باسمه تعالی

پروژه اول درس Distributed optimization and learning

مسیریابی با decomposition متدها

نام و نام خانوادگی : بنفشه کریمیان شماره دانشجویی: ۸۱۰۱۹۹۲۵۲ ایمیل: banafshe.karimian@gmail.com

الگوریتم مسیریابی در مسیریابی رباتها و ایجنت‌های هوشمند اهمیت زیادی دارد. متدهای دیکامپوزیشن می‌توانند برای کم کردن سربار محاسباتی، هزینه‌ی ارتباطات و افزودن به امنیت مسیریابی در محیط‌های چندعامله کمک کنند. همچنین در مسیریابی رباتها و ایجنت‌های هوشمند اگر تعداد ایجنت‌ها از حدی بیشتر شود سختی کار بر اهمیت الگوریتم مسیریابی اضافه می‌کند.

برای مدل‌سازی نیاز است ابتدا q_{ij} را به مختصات ایجنت i در زمان j تفسیر کنیم. q_i را می‌توان مسیر طی شده توسط ایجنت i معرفی کرد. در انتها q مسیر طی شده توسط همه‌ی عامل‌ها را نشان می‌دهد. اولین و مهم‌ترین هدف در مسیریابی یافتن کوتاه‌ترین راه از مبدا به مقصد است که آن را می‌توانیم با تابع زیر نمایش دهیم.

$$f(q_i) = \sum_{\text{for j in T}} \text{norm\_2}(q_{ij}.\ q_{ij-1})$$

$$\text{minimoze} \sum f(q_i)$$

محدودیت‌هایی ک می‌توان تعریف کرد به فرم زیراند:

$$\text{norm}_2\left(q_{ij}.\ q_{ij-1}\right) - V = \ 0 \text{ for velocity constraint}$$

$$\text{and } T - \sum_{\text{for j in T}} \text{norm}_2\left(q_{ij}.\ q_{kj}\right) \le \ 0 \text{ for maximum convergence}$$

$$\text{and } T - \text{norm}_2\left(q_{ij}.\ q_{kj}\right) \le \ 0 \text{ for collision avoidance}$$

$$\text{and} \sum_{\text{for j in T}} \sum_{\text{for each obstacle k}} -1 * \text{norm}_2\left(q_{ij}.\ \text{obstacle}_k\right) - T \le 0 \text{ for obstacle avoidance}$$

$$\text{or } T - \min\left(\text{norm}_2\left(q_{ij}.\ \text{obstacle}_k\right)\right) \le 0 \text{ for obstacle avoidance}$$

$$\text{and} \sum_{\text{for agent i}} \begin{matrix} \text{1 if agent reached goal 1} \\ \text{1 if agent reached goal 2} \\ \vdots \\ \text{1 if agent reached goal n} \end{matrix} = \begin{matrix} 1 \\ 1 \\ \vdots \\ 1 \end{matrix} \text{ for goals}$$

برای بخش تکمیلی و امتیازی در جلسه قرار بر این شد که یا تمام شروط بالا نوشته شود و یا دو شرط اصلی و ADMM+RL پیاده سازی شود که در ابتدا بنا بر این بود که RL استفاده شه اما چون پرایمال و دوال به دلیلی که در ادامه بررسی میکنیم امکان انجام نداشتند هر دو انجام شدن. همچنین بنا بر این شد که اگر باینری بودن شرط هدف مشکل ایجاد کرد می‌توان بجای شرط هدف و شرط سرعت، یکی دیگر از شروط و شرط سرعت را در نظر گرفت که مشکل باینری بودن حل شود که ما شرط تداخل مسیر ایجنت ها را انتخاب میکنیم.

Centralized

هدف اصلی کم ترین مسیر طی شه:

$$\min \sum f(q_i)$$

شروط:

از هم دیگه در یک زمان j از ترشلدی بیشتر فاصله داشته باشند و در هر قدم بیشتر از حدی حرکت نکنند.

$$s.t.\, T - norm_2\left(q_{ij}.\ q_{kj}\right) \le\ 0 \text{ and } norm_2\left(q_{ij}.\ q_{ij-1}\right) - V \le 0$$

برای حل باید از هر کدام نسبت به مختصات مشتق بگیریم.

$$\frac{\partial \sum f(q_i)}{\partial q_{ij}} = \frac{\partial\left(\sqrt{\left(q_{ij}[0]-q_{ij-1}[0]\right)^2+\left(q_{ij}[1]-q_{ij-1}[1]\right)^2}+\sqrt{\left(q_{ij}[0]-q_{ij+1}[0]\right)^2+\left(q_{ij}[1]-q_{ij+1}[1]\right)^2}\right)}{\partial q_{ij}}$$

$$\frac{\partial \sum f(q_i)}{\partial q_{ij}[0]} = \frac{q_{ij}[0]-q_{ij-1}[0]}{\sqrt{\left(q_{ij}[0]-q_{ij-1}[0]\right)^2+\left(q_{ij}[1]-q_{ij-1}[1]\right)^2}} + \frac{q_{ij}[0]-q_{ij+1}[0]}{\sqrt{\left(q_{ij}[0]-q_{ij+1}[0]\right)^2+\left(q_{ij}[1]-q_{ij+1}[1]\right)^2}}$$

$$\frac{\partial \sum f(q_i)}{\partial q_{ij}[1]} = \frac{q_{ij}[1]-q_{ij-1}[1]}{\sqrt{\left(q_{ij}[0]-q_{ij-1}[0]\right)^2+\left(q_{ij}[1]-q_{ij-1}[1]\right)^2}} + \frac{q_{ij}[1]-q_{ij+1}[1]}{\sqrt{\left(q_{ij}[0]-q_{ij+1}[0]\right)^2+\left(q_{ij}[1]-q_{ij+1}[1]\right)^2}}$$

$$\frac{\partial\, T - norm_2\left(q_{ij}.\ q_{kj}\right)}{\partial q_{ij}[0]} = -1 * \sum_{\text{for agent k}} \left(\frac{q_{ij}[0]-q_{kj}[0]}{\sqrt{\left(q_{ij}[0]-q_{kj}[0]\right)^2+\left(q_{ij}[1]-q_{kj}[1]\right)^2}}\right)$$

$$\frac{\partial\, T - norm_2\left(q_{ij}.\ q_{kj}\right)}{\partial q_{ij}[1]} = -1 * \sum_{\text{for agent k}} \left(\frac{q_{ij}[1]-q_{kj}[1]}{\sqrt{\left(q_{ij}[0]-q_{kj}[0]\right)^2+\left(q_{ij}[1]-q_{kj}[1]\right)^2}}\right)$$

$$\frac{\partial\, norm_2\left(q_{ij}.\ q_{ij-1}\right)-V}{\partial q_{ij}[0]} = \frac{(\lambda_{ij}+\frac{p}{2})(q_{ij}[0]-q_{ij-1}[0])}{\sqrt{\left(q_{ij}[0]-q_{ij-1}[0]\right)^2+\left(q_{ij}[1]-q_{ij-1}[1]\right)^2}} + \frac{(\lambda_{ij+1}+\frac{p}{2})(q_{ij}[0]-q_{ij+1}[0])}{\sqrt{\left(q_{ij}[0]-q_{ij+1}[0]\right)^2+\left(q_{ij}[1]-q_{ij+1}[1]\right)^2}}$$

$$\frac{\partial\, norm_2\left(q_{ij}.\ q_{ij-1}\right)-V}{\partial q_{ij}[1]} = \frac{(\lambda_{ij}+\frac{p}{2})(q_{ij}[1]-q_{ij-1}[1])}{\sqrt{\left(q_{ij}[0]-q_{ij-1}[0]\right)^2+\left(q_{ij}[1]-q_{ij-1}[1]\right)^2}} + \frac{(\lambda_{ij+1}+\frac{p}{2})(q_{ij}[1]-q_{ij+1}[1])}{\sqrt{\left(q_{ij}[0]-q_{ij+1}[0]\right)^2+\left(q_{ij}[1]-q_{ij+1}[1]\right)^2}}$$

برای حل باید سیستم نامعادلات زیر حل شه:

$$\frac{\partial \sum f(q_i)}{\partial q_{ij}} + \sum_{\text{for agent i.k and time T}} m_{ikj}\frac{\partial\, T - norm_2\left(q_{ij}.\ q_{kj}\right)}{\partial q_{ij}} + \sum_{\text{for agent i and time T}} n_{ij}\frac{\partial\, norm_2\left(q_{ij}.\ q_{ij-1}\right)-V}{\partial q_{ij}} = 0$$

$$m_{ikj}(T - norm_2\left(q_{ij}.\ q_{kj}\right)) = 0$$

$$n_{ij}\left(norm_2\left(q_{ij}.\ q_{ij-1}\right) - V\right) = 0$$

ADMM

هدف اصلی کم ترین مسیر طی شه:

$$\min \sum f(q_i)$$

شروط:

از هم دیگه در یک زمان j از ترشلدی بیشتر فاصله داشته باشند و در هر قدم بیشتر از حدی حرکت نکنند.

$$\mathrm{s.t.}\, T - \mathrm{norm}_2\left(q_{ij}.\ q_{kj}\right) \le\ 0 \text{ and } \mathrm{norm}_2\left(q_{ij}.\ q_{ij-1}\right) - V \le 0$$

برای تبدیل به فرمتی که قابل استفاده در ADMM باشد:

$$\max\left(0. T - \mathrm{norm}_2\left(q_{ij}.\ q_{kj}\right)\right) =\ 0 \text{ and } \mathrm{norm}_2\left(q_{ij}.\ q_{ij-1}\right) - V \le 0$$

حال تابع Lp را مینویسیم:

$$L_p(q.y) = \sum f(q_i) + \sum_{\text{for agent i.k and time T}} y_{ikj} \max\left(0. T - \mathrm{norm}_2\left(q_{ij}.\ q_{kj}\right)\right)$$

$$+\frac{p}{2} \sum_{\text{for agent i.k and time T}} \left\|\max\left(0. T - \mathrm{norm}_2\left(q_{ij}.\ q_{kj}\right)\right)\right\|_2^2$$

$$+ \sum_{\text{for agent i and time T}} \lambda_{ij} \max(0. \mathrm{norm}_2\left(q_{ij}.\ q_{ij-1}\right) - V) + \frac{p}{2} \sum_{\text{for agent i and time T}} \left\|\max(0. \mathrm{norm}_2\left(q_{ij}.\ q_{ij-1}\right) - V)\right\|_2^2$$

حال الگوریتم ADMM به فرم زیر میشود که در هر سری هر ایجنت مسیر خود را با توجه به مسیر بقیه به طوری که تابع را مینمم کند آپدیت میکند و سپس ضرایب شروط طبق زیر آپدیت می‌شوند:

$$q_i = \mathrm{argmin}_{qi} L_p(q.y)$$

$$y_{ikj} += p(\max\left(0. T - \mathrm{norm}_2\left(q_{ij}.\ q_{kj}\right)\right))$$

$$\lambda_{ij} += p\left(\max(0. \mathrm{norm}_2\left(q_{ij}.\ q_{ij-1}\right) - V)\right)$$

حال با توجه به اینکه نرم ۲ یک عدد مثبت خود آن است تابع را ساده میکنیم. (زیرنویس سیگماها فاکتور گرفته شدند که فرمول در دوخط جا بشه)

$$L_p(q.y) = \sum f(q_i) + \sum y_{ikj} \max\left(0. T - \mathrm{norm}_2\left(q_{ij}.\ q_{kj}\right)\right) + \frac{p}{2}\sum \max\left(0. T - \mathrm{norm}_2\left(q_{ij}.\ q_{kj}\right)\right)$$

$$+ \sum \lambda_{ij}(\max(0. \mathrm{norm}_2\left(q_{ij}.\ q_{ij-1}\right) - V)) + \frac{p}{2}\sum \max(0. \mathrm{norm}_2\left(q_{ij}.\ q_{ij-1}\right) - V)$$

$$L_p(q.y) = \sum f(q_i) + \sum (y_{ikj} + \frac{p}{2}) \max\left(0. T - \mathrm{norm}_2\left(q_{ij}.\ q_{kj}\right)\right) + \sum (\lambda_{ij} + \frac{p}{2}) \max(0. \mathrm{norm}_2\left(q_{ij}.\ q_{ij-1}\right) - V)$$

یک روش حل تابع برای هر qi این است که از تابع برای مختصات x و y هر qij مشتق گرفته برابر صفر قرار دهیم.

$$\frac{\partial L_p(q.y)}{\partial q_{ij}} = I1 + I2 + I3$$

$$I1 = \frac{\partial\left(\sqrt{\left(q_{ij}[0] - q_{ij-1}[0]\right)^2 + \left(q_{ij}[1] - q_{ij-1}[1]\right)^2} + \sqrt{\left(q_{ij}[0] - q_{ij+1}[0]\right)^2 + \left(q_{ij}[1] - q_{ij+1}[1]\right)^2}\right)}{\partial q_{ij}}$$

$$I2 = \partial \frac{\sum_{\text{for each agent k}} \left(y_{ikj} + \frac{p}{2}\right) \max\left(0. T - \mathrm{norm}_2\left(q_{ij}.\ q_{kj}\right)\right)}{\partial q_{ij}}$$

$$I3 = (\lambda_{ij} + \frac{p}{2}) \frac{\partial \sqrt{\left(q_{ij}[0] - q_{ij-1}[0]\right)^2 + \left(q_{ij}[1] - q_{ij-1}[1]\right)^2} + \sqrt{\left(q_{ij}[0] - q_{ij+1}[0]\right)^2 + \left(q_{ij}[1] - q_{ij+1}[1]\right)^2} - 2V}{\partial q_{ij}}$$

$$\frac{\partial I1}{\partial q_{ij}[0]} = \frac{q_{ij}[0] - q_{ij-1}[0]}{\sqrt{\left(q_{ij}[0] - q_{ij-1}[0]\right)^2 + \left(q_{ij}[1] - q_{ij-1}[1]\right)^2}} + \frac{q_{ij}[0] - q_{ij+1}[0]}{\sqrt{\left(q_{ij}[0] - q_{ij+1}[0]\right)^2 + \left(q_{ij}[1] - q_{ij+1}[1]\right)^2}}$$

$$\frac{\partial I1}{\partial q_{ij}[1]} = \frac{q_{ij}[1] - q_{ij-1}[1]}{\sqrt{\left(q_{ij}[0] - q_{ij-1}[0]\right)^2 + \left(q_{ij}[1] - q_{ij-1}[1]\right)^2}} + \frac{q_{ij}[1] - q_{ij+1}[1]}{\sqrt{\left(q_{ij}[0] - q_{ij+1}[0]\right)^2 + \left(q_{ij}[1] - q_{ij+1}[1]\right)^2}}$$

$$\frac{\partial I2}{\partial q_{ij}[0]} = -1 * \sum_{forT - norm_2(q_{ij}.\ q_{kj}) \geq 0} \left(y_{ikj} + \frac{p}{2}\right) \left(\frac{q_{ij}[0] - q_{kj}[0]}{\sqrt{\left(q_{ij}[0] - q_{kj}[0]\right)^2 + \left(q_{ij}[1] - q_{kj}[1]\right)^2}}\right)$$

$$\frac{\partial I2}{\partial q_{ij}[1]} = -1 * \sum_{forT - norm_2(q_{ij}.\ q_{kj}) \geq 0} \left(y_{ikj} + \frac{p}{2}\right) \left(\frac{q_{ij}[1] - q_{kj}[1]}{\sqrt{\left(q_{ij}[0] - q_{kj}[0]\right)^2 + \left(q_{ij}[1] - q_{kj}[1]\right)^2}}\right)$$

If $norm_2\left(q_{ij}.\ q_{ij-1}\right) - V \geq 0$:

$$\frac{\partial I3}{\partial q_{ij}[0]} = \frac{(\lambda_{ij} + \frac{p}{2})(q_{ij}[0] - q_{ij-1}[0])}{\sqrt{\left(q_{ij}[0] - q_{ij-1}[0]\right)^2 + \left(q_{ij}[1] - q_{ij-1}[1]\right)^2}} + \frac{(\lambda_{ij+1} + \frac{p}{2})(q_{ij}[0] - q_{ij+1}[0])}{\sqrt{\left(q_{ij}[0] - q_{ij+1}[0]\right)^2 + \left(q_{ij}[1] - q_{ij+1}[1]\right)^2}}$$

$$\frac{\partial I3}{\partial q_{ij}[1]} = \frac{(\lambda_{ij} + \frac{p}{2})(q_{ij}[1] - q_{ij-1}[1])}{\sqrt{\left(q_{ij}[0] - q_{ij-1}[0]\right)^2 + \left(q_{ij}[1] - q_{ij-1}[1]\right)^2}} + \frac{(\lambda_{ij+1} + \frac{p}{2})(q_{ij}[1] - q_{ij+1}[1])}{\sqrt{\left(q_{ij}[0] - q_{ij+1}[0]\right)^2 + \left(q_{ij}[1] - q_{ij+1}[1]\right)^2}}$$

تمام قيدهاي بالا براي هر ايکس و واي هر ايجنت در مسير بايد نوشته شوند تا يك سيستم از معادلات غير خطي را تشكيل دهند.

اضافه کردن شرایط امتیازی

برای بخش امتیازی شرط نخوردن به موانع و پوشش بیشتر محیط و همچنین شرط تغییر یافته ی گرفتن گول ها (تغییر از باینری به فرم زیر) را اضافه کردیم.

$$T - \sum_{\text{for each agent i.k and j in T}} norm_2\left(q_{ij}.\ q_{kj}\right) \leq\ 0 \text{ for maximum convergence}$$

$$T - \min\left(norm_2\left(q_{ij}.\ obstacle_k\right)\right) \leq 0 \text{ for obstacle avoidance}$$

$$\min\left(norm_2\left(q_{ij}.\ goal_k\right)\right) - T \leq 0\text{for goals}$$

تبديل شروط به فرمي كه بتوان در ADMM استفاده كرد.

$$\max\left(0.T - \sum_{\text{for each agent i.k and j in T}} norm_2\left(q_{ij}.\ q_{kj}\right)\right) = 0 \text{ for maximum convergence}$$

$$\max\left(0.T - \min_{\text{for each agent i and j in T}}\left(norm_2\left(q_{ij}.\ obstacle_k\right)\right)\right) = 0 \text{ for obstacle avoidance}$$

$$\max(0.\min_{\text{for each agent i and j in T and goal k}}\left(norm_2\left(q_{ij}.\ goal_k\right)\right) - T) = 0\text{for goals}$$

فرم تابع Lp به شکل زیر میاید.

$$L_p(q.y) = \sum f(q_i) + \sum \left(y_{ikj} + \frac{p}{2}\right) \max\left(0.T - norm_2(q_{ij}.\ q_{kj})\right) + \sum \left(\lambda_{ij} + \frac{p}{2}\right) \max\left(0.norm_2\left(q_{ij}.\ q_{ij-1}\right) - V\right)$$
$$+ (\gamma_1 + \frac{p}{2}) * \max\left(0.T - \sum norm_2\left(q_{ij}.\ q_{kj}\right)\right)$$

$$+\left(\gamma_2+\frac{p}{2}\right)*\max\left(0.T-\min\left(\mathrm{norm}_2\left(q_{ij}.\ \mathrm{obstacle}_k\right)\right)\right)$$

$$+\left(\gamma_3+\frac{p}{2}\right)*\max(0.\min\left(\mathrm{norm}_2\left(q_{ij}.\ \mathrm{goal}_k\right)\right)-T)$$

روش های Dual، Primal

برای این دو روش باید شروط قابلیت دیکامپوز شدن توسط ما را داشته باشند که هیچ یک از شروط این امکان را نداشتند و قابل حل نیستند.

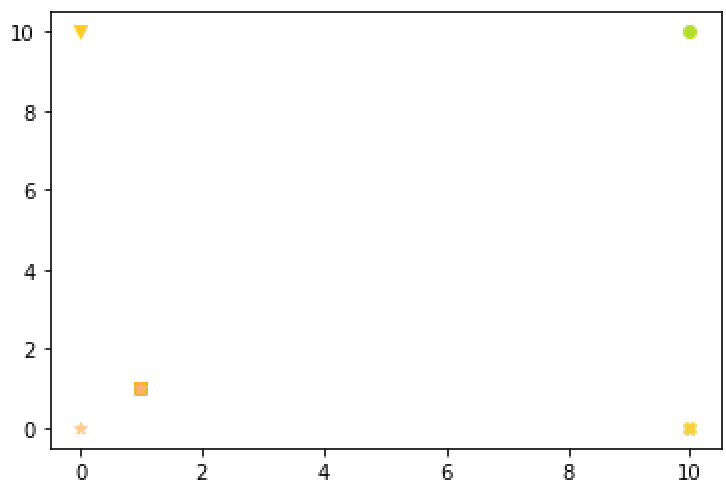
نتایج

نتیجه بخش سنترالایزد

برای حل کردن سیستم معادلات غیر خطی از تابع fsolve پایتون استفاده میکنیم اما چون یکی از محدودیت های این تابع گیر کردن در بهینه محلی برای مسائل سنگین و وابستگی شدیدش به مقدار دهی و تخمین اولیه است و مسئله ی ما تعداد معادله غیر خطی برابر زیر دارد پیدا کردن جواب بهینه برای این تابع مشکل شده.

2*|agents|*| path length|+2*|agents|*|agents|*| path length|+2*|agents|*| path length|

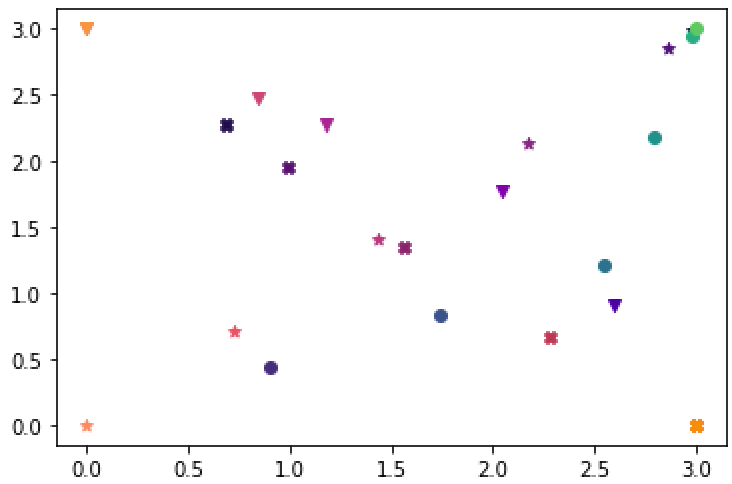
نتیجه برای حالتی که چهار ایجنت در چهار گوشه ی صفحه داریم و هر کدام میخواهد به گوشه رو به روی خود برود به فرم زیر شده که میبینیم حرکت هر ایجنت اصلا مشهود نیست و در حالت لوکال مینیمم که سرعت نزدیک ۰، تداخل ۰ و تنها برای استپ آخر از نقطه ای نزدیک ابتدا پرش به انتها میکند شرط سرعت نقض شده جواب را به ما داده.



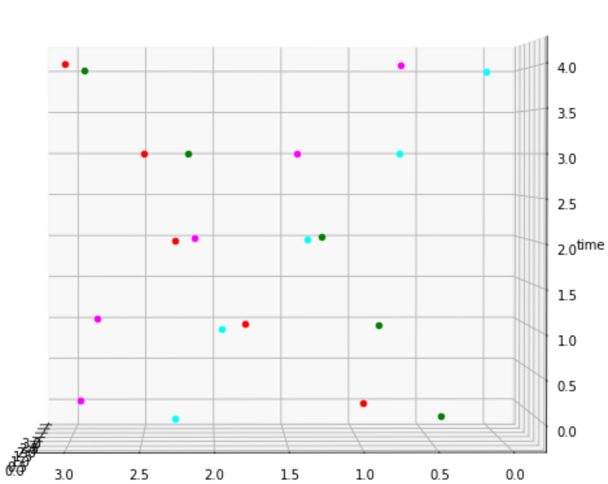
نتیجه ADMM

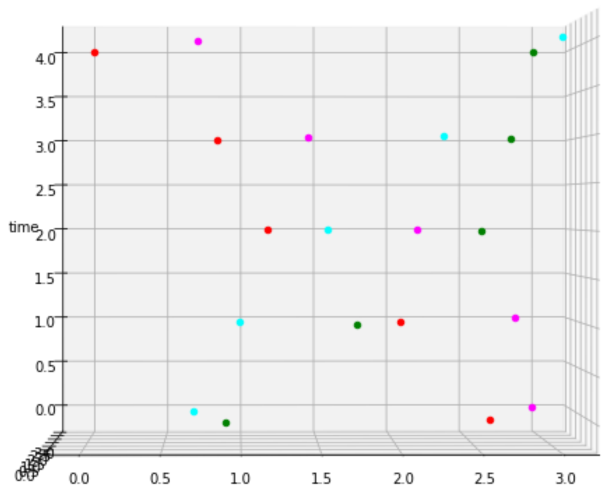
این سیستم از معادلات معرفی شده در بخش را در کد با استفاده از تابع fsolve تعریف میکنیم تا حل شوند. نکته اي که هست این است که حل این خيلي سنگین تر از مینیمایز کردن خود تابع براي ایجنت هاست و جواب خروجي به مقدار خوبي نمیرسد و در جواب لوکال گیر میکند. براي همین از تابع minimize در کد استفاده میکنیم که با دادن ترجکتوري بقیه ایجنت ها ترجکتوري ایجنت مد نظر را جوري تعیین کند که تابع هدف مینیمم شود.

در حالت تست اول ابتدا محیط را کوچک میگیریم به طوری که ایجنت ها در چهار گوشه ی ۰ و ۰ ، ۰ و ۳؛ ۳و ۰، و ۳ و ۳ قرار داشته باشند و هدف هر کدام این باشد که به گوشه مقابلشون برن. در شکل هر مارکر برای یک ایجنت و رنگ ها هر چه تیره تر زمان بیشتری را نشان میدهند. همان طور که میبینیم همه ایجنت ها به مقصد خودشون میرن (دایره از گوشه راست بالا به گوشه چپ پایین، مثلث از گوشه چپ بالا به به راست پایین، ضربدر از راست پایین به چپ بالا و ستاره از گوشه چپ پایین به گوشه راست بالا) و در یک زمان (رنگ های یکسان) ایجنت ها از هم حدود ۱ که ترشلد مورد نظر ماست فاصله دارند.

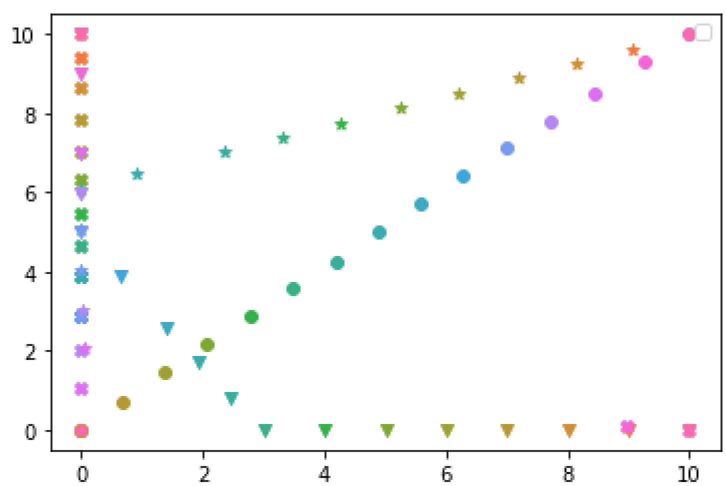


همچنین برای بهتر دیدن کد visualize رو میشه اجرا کرد و آدرس نامپای ترجکتوری ها رو بهش داد (به دلیل اینکه کد خیلی سنگین بود روی سیستم خودم ران نمیشد روی کولب ران کردم و تو کولب سه بعدی با توانایی چرخش نشون داده نمیشه) همون طور که میشه دید اونایی که تو یه زمان یه بعدشون بهم نزدیکه بعد دیگشون از هم دوره (مثلا در زمان ۲ سبز و آبی بهم نزدیکن تو بعد عکس چپ که میبینیم تو بعد عکس راست از هم دورن) که نشون میده فاصله رو رعایت میکنن و تداخلی وجود نداره.

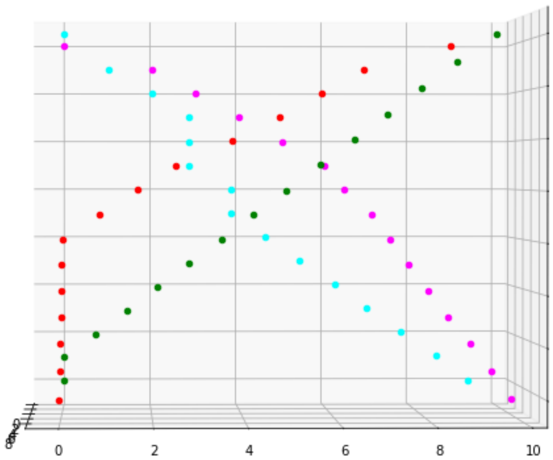


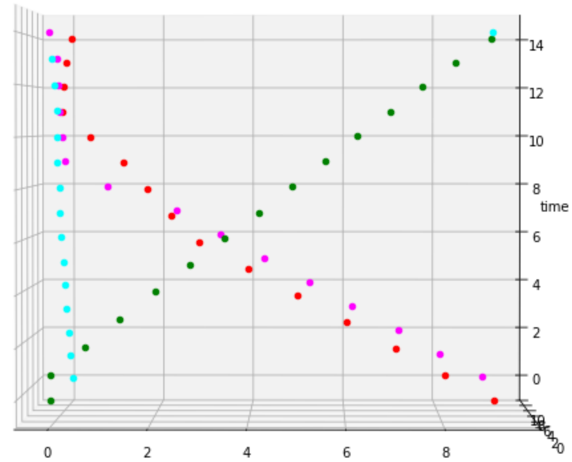


اگر فضا رو بزرگ تر کنیم و بجای ص ۳ در ۳ و پنج قدم تا هدف ص رو ۱۰ در ۱۰ و ۱۵ قدم تا هدف در نظر بگیریم که مسئله سخت تر بشه جواب به فرم یر میشه. همون طور که میبینیم دایره، مثلث، و ستاره شرط سرعت رو رعایت کردن و با استپ سایز مجاز از مبداشون به مقصد رفتن ولی ضربدر تا قدم یکی مونده به آخر رعایت کرده ولی چون نتونسته برسه با ۱۵ قدم تو قدم یکی مونده به آخر پرش کرده و شرط سرعت و استپ سایزو رعایت نکرده.

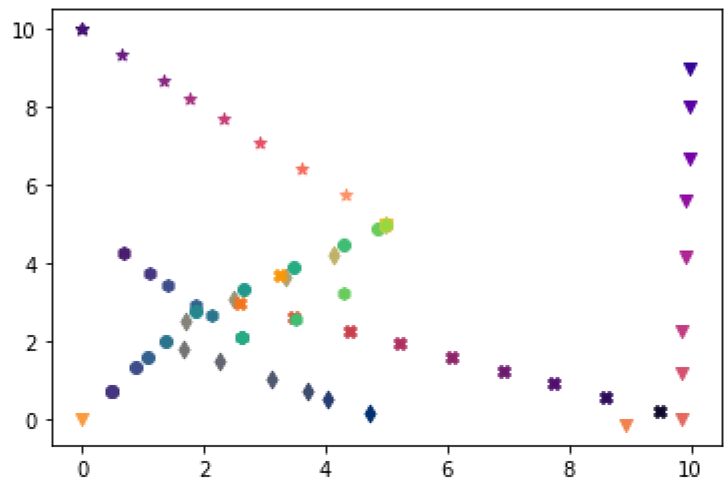


همچنین تو حالت سه بعدی میتونیم تداخل هارو بررسی کنیم. توی بعد راست میبینیم که قرمز و صورتی تو این بعد نزدیک بهم هستن ولی تو بعد چپ فقط تو زمان ۱۱ نزدیک بهم هستن ولی فاصلشون حدود ۱ ئه که مشکلی درست نمیکنه. همچنین نقاط سبز و قرمز و صورتی تو بعد راست نزدیکن تو زمان ۶ که تو بعد چپ فاصله بیشتر از ۱ دارن که اینم مشکلی درست نمیکنه. ولی نقاط آبی و صورتی تو زمان ۱۲ تا ۱۴ تو هر دو بعد نزدیک همن که میشه گفت فاصله اقلیدسیشون نزدیک به ۱ شده و تداخل حدودا رخ داده. پس در کل دو تا تداخل و یه نقض استپ سایز داشتیم با افزایش سختی مسئله توسط افزایش استپ های لازم و ابعاد صفحه. پیش بینی میشه که این سه خطا با بیشتر اجرا شدن تا حدی حل بشن که به دلیل محدودیت زمان بیشتر نشد اجرا شن.

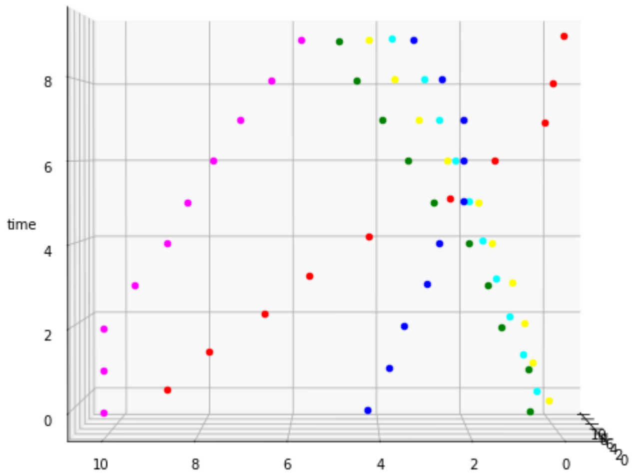


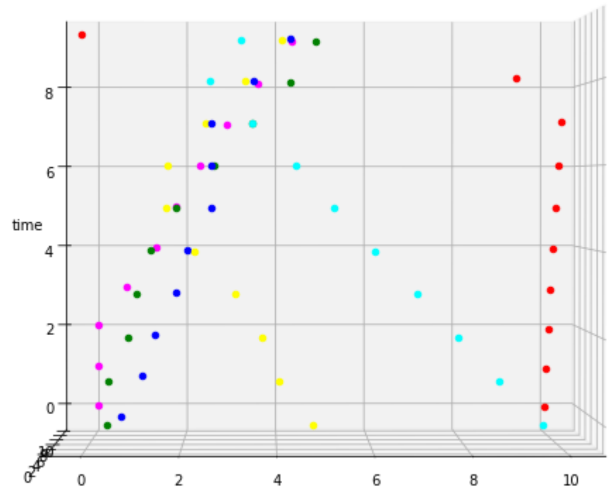


برای سخت تر کردن مسئله میشه تعداد ایجنت ها از ۴ به ۶ زیاد بشن. مختصات چهار ایجنت در چهار گوشه صفحه (۱۰و ۱۰، ۱۰و ۰، ۰ و ۱۰ ، ۰و ۰) و دو ایجنت دیگه در ۵ و ۰ ، ۰ و ۵ هستن و هدف همشون رسیدن به نقطه ی ۵ و ۵ ئه. نمای دو بعدی مکانی این چهار ایجنت رو در پایین میبینیم که چون تعداد ایجنت ها زیاد شده تحلیلش سخته بهتره سه بعدیش رو نگاه کنیم.



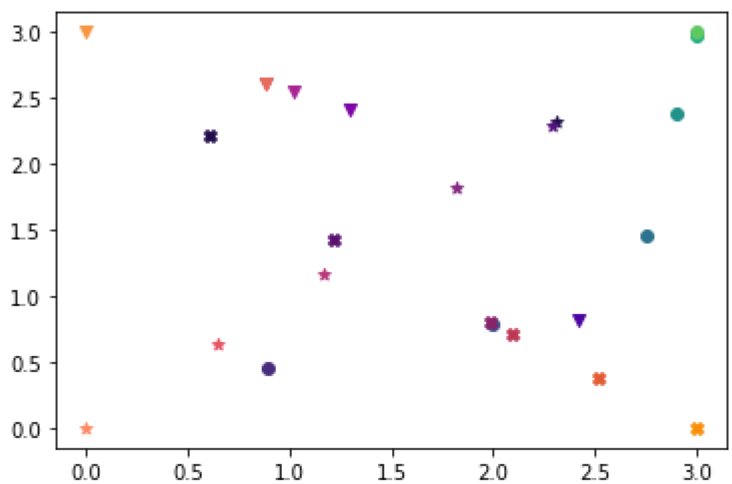
همون طور که در نمودار سه بعدی میبنیم بجز ایجنت قرمز بقیه محدودیت استپ سایز رو رعایت کردن. در رباطه با تداخل ایجنت زرد و سبز و آبی کمرنگ در بعد چپ تا قبل از زمان ۴ نزدیک بهم هستن ولی در بعد راست از هم دورن که نرم فاصلشون رو از یک بیشتر میکنه. از بعد از زمان ۴ سبز به زرد و زرد به آبی کم رنگ و آبی کم رنگ به آبی پر رنگ در بعد چپ فاصله کمتر از یک دارند در بعد راست هم از استیت ۷ تا ۹ هم نزدیک به هم هستن که میشه گفت این سه ایجنت در سه زمان نزدیک به هم بودن که ممکنه با توجه به اینکه مقصد هر سه یجا بوده بشه تا حدی این قضیه رو توجهش کرد. همچنین پیش بینی میشه با بیشتر ران شدن نتیجه ی مسئله ی پیچیده شده بهتر بشه.

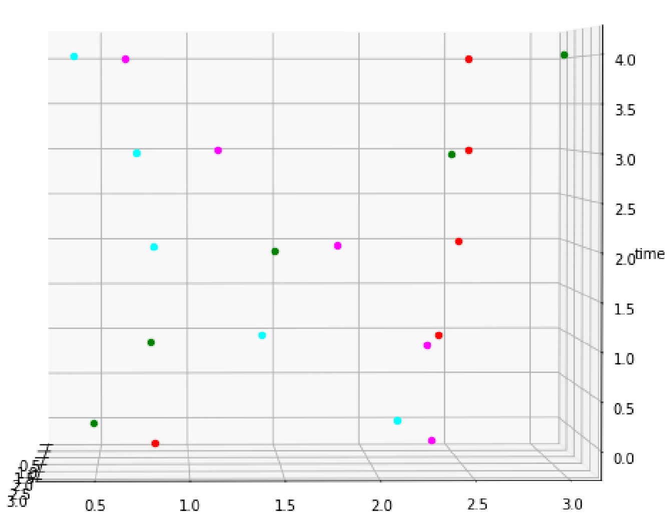


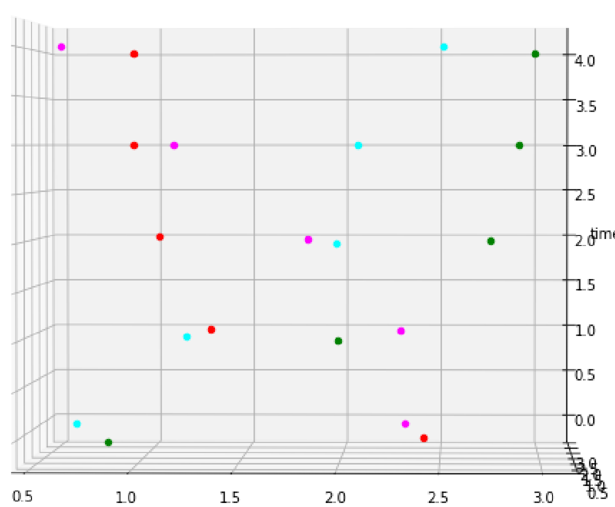


مسئله پایه ایجنت ها در چهار گوشه ی ۰ و ۰ ، ۰ و ۳؛ ۳و ۰، و ۳ و ۳ قرار داشتن رو برای لرنینگ ریت ۰/۵ و ۰/۱ بجای لرنینگ ریت ۱ انجام میدیم.

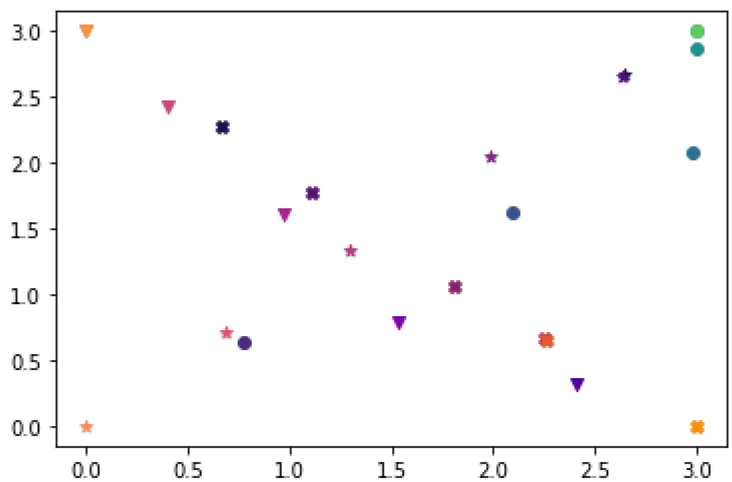
برای لرنینگ ریت ۰/۱ نتیجه به فرم زیر شد همون طور که میبینیم مثل حالت قبل تداخل رو خوب هندل میکنه ولی برای استپ سایز و شرط سرعت یجاهایی رعایت نمیکنه که ازین جهت حالت قبلی بهتر بود.

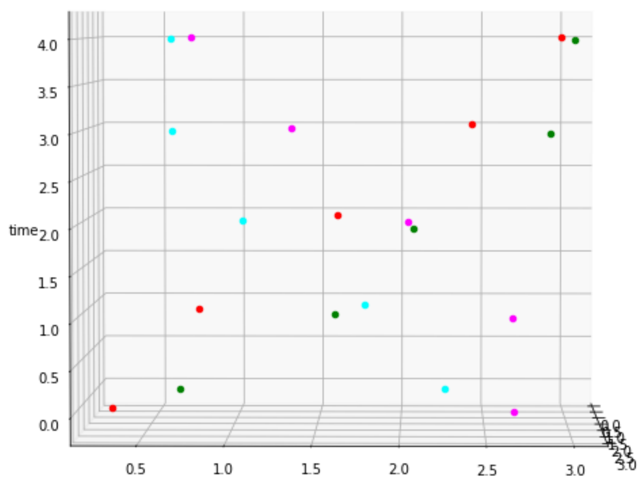


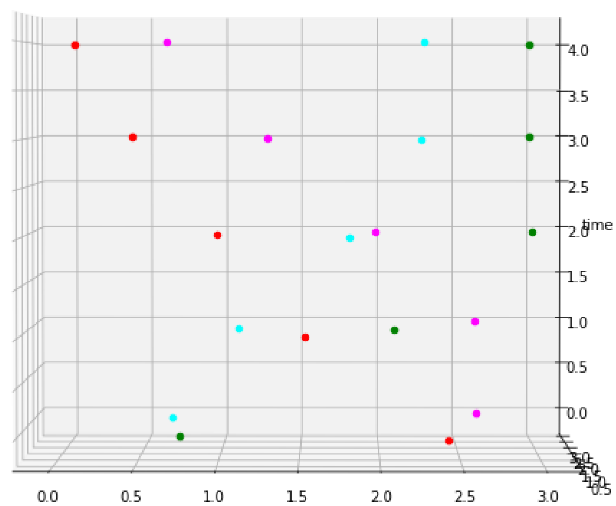




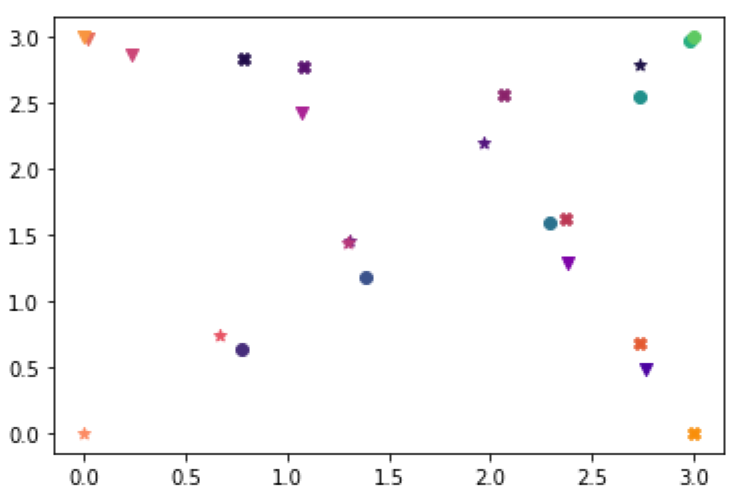
برای لرنینگ ریت ۰/۵ نتیجه به فرم زیر شد همون طور که میبینیم مثل حالت قبل تداخل رو خوب هندل میکنه ولی برای استپ سایز و شرط سرعت یجاهایی رعایت نمیکنه که ازین جهت حالت قبلی بهتر بود.

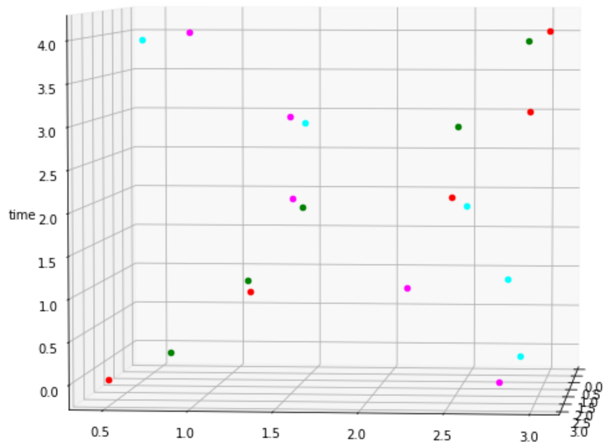


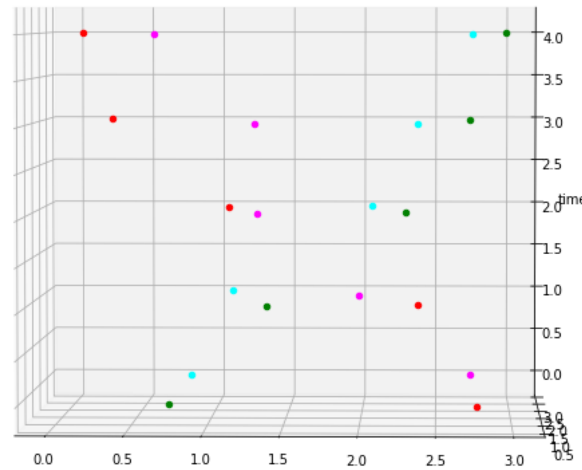




برای لرنینگ ریت 2 نتیجه به فرم زیر شد همون طور که میبینیم مثل حالت قبل تداخل رو خوب هندل میکنه برای استپ سایز و شرط سرعت فقط یجا رعایت نکرده ولی مسیر طی شده توسط ایجنت ها طولانی تر میاد که میشه گفت به شرایط ارزش بیشتری داده تا خود تابع اصلی.



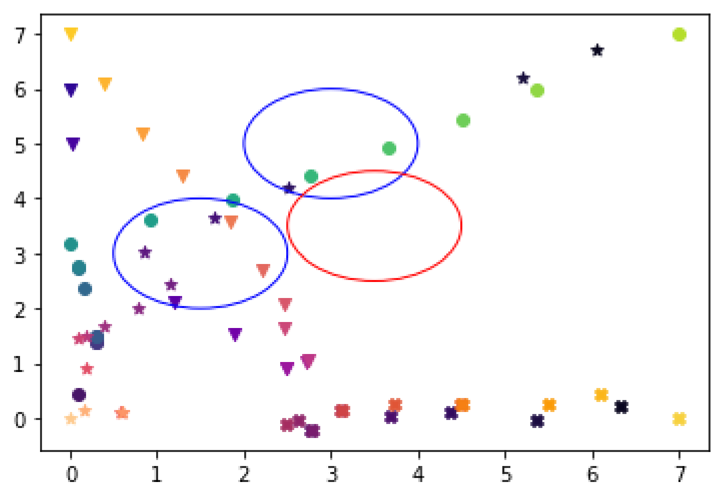




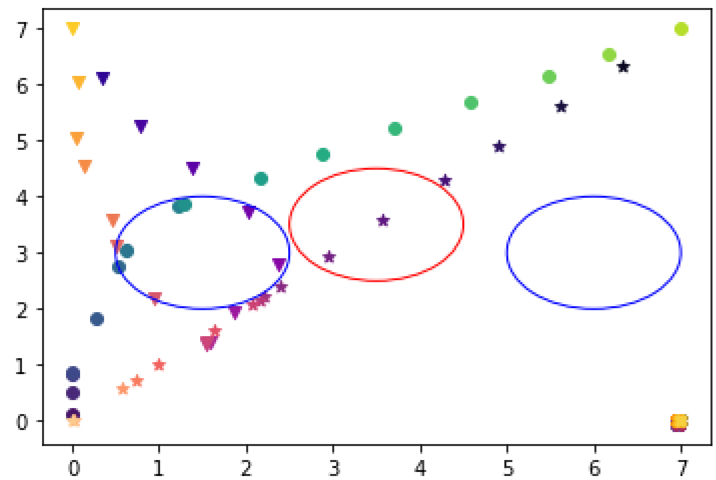
در مقایسه لرنینگ ریت میبینیم که همون لرنینگ ریت قبلی بهتر بوده چون توازن میان شروز و تابع اصلی رو خوب نگه داشته و سرعتش بهتر بوده.

نتیجه ADMM با شروط اضافی

به دلیل سخت شدن پیدا کردن جواب مناسب حل مشتق بخصوص با افزایش طول مسیر از روش مینیمایز کردن تابع هدف استفاده می‌کنیم. برای تست یک مسئله ی اندکی سخت شده با چهار ایجنت که در چهار گوشه ی صفحه قرار دارن با هدف رفتن به گوشه ی مقابلشون رو انتخاب میکنیم. همچنین یک مانع در وسط صفحه قرار میدیم که ایجنت ها نباید از شعاع یک اون مانع رد بشن و همچنین دو گول نزدیک مانع برای سخت شدن میزاریم که حداقل یک ایجنت باید نزدیک گول بشه. در شکل دو بعدی زیر میبنیم که همه ایجنت ها از مانع که دایره قرمز رنگه رد نشدن و گول ها که دو دایره آبی رنگن حتما توسط یک ایجنت دیده شده. تنها نکته ای که نقض شده اینه که ایجنت مثلث و ضربدر تا نزدیک مقصد رفتن و برگشتن که ممکنه بخاطر دوری از تداخل با بقیه از راهشون منحرف شدن. لوپ یادگیری رو برای ده بار تونستیم اجرا کنیم چون همین ده بار هم برای این مسئله حدود دو ساعت و نیم زمان گرفت و پیش بینی میکنیم که اگر بیشتر اجرا کنیمش نتیجه بهتری برسیم.



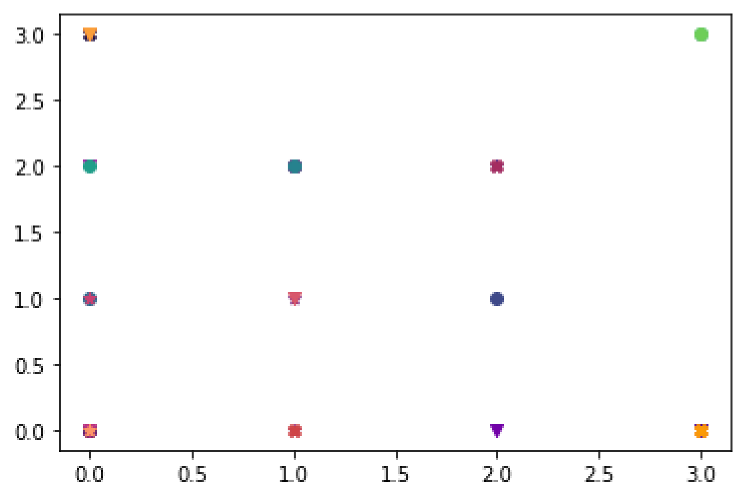
با لرنینگ ریت ۰,۱ میتونیم ببینیم که شروط رو خیلی اهمیت نداده و نیاز به یادگیری بیشتری داره.

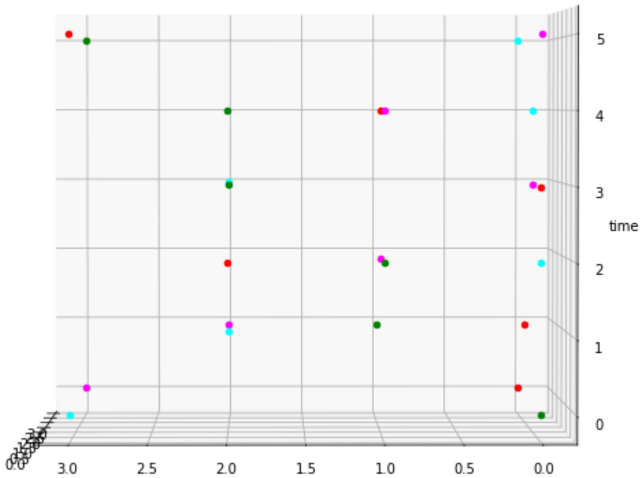


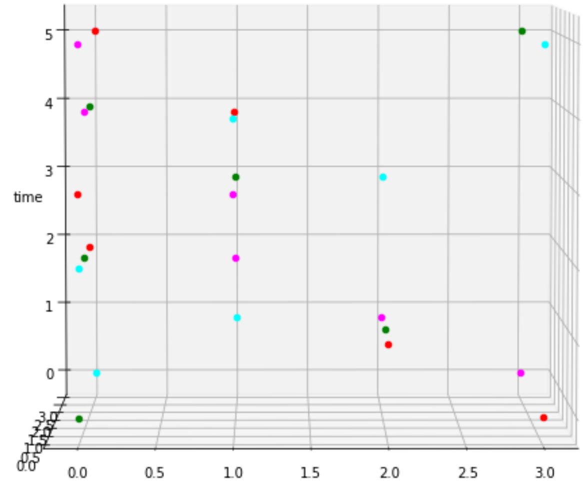
ADMM+RL

چندین روش آر ال برای حل مسئله میتوان پیشنهاد داد. در روش اول آر ال که کل مسیر پیشنهادی یک اکشن باشه و استیت مسیر پیشنهادی بقیه بشه اکشن ها پیوسته می‌شد ولی استیت در هر سری یادگیری تنها یک حالت بود و سمپل ها تکراری و یادگیری سخت میشد. برای همین مسئله را به فرم یک مسئله ی n arm bandit تعریف کرد که مسیر انتخابی یک آرم باشد و ریوارد آن برابر منفی تابع هدفی که میخواهیم بهینه کنیم باشد. چند نکته مطرح میشود که در این صورت باید گرید در نظر گرفته بشه مسیر نه مثل حالت قبل پیوسته، همچنین اندازه گیرید برای اعداد بزرگ مثل ۱۰ رم کولب را پر میکند.

برای مسئله ی ابتدایی با ایجنت ها در ۰و۰، ۳و۰، ۰و۳، ۳و۳ و با هدف رسیدن به نقاط روبه رویی خود با استفاده از روش تامپسون سمپلینگ جواب به فرم زیر شد. این جواب تنها ده بار تکراره برعکس قبلی که صد بار بود. طبق سه تا شکل زیر میبینیم که تداخل رو رعایت کردن و باهم از نرم یک نزدیک تر نمیشن ولی شرط سرعت رو خوب رعایت نکردن.







روش دیگه میشه ایجنت از یک نقطه شروع کنه و به تعداد قدم های محدودی اکشن انجام بده و طبق فاصلش با بقیه ایجنت ها تو هر قدم ریوارد منفی بگیره و در قدم آخر به اندازه فاصلش تا مقصد ریوارد منفی بگیره. روش انتخابی دابل کیو لرنینگه. نتیجه به فرم زیر شد. بدی این روش نسبت به قبلی اینه که قدم به قدم میبینه و هزینه حرکتش زیاد میشه چون فاصلش به بقیه نزدیک میشه ترجیح ممکنه بده که حرکت نکنه. بخاطر همین دیسکانت فاکتور رو باید بالا بزاریم ک به ریواردهای بعدی هم اهمیت بده و هزینه فاصلش از مقصد رو هم زیاد کنیم. نتیجه به فرم زیر شد که باز هم به بهای دور شدن نقطه آخر از مقصد تصمیم میگیرن بهم نزدیک نشن. از جهت بهتر بودن روش قبل بهتر بود ولی این روش برای مسائل بزرگتر بهتر و سریعتره.