باسمه تعالی

دانشگاه صنعتی شریف

دانشکده مهندسی برق

درس سیستم های توزیع شده

پروژه درس

علی محرابیان 96102331

دانش گرمچی 96102289

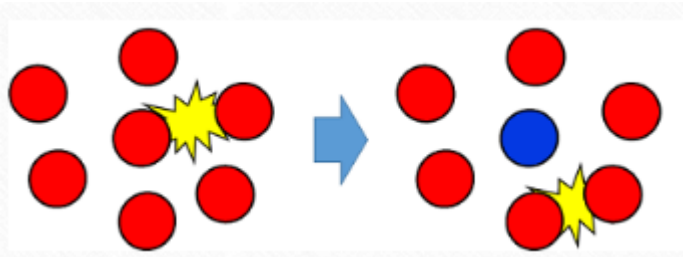
استاد: دکتر صالح

تابستان 1399

مقدمه:

Population protocols یکی از مدل های ریاضی است که برای مدل کردن پروتکل های ارتباطی(به طور خاص Message passing) بیان می شود و به این صورت است که کاربر های ما که در قالب finite state machine هستند،به صورت زوج های دوتایی با یکدیگر ارتباط برقرار کرده و state های خود را طبق قوانین مشخصی، آپدیت می کنند.در این مدل ما از tiny mobile agent ها در واقع استفاده می کنیم.tiny به این معنا که کاربرهای ما توان پردازشی محدودی دارند و mobile به این معناست که این کاربر ها می توانند جابه جا شوند.هدف ما در این مدل،محاسبه یک تابع به صورت گسترده در شبکه به کمک ارتباطات دوتایی کاربر ها با یکدیگر است.

از این مدل برای حل مسائل مختلفی مانند leader election،majority voting،تخمین سایز شبکه و... استفاده می شود که 3 مورد از آن ها را در این پروژه بررسی می کنیم.



توصیف الگوریتم:

برای مدل کردن و توصیف الگوریتم،متغیرهایی که در آن دخیل هستند را بیان می کنیم.

مجموعه متناهی X که بیانگر مجموعه حالت های ورودی و مجموعه متناهی Y که بیانگر حالت های خروجی است.مجموعه متناهی Q هم بیانگر مجموعه state ها و حالت ها هستند.

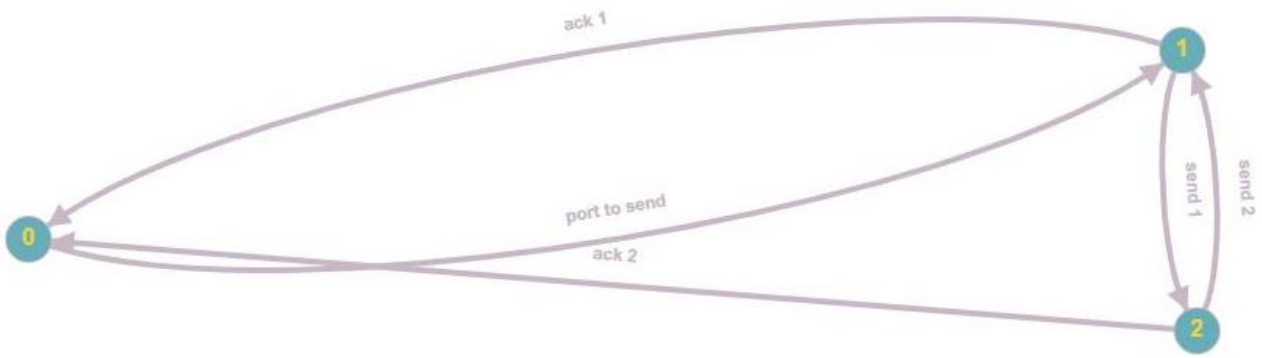
$X \xrightarrow{I} Q \xrightarrow{O} Y$ I:input function O:output function

یکی از بخش های مهم این مدل،transition function است.هر دو کاربری که به صورت زوج و دوتایی با هم ارتباط برقرار کرده،از state های قبلی خود به state جدید می روند.

$(\mathrm{p},\mathrm{q}) \xrightarrow{\delta} (p',q')$ δ: $transition\ function$

و در نهایت از یک fair scheduler برای انتخاب زوج های تصادفی که به صورت غیر قابل پیش بینی انتخاب می شوند،استفاده می کنیم.در این مدل با فرض این که agent های ما دارای حافظه ای به اندازه $\log n$ هستند، می توان termination را داشت ولی بیشتر به دنبال رسیدن به همگرایی و convergence با رسیدن به یک Stable configuration هستیم،به این معنا که agent های ما به state هایی می رسند که خروجی از آن جا به بعد تغییر نمی کند،با این که حتی امکان جابه جایی و تغییر state ها وجود دارد.

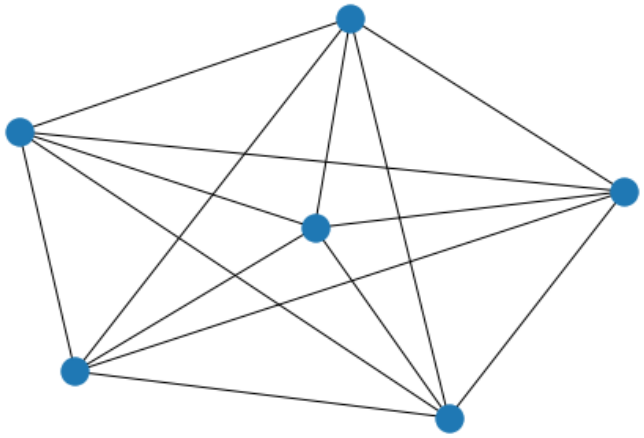
حال در این قسمت به نحوه پیاده سازی و بیان سناریو های مختلف اشاره می کنیم.



در هر مرحله توسط scheduler،دو کاربری که یالی بین آن ها وجود دارد،به صورت تصادفی انتخاب می شوند و برای یکی از آن ها،پیام port to send فرستاده می شود.محتوای این پیام،port همسایه ای است که باید با آن ارتباط برقرار کند.حال این گره که پیام را دریافت کرده،پیام send 1 را که حاوی state و port خودش است،به همسایه خود می فرستد.گره 2 با دریافت پیام send 1،عملیات transition را انجام می دهد.اگر state خود و همسایه را داخل transition rule ها یافت،state ها را آپدیت می کند و در غیر این صورت،state ها بدون تغییر باقی می مانند.حال این گره،پیام send 2 را به همسایه خود فرستاده و state جدید او را به او می گوید.سپس هر دو گره با فرستادن پیام ACK به scheduler،این امکان را به او می دهند تا دوباره،عملیات انتخاب agent های تصادفی را انجام دهد.

سناریو اول:توپولوژی کامل

در این قسمت فرض می کنیم که توپولوژی شبکه به صورت گراف کامل است و در هر مرحله،هر دو گره ای از شبکه،امکان برقراری ارتباط با یکدیگر را دارند.

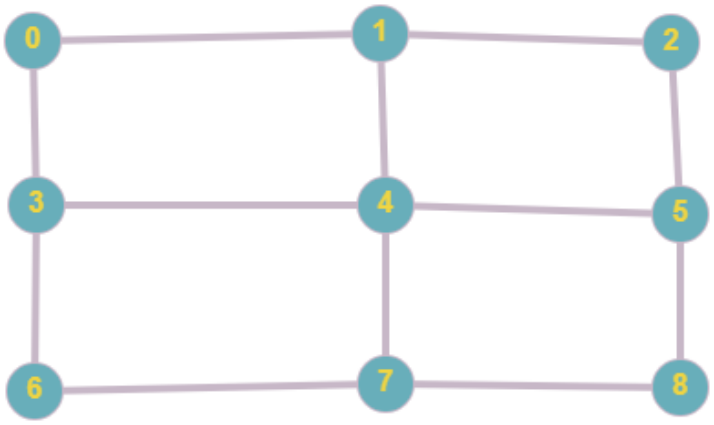


سناریو دوم:توپولوژی grid

در این حالت توپولوژی شبکه را به صورت شبکه مشبک grid در نظر می گیریم.با فرض این که در راستای طول m گره و در راستای عرض p گره داشته باشیم،برای به دست آوردن همسایه های یک گره به این صورت عمل می کنیم که ابتدا مختصات آن را در شبکه به دست می آوریم. شماره گذاری گره ها به ترتیب در راستای طول هستند و فرض کنیم شماره گره n باشد.

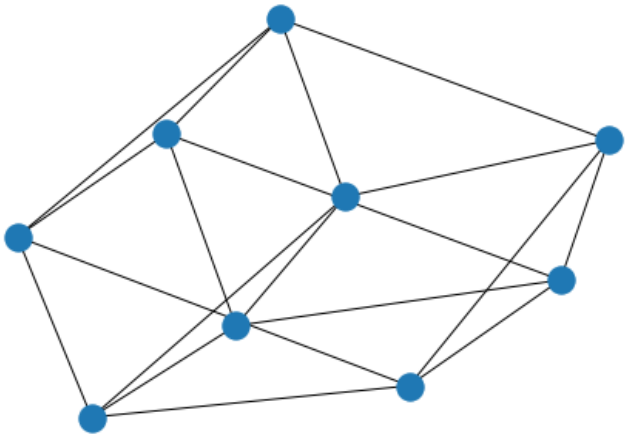
$$n = mq + r \quad \rightarrow \quad (x, y) = (r, m)$$

حال در هر راستا می توانیم یک واحد جابه جا شده تا به گره های همسایه برسیم.پس از انتخاب همسایه به صورت رندوم، با اعمال $n = mq + r$ ،می توانیم به شماره گره همسایه و در نتیجه port متعلق به آن برسیم.



سناریو سوم:توپولوژی مش

این حالت کاملا مشابه با سناریو قبل می باشد،با این تفاوت که راس های ابتدایی و انتهایی در حاشیه ها نظیر به نظیر به همدیگر وصل هستند.پیاده سازی کاملا مشابه با حالت قبل است.



سناریو چهارم:توپولوژی grid همراه با mobility

در این قسمت همان پیاده سازی های قبلی صورت می گیرد،با این تفاوت که گره ها در راستای خود امکان جابه جایی دارند.هر گره در ابتدای شبیه سازی،یک سکه هم شانس می اندازد و جهتی که به صورت عمودی یا افقی تا انتهای شبیه سازی در آن راستا جابه جا می شود را انتخاب می کند.سپس در هر مرحله قبل از فعالیت Scheduler،یکی از خانه های راستای خود را به صورت تصادفی انتخاب کرده و در آن جا قرار می گیرد.در این سناریو،امکان این که چند گره در یک حوزه قرار بگیرند وجود دارد.در واقع محل تقاطع یال ها می تواند حوزه ای برای قرار گیری گره ها باشد.در نتیجه همسایه یک گره می تواند از حوزه او یا حوزه های مجاور که یک واحد در طول یا عرض تفاوت دارند،انتخاب شود.لازم به ذکر است که امکان آن وجود دارد که یک گره در موقعیتی قرار بگیرد که ایزوله بوده و دسترسی به همسایه ای نداشته باشد.در نتیجه scheduler تا جایی که از انتخاب دو گره که eامکان برقراری ارتباط دارند،عملیات انتخاب تصادفی agent ها را انجام می دهد.

سناریو پنجم:توپولوژی mesh همراه با mobility

این حالت نیز مشابه با حالت قبل است به جز تفاوتی که توپولوژی مش با توپولوژی grid دارد.در ادامه خواهیم دید که وجود mobility در حل بعضی از مسائل،بسیار کارا و سودمند خواهد بود.

سناریو ششم:توپولوژی Erdős-Rényi

در این حالت توپولوژی به صورت یک گراف تصادفی مدل می شود که دارای n راس بوده و هر یال با احتمال p به صورت مستقل از بقیه یال ها انتخاب می شود.در قسمت شبیه سازی برای یک n مشخص،پارامتر p را برابر با $p = \frac{\log(n^2)}{n}$ در نظر گرفتیم.

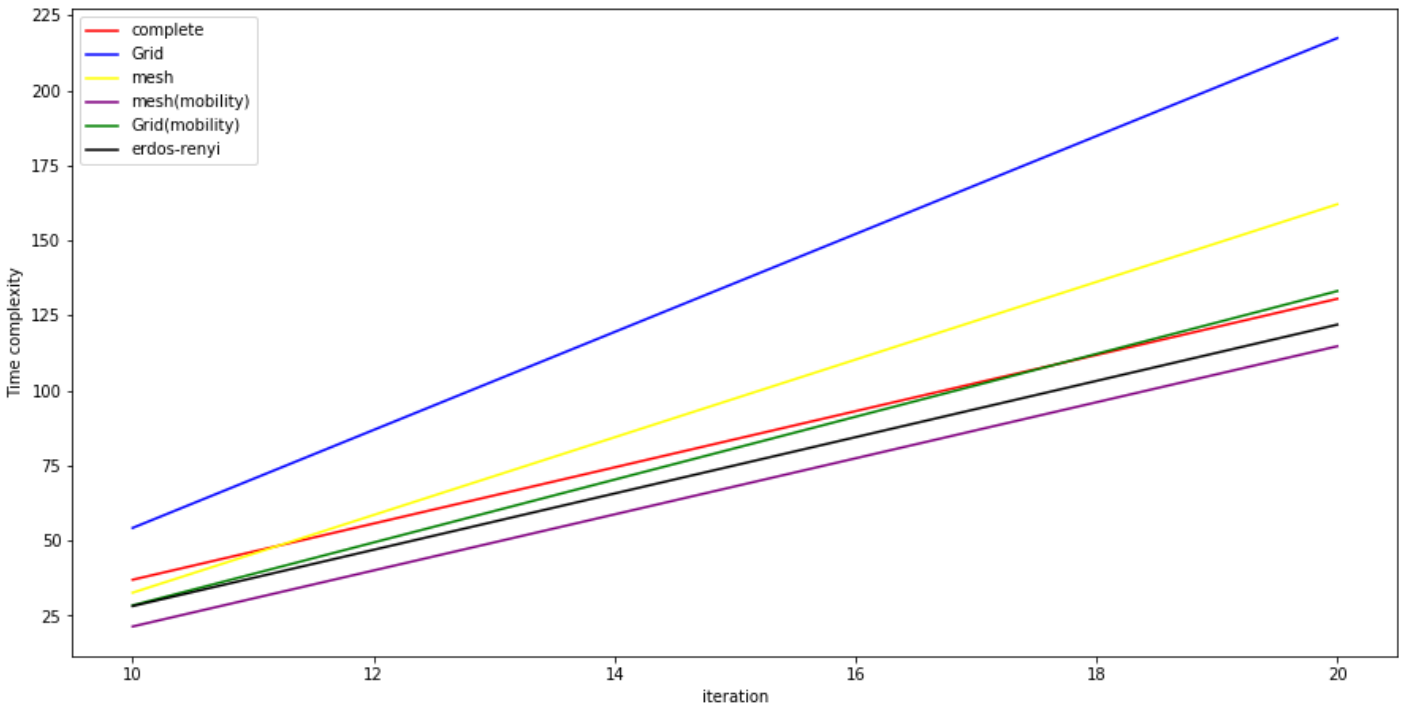
آزمایش های انجام شده و بررسی نتایج:

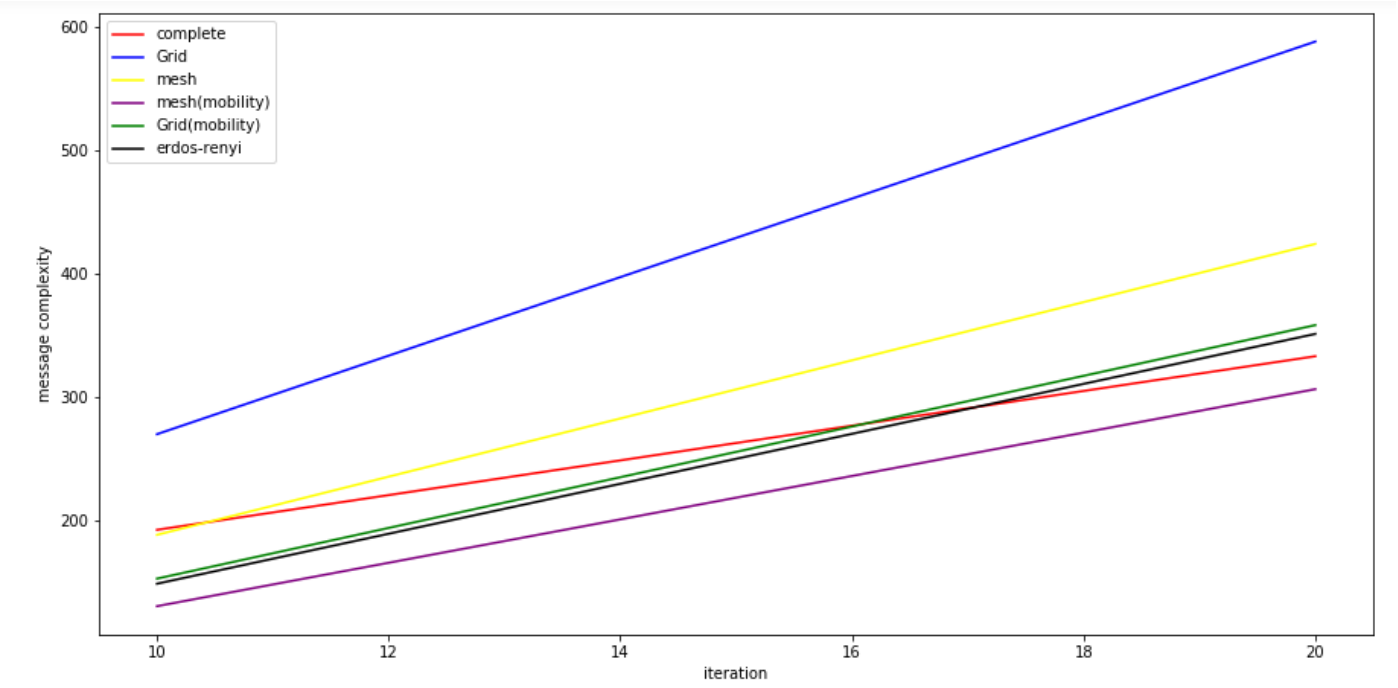
در این قسمت ما به بررسی سه مسئله مختلف که به کمک population protocols امکان تحقق آن ها وجود دارد،می پردازیم.

اولین مسئله،مسئله محاسبه تابع OR در شبکه است.در این مسئله،ورودی یکی از agent ها 1 و بقیه ورودی 0 دارند.ورودی 0 به state 0 و ورودی 1 به state 1 نگاشته می شود.برای محاسبه خروجی هم state 0 به خروجی 0 و state 1 به خروجی 1 نگاشته می شود.transition rule هم به صورت $(1,1) \xrightarrow{\delta} (1,0)$ است.

به این مسئله می توان به این دید نگاه کرد که در جمعیت اولیه یک بیمار وجود دارد و بقیه سالم هستند.در اثر ارتباط یه فرد بیمار با یک فرد سالم،آن فرد سالم هم بیمار می شود.

به طور کلی دو پارامتری که در سنجش شبیه سازی های خود در نظر گرفتیم،time complexity یعنی پیچیدگی زمانی اجرا و message complexity هستند.



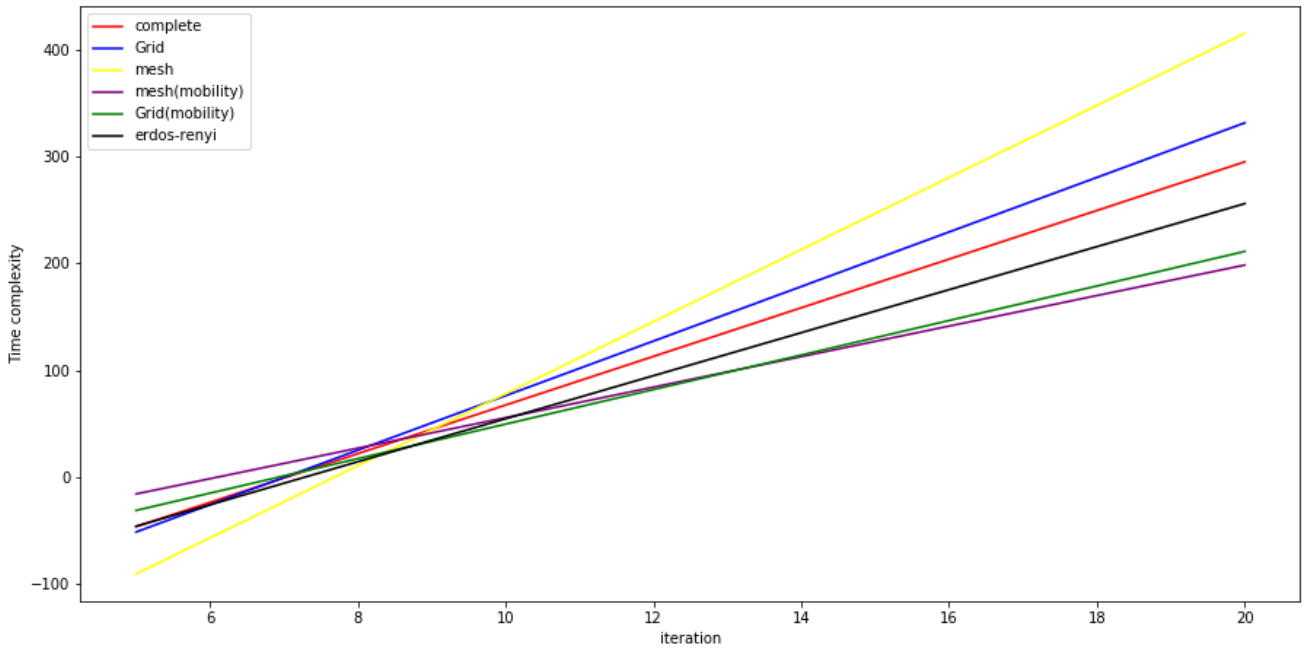


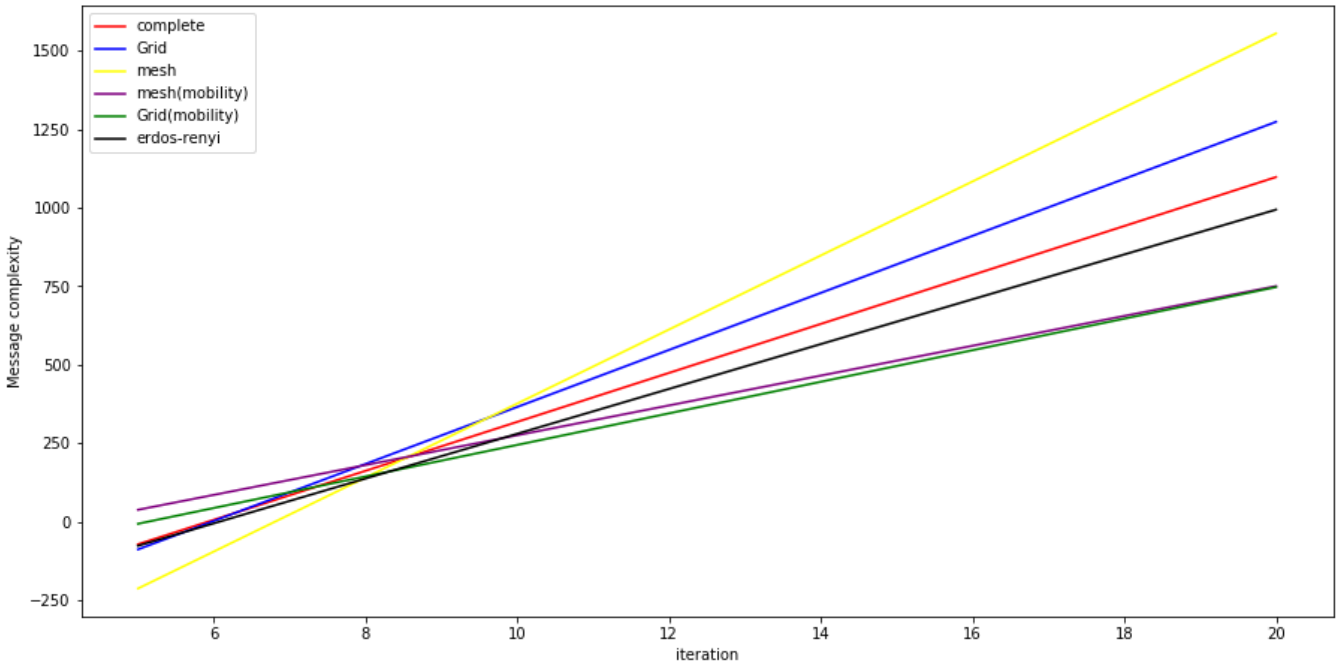
نمودار های رسم شده نشان دهنده پیچیدگی زمانی و تعداد پیام های ردوبدل شده در شبکه است.همان طور که مشخص است،mobility،می تواند تاثیر زیادی در شیوع بیماری داشته باشد به طوری که زمان همگرایی برای توپولوژی های grid و mesh،هنگامی که دارای mobility هستند،به مراتب کمتر است.این مورد با شهود ما نیز سازگار است،زیرا امکان جابه جایی،احتمال پخش بیماری را بیشتر کرده و فرد بیمار با تعداد افراد بیشتری در ارتباط است.در توپولوژی Erdős–Rényi هم از آن جایی که با افزایش جمعیت،احتمال ارتباط بین گره ها کمتر می شود،می توان دید که با افزایش جمعیت،تعداد پیام های جابه جا شده،شیب صعودی قابل توجهی دارد.

دومین مسئله،مسئله leader election است.در این مسئله،ورودی همه agent ها برابر با 1 می باشد.state ها و خروجی های می توانند مقادیر 0 یا 1 را اختیار کنند که این مقادیر نظیر به نظیر نگاشت می شوند.در این مسئله، Transition rule های ما به صورت زیر است.

$$(1,1) \xrightarrow{\delta} (0,1) \qquad (1,0) \xrightarrow{\delta} (0,1) \qquad (0,1) \xrightarrow{\delta} (1,0) \qquad (0,0) \xrightarrow{\delta} (0,0)$$

در این مسئله در نهایت یک نفر به عنوان رهبر انتخاب خواهد شد که خروجی 1 خواهد داشت،یعنی در آخر کار، فقط 1 نفر خروجی 1 خواهد داشت.



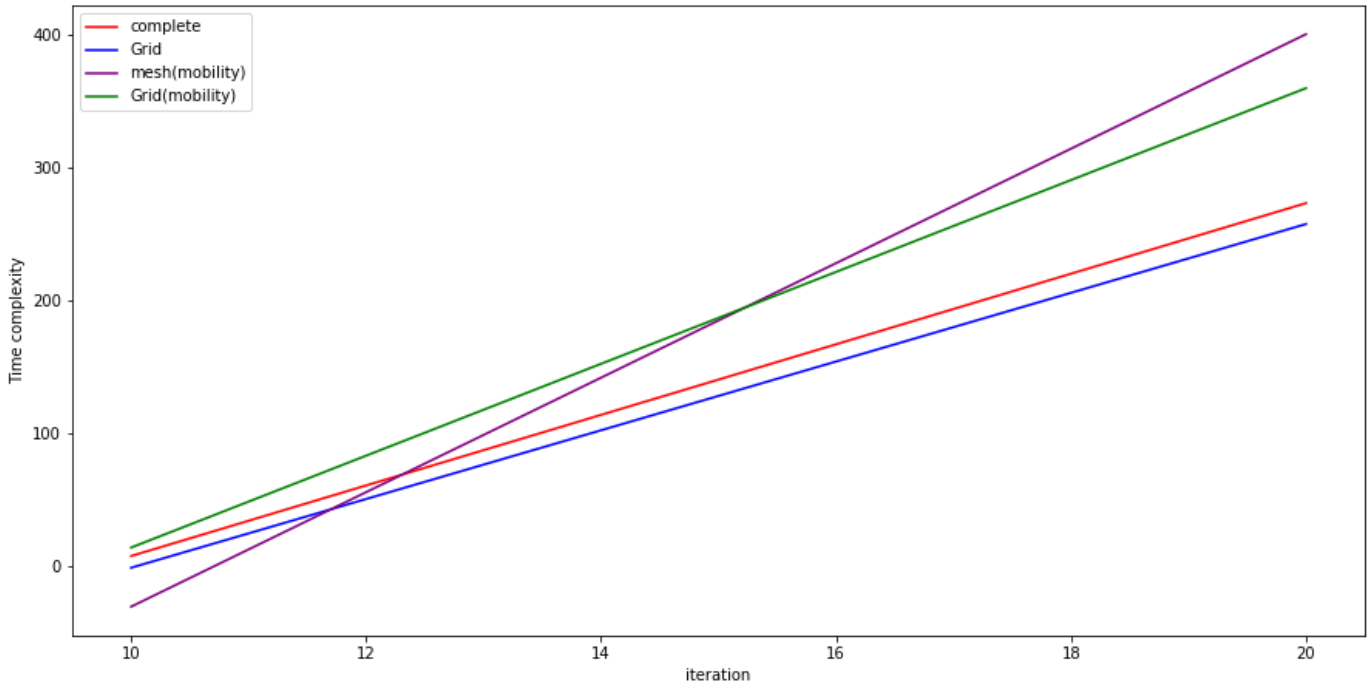


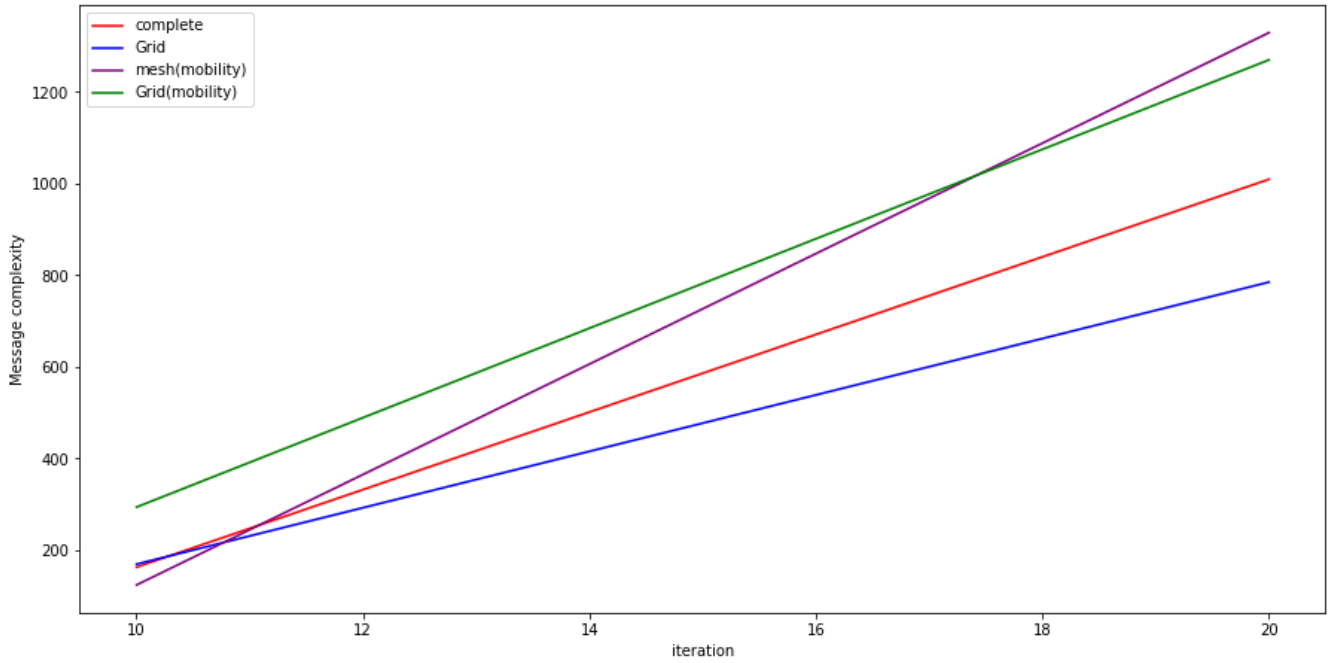
در این مسئله دوباره مشاهده می کنیم که mobility،تاثیر بسزایی در زمان همگرایی و پیام های رد و بدل شده داشته و به وضوح نسبت به سناریو های دیگر خروجی مطلوب تری دارد.با مقایسه این مسئله و مسئله قبل می توان گفت که این دو مسئله ماهیت یکسانی دارند،یعنی یک حالت و مقدار خاص شروع به پخش شدن در شبکه کرده تا زمانی که به خروجی مطلوب برسیم.در چنین حالتی طبیعتاً امکان جابه جایی agent ها و توانایی آن ها در برقراری ارتباط با گره های بیشتر،شانس همگرایی و رسیدن به خروجی مدنظر را به طور چشمگیری افزایش می دهد.

مسئله آخری که بررسی می کنیم،مسئله majority voting برای حالت باینری است.ورودی های agent های ما می توانند دو حالت A یا B داشته باشند که A به 0 و B به 1 نگاشت می شود.state های مسئله یکی از اعضای مجموعه $\{0,1,0^+,1^+\}$ می باشد.خروجی هم می تواند یکی از مقادیر 0 یا 1 را اختیار کند.state های 0 و 0^+ به خروجی 0 و state های 1 و 1^+ به خروجی 1 نگاشته می شوند.transition rule های ما به صورت زیر هستند.

$$(0^+,1^+) \xrightarrow{\delta} (1^+,0^+) \qquad (1,0) \xrightarrow{\delta} (0^+,1^+) \qquad (1,0^+) \xrightarrow{\delta} (1^+,1) \qquad (0,1^+) \xrightarrow{\delta} (0^+,0)$$

در این مسئله در نهایت agent های ما به خروجی می رسند که تعداد آن در ابتدا بیشتر باشد،یعنی اگر تعداد 1 ها در ابتدا بیشتر باشد،خروجی همه 1 خواهد شد یا بالعکس.





در این مسئله متاسفانه نتایج توپولوژی های mesh و Erdós-Rényi مطلوب نبودند و به همین خاطر از گزارش آن ها صرف نظر کردیم.در این مسئله مشاهده می کنیم که بر خلاف مسئله های قبل،mobility نه تنها تاثیر مثبت نداشته،بلکه نسبت به حالت های معمول نیز خروجی های نامطلوب تری دارد.این مسئله نسبت به دو مسئله قبل پیچدگی بیشتری داشته و همگرایی آن به نحوه پحش شدن گره ها در شبکه و میزان غلبه یک گروه به گروه دیگر بستگی بیشتری نسبت به دو مسئله قبل دارد.

منابع:

1) S. Salehkaleybar, H. Bandealinaeini, "Broadcast Distributed Voting Algorithm in Populatio Protocols"

2) J. Aspnes, E. Ruppert, "An Introduction to Population Protocols"

3) A. D. Sarwate, A. G. Dimarkis, "The Impact of Mobility on Gossip Algorithms"