الگوریتم های تصادفی لاس وگاس در مسائل اجماع توزیع شده

نرگس قانعی_۶۱۰۳۹۹۱۵۵
آران باطنی_۶۱۰۳۹۹۱۰۵
فاطمه رشیدی_۶۱۰۳۹۹۱۳۱

تیر ۱۴۰۱

چکیده_ ما مشکلات اجماع توزیع شده را از دیدگاه الگوریتم های احتمالی در نظر می گیریم به خصوص، ما یک مرور کلی در مورد برخی از مشکلات خاص ارائه می دهیم که در آن تصادفی سازی در دستیابی به اجماع بین عوامل متعدد حیاتی است. علاوه بر این، ما نشان می دهیم که الگوریتم های تصادفی شده که در این تنظیمات استفاده می شود به اصطلاح الگوریتم های تصادفی لاس وگاس هستند. (به عنوان مثال [۲۴]). این کلاس با نوع مونت کارلو که اخیرا با موفقیت برای مسائل مختلف محاسباتی دشوار سیستم ها و کنترل استفاده می شود متفاوت است. بنابراین هدف این مقاله این است که ارتباط بین مسائل مختلف اجماع توزیع شده و الگوریتم های تصادفی برای سیستم ها و کنترل را نشان بدهد.

فهرست مطالب

۱ مقدمه

در سال های اخیر، اجماع، توافق، و مسائل انبوه توجه زیادی را در سیستم ها و جامعه کنترل به خود جلب کرده است. کنترل رویکردهای نظری ثابت کرده اند که در تجزیه و تحلیل توزیع شده سیستم هایی برای مشخصاتی مانند ثبات و توافق مفید هستند. منابع اخیر شامل [۳]،[۴]،[۶]، [۷]، [۱۰]، [۱۲]، [۱۶]، [۱۸]، [۲۰]، [۲۱]، [۲۵]،[۲۷]است. برای جزئیات بیشتر به [۵] مراجعه می کنیم که خلاصه ای از توسعه چنین مسائلی همراه با برخی از نتایج جدید می دهد و به ویژه شماره [۱] که تحقیقات جاری در مورد این موضوع را شرح می دهد.

به طور خاص، ما میانگین مسائل اجماع را در نظر می گیریم که می توان آنها را به شرح زیر توصیف کرد: مجموعه ای از N عواملی که دارای مقادیر عددی هستند وجود دارد و ارزش هایشان را به همسایگان خود به شیوه ای تکراری منتقل می کنند. هدف این است که همه عوامل در نهایت به یک مقدار مشترک برسند که این میانگین مقادیر اولیه همه عوامل است. چنین مسائلی در کاربردهای مرتبط با هماهنگی چند وسیله نقلیه، تعادل بار و شبکه های حسگر ایجاد می شود.

هدف این مقاله ارائه میانگین مسائل اجماع از نقطه نظر یکپارچه الگوریتم های احتمالی و لاس وگاس می باشد اخیراً در زمینه سیستم‌ها و کنترل، تکنیک‌های مبتنی بر الگوریتم‌های تصادفی توسعه یافته‌اند (به عنوان مثال، [۲۳]، [۲۶] را ببینید). با این حال، خواهیم دید که چگونه کلاس های الگوریتم ها وجود دارد و در اجماع مسائل متفاوت عمل می کنند. با توجه به یک تعریف رسمی مورد استفاده در علوم کامپیوتر [۱۷]، یک الگوریتم تصادفی الگوریتمی است که در طول اجرای آن برای ایجاد نتیجه انتخاب هایی تصادفی می سازد. این دلالت می کند که حتی برای یک ورودی، الگوریتم ممکن است نتایج متفاوت در اجراهای مختلف تولید کند و علاوه بر آن نتایج حتی ممکن است نادرست باشد.

به طور خاص، در این مقاله، ما قصد داریم دو مورد را روشن کنیم نکته ها: یکی از آنها معرفی الگوریتم‌های تصادفی‌سازی‌شده است که در چندین مورد متفاوت از مسائل اجماع متوسط ظاهر می‌شوند. چنین تکنیک هایی نشان داده شده است که مفید هستند و می توانند حیاتی باشند. در واقع، در برخی موارد، نتایج حتی قوی تری وجود دارد که نشان می دهد هیچ روش قطعی ای نمی تواند به طور کارآمد به دست آید. اجماع [۹]، [۱۷]. دیگری نشان دادن تفاوت در کلاس های الگوریتم هایی که در مسائل اجماع ظاهر می شوند و آنهایی که در رویکرد احتمالی در کنترل هستند است. در پایان، ما یک نمای کلی در مورد رویکرد احتمالی در کنترل ارائه می دهیم و ببینید که الگوریتم های اصلی از نوع مونت کارلو وجود دارند. سپس، نشان می‌دهیم که الگوریتم‌های موجود در اجماع متوسط متعلق به الگوریتم های نوع کلاس لاس وگاس است. این کلاس اخیرا برای مسائل مربوط به سیستم ها و کنترل مورد سوء استفاده قرار گرفته است.[۱۱]، [۲۴].

مقاله بصورت زیر مرتب شده است: در بخش دوم، ما یک مسئله تحلیل استحکام کلی ارائه می دهیم. در بخش سوم، الگوریتم های تصادفی از نوع مونت کارلو برای چنین مسائلی مورد بحث قرار می گیرد. در بخش چهارم، مقدمه ای بر الگوریتم های لاس وگاس ارائه می دهیم. در بخش پنچم، سه مورد از مسائل اجماع متوسط را مورد بحث قرار داده واهمیت تکنیک های احتمالی را نشان می

دهیم. در بخش ششم مقاله نتیجه می گیریم.

۲ رویکرد احتمالی به سیستم های نامشخص

در دهه گذشته روشهای احتمالاتی برای سیستمها و کنترل به طور قابل توجهی پیشرفت کرده است. این تحقیق و همچنین نتایج نشان می دهد که بسیاری از مسائلی که به طور طبیعی در کنترل ایجاد می شوند از نظر محاسباتی دشوار هستند. چنین مسائلی در مناطقی از جمله سیستم های نامشخص و ترکیبی یافت می شود. این توسعه و کاربرد تکنیک های احتمالی برای مسائل کنترل تجزیه و تحلیل و سنتز ثابت شده است که در الگوریتم های محاسباتی موثر است. برای یک حساب دقیق در مورد این موضوع، ما به [۲۳] و[۲۶] مراجعه می کنیم.

در این بخش یک مسئله ی تجزیه و تحلیل استحکام را معرفی می کنیم که عدم قطعیت های مختلف را در آن می توان نشان داد. یک سیستم حاوی اجزای نامشخص داده شده، هدف تجزیه و تحلیل استحکام این است که بفهمد کنترل خاص خواص را برای همه ی عدم قطعیت ها می دارد یا خیر. این مسئله ی کلی می تواند به صورت زیر فرموله شود.

ما ابتدا فرض می کنیم که عدم اطمینان در سیستم توسط یک ماتریس حقیقی / موهومی Δ نشان داده شده است نشان داده شده و Δ متعلق به یک مجموعه محدود $\mathcal{B}$. از سوی دیگر ویژگی سیستم با تابع عملکرد $J:\mathcal{B}\to\mathbb{R}$ اندازه گیری می شود. تابع J فرض می شود تابعی قابل اندازه گیری است. یک مسئله ی تجزیه و تحلیل استحکام کلی این است که بررسی کنید که آیا یک سطح عملکرد خاص γ برای همه عدم قطعیت های ممکن $\Delta\in\mathcal{B}$ تضمین شده است یا خیر. به عبارت دیگر ما علاقه مند هستیم که پیدا کنیم که آیا

$$J(\Delta)\leqslant\gamma \text{ برای هر } \Delta\in\mathcal{B} \quad (۱)$$

اگر عدم اطمینان Δ از نوع عمومی است، از نظر پیچیدگی محاسباتی موانعی برای حل این مسئله وجود دارد ما یک رویکرد احتمالی را دنبال می کنیم و معنی استحکام را از معنای قطعی تغییر می دهیم همانطور که در ۱ اتفاق افتاده است.

در رویکرد احتمالاتی فرض می کنیم که ماتریس عدم قطعیت $\Delta\in\mathcal{B}$ یک ماتریس تصادفی است؛ اینجا متغیرهای تصادفی با حروف پررنگ نشان داده می شوند. اجازه دهید $Prob_{\Delta}(.)$ اندازه گیری احتمال مربوطه مربوط به Δ باشد.

ما دو معیار عملکرد خاص را با استفاده از $J(\Delta)$ در نظر میگیریم. اولین مورد عملکرد بدترین حالت است که به صورت زیر تعریف میشود:

$$J_{max}:=\sup_{\Delta\in\mathcal{B}}J(\Delta) \quad (۲)$$

دیگری عملکرد متوسط است

$$J_{ave}:=E_{\Delta}(J(\Delta))$$

به طوری که $E_\Delta(J(\Delta))$ مقدار مورد انتظار تابع عملکرد را با توجه به مجموعه عدم قطعیت $\mathcal{B}$ نشان می دهد.

در این تنظیمات مشکل تصمیم گیری به یک وضعیت اشاره دارد که پاسخ به یک نمونه بله یا خیر است. اینجا، ما دو مسئله ی تصمیم گیری نامشخص را برطرف می کنیم: برای یک سطح عملکرد $\gamma > 0$ چک کنید که آیا $J_{max} \leq \gamma$ و $J_{ave} \leq \gamma$. ما بعد نشان می دهیم که این مسائل را می توان در معنای احتمالی به طور موثر حل کرد. یک تابع عملکردی که معمولاً در تجزیه و تحلیل استحکام استفاده می شود H^∞ که نورم یک سیستم حلقه بسته است. در این مورد، اجازه دهید تابع $J(\Delta)$ نورم سیستم باشد که از اختلال تا خروجی کنترل شده است. بسته به معیار علاقه، ما ممکن است کار با عملکرد بدترین حالت یا عملکرد متوسط را انتخاب کنیم.

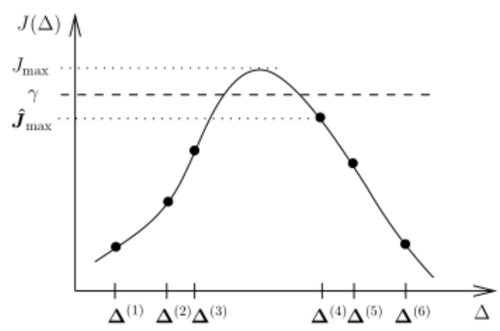
۳ الگوریتم های تصادفی مونت کارلو

اکنون الگوریتم های تصادفی مونت کارلو را معرفی می کنیم. بیشترین نتایج احتمالی حاصل از سیستم ها و کنترل بر اساس این نوع الگوریتم ها هستند. بعدا خواهیم دید که این کلاس متفاوت از آنکه در مسائل اجماع است ظاهر می شود. تعریف به شرح زیر است [۱۷].

تعریف ۱: الگوریتم تصادفی مونت کارلو یک الگوریتم تصادفی است که ممکن است یک نتیجه ای که نادرست است (به معنای قطعی) تولید کند اما احتمال چنین نتیجه ی نادرستی محدود است.

به طور کلی، برای MCRA نتایج و دفعات اجرا از یک اجرا به دیگری متفاوت خواهد بود زیرا الگوریتم بر اساس نمونه گیری تصادفی است. به عنوان یک نتیجه، عملکرد محاسباتی چنین الگوریتم هایی معمولاً توسط زمان اجرای مورد انتظار اندازه گیری می شود. یک MCRA موثر است اگر زمان اجرا از اوردر چند جمله ای در اندازه ی مسئله باشد.

برای مسائل تصمیم گیری می توان MCRA را بر اساس نحوه ارزیابی خطای خروجی ها به دو کلاس تقسیم کرد. MCRA برای یک مسئله ی تصمیم گیری گفته میشود خطای یک طرفه دارد اگر همیشه یک راه حل صحیح در یکی از موارد ممکن را فراهم کند، اما ممکن است یک راه حل اشتباه را برای یکی دیگر ایجاد کند [۱۷].



شکل ۱: MCRA یک طرفه: بدترین عملکرد زمانی که $\hat{J}_{max} < \gamma < J_{max}$

MCRA برای عملکرد بدترین حالت متعلق به این کلاس است. این الگوریتم شامل محاسبه حداکثر تجربی است که با عبارت زیر توصیف میشود:

$$\hat{J}_{max} := \max_{i=1,2,...,N} J(\Delta^{(i)})$$

جایی که $\Delta(i) \in \mathcal{B}, i = 1, 2, ..., N$ نمونه های مستقل و یکسان توزیع شده (شناسه) از عدم قطعیت تصادفی هستند. ماتریس Δ با توجه به اندازه گیری احتمال $Prob_{\Delta}$ تولید می شود. توجه داشته باشید که حداکثر تجربی $\hat{J}_{max}$ خود یک متغیر تصادفی است که ارزش آن بستگی به نمونه های تصادفی انتخاب شده برای محاسبات دارد. شکل. ۱ یک طرح از $J(\Delta)$ را نشان می دهد و مفهوم این مسئله را بیان میکند.

این الگوریتم یک MCRA یک طرفه در زیر است برای یک سطح عملکرد داده شده $\gamma > 0$ اگر $J_{max} \leq \gamma$ سپس به وضوح احتمال این که الگوریتم خروجی پاسخ صحیح بدهد ۱ است. از این رو، این الگوریتم همیشه یک راه حل صحیح برای این مثال فراهم می کند. از طرف دیگر اگر $\gamma < J_{max}$ سپس احتمال اخذ $\hat{J}_{max} \leq \gamma$ ناصفر است. این بدان معنی است که برای این مثال الگوریتم می تواند یک نتیجه اشتباه با احتمال غیر صفر بدهد.

اکنون با توجه به اینکه حداکثر تجربی $\hat{J}_{max}$ همیشه کوچکتر است از J_{max} ما باید یک سوال طبیعی مطرح کنیم چگونه $\hat{J}_{max}$ به خوبی تخمین حداکثر واقعی می کند؟ تحت یک اندازه نمونه به اندازه کافی بزرگ N می توان بیانیه احتمالی ساخت؛ به عنوان مثال،[۲۲]، [۲۳] را ببینید.

کلاس دیگری از MCRA برای مسائل تصمیم گیری این است که از الگوریتم های خطای دو طرفه استفاده شود. چنین الگوریتم هایی ممکن است یک راه حل اشتباه برای هر دو مورد تولید کنند زمانی که پاسخ بله و خیر است [۱۷]. مسئله ی میانگین نمونه ای از یک MCRA دو طرفه است؛ برای اطلاعات بیشتر، ما به [۲۴] مراجعه میکنیم.

۴ الگوریتم های تصادفی لاس وگاس

الگوریتم های لاس وگاس و خواص پایه ی آن را معرفی می کنیم. این نوع زیاد در زمینه کنترل استفاده نشده است اما برای مسائل اجماع مهم است.

۱.۴ مقدمات

تعریف رسمی در [۱۷] به شرح زیر داده شده است:

تعریف ۲: الگوریتم های تصادفی لاس وگاس الگوریتم های تصادفی هستند که همیشه پاسخ صحیح می دهند . تنها تفاوت از یک اجرا به دیگری زمان اجرا است.

به دلایل روشنی، به همچین الگوریتم هایی اشتباه صفر وجهی مونتی کارلو نیز میگویند. به دلیل تصادفی بودن، زمان اجرای الگوریتم نیز رندوم است (مشابه MCRA) و ممکن است در هر مرتبه اجرا، زمانی متفاوت داشته باشد. بنابراین، امید ریاضی زمان اجرا مورد توجه است. اشاره کنیم که امید

ریاضی با احتساب نمونه های تصادفی ساخته‌شده در هر اجرا است و نه ورودی الگوریتم. بعلاوه، اگر امیدریاضی زمان اجرا، از اردر چندجمله ای باشد، الگوریتم کارا خوانده می‌شود.

یک مثال معروف، الگوریتم کوئیک سورت تصادفی RQS است [۱۴]، [۱۷] که در ادامه توضیح داده می‌شود.

مثال ۱ فرض کنید مجموعه

$$\mathcal{S}_1 = x_1, ..., x_N$$

از N عدد حقیقی داده شده‌باشد. مسئله مرتب کردن اعداد را در ترتیب افزایشی در نظر بگیرید. الگوریتم RQS الگوریتم تصادفی ای است که این مسئله را با یک راه حل که از نظر محاسباتی کارا است حل می‌کند. کلیات الگوریتم بدین صورت است

۱ بصورت تصادفی عدد دلخواه $x^{(1)}$ را از مجموعه $\mathcal{S}_1$ انتخاب می‌کنیم.

۲ $x^{(1)}$ و بقیه اعضای $\mathcal{S}_1$ را مقایسه کنید. مجموعه $\mathcal{S}^{(2)}$ مجموعه اعداد کوچکتر از $x^{(1)}$ ، و مجموعه $\mathcal{S}^{(3)}$ مجموعه اعداد بزرگتر از $x^{(1)}$ در نظر بگیرید.

۳ دو مرحله بالا را به صورت بازگشتی روی مجموعه‌های $\mathcal{S}^{(2)}$ و $\mathcal{S}^{(3)}$ اعمال کنید. نسخه مرتب شده $\mathcal{S}^{(2)}$، $x^{(1)}$، و سپس نسخه مرتب شده $\mathcal{S}^{(3)}$ را خروجی بدهید.

منطق تصادفی‌سازی در ۱ به شرح زیر است: اگر مجموعه اصلی $\mathcal{S}_1$ به دو مجموعه با کاردینالیتی‌های یکسان تقسیم شود، الگوریتم در کارآمدترین حالت ممکن خواهد بود. با این حال، این به میانه $\mathcal{S}_1$ به عنوان $x^{(1)}$ نیاز دارد که یافتن آن پرهزینه است. انتخاب تصادفی $x^{(1)}$ یک استراتژی ساده است، اما به طور متوسط تخمین خوبی از میانه ارائه می دهد.

در یک تحلیل رسمی، زمان اجرا با تعداد مقایسه ها اندازه گیری می شود. نتیجه این است که زمان اجرا مورد انتظار از $O(N\log N)$ ست. در واقع، زمان اجرا با احتمال زیاد، یعنی حداقل $11/N$ از این اردر است. [۱۵] RQS کارآمدتر از، برای مثال، یک رویکرد بروت_فورس قطعی است که دارای پیچیدگی $O(N^2)$ است.

برایRQS، بدترین حالت زمانی است که عدد به‌طور تصادفی انتخاب شده در 1 همیشه کوچک‌ترین یا بزرگ‌ترین عدد در مجموعه باشد. سپس، زمان اجرا به اردر $O(N^2)$ می رسد. با این حال، RQS به عنوان یکی از مفیدترین الگوریتم‌های مرتب‌سازی همه‌کاره شناخته می‌شود. [۱۴] ▽

همانطور که در بالا ذکر شد، الگوریتم های لاس وگاس الگوریتم های تصادفی هستند که همیشه نتایج صحیحی را تولید می کنند. واضح است که این ویژگی تنها برای تعداد محدودی از الگوریتم های نوع مونتی کارلو قابل انتظار است. از این رو، کاربرد آنها به طور طبیعی محدود است. یکی از این الگوریتم ها برای مسئله سیستم های سوئیچ شده در [۱۱] توسعه داده شده است.

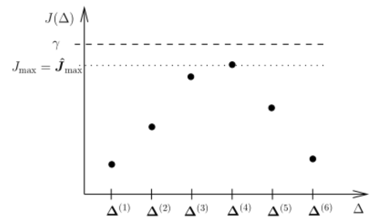
۲.۴ LVRAها برای مسائل انتخابی غیر قطعی

ما اکنون بحثی موازی با بحث سیستم های نامشخص در بخش های ۲ و ۳ برای الگوریتم های لاس وگاس ارائه می کنیم.

ابتدا، به عنوان مجموعه عدم قطعیت، یک زیرمجموعه محدود $\tilde{\mathcal{B}}$ از $\mathcal{B}$ با N عنصر به صورت

$$\tilde{\mathcal{B}} = \left\{\tilde{\Delta}_1, \tilde{\Delta}_2, ..., \tilde{\Delta}_N\right\} \subset \mathcal{B}$$

را در نظر می‌گیریم.



شکل ۲: LVRA: بدترین عملکرد وقتی $J_{max} = \widehat{J} < \gamma$

با فرض اینکه ماتریس های نامشخص در $\tilde{\mathcal{B}}$ متغیرهای تصادفی هستند، یک اندازه گیری احتمال گسسته را در نظر می گیریم؛ دقیقا همانند بخش II، این اندازه گیری با Prob_{Δ} نشان داده می شود. به طور مشابه، فرض کنید تابع عملکرد $J : \tilde{\mathcal{B}}\mathbb{R}$ باشد. مسئله تحلیل استحکام کلی این است که ببینیم آیا برای یک سطح عملکرد معین $\gamma > 0$، $J(\Delta) \leq \gamma$ برای همه $\Delta \in \tilde{\mathcal{B}}$ است یا خیر. بدترین حالت مربوطه بصورت

$$J_{max} := \max_{\Delta \in \tilde{\mathcal{B}}} J(\Delta) \qquad (۳)$$

است، و حالت میانگین بصورت

$$J_{ave} := E_{\Delta}\left[J(\Delta)\right] = \sum_{i=1}^{N} J(\tilde{\Delta}_i) Prob_{\Delta}(\tilde{\Delta}_i)$$

است. ما اکنون مسئله انتخابی غیرقطعی را برای عملکرد بدترین حالت در نظر می گیریم. با فرض یک اسکالر $\gamma > 0$، بررسی کنید که آیا $J_max \leq \gamma$ برقرار است یا خیر. LVRA برای این مورد به شرح زیر است: فرض کنید $\tilde{\mathcal{B}}_0 = \tilde{\mathcal{B}}$ و فرض کنید $k = 1$. همچنین، فرض کنید $\bar{J}^{(0)} = -\inf$. در مرحله k،th به طور تصادفی یک نمونه $\Delta^{(k)}$ از $\tilde{\mathcal{B}}_{k-1}$ انتخاب کنید. سپس، قرار دهید $\bar{J}^{(k)} = max\left\{J(\Delta^{(k)}), \bar{J}^{(k-1)}\right\}$ و $\tilde{\mathcal{B}}_k = \tilde{\mathcal{B}}_{k-1}\ \Delta^{(k)}$، و به مرحله بعد بروید. پس از مرحله Nام، حداکثر عملکرد نسبت به نمونه ها توسط $\hat{J}_{max} = \bar{J}^{(N)}$ محاسبه می‌شود.

این الگوریتم همیشه پاسخ صحیح را خروجی می دهد و از این رو از نوع لاس وگاس است. برای مثال ($J_{max} \leq \gamma$) یا مثال ($J_{max} > \gamma$)، مقدار $\hat{J}_{max}$ که الگوریتم تولید می‌کند با مقدار واقعی J_{max} منطبق می‌شود. این موضوع در شکل ۲ نشان داده شده است.

وقتی N بزرگ است، پیچیدگی محاسباتی را می توان با اصلاح الگوریتمی که توضیح دادیم کاهش داد. الگوریتم به دست آمده از نوع مونتی کارلو است. این را می توان با توقف در مرحله kام با $k < N$

و با محاسبه حداکثر عملکرد روی نمونه k انجام داد. در واقع الگوریتم حاصل به یک MCRA یک طرفه تبدیل می شود. ما خاطرنشان می کنیم که این رویکرد ارتباط نزدیکی با بهینه سازی ترتیبی دارد. به عنوان مثال، [۸] را ببینید. هدف در آنجا یافتن حداکثر مقدار عملکرد نیست، بلکه مقداری است که حداقل در m تا بزرگترین باشد.

عملکرد متوسط مورد نیز می تواند به طور مشابهی بررسی شود.

۵ الگوریتم های تصادفی لاس وگاس برای اجماع متوسط توزیع شده

در این بخش، چندین مسئله در اجماع میانگین توزیع شده ارائه می‌کنیم که در آن کاربرد الگوریتم‌های نوع لاس وگاس می‌تواند مؤثر و گاهی اوقات در واقع حیاتی باشد.

۱.۵ حالت مسئله عمومی

شبکه‌ای از N گره را در نظر بگیرید که گراف $(\mathcal{V}، \mathcal{E})$ می‌نامیم، که در آن $\mathcal{V} := 1, 2, ..., N$ مجموعه گره ها و $\mathcal{E}$ مجموعه یال ها است. فرض می‌شود که گراف بدون جهت و همبند است (به‌عنوان مثال، [۱۷] را برای مقدمه‌ای بر نظریه گراف‌های تصادفی ببینید). این بدان معنی است که یال ها با جهت علامت گذاری نشده اند و برای هر $ij \in \mathcal{V}$، مسیری وجود دارد که گره های i و j را به هم متصل می کند. در زمان k، هر گره i دارای یک مقدار اسکالر $x_i(k)$ است که مقدار اولیه آن $x_i(0)$ است.

هدف ارائه الگوریتمی است که (I) گره ها مقادیر $x_i(k)$ خود را با استفاده از اطلاعات ارسالی از همسایگان خود به روز کنند و (II) مقادیر گره ها در نهایت به میانگین مقادیر اولیه همگرا شوند. برای این منظور، برای یک الگوریتم اجماع، دو عنصر وجود دارد که باید تعیین شود: قوانینی برای به‌روزرسانی مقادیر گره ها و همسایه هایی که هر گره باید با آنها ارتباط برقرار کند.

این مسئله نسخه خاصی از اجماع توزیع شده است. به طور کلی، مسائل اجماع دنبال این نیستند که مقادیر گره ها به کدام عدد باید همگرا شوند، بلکه فقط باید مقدار آنها برای همه گره ها یکسان باشد. در ادامه به سه حالت از این مسئله می پردازیم. تفاوت در محدوده مقادیر گره‌ها است: اعداد واقعی، اعداد صحیح (کوانتیزه شده) و اعداد باینری.

ابتدا برخی نماد را که در سراسر این بخش مورد استفاده قرار خواهند گرفت، معرفی می کنیم. فرض کنید بردار Nـ بعدی متشکل از همه مقادیر گره در زمان k $x(k) = [x_1(k) \dots x_N(k)]^T$ باشد. الگوی ارتباطی برای گره‌ها در زمان k توسط مجموعه یال $\tilde{\mathcal{E}}(k) \subset \mathcal{E}$ مشخص می‌شود، یعنی اگر $\{i, j\} \in \tilde{\mathcal{E}}$ ، سپس $x_i(k)$ با استفاده از $x_j(k)$ به روز می شود و بالعکس. در این حالت، گره‌های i و j در این زمان همسایگان یکدیگر هستند. به طور کلی، همسایگان یک گره ممکن است در طول زمان تغییر کنند.

۲.۵ حالت مقدار حقیقی

ما ابتدا حالتی را ارائه می کنیم که مقادیر گره‌ها اعداد حقیقی اند؛ که در [۲۸] مطالعه شده است. می گوییم اجماع متوسط (به معنای قطعی) در صورتی حاصل می شود که شرط زیر برآورده شود:

$$\lim_{k\to\inf} x_i(k) = \frac{1}{N}\sum_{j=1}^{N} x_j(0) \text{ all for } i = 1, ..., N. \tag{۴}$$

قانون به‌روزرسانی برای گره i یک فرم خطی به شکل

$$x_i(k+1) = W_{ii}(k)x_i(k) + \sum_{j\in\mathcal{N}_i(k)} W_{ij}(k)x_j(k), \tag{۵}$$

می‌گیرد؛ که $\mathcal{N}_i(k) := \{j : \{ij\} \in \tilde{\mathcal{E}}(k)\}$ مجموعه ی همسایگان برای گره i در زمان k، و $W_{ij}(k)$ وزن هستند. وزن ها در زمان تغییر می کنند و بر اساس

$$W_{ij}(k) = \begin{cases} \frac{1}{1+max\{d_i(k),d_j(k)\}} & \text{if}\{i,j\} \in \tilde{\mathcal{E}}(k), \\ 1 - \sum_{l\in\mathcal{N}_i(k)} W_{il}(k) & \text{if} i = j \\ 0 & \text{otherwise،} \end{cases} \tag{۶}$$

مشخص می‌شوند؛ که در آن $d_i(k)$ نشان دهنده کاردینالیتی $\mathcal{N}_i(k)$، یعنی تعداد همسایگان برای گره i است. توجه داشته باشید که مجموع وزن‌ها در هر گره یک است. این باعث می شود ماتریس W یک ماتریس تصادفی باشد. در [۲۸]، اشاره شده است که وزن‌های $Wij(k)$ در (۶) از الگوریتم متروپلیس استفاده شده در زنجیره مارکوف مونتی کارلو اتخاذ شده‌اند.

قانون به روز رسانی در (۵) را می توان به صورت توزیع شده و علّی اجرا کرد. این به این دلیل است که گره ها فقط به دانستن مقادیر رسیده از همسایگان خود در زمان به روز رسانی نیاز دارند. در مورد الگوی ارتباطی مشخص شده توسط $\tilde{\mathcal{E}}(k)$، $k \in \mathbb{Z}_+$، فرض به شرح زیر است: گراف $(\mathcal{V}$ ، $\bigcup_{s\geq k}\tilde{\mathcal{E}}(s))$ یک گراف همبند برای هر k است. به عبارتی، یعنی مجموع مجموعه‌های یال هایی که در طول زمان بی‌نهایت مرتبه رخ می‌دهند، آن را به یک گراف همبند تبدیل می‌کند.

آنچه در زیر آمده است، نتیجه اصلی [۲۸] است.

قضیه ۱: تحت قانون به‌روزرسانی در (۵) و الگوی ارتباطی خاصی که شرایط اینکه گراف $(\mathcal{V}$، $\bigcup_{s\geq k}\tilde{\mathcal{E}}(s))$ یک گراف همبند برای همه k است را ارضا می‌کند، اجماع میانگین توزیع شده با تعریف (۴) برای هر شرط اولیه $x(0) \in \mathbb{R}^N$ بدست می‌آید.

این قضیه شرطی را در $\tilde{\mathcal{E}}(k)$ ارائه می‌کند که باید در زمان پیاده سازی برای اجماع متوسط به معنای قطعی مشخص شود. شرایط مشابهی در طرح‌های دیگر به‌کار می‌رود، به‌عنوان مثال، [۵]، [۱۲] و [۱۶].

یک راه ساده برای پیاده سازی یک الگوی ارتباطی با ویژگی مورد نظر، استفاده از تصادفی سازی است: هر گره i با یک همسایه به طور تصادفی و مستقل انتخاب شده j_k ارتباط برقرار می کند که

$\{i, j_k\} \in \mathcal{E}$ را در زمان k برآورده می کند. به طور خاص، احتمالی مثبت به هر یال $\{i, j\}$ در $\mathcal{E}$ اختصاص می‌دهیم. از آنجایی که $(\mathcal{V}$، $\mathcal{E})$ یک گراف همبند است، شرط در الگوی ارتباطی به طور احتمالی برقرار است. یعنی به ازای هر k، احتمال اینکه نمودار $(\mathcal{V}$، $\bigcup_{s\geq k} \tilde{\mathcal{E}}(s))$ همبند شود یک است. از این رو، الگوریتم حاصل به اجماع متوسط در (۴) با احتمال یک برای هر $x(0)$ دست می یابد.

یک طرح اجماع میانگین تصادفی دیگر نیز در [۱۰] پیشنهاد شده است، که در آن یال های گراف به طور تصادفی و مستقل از یکدیگر انتخاب می‌شوند. این مقاله یک تحلیل احتمالی از همگرایی ارائه می دهد. این طرح بر اساس یک پروتکل ارتباطی داده‌های نمونه‌برداری شده است و وزن‌هایی را به کار می‌گیرد که با موارد (۶) متفاوت است. سایر آثاری که از الگوهای ارتباطی تصادفی سازی شده استفاده می‌کنند عبارتند از: [۷]، [۲۱] و [۲۷].

۳.۵ حالت مقدار کوانتیزه شده

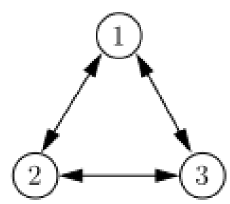
این طرح در [۱۳] پیشنهاد شده است. اینجا منظور از کوانتیزه شدن، این است که مقادیر گره‌ها اعداد صحیح هستند. این مسئله‌ی اجماع، نیازمند چاره‌ای متفاوت از حالت مقدار حقیقی است. برای شروع، میانگین از مقادیر اولیه ممکن است یک عدد صحیح نباشند. بنابراین، مقدار مورد نظر یک تقریب عدد صحیح از میانگین واقعی است و لزوما منحصر به فرد نیست. علاوه بر این، می توان در زمان محدود به اجماع رسید زیرا گره ها در اعداد صحیح در هر لحظه به‌روز می شوند.

به طور خاص تر، اگر شرایط زیر برقرار باشند، گفته می‌شود الگوریتم به متوسط اجماع کوانتیزه‌شده می‌رسد:

۱. همیشه مقادیر اعداد صحیح هستند: $x_i(k) \in \mathbb{Z},\ \forall i, k$.

۲. مجموع مقادیر گره‌ها ثابت می‌ماند: $\sum_{i=1}^{N} x_i(k) = \sum_{i=1}^{N} x_i(0)$ برای هر k.

۳. همه‌ی مقادیر به میانگین کوانتیزه شده همگرا می‌شوند: k^* وجود دارد که $x_i(k) \in \{\overline{x}, \overline{x}+1\}$ برای هر $k > k^*$ و i، که $\overline{x} = \left\lfloor \sum_{i=1}^{N} x_i(0)/N \right\rfloor$.



شکل ۳: گرافی با ۳ گره

در [۱۳] یک کلاس از الگوریتم‌های تصادفی به نام الگوریتم‌های گاسیپ کوانتیزه‌شده پیشنهاد شده است؛ [۶] را نیز ببینید.

موارد زیر، ملزومات چنین الگوریتم‌هایی هستند: در زمان k یک یال $\{i, j\} \in \varepsilon$ به روش i.i.d. به طور تصادفی انتخاب می‌شود. فرض کنید $D_{ij}(k) = |x_i(k) - x_j(k)|$.

۱. اگر $D_{ij}(k) = 0$، آنگاه مقدار گره‌های i و j در زمان $k+1$ یکسان باقی می‌مانند.

۲. اگر $D_{ij}(k) = 1$، آنگاه مقادیر مطابق زیر جابجا می‌شوند:

$$x_i(k+1) = x_(j)(k) \text{ و } x_j(k+1) = x_i(k). \tag{۷}$$

۳. در غیر این صورت، بروزرسانی مقادیر به صورت زیر است:

$$x_i(k+1) + x_j(k+1) = x_i(k) + x_j(k),$$
$$D_{ij}(k+1) < D_{ij}(k).$$

توجه کنید که الگوریتم‌ها در این کلاس با توجه به تعریف، رندوم هستند. یکی از راه های تعیین توزیع برای انتخاب گره‌ها، تخصیص احتمالات مثبت به همه‌ی گره‌ها در گراف است. ما خوانندگان را به [۱۳] برای چند مورد از الگوریتم‌های صریح گاسیپ کوانتیزه شده که ملزومات فوق را ارضا میکنند، ارجاع می‌هیم. همچنین تاکید می‌کنیم که این الگوریتم‌ها با الگوریتم‌های بخش V-B متفاوت هستند.

قضیه ۲: برای هر شرابط اولیه‌ی $x(0) \in \mathbb{Z}^N$، یک الگوریتم گاسیپ کوانتیزه شده با احتمال یک، به یک اجماع متوسط کوانتیزه شده در تعدادی گام متناهی می‌رسد.

یکی از جنبه های جالب این الگوریتم، تصادفی سازی ضروری است. این می‌تواند با یک مثال که در [۱۳] آمده، تصویرسازی شود. یک گراف با سه گره مانند شکل ۳ در نظر بگیرید. در ابتدا، گره i مقادیر $i = 1, 2, 3$ را می‌گیرد و بنابراین، مقدار میانگین برابر 2 می‌شود. فرض کنید ما یک طرح قطعی دوره‌ای برای انتخاب یال با دوره‌ی تناوب 3 به صورت $\{1,2\}, \{1,3\}, \{3,2\}, \{1,2\}, \ldots$ به کار بگیریم. طبق این طرح، واضح است که فقط جابجایی در (۷) همیشه اتفاق می‌افتد. بنابراین، مجموعه‌ی مقادیر گره‌ها 1، 2 و 3 باقی می‌ماند و هرگز به مقادیر اجماع نمی‌رسد. در مقابل، با انتخاب تصادفی گره‌ها، میانگین اجماع در چند مرحله امکان پذیر است.

در [۱۳] تحلیل‌های احتمالی بیشتری درباره‌ی زمان اجرای مورد انتظار برای رسیدن به اجماع کوانتیزه شده آورده شده‌اند. به طور خاص، اگر گراف همبند کامل یا خطی باشد، آنگاه زمان اجرای مورد انتظار، چند جمله‌ای برجسب تعداد گره‌ها است. بنابراین این الگوریتم یک مثال از یک الگوریتم لاس وگاس است. این، با حالت مقدار حقیقی در بخش V-B در تقابل است.

۴.۵ حالت مقدار دودویی

سومین مسئله‌ی اجماع زمانیست که عوامل، مقادیر باینری $x_i(k) \in \{0,1\}$ برای هر i و k را اتخاذ می‌کنند. به طور خاص، مسئله‌ای که در اینجا به آن می‌پردازیم، با عنوان *Byzantine agreement* شناخته می‌شود [۱۷]. ما نشان خواهیم داد که در این حالت، نتایج قوی‌تری به نفع طرح‌های تصادفی قابل دسترسی هستند.

یک ویژگی متمایزکننده‌ی این مسئله این است که بعضی از عوامل N معیوب هستند. این عوامل، ممکن است عوامل نامعیوب را فریب دهند و حتی ممکن است که مخفیانه باهم ارتباط داشته باشند.

ما فرض می‌کنیم به تعداد N_f از این عوامل داریم که ثابت اند اما هویت آنها برای عوامل نامعیوب ناشناخته است. با توجه به الگوی ارتباطی، گراف همبند کامل است. هر عامل در زمان k پیامی را به تمام عوامل دیگر می‌فرستد؛ عوامل معیوب، ممکن است پیام‌های متفاوتی به سایرین بفرستند. درنتیجه، تصادفی سازی در الگوی ارتباطی رخ نمی‌هد. در عوض، یک پرتاب سکه توسط یک طرف قابل اعتماد که نتیجه را برای تمام عوامل در هر k می‌فرستد، انجام می‌شود.

هدف، دستیابی به نوعی اجماع متوسط حتی در بدترین حالت است. اگر هر عامل i مقدار تصمیم $y_i \in \{0,1\}$ را به صورت زیر تعیین کند، می‌گوییم اجماع دودویی حاصل شده است:

۱. همه‌ی عوامل نامعیوب به یک مقدار تصمیم برسند؛

۲. اگر تمام عوامل نامعیوب مقدار اولیه‌ی یکسان $x_i(0)$ را داشته باشند، آنگاه $y_i = x_i(0)$.

حالت ساده‌ای را در نظر می‌گیریم که در آن N مضربی از 8 است و $N_f < N/8$. همچنین قرار می‌هیم:

$$L := 5N/8 + 1,\ H := 3N/4 + 1,\ G := 7N/8.$$

الگوریتمی که برای هر عامل i ارائه می‌دهیم، براساس [۱۹] است؛ اساسا، عامل مورد نظر مقدار خودش را می‌فرستد و با توجه به اکثریت مقادیری که دریافت کرده، تصمیم می‌گیرد.

۱) ارسال مقدار $x_i(k)$ به عوامل دیگر و دریافت $x_j(k),\ j \neq i$ از آنها در زمان k.
۲) تنظیم مقدار اکثریت $m_i(k) \in \{0,1\}$ به چیزی که اکثر عوامل آن را به عنوان مقدار خود فرستاده اند. سپس، قرار دادن مقدار شمارش $t_i(k)$ برابر با تعداد عواملی که مقادیرشان برابر $m_i(k)$ است.
۳) حال، با توجه به نتیجه‌ی پرتاب سکه، مقدار آستانه $\bar{t}(k)$ را برابر L قرار دهید اگر سکه شیر بیاید؛ و درغیراینصورت، مقدار آن را H بگذارید.
۴) تنظیم مقدار $x_i(k)$ به مقدار اکثریت $m_i(k)$ اگر $t_i(k) \geq \bar{t}(k)$ و در غیر اینصورت به 0.
۵) اگر $t_i(k) \geq G$، آنگاه مقدار تصمیم را برابر y_i را برابر $m_i(k)$ قرار دهید.

دو راه حل ساده وجود دارد. (الف) زمانیکه همه‌ی عوامل نامعیوب در یک نقطه مقدار مشترک دارند، همه‌ی آنها در تعدادی گام ثابت، این مقدار را به عنوان مقدار تصمیم خود اتخاذ می‌کنند. (ب) وقتی دو عامل نامعیوب دو مقدار اکثریت متفاوت دارند، آنگاه همه‌ی مقادیر $x_i(k)$ در این گام صفر می‌شوند زیرا هیچ یک از شمارش ها از آستانه عبور نکرده اند.

حالت جالب زمانیست که همه‌ی عوامل نامعیوب مقدار اکثریت یکسانی دارند. آنگاه، عوامل معیوب این شانس را دارند که آنها را سردرگم کنند. این کار می‌تواند با ساخت چند عامل نامعیوب

که شمارش برخی از آنها از آستانه عبور می‌کند و شمارش سایرین، از آستانه کمتر است انجام شود. با این حال، طبق این طرح، شانس محدود است. چون $HL \geq N_f$، عوامل معیوب می‌توانند هر عامل را فقط برای یکی از آستانه های L یا H فریب دهند. علاوه بر این، آستانه به طور رندوم بین L و H به علت پرتاب سکه تغییر می‌کند. در نتیجه، احتمال فریب $1/2$ است. از سوی دیگر، زمانی که آنها نمی توانند فریب دهند، همه عوامل غیر معیوب ارزش یکسانی دارند، که منجر به اجماع می شود.

نتیجه اصلی در مورد Byzantine agreement problem در زیر آورده شده است [۱۷].

قضیه ۳: برای الگوریتمی که در بالا توصیف شد، اجماع دودویی با احتمال یک برای هر مقدار اولیه‌ی $x(0) \in \{0,1\}^N$ به دست می‌آید. همچنین، تعداد گام‌های مورد نیاز، ثابت است.

این قضیه مطرح می‌کند که الگوریتم اجماع، یک نوع موثر از الگوریتم لاس وگاس است. در مقابل، واضح است که هر الگوریتم قطعی نیازمند $N_f + 1$ گام است [۱۷]. تاکید شده است که نتایج قوی‌تری نیز در حوزه‌ی محاسبات توزیع شده بدست آمده است. یکی از آنها نتیجه‌ی ناممکن برای یک ورژن ناهماهنگ از مسئله‌ی اجماع دودویی است [۹]. این نتیجه بیان می‌کند که اگر هیچ ساعت هماهنگی متعلق به عوامل وجود نداشته باشد و اگر هیچ فرضی از نمونه برداری دوره ها از عوامل وجود نداشته باشد، برای هر الگوریتم قطعی، رسیدن به اجماع ناممکن است. طرح های تصادفی مثل پرتاب سکه در این حالت به خوبی نشان داده شده‌اند. برای بررسی این موضوع، که همچنین شامل بسیاری از پیشرفت های اخیر در الگوریتم اصلی است، به [۲] اشاره می‌کنیم.

۶ نتیجه‌گیری

در این مقاله، درباره‌ی چند واریانت از مسئله‌ی اجماع متوسط و نقش الگوریتم‌های احتمالی در این زمینه، بحث کردیم. همچنبن، نشان داده شد که کلاس الگوریتم‌های تصادفی لاس وگاس، که اخیرا در حوزه‌ی سیستم‌ها و کنترل به کار گرفته شده‌اند [۱۱]، [۲۴]، موثر و گاهی حیاتی است.

مراجع

[۱] Technol- the on Issue Special Editors. Guest Baillieul، J. and Antsaklis J. P. .۲۰۰۷ ،(۱)۹۵ IEEE، Proc.Systems. Control Networked of ogy

[۲] Distributed consensus. asynchronous for protocols Randomized Aspnes. J. .۲۰۰۳ ،۱۷۵—۱۶:۱۶۵ Computing،

[۳] Computation: Distributed and ParallelTsitsiklis. N. J. and Bertsekas P. D. NJ، ۱۹۸۹. Cliffs، Englewood Prentice-Hall، Methods. Numerical

[۴] groups of "Coordination on Comments Tsitsiklis. N. J. and Bertsekas P. D. Trans. IEEE rules". neighbor nearest using agents autonomous mobile of .۲۰۰۷ ،۹۶۹—۵۲:۹۶۸ Control، Autom.

[۵] Con- Tsitsiklis. N. J. and Olshevsky، A. Hendrickx، M. J. Blondel، D. V.
Proc. In flocking. and consensus، coordination، multiagent in vergence
Conf.، Control European and Control and Decision on Conf. IEEE ۴۴th
.۲۰۰۵ ،۳۰۰۰—۲۹۹۶ pages

[۶] algo- gossip Randomized Shah. D. and Prabhakar، B. Ghosh، A. Boyd، S.
.۲۰۰۶ ،۲۵۳۰—۵۲:۲۵۰۸ Theory، Information Trans. IEEE rithms.

[۷] Com- Zampieri. S. and Speranzon، A. Focoso، M. Fagnani، F. Carli، R.
Automatica، problem. consensus average the in constraints munication
.۲۰۰۸ ،۶۸۴—۴۴:۶۷۱

[۸] optimal to approach optimization ordinal An Ho. Y.-C. and Deng M.
.۱۹۹۹ ،۳۳۸—۳۵:۳۳۱ Automatica، problems. control

[۹] distributed of Impossibility Paterson. S. M. and Lynch، A. N. Fisher، J. M.
.۱۹۸۵ ،۳۸۲—۳۲:۳۷۴ ACM، J. processor. faulty one with consensus

[۱۰] IEEE networks. random over Agreement Mesbahi. M. and Hatano Y.
.۲۰۰۵ ،۷۲—۵۰:۱۸۶۷ Control، Autom. Trans.

[۱۱] switched of stabilization and sorting Probabilistic Tempo. R. and Ishii H.
.۲۰۰۷ publication، for Submitted systems.

[۱۲] mo- of groups of Coordination Morse. S. A. and Lin، J. Jadbabaie، A.
Autom. Trans. IEEE rules. neighbor nearest using agents autonomous bile
.۲۰۰۳ ،۱۰۰۱—۴۸:۹۸۸ Control،

[۱۳] Automatica، consensus. Quantized Srikant. R. and Başar، T. Kashyap، A.
.۲۰۰۷ ،۱۲۰۳—۴۳:۱۱۹۲

[۱۴] volume edition، ۲nd Programming، Computer of Art The Knuth. E. D.
.۱۹۹۸ MA، Reading، Addison-Wesley، Searching. and Sorting :۳

[۱۵] Random- Computing: and Probability Upfal. E. and Mitzenmacher M.
Press، University Cambridge Analysis. Probabilistic and Algorithms ized
.۲۰۰۵

[۱۶] commu- time-dependent with systems multiagent of Moreau. Stability L.
.۱۸۲،۲۰۰۵—۵۰:۱۶۹ Control، Autom. Trans. IEEE links. nication

[۱۷] Cambridge Algorithms. Randomized Raghavan. P. and Motwani R.
.۱۹۹۵ Press، University

[۱۸] of networks in problems Consensus Murray. R. and Olfati-Saber R.
Autom. Trans. IEEE time-delays. and topology switching with agents
.۲۰۰۴ ،۱۵۳۳—۴۹:۱۵۲۰ Control،

[۱۹] M. O. Rabin. Randomized Byzantin generals. In Proc. Annual Symp. on Foundations of Computer Science، pages ۴۰۳–۴۰۹، ۱۹۸۳.

[۲۰] A. V. Savkin. Coordinated collective motion of groups of autonomous robots: Analysis of Vicsek's model. IEEE Trans. Autom. Control، ۴۹:۹۸۱–۹۸۳، ۲۰۰۴.

[۲۱] A. Tahbaz-Salehi and A. Jadbabaie. Necessary and sufficient conditions for consensus over random independent and identically distributed switch- ing graphs. In Proc. ۴۶th IEEE Conf. on Decision and Control، pages ۴۲۰۹–۴۲۱۴، ۲۰۰۷.

[۲۲] R. Tempo، E. W. Bai، and F. Dabbene. Probabilistic robustness analysis: Explicit bounds for the minimum number of samples. Systems and Control Letters، ۲۳۷:۳۰–۲۴۲، ۱۹۹۷.

[۲۳] R. Tempo، G. Calafiore، and F. Dabbene. Randomized Algorithms for Analysis and Control of Uncertain Systems. Springer، London، ۲۰۰۵.

[۲۴] R. Tempo and H. Ishii. Monte Carlo and Las Vegas randomized algo- rithms for systems and control: An introduction. European J. Control، ۱۸۹:۱۳–۲۰۳، ۲۰۰۷.

[۲۵] J. N. Tsitsiklis. Problems in Decentralized Decision Making and Compu- tation. PhD thesis، Dept. of Electrical Engineering and Computer Science، MIT، ۱۹۸۴. http://web.mit.edu/jnt/www/Papers/PhD-۸۴-jnt.pdf.

[۲۶] M. Vidyasagar. Statistical learning theory and randomized algorithms for control. IEEE Control Systems Magazine، ۱۸ (۶):۶۹–۸۵، ۱۹۹۸

[۲۷] C. W. Wu. Synchronization and convergence of linear dynamics in ran- dom directed networks. IEEE Trans. Autom. Control، ۱۲۰۷:۵۱–۱۲۱۰، ۲۰۰۶.

[۲۸] L. Xiao، S. Boyd، and S. Lall. A scheme for robust distributed sensor fusion based on average consensus. In Proc. Conf. on Information Processing in Sensor Networks، pages ۶۳–۷۰، ۲۰۰۵.