u(k)\langle z_u(k), x\rangle$.

Proposition 1. *[STBS08] Consider a set of N agents evolving on S^1 according to the above update rule. If the underlying communications graph is connected and all agents are initially located within an open semicircle, then they converge towards a consensus state.*

While local consensus is achievable using this protocol, for global consensus, a modified version of the pairwise gossip approach is more adapted.

This algorithm achieves global consensus on the unit circle when the underlying communications network between agents is a connected undirected graph.

In [SS09] the authors show that the consensus results on circular data can be extended to the case of data belonging to a compact, connected homogeneous manifold.

The approach taken by the authors is to reformulate the consensus problem as a minimization problem of an appropriate cost function. The authors embed the data space $\mathcal{M}$ in a Euclidean space $\mathbb{R}^m$ of sufficiently large dimension m. Then they define a cost function ϕ on $\mathbb{R}^m$ that they link the notion of centroid of points. Consensus is no longer described as merely a state where all agents have the same value, instead it is redefined using the notion of *Induced arithmetic mean.* We shall see in the next section how these ideas are implemented.

Algorithm Gossip on the circle

Input: a graph $G = (V, E)$ and the initial nodes configuration $\theta_v(0), v \in V$
for all $k > 0$ **do**
At instant k, uniformly randomly choose a node V_k from V and a node W_k uniformly randomly from $\mathcal{N}(V_k)$.
Update:
$\theta_{V_k}(k) = \arg\left(\frac{\exp(i\theta_{V_k}(k-1))+\exp(i\theta_{W_k}(k-1))}{2}\right)$
$\theta_{W_k}(k) = \arg\left(\frac{\exp(i\theta_{V_k}(k-1))+\exp(i\theta_{W_k}(k-1))}{2}\right)$
$\theta_v(k) = \theta_v(k-1)$ for $v \notin \{V_k, W_k\}$
end for

2.5.2 Extension to Riemannian manifolds

Assume now that the data space is a Riemannian manifold $(\mathcal{M}, \langle , \rangle)$ (see appendix B.1 definitions 35 and 43)and that we place ourselves in the synchronous communications regime. Each agent $v \in V$ has an initial measurement $x_v(0) \in \mathcal{M}$. We want a procedure that drives the agents towards a consensus state $(x_\infty, \ldots, x_\infty) \in \mathcal{M}^N$.

One way to achieve this, is to use methods of optimization. Recall that in the Euclidean setting, one could consider the function:

$$\phi(x_1, \ldots, x_N) = \sum_{i=1}^{N} d^2(x_i, \bar{x})$$

with $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i(0)$ and try to minimize it on $\mathbb{R}^N$ using tools of distributed optimization. Gradient descent writes:

$$\frac{dx_k}{dt}(t) = -\epsilon \operatorname*{grad}_{x_k}(\phi)(x_1, \ldots, x_N)$$

where $\alpha > 0$ is the *step size* of the gradient descent. In discrete time, the algorithm becomes:

$$x_k(t+1) = x_k(t) - \epsilon \operatorname*{grad}_{x_k(t)}(\phi)(x_1(t), \ldots, x_N(t)) \ .$$

In a Riemannian setting, while the notion of a Riemannian gradient exists, the addition operation is not always well defined. In [SS09], the authors circumvent this problem by embedding the data space in a Euclidean vector space of sufficiently large dimension, making the usual gradient descent update using Euclidean gradient and then re-projecting the result on the original data space. This is possible via the *embedding theorem*:

Theorem 8. *[HH06] Let $\mathcal{M}$ be a Riemannian manifold of dimension n, then there exists an integer m and a $\mathcal{C}^\infty$ isometry $f : \mathcal{M} \to \mathbb{R}^m$ compatible with the metric $\langle , \rangle$ and the standard dot product on $\mathbb{R}^m$ i.e. $\forall p \in \mathcal{M}\ \langle u, v \rangle_p = df_p(u).df_p(v)$*

This theorem essentially states that any n-dimensional manifold can be isometrically embedded in a Euclidean space $\mathbb{R}^m$ for m sufficiently large. This embedding will allow us to use Euclidean distance to find a suitable cost function on which we could apply distributed gradient descent. We assume also that $\mathcal{M}$ is a *homogeneous* manifold i.e. a manifold with a transitive action by some Lie group $\mathcal{G}$ (see appendix section B.1.1), the idea of homogeneity allows to focus on the relative positions between agents rather than their absolute positions on the manifold.

Assumption 4. *[SS09] $\mathcal{M}$ is a connected compact homogeneous manifold smoothly embedded in $\mathbb{R}^m$ with the Euclidean norm $\|y\| = r_{\mathcal{M}}$ constant over $y \in \mathcal{M}$. The Lie group $\mathcal{G}$ acts as a subgroup of the orthogonal group on $\mathbb{R}^m$*

First, we define an equivalent for the notion of arithmetic average $\bar{x}$ of a sequence of points $x_1 \dots x_N$:

Definition 2. *[SS09] The induced arithmetic mean $(IAM) \subset \mathcal{M}$ of N agents of weights $w_k > 0$ and positions $x_k \in \mathcal{M}$, $k = 1 \dots N$, is the set of points that globally minimize the weighted sum of squared Euclidean distances in $\mathbb{R}^m$*

$$\bar{x} = \arg\min_{c \in \mathcal{M}} \sum_{k=1}^{N} w_k d^2_{\mathbb{R}^n}(x_k, c)$$

This new definition of centroid, while analogous to that used in Euclidean geometry, does not necessarily reduce to a single point. Indeed in the context of Riemannian manifold the induced arithmetic can be a subset of $\mathcal{M}$ with ;ore than one element.

Definition 3. *The Euclidean centroid $C_e \in \mathbb{R}^m$ of N weighted agents located on $\mathcal{M}$ is:*

$$C_e = \frac{1}{W} \sum_{k=1}^{N} w_k x_k \in \mathbb{R}^m$$

where

$$W = \sum_{k=1}^{N} w_k$$

The sum here is the usual sum in $\mathbb{R}^m$.

The centroid is linked to the IAM by the relationship:

$$IAM = \arg\max_{c \in \mathcal{M}}(c^T C_e)$$

Thus, computing the IAM involves searching for a global maximum of a linear function. If all local maxima of the function were global maxima, then we could use local optimization methods to compute the IAM. Thus, we make the following assumption:

Remark 1. Calculating $\bar{x}$ reduces to calculating the global minima of a linear function. To make the computation even easier; we can assume that the manifold $\mathcal{M}$ is such that the local minima of any linear function are also global minima.

Assumption 5. *[SS09] The local minima of any linear function $f(c) = c^T b$ over $c \in \mathcal{M}$ with b fixed in $\mathbb{R}^n$ are all global minima.*

Since the notion of barycenter is no longer unique, [SS09] updates the notion of consensus.

Definition 4. *A consensus configuration $(x_1, \ldots, x_N)$ on a graph $G = (V, E)$ and a manifold $\mathcal{M}$ is such that:*

$$x_k \in \arg\max \left(c^T \sum_{j=1}^{N} \frac{1}{N} x_j \right) \forall 1 \leq k \leq N$$

A synchronized *configuration $(x_1, \ldots, x_N)$ verifies $x_1 = x_2 = \cdots = x_N$*

In order to transport the tools of distributed optimization to Riemannian manifold, we need a new cost function. By analogy with previous studies on circular data [SSL06], the cost function ϕ is defined as:

$$\phi(x_1 \ldots, x_N) = \frac{1}{2N^2} \sum_{k=1}^{N} \sum_{j=1}^{N} x_j^T x_k$$

In the case of undirected graphs, a gradient flow for ϕ with step size $\alpha > 0$ is:

$$\frac{dx_k}{dt}(t) = -2N^2 \alpha \operatorname*{grad}_{k,\mathcal{M}}(\phi)(x_1, \ldots, x_N)$$

Where $\operatorname{grad}_{k,\mathcal{M}}(\phi)$ is the k-th component of the riemannian gradient of ϕ. We can obtain this gradient from the Euclidean gradient $\operatorname{grad}_{k,\mathbb{R}^m}(\phi)(x_1, \ldots, x_N) = \frac{1}{N^2} \sum_j x_j$.

$$\frac{dx_k}{dt} = -2\alpha N \mathrm{Proj}_{T\mathcal{M},x_k}(C_e(t) - x_k)$$

where $\text{Proj}_{T\mathcal{M},x_k}$ is the projection operator on the tangent space $T_{x_k}\mathcal{M}$ at x_k. The discrete time algorithm has the form:

$$x_k(t+1) = x_k(t) - 2\alpha N \text{Proj}_{T\mathcal{M},x_k}(C_e(t) - x_k(t))$$

In order to characterize the convergence properties of the algorithm, we need the following definitions.

Definition 5 (Lyapunov stability)**.** *[HSD12] X_∞ is a* stable equilibrium *if for every neighborhood U of X_∞ there is a neighborhood $\mathcal{O} \subset U$ of X_∞ such that every $X(t)$ starting in $\mathcal{O}$ remains in U for all $t \geq 0$*

Definition 6 (Asymptotic stability)**.** *[HSD12] An equilibrium $X_\infty \in \mathcal{M}^N$ is* asymptotically stable *if it is Lyapunov stable and additionally $\mathcal{O}$ can be chosen so that: $d(X(t), X_\infty) \to 0$ as $t \to \infty$ for any initial $X(0)$.*

The following proposition, whose proof can be found in [SS09], establishes asymptotic convergence of the gradient descent algorithm.

Proposition 2. *[SS09] If the manifold $\mathcal{M}$ satisfies assumptions 4 and 5 and if $\alpha > 0$, then all stable equilibrium are consensus configurations.*

If the graph $G = (V, E)$ is an undirected, connected graph, then stable equilibrium are synchronized states.

While this approach leads to consensus in a computationally efficient way, it remains highly dependent on the embedding of the manifold, on which additional conditions are imposed. It guarantees only asymptotic stability which is different from convergence.

An intrinsic approach, using only the properties of the manifold $\mathcal{M}$, addresses these issues. This is object of the following section.

2.5.3 The intrinsic approach

The synchronous case

Let $\mathcal{M}$ be a connected, geodesically complete Riemannian manifold (see definition 44). This implies via Hopf-Rinow theorem [Cha06, p.26] that between any two points there exists a minimal geodesic. The unique geodesic γ such that: $\gamma(0) = p$ and $\dot{\gamma}(0) = X$, can be written in the form: $\gamma(t) = \exp_p(tX)$.

We would like to find the minima of a smooth function $\phi : \mathcal{M} \to \mathbb{R}$. Given an initial point $x_0 \in \mathcal{M}$, it is possible to define an iterative algorithm that is purely intrinsic using the notions of geodesic and the exponential function. The algorithm would generate a sequence $x(k)$ such that:

$$\begin{cases} x(0) = & x_0 \\ x(k+1) = & \exp_{x(k)}(-\epsilon_k \mathbf{D}_k) \end{cases}$$

with $\mathbf{D}_k$ the direction of descent and $\epsilon(k)$ the (possibly time-varying) step size of the descent.

The method used in determining the direction $\mathbf{D}_k$ and stepsize ϵ_k will determine the convergence proprieties of the sequences $x(k)$ and $f(x(k))$.

Following the Euclidean gradient descent approach, [TAV11] chooses a direction verifying:

$$\langle \operatorname{grad} f(x(k)), \mathbf{D}_k \rangle < 0$$

We then determine $\epsilon(k)$ such that: $f(\exp_{x(k)}(\epsilon(k)\mathbf{D}_k)) < f(x(k+1))$ and set $x(k+1) = \exp_{x(k)}(\epsilon_k \mathbf{D}_k)$. This iterative process continues until we have satisfied at least one *stopping criteria*, which are of the form:

$$\| \operatorname{grad}(x(k)) \| \leq \eta; |f(x(k+1)) - f(x(k))| \leq \eta$$

.

In the *steepest descent* method, we choose $\mathbf{D}_k = -\operatorname{grad} f(x(k))$:

$$x(k+1) = \exp_{x(k)}(-\epsilon_k \operatorname{grad} f(x(k)))$$

where ϵ_k is chosen so that $f(x(k+1)) < f(x(k))$. If $\operatorname{grad} f(x(k)) = 0$ then we stop the algorithm.

In order to apply the gradient descent algorithm to solve the consensus problem, we use the following cost function:

$$\phi : \mathcal{M}^N \to \mathbb{R} \qquad (x_1 \ldots, x_N) \to \sum_{i \sim j} d^2(x_i, x_j)$$

The gradient of ϕ with respect to the i-th component is [TAV11]:

$$\operatorname*{grad}_{x_i} \phi = \frac{1}{2} \operatorname*{grad}_{x_i} \sum_{j \in \mathcal{N}(i)} d^2(x_i, x_j) = - \sum_{j \in \mathcal{N}(i)} \exp_{x_i}^{-1}(x_j).$$

The idea is to use a distributed version of the Riemannian gradient descent with a constant step-size $\epsilon > 0$, by iterating the following updates for each node $i \in \{1 \ldots N\}$ in parallel:

$$x_i(k+1) = \exp_{x_i(k)}(\epsilon \operatorname*{grad}_{x_i} \phi(x_i(k)))$$

When $\mathcal{M} = \mathbb{R}$ and the metric is the standard metric, we obtain the distributed Euclidean gradient descent.

We also want a constant step size $\epsilon > 0$, it has to ensure that: $\phi(x(k+1)) < \phi(x(k))$ for all $k \geq 0$.

Definition 7. *We say that a function ϕ is of uniformly bounded Hessian if and only if there exists a constant M such that for any $x_0 \in \mathcal{M}$, $v \in T_{x_0}\mathcal{M}$, and $\gamma_{x_0}(t) = \exp_{x_0}(tv)$ we have:*

$$\frac{d^2\phi}{dt^2} \leq M\|v\|^2$$

In order to see if such a step size exists, we will use the following lemma:

Lemma 1. *[Bon13] For any function $f : \mathcal{M} \to \mathbb{R}$ twice continuously differentiable, $p \in \mathcal{M}$ and $v \in T_p\mathcal{M}$ we have:*

$$f(\exp_p(tv)) - f(p) \leq t\left\langle v, \operatorname{grad} f(p)\right\rangle_p + \frac{t^2}{2}\|v\|^2 M_f$$

where M_f is a bound on the hessian of the function f.

The following proposition characterizes the admissible values of ϵ:

Proposition 3. *[TAV11] Assume $f : \mathcal{M} \to \mathbb{R}$ is a twice differentiable function with bounded Hessian. Let $x_0 \in \mathcal{M}$ and $\gamma_{x_0}(t) = \exp(-t \operatorname{grad} f(x_0))$.*

If $t \in \left(0, \frac{2}{M_f}\right)$ (with M_f being the uniform bound on the Hessian of f), then we have $f(\gamma_{x_0}(t)) - f(x_0) \leq 0$

Proof. Applying the previous lemma with $v = -\operatorname{grad} f(x_0)$ we get:

$$f(\exp(-t \operatorname{grad} f(x_0))) - f(x_0) \leq -t\left\langle \operatorname{grad} f(x_0), \operatorname{grad} f(x_0)\right\rangle_{x_0} + \frac{t^2}{2}\|v\|^2 M_f$$

Thus:

$$f(x_0) - f(\exp(-t \operatorname{grad} f(x_0))) \geq t\|\operatorname{grad} f(x_0)\|^2\left(1 - \frac{M_f t}{2}\right)$$

The right hand side of the inequality is positive when $t \leq \frac{2}{M_f}$ which implies that: $f(x_0) \geq f(\exp(-t \operatorname{grad} f(x_0)))$. □

In order to have a uniform bound on the cost function $\phi(x_1, \ldots, x_N) = \sum_{i\sim j} d^2(x_i, x_j)$, we make the additional assumption that $\mathcal{M}$ is a manifold of bounded curvature, this gives us the following uniform bound (see: Tron et al [TAV11]):

Proposition 4. *[TAV11] Assume that $\mathcal{M}$ is a manifold of bounded curvature, with curvature bounds δ and Δ. Then a universal bound on the Hessian of the function ϕ on the domain: $\mathcal{D} = \{(x_1, \ldots, x_N) \in \mathcal{M}^N : d(x_i, x_j) < r_{\mathcal{M}}\}$ is:*

$$M_\phi = \deg(G) \max\left\{2, r_{\mathcal{M}}\left(\frac{C_\delta(r_{\mathcal{M}})}{S_\delta(r_{\mathcal{M}})} + \frac{1}{S_\Delta(r_{\mathcal{M}})}\right)\right\}$$

Function S_κ is defined such that: $S_\kappa(t) = \frac{1}{\sqrt{\kappa}}\sin(\sqrt{\kappa}t)$ when $\kappa > 0$, $S_\kappa(t) = t$ when $\kappa = 0$, and $S_\kappa(t) = \frac{1}{\sqrt{\|\kappa\|}}\sinh(\sqrt{\|\kappa\|}t)$ when $\kappa < 0$.

Function C_κ is defined such that: $C_\kappa = \cos(\sqrt{\kappa}t)$ when $\kappa > 0$, $C_\kappa = 1$ when $\kappa = 0$, and $C_\kappa = \cosh(\sqrt{\kappa}t)$ when $\kappa < 0$.

The following theorem establishes the local convergence of the gradient descent algorithm.

Theorem 9. *[TAV11] Assume that $\mathcal{M}$ is a Riemannian manifold with bounded curvature. Let $\mathcal{S}_{\text{conv}}$ be the set defined as $\mathcal{S}_{\text{conv}} = \{X = (x_1, \ldots, x_N) \in \mathcal{M} | \phi(X) \leq \frac{(r_{\mathcal{M}})^2}{2\,\text{diam}(G)}\}$.*

If the initial set of points $X(0) = (x_1(0), \ldots, x_N(0))$ is initialized in $\mathcal{S}_{\text{conv}}$, then the gradient descent sequence $(X(k))_{k\in\mathbb{N}}$ converges towards a consensus state.

For a complete proof, see [TAV11].

This gradient algorithm is a synchronous one since updates need be performed simultaneously. We now follow the work of [Bon13] for the asynchronous case.

The asynchronous case

The objective of stochastic optimization consists of minimizing a cost function $C(x) = \mathbb{E}_z H(z, x) = \int H(z, x) dP(z)$ which we assume here to be three times continuously differentiable with respect to x, where x is a minimization parameter belonging to a Riemannian manifold $\mathcal{M}$. $H(z, x)$ is such that $\mathbb{E}_z H(z, x) = \int H(z, x) dP(z) = \nabla C(x)$ and dP is is assumed to be unknown.

The gradient descent update rule here has the form:

$$x_{k+1} = \exp_{x_k}(-\epsilon_k H(z_k, x_k))$$

where ϵ_t is the time dependent stepsize which assumed to verify:

$\sum \epsilon_k^2 < \infty$ and $\sum \epsilon_k = +\infty$

We have the following convergence theorem:

Theorem 10. *[Bon13] Suppose the stochastic gradient descent is applied on a connected Riemannian manifold $\mathcal{M}$ such that $\text{inj}(M) > 0$, and the stepsize verify the relations described above. Assume that there exists a compact $\mathcal{K}$ such that $w_k \in \mathcal{K}$ for all $k \geq 0$. Assume also that $H(x, w)$ is bounded on $\mathcal{K}$, meaning that $\exists A > 0$ such that $\forall z$, and $\forall x \in \mathcal{K}$: $\|H(x, w)\| \leq A$.*

Then $C(x_k)$ converges almost surely and $\lim_{k\to\infty} \text{grad}\, C(x_k) = 0$.

In the stochastic version of the algorithm we use the Poisson clock model previously used for the asynchronous Euclidean gossip. At each count of the virtual global clock one node v is selected uniformly randomly from the set of agents V. The node v then randomly selects a node w from $\mathcal{N}(v)$ and both nodes make the update:

$$x_v(k+1) = \exp_{x_v(k)}(\epsilon_k \exp^{-1}_{x_v(k)}(x_w(k)))$$

$$x_w(k+1) = \exp_{x_w(k)}(\epsilon_k \exp^{-1}_{x_w(k)}(x_v(k)))$$

Meaning that the two agents advance towards each other along the geodesic $[x_v(k), x_w(k)]$ with a small stepsize ϵ_k. This is equivalent to applying the stochastic descent algorithm on the cost function:

$$C(x_1 \ldots, x_N) = \sum_{v \sim w} \mathbb{P}[v, w] d^2(x_v, x_w)$$

And:

$$H(x(k), V_k, W_k) = \sum_{v \sim w} \mathbb{1}_{\{V_k = v, W_k = w\}} d^2(x_v, x_w)$$

2.6 Conclusion

In this chapter, we presented distributed gossip in the case of Euclidean and Riemannian data spaces in both synchronous and asynchronous cases. While Euclidean gossip provides satisfactory convergence results in synchronous as well as asynchronous cases, it is limited in its scope of application. This was the reason for introducing the Riemannian framework. However, convergence results for the Riemannian framework were not entirely satisfactory. Indeed, for the embedding approach, only an asymptotic stability result has been established with strong constraints on the embedding space. The intrinsic approach works well for the synchronous case, but in the asynchronous case we have only convergence in the case of non positive sectional curvature which limits the applicability of the result. A clear statement concerning convergence speed is also lacking.

Chapter 3

Our extension of gossip to $CAT(\kappa)$ metric spaces

3.1 Motivation

In this chapter, we present our approach in generalizing the gossip algorithm to nonlinear spaces. We advocate that the *CAT(κ) metric space framework* generalizes and simplifies previous work on the consensus problem in non Euclidean spaces by avoiding usage of differential geometry tools. We mainly address the gossip problem in the asynchronous case by proposing a distributed gossip protocol: the Random Pairwise Midpoint (RPM) algorithm. We establish convergence to consensus state for data spaces with positive as well as negative sectional curvature bounds. Convergence speed at a linear rate is also established and tested via numerical simulations.

3.2 Generalities on metric spaces

3.2.1 First definitions

Let $\mathcal{M}$ be some set. A function $d : \mathcal{M}^2 \to \mathbb{R}$ is called a *distance function* if and only if the following conditions hold:

- $d(x, y) \geq 0$ for all $x, y \in \mathcal{M}$
- $\forall\ x, y \in \mathcal{M}$: $d(x, y) = 0$ if and only if $x = y$
- $\forall\ x, y \in \mathcal{M}\ d(x, y) = d(y, x)$
- $\forall\ x, y, z \in \mathcal{M}\ d(x, z) \leq d(x, y) + d(y, z)$

The set $\mathcal{M}$ equipped with a distance function is called a *metric space* and is denoted by $(\mathcal{M}, d)$.

Definition 8 (Geodesic, Segments)**.** *A path $c : [0, l] \to \mathcal{M}$, $l \geq 0$ is said to be a* geodesic *if $d(c(t), c(t')) = |t - t'|$, for all $(t, t') \in [0, l]^2$; $x = c(0)$ and $y = c(1)$ are the* endpoints *of the geodesic and $l = d(x, y)$ is the* length *of the geodesic. The image of c is called a* geodesic segment *with endpoints x and y. If there is a single segment with endpoints x and y, it is denoted $[x, y]$.*

Definition 9 (Midpoint)**.** *The* midpoint *of a segment $[x, y]$ is denoted $\frac{x+y}{2}$, it is defined as the unique point m such that $d(x, m) = d(y, m) = d(x, y)/2$.*

Next, we shall construct the metric spaces $\mathcal{M}_\kappa^n$ which will serve as standard models against which one can compare more general metric spaces. They will play a central role in defining the notion of $CAT(\kappa)$ metric spaces.

Model space $\mathcal{M}_\kappa^n$

In order to properly define the model space, we need three prototype spaces: the Euclidean space, the sphere and the hyperbolic space. General model spaces are then simply derived by dilation.

Let us denote $\mathcal{E}^n$ the vector space $\mathbb{R}^n$ equipped with its standard Euclidean norm $\|x\|^2 = \sum_{i=0}^{n-1} x_i^2$ with $x = (x_0, \ldots, x_{n-1}) \in \mathcal{E}^n$. Denote

$$\mathcal{S}^n = \{(x_0, \ldots, x_n) \in \mathcal{E}^{n+1} : \sum_{i=0}^{n} x_i^2 = 1\},$$

the n-dimensional unit Euclidean sphere and

$$\mathcal{H}^n = \{(x_0, \ldots, x_n) \in \mathcal{E}^{n+1} : (-x_0^2 + \sum_{i=1}^{n} x_i^2) = -1, x_0 > 0\},$$

one sheet of a two-sheets n-dimensional hyperboloid. As a metric space $\mathcal{E}^n$ is equipped with distance $d_E(x, y) = \|x - y\|^2$, $\mathcal{S}_\kappa^n$ is equipped with distance $0 \leq d_S(x, y) \leq \pi$ such that

$$\cos(d_S(x, y)) = \sum_{i=0}^{n} x_i y_i,$$

and $\mathcal{H}_n$ is equipped with distance $d_H(x, y) \geq 0$ such that:

$$\cosh(d_H(x, y)) = x_0 y_0 - \sum_{i=1}^{n} x_i y_i .$$

Remark 2. Function d_S is well defined since for $(x,y) \in (\mathcal{S}^n)^2$, $-1 \leq \sum x_i y_i \leq 1$. Note that if $(x_0,\ldots,x_n) \in \mathcal{H}^n$ then necessarily $x_0 \geq 1$. For all $(x,y) \in \mathcal{H}^n$,

$$\sum_{i=1}^{n} x_i y_i \leq \left(\sum_{i=1}^{n} x_i^2\right)^{1/2} \left(\sum_{i=1}^{n} y_i^2\right)^{1/2}$$

$$\leq (x_0^2-1)^{1/2}(y_0^2-1)^{1/2} \leq x_0 y_0 \ .$$

Thus, function d_H is well defined. We refer the reader to [BH99, chap I.2] for the proof that d_S and d_H satisfy the requirements of distance functions (triangle inequality is not obvious).

In what follows we also use notation $r_\kappa = \frac{D_\kappa}{2}$.

Definition 10 (Comparison Triangles in $\mathcal{M}_\kappa^n$)**.** *Assume p,q and r are three points in $\mathcal{M}$. A comparison triangle in $\mathcal{M}_\kappa^n$ refers to any three points, provided there exist, $\bar{p},\bar{q}$ and $\bar{r}$ in $\mathcal{M}_\kappa^n$ such that $\bar{d}(\bar{p},\bar{q}) = d(p,q)$, $\bar{d}(\bar{q},\bar{r}) = d(q,r)$, and $\bar{d}(\bar{r},\bar{p}) = d(r,p)$.*

Concerning the existence and uniqueness of such comparison triangles, we provide without proof the following proposition (see for instance [BH99] for a proof) .

Proposition 5. *[BH99, chap I.2] Assume p,q and r are three points in $\mathcal{M}$ such that $d(p,q)+d(q,r)+d(r,p) < 2D_\kappa$ and $max(d(p,q),d(q,r),d(r,p)) \leq D_\kappa$. Then there exists a comparison triangle in $\mathcal{M}_\kappa^n$. Moreover, this comparison triangle is unique up to an isometry.*

Remark 3. Note that the proposition is straightforward for $\kappa = 0$, where triangle inequality is a necessary and sufficient condition for the There is also a notion of angle in this "metric" context, as illustrated by the next definition.

Definition 11 (Alexandrov Angle)**.** *Assume $c : [0,l] \to \mathcal{M}$ and $c' : [0,l'] \to \mathcal{M}$ are two geodesics such that $x = c(0) = c'(0)$, $y = c(l)$ and $z = c'(l')$. For each $0 \leq t \leq l$ and $0 \leq t' \leq l'$, consider a comparison triangle $(\bar{x},\bar{y}_t,\bar{z}_{t'})$ in $\mathcal{M}_0^2$ for the triplet $(x,c(t),c'(t'))$. The angle between c and c' at x, denoted by $\angle(c,c')$, is defined by:*

$$\angle(c,c') = \lim_{\epsilon \to 0} \sup_{0 \leq t,t' \leq \epsilon} \bar{\angle}_{\bar{x}}([\bar{x},\bar{y}_t],[\bar{x},\bar{z}_{t'}])$$

where $\bar{\angle}_{\bar{x}}([\bar{x},\bar{y}_t],[\bar{x},\bar{z}_{t'}])$ denotes the angle at $\bar{x}$ of the comparison triangle $(\bar{x},\bar{y}_t,\bar{z}_{t'})$.

3.2.2 $CAT(\kappa)$ metric spaces

The concepts from metric triangle geometry presented in the previous subsections yield the following definition of $CAT(\kappa)$ spaces.

Definition 12 ($CAT(\kappa)$ inequality)**.** *Assume $(\mathcal{M}, d)$ is a metric space and $\Delta = (c_0, c_1, c_2)$ is a geodesic triangle with vertices $p = c_0(0)$, $q = c_1(0)$ and $r = c_2(0)$ and with perimeter strictly less than $2D_\kappa$. Let $\bar{\Delta} = (\bar{p}, \bar{q}, \bar{r})$ denote a comparison triangle in $\mathcal{M}^2_\kappa$. Δ is said to satisfy the $CAT(\kappa)$ inequality if for any $x = c_0(t)$ and $y = c_2(t')$, one has:*

$$d(x, y) \leq \bar{d}(\bar{x}, \bar{y})$$

where $\bar{x}$ is the unique point of $[\bar{p}, \bar{q}]$ such that $d(p, x) = \bar{d}(\bar{p}, \bar{x})$ and $\bar{y}$ on $[\bar{p}, \bar{r}]$ such that $d(p, y) = \bar{d}(\bar{p}, \bar{y})$.

Remark 4. Applying this inequality to the case where $\kappa \leq 0$ and $d(q, r) = 0$, the uniqueness of geodesics is recovered when $\kappa \leq 0$.

Definition 13 ($CAT(\kappa)$ metric space)**.** *A metric space $(\mathcal{M}, d)$ is said to be $CAT(\kappa)$ if every geodesic triangle with perimeter less than $2D_\kappa$ satisfies the $CAT(\kappa)$ inequality.*

Proposition 6. *If x and y are two points in a $CAT(0)$ space, there is a unique geodesic $[x, y]$ and the midpoint $\frac{x+y}{2}$ is always well defined and unique.*

One has the so-called Bruhat-Tits inequality [BH99], which is a straightforward application of the $CAT(0)$ inequality:

Proposition 7. *[BH99] Assume $(\mathcal{M}, d)$ is $CAT(0)$, and Δ is a geodesic triangle with vertices (p, q, r) such that m is the midpoint of q and r along the side of the triangle. Then,*

$$2d(p, m)^2 \leq d(p, q)^2 + d(p, r)^2 - d(q, r)^2/2 \tag{3.1}$$

Remark 5. Note that $CAT(0)$ inequality does apply to all geodesic triangles since the diameter restriction is void when $\kappa \leq 0$.

An important notion for studying consensus algorithm is that of convexity. We define the notion of convex sets of metric spaces and study how convexity interacts with curvature.

Convex Sets

Convexity can have several meanings in the context of metric spaces (*cf.*, *e.g.* [Cha06, p.403]).

Definition 14 (Convexity)**.** *A subset S of $\mathcal{M}$ is said to be* convex *when for every couple of points $(x, y) \in S^2$, every geodesic segment γ joining x and y in $(\mathcal{M}, d)$ is such that $\gamma \subset S$.*

The notion of convex hull is going to be useful in the sequel.

Definition 15 (Convex Hull)**.** *Assume S is a subset of $\mathcal{M}$. Then the* convex hull *of S, denoted* $\mathrm{conv}(S)$*, is the intersection of all closed convex sets containing S.*

One can easily check that $\mathrm{conv}(S)$ is indeed convex (and closed).

Proposition 8. *If $(\mathcal{M}, d)$ is $CAT(0)$, then for each $x_0 \in \mathcal{M}$ and $r \geq 0$, every ball $B_{x_0,r} = \{x \in \mathcal{M} : d(x, x_0) \leq r\}$ is convex.*

Proof. Consider $x, y \in B_{x_0,r}$ and a comparison triangle $\Delta(x_0, x, y)$. Then for each $z \in [x, y]$, $CAT(0)$ inequality implies $d(x_0, z) \leq \max(d(x_0, x), d(x_0, y)) \leq r$. Hence $z \in B_{x_0,r}$, which finishes the proof. □

Proposition 9 ([BH99][prop. II.1.4)**.** *] Let $\mathcal{M}$ denote a $CAT(\kappa)$ metric space.*

1. *If x and y in $\mathcal{M}$ are such that $d(x, y) < D_\kappa$, there exists a unique geodesic $[x, y]$ joining them.*
2. *For any $x \in \mathcal{M}$, the ball $B_{x,r}$ with $r < r_\kappa$ is convex.*

Remark 6. $CAT(\kappa)$ metric spaces appear in many situations:

- An example of a data space where $CAT(0)$ geometries involved is the space of phylogenetic tree (see [BHV01]). $CAT(0)$ geometry also appears in the context of reconfigurable systems where the space of all possible configurations of the system can be modeled by a $CAT(0)$ cubical complex [ABY12].
- Examples of $CAT(\kappa)$ spaces with $\kappa > 0$ are: The projective space, the Grassmanians, and the group of N-dimensional rotations. These spaces appear frequently in computer vision [SBS10, Lui12, TVT08].

For notational convenience, define the functions: $C_\kappa(t) = \cos(\sqrt{\kappa}t)$, $S_\kappa(t) = \frac{\sin(\sqrt{\kappa}t)}{\sqrt{\kappa}}$, $\chi_\kappa(t) = 1 - C_\kappa(t)$.

Lemma 2 (Law of Cosines)**.** *Given a complete manifold $\mathcal{M}^n_\kappa$ with constant sectional curvature κ and a geodesic triangle $\Delta(pqr)$ in $\mathcal{M}^n_\kappa$, assume $\max\{d(p, r), d(q, r), d(p, q)\} < r$ and let $\alpha := \widehat{prq}$. We have:*

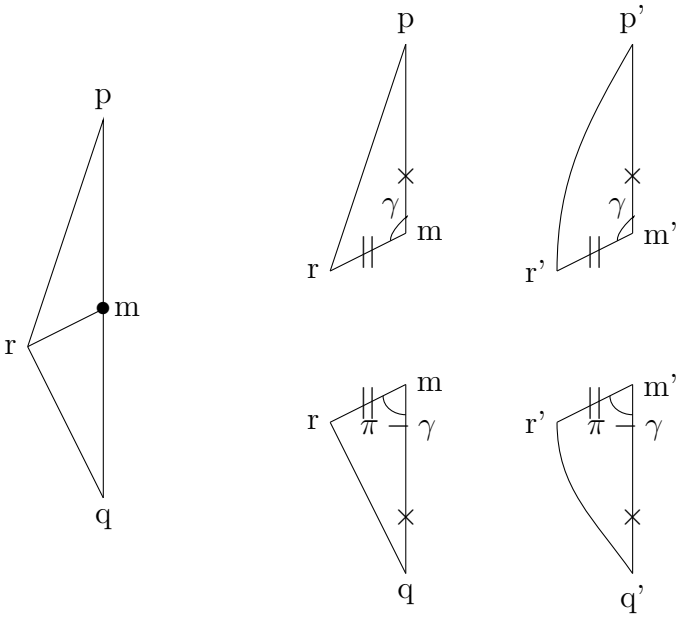
$$C_\kappa(d(p, q)) = C_\kappa(d(p, r))C_\kappa(d(q, r)) + S_\kappa(d(p, r))S_\kappa(d(q, r))\cos(\alpha)$$

We deduce the following result.

Lemma 3. *For for any triangle $\Delta(pqr)$ in $\mathcal{M}$ where m is the midpoint of $[p,q]$ we have:*

$$C_\kappa(d(p,r)) + C_\kappa(d(q,r)) \leq 2C_\kappa(d(m,r))C_\kappa\left(\frac{d(p,q)}{2}\right)$$

Proof. Let us consider the triangle $\Delta(pqr)$ in $\mathcal{M}$. We denote the geodesic midpoint of p and q by $m = \langle\frac{p+q}{2}\rangle$. Let $\Delta(p'q'r')$ be a comparison triangle to $\Delta(pqr)$ in $\mathcal{M}^n_\kappa$ and m' a comparison point to m. A fundamental characterization of $CAT(\kappa)$ spaces is that $d(r,m) < d(r',m')$. We apply lemma 3 to comparison triangles $\Delta(p'm'r')$ and $\Delta(r'm'q')$.



$$C_\kappa(d'(p',r')) = C_\kappa(d'(m',r'))C_\kappa\left(\frac{d'(p',q')}{2}\right) + S_\kappa(d'(m',r'))S_\kappa\left(\frac{d'(p',q')}{2}\right)\cos(\gamma')$$

and

$$C_\kappa(d'(q',r')) = C_\kappa(d'(m',r;))C_\kappa\left(\frac{d'(p',q')}{2}\right) + S_\kappa(d'(m',r'))S_\kappa\left(\frac{d'(p',q')}{2}\right)\cos(\pi - \gamma')$$

summing the two equations we get:

$$C_\kappa(d'(p',r')) + C_\kappa(d'(q',r')) = 2C_\kappa(d'(m',r'))C_\kappa\left(\frac{d'(p',q')}{2}\right)$$

Since $\Delta(pqr)$ and $\Delta(p'q'r')$ are comparison triangles: $C_\kappa(d(p,r))+C_\kappa(d(q,r)) = 2C_\kappa(d'(m',r'))C_\kappa\left(\frac{d(p,q)}{2}\right)$. Since C_κ is decreasing in $[0, \frac{\pi}{\sqrt{K}}]$ and $d(r,m) < d(r',m')$, we get:

$$C_\kappa(d(p,r)) + C_\kappa(d(q,r)) \leq 2C_\kappa(d(m,r))C_\kappa\left(\frac{d(p,q)}{2}\right) ;$$

which is the desired result. □

Proposition 10 ([prop. II.1.4)**.** *] [BH99] Let $\mathcal{M}$ denote a $CAT(\kappa)$ metric space.*

1. *If x and y in $\mathcal{M}$ are such that $d(x,y) < D_\kappa$, there exists a unique geodesic $[x,y]$ joining them.*

2. *For any $x \in \mathcal{M}$, the ball $B_{x,r}$ with $r < r_\kappa$ is convex.*

3.3 Distributed consensus protocol proposal

3.3.1 The Random Pairwise Midpoint (RPM) algorithm

At each count of the virtual global clock one node v is selected uniformly randomly from the set of agents V; the node v then randomly selects a node w from $\mathcal{N}(v)$. Both nodes v and w then compute and update their value to $\langle\frac{x_v+x_w}{2}\rangle$.

Algorithm RPM

Input: a graph $G = (V,E)$ and the initial nodes configuration $X_v(0), v \in V$
for all $k > 0$ **do**
At instant k, uniformly randomly choose a node V_k from V and a node W_k uniformly randomly from $\mathcal{N}(V_k)$.
Update:
$x_{V_k}(k) = \frac{x_{V_k}(k-1)+x_{W_k}(k-1)}{2}$
$x_{W_k}(k) = \frac{x_{V_k}(k-1)+x_{W_k}(k-1)}{2}$
$x_v(k) = x_v(k-1)$ for $v \notin \{V_k, W_k\}$
end for

Remark 7. Please note that the previous algorithm is well defined in the case where data belongs to some $CAT(0)$ space thanks to proposition 6. Otherwise, midpoints are not necessarily well-defined, and the algorithm should read: compute any midpoint between $x_{V_k}(k-1)$ and $x_{W_k}(k-1)$, if there exists any. However, it will be seen in the next sections that, in this case, the algorithm might fail to converge to a consensus.

Remark 8. The complexity to compute midpoints depends a lot on the considered space. For some data spaces ($SO(3)$, the sphere, etc), the formula for the midpoint is available in closed form. In other instances (cubical complexes, pylogenetic spaces, etc ...) a closed form cannot be found but there exists an algorithm for computing the midpoints [Bel15].

3.3.2 Convergence in the case of $CAT(0)$ spaces

In order to study convergence, we recall the following assumptions.

Assumption 6.

1. *$G = (V, E)$ is connected*
2. *$(V_k, W_k)_{k\geq 0}$ are i.i.d random variables, such that:*
   (a) *(V_k, W_k) is independent from $x_0, \dots, x_{k-1}, (V_0, W_0), \dots, (V_{k-1}, W_{k-1})$,*
   (b) $\mathbb{P}[\{V_0, W_0\} = \{v, w\}] = \frac{1}{N}(\deg^{-1}(v) + \deg^{-1}(w))\delta\{v \sim w\}$

Assumption 7. *$(\mathcal{M}, d)$ is a* complete $CAT(0)$ *metric space.*

We now define the *disagreement function*.

Definition 16. *Given a configuration $x = (x_1, \dots, x_N) \in \mathcal{M}^N$, the disagreement function $\Delta(x)$ is given by:*

$$\Delta(x) = \sum_{\substack{v\sim w \\ \{v,w\}\in\mathcal{P}_2(V)}} \frac{1}{N}(\deg(v)^{-1} + \deg(w)^{-1})d^2(x_v, x_w)$$

Function Δ measures how much disagreement is left in the network. Indeed, since the network is connected, Δ is 0 if and only if the network is at consensus. It would be a graph Laplacian in the Euclidean setting. The normalizing term involving degrees gives less weight to more connected vertices, since they are more likely to be solicited by neighbors, in order to give equal weight to each edge in the graph. This normalization will turn out to be convenient in the analysis. Notice also that $\Delta(x)$ at each link vw is proportional to the variance of distance between x_v and x_w.

Another important function is the *variance* function.

Definition 17. *Given a configuration $x = (x_1, \dots, x_N) \in \mathcal{M}^N$, the* variance *function is defined as:*

$$\sigma^2(x) = \frac{1}{N} \sum_{\{v,w\}\in\mathcal{P}_2(V)} d^2(x_v, x_w)$$

Remark 9. The $\frac{1}{N}$ normalizing constant accounts for the fact that when d is the Euclidean distance then $\sigma^2(x)$ equals $\sum_{v\in V} \|x_v - \bar{x}\|^2$, with $\bar{x} = \frac{1}{N}\sum_{v\in V} x_v$.

The next proposition measures the average decrease of variance at each iteration.

Proposition 11. *[BJ16] Under Assumptions 6 and 7, for X_k given by (RPM) algorithm, the following inequality holds, for every $k \geq 1$.*

$$\mathbb{E}[\sigma^2(x(k)) - \sigma^2(x(k-1))] \leq -\frac{1}{2}\mathbb{E}[\Delta(x(k-1))]$$

Proof. Taking into account that at round k, two nodes with indices V_k and W_k woke up, it follows that:

$$N(\sigma^2(x(k)) - \sigma^2(x(k-1))) = -d^2(x_{V_k}(k-1), x_{W_k}(k-1))$$

$$+ \sum_{\substack{u\in V \\ u\neq V_k, u\neq W_k}} T(V_k, W_k, u)$$

where $T(V_k, W_k, u) = 2d^2(x_u(k), M_k) - d^2(x_u(k), x_{V_k}(k-1)) - d^2(x_u(k), x_{W_k}(k-1))$ and M_k denotes the midpoint $\langle \frac{x_{V_k}(k-1)+x_{W_k}(k-1)}{2} \rangle$.

Notice that $x_{W_k}(k) = x_{V_k}(k) = M_k$. Now, using the CAT(0) inequality, one has:

$$N(\sigma^2(x(k)) - \sigma^2(x(k-1))) \leq -\frac{N}{2} d^2(x_{V_k}(k-1), x_{W_k}(k-1)) \ .$$

taking expectations on both sides and dividing by N gives:

$$\mathbb{E}[\sigma^2(x(k)) - \sigma^2(x(k-1))] \leq -\frac{1}{2}\mathbb{E}[d^2(x_{V_k}(k-1), x_{W_k}(k-1))]$$

recalling that $\mathbb{P}[\{V_k, W_k\} = \{u, v\}] = \frac{1}{N}(\frac{1}{\deg u} + \frac{1}{\deg v})$ when $u \sim v$ and 0 otherwise, and that (V_k, W_k) are independent from X_{k-1}, one can deduce:

$$\mathbb{E}[d^2(x_{V_k}(k-1), x_{W_k}(k-1))] = \mathbb{E}[\Delta(x(k-1))]$$

□

Proposition 12. *[BJ16] Assume $G = (V, E)$ is an undirected connected graph, there exists a constant $C_G \geq 1$ depending on the graph only such that:*

$$\forall x \in \mathcal{M}^N, \quad \frac{1}{2}\Delta(x) \leq \sigma^2(x) \leq C_G \Delta(x)$$

Proof. First:

$$\begin{aligned}\Delta(x) &= \sum_{v\sim w}\frac{1}{N}(\deg(v)^{-1}+\deg(w)^{-1})d^2(x_v,x_w)\\ &\leq \frac{2}{N}\sum_{v\sim w}d^2(x_v,x_w)\\ &\leq \frac{2}{N}\sum_{\{v,w\}\in\mathcal{P}_2(V)}d^2(x_v,x_w)=2\sigma^2(x)\end{aligned}$$

For the second inequality, consider $v\neq w$ two vertices in V, not necessarily adjacent. Since G is connected, there exists a path $u_0=v,\ldots,u_l=w$ such that $u_i\sim u_{i+1}$. Then, using Cauchy-Schwartz inequality:

$$d(x_v,x_w)^2\leq l\sum_{i=0}^{l-1}d^2(x_{u_i},x_{u_{i+1}})$$

$$\leq C\sum_{i=0}^{l-1}(\deg(u_i)^{-1}+\deg(u_{i+1})^{-1})d^2(x_{u_i},x_{u_{i+1}})$$

with: $C=\frac{\deg(G)\,\mathrm{diam}(G)}{2}$, where $\deg(G)$ denotes the *maximum degree* $\max\{\deg(v):v\in V\}$ and $\mathrm{diam}(G)$ the *diameter* of G. Hence taking $C_G=(N-1)\frac{\deg(G)}{2}\,\mathrm{diam}(G)\geq 1$, one recovers the desired inequality. □

Remark 10. Both functions Δ and σ^2 measure disagreement in the network, Δ takes into account the graph connectivity while σ^2 does not. The previous result shows that Δ and σ^2 are nonetheless equivalent up to multiplicative constants.

We now state a first convergence result.

Theorem 11 (Almost-sure convergence to consensus)**.** *[BJ16] Under Assumptions 6 and 7, there exists a random variable $x_\infty=(x_{\infty,v})_{v\in V}$, such that: (i) x_k converges almost surely to x_∞ and, (ii) almost surely, $\forall(v,w)\in V^2$, $x_{\infty,v}=x_{\infty,w}$,* i.e. *$x_\infty$ takes consensus values.*

Proof. Let us first show that $\Delta(x_k)$ converges almost surely to 0. From proposition 11, $\mathbb{E}[\sigma^2(x_k)]$ is non-increasing; which implies again from proposition 11:

$$\sum_k\mathbb{E}[\Delta(x_k)]<2\sigma^2(x_0)<\infty$$

hence, $\sum_k\Delta(x_k)$ has a finite expectation and $\Delta(x_k)$ converges almost surely to 0. Therefore, using the first inequality in proposition 12, $\sigma^2(x_k)$ converges to 0. As

a direct consequence, the diameter $\max\{d(x_v(k), x_w(k)) : (v, w) \in V^2\}$ also tends to 0 when k goes to ∞.

Now denote by S_k the set $\{x_v(k) : v \in V\}$ and by $\mathrm{conv}(S_k)$ its *convex hull*. One has $\mathrm{diam}(\mathrm{conv}(S_k)) \leq 2\,\mathrm{diam}(S_k)$: every ball centered at $x_v(k)$ with radius $\mathrm{diam}(S_k)$ is a convex set containing S_k and hence $\mathrm{conv}(S_k)$. Moreover, using the definition of convexity, one has $S_{k+1} \subset \mathrm{conv}(S_k)$. Therefore $\mathrm{conv}(S_k)$ forms a family of nested closed sets with diameter converging to 0. It is an easy result that in a complete metric space, the intersection of a family of nested closed subsets with diameter converging to 0 reduces to a singleton. □

Remark 11. A precise relationship between the distribution of x_∞ and x_0 seems out of the scope of this work. However, notice that the previous proof straightforwardly implies that x_∞ lies in the convex hull of the initial points $\{x_0(v); v \in V\}$.

Actually the previous proof can be adapted to give information on the convergence speed of the algorithm. Let us first prove an elementary lemma.

Lemma 4. *Assume a_n is a sequence of nonnegative numbers such that $a_{n+1} - a_n \leq -\beta a_n$ with $\beta \in (0, 1)$. Then,*

$$\forall n \geq 0, \quad a_n \leq a_0 \exp(-\beta n)$$

Proof. Indeed if $l_n = \log a_n$, then $l_{n+1} - l_n \leq \log(1 - \beta) \leq -\beta$. Hence $l_n \leq l_0 - \beta n$. Taking exponential on both side gives the expected result. □

We are now in a position to prove the following result:

Theorem 12 (Convergence speed)**.** *[BJ16] Let $x_k = (x_1(k), ..., x_N(k))$ denote the sequence of random variables generated by (RPM) algorithm. Under Assumptions 6 and 7, there exists $L \in (-1, 0)$ such that $\forall\ k \in \mathbb{N}$:*

$$\mathbb{E}\sigma^2(x(k)) \leq \exp(Lk)$$

Proof. Denote by $a_n = \mathbb{E}\sigma^2(x(k))$. Using the same argument as in the proof of theorem 11 and proposition 12, we know that the constant $L = -\frac{1}{2C_G}$ verifies $a_{n+1} - a_n \leq La_n$ and $L \in (-1, 0)$ since $C_G \geq 1$. We conclude using lemma 4. □

Remark 12.

- Using Proposition 12, it is straightforward to see that an analogous inequality holds for $\mathbb{E}\Delta(x_k)$.
- The convergence rate places a linear bound of negative slope on $\log \mathbb{E}\sigma^2(x_k)$. This means that on average, the Δ function converges to 0 at a linear rate.

Remark 13. What we have shown so far is that for $CAT(0)$ spaces, both convergence and convergence speed are similar to the Euclidean case; yet the proof techniques only rely on metric comparisons, whereas spectral techniques are mainly used in the Euclidean case (*e.g.* [BGPS06]).

Remark 14. The convergence and speed results are still true in the case where instead of explicitly giving the probability of the gossiping on each link, we now only assume that it is strictly positive for neighboring nodes.

- This means that: $\mathbb{P}[\{V_0, W_0\} = \{v, w\}] = P_{v,w}$ with $P_{v,w} > 0$ if $v \sim w$ and $P_{v,w} = 0$ otherwise.
- Proposition 11 remains true by changing the expression of the disagreement function to:
$$\Delta(X) = \sum_{v,w \in \mathcal{P}_2(V)} P_{v,w} d^2(x_v, x_w)$$
Proposition 12 also remains correct by taking:
$$C_G = \frac{N(N-1)\operatorname{diam}(G)}{2\min_{v \sim w}(P_{v,w})}$$
.

Thus theorems 11 and 12 remain valid.

Remark 15. We can obtain the real time convergence speed. If K_t is the Poisson process if the global clock of parameter $N\lambda$ then we define the real time measurements vector as: $X(t) = (x_1(K_t), \ldots, x_N(K_t))$, the variance function becomes $\sigma(X(t)) = \sum_{u,v} d^2(x_u(K_t), x_v(K_t))$.

We know that for any positive integer k we have:

$$\mathbb{E}[\sigma(X(k))] \leq \exp(-Lk)$$

And that:

$$\mathbb{P}(K_t = k) = \frac{(N\lambda t)^k}{k!} \exp(-N\lambda t)$$

Thus:

$$\begin{aligned}
\mathbb{E}[\sigma^2(X(K_t))] &= \sum_{k=0}^{\infty} \mathbb{E}[\sigma^2(X(K_t))|K_t = k]\mathbb{P}(K_t = k) \\
&= \sum_{k=0}^{\infty} \mathbb{E}[\sigma^2(X(k))]\mathbb{P}(K_t = k) \\
&\leq \sum_{k=0}^{\infty} \exp(-Lk)\frac{(N\lambda t)^k}{k!} \exp(-N\lambda t) \\
&= \exp(-N\lambda t) \sum_{k=0}^{\infty} \frac{(\exp(-L)N\lambda t)^k}{k!}
\end{aligned}$$

$$= \exp(-(1-I)N\lambda t)$$

with $I = \exp(-L) < 1$.
And thus:

$$\mathbb{E}[\sigma^2(X(K_t))] \leq \exp(-(1-I)N\lambda t)$$

3.3.3 Convergence in the case of $CAT(\kappa)$ spaces with $\kappa > 0$

In this section, we replace Assumption 7 by the following:

Assumption 8.

1. $\kappa > 0$
2. $(\mathcal{M}, d)$ *is a* complete $CAT(\kappa)$ *metric space.*
3. $\text{diam}(\{x_v(0) : v \in V\}) < r_\kappa$

By proposition 10 we are ensured that Random Pairwise Midpoint algorithm is well-defined. Indeed, by convexity of balls with radius smaller than r_κ, points will remain within distance less than r_κ of each other. Moreover midpoints are well-defined and unique since $r_\kappa < D_\kappa$.

The trick used to study $CAT(\kappa)$ configurations is to replace distance $d(x, y)$ by: $\chi_\kappa(d(x, y))$ with $\chi_\kappa = 1 - C_\kappa$ being pointwise non negative. We adapt the definitions used in the $CAT(0)$ setting as follows:

Definition 18. *For $x \in \mathcal{M}^n$ define:*

$$\Delta_\kappa(x) = \frac{1}{2N} \sum_{\substack{v \sim w \\ \{v,w\} \in E}} \left(\frac{\chi_\kappa(d(x_v, x_w))}{\deg(v)} + \frac{\chi_\kappa(d(x_v, x_w))}{\deg(w)}\right)$$

$$\sigma^2_\kappa(x) = \frac{2}{N} \sum_{\{v,w\} \in \mathcal{P}_2(V)} \chi_\kappa(d(x_v, x_w))$$

One can observe that for all $k \geq 0$, $(v, w) \in V^2$: $\sigma^2_\kappa(X_k) \geq 0$ and $\Delta_\kappa(X_k) \geq 0$. Notice that $\sigma^2_\kappa(x) = 0$ implies that for all $\{v, w\} \in \mathcal{P}_2(V)$: $\chi_\kappa(d(v, w)) = 0$; and, since $0 \leq d(v, w) \leq \frac{\pi}{2\sqrt{\kappa}}$, it implies that $d(v, w) = 0$, hence the system is in a consensus state. Moreover, when $\kappa \to 0$, $\Delta_\kappa \to \Delta$ and $\sigma^2_\kappa \to \sigma^2$.

The following proposition is a direct consequence of lemma 3.

Proposition 13. *[BJ16] Under Assumption 8, for any triangle $\Delta(pqr)$ in $\mathcal{C}$ where m is the midpoint of $[p, q]$ we have:*

$$\chi_\kappa(d(m, r)) \leq \frac{\chi_\kappa(d(p, r)) + \chi_\kappa(d(q, r))}{2}$$

Proof. From lemma 3 we get:

$$C_\kappa(d(p,r)) + C_\kappa(d(q,r)) - 2C_\kappa(d(m,r))$$
$$\leq 2C_\kappa(d(m,r))C_\kappa(d(p,q)) - 2C_\kappa(d(m,r))$$

since $\max\{d(m,r), d(p,q)\} < \frac{\pi}{2\sqrt{\kappa}}$ we have:

$$0 \leq C_\kappa(d(p,q)) \leq 1 \qquad 0 \leq C_\kappa(d(m,r)) \leq 1$$

which means that:

$$2C_\kappa(d(m,r))C_\kappa(d(p,q)) - 2C_\kappa(d(m,r)) \leq 0$$

and thus:

$$2\chi_\kappa(d(m,r)) \leq \chi_\kappa(d(p,r)) + \chi_\kappa(d(q,r))$$

□

With this result, it is now possible to obtain an analogous result to that of proposition 11. The techniques are the same but the details differ slightly. For the sake of completeness, we give the details below.

Proposition 14. *[BJ16]*

$$\mathbb{E}[\sigma^2_\kappa(x_{k+1}) - \sigma^2_\kappa(x_k)] \leq \ -\frac{1}{N}\mathbb{E}\Delta_\kappa(x_k)$$

Proof. At round k, two nodes with indices V_k and W_k wake up, it follows that:

$$N(\sigma^2_\kappa(x_k) - \sigma^2_\kappa(x_{k-1})) = -\chi_\kappa(d(x_{V_k}(k-1), x_{W_k}(k-1)))$$

$$+ \sum_{\substack{u \in V \\ u \neq V_k, u \neq W_k}} T_\kappa(V_k, W_k, u)$$

where T_κ is defined as:

$$T_\kappa(V_k, W_k, u) = 2\chi_\kappa(d(x_u(k), M_k)) - \chi_\kappa(d(x_u(k), x_{V_k}(k-1)))$$
$$-\chi_\kappa(d(x_u(k), x_{W_k}(k-1))) \ .$$

Now, using the inequality of proposition 13, one gets that $T_\kappa(V_k, W_k, u) \leq 0$ and:

$$N(\sigma^2_\kappa(x_k) - \sigma^2_\kappa(x_{k-1})) \leq \chi_\kappa(d(x_{V_k}(k-1), x_{W_k}(k-1))) \ .$$

taking expectations on both sides and dividing by N gives:

$$N\mathbb{E}[\sigma^2_\kappa(x_k) - \sigma^2_\kappa(x_{k-1})] \leq -\mathbb{E}[\chi_\kappa(d(x_{V_k}(k-1), x_{W_k}(k-1)))]$$

using similar reasoning as in the proof of proposition 11 we have:

$$\mathbb{E}[\chi_\kappa(d(x_{V_k}(k-1), x_{W_k}(k-1)))] = \mathbb{E}[\Delta_\kappa(x_{k-1})]$$

which yields:

$$\mathbb{E}[\sigma_\kappa^2(x_{k+1}) - \sigma_\kappa^2(x_k)] \leq -\frac{1}{N}\mathbb{E}\Delta_\kappa(x_k)$$

□

Remark 16. Notice the constant $1/N$ is in the right hand side which differs from the case of non positive curvature (compare with Proposition 11).

In order to derive a convergence result, we need an analogous result to Proposition 12 for $CAT(\kappa)$ spaces.

Proposition 15. *[BJ16] Assume $G = (V, E)$ is an undirected connected graph, there exists a constant C_κ depending on the graph only such that:*

$$\forall x \in \mathcal{M}^N, \quad \frac{\kappa}{\pi^2}\Delta_\kappa(x) \leq \sigma_\kappa^2(x) \leq C_\kappa \Delta_\kappa(x)$$

Proof. One has: $\frac{2\kappa}{\pi^2}x^2 \leq \chi_\kappa(x) \leq \frac{\kappa}{2}x^2$ when $0 \leq x < \frac{\pi}{2\sqrt{\kappa}}$. Hence, under Assumption 8, χ_κ and d are equivalent. The result then follows from Proposition 15. □

All the tools needed to show almost-sure convergence and speed are in place. The proofs of the following two results are exactly the same than in the $CAT(0)$ case, with Δ and σ replaced by Δ_κ and σ_κ.

Theorem 13. *[BJ16] Let $x(k) = (x_1(k), ..., x_N(k))$ denote the sequence generated by RPM algorithm, then under Assumptions 6 and 8, there exists a random variable x_∞ taking values in the consensus subspace, such that x_k tends to x_∞ almost surely.*

Theorem 14. *Let $x(k) = (x_1(k), ..., x_N(k))$ denote the sequence of random variables generated by algorithm RPM; under Assumptions 6 and 8, there exists $L \in (-1, 0)$ such that,*

$$\mathbb{E}\Delta_\kappa(x_k) \leq \exp(Lk)$$

These results show that – *provided all the initial points are close enough from each other*, as detailed by Assumption 8.3 – the situation is the same as in non positive curvature, specifically, almost sure convergence taking place at least exponentially fast. Notice that, by contrast, there are no constraints on the initialization, for the result to hold true in $CAT(0)$. Notice also that the radius involved in Assumption 8.3 depends on the curvature upper bound κ and ensures convexity of corresponding balls. It gives a hint that convexity plays an important role in the behavior of the algorithm, which is not surprising, since the algorithm basically amounts to taking random midpoints.

3.4 Numerical applications

In this section we simulate the (RPM) algorithm for four examples. The first example is the space of covariance matrices; it is a Hadamard manifold (*i.e.*, a complete, simply connected manifold with nonpositive sectional curvature, see, *e.g.* [Lan99, Chap XI.3]). The second is the metric graph, (a complex of $(0, 1]$ segments), which is a $CAT(0)$ metric space with no differential structure. The other two examples are of $CAT(\kappa)$ spaces with $\kappa > 0$. They are the three dimensional unit sphere S^2 and three dimensional rotation matrices $SO(3)$.

When one of the above mentioned spaces happens to be stable by addition and multiplication by a scalar (this is the case for positive definite matrices), we compare the performance of the RPM algorithm with that of the Arithmetic Gossip. In order to distinguish between the two algorithms, we use the term RPM for the Random Pairwie Midpoint algorithm and the term Arithmetic Gossip for the classical random pairwise algorithm $x_{n+1,v} = x_{n+1,w} = \frac{1}{2}(x_{n,v} + x_{n,w})$ [BGPS06] which is equivalent to Midpoint Gossip when the distance is the Euclidean one.

The results of these comparisons, as we shall see, might depend on the distance function used to define the disagreement function, or equivalently, the variance function.

3.4.1 Positive definite matrices

The scenario in this experiment is as follows. Each sensor in the network estimates a covariance matrix for some observed multivariate process. Then the network seeks a consensus on these covariance matrices. We implemented the proposed algorithm on the of positive definite matrices $\mathrm{Pos}(n)$, [Lan99, chap. 12]. $\mathrm{Pos}(n)$ is equipped with distance

$$d(M, N)^2 = \mathrm{tr}\{\log(N^{-1/2}MN^{-1/2}) \log(N^{-1/2}MN^{-1/2})^T\} = \| \log(MN^{-1})\|^2$$

and:

$$\frac{M+N}{2} = M^{1/2}(M^{-1/2}NM^{-1/2})^{1/2}M^{1/2} \, .$$

using this distance, which comes from a Riemannian metric, $\mathrm{Pos}(n)$ is a Hadamard manifold [Lan99, p.326], see also [Bar13] for an in-depth presentation, and as such, it is a $CAT(0)$ space [Lan99, prop 3.4, p.311]. Using the previous relationships, it is straightforward to implement the RPM algorithm and compute $\log \sigma^2\big(M(n)\big)$ at each iteration n; where $x(n) = (x_1(n), \ldots, x_N(n))$ denotes the tuple of positive definite matrix held by the agents $1 \leq v \leq N$ at time n.

Regarding the initialization step, we generate N iid matrices $x_v(0)$, following a Wishart distribution on $q \times q$ positive definite matrices with parameters $(q, 1)$, *i.e.*,

as $\sum_{k=1}^{q} M_{k,v} M_{k,v}^T$ where $M_{k,v} \sim \mathcal{N}(0, I_q)$ are independent standard multivariate Gaussian vectors of dimension q (in this numerical experiment $q = 3$ and $N = 30$).

Regarding the network, the experiments are carried out using randomly generated graphs. Such graphs have N nodes and a probability $p \in (0, 1)$ that each two nodes are connected. For a given p, we generate a graph and check if it is connected; if it is, we run the RPM algorithm.

We plot in the same figure the curve $\sigma_n^2 : n \to \log \mathbb{E}\sigma^2(x(n))$ for graphs of $N = 30$ nodes with parameters $p = 0.5, 0.7$, and 0.9, and see how the speed of convergence is affected by the connectivity parameter p.

It is also interesting in this case to make a comparison for positive definite matrices between the RPM algorithm and the Euclidean arithmetic gossip. In figure 3.2, we plot $n \mapsto \sigma_n^2$ where n is the number of iterations and σ^2 the sum of "non Euclidean" distances squared. The result suggests that the Euclidean gossip algorithm has a slight advantage over RPM gossip in terms of convergence speed. However, if we plot $n \mapsto \sigma_{n,E}^2$ where $\sigma_{n,E}^2 = \frac{1}{N} \sum_{\{i,j\} \in \mathcal{P}_2(V)} ||x_i(n) - x_j(n)||_F^2$ and $||.||_F$ is the Frobenius Euclidean norm, the opposite seems to be true. As shown in figure 3.3 RPM algorithm performs slightly better. The midpoint gossip algorithm converges faster than Euclidean gossip when the variance is expressed in Euclidean distances.

3.4.2 Distributed classification

The situation is the following: Each node of the network must make a decision on a finite number $p < \infty$ of hypothesis shared between the elements of the network, which then must reach a consensus. Typical examples include intrusion detection, classification of planes by thermal sensor, etc.

Each node $v \in V$ has a discrete probability distribution $x_v = (x_{1,v}, \ldots, x_{p,v})$, with $x_{i,v}$ being the probability assigned to the i-th hypothesis. Naturally we have the relation: $\sum_{i=1}^{p} x_{v,i} = 1$. The data space considered here is the $p-1$ dimensional simplex $\mathcal{S}_p$ with the fisher information metric [AN00]. Using the map $T : (x_1, \ldots, x_p) \to (2\sqrt{x_1}, \ldots, 2\sqrt{x_p})$, we can show that the simplex can be isometrically identified to the $p-1$ dimensional sphere with radius 2, which is a metric space of positive curvature $\kappa = \frac{1}{2}$.

In this example, we shall consider the 3-dimensional simplex, which can be mapped to the quarter of the 3 dimensional sphere $S^2 = \{(x, y, z) \in \mathbb{R}^3 | x^2 + y^2 + z^2 = 4 | x \geq 0, y \geq 0, z \geq 0\}$ equipped with the distance $d(a, b) = \cos^{-1}(\langle a, b \rangle)$ such that $0 \leq d(a, b) \leq \pi$ for all $a, b \in S^2$, which is of diameter $r_1 = \frac{\pi}{2}$ thus convex and thus $CAT(\frac{1}{2})$ (as a convex subset of the model space $\mathcal{M}_2^3$ – with an abuse of language since $\mathcal{M}_\kappa^n$ is only defined up to an isometry).

Note that the sphere does not possess a vector space structure and thus one cannot use Euclidean gossip without a re-projection step.

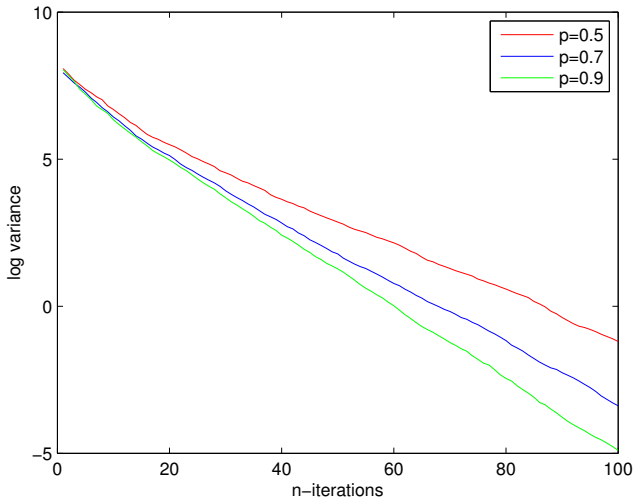


Figure 3.1 – Plot of $n \mapsto \log \sigma_n^2$ for the positive definite matrices space; the underlying network is the randomly generated graph for parameter $p = 0.5, 0.7, 0.9$. Because of the stochastic nature of the RPM algorithm, 50 simulations are done and we plot $\log \sigma_n^2$. The variance function behaves like an exponential, in accordance with the prediction of theorem 12. We can also observe that the slope of $\log \sigma_n^2$ decreases with p which highlights the importance of graph connectivity in information mixing.

We sample a set of $N = 30$ points uniformly from $\mathcal{Q}$ as initial step. The expression of a geodesic $\gamma(t)$ on $\mathcal{Q}$ such that $\gamma(0) = p$ and $\gamma(1) = q$ and $p \neq q$ is given by:

$$\gamma(t) = \sin\left(\cos^{-1}(\langle p, q\rangle)t\right)\frac{q - \langle p, q\rangle p}{\sqrt{1 - \langle p, q\rangle^2}}$$

$$+ \cos\left(\cos^{-1}(\langle p, q\rangle)t\right)p\ .$$

The total number of iterations is 500. We use a randomly generated graph.

By plotting the variance function $\sigma_1^2(X_n) = \frac{1}{N}\sum_{\{i,j\}\in\mathcal{P}_2(V)} \chi_1(d(x_i(n), x_j(n))$ with respect to the number of iterations we get from figure 3.5 that $n \mapsto \log \sigma_1^2(X_n)$ (or equivalently $n \mapsto \log \Delta_1$) is a linear function with negative slope which is in accordance with theorem 14. Convergence slows as p decreases, this again highlights the influence of graph connectivity on the speed of convergence.

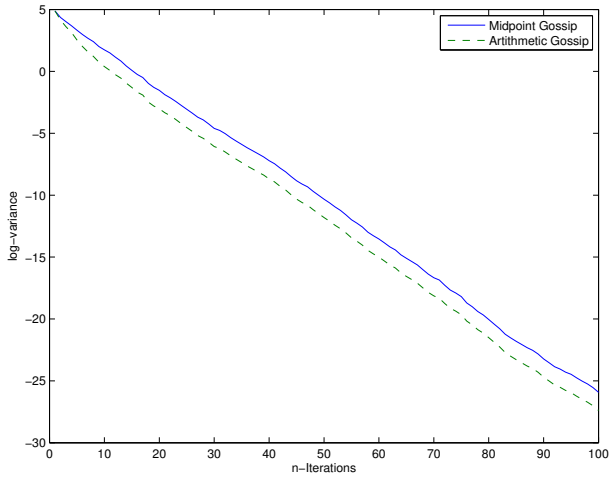


Figure 3.2 – Plot of $n \mapsto \log \sigma_n^2$ (non euclidean distances) for the positive definite matrices manifold, the full curve represents the RPM algorithm, while the dashed curve represents the Euclidean gossip based on arithmetic averaging. Euclidean gossip seems to converge faster.

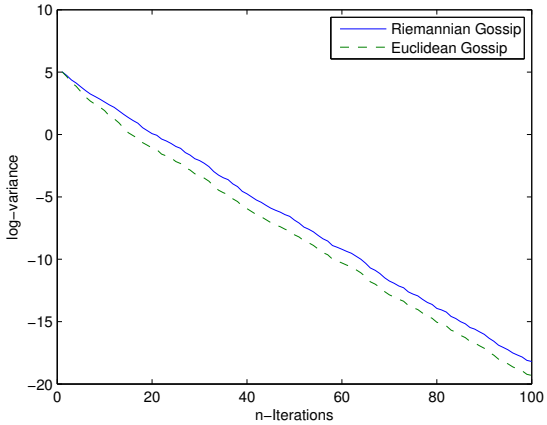


Figure 3.3 – Plot of $n \mapsto \log \sigma_{n,E}^2$ (using euclidean norm) for the positive definite matrices manifold, the full curve represents the RPM Algorithm, while the dashed curve represents Euclidean gossip. The midpoint gossip seems to have faster convergence.

3.4.3 Group of rotations

We shall be interested in what follows in the *rotations group* SO_3 of the Euclidean space $\mathbb{R}^3$. A rotation $R_{a,\theta}$ acting on $\mathbb{R}^3$ is characterized by its axis of rotation $a \in \mathbb{R}^3$ and its rotation angle $\theta \in [-\pi, \pi)$. One of the possible applications of

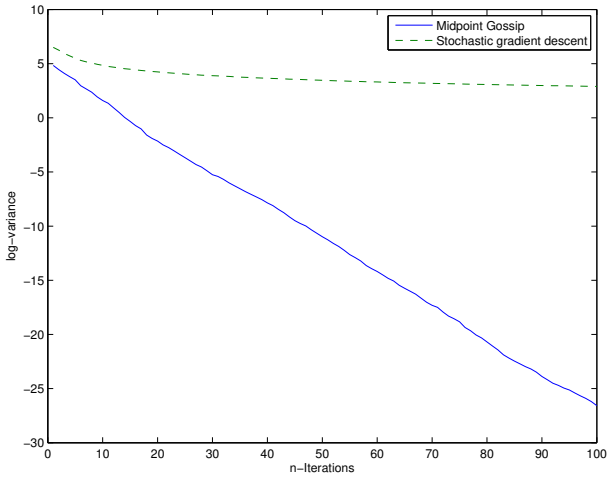


Figure 3.4 – Plot of $n \mapsto \log \sigma_n^2$ for the positive definite matrices manifold. The full curve represents RPM gossip while the dashed curve represents the stochastic gradient descent method applied to the function σ^2. Convergence is exponential in the first case while it is not for the second.

RPM algorithm with data in SO_3 is a network of $3D$ cameras [TVT08] that seeks to achieve a consensus in order to estimate the pose of an object.

The convexity radius for the 3-D rotations group is: $r_\kappa = \frac{\pi}{2}$ [TAV11]. Toponogov comparison theorem [Cha06, p.400] shows that the geodesic ball $\mathcal{B}$ with center I_3 and diameter r_κ is a $CAT(\frac{1}{4})$ space.

Let, $(R_{a_1,\theta_1}, R_{a_2,\theta_2}) \in \mathcal{B}^2$. If -1 is not an eigenvalue of $R_{a_1,\theta_1}^T R_{a_2,\theta_2}$, then the distance between the two elements is:

$$d(R_{a_1,\theta_1}, R_{a_2,\theta_2})^2 = \frac{1}{2}|| \log(R_{a_1,\theta_1}^T R_{a_2,\theta_2})||^2 = [\alpha]^2$$

where $\{e^{i[\alpha]}, e^{-i[\alpha]}, 1\}$ are the eigenvalues of $(R_{a_1,\theta_1}^T R_{a_2,\theta_2})$, such that $[\alpha] \in (-\pi, \pi)$. If -1 is an eigenvalue of $R_{a_1,\theta_1}^T R_{a_2,\theta_2}$ then $d(R_{a_1,\theta_1}, R_{a_2,\theta_2}) = \pi$, and $(R_{a_1,\theta_1}, R_{a_2,\theta_2})$ are said to be *antipodal points*.

For $R_{a,\theta} \in \mathcal{B}$ we have $d(I_3, R_{a,\theta}) = |\theta| < \frac{\pi}{4}$, and for $(R_{a_1,\theta_1}, R_{a_2,\theta_2}) \in \mathcal{B}^2$ we have $|[\alpha]| = d(R_{a_1,\theta_1}, R_{a_2,\theta_2}) \leq d(I_3, R_{a_1,\theta_1}) + d(I_3, R_{a_2,\theta_2}) < \frac{\pi}{2}$ which implies that -1 cannot be an eigenvalue of $R_{a_1,\theta_1}^T R_{a_2,\theta_2}$. Thus $\mathcal{B}$ does not contain antipodal points.

For all $(R_{a_1,\theta_1}, R_{a_2,\theta_2}) \in \mathcal{B}^2$ there exists a unique minimizing geodesic $\gamma(t)$ such that $\gamma(0) = R_{a_1,\theta_1}$ and $\gamma(1) = R_{a_2,\theta_2}$, and it has the following expression:

$$\gamma(t) = R_{a_1,\theta_1} \exp\left(t \log(R_{a_1,\theta_1}^T R_{a_2,\theta_2})\right)$$

since $\mathcal{B}$ is strongly convex [Cha06, p.404], $\gamma(t) \in \mathcal{B}$ for all $t \in [0, 1]$. The expression of the midpoint is thus: $\langle \frac{R_{a_1,\theta_1} + R_{a_2,\theta_2}}{2} \rangle = \sqrt{R_{a_1,\theta_1} R_{a_2,\theta_2}}$.

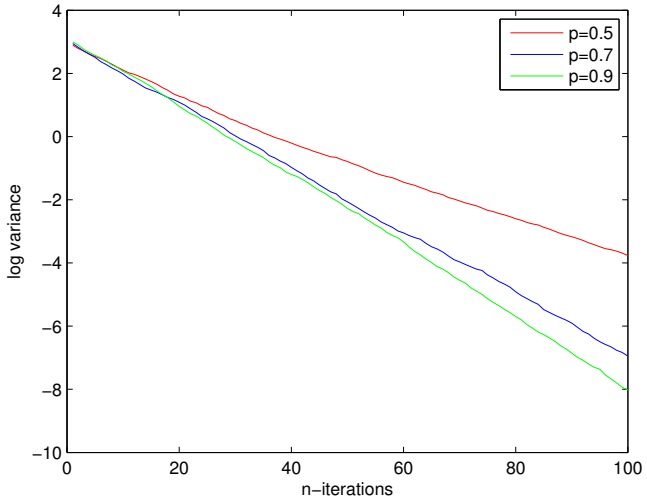


Figure 3.5 – Plot of $n \mapsto \log \sigma_1^2(X_n)$ where n is the iteration index and $\mathcal{M} = S^2$. The underlying network is the randomly generated graph for parameters $p = 0.5, 0.7, 0.9$. Again, because of the stochastic nature of the RPM algorithm, 50 simulations are done and we plot $\log \sigma_n^2$. The variance function behaves like an exponential, in accordance with the prediction of theorem 12; we can also observe that the slope of $\log \sigma_n^2$ decreases with p which highlights the importance of graph connectivity in information mixing.

In the numerical simulation presented in this chapter, we sample $N = 30$ rotation matrices $(R_i)_{1 \leq i \leq N} \in \mathcal{B}$. The underlying graph is randomly generated. The results of the experiment are displayed in figure 3.6 where we plot the logarithm of: $\sigma_{\frac{1}{4}}^2(X_n)$ as a function of n. We observe that $n \mapsto \log \sigma_{\frac{1}{4}}^2(X_n)$ decreases linearly, which is in accordance with theorem 14.

3.4.4 Conclusion

We presented in this chapter a new proposal: RPM algorithm, which achieves consensus at a linear rate in the asynchronous case for both positive and negative sectional curvature bounds. Our numerical results for RPM algorithm match with the predictions of the convergence and speed theorems established in section 3.3.1. The above seen examples can also be treated efficiently using Riemannian framework. The added value of the RPM algorithm is the simplification of proofs and an explicit result on the convergence rate in the asynchronous case.

In the next chapter, we will investigate a purely metric application that cannot be treated within Euclidean or Riemannian frameworks: Metamorphic/Reconfigurable

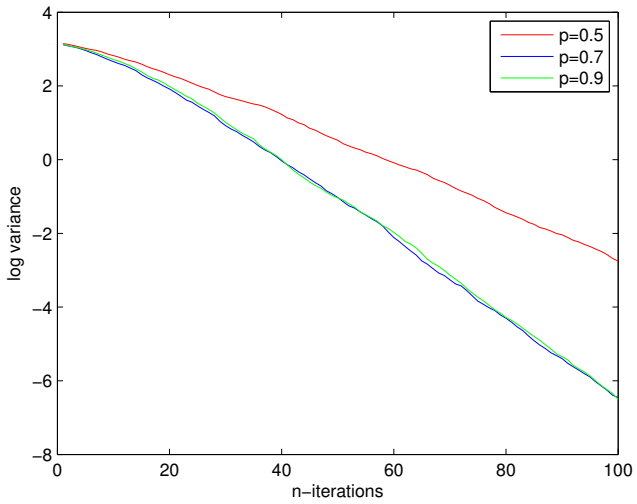


Figure 3.6 – Plot of $n \mapsto \log \sigma^2_{\frac{1}{4}}(n)$ where n is the iteration index, and $\mathcal{M} = SO_3$. The underlying network is the randomly generated graph for parameter $p = 0.5, 0.7, 0.9$. Because of the stochastic nature of the algorithm, 50 simulations are done and we plot $\log \sigma^2_n$. The variance function behaves like an exponential, in accordance with the prediction of theorem 12; we can also observe that the slope of $\log \sigma^2_n$ decreases with p which highlights the importance of graph connectivity in information mixing.

systems. They constitute one example of a novel extension of gossip to cases where only a metric setting can be applied.

Chapter 4

Application of RPM algorithm to metamorphic systems

4.1 Introduction

So far, the examples of $CAT(\kappa)$ data spaces where our RPM algorithm was applied could also be treated within the Riemannian framework. In this chapter, we apply the RPM algorithm to a purely metric data space where no differentiable structure can be applied.

4.2 Definitions and examples

There are many examples of metamorphic/reconfigurable systems, such as reconfigurable manufacturing systems [KHJ+03], and phylogenetic trees [BHV01]. Another example, one in which we will be more interested in this chapter, is that of metamorphic robots. They were described by E.H. Ostergaard in [ØKBL06] as robotic systems that:

1. Consist of several identical and physically independent unit modules,
2. Their modules can be connected to each other in many possible ways in order to form rigid structures,
3. Their modules can disconnect and reconnect while the system is active,
4. They can change the way their modules are connected, i.e. they are fully automatic.

Some of these robots are lattice-based, meaning that the robotic modules that occupy a discrete set of possible positions, this set of possible positions forms a

lattice. The nature of the lattice depends on the geometry of the modules, it can be hexagonal, squared, dodecahedral, etc. These metamorphic systems can be mathematically described as a collection of states on a graph (see section 4.3).

Representing the various states of a metamorphic system by their lattice configuration will prove to be insufficient for finding a simple consensus protocol. Indeed, the lattice representation does not provide an ordering of states. Following [AG04], we represent a system configuration as a point in a cubical complex $\mathcal{S}$ called the *state complex* (see also section 4.4 for a definition). The 0-dimensional skeleton of this complex is the set of states, and two vertices are linked by an edge if their corresponding states differ by a single action of a *generator* (see section 4.3). A k-cube of the complex represents k commutative movements – *i.e.* movements that are non-overlapping whatever their order.

An example of a metamorphic system is the robotic arm which consists of attached links, inside a grid with one of its extremities attached to the basepoint $(0, 0)$ of the grid (see figure 4.1).

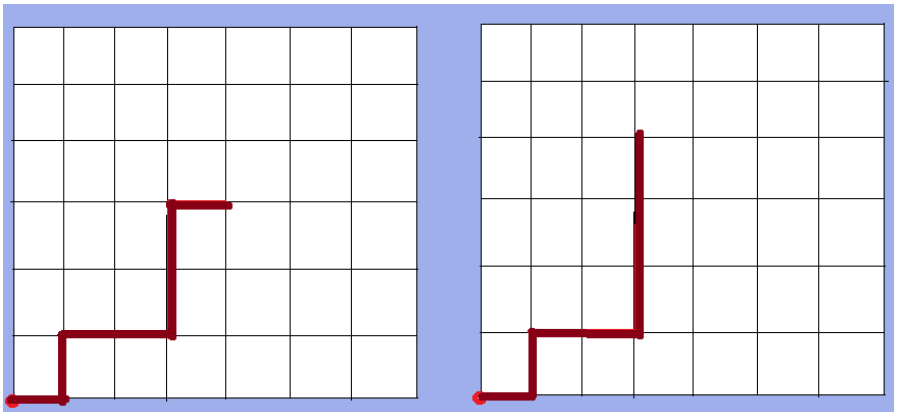


Figure 4.1 – Example of a lattice based reconfigurable system: the robotic arm. The edges in bold indicate the presence of a unit module, a black edge indicates its absence. The arm is attached at its basepoint $(0, 0)$. Here, an elementary movement has been performed by the module at the end of the arm, which changes the overall form of the system.

We assume here that metamorphic systems form a network; two systems are connected whenever they are able to communicate with each other. Our aim is to synchronize all the systems that compose the network, *i.e.* all the systems should have the same configuration, as shown in the example of figure 4.2.

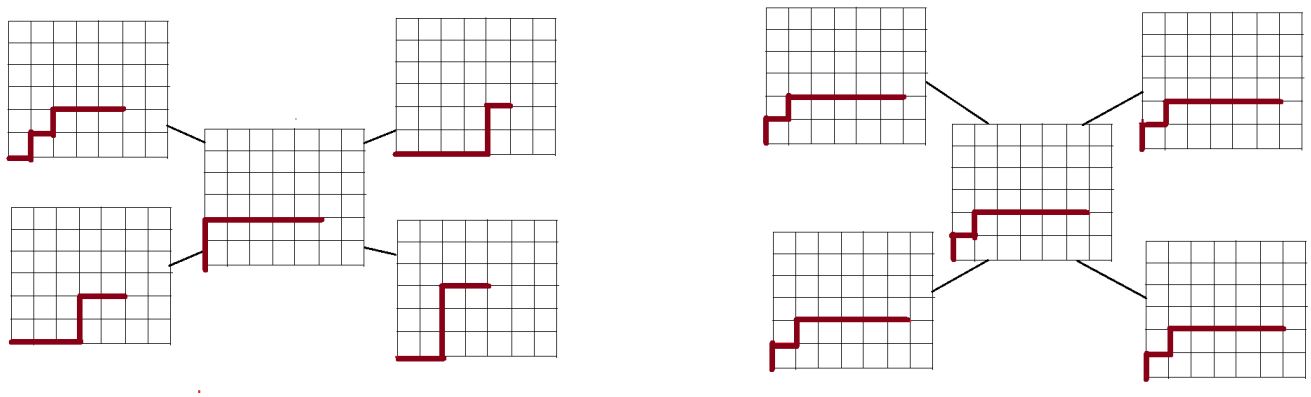


Figure 4.2 – In this example we are given five robotic arms. In the leftmost figure – describing the initial state – each arm has its own configuration. The rightmost figure represents a consensus state, in which all the arms share a common configuration.

4.3 Modeling metamorphic systems

For a rigorous definition of metamorphic systems, we follow the approach of [AG04]. We represent the lattice by its dual graph $\mathcal{G} = (\mathcal{V}(\mathcal{G}), \mathcal{E}(\mathcal{G}))$ whose vertices represent the individual cells of the lattice; two vertices are joined by an edge if and only if their corresponding cells are adjacent in the lattice. We associate to this graph a set of labels $\mathcal{A}$ on the set of vertices to indicate whether the corresponding cell is occupied or not and in the former case, whether it is occupied by an obstacle or by a module and of which type. A *state* of the system is any map $U : \mathcal{G} \to \mathcal{A}$. The metamorphic system dynamically changes its state through a set of elementary movements which satisfy the following rules: (i) units cannot overlap during reconfiguration and (ii) overall connectivity should always be maintained. This is done via *generators* which are defined below:

Definition 19. *[GP07] Let $\mathcal{G} = (\mathcal{V}(\mathcal{G}), \mathcal{E}(\mathcal{G}))$ be a graph and $\mathcal{A}$ a set of labels. A* generator *ϕ is a collection of three objects:*

- *A* support $\mathrm{SUP}(\phi) \subset \mathcal{G}$ *which is a subgraph of* $\mathcal{G}$
- *A* trace $\mathrm{TR}(\phi) \subset \mathrm{SUP}(\phi)$ *which is a subgraph of* $\mathrm{SUP}(\phi)$
- *An non-ordered pair of states* $u_0, u_1 : \mathcal{V}(\mathrm{SUP}(\phi)) \to \mathcal{A}$ *verifying :*

$$u_0|_{\mathcal{V}(\mathrm{SUP}(\phi))-\mathcal{V}(\mathrm{TR}(\phi))} = u_1|_{\mathcal{V}(\mathrm{SUP}(\phi))-\mathcal{V}(\mathrm{TR}(\phi))}$$

where $\mathcal{V}(\mathrm{SUP}(\phi))$ *are the vertices of* $\mathrm{SUP}(\phi)$

The support of the generator corresponds to the region of the graph that contains the necessary information to verify whether the movement is feasible (*i.e.* absence of obstacles, no modules overlapping, and connectivity of the system is maintained). The trace of the generator is the region of the graph where the movement actually takes place.

Definition 20. *[GP07] A generator is said to be* admissible *at a state U if: $U|_{\mathcal{V}(\mathrm{SUP}(\phi))} = \hat{u}_0$. The action of ϕ denoted by Φ maps a state $u : \mathcal{V}(\mathcal{G}) \to \mathcal{A}$ to a new one $\Phi[u] : \mathcal{V}(\mathcal{G}) \to \mathcal{A}$ given by:*

$$\Phi[U] := \begin{cases} U & : \text{on } \mathcal{V}(\mathcal{G}) - \mathcal{V}(\mathrm{TR}(\phi)) \\ \hat{u}_1 & : \text{on } \mathcal{V}(\mathrm{TR}(\phi)) \end{cases}$$

Thus, a generator ϕ acts on a state U by modifying its restriction on the vertices of $\mathrm{TR}(\phi)$ from u_0 to u_1. This corresponds to an elementary movement of unit modules. However a metamorphic system should be able to perform many such elementary movements simultaneously. In order for this to be feasible, the movements have to be compatible, *i.e.*, their simultaneous execution does not lead to module overlap or loss of system connectivity. For this, we introduce the following definition:

Definition 21. *[GP07] In a metamorphic system, a collection of generators $\{\phi_i\}$ is said to be* commutative *if:*

$$i \neq j \Rightarrow \mathrm{TR}(\phi_i) \cap \mathrm{SUP}(\phi_j) = \emptyset,$$

Next we shall see an important tool for visually representing the space of available states of a metamorphic system which is the state complex.

4.4 The state complex

4.4.1 Definition

A *cubical complex* can be seen as a collection of cubes glued together using isometries.

Definition 22. *[BH99, p.112] Let $\Gamma \subset \mathbb{N}$ and $(C_i)_{i\in\Gamma}$ be a collection of Euclidean unit cubes of various dimensions and $X = \coprod_{i\in\Gamma} C_i$ a disjoint union of these cubes. A cubical complex $\mathcal{S}$ is the quotient of X by an equivalence relation $\sim$ such that if $p : X \to \mathcal{S}$ is the natural projection then:*

- *For every $i \in \Gamma$ the restriction p_i of p to the cube C_i is injective.*

- *If $p_i(C_i) \cap p_j(C_j) \neq \emptyset$ then there is an isometry $h_{i,j}$ from a face $T_i \subset C_i$ onto a face $T_j \subset C_j$ such that $p_i(x) = p_j(x')$ if and only if $x' = h_{i,j}(x)$.*

Definition 23. *[GP07] The* state complex $\mathcal{S}$ *of a metamorphic system is a cubical complex. Each k-cube $e^{(k)}$ of $\mathcal{S}$ is an equivalence class $[u, (\phi_i)_{i=1}^k]$ where:*

- *$(\phi_i)_{i=1}^k$ is a k-tuple of commuting generators.*
- *u is a state for which the generators $(\phi_i)_{i=1}^k$ are admissible.*
- *$[u_0, (\phi_i)_{i=1}^k] = [u_1; (\phi'_i)_{i=1}^k]$ if and only if $\exists \sigma \in \mathfrak{t}_k$ such that: $\forall i \in \{1, \ldots, k\}$ we have: $\phi_i = \phi'_{\sigma(i)}$ and $u_0 = u_1$ on the subgraph: $\mathcal{G} - \bigcup \mathrm{TR}(\phi_{\alpha_i})$*

The boundary of each k-cube is a collection of $2k$ faces:

$$\partial[u; (\phi_{\alpha_i})_{i=1}^k] = \bigcup_{i=1}^{k} ([u; (\phi_{\alpha_j})_{j \neq i}] \cup [\phi_{\alpha_i}[u]; (\phi_{\alpha_j})_{j \neq i}])$$

One advantage of using the cubical complex representation over that of the transition graph is that it shows which elementary movements can be performed simultaneously, and thus contains more information than the transition graph. It is not generally computationally feasible to construct the cubical complex associated to a given reconfigurable system but there are some interesting cases where it is possible to do so like the case of the robotic arm [ABY12].

4.4.2 Encoding as a partially ordered set

Given a cubical complex $\mathcal{S}$ and a vertex v (called the *root* vertex) of $\mathcal{S}$, one can introduce a partial order relation in the set of vertices of $\mathcal{S}$ by stating that $u \prec w$ if and only if there is an edge path geodesic (i.e, a geodesic on the metric graph associated to $\mathcal{S}$) from the root v to w that passes through u. The complex $\mathcal{S}$ can thus be seen as a partially ordered set. In [AOS12] it is shown that if one can choose the special node v such that the set of vertices of $\mathcal{S}$ equipped with the partial order relation $\prec$ is a *poset with inconsistent pairs* then $\mathcal{S}$ is a globally $CAT(0)$ cubical complex. This last property guarantees the existence and uniqueness of the midpoint of any given two points, and allows us to apply the RPM algorithm on a network of agents whose data are encoded in a cubical complex. We assume in this chapter that all the studied complexes are $CAT(0)$.

Reciprocally, if given a poset with inconsistent pairs $(P, \prec)$ one can build a lattice whose summits are the order ideals of $(P, \prec)$ that do not contain any inconsistent pairs (such order ideals are said to be *consistent*). An edge is drawn between two summits if their corresponding ideals differ by exactly one element. The vertices are ordered according to their corresponding ideals, $u < w$ if and only

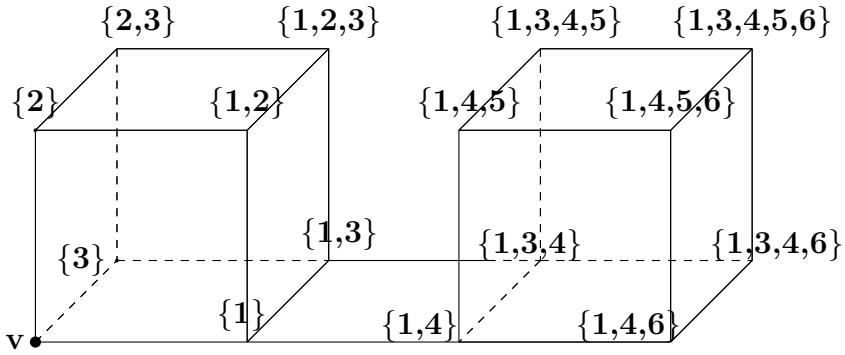


Figure 4.3 – The cubical complex associated with the poset ($P = \{1, 2, 3, 4, 5, 6\}, \prec$) such that: $1 \prec 4$, $1 \prec 5$ and $4 \prec 6$. Each vertex is labeled by a consistent order ideal of P.

if the order ideal I_u corresponding to the vertex u is a subset of I_w the order ideal associated to the vertex w ($u < w \Leftrightarrow I_u \subset I_w$) as shown in figure 4.3.

Individual cubes are represented by a pair (I, M) with I a consistent order ideal and M a subset of the maximal element of I. Such a cube is of dimension $|M|$ and its vertices are found by removing from I all the possible subsets of M.

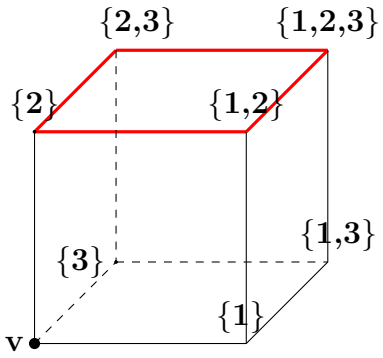


Figure 4.4 – In this example, the 2-dimensional cube highlighted in color is represented by the pair (I, M) with $I = \{1, 2, 3\}$ and $M = \{1, 3\}$

4.5 The standard embedding

One way to embed the cubical complex associated to a poset with inconsistent pairs of cardinal N is the so called *standard embedding* [AOS12]. An element $u \in \mathcal{S}$ is represented by a N-tuple $u = (u_1, \ldots, u_N)$ where $\forall i \; u_i \in [0, 1]$, and if for

some $(i,j) \in \{1,\dots,N\}^2$ we have: $u_j \neq 0$ and $u_j > u_i$, then we must have: $u_i = 1$. Also, if x_i and x_j are inconsistent then: $u_i u_j = 0$.

Figure 4.5 shows an example of a cubical complex where the standard embedding coordinates of the points x and y are: $x = (1; 0.25; 0.25; 0)$ and $y = (1; 0.75; 1; 0.75)$.

Computing the midpoint

In [Bac14], an algorithm is given for computing means and medians in general $CAT(0)$ metric spaces. We are here interested in finding the midpoint $\frac{x+y}{2}$ of any two given points x and y of a $CAT(0)$ cubical complex $\mathcal{S}$. We first determine the geodesic $[x,y]$ between x and y.

Let $x = (x_1 \dots x_N)$ and $y = (y_1 \dots y_N)$ be the coordinates of x and y in the current standard embedding of $\mathcal{S}$ in $\mathbb{Z}^N$.

Then, we define $v = (v_1,\dots,v_N)$ and $w = (w_1,\dots,w_N)$ such that for $i \in \{1,\dots,N\}$:

$$\begin{cases} v_i = 0 & \text{if } x_i \leq 0.5 \\ v_i = 1 & \text{if } x_i > 0.5 \end{cases}$$

and:

$$\begin{cases} w_i = 0 & \text{if } y_i \leq 0.5 \\ w_i = 1 & \text{if } y_i > 0.5 \end{cases}$$

v and w are respectively the closest vertices of $\mathcal{S}$ to x and y (see figure 4.5).

Next, we re-root the cubical complex at v and change the labeling of its vertices consequently as well as the coordinates of x and y . To obtain the new coordinates of any point $a = (a_1,\dots,a_N)$ in the new standard embedding, we update: $a_{i,new} = 1 - a_{i,old}$ if $v_i = 1$ and $a_{i,new} = a_{i,old}$ if $v_i = 0$ (see figures 4.5 and 4.6).

We denote by $I_w = \{i \in \{1,\dots N\} | w_i = 1\}$ the order ideal associated with the vertex w. Now that the complex is rerooted, we need to find the cubical sequence that contains the geodesic between x and y. According to [AOS12], a *valid sequence* $(C_k)_{1\leq k\leq n} = (I_k, M_k)_{1\leq k\leq n}$ of cubes containing the geodesic, consists of a sequence of order ideals $I_1 \subset I_2 \subset \cdots \subset I_n = I_w$ and maximal antichains $M_k \subset I_k$, $1 \leq k \leq n$ (see section 4.4 for the representation of individual cubes of a complex). To find this sequence, we take the subset of minimal elements of I_w and call it I_1, then set $M_1 = I_1$. Then, in order to form $C_k = (I_k, M_k)$ from $C_{k-1} = (I_{k-1}, M_{k-1})$, we take the subset of minimal elements m_k of $Q - I_{k-1}$ and put $I_k = I_{k-1} \cup m_k$ and let M_k be the maximal antichain of I_k. Using this procedure until $I_k = P$, we obtain a valid cube sequence which contains the geodesic. In the example of the figures 4.5 and 4.6, we have: $I_1 = \{2,3\}$, $M_1 = \{2,3\}$ and $I_2 = \{2,3,4\}$, $M_2 = \{2\}$.

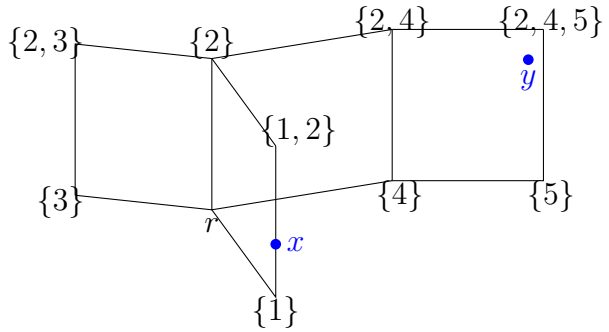


Figure 4.5 – Example of a cubical complex with an initial root vertex r. We want to compute the geodesic between the points $x = (1, 0.25, 0, 0, 0)$ and $y = (0, 0.75, 0, 1, 0.75)$

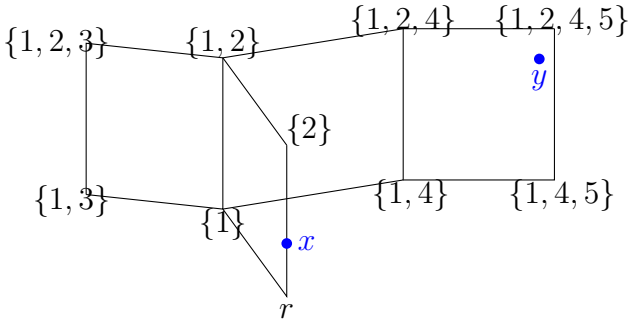


Figure 4.6 – We reroot cubical complex at v, and the new coordinates are: $x = (0, 0.25, 0, 0, 0)$ and $y = (1, 0.75, 0, 1, 0.75)$. The coordinates of w are $(0, 1, 0, 1, 1)$

After the cube sequence has been determined, we then have to find the 'break-points' from which the geodesic passes. The points $\{p_1, \ldots, p_{n-1}\}$ are such that: $\|x - p_1\| + \|p_1 - p_2\| + \cdots + \|p_{n-1} - p_n\| + \|p_n - y\|$ is minimal and that for each $k \in \{1, \ldots, n-1\}$ $p_k \in F_k$ with F_k being the common frontier of (I_k, M_k) and (I_{k+1}, M_{k+1}) which is the cubical cell $(I_k, M_k \cap M_{k+1})$. This problem can be cast as a touring problem with n polyhedral regions and $2n$ facets [AOS12].

$$\min t_0 + t_1 + \cdots + t_n$$

$$\forall k \in \{0 \ldots n\} : \; t_k \geq \|p_k - p_{k+1}\|$$

$$p_k \in F_k, \;\; k = 1 \ldots n$$

$$p_0 = x, p_{n+1} = y$$

the touring problem is a second order cone optimization problem, for which numerical solvers exist. We solve the touring problem and obtain a series of points $p_1 \ldots p_n$ and the distances $t_1 \ldots t_{n-1}$ between p_k and p_{k+1}.

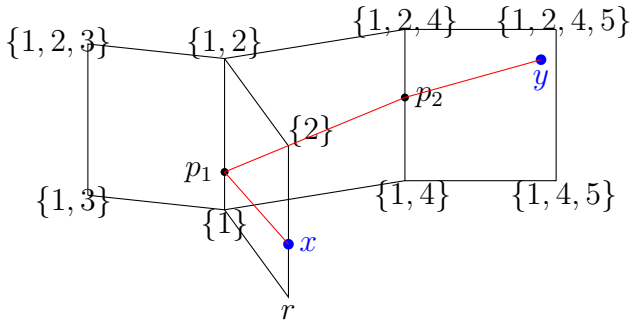


Figure 4.7 – In this example, the cube complex and the points x and y are such that we have non co-linear sequence of points $\{p_1, p_2\}$ that the geodesic between x and y passes through. The cube sequence $\{C_1, C_2, C_3\}$ contains this geodesic.

Knowing the sequence of points $(p_i)_{1\leq i\leq n}$ and distances $(t_i)_{0\leq i\leq n}$, we want to determine the midpoint of x and y. We have two cases:

Case 1: $n > 0$

First we determine the cubical cell of $(C_k)_{1\leq k\leq n+1}$ that contains the midpoint. Let $i_0 = \min\left\{1 \leq i \leq n | d(x, p_i) \geq \frac{d(x,y)}{2}\right\}$, then we have: $\frac{x+y}{2} \in C_{i_0}$ and $\frac{x+y}{2} \in [p_{i_0}, p_{i_0+1}]$. Using the convention $p_0 = x$ and $p_{n+1} = y$, the analytical expression of $\frac{x+y}{2}$ is:

$$\frac{x+y}{2} = p_{i_0} + \frac{D}{t_{i_0}}(p_{i_0+1} - p_{i_0})$$

with:

- $D = \frac{1}{2}\left|\sum_{k=0}^{i_0-1} t_k - \sum_{k=i_0+1}^{n+1} t_k - t_{i_0}\right|$: If $2 \leq i_0 \leq n-1$.
- $D = \frac{1}{2}t_{i_0}$: If $i_0 = 1$.
- $D = \frac{1}{2}\left|t_{i_0} - \sum_{k=0}^{i_0-1} t_k\right|$: If $i_0 = n$.

Case 2: $n = 0$.

In this case, x and y belong to the same cubical cell and we have: $\frac{x+y}{2} = \frac{x+y}{2}$.

In the rest of this section, we apply the RPM algorithm to two examples of lattice-based metamorphic systems: the robotic arm and the hexagonal lattice robot.

4.5.1 Application to the robotic arm

To understand why the robotic arm is $CAT(0)$, one must construct a poset with inconsistent pairs associated with the state complex of the arm (see section 4.4.2).

Definition 24. *Define for any robotic arm with n articulations R_n the set: $P_n = \{(x, y) | y \geq 0, y \leq x, x \leq n-1\}$ and define the partial order relation $\leq$ such that: $(x_1, y_1) \leq (x_2, y_2)$ if and only if: $x_1 \leq x_2$, and $y_1 \leq y_2$.*

Using the partially ordered set P_n, we can show that the state complex of R_n is $CAT(0)$ through the following proposition:

Proposition 16. *[ABY12] Let $\mathcal{S}_n$ be the cubical complex of R_n rooted at the state where the arm is completely horizontal. Then there is a bijection between the possible states of R_n and the order ideals of P_n.*

We plot the log-variance $\log \sigma^2$ as a function of the number of iterations k, for $N = 10$ robotic arms of $n = 7$ joints (see figure 4.8), first on a complete graph and then for a path graph. In both cases we observe a linear curve of negative slope in accordance with the results of [BJ14]. In the case of the path graph however the slope is smaller than for the complete graph.

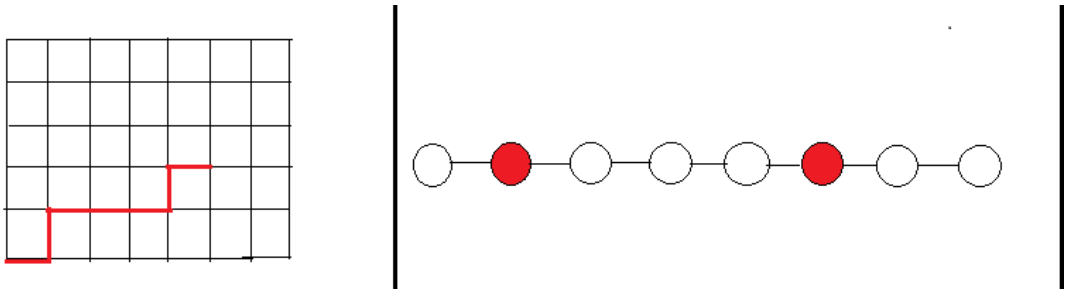


Figure 4.8 – In the leftmost figure, a robotic arm with $n = 7$ articulations; and in the rightmost figure, its associated state graph

4.5.2 Application to the planar hexagonal lattice

The same analysis is applied to the hexagonal lattice system [AG04] which is a connected aggregate of hexagonal shaped modules that occupy a planar lattice. Its graph representation and associated cubical complex can be seen in figure 4.9.

We observe similar patterns as for the robotic arms experiment; the function $\log \sigma^2$ decreases at a linear rate for both the complete graph and the path graph cases.

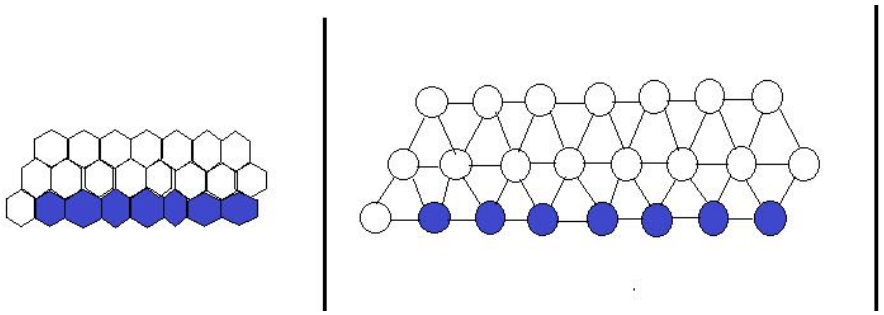


Figure 4.9 – In the left, a hexagonal lattice system, and in the right, its associated state graph

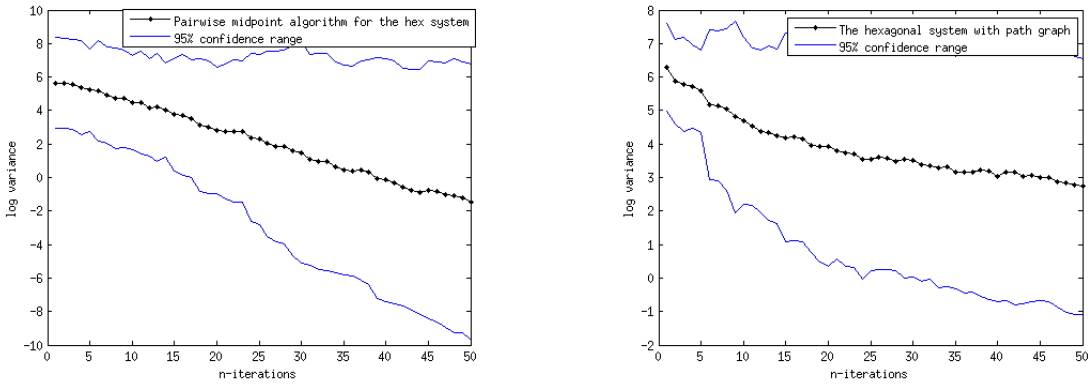


Figure 4.10 – Plot of $n \to \log \sigma^2$ for a network of $k = 10$ hexagonal systems. In the left figure, the underlying communications graph is the complete graph, while on the right it is the path graph. Because of the stochastic nature of the algorithm the procedure is averaged over 30 runs. The resulting curve is a line of negative slope of larger magnitude for the complete graph than for the path graph.

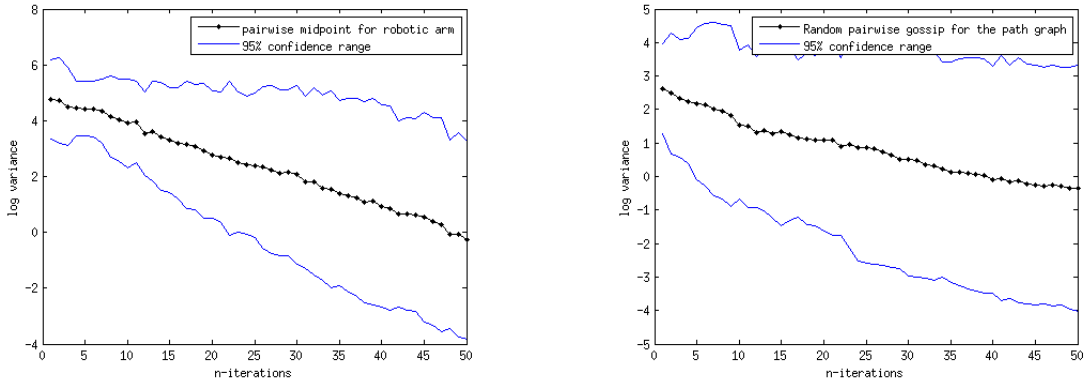


Figure 4.11 – Plot of $n \to \log \sigma^2$ for a network of $k = 10$ robotic arm with $n = 7$ articulations. In the left figure, the underlying communication graph is the complete graph, while on the right it is the path graph. Because of the stochastic nature of the algorithm, the procedure is averaged over 30 runs. The resulting curve is a line of negative slope of larger magnitude for the complete graph than for the path graph.

Chapter 5

RPM for time varying graphs and for directed graphs

5.1 Motivation for time varying graphs

While previous approaches in gossip generalization assumed a fixed network topology. We would like to see how changes in the underlying graph topology affects the convergence properties of the (RPM) algorithm,. Here, the graph is no longer assumed to be fixed with time; such restriction is important to study dynamics of mobile autonomous agents [OSM02, JL+03, BT+07], where new links can form and existing one disappear.

5.2 Framework

5.2.1 Network

Here, our network of N agents will be represented by a time varying graph $G(t) = (V, E(t))$, where $E(t)$ is a time-varying set reflecting the time-dependent nature of the communication links. When at a given instant t, a communication link $e = \{v, w\} \in E(t)$ exists between two agents, both agents are said to be neighbors. We denote by $\mathcal{N}(v, t)$ the set of all neighbors of the agent $v \in V$ at instant t.

Also, if t_1 and t_2 are two distinct times, we can define the union of the graphs $G(t_1) = (V, E(t_1))$ and $G(t_2) = (V, E(t_2))$ as: $G(t_1) \cup G(t_2) = (V, E(t_1) \cup E(t_2))$.

5.2.2 Synchronization

We assume the same Poisson clock model as in the asynchronous gossip case. We use the notation $G(k) = (V, E(k))$ to mean the set of communication links at

instant k, and $\mathcal{N}(v,k)$ to mean the set of available communication neighbors of v at instant k.

5.2.3 Communication

At each time k, there are only two communicating agents in the whole network: V_k and W_k. We assume that (V_k, W_k) are independent but not identically distributed anymore, the probability distribution of (V_k, W_k) is given by:

$$\mathbb{P}[V_k = v, W_k = w] = \begin{cases} P_{v,w}(k) = \frac{1}{\deg_v(k)} + \frac{1}{\deg_w(k)} \text{ if } \{v,w\} \in E(k) \\ 0 \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{otherwise} \end{cases}$$

We have $P_{u,v}(k) > \frac{1}{N}$, for any $k > 0$ and $\{u,v\} \in E(k)$. We define the constant $\alpha > 0$ as $\alpha = \frac{1}{N}$.

5.2.4 Data

Each node $v \in V$ stores data represented as an element x_v belonging to a $CAT(0)$ space $\mathcal{M}$. Initially each node v has a value $x_v(0)$ and $X(0) = (x_1(0), \ldots, x_N(0))$ is the tuple of initial values of the network.

The new scheme is similar to the one for static graphs: when the global clock of the system ticks for k-th time the measurements vector of the network is $X(k-1)$, and the communication links are described by $G(k)$ the underlying graph of the network at instant k. Agents V_k and W_k wake up, communicate and perform some computation, and then change the measurements vector to $X(k)$.

5.3 Convergence results

Assumption 9. *The following assumptions are adapted from [NO09]:*

1. $(\mathcal{M}, d)$ *is a* $CAT(0)$ *metric space*
2. $(V_k, W_k)_{k\geq 0}$ *are independent random variables, such that:*
   - (V_k, W_k) *is independent from* $X(0), \ldots, X(k-1), (V_0, W_0), \ldots, (V_{k-1}, W_{k-1})$,
   - $\mathbb{P}[\{V_k, W_k\} = \{v, w\}] = P_{v,w}(k)\delta_{\{\{v,w\}\in E(k)\}}$
3. $\exists\, T > 0$ *such that for any* $k > 0$ *we have:* $G(k) \cup G(k+1) \cup \cdots \cup G(k+T)$ *is a connected graph.*

We now define time variant versions of the *disagreement* and *variance* functions.

Definition 25. *Given a configuration $x = (x_1, \ldots, x_N) \in \mathcal{M}^N$, the disagreement function at instant k is:*

$$\Delta(x, k) = \sum_{\{v,w\} \in E(k)} P_{v,w}(k) d^2(x_v, x_w)$$

The variance *function is defined for any configuration x as:*

$$\sigma^2(x) = \frac{1}{N} \sum_{\{v,w\} \in \mathcal{P}_2(V)} d^2(x_v, x_w)$$

The next proposition measures the average decrease of variance at each iteration.

Proposition 17. *Under Assumptions 9, for X_k given by Algorithm RPM, the following inequality holds, for every $k \geq 1$.*

$$\mathbb{E}[\sigma^2(X(k)) - \sigma^2(X(k-1))] \leq -\frac{1}{2}\mathbb{E}[\Delta(X(k-1), k)]$$

Next we shall need the following definitions:

Definition 26. *For a given integer $k \geq 1$ we define the stochastic processes:*

$$\mathcal{L}(k) = \sum_{n=k}^{k+T} \sigma^2(X(n))$$

$$\mathcal{D}(k) = \sum_{n=k}^{k+T} \Delta(X(n), n+1)$$

We shall see later that $\mathcal{L}$ reaches 0 if and only if the system reaches a consensus state.

Proposition 18. *We have the following relationship between $\mathcal{L}$ and $\mathcal{D}$ for every integer k:*

$$\mathbb{E}[\mathcal{L}(k+1) - \mathcal{L}(k)] \leq -\frac{1}{2}\mathbb{E}[\mathcal{D}(k)]$$

Proof. We know that: $\mathcal{L}(k) = \sum_{n=k}^{k+T} \sigma^2(X(n))$.

$$\mathcal{L}(k+1) - \mathcal{L}(k) = \sum_{n=k+1}^{k+T+1} \sigma^2(X(n)) - \sum_{n=k}^{k+T} \sigma^2(X(n))$$

$$= \sum_{n=k}^{k+T} \left(\sigma^2(X(n+1)) - \sigma^2(X(n))\right)$$

Taking the expected value and using proposition 17 we have:

$$\mathbb{E}\left[\mathcal{L}(k+1) - \mathcal{L}(k)\right] = \sum_{n=k}^{k+T} \mathbb{E}\left[\sigma^2(X(n+1)) - \sigma^2(X(n))\right]$$

$$\leq -\frac{1}{2}\sum_{n=k}^{k+T} \mathbb{E}[\Delta(X(n), n+1)] = -\frac{1}{2}\mathbb{E}\left[\mathcal{D}(k)\right]$$

Which is the desired result. □

Next we prove the following proposition:

Proposition 19. *Given assumptions 9, there exists a constant $C_{N,\alpha,T}$ depending on N, α and T only such that:*

$$\forall k > 0, \quad \mathcal{L}(k) \leq C_{N,\alpha,T}\mathcal{D}(k)$$

Proof. For a given $k > 0$, we have:

$$\mathcal{L}(k) = \sum_{n=k}^{k+T} \sum_{\{v,w\}\in\mathcal{P}_2(V)} d^2(x_v(n), x_w(n))$$

And:

$$\mathcal{D}(k) = \sum_{n=k}^{k+T} \sum_{\{v,w\}\in E(n)} P_{u,v}(n) d^2(x_v(n-1), x_w(n-1))$$

Given $x_v(n)$ and $x_w(n)$ with: $\{v, w\} \in \mathcal{P}_2(V)$ and $k \leq n \leq k+T$, we want to find a bound for $d^2(x_v(n), x_w(n))$ of the form:

$$d^2(x_v(n), x_w(n)) \leq C\mathcal{D}(k)$$

where the constant C is independent of v, w and n.

We know from assumptions 9 that $\bigcup_{n=k+1}^{k+T+1} G(n)$ is connected; and thus, there exists $l > 0$, $u_1, \ldots, u_{l+1} \in V$ and $k_1 \ldots k_l \in \{k+1, \ldots, k+T+1\}$ such that:

- $u_1 = v$ and $u_{l+1} = w$
- $\forall\ i \in \{0, \ldots, l\}$: $\{u_i, u_{i+1}\} \in E(k_i)$

Next, define the sequence $t_0, \ldots, t_l$ such that: $t_0 = t_{l+1} = n$ and for any $i \in \{1, \ldots, l\} : t_i = k_i - 1$.

The idea is that while in general it may not be possible to find in $G(n)$ a path between v and w, one can construct a path involving displacements along edges that are available at different times. From this path between v and w, we can create a path between $x_v(n)$ and $x_w(n)$ that involves only two types of displacements: displacements along an edge at a fixed time, and time increments (or decrements) along a fixed node.

This will allow us via the triangular inequality to write:

$$d(x_v(n), x_w(n)) \leq \sum_{i=1}^{l} d(x_{u_i}(t_i), x_{u_{i+1}}(t_i)) + \sum_{i=0}^{l} d(x_{u_{i+1}}(t_i), x_{u_{i+1}}(t_{i+1}))$$

The right hand side of this inequality is the sum of two terms that we will call T_1 and T_2 with:

$$T_1 = \sum_{i=1}^{l} d(x_{u_i}(t_i), x_{u_{i+1}}(t_i))$$

and:

$$T_2 = \sum_{i=0}^{l} d(x_{u_{i+1}}(t_i), x_{u_{i+1}}(t_{i+1}))$$

T_1 sums over displacements between nodes u_i and u_{i+1} at time k_i, while T_2 sums over displacements between time events k_i and k_{i+1} at node u_i.

Using Cauchy-Schwartz inequality, we have:

$$T_1^2 = (\sum_{i=1}^{l} d(x_{u_i}(k_i - 1), x_{u_{i+1}}(k_i - 1)))^2 \leq l \sum_{i=1}^{l} d^2(x_{u_i}(k_i - 1), x_{u_{i+1}}(k_i - 1))$$

$$\leq T(N-1) \sum_{i=1}^{l} \sum_{\{a,b\} \in E(k_i)} d^2(x_a(k_i - 1), x_b(k_i - 1))$$

$$\leq T(N-1) \sum_{i=k}^{k+T} \sum_{\{a,b\} \in E(k_i)} d^2(x_a(i-1), x_b(i-1))$$

$$\leq \frac{T(N-1)}{\alpha} \sum_{i=k+1}^{k+T+1} \sum_{\{a,b\} \in E(i)} P_{a,b}(i) d^2(x_a(i-1), x_b(i-1)) = \frac{T(N-1)}{\alpha} \mathcal{D}(k)$$

Thus, by taking the expected value on both sides we get:

$$\mathbb{E}[T_1^2] \leq \frac{T(N-1)}{\alpha}\mathbb{E}[\mathcal{D}(k)] \tag{5.1}$$

For the second term we have:

$$T_2^2 = (\sum_{i=0}^{l} d(x_{u_{i+1}}(t_i), x_{u_{i+1}}(t_{i+1})))^2 \leq l\sum_{i=0}^{l} d^2(x_{u_{i+1}}(t_i), x_{u_{i+1}}(t_{i+1}))$$

$$\leq T(N-1)\sum_{i=0}^{l}\sum_{a\in V} d^2(x_a(t_i), x_a(t_{i+1}))$$

$$\leq T(N-1)\sum_{i=k+1}^{k+T}\sum_{a\in V} d^2(x_a(i), x_a(i+1))$$

Taking the expected value on both sides we get:

$$\mathbb{E}[T_2^2] \leq T(N-1)\sum_{i=k+1}^{k+T}\sum_{a\in V}\mathbb{E}[\mathbb{E}[d^2(x_a(i), x_a(i+1))|F_i]]$$

$$\leq T(N-1)\sum_{i=k+1}^{k+T}\sum_{a\in V}\mathbb{E}[\sum_{b:\{a,b\}\in E(i+1)} P_{a,b}(i+1)d^2(x_a(i), x_b(i))]$$

Thus:

$$\mathbb{E}[T_2^2] \leq T(N-1)\mathbb{E}[\mathcal{D}(k)] \tag{5.2}$$

By adding equations 5.1 and 5.2 we get:

$$\mathbb{E}[d^2(x_v(n), x_w(n))] \leq \mathbb{E}[(T_1+T_2)^2] \leq 2\mathbb{E}[(T_1^2+T_2^2)] \leq \frac{2T(N-1)(\alpha+1)}{\alpha}\mathbb{E}[\mathcal{D}(k)]$$

And since: $\mathbb{E}[\mathcal{L}(k)] = \sum_{n=k}^{k+T}\sum_{\{v,w\}\in\mathcal{P}_2(V)}\mathbb{E}[d^2(x_v(n), x_w(n))]$ we get:

$$\mathbb{E}[\mathcal{L}(k)] \leq \frac{2T^2(N-1)^2N(\alpha+1)}{\alpha}\mathbb{E}[\mathcal{D}(k)]$$

Taking $C_{N,\alpha,T} = \frac{2T^2(N-1)^2N(\alpha+1)}{\alpha}$ gives us the desired result. □

We are now in a position to prove the following result:

Theorem 15 (Convergence speed)**.** *Let $X_k = (x_1(k), ..., x_N(k))$ denote the sequence of random variables generated by Algorithm RPM, under Assumptions 9 , there exists $L < 0$ such that,*

$$\mathbb{E}\sigma^2(\mathcal{L}(k)) \leq \exp(Lk)$$

Proof. We know from propositions 18 and 19 that:

$$\mathbb{E}\sigma^2(\mathcal{L}(k+1)) - \mathbb{E}\sigma^2(\mathcal{L}(k)) \leq -\frac{1}{2C_{N,\alpha,T}}\mathbb{E}\sigma^2(\mathcal{L}(k))$$

Denote by $a_n = \mathbb{E}\sigma^2(\mathcal{L}(k))$ and $\beta = \frac{1}{C_{N,\alpha,T}} < 1$, by applying lemma 4 we get the desired result with $L = -\beta = -\frac{1}{C_{N,\alpha,T}}$. □

Theorem 16 (Almost-sure convergence to consensus)**.** *There exists a random variable $X_\infty = (X_{\infty,v})_{v\in V}$, such that: (i) X_k converges almost surely to X_∞ and, (ii) almost surely, $\forall (v,w) \in V^2$, $X_{\infty,v} = X_{\infty,w}$,* i.e. *$X_\infty$ takes consensus values.*

Proof. From Theorem 15, $\mathbb{E}[\mathcal{L}(k)]$ is bounded by a decreasing exponential, which implies that $\mathcal{L}(k)$ converges almost surely to 0. As a direct consequence, the diameter $\max\{d(x_v(k), x_w(k)) : (v,w) \in V^2\}$ also tends to 0 when k goes to ∞. And as seen previously in the static case, this implies the convergence towards a consensus state

□

5.3.1 The case of $CAT(\kappa)$ spaces with $k > 0$

It is possible to extend the results above to any $CAT(\kappa)$ metric space with $\kappa > 0$ provided some additional assumptions on the initial set of data points.

Assumption 10. *The following assumptions are adapted from [NO09]:*

1. *$(\mathcal{M}, d)$ is a $CAT(k)$ metric space with $\kappa > 0$*
2. *The initial data set is of diameter $< D_\kappa$*
3. *$(V_k, W_k)_{k\geq 0}$ are i.i.d random variables, such that:*
   - *(V_k, W_k) is independent from $X(0), \ldots, X(k-1), (V_0, W_0), \ldots, (V_{k-1}, W_{k-1})$,*
   - $\mathbb{P}[\{V_k, W_k\} = \{v, w\}] = P_{u,v}(k)\delta_{\{\{v,w\}\in E(k)\}}$
4. *$\exists\, T > 0$ such that for any $k > 0$ we have: $G(k) \cup G(k+1) \cup \cdots \cup G(k+T)$ is a connected graph.*

To prove that assumptions 10 lead to a consensus state we need to define the $CAT(\kappa)$ version of the disagreement and variance functions.

Definition 27. *Given a configuration $x = (x_1, \dots, x_N) \in \mathcal{M}^N$, the $CAT(\kappa)$-disagreement function at instant k is:*

$$\Delta_\kappa(x,k) = \sum_{\{v,w\}\in E(k)} P_{v,w}(k)\chi_\kappa\left(d(x_v,x_w)\right)$$

*The $CAT(\kappa)$-*variance *function is defined for any configuration x as:*

$$\sigma^2_\kappa(x) = \frac{1}{N}\sum_{\{v,w\}\in\mathcal{P}_2(V)} d^2(x_v,x_w)$$

Definition 28. *For a given integer $k \geq 1$ we define the stochastic processes:*

$$\mathcal{L}_\kappa(k) = \sum_{n=k}^{k+T} \sigma^2_\kappa(X(n))$$

$$\mathcal{D}_\kappa(k) = \sum_{n=k}^{k+T} \Delta_\kappa(X(n), n+1)$$

The following proposition is analogous to proposition 14 and can be derived using a similar proof

Proposition 20.

$$\mathbb{E}[\sigma^2_\kappa(x(k)) - \sigma^2_\kappa(x(k-1))] \leq \ -\frac{1}{N}\mathbb{E}\Delta_\kappa(x(k-1),k)$$

Summing the relationship over a period we get:

Proposition 21. *We have the following relationship between $\mathcal{L}_\kappa$ and $\mathcal{D}_{kappa}$ for every integer k and real number $\kappa > 0$:*

$$\mathbb{E}[\mathcal{L}_\kappa(k+1) - \mathcal{L}_\kappa(k)] \leq -\frac{1}{N}\mathbb{E}[\mathcal{D}_\kappa(k)]$$

We also have the proposition:

Proposition 22. *Given assumptions 9, there exists a constant $C_{N,\alpha,\kappa,T}$ depending on N, α, κ and T only such that:*

$$\forall k > 0, \quad \mathcal{L}(k) \leq C_{N,\alpha,\kappa,T}\mathcal{D}(k)$$

Proof. We have that $\frac{2\kappa}{\pi^2}x^2 \leq \chi_\kappa(x) \leq \frac{\kappa}{2}x^2$ when $0 \leq x < \frac{\pi}{2\sqrt{\kappa}}$. Hence, under Assumption 10, χ_κ and d are equivalent and using proposition 19 we have the result □

propositions 21 and 22 coupled with lemma 4 lead to a convergence result with exponential speed for the $CAT(\kappa)$ case similar to theorems 15 and 16 under assumptions 10.

5.4 Directed graphs

5.4.1 Motivation for directed graphs

In the case where two way communications become impossible, we want to adapt the framework and the (RPM) algorithm so as to be able to perform some form of gossip in the network.

5.4.2 Framework

The network topology is now modeled by a time-invariant directed graph $G = (V, E)$, with $E \subset V \times V$ being the set of directed edges which represents the communication links between agents. We denote by $e = (v, w)$ the communication link from $v \in V$ towards $w \in V$, the set of neighbors of a given agent v is denoted by $\mathcal{N}(v)$. This time the 'neighbor' relationship is not symmetric, meaning that if w is a neighbor of v, that does not necessarily mean that w is a neighbor of v. For this reason we will drop the term 'neighbors' in what follows. Instead, we will cal $\mathcal{N}(v)$ the set of *reachable agents* of v.

The synchronization model of the agents is still the same, the system is modeled by a global Poisson clock of intensity $N\lambda$. At each tick of the clock one agent is awakened, this agent can perform some computation and communicate with his reachable agents. Like in the original (RPM) framework, the agent can awaken another agent from its set of reachable agents.

The adapted (RPM) algorithm is as follows, at each step k, one V_k agent is awakened, this agent wakes up another agent W_k form its reachable agents set (assuming it is not empty, else the agent does nothing), V_k communicates its value $x_{V_k}(K-1)$ to W_k which then updates its value to $\frac{x_{V_k}(k-1)+x_{W_k}(k-1)}{2}$, the algorithm becomes:

Algorithm Directed (RPM)

Input: a graph $G = (V, E)$ and the initial nodes configuration $X_v(0), v \in V$
for all $k > 0$ **do**
At instant k, uniformly randomly choose a node V_k from V and a node W_k uniformly randomly from $\mathcal{N}(V_k)$ (If $\mathcal{N}(V_k)$ is not empty).
Update:
$x_{W_k}(k) = \frac{x_{V_k}(k-1)+x_{W_k}(k-1)}{2}$
$x_v(k) = x_v(k-1)$ for $v \neq W_k$.
end for

5.4.3 Convergence study

We say that a directed graph $G = (V, E)$ is strongly connected if and only if for every $v, w \in V$ there exists a series of directed edges in E linking v to w and a series of edges (not necessarily the same), linking w to v. We make the following assumptions on the graph and data space:

Assumption 11.

$G = (V, E)$ is a directed, strongly connected graph, and it does not change over time

$\mathcal{M}$ is a $CAT(0)$ metric space

Using the directed (RPM) algorithm under the above assumptions we should be able to reach a consensus state with a linear rate of convergence. This expressed via the thorem:

Proposition 23. *Under Assumptions 11 and for X_k given by the new (RPM) algorithm, the following inequality holds, for every $k \geq 1$.*

$$\mathbb{E}[\sigma^2(x_k) - \sigma^2(x_{k-1})] \leq -\frac{1}{4}\mathbb{E}[\Delta(x_{k-1})]$$

Where Δ and σ^2 are respectively the disagreement and variance function as defined in the static graphs case.

Proof. Taking into account that at round k, two nodes with indices V_k and W_k woke up, it follows using the Bruhat-Tits inequality:

$$N(\sigma^2(x_k) - \sigma^2(x_{k-1})) \leq -\frac{1}{4}d^2(x_{V_k}(k-1), x_{W_k}(k-1))$$

$$+\frac{1}{2}\sum_{\substack{u \in V \\ u \neq V_k, u \neq W_k}} T(V_k, W_k, u)$$

where $T(V_k, W_k, u) = d^2(x_u(k-1), x_{W_k}(k-1)) - d^2(x_u(k-1), x_{V_k}(k-1))$

And Since V_K is chosen randomly in the graph while $W_k \in \mathcal{N}(V_K)$

$$\frac{1}{2}\mathbb{E}[d^2(x_{V_k}(k-1), x_u(k-1))] \geq \frac{1}{2}\mathbb{E}[d^2(x_u(k-1), x_{W_k}(k-1))]$$

Which gives the proposition. □

Proposition 19 remains true under assumptions 11

Proposition 24. *Given assumptions 11, there exists a constant C_N depending on N and the graph only such that:*

$$\forall k > 0, \quad \sigma^2(k) \leq C_N \Delta(k)$$

The two propositions plus lemma 4 gives the convergence and speed theorem.

Theorem 17 (Convergence speed)**.** *Let $x_k = (x_1(k), ..., x_N(k))$ denote the sequence of random variables generated by the undirected (RPM) algorithm. Under Assumptions 11, there exists $L \in (-1, 0)$ such that $\forall\ k \in \mathbb{N}$:*

$$\mathbb{E}\sigma^2(x_k) \leq \exp(Lk)$$

Chapter 6

Conclusion

The context of this thesis was on gossip protocol and how it can be used to solve the consensus problem in distributed sensor networks. We first reviewed how gossip is implemented for both Euclidean and Riemannian data spaces, in synchronous and asynchronous time models. We reviewed current techniques that allow distributed gradient descent and distributed averaging to be extended to Riemannian manifolds: either by embedding the manifold into a larger Euclidean space, or by using intrinsic geometry. We discussed the conditions under which one can obtain convergence of these algorithms.

Next, we presented our proposal which extends the gossip protocol to the case of $CAT(\kappa)$ metric spaces in the asynchronous pairwise case: the RPM algorithm. We identified a set of conditions on the curvature ($\kappa \leq 0$) that guarantee a global convergence of RPM. For $\kappa > 0$, a local convergence result has been proven. The algorithm converges in each case towards an arbitrary consensus state at exponential speed. Our experiments with positive definite matrices, the sphere and the three dimensional special orthogonal group agree with theoretical results and validate our approach. We also saw how the $CAT(\kappa)$ framework can lead to new applications that were not covered by Euclidean and Riemannian frameworks, such as metamorphic systems.

The $CAT(\kappa)$ metric framework allows to generalize the pairwise gossip protocol in many cases of non Euclidean data while preserving the exponential rate of convergence to a consensus state. Our approach is however limited by its ability to find either a closed-form expression for the midpoint of any two points, or a least to find a way to compute it. Another limitation in the case of positive sectional curvature, is that we need extra conditions on the initial configuration of the system, to ensure that data points are close enough in their initial state.

An interesting question to examine would be whether or not one can drop the conditions on the initial configuration of the network in the case of $CAT(\kappa)$ metric spaces with $\kappa > 0$. Dropping the condition on the initial state will likely cause

the loss of the uniqueness of midpoints of a given segment. However, the RPM algorithm could be modified so as to choose one of the midpoints at random. It could be the case that such a random choice eventually lead to a close enough configuation of points so that convergence can take place.

Appendix A

A.1 Graph theory reminder

In the framework of pairwise gossip in a network, we usually represent the communication links of the network by a graph. Since future definitions will rely on graph theory concepts, this section contains reminders of notions like graph connectivity and the Laplacian matrix of a graph.

A.1.1 Fundamental notions

Definition 29. *A graph $G = (V, E)$ is composed of a finite set V of vertices and a finite set E whose elements are called edges. We can distinguish two main types of graphs:*

- *The directed graph: Where E is a subset of $V \times V$, the edges are ordered pairs (v, w) with v and $w \in V$.*
- *The undirected graph: Where E is a subset of $\mathcal{P}_2(V)$, the edges are unordered pairs $\{v, w\} = \{w, v\}$ with v and w in V.*

In both types of graph we exclude single vertex edges or loops, which are of the form: (v, v) or $\{v, v\}$. In an undirected graph, if an edge exists between two nodes, they are said to be neighbors.

Definition 30. *Neighborhood and Degree We denote by $\mathcal{N}(v)$ the set of all graph neighbors of the agent $v \in V$. The number of elements in $\mathcal{N}(v)$ is referred to as the* degree *of v and denoted* $\deg(v)$*. The degree of a graph* $\deg(G)$ *is the maximum of its node degrees.*

Definition 31. *Connectivity An undirected graph G is said to be connected if and only if for every two agents u, v, there exists a finite sequence of vertices $w_0 = u, \ldots, w_d = v$ such that:*

$$\forall 0 \leq i \leq d-1 : \{w_i, w_{i+1}\} \in E$$

This means that each two agents are at least indirectly related.

The distance between u and v is the cardinal of the smallest of such sequences minus one.

The diameter of the graph: $\mathrm{diam}(G)$ *is the maximum distance between any two of its vertices.*

For directed graphs we have the notion of strong connectivity which defines a graph G as being strongly connected if and only if for every two agents v and w in V there exists a finite sequence of vertices $w_0 = u, \ldots, w_d = v$ such that:

$$\forall 0 \leq i \leq d-1 : (w_i, w_{i+1}) \in E$$

A.1.2 Stochastic matrices

Definition 32. *A $N \times N$ matrix is called stochastic if all its entries are nonnegative and the sum of each column vector is* 1*. It is said ti be bi-stochastic if all its entries are nonnegative and the sum of each row and column vectors is* 1*.*

A stochastic matrix is said to be regular if and only if all the entries of some power of the matrix are positive.

Proposition 25. *[HJ12] For any integer p, if A is a stochastic matrix, then A^p is also a stochastic matrix.*

Proposition 26 (Eigenvalues of stochastic matrices)**.** *[Chu97] If A is a stochastic matrix then:*

- *A has eigenvalue* 1*.*
- *all the other eigenvalues of A have absolute value smaller or equal to* 1*.*

A.1.3 Laplacian matrix

One important notion in the theory of graphs is that of the adjacency matrix, which provides a simple way to encode relationships between the nodes of a graph. The Laplacian of a graph will be very important for studying the convergence of synchronous gossip algorithms in this chapter.

Definition 33 (Adjacency matrix and Laplacian)**.** *If $G = (V, E)$ is a graph with N vertices, then define the* adjacency matrix *as the $N \times N$ matrix $(A)_{1 \leq i,j \leq N}$ such that:*

$$A_{i,j} = \begin{cases} 1 & \text{if } i \sim j \\ 0 & \text{otherwise} \end{cases}$$

Define also the Laplacian *of the graph G as the $N \times N$ matrix $(L)_{1\leq i,j\leq N}$ such that:*

$$L_{i,j} = \begin{cases} \deg(i) & \text{if } i = j \\ -1 & \text{if } i \sim j \\ 0 & \text{otherwise} \end{cases}$$

The Laplacian can be written as: $L = D - A$ with $D = \text{diag}(\deg(v_1), \ldots, \deg(v_N))$ and A the adjacency matrix of G. It is a symmetric matrix, its eigenvalues satisfy: $0 = \lambda_1^L \leq \lambda_2^L \leq \cdots \leq \lambda_N^L$ (see [HJ12, Th. 7.2.1]). The number of eigenvectors associated with this eigenvalue is equal to the number of connected components of the graph G [Chu97, Lemma 1.7].

This means that if i and j belong to the same connected component then $x_i = x_j$. If the graph G is connected then the only eigenspace associated with the eigenvalue 0 is $\text{Vect}\{(1, \ldots, 1)\}$, this observation will be useful in a future section.

Appendix B

B.1 Generalities on Riemannian manifolds

B.1.1 Manifolds and tangent space

If the structure we wish to study does not behave like a Euclidean space, we could see if locally we can still study it like a Euclidean space. This can be achieved via *charts*:

Definition 34 (Charts)**.** *Let $\mathcal{M}$ be a set. A* d-dimensional C^k chart (U, ϕ) *consists of a subset $U \subset \mathcal{M}$ and a C^k-diffeomorphism ϕ from U towards an open set of $\mathbb{R}^d$. The dimension d may depend on the chart (U, ϕ).*

The point of a chart is to reduce the study of $U \in \mathcal{M}$ to that of a subset of $\mathbb{R}^n$, the set $\mathcal{M}$ can thus be assigned a local coordinate system.

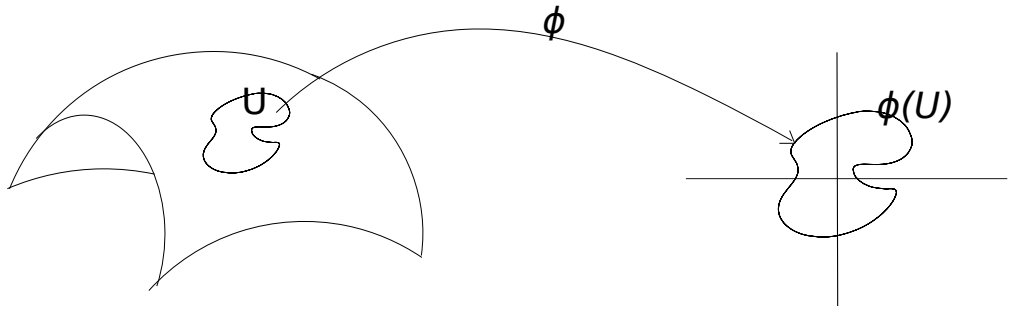


Figure B.1

If a point $x \in \mathcal{M}$ belongs to two charts (U_1, ϕ_1) and (U_2, ϕ_2) then the two charts must be *compatible* with one another. For example if a function f is defined on $U_1 \cap U_2$ then $f \circ \phi_1^{-1}$ and $f \circ \phi_2^{-1}$ must have the same differentiability proprieties. This will allow us to transpose some notions of differentiable calculus to our set.

For this, the notion of *Atlas* is introduced, and as we shall see later, it will allow us to define the notions of differentiable function in a consistent manner.

Definition 35 (Atlas)**.** *Let $\mathcal{M}$ be a set; a* $\mathcal{C}^\infty$-atlas *of $\mathcal{M}$ is a collection of charts (U_α, ϕ_α) of the set $\mathcal{M}$ such that:*

- $\cup_\alpha U_\alpha = \mathcal{M}$
- *For any pair α, β with: $\phi_\alpha(U_\alpha) \subset \mathbb{R}^{d_1}$, $\phi_\beta(U_\beta) \subset \mathbb{R}^{d_2}$, and $U_\alpha \cap U_\beta \neq \emptyset$, we have: $d_\alpha = d_\beta$ and the sets $\phi_\alpha(U_\alpha \cap U_\beta)$ and $\phi_\beta(U_\alpha \cap U_\beta)$ are open sets in $\mathbb{R}^d$, and the map:*

$$\tau_{\alpha,\beta} = \phi_\beta \circ \phi_\alpha^{-1} : \mathcal{R}^d \to \mathcal{R}^d$$

  is $\mathcal{C}^\infty$

Equipped with an atlas, $\mathcal{M}$ is called a smooth real manifold*. And if $\mathcal{M}$ is connected, then d is independent of the chosen chart (U, ϕ), and we say that $\mathcal{M}$ is a d-dimensional manifold.*

In this manuscript, we will consider only connected manifolds.

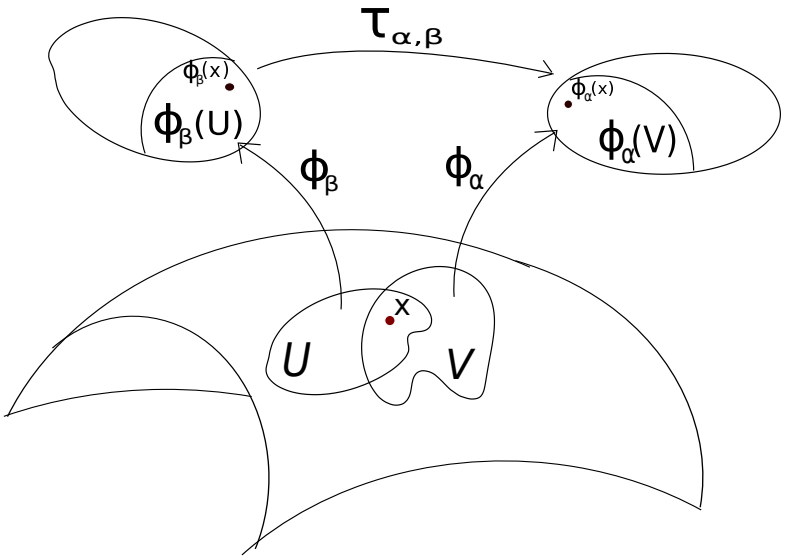


Figure B.2 – In an atlas, any overlapping charts are in correspondence via a C^∞ invertible map.

Examples of real manifolds are:

- The sphere:

$$S^n = \{x = (x_1, \ldots, x_{n+1}) \in \mathbb{R}^{n+1} | x_1^2 + \cdots + x_{n+1}^2 = 1\}$$

is an n-dimensional manifold with two charts (U_1, ϕ_1) and (U_2, ϕ_2) with:

$$U_1 = \{x \in S^n | x_{n+1} > -1\}, \quad \phi_1(x) = \left(\frac{x_1}{1+x_{n+1}}, \ldots, \frac{x_n}{1+x_{n+1}}\right)$$

and:

$$U_2 = \{x \in S^n | x_{n+1} < 1\}, \quad \phi_2(x) = \left(\frac{x_1}{1-x_{n+1}}, \ldots, \frac{x_n}{1-x_{n+1}}\right)$$

(See figure B.3).

- The projective space $\mathbb{R}P^n$ is the set of lines of $\mathbb{R}^{n+1}$ that pass through the origin. Another way to define it is: $\mathbb{R}P^n = (\mathbb{R}^{n+1} - \{0\})/\sim$ with $\sim$ being the equivalence relationship $(x_1, \ldots, x_{n+1}) \sim (y_1, \ldots, y_{n+1})$ if and only if $\exists \lambda \in \mathbb{R} - \{0\}$ such that: $(x_1, \ldots, x_{n+1}) = (\lambda y_1, \ldots, \lambda y_{n+1})$.

  For $i = 1 \ldots n+1$ denote the set:

$$U_i = \{[x_1, \ldots, x_{n+1}] : x_i \neq 0\}$$

  and the map

$$\phi_i : U_i \to \mathbb{R}^n \quad [x_1, \ldots, x_{n+1}] \to \left(\frac{x_1}{x_i}, \ldots, \frac{x_{n+1}}{x_i}\right)$$

  Equipped with the charts, $(U_i, \phi_i)_{1 \leq i \leq n+1}$ is an n-dimensional smooth manifold.

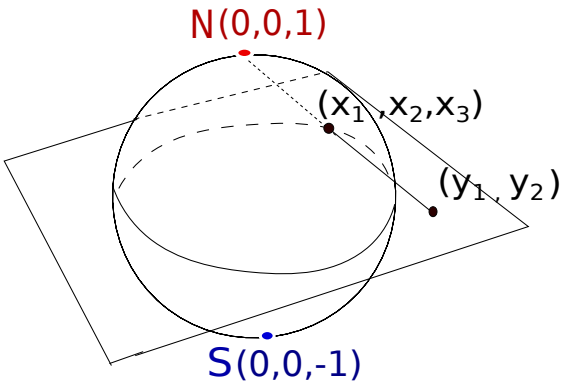


Figure B.3 – Example of a chart for the 3-dimensional sphere. The point (x_1, x_2, x_3) with $x_3 > 0$ is mapped to the point $(y_1, y_2) = \left(\frac{x_1}{1-x_3}, \frac{x_2}{1-x_3}\right)$

Definition 36. *Given a map F between a manifold $\mathcal{M}_1$ to another manifold $\mathcal{M}_2$. We say that F is smooth at a point $x \in \mathcal{M}_1$ if we can find charts ϕ_1 and ϕ_2 around x and $F(x)$ such that the function: $\phi_2 \circ F \circ \phi_1^{-1}$ is $\mathcal{C}^\infty$.*

Definition 37. *A manifold $\mathcal{L}$ is said to be a* submanifold *of $\mathcal{M}$ if and only if:*

1. *$\mathcal{L}$ is a topological subspace of $\mathcal{M}$*
2. *The inclusion map $i : \mathcal{L} \to \mathcal{M}$ is smooth a each point $p \in \mathcal{L}$ and $d(i)(p)$ is a bijection.*

Given a real valued function $f : \mathcal{M} \to \mathbb{R}$, we would like to find a way to define directional derivatives of f on $\mathcal{M}$. Directional derivatives are useful in optimization, like for example in gradient descent; if one wishes to choose a descent direction it would generally be the one minimizing the directional derivative.

Since $\mathcal{M}$ is not necessarily a vector space, the formula

$$df(x)[v] = \lim_{t \to 0} \frac{f(x+tv) - f(x)}{t}$$

is no longer valid. We instead use the notion of curves of $\mathcal{M}$, and the tangent vectors to these curves. A curve is a map $c : \mathbb{R} \to \mathcal{M}$, if the function $t \to f(c(t))$ can be shown to be differentiable then the quantity $\frac{df \circ c}{dt}(0)$ can be defined as the directional derivative of f.

Definition 38 (Equivalent curves)**.** *[AMS09] Given a smooth manifold $\mathcal{M}$ of dimension d, and two curves $c_1, c_2 : [-\epsilon, \epsilon] \to \mathcal{M}$ such that: $c_1(0) = c_2(0) = p$.*

c_1 and c_2 are said to be equivalent if and only if there exists some chart (U, ϕ) at p such that:

$$\frac{d(\phi \circ c_1)}{dt}(0) = \frac{d(\phi \circ c_2)}{dt}(0)$$

Definition 39 (Tangent vector)**.** *[AMS09] Let $\mathcal{M}$ be a manifold, $p \in \mathcal{M}$. Let $\mathbb{F}_p(\mathcal{M})$ be the set of real valued functions on $\mathcal{M}$ that are locally smooth at p. A tangent vector ∂_p to $\mathcal{M}$ at a point p is a mapping from $\mathbb{F}_p(\mathcal{M})$ to $\mathbb{R}$ such that there exists a curve c on $\mathcal{M}$ with $c(0) = p$ satisfying:*

$$\partial_p f := \frac{d(f \circ c)}{dt}(0)$$

$\forall f \in \mathbb{F}_p(\mathcal{M})$

This definition does not depend on a specific choice of c since any equivalent curve would satisfy the definition.

Proposition 27 (Tangent space, tangent bundle)**.** *[AMS09] The set of all tangent vectors to a manifold $\mathcal{M}$ at a given point $p \in \mathcal{M}$ forms a vector space and is denoted $T_p\mathcal{M}$.*

For $p \in \mathcal{M}$, it is possible to define a basis for the tangent space $T_p\mathcal{M}$, $(\partial_1, \partial_2, \ldots, \partial_d)$, this means that $T_p\mathcal{M}$ is of dimension d for all p.

The set $T\mathcal{M} = \cup_{p\in\mathcal{M}} T_p\mathcal{M}$ is called, the tangent bundle

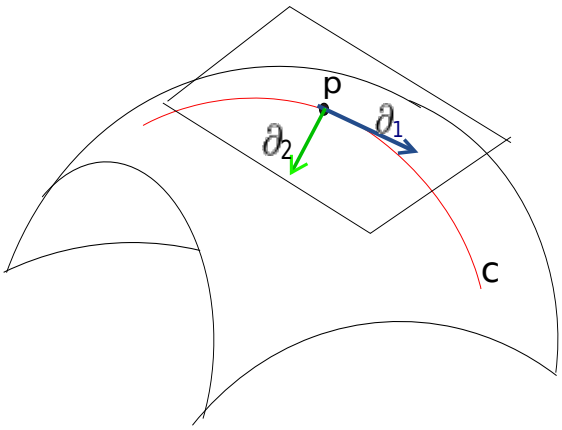


Figure B.4 – A tangent space at $p \in \mathcal{M}$, with basis ∂_1 and ∂_2.

Definition 40 (Vector fields)**.** *[AMS09]*

A smooth vector field *is a $\mathcal{C}^\infty$ map $X : \mathcal{M} \to T\mathcal{M}$.*

The set of vector fields on $\mathcal{M}$ is denoted by $\Gamma(T\mathcal{M})$.

Let $\gamma : [0, 1] \to \mathcal{M}$. We want to differentiate a vector field $X = \frac{d\gamma}{dt}(t)$. The formula $\frac{dX}{dt}(t_0) = \lim_{t\to 0} \frac{X(\gamma(t))-X(\gamma(0))}{t}$ cannot work since $X(\gamma(t))$ and $X(\gamma(0))$ belong to two different vector spaces (respectively $T_{\gamma(t)}\mathcal{M}$ and $T_{\gamma(0)}\mathcal{M}$). There is no general correspondence between tangent vector spaces that can allow us to define a differential of a vector space. We need to introduce such a correspondence using *affine connections*, which are the equivalent of directional derivatives for vector fields.

Definition 41 (Connections)**.** *A* connection ∇ *on a Riemannian manifold $\mathcal{M}$ is a map $\nabla : T\mathcal{M} \times \Gamma(T\mathcal{M}) \to T\mathcal{M}$, $(v, Y) \to \nabla_v Y$ such that $\forall\ p \in \mathcal{M}$, $\alpha, \beta \in \mathbb{R}$ $u, v \in T\mathcal{M}$, $Y \in \Gamma(T\mathcal{M})$:*

$$\nabla_{\alpha u+\beta v} Y = \alpha \nabla_u Y + \beta \nabla_v Y$$

And for all $Y_1, Y_2 \in \Gamma(T\mathcal{M}))$, $u \in T\mathcal{M}$, and f a smooth function on $\mathcal{M}$:

$$\nabla_u(Y_1 + Y_2) = \nabla_u Y_1 + \nabla_u Y_2$$

$$\nabla_u(fY) = (v.f)Y(p) + f(p)\nabla_u Y$$

Definition 42. *Let $\mathcal{M}$ be a manifold with a connection ∇. A vector field X along a curve c is said to be* parallel *along c if $\nabla_{\dot{c}} X = 0$.*

Theorem 18. *[Lee06] Given a curve $c : [0,1] \to \mathcal{M}$ and a vector $\xi_0 \in T_{\dot{c}}\mathcal{M}$, there exists a unique parallel vector field X along c such that: $X(0) = X_0$*

If $t_0, t_1 \in I$, Parallel translation *defines an operator:*

$$P_{t_0,t_1} : T_{c(t_0)}\mathcal{M} \to T_{c(t_1)}\mathcal{M}$$

such that: $P_{t_0,t_1}(\xi_0) = X(t_1)$, where X is a parallel vector field along c with respect to ∇ such that: $X(0) = \xi_0$

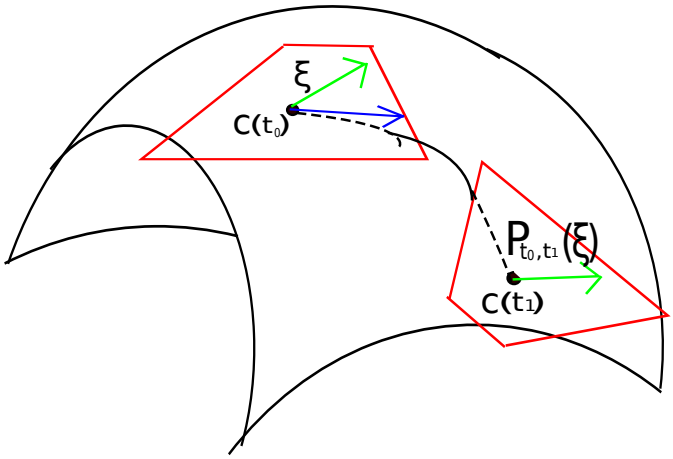


Figure B.5

Riemannian metrics

A metric $\langle , \rangle$ on a smooth Riemannian manifolds is a collection of Euclidean scalar products $(\langle , \rangle_p)_{p \in \mathcal{M}}$ on each of the tangent spaces $T_p\mathcal{M}$ of $\mathcal{M}$. We also require that $\langle , \rangle_p$ varies *smoothly* with respect to p.

Definition 43 (Riemannian metric)**.** *Given a smooth manifold $\mathcal{M}$, define a Riemannian metric $\langle , \rangle$ on $\mathcal{M}$ to be a mapping that associates each $p \in \mathcal{M}$ to a scalar dot product $\langle , \rangle_p : T_p\mathcal{M} \times \mathcal{M} \to \mathbb{R}$ such that: If X and Y are smooth vector fields of $\mathcal{M}$ then the function: $p \to \langle X_p, Y_p \rangle$ is smooth.*

A Riemannian manifold is a smooth manifold equipped with a Riemannian metric.

A special kind of connection we will be interested in is the *Levi-Civita* connection, whose existence and uniqueness is given by the following lemma:

Lemma 5. *[Cha06] If $\mathcal{M}$ is a Riemannian manifold, then there exists a unique connection ∇ such that for all smooth vector fields X, Y and Z we have:*

$$\nabla_X Y = \nabla_Y X + [X, Y]$$
$$X\langle Y, Z\rangle = \langle \nabla_X Y, Z\rangle + \langle Y, \nabla_X Z\rangle$$

Definition 44. *Let $v \in T_p\mathcal{M}$ be a tangent vector to the manifold $\mathcal{M}$ at p. Then there is a unique geodesic γ_v satisfying $\gamma_v(0) = p$ with initial tangent vector $\gamma_v'(0) = v$. The exponential map at p is defined by $\exp_p(v) = \gamma_v(1)$ for any $v \in T_p\mathcal{M}$.*

The injectivity radius at a point $p \in \mathcal{M}$ is the quantity: $\mathrm{inj}(p, \mathcal{M}) = \sup_{r>0}\{r | \exp_p : B(0, r) \to B(p, r)$ is a diffeomorphism $\}$. The injectivity radius of the whole manifold $\mathcal{M}$ is defined as: $\mathrm{inj}(\mathcal{M}) = \inf_{p\in\mathcal{M}} \mathrm{inj}(p, \mathcal{M})$

A manifold is said to be geodesically complete *if the exponential map is defined on all of TM*

Sectional curvature and convexity

If c is a curve on $\mathcal{M}$, we define the length of the curve as the quantity:

$$L(c) := \int_0^1 \langle c'(t), c'(t)\rangle_{c(t)} \mathrm{d}t$$

We denote as ∇ the Levi-Civita connection on $\mathcal{M}$. If X, Y and Z are three vector fields, the *Riemannian curvature* tensor R is defined as:

$$R(X, Y)Z = \nabla_X \nabla_Y Z - \nabla_Y \nabla_X Z - \nabla_{[X,Y]} Z$$

In what follows, we shall make use of the following notation: $R(X, Y, Z, W) = \langle R(X, Y)Z, W\rangle$. Given a point $p \in \mathcal{M}$ and two linearly independent vectors $v, w \in T_p\mathcal{M}$ spanning a two dimensional space $\sigma = \mathrm{span}\{v, w\}$, the *sectional curvature* at p for the subspace σ is defined as:

$$K_\sigma(p) = \frac{R(v, u, v, u)}{\|u\|^2\|v\|^2 - \langle u, v\rangle^2}$$

Using this definition, the notion of distance between two points x and $y \in \mathcal{M}$ is defined by:

$$d(x, y) := \inf_{\{c | c(0)=x; c(1)=y\}} L(c)$$

A curve $\gamma(t)$ between x and y verifying: $L(\gamma) = d(x, y)$ is called a *geodesic*. If such a curve exists and is unique, we denote it by $[x, y]$. This notation suggests that geodesics are the Riemannian analog to the Euclidean line segment. The manifold $\mathcal{M}$ is said to be *complete* if for every couple $(x, y) \in \mathcal{M}^2$ a geodesic between x and y exists.

We also define *the midpoint* of x and y (assuming existence and uniqueness of the geodesic):

$$\frac{x+y}{2} := [x, y]\big(\frac{1}{2}\big)$$

It is important to notice that notation $\langle\frac{x+y}{2}\rangle$ denotes the midpoint of $[x, y]$ in the manifold and involves actually no addition nor dilation. Obviously in the Euclidean case the two notions ("midpoint" and "arithmetic mean") coincide.

For $p \in \mathcal{M}$ and σ a 2-dimensional subspace of $T_p\mathcal{M}$, we denote by $K_p(\sigma)$ the *sectional curvature* of σ [DC92]. A manifold is said to be of *bounded curvature* if and only if: $\exists\ \delta, \Delta \in \mathbb{R}$ such that: $\delta \leq K_\sigma(p) \leq \Delta$ for any point $p \in \mathcal{M}$.

We define the *convexity radius* $r_\mathcal{M}$ as:

$$r_\mathcal{M} = \frac{1}{2} \min\left\{\text{inj}(\mathcal{M}), \frac{\pi}{\sqrt{\Delta}}\right\}$$

where $\text{inj}(\mathcal{M})$ is the injectivity radius.

With these notions defined, we can start describing the first optimization tools: the gradient and the Hessian.

Differential operators

Definition 45 (gradient)**.** *Let $f\mathcal{M} \to \mathbb{R}$ be a smooth function. The vector field* grad f *is defined so that the relationship:*

$$\langle X, \text{grad}\, f\rangle = X(f) = df(X)$$

is valid for any vector field X on $\mathcal{M}$.

Remark 17. It can be shown that the gradient field verifies the following relationships:

1. $\text{grad}(\lambda_1 f_1 + \lambda_2 f_2) = \lambda_1\, \text{grad}\, f_1 + \lambda_2\, \text{grad}\, f_2$
2. $\text{grad}(f_1 f_2) = f_1\, \text{grad}(f_2) + f_2\, \text{grad}(f_1)$
3. $\text{grad}(\frac{f_1}{f_2}) = \frac{f_2\, \text{grad}\, f_1 - f_1\, \text{grad}\, f_2}{f_2^2}$

Definition 46. *Let $\mathcal{M} \to \mathbb{R}$ be a smooth function. The* Hessian *of f is defined as:* $\text{Hess}(f) = \nabla(df)$.

Remark 18. The Hessian also verifies the relationship:

$$\text{Hess } f(X,Y) = \langle \nabla_X \text{ grad } f, Y \rangle$$

for any vector fields X and Y.

A word on Lie Groups and Homogeneous manifolds

A Lie group is a group that is also a finite-dimensional smooth manifold, such that the multiplication and inversion mappings are smooth. These two conditions can be summed up by stating that the map $(x, y) \to x.y^{-1}$ is a smooth map from the product manifolds $G \times G$ to G.

Examples of Lie groups are: The unit circle S^1, the n-dimensional torus $T^n = S^1 \times \cdots \times S^1$, the group of invertible $n \times n$ matrices $GL_n(\mathbb{R})$.

A manifold $\mathcal{M}$ is said to be homogeneous with respect to a Lie group G if there exists a transitive action of G on $\mathcal{M}$.

B.1.2 Examples

Grassmannian manifolds

The first example of interest are Grassmann manifolds, which arise naturally in several vision applications such as image matching and learning theory [TVC08, Chi12].

A Grassmanian manifold $G(r, n)$ with $r, n \in \mathbb{N}$ is the quotient of $\mathbb{R}^{n \times r}$ by the equivalence relationship:

$$X \sim Y \Leftrightarrow \text{span}(X) = \text{span}(Y)$$

It can be shown [AMS09, p.48] that Grassmannians admit a manifold structure.

The tangent space at an r-dimensional subspace $W \subset \mathbb{R}^n$, $T_W G(r, n)$ can be identified to the space of linear maps from W to $\mathbb{R}^n/W$: $\text{Lin}(W, \mathbb{R}^n/W)$ [PT00].

This tangent space can be endowed with the Riemannian metric [AMS09, p.83]:

$$g_W : \text{Lin}(W, \mathbb{R}^n/W) \times \text{Lin}(W, \mathbb{R}^n/W) \to \mathbb{R}$$
$$(Z_1, Z_2) \to \text{tr}\left((Y^T Y)^{-1} Z_1^T Z_2\right)$$

Assuming Levi-Civita connection, a geodesic $\mathcal{Y}(t) \in G(r, n)$ has the form [AMS09, p.104]:

$$\mathcal{Y}(t) = \text{span}\left(Y_0 (Y_0^T Y_0)^{-\frac{1}{2}} V \cos(\Sigma t) + U \sin(\Sigma t)\right)$$

such that $\mathcal{Y}(0) = \text{span}(Y_0)$ and $\frac{d\mathcal{Y}}{dt}(0) = U \Sigma V^T$ where $U \Sigma V^T$ is the singular value decomposition of the tangent vector.

The curvature bounds are $\Delta = 2$ and $\delta = 0$ [TAV11], the injectivity radius is $\text{inj}(G(r, n)) = \frac{\pi}{2}$.

Positive definite matrices

The space of positive definite matrices is defined as:

$$\mathrm{Pos}(n) = \{M \in \mathbb{R}^{n\times n} : M^T = M, x^T M x > 0, \forall x \in \mathbb{R}^n\}$$

The tangent space at a given point $M \in \mathrm{Pos}(n)$ can be identified with the space of symmetric matrices. Equipped with the metric:

$$\langle X_1, X_2 \rangle_P = \mathrm{tr}\left(X_1 P^{-1} X_2 P^{-1}\right)$$

$\mathrm{Pos}(n)$ is a Riemannian manifold.

Equipped with this metric and the Levy-Civita connection, $\mathrm{Pos}(n)$ is a Riemannian manifold of constant negative curvature [Lan99, chap. 12], and the injectivity radius is infinite $\mathrm{inj}(\mathrm{Pos}(n)) = \infty$. The geodesic $P(t)$ such that $P(0) = P$ and $\dot{P}(0) = X$ is [Lan99, p.326]:

$$P(t) = P^{\frac{1}{2}} \exp(tP^{-\frac{1}{2}} X P^{-\frac{1}{2}}) P^{\frac{1}{2}}$$

The rotation group

Another data space of interest is the 3-dimensional rotations group $SO(3) = \{R \in \mathbb{R}^{3\times 3} : R^T R = I, \det(R) = I\}$. The tangent of $SO(3)$ at a point R is given by $T_R SO(3) = \{RV : V^T = -V\}$.

The Riemannian metric at identity is $\langle v_1, v_2 \rangle = -\frac{1}{2}\mathrm{tr}(V_1 v_2)$. In this metric, the sectional curvature is constant and positive $\Delta = \delta = \frac{1}{4}$, and the injectivity radius is $\mathrm{inj}(SO(3)) = \pi$.

For all $(R_{a_1,\theta_1}, R_{a_2,\theta_2}) \in \mathcal{B}^2$ there exists a unique minimizing geodesic $\gamma(t)$ such that $\gamma(0) = R_{a_1,\theta_1}$ and $\gamma(1) = R_{a_2,\theta_2}$, and it has the following expression:

$$\gamma(t) = R_{a_1,\theta_1} \exp\left(t \log(R_{a_1,\theta_1}^T R_{a_2,\theta_2})\right)$$

In the next section, we shall see how we can extend the gradient descent method to Riemannian manifolds. This first approach involves embedding the manifold into a Euclidean space of sufficiently large dimension.

Bibliography

[ABY12] Federico Ardila, Tia Baker, and Rika Yatchak. Moving robots efficiently using the combinatorics of CAT(0) cubical complexes. *arXiv preprint arXiv:1211.1442*, 2012.

[AG04] Aaron Abrams and Robert Ghrist. State complexes for metamorphic robots. *The International Journal of Robotics Research*, 23(7-8):811–826, 2004.

[AMS09] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

[AN00] Shun-Ichi Amari and Hiroshi Nagaoka. *Methods of Information Geometry*. Amer. Math. Soc., 2000.

[AOS12] Federico Ardila, Megan Owen, and Seth Sullivant. Geodesics in cat (0) cubical complexes. *Advances in Applied Mathematics*, 48(1):142–163, 2012.

[Bac14] Miroslav Bacak. Computing medians and means in hadamard spaces. *SIAM Journal on Optimization*, 24(3):1542–1566, 2014.

[Bar13] Frédéric Barbaresco. Information geometry of covariance matrix: Cartan-Siegel homogeneous bounded domains, Mostow/Berger fibration and Frechet median. In *Matrix Information Geometry*, pages 199–255. Springer, 2013.

[Bel15] Anass Bellachehab. Distributed consensus for metamorphic systems using a gossipalgorithm for cat (0) metric spaces. *Entropy*, 17(3):1165–1180, 2015.

[BGPS06] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. "Randomized gossip algorithms". *IEEE Trans. Inf. Theory*, 52(6):2508–2530, Jun 2006.

[BH99] Martin Bridson and André Haefliger. *Metric Spaces of Non-Positive Curvature*. Springer Verlag, 1999.

[BHT09] Vincent D Blondel, Julien M Hendrickx, and John N Tsitsiklis. On krause's multi-agent consensus model with state-dependent connectivity. *IEEE transactions on Automatic Control*, 54(11):2586–2597, 2009.

[BHV01] Louis J Billera, Susan P Holmes, and Karen Vogtmann. Geometry of the space of phylogenetic trees. *Advances in Applied Mathematics*, 27(4):733–767, 2001.

[BJ14] Anass Bellachehab and Jérémie Jakubowicz. Random pairwise gossip on $CAT(\kappa)$ metric spaces. *arXiv preprint arXiv:1405.4190*, 2014.

[BJ16] Anass Bellachehab and Jérémie Jakubowicz. Random pairwise gossip on $cat(\kappa)$ metric spaces. *IEEE Transactions on Automatic Control*, 61(12):3857–3869, 2016.

[Bon13] Silvère Bonnabel. "Stochastic gradient descent on Riemannian manifolds". *IEEE Trans. Autom. Control*, 58(9):2217–2229, Sep 2013.

[BT+07] Dimitri P Bertsekas, John N Tsitsiklis, et al. Comments on" coordination of groups of mobile autonomous agents using nearest neighbor rules". *IEEE Transactions on Automatic Control*, 52(5):968, 2007.

[Cha06] Issac Chavel. *Riemannian Geometry*. Cambridge University Press, second edition, 2006.

[Chi12] Yasuko Chikuse. *Statistics on special manifolds*, volume 174. Springer Science & Business Media, 2012.

[Chu97] Fan RK Chung. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997.

[CS77] Samprit Chatterjee and Eugene Seneta. Towards consensus: some convergence theorems on repeated averaging. *Journal of Applied Probability*, pages 89–97, 1977.

[DC92] Manfredo Do Carmo. *Riemannian Geometry*. Birkhäuser, 1992.

[DeG74] Morris H DeGroot. Reaching a consensus. *Journal of the American Statistical Association*, 69(345):118–121, 1974.

[DGH+87] Alan Demers, Dan Greene, Carl Hauser, Wes Irish, John Larson, Scott Shenker, Howard Sturgis, Dan Swinehart, and Doug Terry. Epidemic algorithms for replicated database maintenance. In *Proceedings of the sixth annual ACM Symposium on Principles of distributed computing*, pages 1–12. ACM, 1987.

[FLP85] Michael J Fischer, Nancy A Lynch, and Michael S Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the ACM (JACM)*, 32(2):374–382, 1985.

[GKG06] Indranil Gupta, Anne-Marie Kermarrec, and Ayalvadi J Ganesh. Efficient and adaptive epidemic-style protocols for reliable and scalable multicast. *Parallel and Distributed Systems, IEEE Transactions on*, 17(7):593–605, 2006.

[GP07] Robert Ghrist and Valerie Peterson. The geometry and topology of reconfiguration. *Advances in applied mathematics*, 38(3):302–323, 2007.

[Gro07] Mikhail Gromov. *Metric structures for Riemannian and non-Riemannian spaces*. Springer Science & Business Media, 2007.

[HH06] Qing Han and Jia-Xing Hong. *Isometric embedding of Riemannian manifolds in Euclidean spaces*, volume 130. American Mathematical Society, 2006.

[HJ12] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

[HSD12] Morris W Hirsch, Stephen Smale, and Robert L Devaney. *Differential equations, dynamical systems, and an introduction to chaos*. Academic press, 2012.

[HSJ15] Julien M Hendrickx, Guodong Shi, and Karl H Johansson. Finite-time consensus using stochastic matrices with positive diagonals. *Automatic Control, IEEE Transactions on*, 60(4):1070–1073, 2015.

[JL+03] Ali Jadbabaie, Jie Lin, et al. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *Automatic Control, IEEE Transactions on*, 48(6):988–1001, 2003.

[KHJ+03] Dr Sc Y Koren, U Heisel, F Jovane, Dr Eng G Pritschow, G Ulsoy, Dr Eng H Van Brussel, et al. *Reconfigurable manufacturing systems*. Springer, 2003.

[Kra00] Ulrich Krause. A discrete nonlinear and non-autonomous model of consensus formation. *Communications in difference equations*, pages 227–236, 2000.

[Lan99] Serge Lang. *Fundamentals of Differential Geometry*. Springer, 1999.

[Lee06] John M Lee. *Riemannian manifolds: an introduction to curvature*, volume 176. Springer Science & Business Media, 2006.

[LR06] Qui Li and Daniela Rus. Global clock synchronization in sensor networks. *Computers, IEEE Transactions on*, 55(2):214–226, 2006.

[Lui12] Yui Man Lui. Advances in matrix manifolds for computer vision. *Image and Vision Computing*, 30(6):380–388, 2012.

[NO09] Angelia Nedić and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *Automatic Control, IEEE Transactions on*, 54(1):48–61, 2009.

[ØKBL06] Esben Hallundbæk Østergaard, Kristian Kassow, Richard Beck, and Henrik Hautop Lund. Design of the atron lattice-based self-reconfigurable robot. *Autonomous Robots*, 21(2):165–183, 2006.

[OSFM07] Reza Olfati-Saber, Alex Fax, and Richard M Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

[OSM02] Reza Olfati-Saber and Richard M Murray. Distributed structural stabilization and tracking for formations of dynamic multi-agents. In *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, volume 1, pages 209–215. IEEE, 2002.

[PT00] Paolo Piccione and Daniel Victor Tausk. *On the geometry of Grassmannians and the symplectic group: the Maslov index and its applications*. UFF, 2000.

[SBS10] Alain Sarlette, Silvere Bonnabel, and Rodolphe Sepulchre. Coordinated motion design on lie groups. *IEEE Transactions on Automatic Control*, 55(5):1047–1058, 2010.

[SG07] Luca Schenato and Giovanni Gamba. A distributed consensus protocol for clock synchronization in wireless sensor network. In *Decision and Control, 2007 46th IEEE Conference on*, pages 2289–2294. IEEE, 2007.

[SJJ12] Guodong Shi, Mikael Johansson, and Karl Henrik Johansson. When do gossip algorithms converge in finite time? *avaiable at http://arxiv.org/pdf/1206.0992 v1. pdf*, 2012.

[SS09] Alain Sarlette and Rodolphe Sepulchre. Consensus optimization on manifolds. *SIAM Journal on Control and Optimization*, 48(1):56–76, 2009.

[SSL06] Alain Sarlette, Rodolphe Sepulchre, and Naomi Leonard. Discrete-time synchronization on the n-torus. In *17th International Symposium on the Mathematical Theory of Networks and Systems*, 2006.

[STBS08] Alain Sarlette, Sezai Emre Tuna, Vincent Blondel, and Rodolphe Sepulchre. Global synchronization on the circle. In *17th IFAC World Congress*, pages 9045–9050, 2008.

[TAV11] Roberto Tron, Bijan Afsari, and René Vidal. "Average consensus on Riemannian manifolds with bounded curvature". In *IEEE Conf. on Decision and Control*, pages 7855–7862, 2011.

[Tsi84] John Tsitsiklis. *Problems in Decentralized Decision Making and Computation*. PhD thesis, MIT, 1984.

[TVC08] Pavan Turaga, Ashok Veeraraghavan, and Rama Chellappa. Statistical analysis on stiefel and grassmann manifolds with applications in computer vision. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[TVT08] Roberto Tron, René Vidal, and Andreas Terzis. "Distributed pose averaging in camera networks via consensus on SE(3)". In *Second ACM/IEEE International Conference on Distributed Smart Cameras. ICDSC*, pages 1–10, 2008.

[XB04] Lin Xiao and Stephen Boyd. Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53(1):65–78, 2004.