



به نام خدا  
دانشگاه تهران  
دانشکده مهندسی برق و کامپیوتر



**مخابرات بی سیم**

**استاد مریم صباغیان**

**گزارش ارائه مقاله**

Clustering-Based Activity Detection Algorithms  
for Grant-Free Random Access in  
Cell-Free Massive MIMO

نام و نام خانوادگی	فاطمه جلیلی
شماره دانشجویی	۸۱۰۱۹۹۳۹۸
تاریخ تحویل	۱۴۰۳ / ۰۴ / ۱۹

## فهرست

۳	مقدمه
۴	فرمول بندی مسئله
۵	الگوریتم ها
۵	الگوریتم اول
۵	الگوریتم دوم
۶	الگوریتم سوم
۶	نتایج شبیه سازی

با توجه به افزایش ترافیک داده در شبکه های بیسیم، شبکه های نسل جدید بایستی در سه حوزه mMTC، eMMB و URLLC قابل اتکا باشند، به این معنا که بتوانند پهنای باند گسترده تر، پوشش تعداد دستگاه های زیاد و تاخیر کم تر را فراهم کنند. در این مقاله ما سیستم های cell-free massive MIMO را در حوزه mMTC بررسی می کنیم به این معنی که این سیستم ها باید بتوانند تعداد کاربران فراوان شبکه که data packet های کوچکی را ارسال می کنند در یک زمان و باند فرکانسی یکسان پشتیبانی کنند. در حوزه mMTC نسبت تعداد device های فعال در لحظه به تعداد کل آن ها به دلیل packet size کوچک از یک بسیار کم تر است.

در سیستم های سنتی grant-based، هر کاربر فعال به صورت رندوم یک pilot از مجموعه ای از pilot های عمود برهم انتخاب می کند و به وسیله آن به base station اطلاع می دهد که داده ای برای انتقال دارد، base station باید مسئله تداخل موجود را حل کند و بر اساس آن grant به device مورد نظر برای ارسال data اختصاص دهد. در سیستم های مخابراتی زمان همبستگی کانال محدود است بنابراین تعداد pilot های عمود نیز محدود خواهد بود و در حوزه mMTC که تعداد device ها بسیار زیاد است تعداد درخواست ها با pilot های یکسان در این سیستم سنتی افزایش می یابد لذا base station نمی تواند همزمان grant به همه این درخواست ها اختصاص دهد و زمان انتظار device ها برای دریافت grant نتیجتاً تاخیر سیستم افزایش خواهد یافت پس به سراغ سیستم های grant-free می رویم که هر device یک pilot مخصوص به خود دارد که این pilot ها دیگر بر هم عمود نیستند و ما نیاز به الگوریتم activity detection برای شناسایی کاربرانی که فعال هستند و می خواهند داده ارسال کنند پیدا می کنیم.

در سیستم های Massive MIMO که در 5G استفاده می شوند تعداد زیادی آنتن در کنار هم در هر cell قرار می گیرند، برای افزایش نرخ ارسال نیاز هست تا تعداد این آنتن ها در هر cell بیش تر و بیش تر شود و از تعداد زیادی access point استفاده شود ولی افزایش چگالی آنتنی شبکه inter-cell interference را افزایش می دهد، برای گذر از این محدودیت به سوی سیستم های cell-free پیش می رویم که در آن ها همه هر device می تواند با همه AP ها ارتباط برقرار کند و همه AP ها به یک CPU مرکزی متصل هستند.

در این مقاله توسعه چند الگوریتم activity detection برای grant-free random access در سیستم های cell-free بیان می شود. در حالت بهینه برای تشخیص فعالیت هر کاربر نیاز داریم تا از اطلاعات همه AP ها استفاده کنیم ولی این روش پیچیدگی محاسباتی بالایی دارد لذا از چند AP که کانال بهتری با کاربر مورد نظر دارند برای تشخیص فعالیت کاربر استفاده می کنیم لذا الگوریتم های معرفی شده از نوع cluster-based هستند که برای هر تشخیص فعالیت هر کاربر از یک cluster از AP ها استفاده می کنند.

در ادامه در بخش اول مسئله را فرمول بندی می کنیم، در بخش دوم الگوریتم ها معرفی می شوند و در آخر نتایج شبیه سازی را خواهیم داشت.

ارائه مقاله از طریق این [لینک](#) قابل دسترسی می باشد.

## فرمول بندی مسئله

با فرض های اولیه مثل مستقل بودن کانال های کاربران و دانسته بودن ( $s_k$ ) signature sequence یا همان pilot های مخصوص هر کاربر در AP ها مسئله را فرمول بندی می کنیم.

گین کانال بین کاربر  $k$ ام، آنتن  $n$ ام در AP  $m$ ام را با  $g_{mnk}$ ، ضریب مربوط به فعالیت کاربر  $k$ ام را با  $a_k$  و توان ارسالی توسط کاربر را با  $p_k$  نشان می دهیم بنابراین سیگنال دریافتی در آنتن  $n$ ام از AP  $m$ ام برابر خواهد بود با:

$$y_{mn} = \sum_{k=1}^K a_k \rho_k^{\frac{1}{2}} g_{mnk} s_k + w_{mn}$$

$$= \text{SD}_a \text{D}_{\rho}^{\frac{1}{2}} g_{mn} + w_{mn},$$

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_M \end{bmatrix} = \begin{bmatrix} \text{SD}_a \text{D}_{\rho}^{\frac{1}{2}} G_1 \\ \text{SD}_a \text{D}_{\rho}^{\frac{1}{2}} G_2 \\ \vdots \\ \text{SD}_a \text{D}_{\rho}^{\frac{1}{2}} G_M \end{bmatrix} + W$$

$$= \begin{bmatrix} S & 0 & \dots & 0 \\ 0 & S & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & S \end{bmatrix} \begin{bmatrix} \text{D}_a \text{D}_{\rho}^{\frac{1}{2}} & 0 & \dots & 0 \\ 0 & \text{D}_a \text{D}_{\rho}^{\frac{1}{2}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \text{D}_a \text{D}_{\rho}^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} G_1 \\ G_2 \\ \vdots \\ G_M \end{bmatrix} + W,$$

با ترکیب سیگنال های دریافتی در آنتن های هر AP در بعد دیگر ماتریس، سیگنال های دریافتی در AP M را می توانیم به صورت زیر بنویسیم:

که  $\gamma$  حاصل ضرب  $a \times p$  است.

از آن جایی که ستون های  $Y$  مستقل هستند ماتریس کواریانس

آن ماتریسی قطری است. با استفاده از ویژگی های ماتریس کواریانس احتمال  $Y$  به شرط  $\gamma$  مطابق زیر در حالت ماتریسی نوشته می شود:

$$Q = \begin{bmatrix} \text{SD}_{\gamma} \text{D}_{\beta_1} S^H & 0_L & \dots & 0_L \\ 0_L & \text{SD}_{\gamma} \text{D}_{\beta_2} S^H & \dots & 0_L \\ \vdots & \vdots & \ddots & \vdots \\ 0_L & 0_L & \dots & \text{SD}_{\gamma} \text{D}_{\beta_M} S^H \end{bmatrix} + \sigma^2 \mathbf{I}_{LM}$$

$$p(Y|\gamma) = \prod_{m=1}^M \prod_{n=1}^N \frac{1}{|\pi Q_m|} \exp(-y_{mn}^H Q_m^{-1} y_{mn})$$

$$= \prod_{m=1}^M \frac{1}{|\pi Q_m|^N} \exp(-\text{Tr}(Q_m^{-1} Y_m Y_m^H)),$$

که می خواهیم این احتمال ماکسیمم شود لذا با لگاریتم گیری به مسئله بهینه سازی رو به رو می رسیم:

به ازای هر  $m$  تابع رو به رو را تعریف می کنیم:

$$f^m(\gamma) = \log |Q_m| + \text{Tr} \left( Q_m^{-1} \frac{Y_m Y_m^H}{N} \right)$$

$$Q_m(\gamma) = \text{SD}_{\gamma} \text{D}_{\beta_m} S^H + \sigma^2 \mathbf{I}_L$$

$$= \sum_{k=1}^K \gamma_k \beta_{mk} s_k s_k^H + \sigma^2 \mathbf{I}_L,$$

به جای مینیمم کردن کل تابع فقط  $Q_m$  که درایه های قطری ماتریس کواریانس هستند را مینیمم می کنیم:

$Q_m$  حاصل جمع  $K$  تا ماتریس رنک ۱ با مقدار اولیه نويز است بنابراین به جای مینیمم کردن کل آن به صورت یکجا

در هر iteration آن را از یک بعد  $k$ ام مینیمم می کنیم تا جایی که نتوانیم بیش تر پیش برویم، از notation زیر استفاده

می کنیم که بیانگر آپدیت تابع با  $d$  step هر سری از یک بعد  $k$ ام است:

$$f_k^m(d) = f^m(\gamma + d e_k),$$

با استفاده از Sherman-Morrison rank-1 update:  $|Q_m + d \beta_{mk} s_k s_k^H| = (1 + d \beta_{mk} s_k^H Q_m^{-1} s_k) |Q_m|$ .

با جایگذاری عبارت فوق در تابع  $f$  داریم: (\*)

$$f_k(d) = c + \sum_{m=1}^M \left( \log(1 + d \beta_{mk} s_k^H Q_m^{-1} s_k) - d \beta_{mk} \frac{s_k^H Q_m^{-1} Q Y_m Q_m^{-1} s_k}{1 + d \beta_{mk} s_k^H Q_m^{-1} s_k} \right),$$

## الگوریتم ها

### الگوریتم اول

در این الگوریتم اندیس AP ای که بهترین کانال با کاربر مورد نظر را دارد با  $m'$  نشان می دهیم، با محاسبه فرمول (\*) به ازای  $m'$  مشتق گرفتن از آن و برابر صفر گذاشتن update

$$d^* = \frac{s_k^H Q_{m'}^{-1} Q_Y Y_{m'} Q_{m'}^{-1} s_k - s_k^H Q_{m'}^{-1} s_k}{\beta_{m'k} (s_k^H Q_{m'}^{-1} s_k)^2}.$$

step مطابق رو به رو بدست می آید:

برای جلوگیری از منفی شدن  $\gamma$  برای update کردن Qm از  $\max\{d^*, -\gamma_k\}$  استفاده می کنیم.

#### Algorithm 1 Coordinate Descend Algorithm for Estimating $\gamma$

**Input:** Observations  $Y_m, \forall m = 1, 2, \dots, M, \beta_{mk}, \forall m = 1, 2, \dots, M, k = 1, 2, \dots, K$

**Initialize:**  $Q_m^{-1} = \sigma^{-2} I_L, \forall m = 1, 2, \dots, M, \hat{\gamma}^0 = 0_K$

- 1: Compute  $Q_Y Y_m = \frac{1}{N} Y_m Y_m^H, \forall m = 1, 2, \dots, M$
- 2: **for**  $i = 1, 2, \dots, I$  **do**
- 3: Select an index set  $\mathcal{K}$  from the random permutation of set  $\{1, 2, \dots, K\}$
- 4: **for**  $k \in \mathcal{K}$  **do**
- 5: Find the strongest link or AP for device  $k$ , i.e.,  $m' = \operatorname{argmax}\{\beta_{mk}\}$
- 6:  $\delta = \max \left\{ \frac{s_k^H Q_{m'}^{-1} Q_Y Y_{m'} Q_{m'}^{-1} s_k - s_k^H Q_{m'}^{-1} s_k}{\beta_{m'k} (s_k^H Q_{m'}^{-1} s_k)^2}, -\hat{\gamma}_k \right\}$
- 7:  $\hat{\gamma}_k^i = \hat{\gamma}_k^{i-1} + \delta$
- 8: **for**  $m = 1, 2, \dots, M$  **do**
- 9:  $Q_m^{-1} \leftarrow Q_m^{-1} - \delta \frac{\beta_{mk} Q_m^{-1} s_k s_k^H Q_m^{-1}}{1 + \delta \beta_{mk} s_k^H Q_m^{-1} s_k}$
- 10: **end for**
- 11: **end for**
- 12: **if**  $f(\hat{\gamma}^i) \geq f(\hat{\gamma}^{i-1})$  **then**
- 13:  $\hat{\gamma} = \hat{\gamma}^{i-1}$
- 14: **break**
- 15: **end if**
- 16:  $\hat{\gamma} = \hat{\gamma}^i$
- 17: **end for**
- 18: **return**  $\hat{\gamma}$

به این ترتیب در هر حلقه یک ترتیب رندوم از کاربران انتخاب می کنیم و مقدار اولیه Qm را برابر نویز در نظر می گیریم که در فرمول های صفحه قبل بدست آمد و در هر مرحله با محاسبه update step مقدار  $\gamma$  و از روی آن مقدار Qm را بدست می آوریم و این مراحل را تا جایی ادامه می دهیم که مقدار تابع  $f$  کم تر نشود. در نهایت بعد از پایان iteration ها با استفاده از یک threshold فعال بودن یا نبودن هر کاربر را از روی  $\gamma$  بدست می آوریم.

### الگوریتم دوم

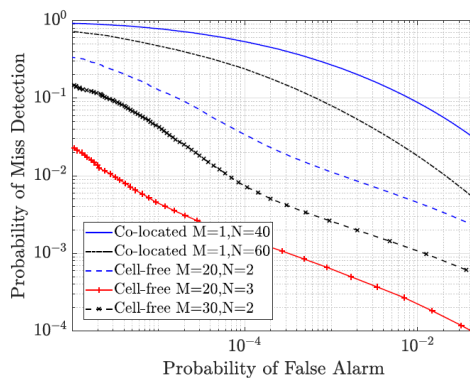
این الگوریتم دقیقاً مشابه الگوریتم اول است با این تفاوت که به جای استفاده از صرفاً یکی از AP ها که بهترین کانال را با کاربر مورد نظر دارد، T تا از بهترین ها را انتخاب می کنیم و با مجموعه اندیس  $\mathcal{M}_k$  نشان می دهیم. برای محاسبه update step تابع  $f_k$  را به ازای همه این اندیس ها باهم جمع می کنیم، از حاصل مشتق گرفته و برابر صفر قرار می دهیم، مجموعه ی ریشه های این معادله را با D نشان می دهیم و مشابه قبل برای جلوگیری از منفی شدن  $\gamma$ ،  $-\gamma$  را هم به این مجموعه اضافه می کنیم، تابع  $f_k$  را به ازای همه اعضا مجموعه محاسبه و آن update step ای که کمترین مقدار تابع را نتیجه می دهد به عنوان update step نهایی بر می گزینیم. این کار ها در هر iteration تکرار می شود و مشابه الگوریتم ۱ با این update step بدست آمده روند به جای قبلی روند را دنبال می کنیم.

## الگوریتم سوم

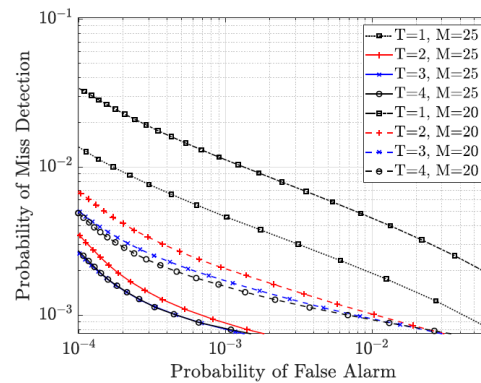
با توجه به اینکه ما در حوزه mMTC پژوهش می کنیم تعداد کاربران فعال در هر لحظه بسیار اندک است و آپدیت  $Q_m$  تنها در برخی درایه ها در حالی که ابعاد ماتریس بسیار بزرگ است انجام می شود لذا می توان نتیجه گرفت نیاز به آپدیت  $Q_m$  در هر مرحله با  $\gamma$  جدید نداریم و می توان مرحله ۹ از الگوریتم اول را به بیرون for loop منتقل کرد و نتیجتاً for loop را به جای سری انجام دادن به ازای هر کاربر در چندین شاخه موازی با تعداد چندین کاربر انجام داد، در این الگوریتم به جای آپدیت  $Q_m$  بعد از iteration چند iteration یک بار  $Q_m$  را آپدیت می کنیم.

## نتایج شبیه سازی

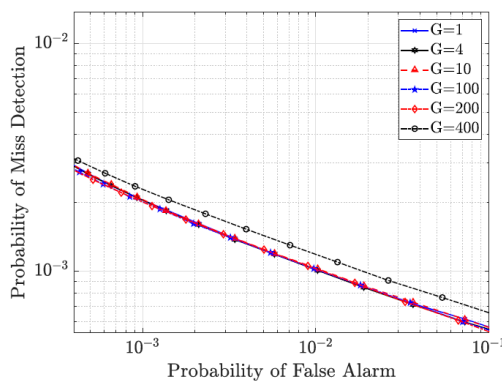
نتایج شبیه سازی را با استفاده از نمودارهای ROC نمایش می دهیم لذا هر چه قدر نمودار تقعر بیش تری پیدا کند و از خط  $y=-x$  به سمت پایین دور تر شود در واقع الگوریتم performance بهتری داشته است.



تصویر ۱: نتایج شبیه سازی الگوریتم اول



تصویر ۲: نتایج شبیه سازی الگوریتم دوم



تصویر ۳: نتایج شبیه سازی الگوریتم سوم

برای بررسی عملکرد الگوریتم اول نمودار عملکرد آن در کنار Massive MIMO با آنتن های کنار هم در مقابل cell-free رسم شده است که مشاهده می شود cell-free بسیار بهتر عمل کرده است، از طرفی با افزایش تعداد آنتن ها به ازای هر AP عملکرد الگوریتم اول بهبود می یابد.

برای بررسی عملکرد الگوریتم دوم، نتایج آن به ازای تعداد مختلف از بهترین AP ها که برای انجام الگوریتم بر می گزینیم رسم شده است، مشاهده می شود تقعر نمودار زمانی که  $T$  افزایش می یابد به شدت بیش تر می شود و در مقایسه با الگوریتم اول که همان  $T=1$  است بهبود قابل توجهی می یابد.

عملکرد الگوریتم سوم تا زمانی که تعداد شاخه ها خیلی زیاد نشود ( $G=400$ ) مشابه الگوریتم دوم است با این تفاوت که performance زمانی بهتری دارد.