Computer Assignment #2

AmirHossein Mohammadi

Date

6/10/2022

Course title

Wireless Communication

Dr. Sabbaghian

الف)

با توجه به مسئله بهینه سازی داده شده نیاز است که ابتدا بدانیم که تابع مورد نیاز برای بهینه سازی چیست و سپس شروط آن را بنویسیم.
میزان SINR در حقیقت میزان توان سیگنال دریافتی کاربر i که توسط کانال i گین گرفته نسبت به توان نویز و تداخل توسط باقی کاربران است. در نتیجه تابع به شکل زیر در می آید (ممکن است نوتیشن اندکی متفاوت باشد).

$$f(p_i) = \sum_{1}^{N} \log\left(log\left(1 + \frac{p_i G_{ii}}{N_0 + \sum_{\substack{j=1 \\ j\neq i}}^{N} P_j G_{ij}} \right)\right)$$
$$st: \qquad 0 < p_i < p_i^{max}$$

که در بالا پارامتر ها دارای ابعاد زیر است:

$$G \rightarrow N \times N; G_{ij} \rightarrow \textit{Gain of jth channel for i th person}$$
$$p \rightarrow 1 \times N\,; p_i \rightarrow \textit{ith person}$$
$$N_0 \rightarrow 1 \times N\,; N_{0_i} \rightarrow \textit{noise for ith person}$$

با توجه به اینکه خواسته شده که به وسیله الگوریتم Gradient Ascent تابع هزینه را بهینه کنیم، و توجه به مقعر بودن تابع (وجود یک ماکسیمم سراسری) میتوان از الگوریتم GA استفاده کرد. با توجه به محدودیتی که در p وجود دارد نمیتوان از GA معمول استفاده کرد چرا که ممکن است p از بیشترین توان بیشتر یا از صفر کمتر شود درنتیجه باید با روشی ازین اتفاق جلوگیری کرد.
از روش هایی مانند interior point یا Barrier Method میتوان استفاده کرد که سرعت بالایی نیز دارند، اما چون در Barrier ها KKT conditionرا ارضا کرد وبا توجه به پیچیدگی محاسبات از آن صرف نظر شد. روش دیگری که اینجا مورد استفاده میشود همان GA است که ویژگی اضافه ای که دارد این است که تنها به p هایی اجازه update میدهد که از مقدار max بیشتر نشود و اگر از مقدار max بیشتر شد همان مقدار max را درون آن مقدار قرار میدهد، در مورد صفر ها نیز به همین شکل عمل میشود. البته اگر منفی شود مقدار صفر را نمیگذاریم بلکه مقداری اندکی بزرگتر از صفر تا مقدار Log بینهایت نشود.
برای این الگوریتم نیاز است که ابتدا مشتق تابع رو گرفته و سپس در جهت مثبت گرادیان حرکت کنیم. به دلیل مقعر بودن تابع هدف میدانیم که اگر در جهت شیب حرکت کنیم (با گام مناسب) همواره به نقطه بهینه نزدیک میشویم. مشتق را به شکل زیر حساب میکنیم:

$$\frac{\partial f(p)}{\partial p_k} = \frac{G_{kk}}{N0 + \sum_{\substack{j=1 \\ j\neq k}}^{N} p_j G_{jk}} \frac{1}{(\gamma_k + 1)} \frac{1}{log(\gamma_k + 1)} + \sum_{\substack{i=1 \\ i\neq k}}^{N} \frac{-p_i G_{ii} G_{ik}}{\left(N0_i + \sum_{\substack{j=1 \\ j\neq i}}^{N} p_j G_{ij}\right)^2} \frac{1}{(\gamma_i + 1)} \frac{1}{log(\gamma_i + 1)}$$

مشاهده میشود که دو ترم در مشتق ظاهر شده است. ترم اول به علت وجود p_k در summation داده شده است و ترم دوم به دلیل وجود p_k در مخرج summation و درون sum است. نتیجه مشتق به صورت بالا است.

تلاش **بسیاری** شده است که عبارت بالا به صورت ماتریسی پیاده سازی شود. در نتیجه ابتدا باید sum ها رو به صورت زیر در آورده شوند:

$$\frac{\partial f(p)}{\partial p_k} = \frac{G_{kk}}{N0 + \sum_{j=1}^{N} p_j G_{jk} - p_k G_{kk}} \frac{1}{(\gamma_k + 1)} \frac{1}{\log(\gamma_k + 1)} + \sum_{i=1}^{N} \frac{-p_i G_{ii} G_{ik}}{\left(N0_i + \sum_{j=1}^{N} p_j G_{ij} + p_i G_{ii}\right)^2} \frac{1}{(\gamma_i + 1)} \frac{1}{\log(\gamma_i + 1)} - \frac{-p_k G_{kk}^2}{\left(N0 \quad + \sum_{j=1}^{N} p_j G_{kj} + p_k G_{kk}\right)^2} \frac{1}{(\gamma_k + 1)} \frac{1}{\log(\gamma_k + 1)}$$

حال که از دست استثنا هایی که در summation ها ظاهر شده بودند خلاص شدیم میتوانیم به صورت ماتریسی مشتق را پیاده سازی کرد.

برای پیاده سازی فرض شده که نتیجه نهایی مشتق قرار است یک بردار سطری باشد در نتیجه برای اینکه بتوانیم عبارت دوم را پیاده سازی کنیم نیاز بود یک ماتریس دو بعدی در نظر گرفت که اگر sum را پیاده سازی کنیم کافیست روی ستون ها جمع را انجام داد(اندیس i برای هر سطر است) و در نهایت یک بردار سطری حاصل میشود که قابل قبول است.(k اندیس برای هر ستون است.) کد سعی شده خوانا زده شود و دارای comment است. تابع `derivative_cal` برای گرفتن مشتق پیاده سازی شده است.

انتخاب learning rate وابسته است به داده ای که داده شده است و برای آن مقادیر مختلفی استفاده شده که در نمودار هر کدام ذکر شده است.

شرط خاتمه ای که در نظر گرفته شده به شکل رو به رو است:

$$\left| f_{prev}(p) - f(p) \right| < 10^{-4}$$

یعنی اگر تابع هدف ما کمتر از 10^{-4} تغیر کند الگوریتم خاتمه پیدا میکند.

همچنین زمانی الگوریتم همگرا میشود که مقدار p تغییر محسوسی نکند یعنی مقدار norm(gradient) مقدار بسیار کمی داشته باشد در نتیجه f نیز تغییر نمیکند. اگر learning rate مناسب انتخاب شده باشد و با توجه به concave بودن تابع میتواند اطمینان داشت که همگرا میشود.

ب)

نمودار $f(p)$ به ازای mat فایل های متفاوت رسم شده و به شکل زیر است:

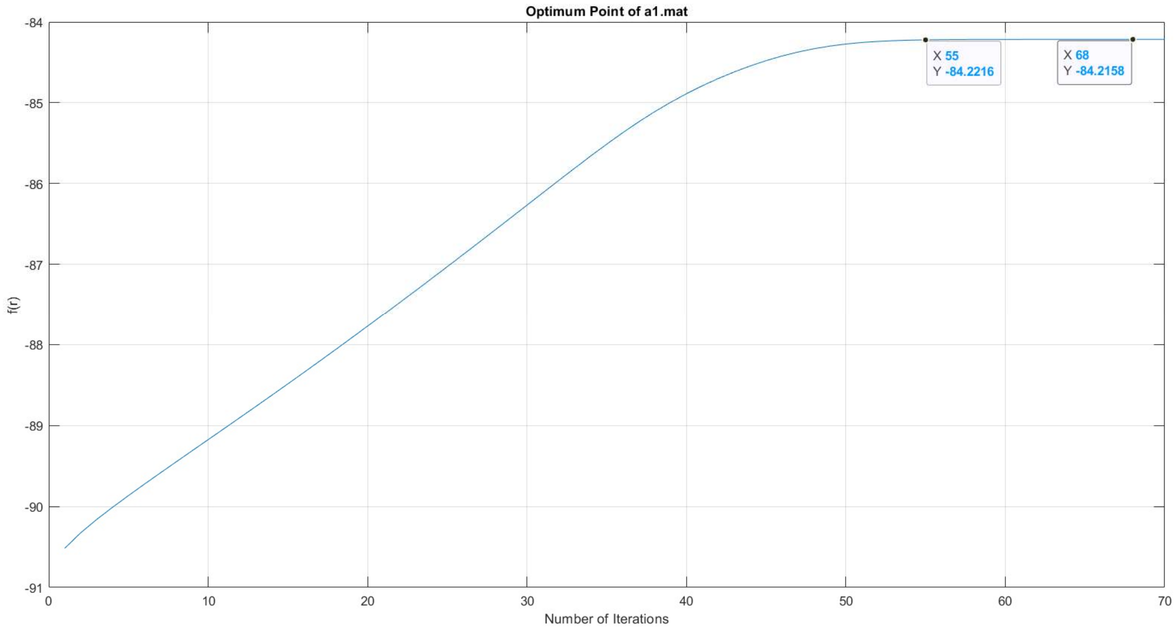


Figure1 , Values of f(r) for a1.mat

مشاهده میشود بعد از حدود 68 *iteration* به مقدار -81.215 همگرا شده است. مقدار آلفای انتخاب شده: 15

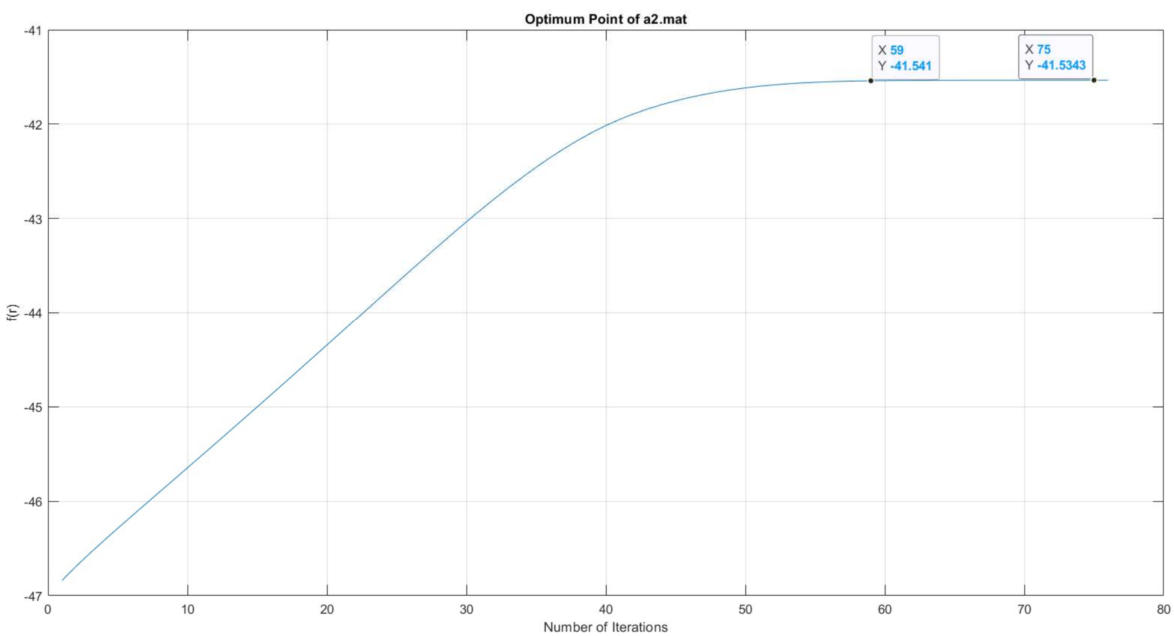


Figure , 2 Values of f(r) for a2.mat

مشاهده میشود بعد از حدود *70* *iteration* به مقدار - *41.5343* همگرا شده است. مقدار آلفای انتخاب شده:*15*

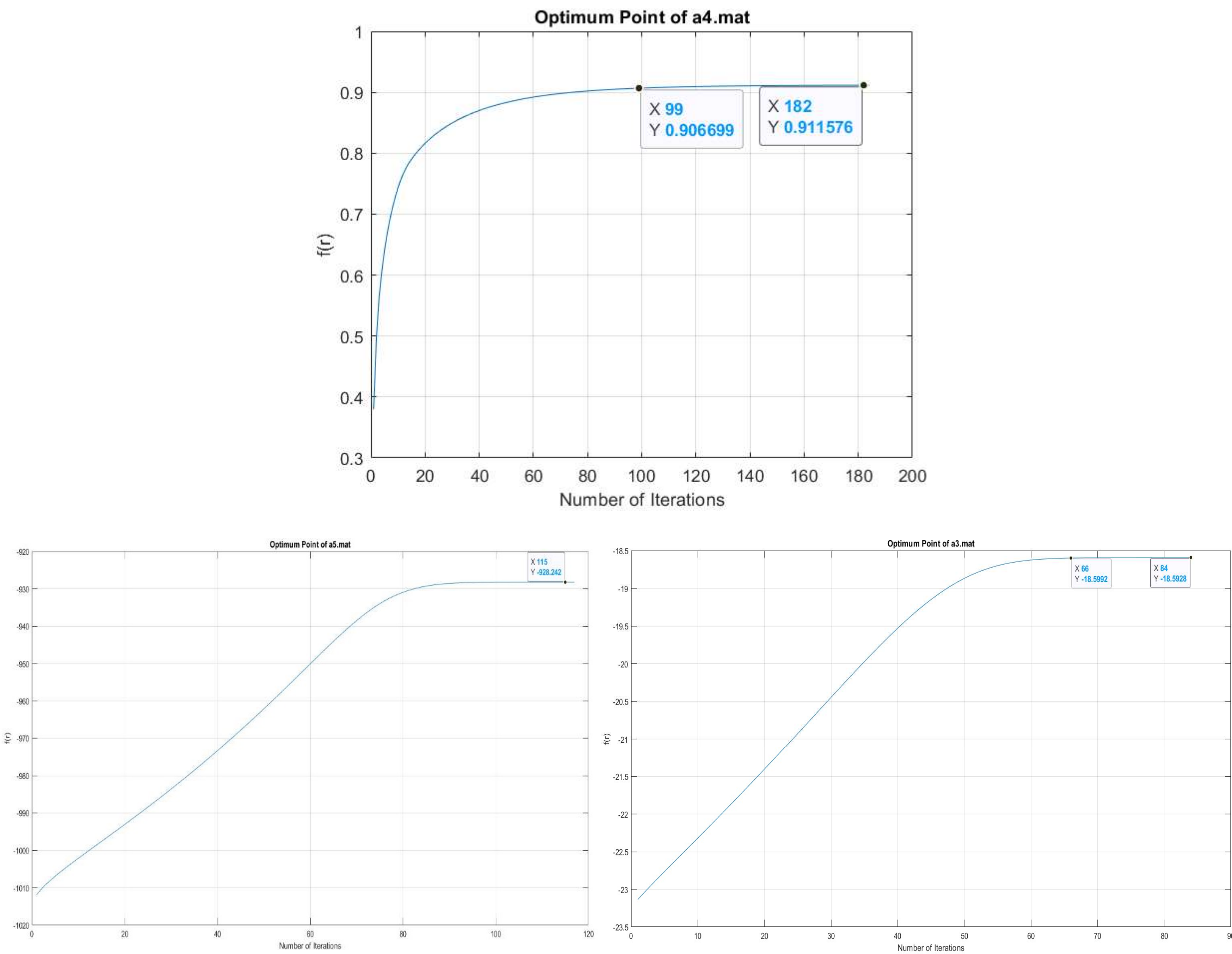


Figure4 , A3, A4, A5 mat file

همچنین برای a3.mat و a4.mat و a5.mat به ترتیب مقادیر -18.5928 و 0.9116 و -928.242 هر کدام بعد از حدود 130و 70 و 110 iteration به دست آمده است. که مقادیر آلفای آنان به ترتیب 15و 70و 10 بوده است. متوجه میشویم که فایل چهارم داری تابع flat تری است و نیاز به learning rate بیشتری دارد.

نمودار stem برای p_i ها به صورت زیر رسم شده است:

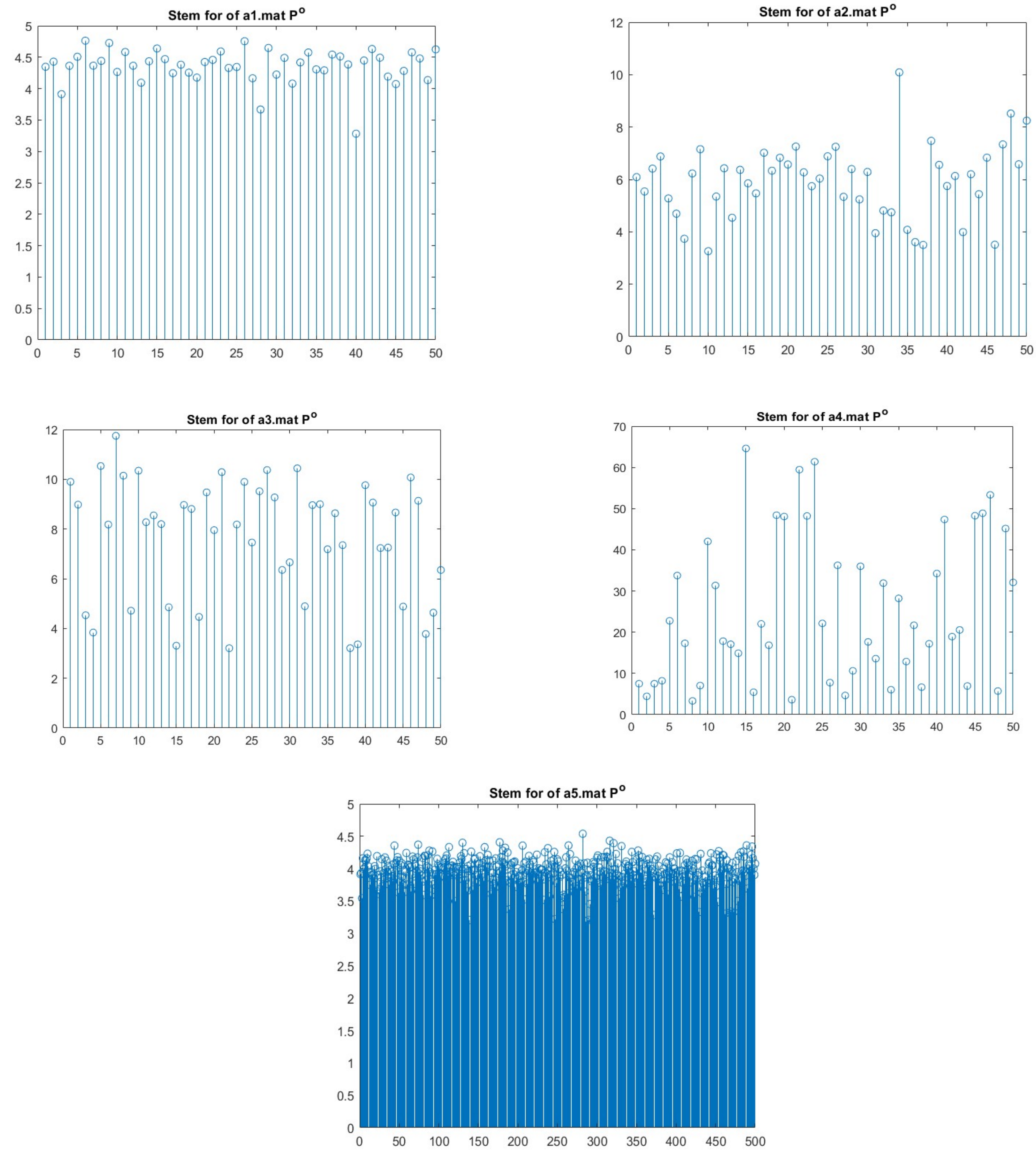
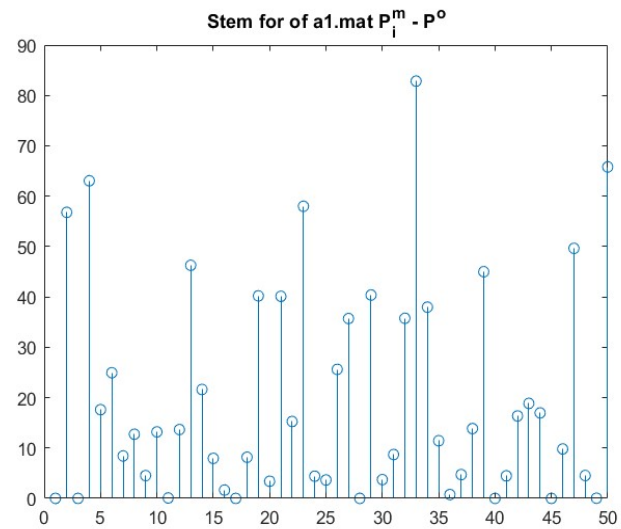
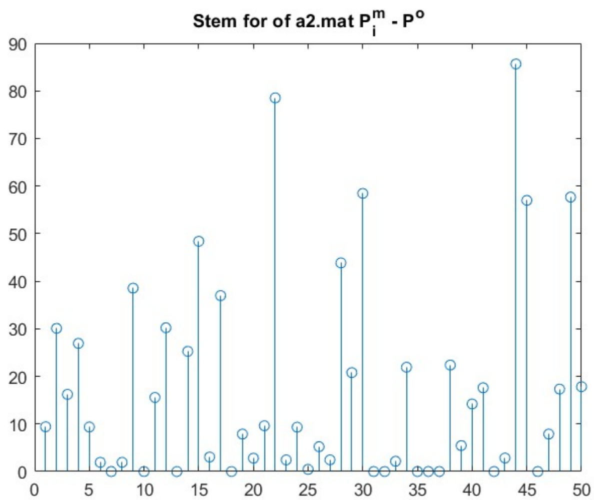
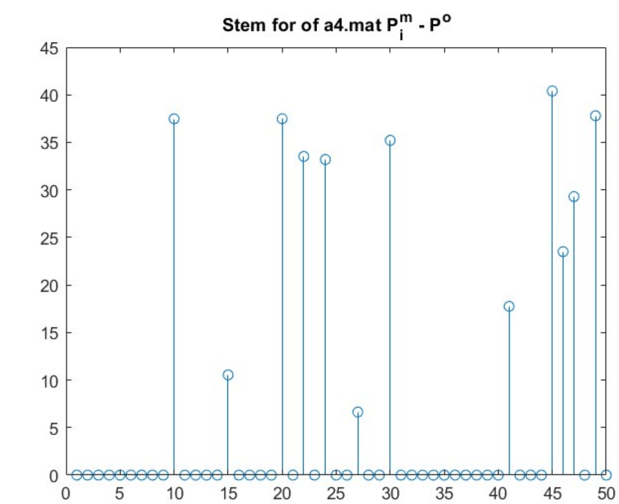


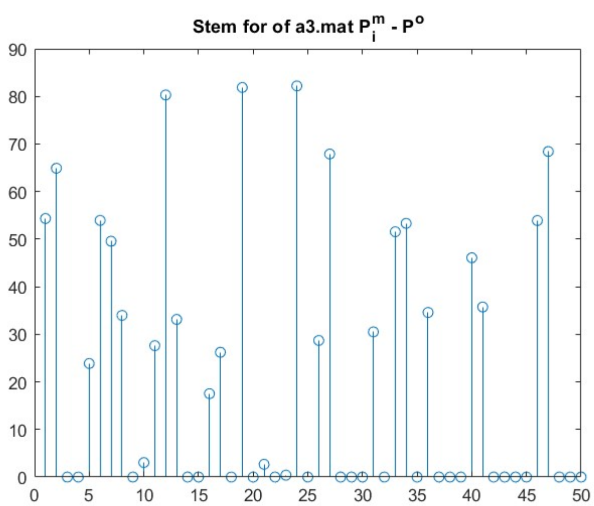
Figure5

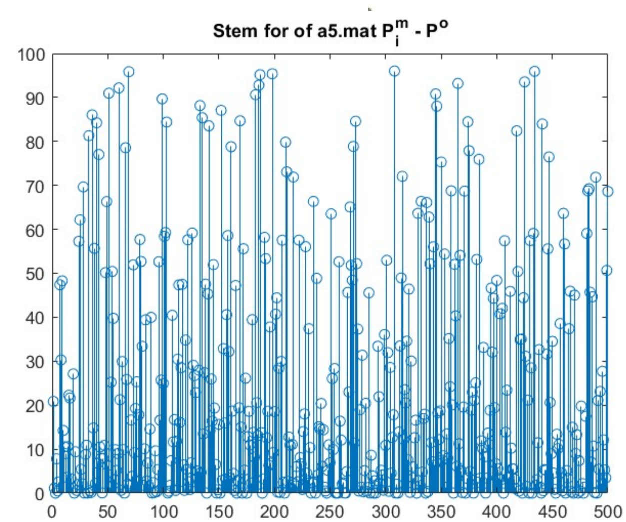
همچنین برای تفاضل ماکسیمم توان و بهترین توان داریم:











برای یافتن مقدار f افزایش یافته کد زیر نوشته شده و خروجی نیز به همراه آن ضمیمه شده است.

```
f_max = f_r(P_max);
f_optimum = f_r(p);
increase = f_optimum - f_max;
saved_power = sum(P_max - p);
sprintf("We have %.4f Imporvement for Optimum value for a%d.mat and also %.2f Power has been saved"...
,increase, j, saved_power);
sprintf("We have %.4f Imporvement for Optimum value for a%d.mat",increase, j)
```

```
We have 8.3223 Imporvement for Optimum value for a1.mat and also 998.62 Power has been saved
We have 7.2847 Imporvement for Optimum value for a2.mat and also 865.60 Power has been saved
We have 7.1923 Imporvement for Optimum value for a3.mat and also 1106.29 Power has been saved
We have 103.0464 Imporvement for Optimum value for a5.mat and also 11844.24 Power has been saved
We have 0.2061 Imporvement for Optimum value for a4.mat and also 337.66 Power has been saved
```

ج)

با توجه به اینکه scale مقادیر f(r) متفاوت است میتوان متوجه شد که سرعت همگرایی آنان متفاوت است، به طور مثال A4 مقادیر کمتری میگیرد و تغییراتی بسیار کمتر و در نتیجه گرادیان کمتری دارد.

میتوان متوجه شد که در دیتای 4 و کمی در دیتای 3 مقادیر توان آپتیمم بسیار نزدیک به توان ماکزیمم است، میتوان حدس زد که به احتمال زیاد تداخل کمتری دارد که میتوان آزادانه به هر میزان توانی که خواست فرستاد و بر عکس برای دیتا هایی که مقادیر بسیار کمتر از توان ماکسیمم میفرستیم میتوان دریافت که ماتریس G مقادیری غیر قطری بسیار بزرگتری دارد.

درنتیجه توان ذخیره شده کمتری دارد. همچنین در P_5 میتوان مشاهده کرد که فضای جست و جوی بزرگ تر است، که میتواند برای جست و جوی کار را سخت تر کند.