

شبیه سازی رایانه ای در فیزیک

تمرین ۱۰

علی ستاره کوب
شماره دانشجویی: ۹۵۱۰۰۴۹۱

۳۰ دی ۱۳۹۹

۱ مقدمه

در این تمرین می خواهیم دینامیک یک بازی را بر روی یک شبکه $N * N$ بررسی کنیم.

۲ کد شبیه سازی

در ابتدا یک کلاس بازیکن (player) می سازیم. این کلاس شامل ویژگی ها و توابعی است که هر بازیکن در این بازی دارا می باشد. از آنجا که بازی بر روی یک شبکه اتفاق می افتد پس هر بازیکن یک مختصه بر روی صفحه ی دوبعدی دارد که برای ساخت بازیکن لازم می باشد. ویژگی دیگری که بازیکن دارا می باشد، یک ماتریس سود می باشد که با عنوان pay off مات به تابع سازنده ی بازیکن در کلاس player داده می شود. علاوه بر این ها برای هر بازیکن یک متغیر با عنوان reward می سازیم که در آن سود آن بازیکن را نگه داری می کنیم. هر بازیکن در این بازی می تواند یکی از دو حالت همکاری 1 و عدم همکاری را به خود بگیرد که این ویژگی نیز در متغیر state برای هر بازیکن تعریف شده است. طبق صورت سوال، در ابتدای بازی، بازیکنان حالت خود را بصورت تصادفی انتخاب می کنند، بنابراین به هنگام ساخت یک بازیکن، حالت آن بصورت تصادفی میان صفر و یک انتخاب می شود. حال که تابع ساخت بازیکن را توضیح دادیم، به سراغ سایر توابع دیگر این کلاس می رویم.

تابع comparison تابعی می باشد که وظیفه دارد تا سود بازیکن مورد نظر را با یک بازیکن دیگر در شبکه بصورت تصادفی مقایسه کند. برای این کار اگر سود بازیکن از بازیکن تصادفی انتخاب شده کمتر بود، بازیکن حالتش را به حالت بازیکن با سود بیشتر می دهد و اگر این گونه نبود، با احتمال exp این تغییر حالت صورت می گیرد.

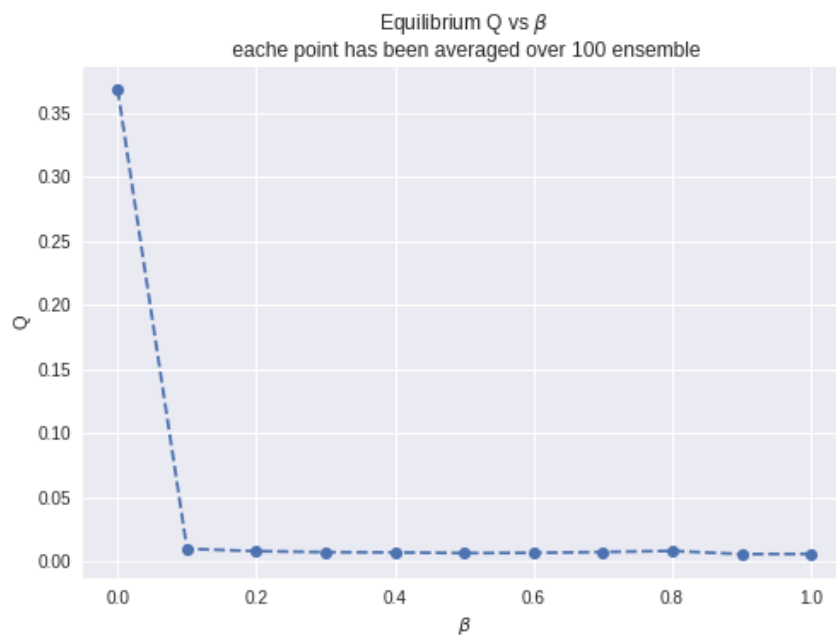
تابع mutation وظیفه ی جهش را بر عهده دارد. جهش با احتمال p رخ می دهد و طبق گفته

ی صورت سوال بدین گونه تعریف می شود که هر بازیکن فارغ از آنکه با چه بازیکنی خود را مقایسه می کند، حالتش را به حالتی مخالف حالت فعلی خود تبدیل می کند. تابع punish تابعی است که وظیفه دارد بازیکنی را عدم همکاری می کند (حالتش صفر می باشد) را مجازات کند. این مجازات بدین صورت رخ می دهد که تمام سود این بازیکن از آن گرفته می شود و همچنین ماتریس سود او نیز به ماتریس صفر تغییر می کند. تا این جای کار کلاس player را توصیف کردیم. حال به توابع اصلی این برنامه می پردازیم. تابع initialize وظیفه دارد یک شبکه یه شکل $N * N$ که هر عضو آن یک شی از کلاس player را می باشد را بسازد. تابع time step تابعی که برنامه را یک گام به جلو می برد. در این یک گام تمام بازیکنان باید با یکدیگر بازی کنند و در آخر با مقایسه خود با یک عضو تصادفی به شیوه ای که توضیح داده شد، استراتژی خود در گام بعد را تعیین کنند. تابع game تابعی که دو متغیر از جنس بازیکن را می گیرد و اجازه می دهد که این دو بازیکن با یکدیگر بازی کنند و پس از بازی، متغیر reward هر دو بازیکن با توجه به سودی که بدست آورده اند بروزرسانی می شود. تابع comp mutation تابعی است که در آن به هر بازیکن این اجازه می دهد که سود خود را با یک بازیکن دیگر در شبکه بصورت تصادفی مقایسه کند و در نهایت حالت جدید هر بازیکن تعیین شود. در این تابع پارامتر Q که در صورت سوال تعریف شده است محاسبه می شود. تابع zero reward نیز برای آن است که پس از اتمام هر مرحله سود تمام بازیکنان را صفر کند چون در اینجا تنها سود هر بازیکن در هر مرحله است که با یکدیگر باید مقایسه کنیم. توابع animate و state mat توابع کمکی می باشند که اولی برای درست کردن انیمیشن شبکه بکار برده می شود و دومی حالت تمام بازیکنان را به فرم ماتریسی می دهد. این نمایی کلی از کد این تمرین بود. حال به سراغ گرفتن داده و رسم نمودار ها از این کد می رویم.

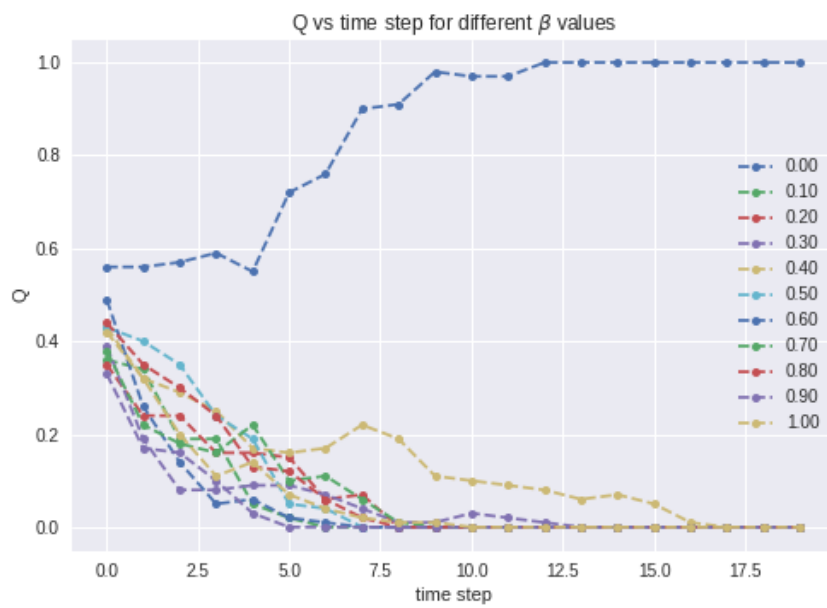
۳ سوال ۱

برای سوال اول، پارامتر بتا را برابر 1 قرار می دهیم و چون در این قسمت جهش نداریم احتمال جهش (p) را نیز برابر صفر می گذاریم و برنامه را برای بیست گام زمانی و برای بتا های مختلف اجرا می کنیم. برای آنکه آمارمان بهتر شود برای هر بتا بر روی بیست آنسامبل برای هر نقطه داده گیری را انجام می دهیم و روی این اعداد میانگین گیری انجام می دهیم و نمودار Q بر حسب زمان را برای β های مختلف رسم می کنیم. همچنین نمودار Q تعادلی بر حسب β را نیز رسم می کنیم. همانطور که انتظار داشتیم برای تمام β ها غیر از $\beta = 0$ در نهایت تمام بازیکنان استراتژی عدم همکاری را انتخاب می کنند. به ازای $\beta = 0$ در برخی از اجرا ها سیستم به حالت 1 و در برخی دیگر سیستم به حالت 0 می رسد. علت این امر آن است که در این حالت عبارت

exp برابر صفر می شود و در نتیجه همواره تغییر حالت برای بازیکنان رخ می دهد و در نتیجه این شانس را به سیستم می دهد که اگر سود بازیکنان همکاری کننده بیشتر باشد به مرور آن بازیکنانی که استراتژی آنها عدم همکاری است با احتمال یک تغییر حالت دهد و در نهایت تمام بازیکنان به استراتژی همکاری روی آورند.



[الف]



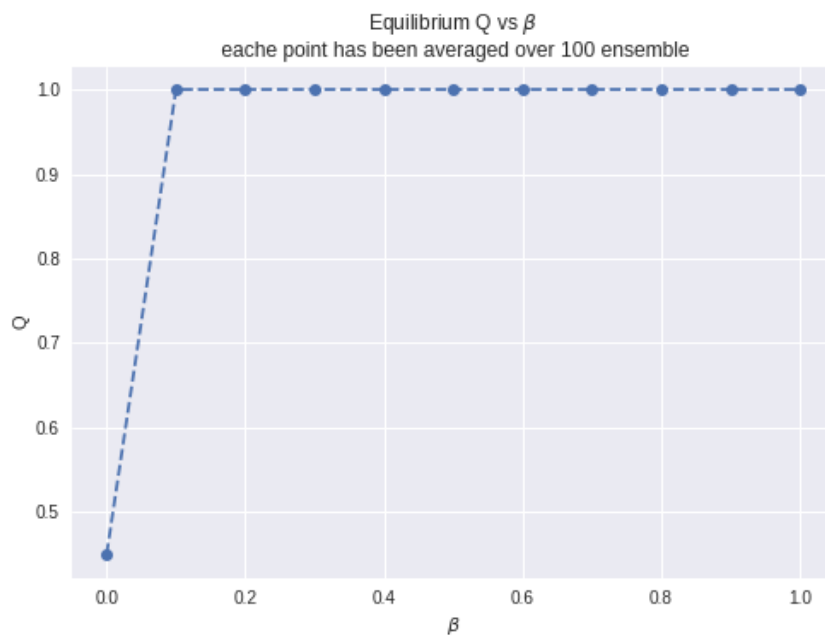
[ب]

شکل ۱: الف) نمودار Q تعادلی بر حسب β . ب) نمودار Q بر حسب زمان

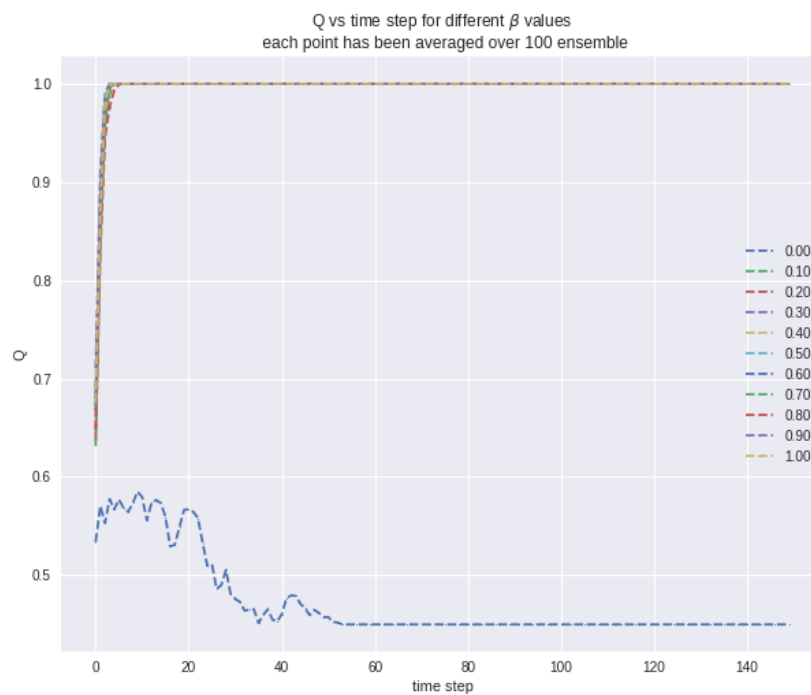
۴ سوال ۲

اگر ماتریس هزینه فایده را بصورت زیر تغییر دهیم، حالت تعادلی بازی به همکاری تغییر می کند و اگر مجدداً نمودار های قسمت قبل را رسم کنیم درستی این مساله را مشاهده می کنیم. همانطور که مشاهده می کنید در این حالت Q تعادلی به جز برای حالت $\beta = 0$ برابر 1 می باشد. علت 1 نبودن تعادل برای $\beta = 0$ مشابه قسمت قبل می باشد.

$$M = \begin{bmatrix} 2 & 5 \\ 0 & 12 \end{bmatrix}$$



[الف]

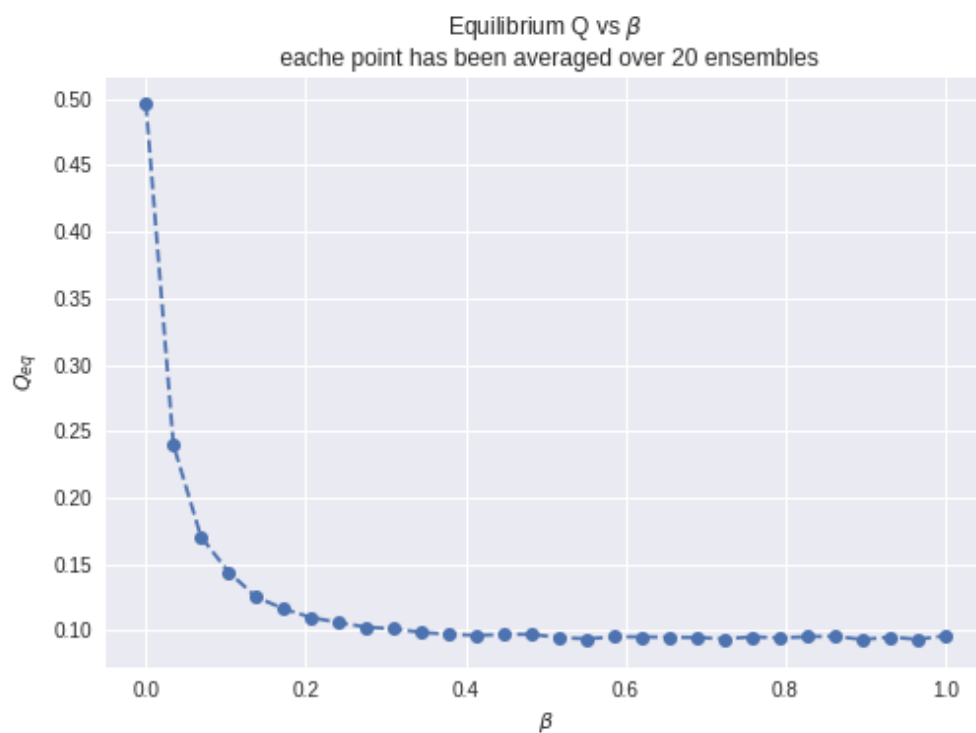


[ب]

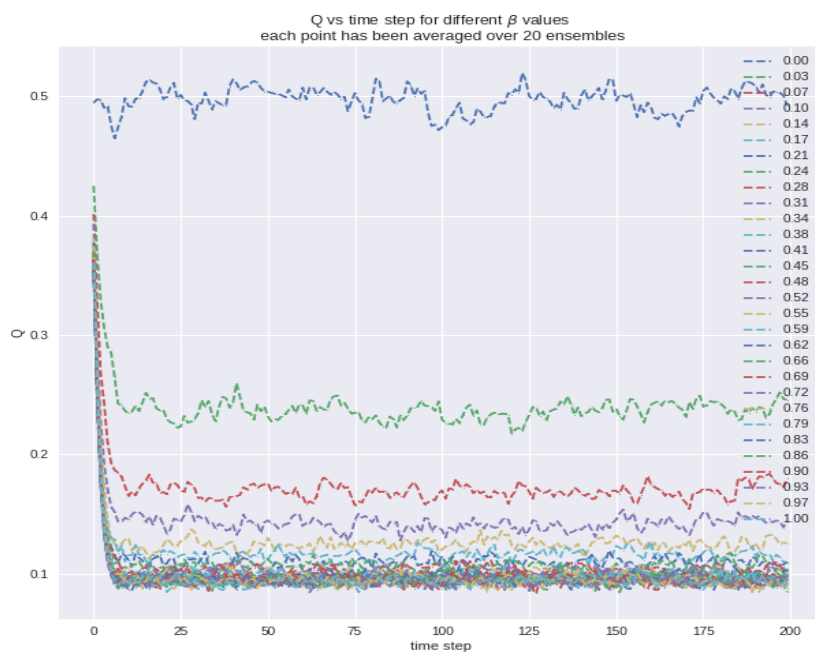
شکل ۲: الف) نمودار Q تعادلی بر حسب β . ب) نمودار Q بر حسب زمان

۵ سوال ۳

در این قسمت احتمال جهش نیز وجود دارد بنابراین کافی است احتمال جهش که با متغیر p در برنامه وجود دارد از صفر به مقدار گفته شده در صورت سوال یعنی 0.05 تغییر دهیم و مراحل گفته شده در دو قسمت قبل را تکرار کنیم که نتایج حاصل شده را در نمودار های زیر می بینید. در این قسمت برای آنکه آمارمان قوی تر باشد، هر نقطه را برای بیست آنسامبل محاسبه می کنیم و میانگین می گیریم. همچنین برای بدست آوردن مقدار تعادلی Q پس از به تعادل رسیدن سیستم، بر روی پنجاه داده آخر میانگین گیری می کنیم.



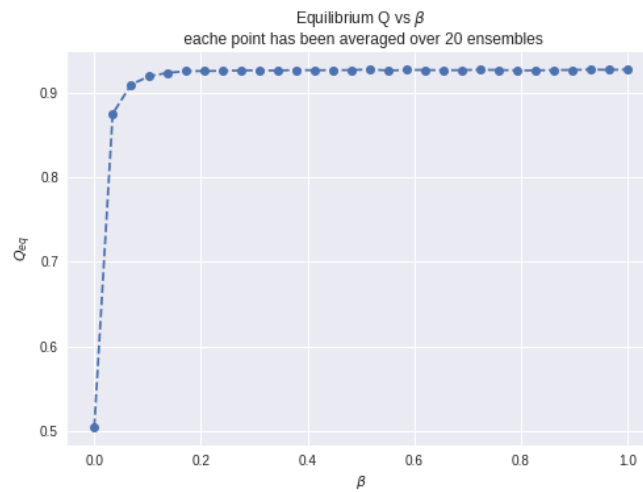
[الف]



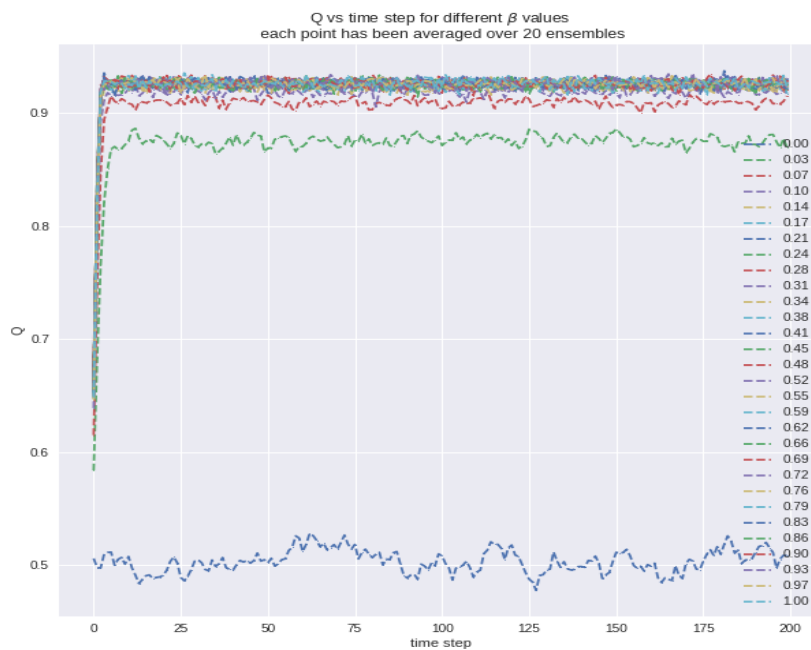
[ب]

شکل ۳: الف) نمودار Q تعادلی بر حسب β . ب) نمودار Q بر حسب زمان برای $p = 0.05$

در زیر این دو نمودار را برای حالتی که ماتریس هزینه فایده به همکاری می انجامد رسم می کنیم.



[الف]

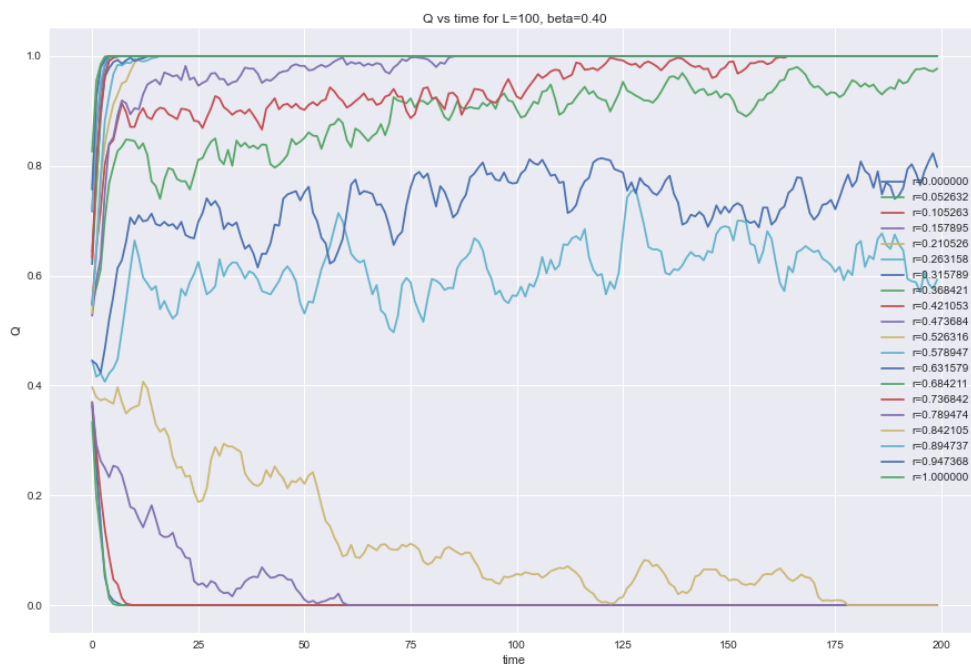


[ب]

شکل ۴: الف) نمودار Q تعادلی بر حسب β . ب) نمودار Q بر حسب زمان برای $p = 0.05$ حالت همکاری

۶ سوال ۴

در این بخش احتمال مجازات شدن بازیکنانی که عدم همکاری می کنند وجود دارد. برای این بخش کافی که لیستی از بازیکنان خطا کار درست کنیم و سود و ماتریس هزینه فایده آنها را تغییر دهیم. طبق خواسته ی سوال، هر بازیکنی که استراتژی خود را از عدم همکاری به همکاری تغییر دهد و اکنون در حال مجازات باشد، با احتمال $1 - r$ می تواند مجازات او برداشته شود. برای اجرای این کار در تابع comp mutation پس از آنکه هر بازیکن استراتژی خود را برای گام بعدی تعیین کرد، بررسی می کنیم که اگر این بازیکن در بین مجازات شوندگان بود، این شانس را به او می دهیم که از حالت مجازات خارج شود. این کار را برای مقادیر مختلف β و r انجام می دهیم و نمودار حالت تعادلی Q را برحسب این دو پارامتر رسم می کنیم. برای رسم نمودار ها فضای β را با گام های 0.1 و فضای r را با گام های 0.05 جاروب می کنیم. نمودار های کامل در فایل ارائه شده و در پوشه ارائه شده موجود می باشند که ما یکی از آنها را در اینجا آورده ایم.

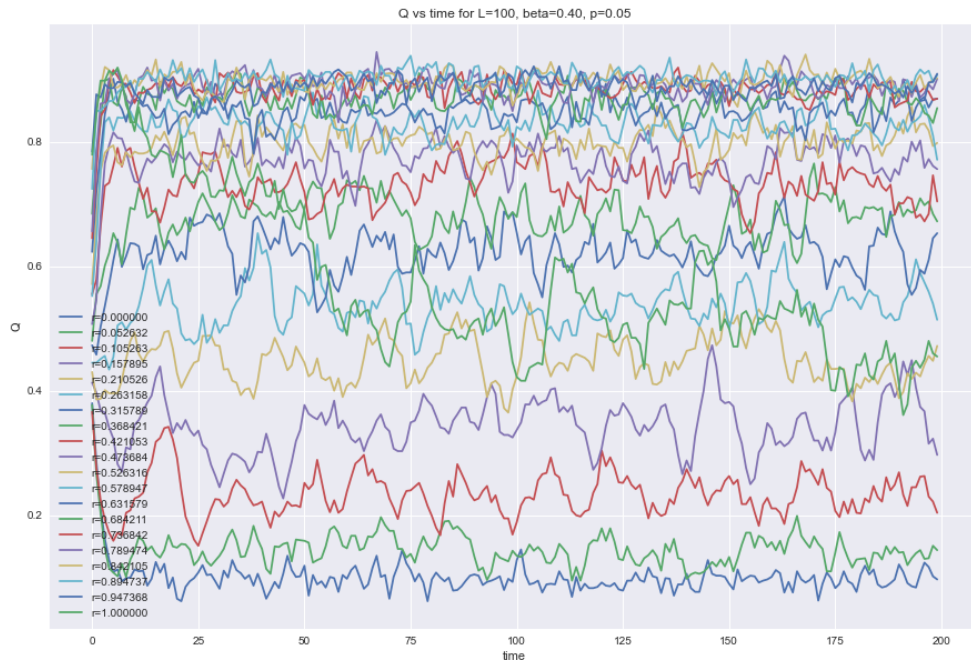


شکل ۵: نمودار Q برحسب زمان $\beta = 0.4$ برای r های مختلف

۷ سوال ۵

در این قسمت برخلاف قسمت قبل احتمال جهش را از صفر به 0.05 تغییر می دهیم و نمودار های مربوطه را رسم می کنیم. در اینجا فقط یک نمونه از نمودار های Q برحسب زمان را آورده

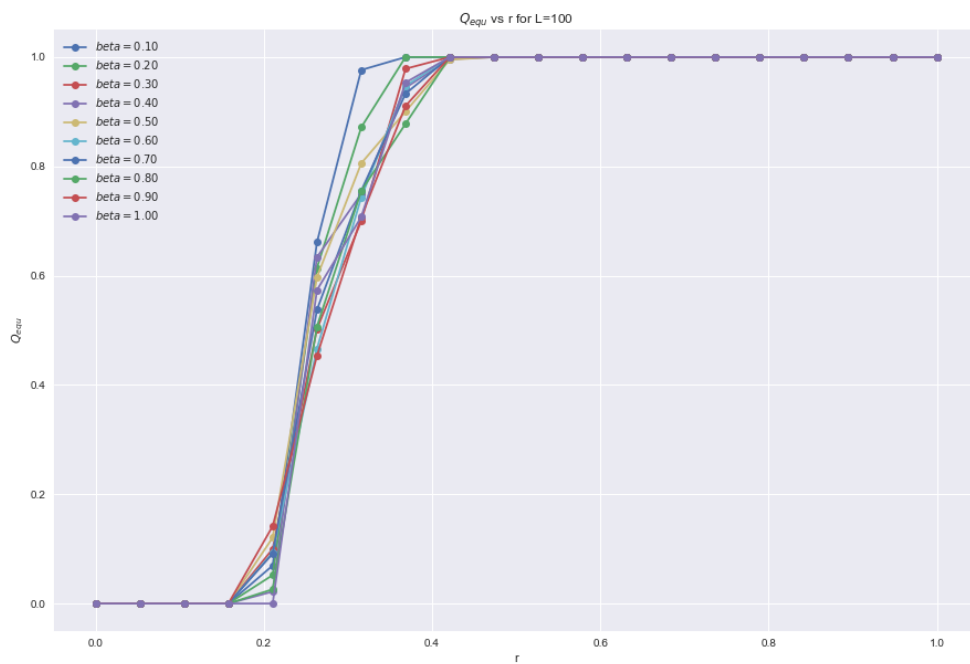
ایم اما فایل کامل در فایل ارائه شده موجود است.



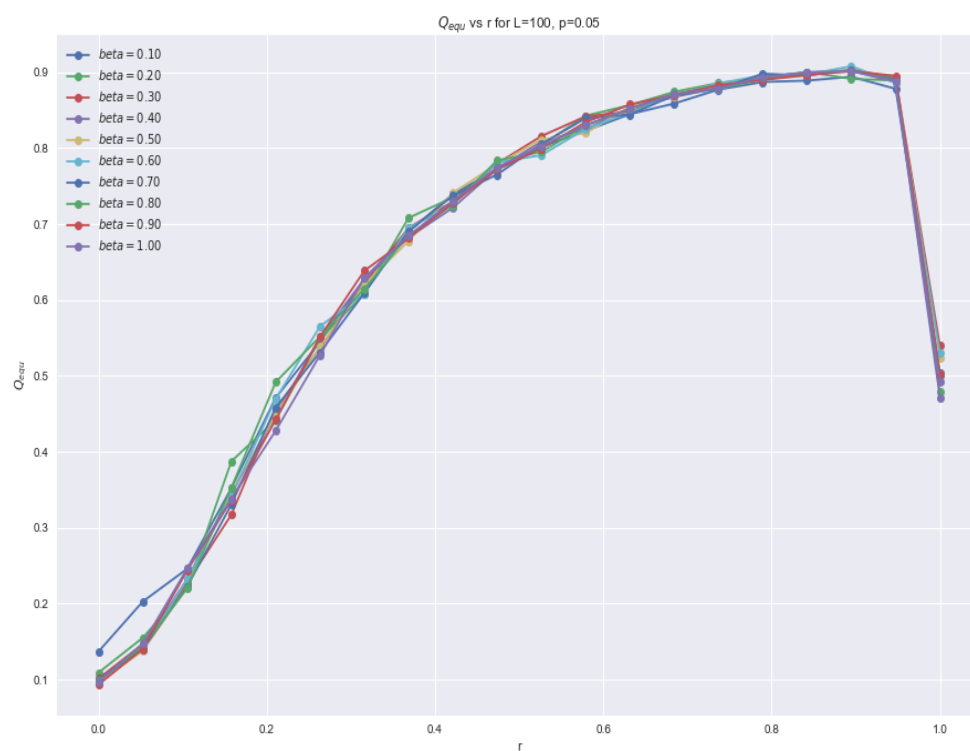
شکل ۶: نمودار Q بر حسب زمان $\beta = 0.4$ برای r های مختلف

۸ سوال ۶

برای تحلیل نتایج حاصل شده Q تعادلی را بر حسب β رسم می کنیم که در زیر می بینید. همانطور که دیده می شود، در حالتی که بازیکنان خطا نمی کنند با اعمال جریمه، در محدوده $r = 0.2$ شاهد یک تغییر فاز گسسته مانند مدل آیزینگ هستیم. اما هنگامی که بازیکنان خطا نیز می کنند، حالت تعادل بصورت پیوسته تغییر می کند.



[الف]

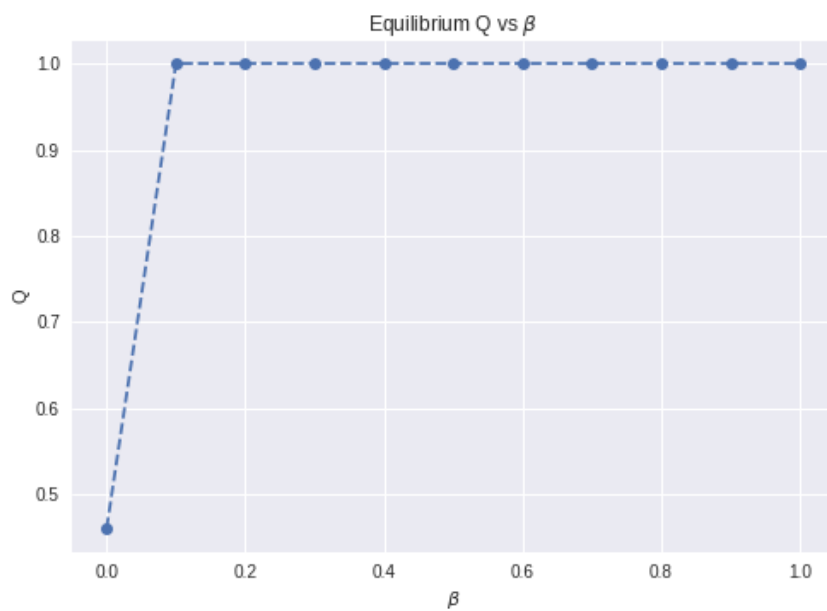


[ب]

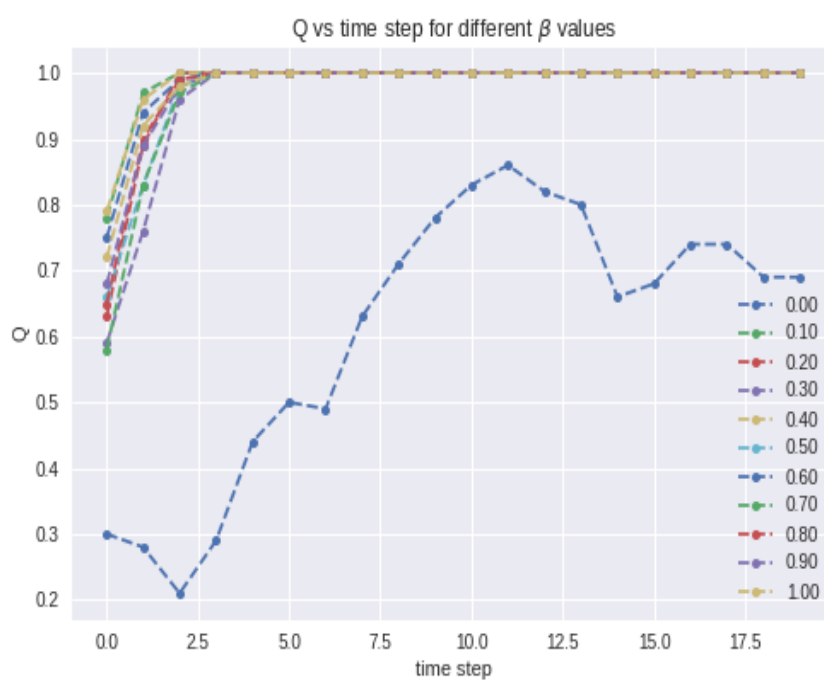
شکل ۷: الف) نمودار Q تعالی بر حسب β برای $p = 0$. نمودار Q تعادلی بر حسب β برای $p = 0.05$

۹ بخش اضافه

در قسمت های پیشین، هر بازیکن برای مقایسه خود با بازیکن دیگر، تنها سود خود در آن زمان را با سود بازیکن دیگر مقایسه می کرد. حال در اینجا می خواهیم حالتی را بررسی کنیم که بازیکنان سود تجمعی خود را با یکدیگر مقایسه کنند. کد این قسمت عمان کد قبلی است با این تفاوت که تابع zero reward را خاموش می کنیم. اطلاعات مربوط به این قسمت در پوشه ی total reward آمده است. ابتدا رفتار Q را برای حالتی که جهش نداریم و برای دو حالت همکاری و عدم همکاری بررسی می کنیم که نمودارهای مربوط را در زیر می بینید.

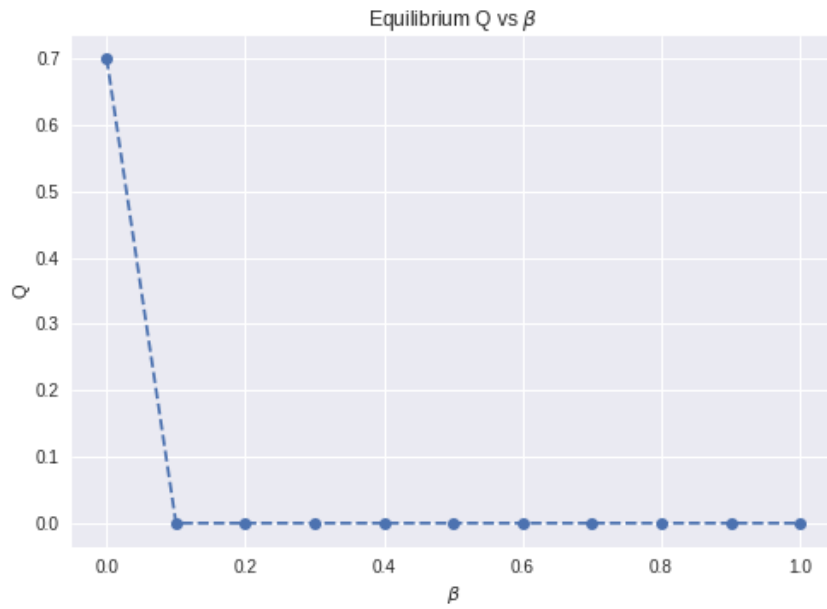


[الف]

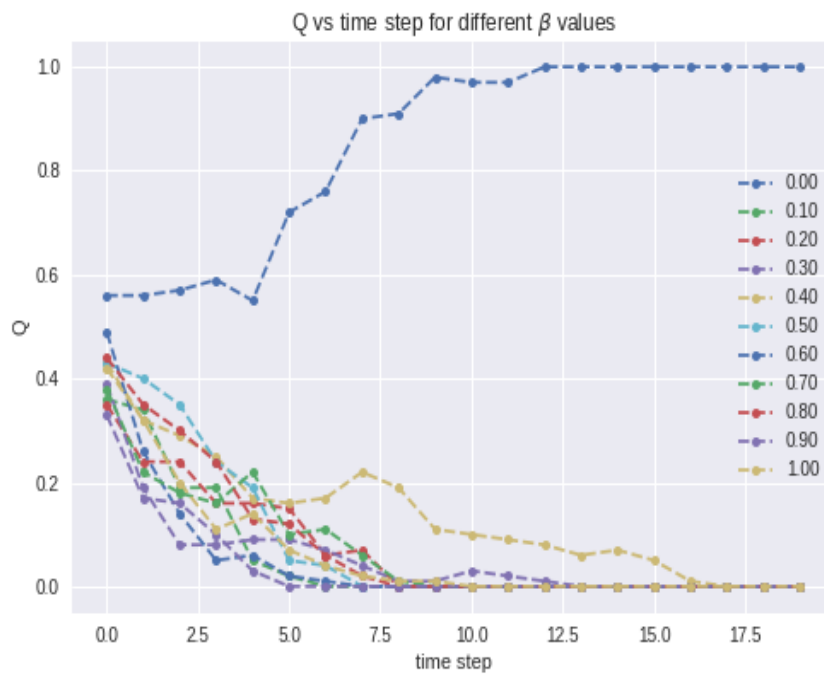


[ب]

شکل ۸:



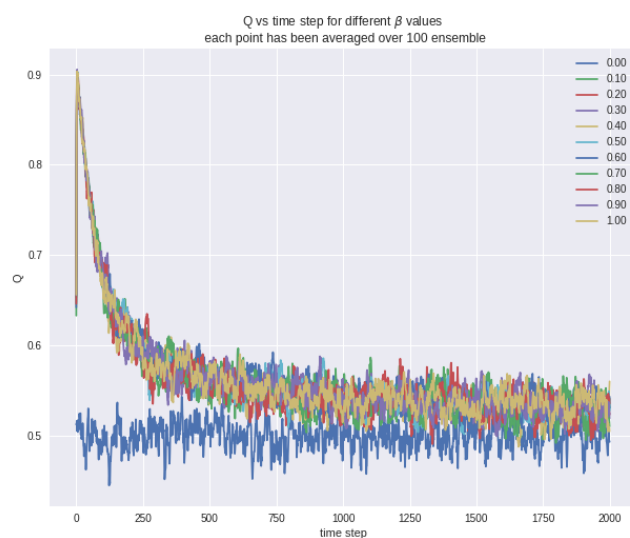
[الف]



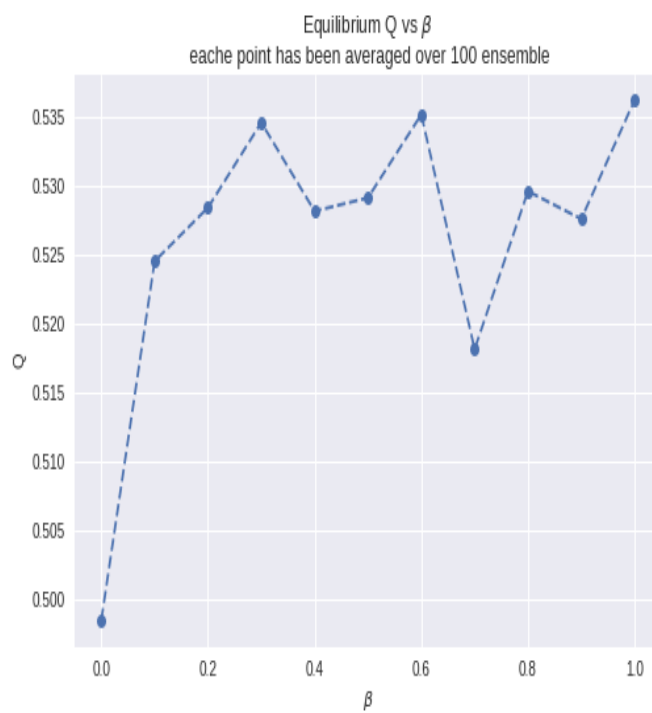
[ب]

شکل ۹:

حال احتمال خطا را اضافه می کنیم. در اینجا مانند قسمت های قبل $p = 0.05$ در نظر می گیریم. نکته ی جالبی که در اینجا وجود دارد آن است که در این حالت برخلاف حالت قبلی، رفتار تعادلی برای هر دو ماتریس همکاری و عدم همکاری یکسان می باشد و هر دو در نهایت به سمت حوالی 0.5 میل می کنند. اجرا برنامه با ماتریس همکاری:



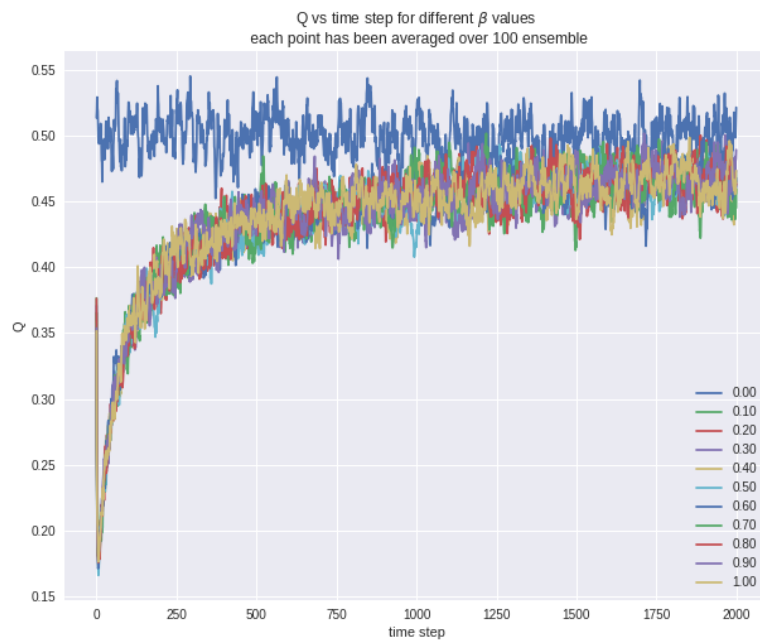
[الف]



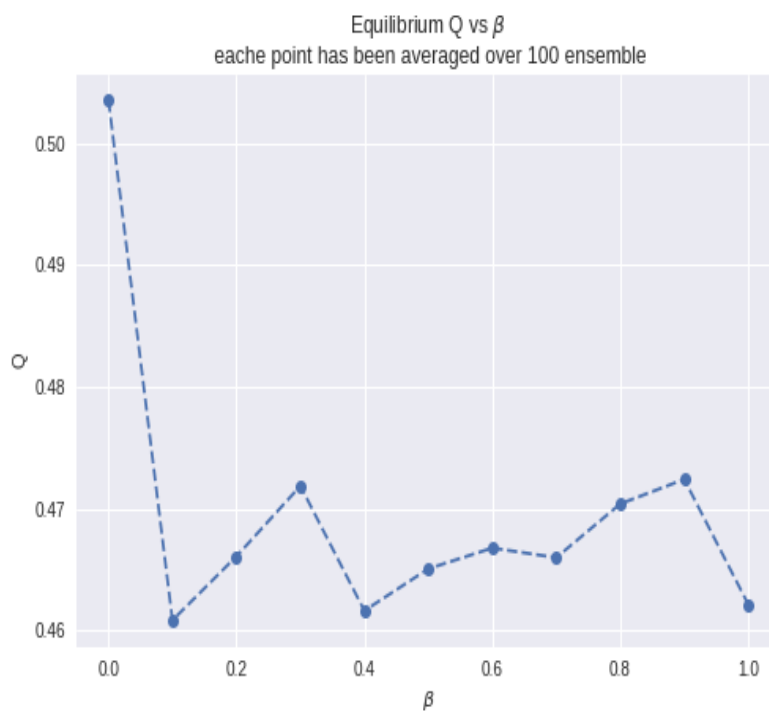
[ب]

شکل ۱۰: الف) نمودار Q بر حسب زمان برای β های مختلف. ب) نمودار Q تعادلی بر حسب β حالت همکاری

اجرای برنامه با ماتریس هزینه فایده عدم همکاری:



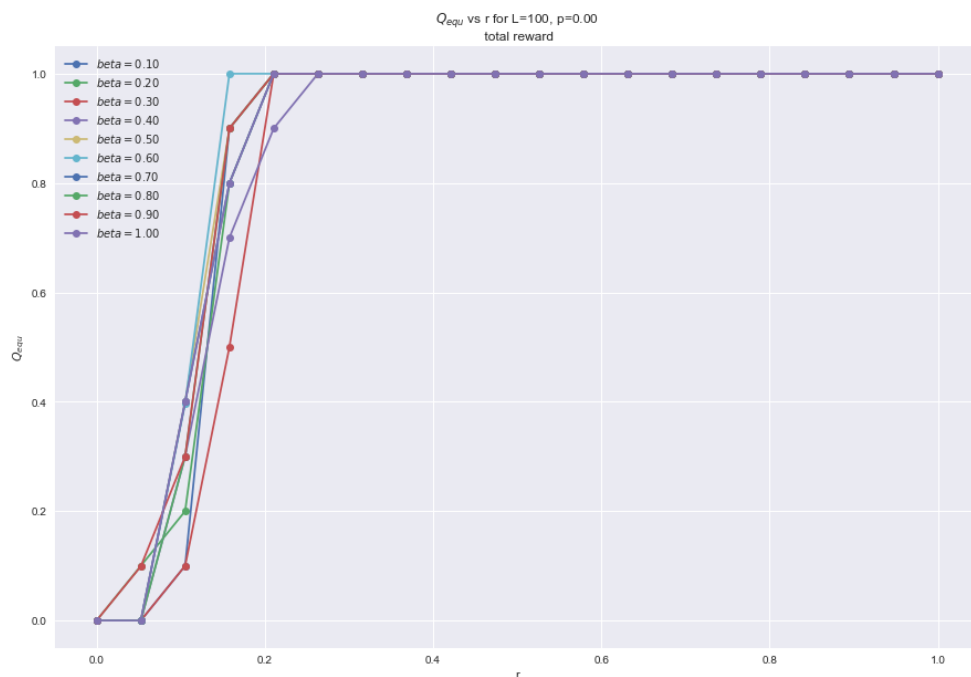
[الف]



[ب]

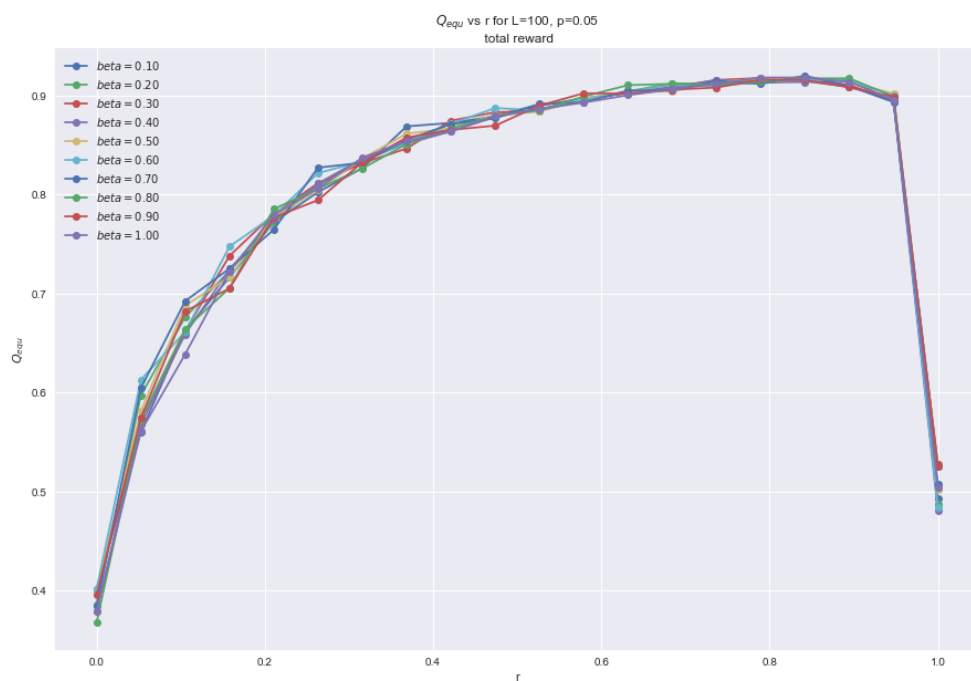
شکل ۱۱: الف) نمودار Q بر حسب زمان برای β های مختلف. ب) نمودار Q تعادلی بر حسب β حالت عدم همکاری

حال جریمه را اضافه می کنیم و احتمال خطا را صفر می کنیم. همانطور که در نمودار زیر می بینید، در این حالت نیز شاهد تغییر فاز گسسته هستیم اما این تغییر فاز در r کوچکتر اتفاق می افتد. در اینجا ما فقط نمودار حالت تعادلی Q را بر حسب β آورده ایم اگر به دنبال نمودارهای Q بر حسب زمان هستیم می توانید آنها را در فایل Q vs t در فولدر total reward بیابید.



شکل ۱۲: نمودار حالت تعادلی Q بر حسب β برای $p = 0$.

حال $p = 0.05$ را نیز اعمال می کنیم که نتیجه را در زیر می بینید:



شکل ۱۳: نمودار حالت تعادلی Q بر حسب β برای $p = 0.05$

همانطور که می بینید در هر دو حالت، هم هنگامی که فقط سود مقطعی و هم سود تجمعی را در نظر می گیریم با اعمال جریمه شاهد تغییر فاز خواهیم بود. اما هنگامی که جریمه نداریم، در این حالت برخلاف حالت قبل حتی اگر ماتریس هزینه فایده را نیز تغییر دهیم در حالت تعادلی سیستم تغییری ایجاد نمی شود.