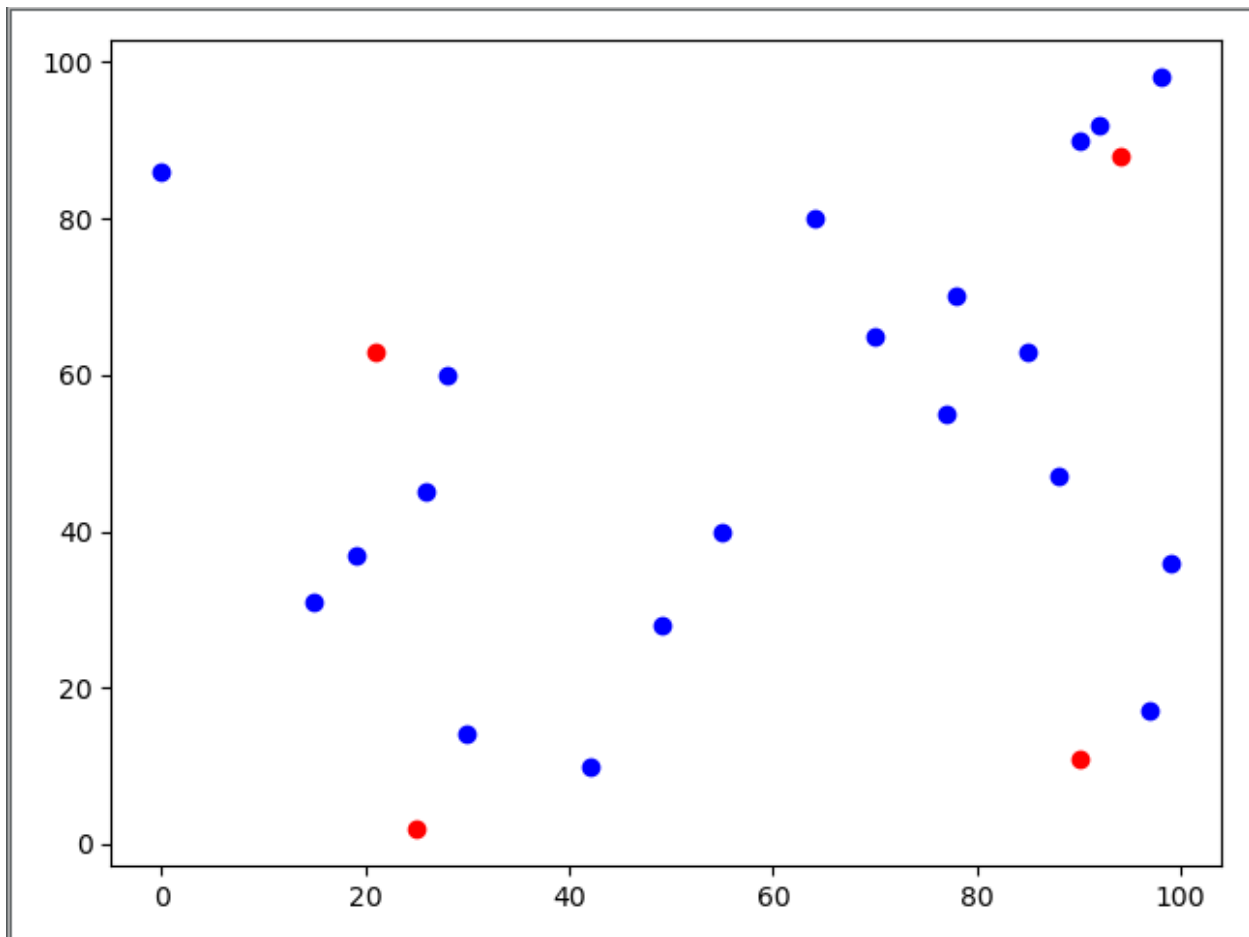


1. За вежбава користев Python scripts кои што се имплементирани на начин кои што генерираат виртуелни мрежи од нодови , во кодот може да се видат параметрите за истите. Со генерирање на различни околина со различни нодови и користење на различен број анкор јазли се добиваат различни резултати. Истите тие ги експортирам во ексел табели кои што се во датотекава која што е пратена и може да ги прочитате.



Пример на оваа околина која што е 100x100 , 1unit = 1m во вистинско време , истото може да се погледне во кодот. Имаме 4 анкор јазли кои што се со црвена боја и имаме 20 јазли кои што се обични и истите треба да ги локализираме според 4те анкор јазли, тест примерот е пуштен на 10 околина и за истите користам 2 алгоритми ToA трилатерација и RSS трилатерација. Шумот на пристигање е представен како noise во 2те скрипти е 0.3 , за секоја околина добиваме вистинска локација и калкулирана локација соодветно со 2те скрипти. Па за споредба ќе имаме 10 околина со 4 анкор јазли и вкупно 20 пресметки по 10 од 2те трилатерации.

2. Подоле имаме резултати од генерираниот примерок , односно околина и се гледа тука дека имаме 4 анкор нодови и 20 обични нодови , потоа секое е еквивалентно со вистинската далечина до анкорите и тоа каков шум имаме и резултати кои што ги добиваме со ToA алгоритмот.

Anchors

```
[[0.08 0.62]
 [0.11 0.57]
 [0.99 0.55]
 [0.55 0.31]]
```

Nodes

```
[[0.17 0.99]
 [0.61 0.1 ]
 [0.81 0.9 ]
 [0.91 0.78]
 [0.31 0.6 ]
 [0.3 0.36]
 [0.5 0.45]
 [0.75 0.51]
 [0.98 0.8 ]
 [0.16 0.4 ]
 [0.65 0.65]
 [0.57 0.58]
 [0.95 0.55]
 [0.37 0.31]
 [0.4 0.88]
 [0.96 0.29]
 [0.32 0.04]
 [0.72 0.01]
 [0.06 0.93]
 [0.51 0.7 ]]
```

Distances

```
[[0.26925824 0.3      0.65802736 0.55081757]
 [0.52502381 0.4852319 0.41647329 0.15443445]
 [0.55285622 0.54722025 0.2782984 0.45590569]
 [0.59770394 0.58485041 0.17219175 0.41862871]
 [0.16324828 0.1430035 0.48213069 0.26617663]
 [0.24083189 0.20024984 0.50606324 0.18027756]
 [0.32039039 0.28853076 0.35362409 0.10511898]
 [0.48010416 0.45453273 0.17204651 0.2      ]
 [0.64899923 0.63631753 0.17691806 0.46097722]
 [0.16552945 0.12529964 0.5964059 0.28301943]
 [0.40360872 0.38600518 0.25059928 0.25059928]
 [0.34763487 0.32534597 0.2977415 0.1914419 ]
 [0.61717096 0.59413803 0.02828427 0.32984845]
```

[0.30016662 0.26 0.47010637 0.12727922]
[0.29154759 0.30016662 0.47801674 0.41677332]
[0.66456753 0.63281119 0.18506756 0.29025851]
[0.44384682 0.40311289 0.59539903 0.25079872]
[0.62517997 0.58553394 0.42690748 0.24382371]
[0.21965883 0.25700195 0.71038722 0.55879334]
[0.30927334 0.29740545 0.35559809 0.27721833]]

Path Loss

[[0.84575339 0.86572758 0.93464301 0.91106255]
[0.91084238 0.90691852 0.8820076 0.78523658]
[0.91511306 0.90465029 0.84868491 0.90369094]
[0.92988573 0.91325606 0.80028893 0.88661011]
[0.80110036 0.78226556 0.91003286 0.84021497]
[0.83988323 0.81334161 0.91070182 0.80224504]
[0.85447742 0.85241257 0.87697506 0.74946233]
[0.9038437 0.9058323 0.80465978 0.81508201]
[0.93667459 0.93204344 0.80252669 0.90257926]
[0.79958738 0.76720385 0.93248717 0.85164894]
[0.88079889 0.87718711 0.84264835 0.84204422]
[0.87259808 0.8594162 0.84891645 0.80537393]
[0.92820433 0.92116929 0.6272232 0.86889201]
[0.85928909 0.84244302 0.90812653 0.76691165]
[0.84843115 0.86323222 0.90957905 0.89183001]
[0.93617051 0.92777267 0.80689698 0.85345894]
[0.90330261 0.88452737 0.93221537 0.83878338]
[0.94083945 0.92264725 0.89314661 0.83621037]
[0.82636364 0.82933551 0.94236663 0.92034123]
[0.85275269 0.85163602 0.88779425 0.85310717]]

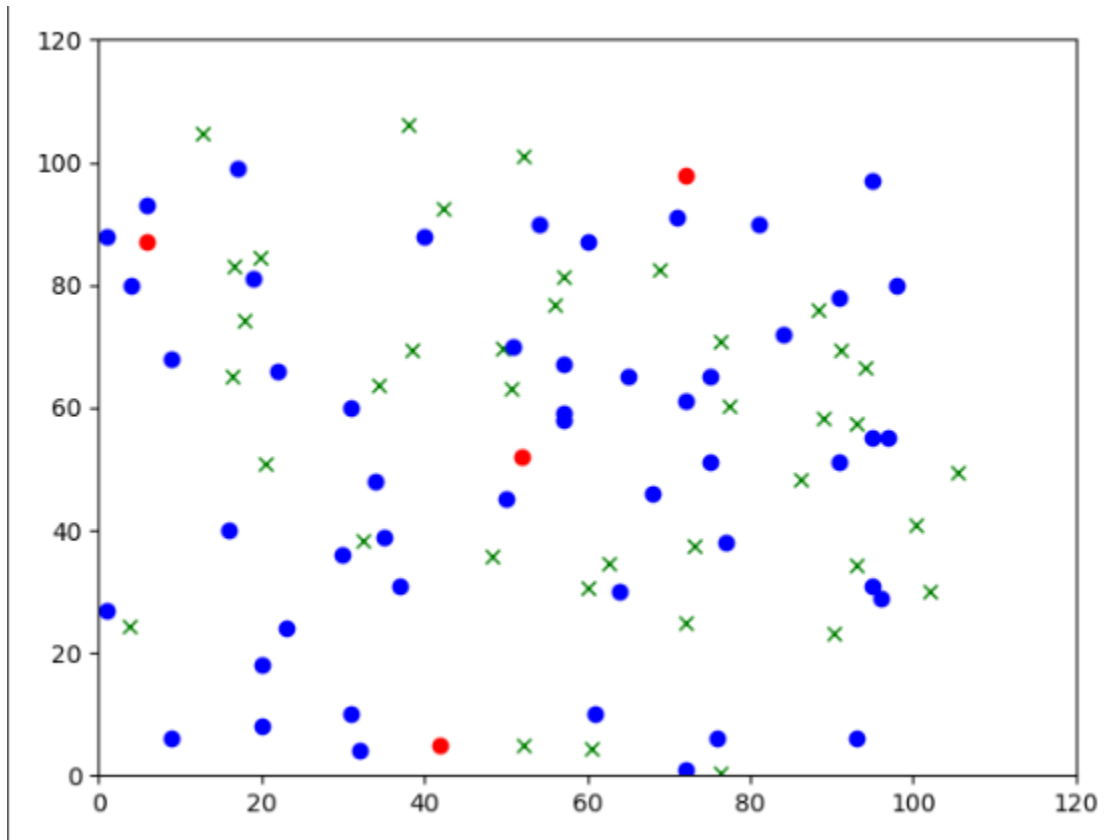
ToA

[[0.25272218 0.27048017 0.62134622 0.52179377]
[0.50772389 0.46386999 0.39554907 0.14082344]
[0.54297791 0.52437876 0.28206568 0.43841872]
[0.57770577 0.56428582 0.1723271 0.39821851]
[0.15918865 0.14140169 0.44965528 0.26947433]
[0.221578 0.17782589 0.4734361 0.16376264]
[0.29266015 0.27404473 0.33793437 0.10391878]
[0.47343784 0.43396367 0.17991225 0.18587293]
[0.61666839 0.59528432 0.14536511 0.43013523]
[0.14921086 0.12104098 0.5615516 0.2645532]
[0.38747605 0.34916086 0.22374157 0.23695637]
[0.3195038 0.31418205 0.29194655 0.17493307]
[0.57752142 0.56046152 0.03062886 0.31794411]
[0.26758141 0.26055519 0.46377999 0.12883379]
[0.26881359 0.28073854 0.4428833 0.40997005]

```
[0.62997929 0.61249201 0.18770382 0.26873031]
[0.42661108 0.37642416 0.563229 0.22866463]
[0.59830324 0.53437073 0.40880043 0.2333528 ]
[0.20675376 0.25337598 0.66590063 0.52169167]
[0.29741331 0.28541787 0.36166872 0.25208809]]
```

Generated Env # 2

3.Резултати



На генерирана пример мрежа добиваме X за локациите кои што ги предвидуваме наспроти оригиналните локации. Како на сликата горе. Со тоа добиваме 2 ексел фајлови кои што ги имаат предвидувањата и MSE(Mean Squared Error) и може да споредиме каде има повеќе грешки и каде има помалце грешки.

```
[3280.0, 47406.0, 15173.0, 3404.0, 8.0, 631.0, 932.0, 17.0, 59.0, 131.0, 28.0, 10771.0, 47590.0, 535.0, 38.0, 11298.0, 21539.0, 8316.0, 968.0, 3753.0, 110.0, 16.0, 447.0, 5.0, 12.0, 6.0, 162.0, 343.0, 264.0, 68.0, 464.0, 271.0, 111.0, 739.0, 13.0, 706.0, 64.0, 1438.0, 90.0, 1.0, 135.0, 113.0, 7.0, 822.0, 343.0, 371.0, 52.0, 1074.0, 486.0, 93.0] ToA MSE
```

```
[7497.0, 495817.0, 2878.0, 22087.0, 173.0, 607.0, 23453.0, 989.0, 10076.0, 454368.0, 27.0, 38671.0, 30546.0, 54.0, 2997.0, 757874.0, 25716.0, 5645.0, 33.0, 23925.0, 914.0, 788.0, 6572.0, 24594.0, 88.0, 317.0, 465.0, 2720.0, 1450.0, 150.0, 11.0, 562.0, 1517.0, 186.0, 1054.0, 2520.0, 4818.0, 28460.0, 593.0, 42.0, 178.0, 893.0, 1926.0, 1278.0, 2764.0, 1849.0, 81.0, 182.0, 835.0, 454.0] RSS MSE
```

Гледаме дека грешките се движат од блиску до нулата до грешки кои што не се прифатливи во ваков систем. Но кога ќе споредиме можеме да видиме дека грешките кои што ги има ToA се многу помали одколку оние кои што ги има RSS.

