به نام خدا

دانشگاه تهران

پردیس دانشکده‌های فنی

دانشکده برق و کامپیوتر

مخابرات بی سیم

تمرین کامپیوتری دوم

محمد حیدری

810197494

خرداد ماه 1401

فهرست:

1. نحوه پیاده سازی و روابط تئوری:

دراین پروژه قصدداریم نحوه تخصیص بهینه توان در کانال های مخابراتی در یکی از ساده ترین مدل های سیستم مخابرات بیسیم در حضور تداخل و نویز را بررسی کنیم.

اطلاعات داده شده در این سوال به شرح زیر است.

- یک سیستم مخابراتی بیسیم شامل یک BS و n کاربر دراختیار داریم.
- همچنین تعداد n کانال مخابراتی برای ارسال تعدادی رشته داده به n کاربر ایجاد میکنیم.
- پارامترهای مسئله توان های ارسالی در کانال ها هستند که با P_j نمایش میدهیم.
- همچنین فرض میکنیم G_{ij} بهره کانال j ام برای کاربر i ام باشد.
- دراین مسئله محدودیت ارسال داریم که قیودی را برای حداکثر توان ارسال بما تحمیل خواهند کرد.

$$0 < P_i \leq \; P_i^{max} \quad \forall i \; \in \{1,2,\dots,n\}$$

عملا دراین مسئله ما قصد داریم بااستفاده از الگوریتم *gradient Ascent* یک مسئله بهینه سازی را حل نماییم که تابع هدف آن بصورت زیر تعریف میشود و پارامترهای آن نیز یک بردار n تایی از توان های ارسالی در n کانال مورد بررسی خواهند بود.(درواقع بروزرسانی الگوریتم روی توان های ارسالی انجام خواهد شد.)

$R_i = \log{(1 + \gamma_i)}$

Target-function : $\sum_{i=1}^{n} \log{(Ri)}$

باتوجه به ماتریس G داده شده میدانیم که قطراصلی این ماتریس بهره قسمت مطلوب سیگنال را بدست میدهد و قطرفرعی آن ترم های ناشی از تداخل را بما میدهد لذا *SINR* را میتوانیم با رابطه ی زیر بدست آوریم:

$$SINR_i = \frac{P_i . G_{ii}}{\sum_{i \neq j} G_{ij} . P_j}$$

مهمترین بخش درپیاده سازی یک بهینه ساز برای حل این مسئله دردست داشتن توابع *f(R)* و همچنین تابع گرادیان متناظر با تابع هدف میباشد لذا قبل از شروع حل و پیاده سازی مسئله میبایست توابعی تعریف کنیم که تابع هدف و ماتریس گرادیان متناظر با آن را با گرفتن هر ست n تایی از توان های ارسالی بدست دهد. لذا درادامه درپی آن هستیم که این توابع را پیاده سازی نماییم:

مطابق داده های سوال 5 دیتای مختلف دردست داریم که هریک ماتریس های *G,N0,P_max* را بما خواهد داد.

پیاده سازی تابع هدف:

دراین قسمت تابعی پیاده سازی کرده ایم که با گرفتن ماتریس *G* و *N0* و *P* مقادیر *SINR,Ri* و درنهایت مقدار تابع هدف درآن نقطه را بما خواهد داد.

$$SINR_i = \frac{P_i . G_{ii}}{\sum_{i \neq j} G_{ij} . P_j}$$

$$R_i = \log\left(1 + \gamma_i\right)$$

$$\textit{Target-function}: \sum_{i=1}^{n} \log\left(Ri\right)$$

```
function [f_res,Ri,SINR_vector]=f(P,G,N0)
signal=0;
for i=1:1:length(P)
signal=G(i,i)*P(i);
ISI=0;
for j=1:1:length(P)
if i ~=j
ISI=ISI+G(i,j)*P(j);
end
end
SINR(i)=signal/(N0(i)+ISI);
end
SINR_vector=SINR;
Ri=log10(1+SINR);
f_res=sum(log10(Ri));
end
```

Figure 1. implementing f function using MATLAB

پیاده سازی تابع محاسبه گرادیان:

دراین قسمت تابعی پیاده سازی کرده ایم که با گرفتن ماتریس *G* و *N0* و *P* بردار گرادیان به ازای ست *n* تایی توان های ارسال را بدست میدهد.

میتوان گفت این تابع مهم ترین قسمت پیاده سازی را تشکیل میدهد زیرا محاسبه ی درست گرادیان تابع هدف در عملکرد تابع بهینه ساز به ویژه در سرعت همگرایی نقش مهمی را خواهد داشت.

نکته قابل توجه دیگر نیز دراین پیاده سازی این است که ما میبایست بصورت دستی محاسبات مربوط به گرادیان را انجام دهیم که در ادامه تمامی این مراحل ضمیمه شده اند:

بااستفاده ازمشتق زنجیره ای داریم:

$R_i = \log(1+\gamma_i)$

Target-function : $\sum_{i=1}^{n} \log(Ri)$

$$\frac{\partial f}{\partial P_i} = \frac{1}{Ri}\cdot\frac{1}{\ln(10)}\cdot\frac{1}{1+\gamma_i}\cdot\frac{1}{\ln(10)}\cdot\frac{\partial \gamma_i}{\partial P_i} + \sum_{k\neq i}\frac{1}{R_k}\cdot\frac{1}{\ln(10)}\cdot\frac{1}{1+\gamma_k}\cdot\frac{1}{\ln(10)}\cdot\frac{\partial \gamma_k}{\partial P_i}$$

$$SINR_i = \frac{P_i.G_{ii}}{\sum_{i\neq j} G_{ij}.P_j}$$

$$\frac{\partial \gamma_i}{\partial P_i} = \frac{G_{ii}}{\sum_{i\neq j} G_{ij}.P_j} =$$

$$\frac{\partial \gamma_k}{\partial P_i} = \sum_{k\neq i}\frac{-P_k G_{kk} G_{ki}}{\sum_{k\neq j}(G_{kj}.P_j)^2} = \sum_{k\neq i}\frac{-{\gamma_k}^2 G_{ki}}{P_k G_{kk}} \xrightarrow{yields}$$

$$\boxed{\xrightarrow{yields} \frac{\partial f}{\partial P_i} = \frac{1}{Ri}\cdot\frac{1}{\ln(10)}\cdot\frac{1}{1+\gamma_i}\cdot\frac{1}{\ln(10)}\cdot\frac{\gamma_i}{P_i} + \sum_{k\neq i}\frac{1}{R_k}\cdot\frac{1}{\ln(10)}\cdot\frac{1}{1+\gamma_k}\cdot\frac{1}{\ln(10)}\cdot\frac{-{\gamma_k}^2 G_{ki}}{P_k G_{kk}}}$$

رابطه بدست آمده دربالا الگوی کار من قرارگرفت و بااستفاده از 2 حلقه تو در تو تابع محاسبه ی گرادیان را درمتلب پیاده سازی کرده ام :

```
function grad=gradient_calc(P,G,N0)
[~,Ri,SINR_vector]=f(P,G,N0);
for i=1:1:length(P)
collector=(SINR_vector(i)/((log(10)^2)*Ri(i)*(1+SINR_vector(i))))/P(i);
for k=1:1:length(P)
if i~=k
collector=collector+(1/((log(10)^2)*Ri(k)*(1+SINR_vector(k))))*((-SINR_vector(k)^2*G(k,i))/(G(k,k)*P(k)));
end
end
grad(i)=collector;
end
end
```

Figure 2. implementing gradient-calc function using MATLAB

الف: الگوریتم *gradient descent* :

همانطور که میدانیم این الگوریتم مبتنی بر یک روش بهینه سازی برای یافتن مینیمم تابع میباشد که دراینجا ما با حرکت کردن در راستای مثبت گرادیان بجای حرکت در راستای منفی گرادیان عملا آزان به عنوان یک بهینه ساز برای یافتن ماکسیمم تابع استفاده خواهیم کرد.

مراحل الگوریتم:

1) انتخاب رندوم یک نقطه اولیه
2) محاسبه گرادیان درآن نقطه
3) حرکت درجهت مثبت گرادیان بااستفاده از یک ضریب اسکیل کننده به نام پارامتر یادگیری
4) تکرار مراحل 2 و 3 تا رسیدن به همگرایی
5) پایان الگوریتم با ارضاشدن شرط خاتمه یا رسیدن به حداکثر تکرار.

حال توضیح خواهیم داد که چرا این الگوریتم به نقطه بهینه همگرا خواهد شد:

فرض کنیم مشتق مرتبه اول مخالف صفر باشد میتوانیم بیان کنیم که مقادیر تابع دراطراف یک نقطه خاص را میتوان بصورت زیر نوشت : (توجه شود این notation از لکچرهای دکتر کلهر از درس شبکه های عصبی برداشت شده است.)

$$J(p_0 + \delta p) \cong J(p_0) + (\delta p)^T \nabla j(p_0) \qquad ||\delta p|| \to 0$$

اگر تعریف کنیم که$\delta p = a\nabla j(p_0)$ و فرض کنیم که الفا همان لرنینگ ریت الگوریتم و یک مقدار به اندازه کافی کوچک مثبت میباشد داریم :

$$J(p_0 + \delta p) \cong J(p_0) + a\left|\left|\nabla j(p_0)\right|\right|^2 > J(p_0)$$

همانطور که از رابطه بالا مشخص است درصورتی که درجهت مثبت گرادیان حرکت کنیم ینی الفا عدد کوچک مثبتی باشد قطعا درجهت ماکسیمم حرکت خواهیم کرد که به معنای همگراشدن به ماکسیمم خواهد بود.

همچنین نکته مهمی دراین بین توجه به قیدهای مسئله میباشد که میبایست به نوعی آنها را در حل این مسئله لحاظ نماییم:

$$0 < P_i \leq P_i^{max} \quad \forall i \in \{1,2,\dots,n\}$$

برای اعمال قیود مسئله کافی است همان مسئله یافتن ماکسیمم را حل نماییم بااین تفاوت که درهرمرحله آپدیت چک نماییم که تمامی مقادیر توان ارسالی از مقادیر ماکسیمم متناظرخود کمتر باشند و دراین صورت الگوریتم را ادامه خواهیم داد اما درصورتی که هریک از مقادیر توان ارسالی از مقادیر ماکسیمم متناظر خود بزرگ تر باشند ابتدا مقدار ماکسیمم را جایگزین آنها خواهیم کرد و سپس ادامه الگوریتم را پیش خواهیم گرفت دراین حالت قیدهای مسئله را نیز اعمال کرده ایم.

بروزرسانی پارامترها:

این الگوریتم از قاعده آپدیت ساده ای پیروی میکند که بنا به دلایلی که درادامه توضیح داده خواهد شد همگرایی آن تضمین شده میباشد:

با درنظر گرفتن پارامترهای مسئله:

$$P_{k+1} = P_k + alpha * g(P_k)$$

پارامترهای مسئله :

1. نرخ یادگیری: (learning-rate) : این پارامتر مقدار گرادیان را اسکیل میکند و به نوعی مقدارطول گام را در مسائل بهینه سازی کنترل میکند.
2. خطای همگرایی(tolerance) : این پارامتر مقدار خطایی را مشخص میکند که اگر شرط همگرایی ازآن کمتر شد به معنای رسیدن به همگرایی است.
3. ماکزیمم تکرار (max-iteration) : این پارامتر صرفا باین هدف تعریف شده است که درصورتی که الگوریتم به دلیل کوچک بودن خطا همگرا نشد بعد از تعداد مشخصی تکرار پایان یابد.
4. نقطه شروع: (start-point) : این پارامتر معمولا بصورت رندوم انتخاب میگردد و نقطه شروع را مشخص میکند.

شرط پایان های مختلفی برای این الگوریتم میتوان تعریف کرد:

$$\text{norm(gradient vector)} < epsilon$$

$$\text{norm(Pnew} - \text{P)} < epsilon$$

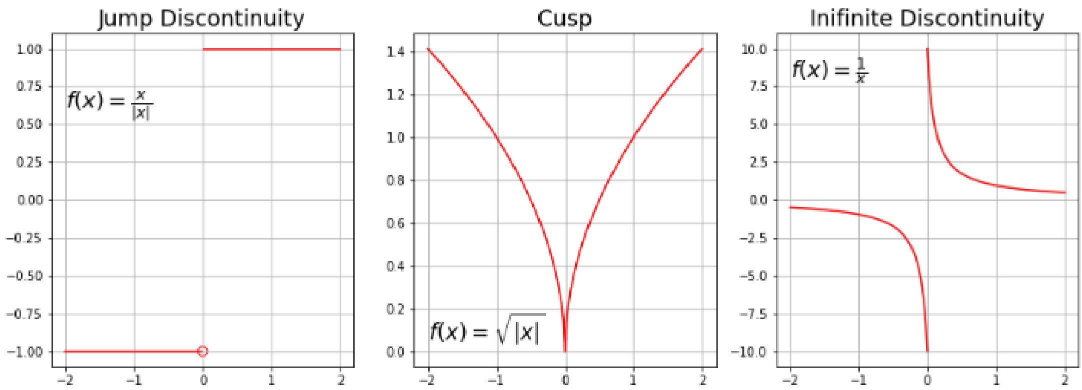
که دراین پیاده سازی من از شرط پایان دوم برای خارج شدن از حلقه while استفاده کرده ام.

شرط تضمین همگرایی الگوریتم:

الگوریتم گرادیان معرفی شده درقسمت های قبل برای همه انواع توابع کار نمیکند و دو شرط لازم برای تضمین همگرایی آن وجود دارد که درادامه به آن خواهیم پرداخت:

1. تابع مفروض میبایست مشتق پذیر باشد. درواقع میبایست در تمام دامنه تعریف خود مشتق پذیر باشد.

غالبا توابع غیرمشتق پذیر دارای ناپیوستگی، جهش و مواردی ازاین دست هستند.



Examples of non-differentiable functions; Image by author

2. تابع مورد بررسی میبایست حتما محدب باشد تا بتوان همگرایی آن را تضمین کرد. (درواقع این شرط تضمین میکند که در ماکسیمم های محلی الگوریتم به دام نیفتد و مستقیما به ماکسیمم گلوبال همگرا شویم)

برای محدب بودن تابع میبایست شرط زیر برقرار باشد:

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda)f(x_2)$$

یا اینکه مشتق دوم آن مثبت باشد:

$$\frac{d^2 f(x)}{dx^2} > 0$$

همچنین اینکه میبایست پارامترهای اولیه مسئله را به درستی انتخاب کنیم و عملا انتخاب هایپرپارامترها بصورت تجربی و با تست کیس هایی انجام خواهد شد.

برای مثال مقدار لرنینگ ریت میبایست به اندازه کافی کوچک باشد که هم تضمین همگرایی را داشته باشد و هم اینکه خیلی سرعت اجرای الگوریتم را کند و محدود نکند.

لذا انتخاب طول گام مناسب و همچنین شروع از یک نقطه مناسب نیز میتوانند تضمین کننده همگرایی الگوریتم باشند چرا که همانطور که دربالا اثبات شد با انتخاب لرنینگ ریت مثبت و قرار دادن علامت مثبت در گرادیان درجهت رسیدن به ماکسیمم حرکت خواهیم کرد.

ب: گزارش نتایج:

دراین قسمت مطابق توضیحات ارایه شده در بالا و بااستفاده ازتوابع تعریف شده الگوریتم را پیاده سازی کرده ایم که در ادامه و به ترتیب خواسته های سوال را برای هریک از دیتاهای گفته شده ضمیمه خواهیم کرد:

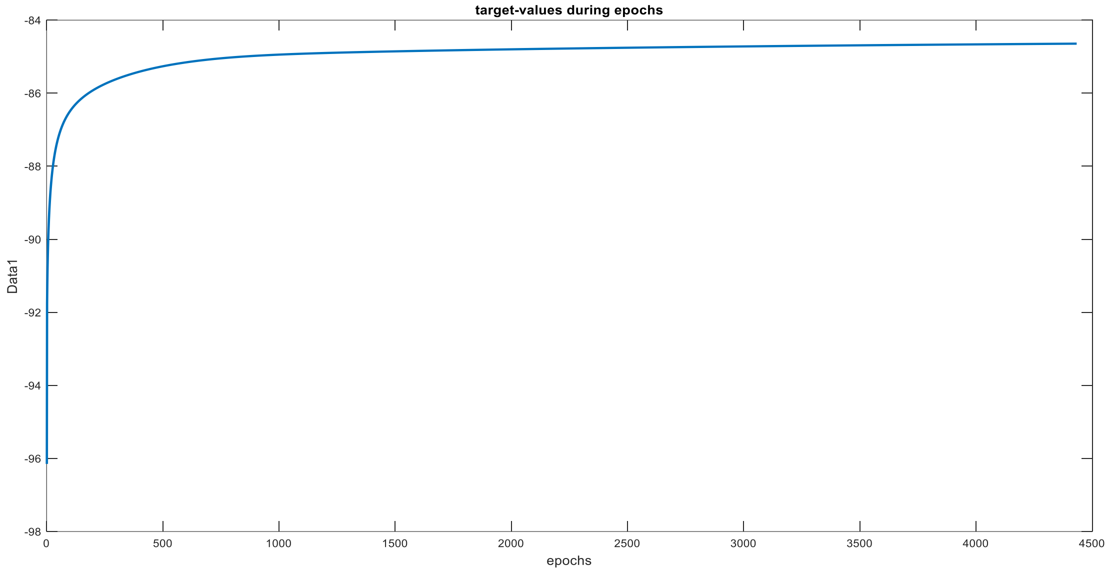


Figure 3. function-values during epochs(Dataset #1)

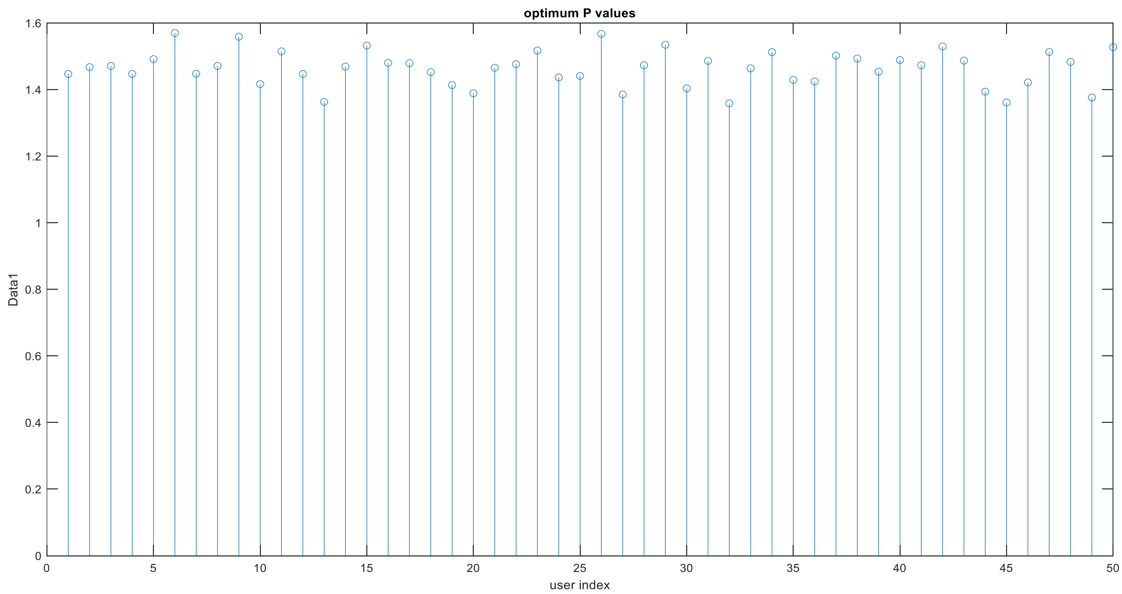


Figure 4. optimum P values during user index(Dataset #1)

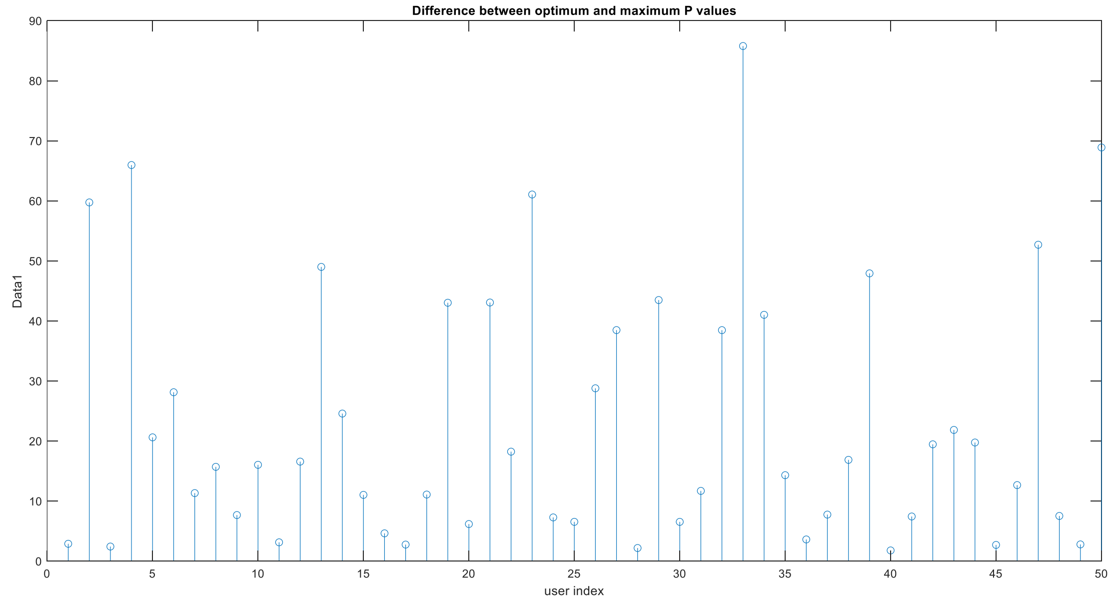


Figure 5. diff between opt and max P values during user index(Dataset #1)

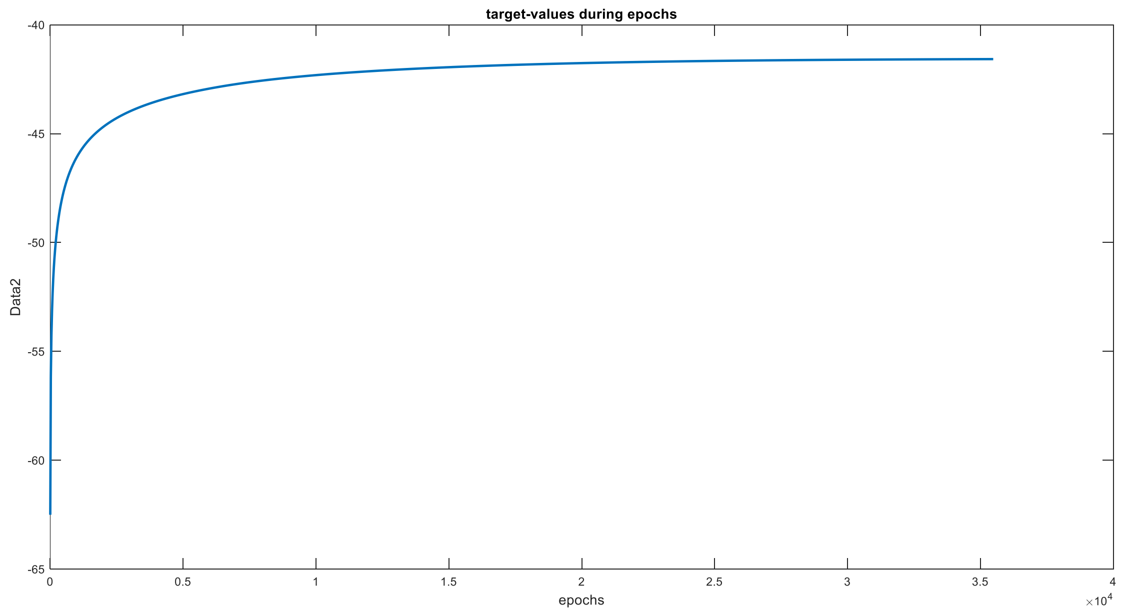


Figure 6. function-values during epochs(Dataset #2)

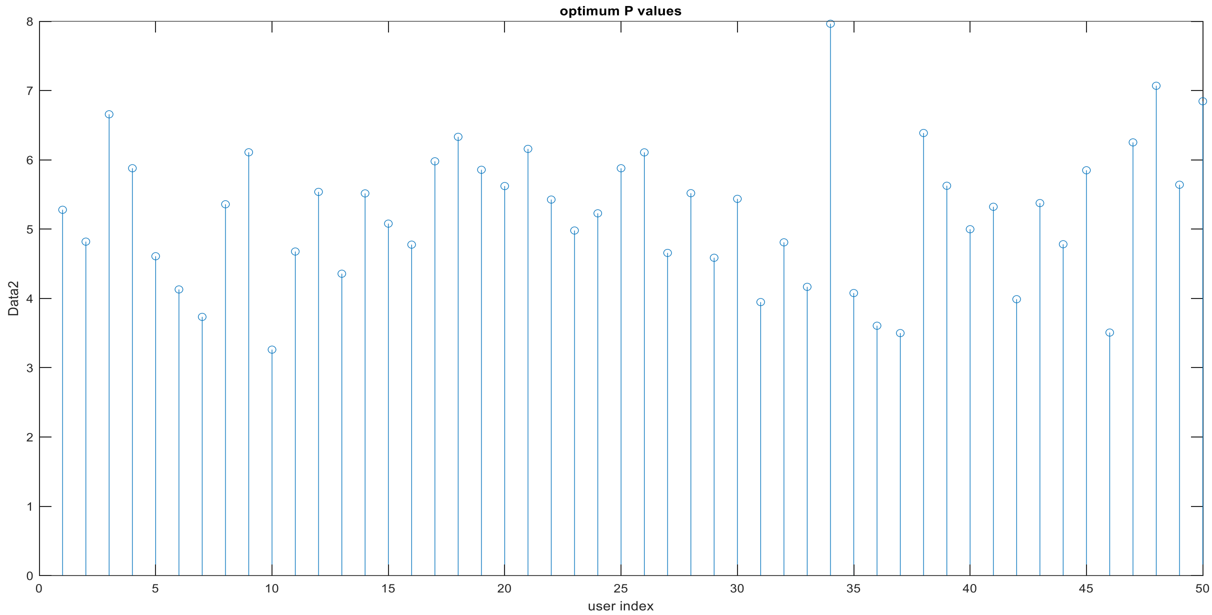


Figure 7. optimum P values during user index(Dataset #2)

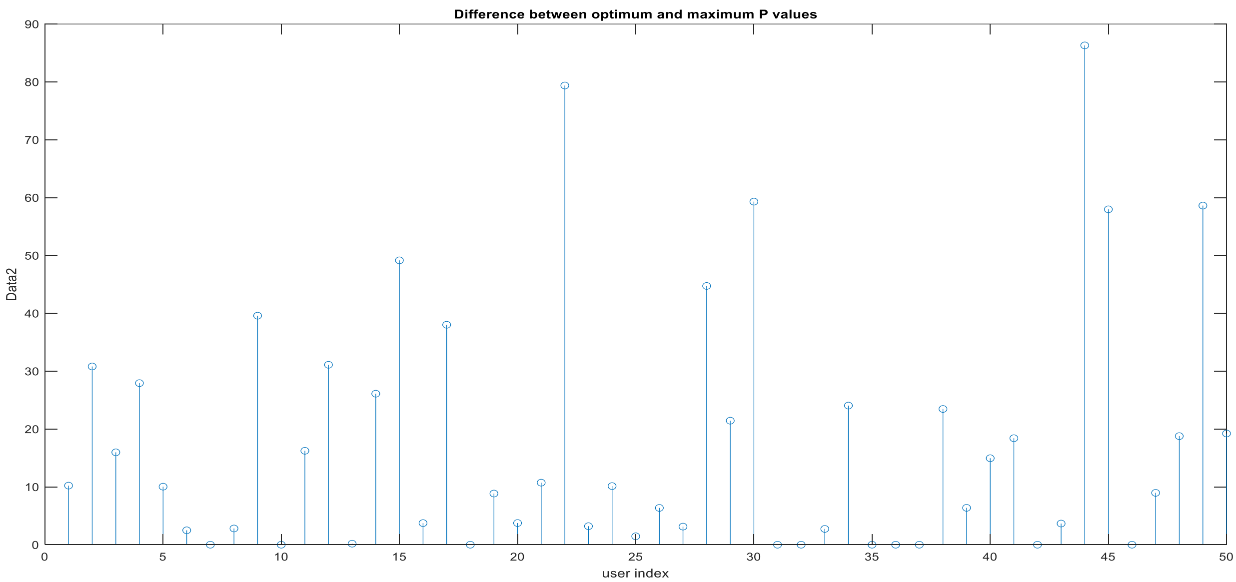


Figure 8. diff between opt and max P values during user index(Dataset #2)

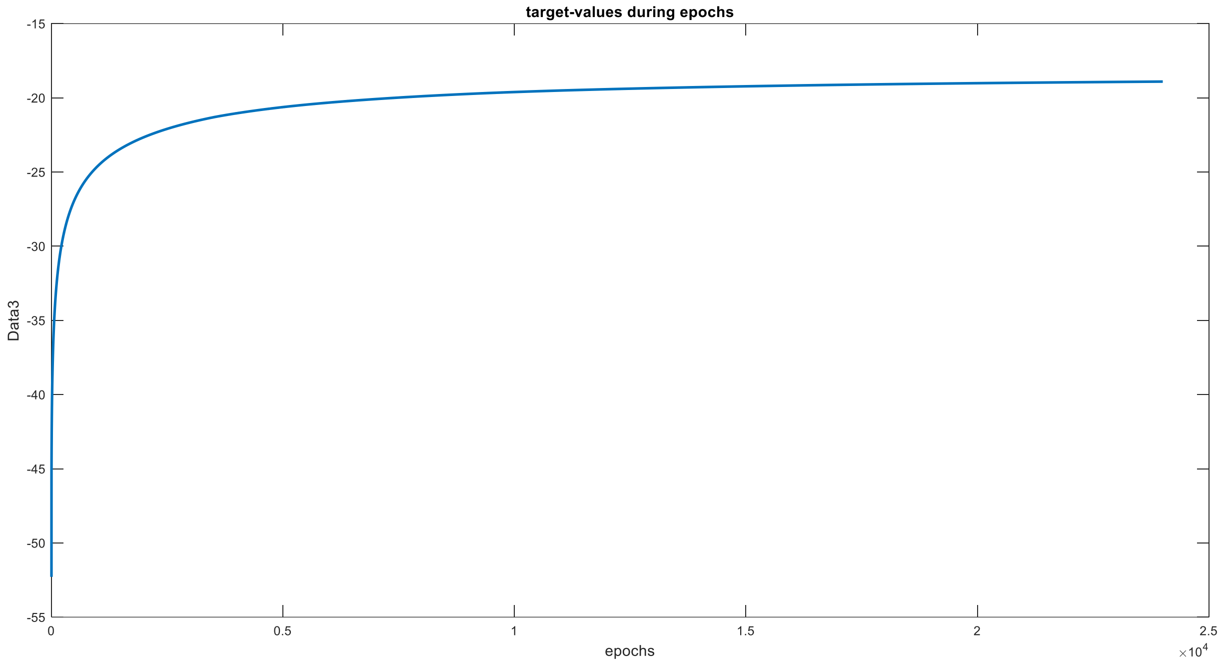


Figure 9. function-values during epochs(Dataset #3)

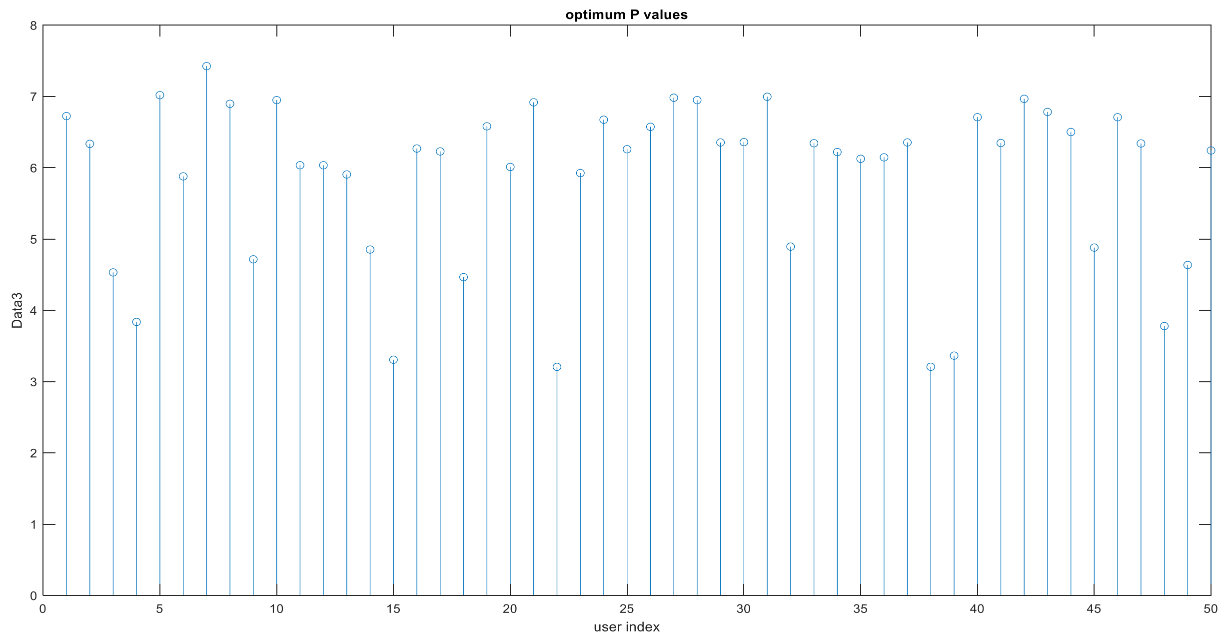


Figure 10. optimum P values during user index(Dataset #3)

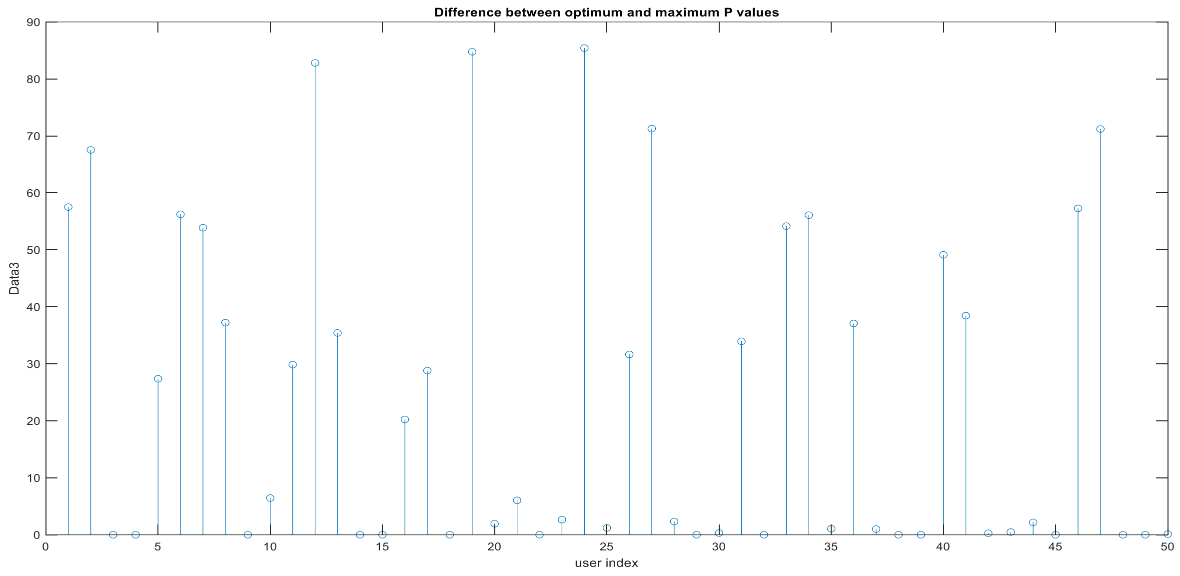


Figure 11. diff between opt and max P values during user index(Dataset #3)

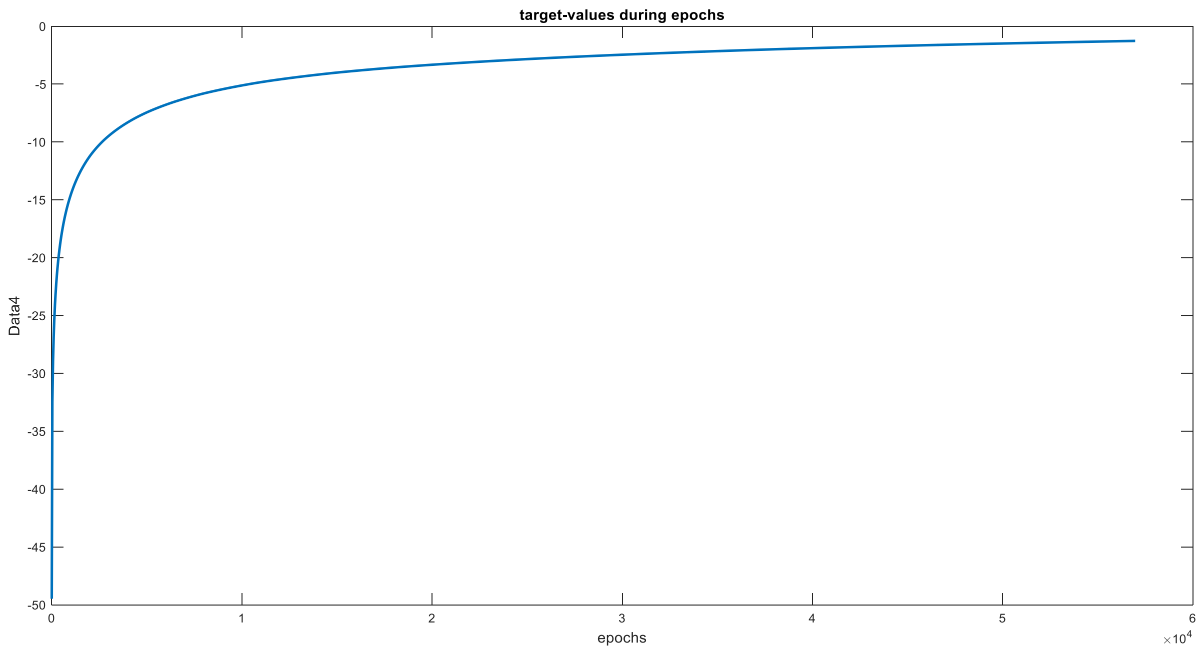


Figure 12. function-values during epochs(Dataset #4)

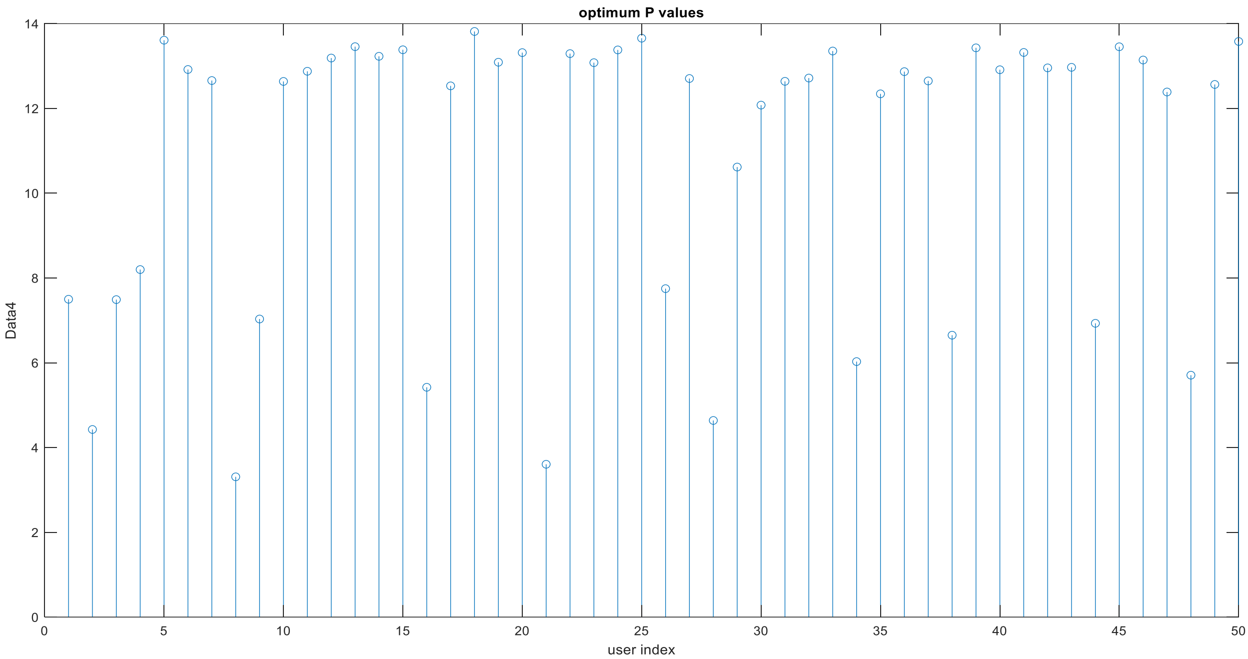


Figure 13. optimum P values during user index(Dataset #4)

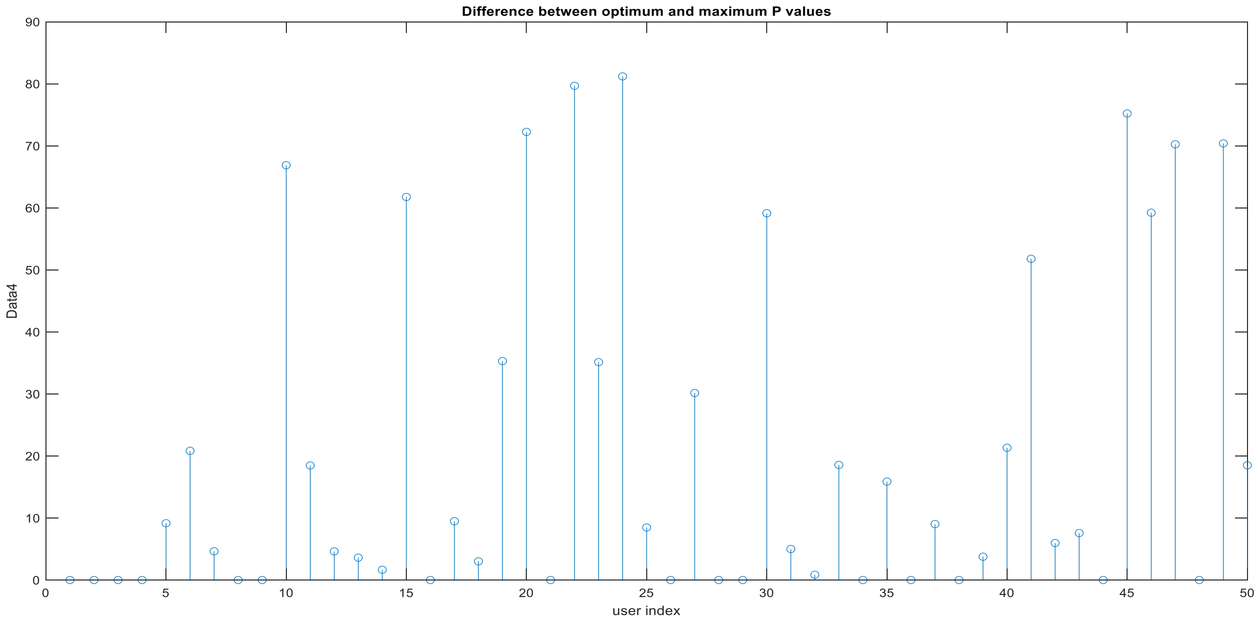


Figure 14. diff between opt and max P values during user index(Dataset #4)

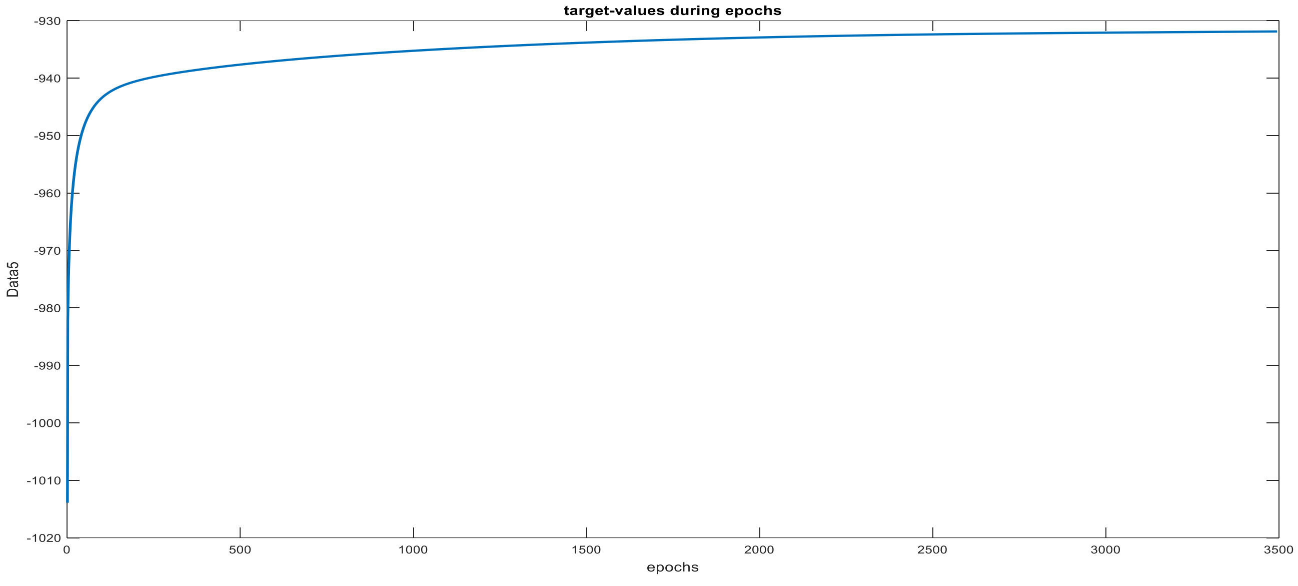


Figure 15. function-values during epochs(Dataset #5)

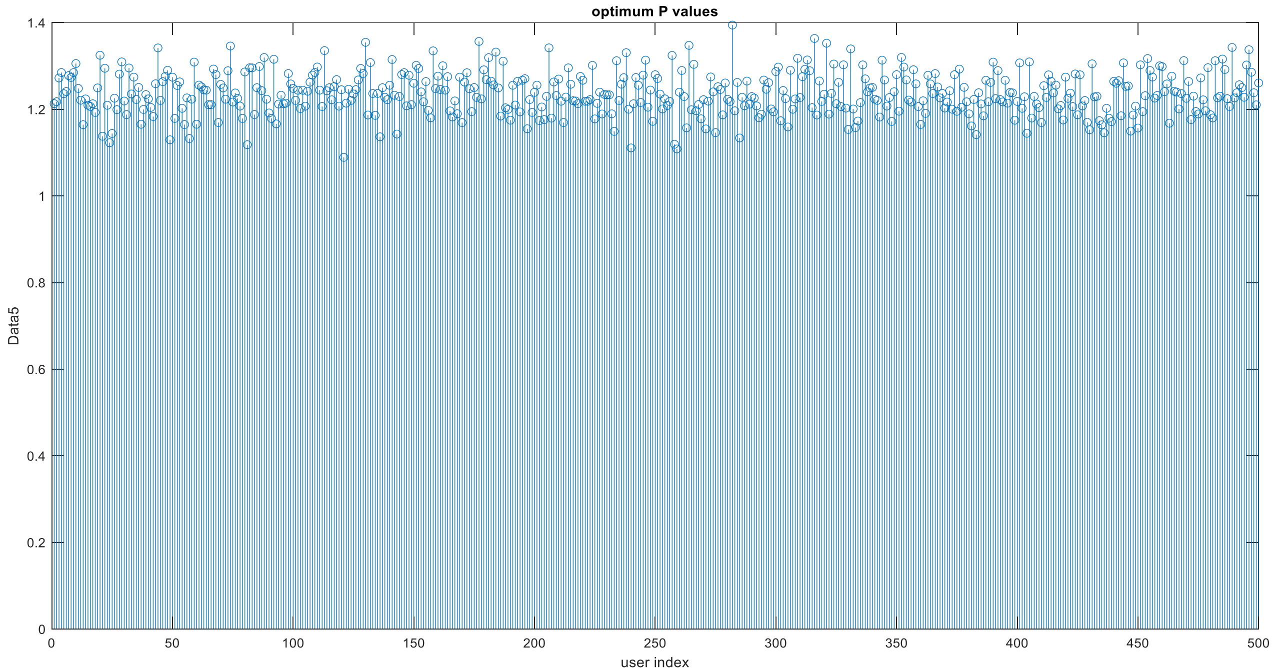


Figure 16. optimum P values during user index(Dataset #5)

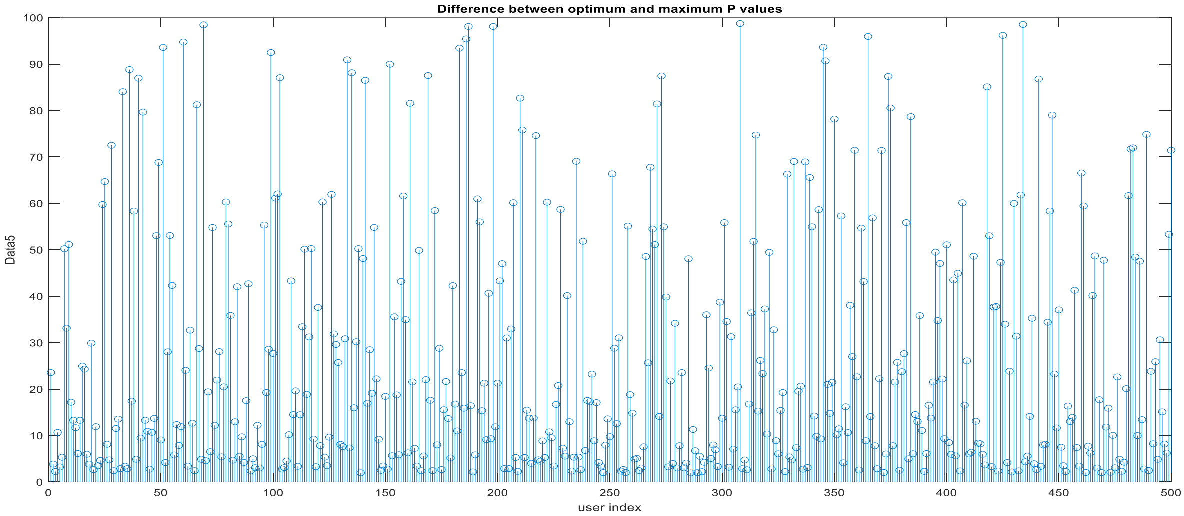


Figure 17. diff between opt and max P values during user index(Dataset #5)

```
Command Window
>> untitled
f-optimum1: -84.6439
f-max1: -92.5384
improvement1: 7.8945
Save_Power1: 1143.5912
Iteratation_Num1: 4433
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
f-optimum2: -41.5764
f-max2: -48.8192
improvement2: 7.2429
Save_Power2: 900.3418
Iteratation_Num2: 35480
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
f-optimum3: -18.9034
f-max3: -25.7855
improvement3: 6.8821
Save_Power3: 1193.3562
Iteratation_Num3: 24003
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
f-optimum4: -1.2624
f-max4: 0.68612
improvement4: -1.9485
Save_Power4: 1038.7446
Iteratation_Num4: 56966
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
f-optimum5: -931.8648
f-max5: -1031.2882
improvement5: 99.4234
Save_Power5: 13205.5902
Iteratation_Num5: 3494
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Figure 18. optimization-results

همانطور که دربالا مشاهده میشود تمامی خواسته های سوال درقسمت ب دراینجا ضمیمه شد.

(مقادیر تعداد ایپاک ها برای همگرایی، مقادیر بهینه و ماکسیمم، مقدار بهبود عملکرد، مقدارتوان صرفه جویی شده درهر حالت مدل سازی کانال و همچنین منحنی های خواسته شده تماما ضمیمه شده اند.)

درارابطه با تحلیل منحنی ها و مقادیر بهبود درهریک از دیتاست ها و همچنین روال تغییر آنها نیز درادامه و درقسمت پ به تفصیل صحبت خواهیم کرد.

ج: تحلیل نتایج و پاسخ به سوالات:

دراین قسمت قصدداریم تا با بررسی دقیق دیتاست های داده شده و همچنین روال تغییر منحنی ها و دیتاهای بدست آمده درقسمت قبل هدف از این پروژه را بیابیم و در رابطه با تفاوت های مدلسازی های مختلف کانال ها که دردیتاست های مختلف نهفته شده اند توضیحاتی را ارایه دهیم:

1)

درگام اول برای بررسی تفاوت دیتاست های داده شده بردار نویزها را برای هریک از دیتاست های داده شده بررسی خواهم کرد:

بعد از یک بررسی مختصر و رسم منحنی های توان نویزها برای دیتاست های مختلف به این نتیجه خواهیم رسید که بطور تقریبی این بردارها ازلحاظ اندازه دریک رنج قرار دارند و دراین زمینه تمامی دیتاست های مفروض نویز مشابهی را متحمل خواهند شد.

لذا نتیجه خواهیم گرفت که علت تفاوت نتایج بدست آمده درقسمت های قبل نمی تواند ناشی از تفاوت در نویزهای دیتاست های مختلف باشد!

2)

حال به بررسی بردارهای G داده شده برای 4 دیتاست اول خواهیم پرداخت:

بعد از یک بررسی نسبتا مختصر خواهیم دانست که با حرکت از دیتاست اول به سمت دیتاست 4 ام به مرور بهره های واقع روی قطر فرعی نسبت به قطر اصلی کوچک و کوچک تر میشوند.

همانطور که میدانیم قطراصلی این ماتریس بیانگر و سازنده بخش مطلوب از SINR خواهد بود که با یک بررسی مختصر در دیتاست های مختلف متوجه خواهیم شد که مقدار بهره های قطر اصلی بطور تقریبی برای همه دیتاست ها یکسان و رنج اعداد آن درحد 1 خواهد بود.

همچنین اینکه باحرکت ازدیتاست1 به سمت دیتاست 4 اعداد قطرفرعی کوچک و کوچک تر خواهند شد مادامی که در دیتاست 4 این کوچک شدن به کمترین مقدار خود خواهد رسید و اعدادی در رنج

1e-4 برروی قطر فرعی باقی خواهند ماند که عملا معرف این حقیقت هستند که ISI بسیار کوچک و درحد صفر خواهد بود و لذا در مخرج کسر SINR دیگر هیچ ترم ISI تاثیر نخواهد گذاشت که به منزله تبدیل شدن SINR به SNR خواهد بود :

SINR ---------- > SNR

همچنین اگر به روال تغییر بهبود ها در نتایج قسمت قبل دقت کنیم متوجه میشویم که مقدار improvement از دیتاست 1 تا 4 بصورت نزولی و کاهشی است تا اینکه در دیتاست 4 ام این مقدار منفی میشود که به معنای این است که بطور تقریبی در دیتاست چهارم عملا ما به بردار ماکسیمم همگرا شده ایم و اختلاف f های ناشی از p-optimum و p-max بسیار ناچیز و درحد صفر است لذا هیچ بهبودی دراین حالت نخواهیم داشت.

توجه شود که به دلیل تریکی که برای اعمال قیود دراین مسئله استفاده کرده ایم انتظار نداریم که دقیقا به بردار p-max داده شده برای این دیتاست برسیم و علت صفر نشدن improvement در دیتاست 4 ام این امر خواهد بود.

پس درقسمت اول سوال و واضحا روال متغیرimprovement یک روال نزولی است زیرا هرچه ISI کاهش بیشتری یابد SINR به SNR بیشتر میل میکند و میدانیم که برای داشتن ماکسیمم نرخ دریافت روی تمامی کاربرها با ثابت بودن توان نویز میبایست صورت کسر یا همان توان های ارسالی را با بیشترین مقدار عملی ممکن در BS ارسال نماییم که این خود دلیلی بر یکسان شدن مقادیر p-optimum و p-max در دیتاست چهارم میباشد.(عملا درمنحنی مربوط به اختلاف p-opt و p-max تعداد بیشتری صفر موجود میباشد که معرف این است که بطور میانگین این حالت بیشترین شباهت را به مقدار ماکسیمم خود دارد.)

عملا با حرکت به سمت دیتاست چهارم فاصله بین f-opt و f-max کمتر و کمتر خواهد شد تا این 2 مقدار کاملا یکسان شوند که تابع بهینه ساز ما نیز به وضوح و به درستی همین امر را به تصویر میکشد.

همچنین اینکه درمورد دیتاست پنجم نیز مشاهده میکنیم که علی رغم مقدار معقول ISI مقدار improvement تفاوت فاحشی با 4 مقدار اول دارد علت این قضیه نیز این مهم است که در حالت 5 ام ما از کانال های بیشتری (500 کانال) در مقایسه با 4 حالت قبلی استفاده میکنیم که با یکسان بودن مقادیر قطر اصلی آن عملا تعداد بیشتری کانال در ایجاد تداخل روی سیگنال دریافتی توسط کاربران دخیل اند که همین امر موجب میشود که ISI در مخرج باعث ایجاد تفاوت بیشتری بین مقدار بهینه و مقدار ماکسیمم شود.

توضیح دهید تحت چه شرایطی پاسخ بهینه به پاسخ ماکسیمم نزدیکتر میشود؟

همانطور که در بالا نیز بطور مفصل دراین مورد توضیح ارایه شد میدانیم اگر درایه های قطر فرعی ماتریس G تماما صفر باشند عملا ISI ای نخواهیم داشت و داریم :

$$\mathrm{SINR} = \frac{signal}{N0 + 0} = \frac{signal}{N0}$$

عملا دراین حالت مساله بهینه سازی ما به مساله ماکسیمم کردن نرخ دریافت در گیرنده ها یا همان SNR ای تبدیل خواهد شد که همگی ما میدانیم با فرض یکسان بودن توان نویز به ازای ماکسیمم توان ارسالی ممکن یا به عبارتی بازای Pi های ماکسیمم به به مقدار بهینه خود همگرا خواهد شد.