TSP with Genetic Algorithm-pr1002

November 6, 2021

1 Getting input

saving Adjacency matrix with weight of edges

[1]:
```
graph = []
n = int(input("Number of cities: "))
x = []
y = []
s = list(map(float, input().split()))
for i in range(0, len(s)-2, 3):
    graph.append([-1 for i in range(n)])
    a, x_in, y_in = s[i], s[i+1], s[i+2]
    x.append(x_in)
    y.append(y_in)
print(a, len(x), len(y))

for i in range(n):
    for j in range(n):
        dis = ((x[i] - x[j])**2 + (y[i] - y[j])**2) ** 0.5
        graph[i][j] = dis
        graph[j][i] = dis

#sample view
print("Sample ###############")
for i in range(4):
    for j in range(4):
        print(graph[i][j], end=' ')
    print('\n')
```

```
Number of cities: 1002
1 1150 4000 2 1050 2750 3 1150 2250 4 1250 2050 5 1350 2350 6 1050 1550 7 3350
1700 8 3450 1450 9 3550 1600 10 3950 1700 11 4050 2000 12 4050 2150 13 4250 1650
14 4150 1500 15 4450 1450 16 4400 1700 17 4600 1850 18 4900 1550 19 5100 1550 20
5350 1450 21 4950 1700 22 4850 1900 23 4900 2050 24 5000 2150 25 5100 2050 26
5400 2050 27 5750 2000 28 5900 2050 29 5600 2250 30 5400 2300 31 5250 2250 32
5000 2350 33 5000 2550 34 5050 2800 35 5250 2750 36 5450 2750 37 5400 2950 38
5200 3150 39 5050 3100 40 4950 3300 41 5100 3600 42 5200 3650 43 5350 3750 44
```

5450 3750 45 5600 3750 46 5600 4250 47 5450 4250 48 5350 4150 49 5050 3800 50
4950 3500 51 4700 3500 52 4400 3700 53 4450 3500 54 4100 3500 55 4150 3300 56
4100 3150 57 4300 3300 58 4500 3150 59 4500 2950 60 4700 3000 61 4700 2800 62
4700 2500 63 4600 2350 64 4550 2500 65 4550 2800 66 4300 2800 67 4100 2950 68
3700 2800 69 3550 2800 70 3400 2700 71 3400 3200 72 3700 3100 73 3550 3300 74
3350 4250 75 3350 4650 76 3250 5200 77 4000 5050 78 4100 4750 79 3950 4650 80
3950 4250 81 4200 4150 82 4200 4550 83 4500 4400 84 4700 4350 85 4700 4750 86
4500 4800 87 4500 4950 88 4350 4750 89 4350 4900 90 4247 5150 91 4354 5256 92
4247 5256 93 4100 5250 94 4100 5350 95 4000 5550 96 4199 5554 97 4305 5554 98
4304 5447 99 4500 5450 100 4500 6050 101 4650 6000 102 4800 6100 103 5050 6050
104 5150 6150 105 5300 6050 106 5400 6050 107 5150 5500 108 5250 5350 109 5300
5150 110 5650 5350 111 5800 5350 112 5650 5500 113 5800 5500 114 5750 5700 115
5750 5850 116 5700 6000 117 5700 6150 118 5550 6000 119 5550 6150 120 6000 6600
121 6000 7000 122 5550 6650 123 5400 6550 124 5300 6450 125 5150 6550 126 5150
7000 127 5050 6550 128 4800 6500 129 4650 6400 130 4500 6550 131 4250 6700 132
4250 6350 133 4250 5950 134 4100 5850 135 3950 5950 136 3800 5950 137 3950 6350
138 3800 6350 139 3250 7000 140 3300 7450 141 3300 7550 142 3900 7850 143 4250
7800 144 4400 7750 145 4250 7300 146 4450 7450 147 4650 7450 148 4550 7650 149
4550 7900 150 4650 8150 151 4400 8250 152 4250 8250 153 4550 8750 154 4550 8950
155 4250 9250 156 4300 9450 157 4500 9750 158 4300 9950 159 4300 10150 160 4100
10150 161 3900 10150 162 4200 9850 163 4050 9700 164 4200 9450 165 4050 9200 166
3900 9250 167 3750 9500 168 3450 9450 169 3300 9350 170 3300 10650 171 3450
10450 172 3750 10500 173 3600 11100 174 3750 11200 175 3750 11400 176 3600 11600
177 3900 11550 178 3900 11400 179 3900 11150 180 4100 11150 181 4300 11150 182
4600 11100 183 4600 10850 184 4550 10550 185 4500 10250 186 5600 9750 187 5600
10250 188 5950 10250 189 6100 10250 190 6250 10200 191 6400 10300 192 6200 10300
193 6150 10400 194 5950 10550 195 5650 10550 196 5600 10850 197 5950 10850 198
6050 11050 199 6150 11050 200 6200 11150 201 5950 11150 202 5600 11100 203 5950
11650 204 6050 11550 205 6450 11550 206 6350 11350 207 6550 11350 208 6500 11150
209 6650 11050 210 6950 11250 211 6950 10650 212 7000 10500 213 7200 10500 214
7200 10200 215 7100 10250 216 7050 10400 217 6900 10450 218 6700 10300 219 6850
10200 220 7000 10050 221 6800 9950 222 7000 9800 223 7100 9750 224 7300 9650 225
7300 9350 226 7300 9150 227 7300 9000 228 7000 9000 229 6900 9000 230 6950 9250
231 6700 9150 232 6800 9350 233 6700 9650 234 6700 9800 235 6600 9800 236 6500
10050 237 6400 10050 238 6200 9950 239 6100 10050 240 6100 9800 241 5950 9750
242 6000 9600 243 6150 9200 244 6250 8950 245 6250 8850 246 6350 8850 247 6150
8600 248 6150 8500 249 6000 8450 250 6000 8600 251 5550 8950 252 5550 8750 253
5650 8150 254 5550 7900 255 5550 7650 256 5650 7450 257 5900 7350 258 5950 7650
259 6000 7850 260 6400 8350 261 6450 8600 262 6500 8700 263 6500 9000 264 6600
9000 265 6699 8856 266 6699 8750 267 6805 8749 268 6750 8600 269 6750 8500 270
6600 8000 271 6900 8350 272 7200 8000 273 7350 7650 274 7000 7350 275 7100 6750
276 7100 6600 277 6800 6000 278 6603 6057 279 6498 6057 280 6604 5950 281 6700
5800 282 6600 5700 283 6550 5500 284 6800 5500 285 7100 5500 286 7100 5100 287
7350 4900 288 7350 4750 289 7350 4150 290 7350 4050 291 7350 3900 292 7350 3800
293 6950 3900 294 6850 3800 295 6850 4050 296 6950 4150 297 6650 4900 298 6700
5050 299 6700 5300 300 6600 5200 301 6250 5500 302 6106 5606 303 5999 5607 304
5998 5500 305 6250 5000 306 6100 4800 307 6000 4450 308 6100 4400 309 6100 4150
310 6000 3950 311 6000 3800 312 5850 3750 313 5750 3450 314 5850 3050 315 6000

3200 316 6000 3300 317 6100 3650 318 6148 3556 319 6148 3450 320 6254 3557 321
6450 3750 322 6450 3650 323 6450 3450 324 6254 3206 325 6148 3206 326 6148 3099
327 6450 2950 328 6450 2800 329 6300 2800 330 6150 2800 331 6000 2800 332 6150
2300 333 6300 2300 334 6450 2300 335 8100 1700 336 8200 1450 337 8300 1600 338
8700 1700 339 8800 2000 340 8800 2150 341 9000 1650 342 8900 1500 343 9200 1450
344 9150 1700 345 9350 1850 346 9650 1550 347 9850 1550 348 10100 1450 349 9700
1700 350 9600 1900 351 9650 2050 352 9750 2150 353 9850 2050 354 10150 2050 355
10500 2000 356 10650 2050 357 10350 2250 358 10150 2300 359 10000 2250 360 9750
2350 361 9750 2550 362 9800 2800 363 10000 2750 364 10200 2750 365 10150 2950
366 9950 3150 367 9800 3100 368 9700 3300 369 9850 3600 370 9950 3650 371 10100
3750 372 10200 3750 373 10350 3750 374 10350 4250 375 10200 4250 376 10100 4150
377 9800 3800 378 9700 3500 379 9450 3500 380 9150 3700 381 9200 3500 382 8850
3500 383 8900 3300 384 8850 3150 385 9050 3300 386 9250 3150 387 9250 2950 388
9450 3000 389 9450 2800 390 9450 2500 391 9350 2350 392 9300 2500 393 9300 2800
394 9050 2800 395 8850 2950 396 8450 2800 397 8300 2800 398 8150 2700 399 8150
3200 400 8450 3100 401 8300 3300 402 8100 4250 403 8100 4650 404 8000 5200 405
8750 5050 406 8850 4750 407 8700 4650 408 8700 4250 409 8950 4150 410 8950 4550
411 9250 4400 412 9450 4350 413 9450 4750 414 9250 4800 415 9250 4950 416 9100
4750 417 9100 4900 418 8997 5150 419 9104 5256 420 8997 5256 421 8850 5250 422
8850 5350 423 8750 5550 424 8949 5554 425 9055 5554 426 9054 5447 427 9250 5450
428 9250 6050 429 9400 6000 430 9550 6100 431 9800 6050 432 9900 6150 433 10050
6050 434 10150 6050 435 9900 5500 436 10000 5350 437 10050 5150 438 10400 5350
439 10550 5350 440 10400 5500 441 10550 5500 442 10500 5700 443 10500 5850 444
10450 6000 445 10450 6150 446 10300 6000 447 10300 6150 448 10750 6600 449 10750
7000 450 10300 6650 451 10150 6550 452 10050 6450 453 9900 6550 454 9900 7000
455 9800 6550 456 9550 6500 457 9400 6400 458 9250 6550 459 9000 6700 460 9000
6350 461 9000 5950 462 8850 5850 463 8700 5950 464 8550 5950 465 8700 6350 466
8550 6350 467 8000 7000 468 8050 7450 469 8050 7550 470 8650 7850 471 9000 7800
472 9150 7750 473 9000 7300 474 9200 7450 475 9400 7450 476 9300 7650 477 9300
7900 478 9400 8150 479 9150 8250 480 9000 8250 481 9300 8750 482 9300 8950 483
9000 9250 484 9050 9450 485 9250 9750 486 9050 9950 487 9050 10150 488 8850
10150 489 8650 10150 490 8950 9850 491 8800 9700 492 8950 9450 493 8800 9200 494
8650 9250 495 8500 9500 496 8200 9450 497 8050 9350 498 8050 10650 499 8200
10450 500 8500 10500 501 8350 11100 502 8500 11200 503 8500 11400 504 8350 11600
505 8650 11550 506 8650 11400 507 8650 11150 508 8850 11150 509 9050 11150 510
9350 11100 511 9350 10850 512 9300 10550 513 9250 10250 514 10350 9750 515 10350
10250 516 10700 10250 517 10850 10250 518 11000 10200 519 11150 10300 520 10950
10300 521 10900 10400 522 10700 10550 523 10400 10550 524 10350 10850 525 10700
10850 526 10800 11050 527 10900 11050 528 10950 11150 529 10700 11150 530 10350
11100 531 10700 11650 532 10800 11550 533 11200 11550 534 11100 11350 535 11300
11350 536 11250 11150 537 11400 11050 538 11700 11250 539 11700 10650 540 11750
10500 541 11950 10500 542 11950 10200 543 11850 10250 544 11800 10400 545 11650
10450 546 11450 10300 547 11600 10200 548 11750 10050 549 11550 9950 550 11750
9800 551 11850 9750 552 12050 9650 553 12050 9350 554 12050 9150 555 12050 9000
556 11750 9000 557 11650 9000 558 11700 9250 559 11450 9150 560 11550 9350 561
11450 9650 562 11450 9800 563 11350 9800 564 11250 10050 565 11150 10050 566
10950 9950 567 10850 10050 568 10850 9800 569 10700 9750 570 10750 9600 571
10900 9200 572 11000 8950 573 11000 8850 574 11100 8850 575 10900 8600 576 10900

8500 577 10750 8450 578 10750 8600 579 10300 8950 580 10300 8750 581 10400 8150
582 10300 7900 583 10300 7650 584 10400 7450 585 10650 7350 586 10700 7650 587
10750 7850 588 11150 8350 589 11200 8600 590 11250 8700 591 11250 9000 592 11350
9000 593 11449 8856 594 11449 8750 595 11555 8749 596 11500 8600 597 11500 8500
598 11350 8000 599 11650 8350 600 11950 8000 601 12100 7650 602 11750 7350 603
11850 6750 604 11850 6600 605 11550 6000 606 11353 6057 607 11248 6057 608 11354
5950 609 11450 5800 610 11350 5700 611 11300 5500 612 11550 5500 613 11850 5500
614 11850 5100 615 12100 4900 616 12100 4750 617 12100 4150 618 12100 4050 619
12100 3900 620 12100 3800 621 11700 3900 622 11600 3800 623 11600 4050 624 11700
4150 625 11400 4900 626 11450 5050 627 11450 5300 628 11350 5200 629 11000 5500
630 10856 5606 631 10749 5607 632 10748 5500 633 11000 5000 634 10850 4800 635
10750 4450 636 10850 4400 637 10850 4150 638 10750 3950 639 10750 3800 640 10600
3750 641 10500 3450 642 10600 3050 643 10750 3200 644 10750 3300 645 10850 3650
646 10898 3556 647 10898 3450 648 11004 3557 649 11200 3750 650 11200 3650 651
11200 3450 652 11004 3206 653 10898 3206 654 10898 3099 655 11200 2950 656 11200
2800 657 11050 2800 658 10900 2800 659 10750 2800 660 10900 2300 661 11050 2300
662 11200 2300 663 12850 1700 664 12950 1450 665 13050 1600 666 13450 1700 667
13550 2000 668 13550 2150 669 13750 1650 670 13650 1500 671 13950 1450 672 13900
1700 673 14100 1850 674 14400 1550 675 14600 1550 676 14850 1450 677 14450 1700
678 14350 1900 679 14400 2050 680 14500 2150 681 14600 2050 682 14900 2050 683
15250 2000 684 15400 2050 685 15100 2250 686 14900 2300 687 14750 2250 688 14500
2350 689 14500 2550 690 14550 2800 691 14750 2750 692 14950 2750 693 14900 2950
694 14700 3150 695 14550 3100 696 14450 3300 697 14600 3600 698 14700 3650 699
14850 3750 700 14950 3750 701 15100 3750 702 15100 4250 703 14950 4250 704 14850
4150 705 14550 3800 706 14450 3500 707 14200 3500 708 13900 3700 709 13950 3500
710 13600 3500 711 13650 3300 712 13600 3150 713 13800 3300 714 14000 3150 715
14000 2950 716 14200 3000 717 14200 2800 718 14200 2500 719 14100 2350 720 14050
2500 721 14050 2800 722 13800 2800 723 13600 2950 724 13200 2800 725 13050 2800
726 12900 2700 727 12900 3200 728 13200 3100 729 13050 3300 730 12850 4250 731
12850 4650 732 12750 5200 733 13500 5050 734 13600 4750 735 13450 4650 736 13450
4250 737 13700 4150 738 13700 4550 739 14000 4400 740 14200 4350 741 14200 4750
742 14000 4800 743 14000 4950 744 13850 4750 745 13850 4900 746 13747 5150 747
13854 5256 748 13747 5256 749 13600 5250 750 13600 5350 751 13500 5550 752 13699
5554 753 13805 5554 754 13804 5447 755 14000 5450 756 14000 6050 757 14150 6000
758 14300 6100 759 14550 6050 760 14650 6150 761 14800 6050 762 14900 6050 763
14650 5500 764 14750 5350 765 14800 5150 766 15150 5350 767 15300 5350 768 15150
5500 769 15300 5500 770 15250 5700 771 15250 5850 772 15200 6000 773 15200 6150
774 15050 6000 775 15050 6150 776 15500 6600 777 15500 7000 778 15050 6650 779
14900 6550 780 14800 6450 781 14650 6550 782 14650 7000 783 14550 6550 784 14300
6500 785 14150 6400 786 14000 6550 787 13750 6700 788 13750 6350 789 13750 5950
790 13600 5850 791 13450 5950 792 13300 5950 793 13450 6350 794 13300 6350 795
12750 7000 796 12800 7450 797 12800 7550 798 13400 7850 799 13750 7800 800 13900
7750 801 13750 7300 802 13950 7450 803 14150 7450 804 14050 7650 805 14050 7900
806 14150 8150 807 13900 8250 808 13750 8250 809 14050 8750 810 14050 8950 811
13750 9250 812 13800 9450 813 14000 9750 814 13800 9950 815 13800 10150 816
13600 10150 817 13400 10150 818 13700 9850 819 13550 9700 820 13700 9450 821
13550 9200 822 13400 9250 823 13250 9500 824 12950 9450 825 12800 9350 826 12800
10650 827 12950 10450 828 13250 10500 829 13100 11100 830 13250 11200 831 13250

```
11400 832 13100 11600 833 13400 11550 834 13400 11400 835 13400 11150 836 13600
11150 837 13800 11150 838 14100 11100 839 14100 10850 840 14050 10550 841 14000
10250 842 15100 9750 843 15100 10250 844 15450 10250 845 15600 10250 846 15750
10200 847 15900 10300 848 15700 10300 849 15650 10400 850 15450 10550 851 15150
10550 852 15100 10850 853 15450 10850 854 15550 11050 855 15650 11050 856 15700
11150 857 15450 11150 858 15100 11100 859 15450 11650 860 15550 11550 861 15950
11550 862 15850 11350 863 16050 11350 864 16000 11150 865 16150 11050 866 16450
11250 867 16450 10650 868 16500 10500 869 16700 10500 870 16700 10200 871 16600
10250 872 16550 10400 873 16400 10450 874 16200 10300 875 16350 10200 876 16500
10050 877 16300 9950 878 16500 9800 879 16600 9750 880 16800 9650 881 16800 9350
882 16800 9150 883 16800 9000 884 16500 9000 885 16400 9000 886 16450 9250 887
16200 9150 888 16300 9350 889 16200 9650 890 16200 9800 891 16100 9800 892 16000
10050 893 15900 10050 894 15700 9950 895 15600 10050 896 15600 9800 897 15450
9750 898 15500 9600 899 15650 9200 900 15750 8950 901 15750 8850 902 15850 8850
903 15650 8600 904 15650 8500 905 15500 8450 906 15500 8600 907 15050 8950 908
15050 8750 909 15150 8150 910 15050 7900 911 15050 7650 912 15150 7450 913 15400
7350 914 15450 7650 915 15500 7850 916 15900 8350 917 15950 8600 918 16000 8700
919 16000 9000 920 16100 9000 921 16199 8856 922 16199 8750 923 16305 8749 924
16250 8600 925 16250 8500 926 16100 8000 927 16400 8350 928 16700 8000 929 16850
7650 930 16500 7350 931 16600 6750 932 16600 6600 933 16300 6000 934 16103 6057
935 15998 6057 936 16104 5950 937 16200 5800 938 16100 5700 939 16050 5500 940
16300 5500 941 16600 5500 942 16600 5100 943 16850 4900 944 16850 4750 945 16850
4150 946 16850 4050 947 16850 3900 948 16850 3800 949 16450 3900 950 16350 3800
951 16350 4050 952 16450 4150 953 16150 4900 954 16200 5050 955 16200 5300 956
16100 5200 957 15750 5500 958 15606 5606 959 15499 5607 960 15498 5500 961 15750
5000 962 15600 4800 963 15500 4450 964 15600 4400 965 15600 4150 966 15500 3950
967 15500 3800 968 15350 3750 969 15250 3450 970 15350 3050 971 15500 3200 972
15500 3300 973 15600 3650 974 15648 3556 975 15648 3450 976 15754 3557 977 15950
3750 978 15950 3650 979 15950 3450 980 15754 3206 981 15648 3206 982 15648 3099
983 15950 2950 984 15950 2800 985 15800 2800 986 15650 2800 987 15500 2800 988
15650 2300 989 15800 2300 990 15950 2300 991 6450 4950 992 11200 4950 993 15950
4950 994 5050 5750 995 5050 8450 996 5050 11650 997 9800 5750 998 9800 8450 999
9800 11650 1000 14550 5750 1001 14550 8450 1002 14550 11650
1002.0 1002 1002
Sample ##############
0.0 1253.9936203984453 1750.0 1952.5624189766636

1253.9936203984453 0.0 509.9019513592785 728.0109889280518

1750.0 509.9019513592785 0.0 223.60679774997897

1952.5624189766636 728.0109889280518 223.60679774997897 0.0
```

2 Creating random chromosomes

generating a random permutation

[4]:
```
import random
def first_population(number_of_samples, size):
    l = []
    for i in range(number_of_samples):
        chromosome = []
        for j in range(size):
            a = random.randint(0,size-1)
            while a in chromosome:
                a = random.randint(0,size-1)
            chromosome.append(a)
        l.append(chromosome)
    return l

first_population(3,5)
```

[4]:
```
[[4, 1, 3, 2, 0], [4, 2, 1, 0, 3], [0, 1, 3, 2, 4]]
```

3 Fitness function (sum of weights in chromosome path)

[5]:
```
def fitness(chromosome):
    w = 0
    for i in range(len(chromosome)):
        w += graph[chromosome[i]][chromosome[(i+1) % len(chromosome)]]
    return w

chromosome = first_population(1,29)[0]
print(chromosome)
print(fitness(chromosome))
```

```
[7, 8, 4, 23, 20, 28, 1, 9, 15, 5, 2, 0, 12, 6, 26, 27, 16, 11, 24, 17, 3, 22,
10, 18, 21, 14, 13, 25, 19]
46821.68432989189
```

4 Cross over(ordered crossover)

[6]:
```
def cross_over(parent1, parent2):

    size = len(parent1)
    # result
    res = [-1 for i in range(size)]
    a = random.randint(0, size-1)
    b = -1
    while b == a or b == -1:
        b = random.randint(0, size-1)
    if (a > b):
        a, b = b, a
    # putting parent1 genes
    i = a
    while i != b:
        res[i] = parent1[i]
        i = (i + 1) % size

    # putting parent2 genes
    cnt = 0
    i = b
    while cnt < size:

        if parent2[cnt] not in res:
            res[i] = parent2[cnt]
            i = (i + 1) % size
        cnt += 1


    return res
parents = first_population(2,5)
print(parents)
print(cross_over(parents[0], parents[1]))
```

```
[[1, 4, 2, 0, 3], [1, 4, 3, 0, 2]]
[2, 4, 1, 3, 0]
```

5 Mutation

[7]:
```
def mutation(chromosome, mutation_rate):
    for i in range(len(chromosome)):
        if (random.random() < mutation_rate):
            x = random.randint(0,len(chromosome)-1)
```

```
            # swap:
            chromosome[i], chromosome[x] = chromosome[x], chromosome[i]
    return chromosome

print(mutation([1,2,3,4,5], 0.5))
```

```
[3, 2, 4, 1, 5]
```

6 Main Function

improvement in order(time complexity) of selction with new approach

[12]:
```
import matplotlib.pyplot as plt
import bisect
def reverse(l):
    return float(1) / float(l)

def genentic(iterations, population_number, mutation_rate, elite_size):
    print("D")
    population = first_population(population_number, n)
    print("SS")
    progress = []
    for i in range(iterations):



        print(i,mutation_rate, fitness(population[0]), fitness(population[1]),␣
 ↪end = '\r', flush=True)
        population.sort(key = fitness)
        progress.append(fitness(population[0]))
        child = []
        # moving elites to next generation
        for j in range(elite_size):
            child.append(population[j])

        # selection based on fitness(normalize fitness)
        select = []
        normalized_fitness = [(reverse(fitness(j))) for j in population]
        reversed(population)
        sum_fitness = sum(normalized_fitness)
        prefix_sum = [normalized_fitness[0]/sum_fitness]
        for k in range(1, len(population)):
            prefix_sum.append(prefix_sum[k-1] + float(normalized_fitness[k]) /␣
 ↪float(sum_fitness))

        while(len(select) < population_number - elite_size):
```

```
            x = random.random()
            select.append(population[bisect.bisect(prefix_sum, x)])

        # making new child from selected parents
        j = 0
        population.sort(key = fitness)
        while (len(child) < population_number):
            child.append(cross_over(select[j], select[len(select) - 1 - j]))
            j = (j + 1) % len(select)
        # mutate next generation
        for j in range(elite_size, population_number):
            if (random.random() < mutation_rate):
                child[j] = mutation(child[j], mutation_rate)
        population = child

    population.sort(key = fitness)
    print(population[0], fitness(population[0]))

    plt.plot(progress)
    plt.ylabel('Distance')
    plt.xlabel('Generation')
    plt.show()

genentic(500, 100, 0.001, 20)
```

```
D
SS
[226, 535, 749, 834, 3, 49, 446, 334, 419, 950, 333, 849, 459, 383, 951, 164,
845, 509, 704, 117, 115, 464, 367, 248, 156, 104, 61, 986, 672, 295, 972, 964,
454, 239, 399, 32, 664, 119, 863, 211, 160, 621, 936, 963, 166, 77, 461, 961,
847, 26, 189, 957, 823, 423, 101, 676, 991, 802, 708, 477, 417, 450, 396, 254,
105, 95, 389, 698, 723, 661, 379, 647, 671, 222, 340, 838, 549, 287, 724, 65,
137, 524, 402, 114, 653, 362, 862, 1000, 877, 583, 811, 21, 142, 966, 702, 318,
411, 629, 762, 173, 841, 892, 469, 186, 428, 451, 297, 673, 937, 721, 357, 303,
15, 593, 197, 44, 152, 274, 669, 727, 909, 617, 217, 478, 364, 363, 894, 259,
147, 22, 988, 577, 103, 138, 532, 547, 377, 401, 968, 39, 332, 404, 213, 687,
640, 64, 16, 341, 576, 437, 485, 456, 326, 109, 449, 953, 316, 134, 427, 475,
826, 980, 315, 444, 914, 861, 859, 144, 205, 269, 917, 482, 512, 623, 854, 764,
33, 361, 568, 107, 193, 132, 343, 967, 975, 591, 898, 601, 227, 684, 382, 879,
93, 720, 860, 230, 421, 666, 683, 360, 342, 626, 499, 959, 175, 121, 806, 897,
453, 872, 194, 886, 800, 339, 631, 912, 947, 25, 272, 649, 184, 471, 483, 500,
939, 919, 204, 534, 496, 169, 381, 100, 246, 618, 799, 850, 199, 827, 565, 529,
330, 718, 432, 587, 820, 722, 325, 662, 224, 14, 941, 663, 874, 228, 805, 697,
407, 434, 422, 245, 281, 263, 506, 633, 344, 463, 163, 43, 476, 931, 590, 795,
517, 305, 731, 685, 690, 882, 875, 415, 507, 298, 143, 150, 508, 438, 368, 665,
371, 356, 730, 310, 930, 255, 701, 0, 42, 934, 974, 905, 159, 276, 313, 73, 607,
548, 236, 2, 11, 317, 420, 619, 913, 846, 692, 987, 63, 572, 306, 249, 262, 648,
```

129, 231, 86, 266, 288, 559, 414, 69, 195, 111, 993, 429, 405, 176, 5, 921, 561, 409, 585, 17, 280, 398, 725, 479, 179, 691, 703, 818, 797, 924, 369, 825, 783, 808, 244, 75, 885, 798, 796, 956, 709, 219, 502, 777, 537, 530, 136, 98, 758, 944, 375, 491, 852, 835, 220, 837, 543, 497, 759, 604, 71, 243, 780, 706, 300, 525, 765, 770, 682, 739, 902, 455, 331, 418, 131, 996, 359, 979, 564, 370, 705, 622, 628, 573, 441, 562, 792, 29, 589, 970, 694, 445, 355, 284, 238, 183, 634, 788, 395, 613, 586, 597, 822, 110, 27, 915, 940, 489, 484, 606, 412, 380, 135, 558, 588, 153, 938, 760, 733, 848, 511, 48, 448, 154, 513, 615, 948, 233, 594, 906, 515, 394, 641, 392, 447, 761, 935, 790, 693, 329, 292, 264, 201, 366, 636, 853, 883, 6, 620, 165, 971, 1, 126, 354, 596, 830, 426, 675, 677, 473, 553, 901, 907, 440, 442, 458, 261, 742, 688, 807, 493, 216, 545, 397, 600, 582, 504, 828, 433, 123, 139, 79, 207, 614, 888, 286, 250, 981, 19, 667, 954, 650, 349, 10, 118, 302, 190, 696, 494, 605, 90, 603, 460, 212, 237, 168, 686, 768, 794, 120, 639, 178, 112, 83, 34, 855, 983, 656, 625, 465, 424, 470, 812, 480, 579, 997, 592, 644, 965, 982, 624, 372, 769, 462, 754, 715, 962, 880, 148, 646, 984, 378, 37, 162, 92, 557, 923, 208, 580, 756, 567, 174, 920, 932, 258, 735, 338, 771, 91, 651, 969, 750, 660, 351, 871, 918, 181, 203, 102, 712, 408, 328, 611, 658, 670, 492, 745, 188, 608, 187, 505, 45, 481, 393, 430, 740, 185, 60, 57, 743, 836, 976, 864, 487, 552, 472, 346, 291, 7, 645, 81, 752, 899, 884, 878, 221, 38, 89, 327, 84, 324, 336, 928, 942, 635, 713, 781, 546, 958, 58, 689, 789, 516, 813, 403, 67, 140, 542, 655, 556, 191, 149, 391, 88, 320, 374, 737, 949, 257, 520, 599, 998, 659, 832, 113, 612, 755, 858, 551, 223, 521, 322, 734, 510, 498, 856, 584, 581, 308, 277, 215, 85, 66, 431, 642, 773, 873, 922, 146, 384, 35, 637, 716, 767, 30, 566, 293, 652, 560, 452, 776, 569, 466, 78, 973, 206, 994, 804, 229, 311, 563, 893, 779, 252, 196, 933, 539, 214, 741, 170, 9, 202, 889, 791, 910, 926, 554, 632, 772, 390, 335, 24, 887, 978, 365, 225, 821, 234, 265, 609, 786, 436, 116, 400, 747, 56, 851, 232, 210, 925, 638, 155, 172, 74, 550, 911, 960, 736, 410, 775, 900, 810, 990, 46, 488, 574, 774, 570, 474, 36, 348, 157, 270, 729, 260, 337, 279, 710, 18, 307, 271, 540, 778, 358, 766, 945, 916, 955, 522, 946, 76, 985, 728, 578, 200, 523, 503, 490, 541, 717, 106, 753, 711, 744, 857, 51, 40, 273, 20, 68, 47, 782, 801, 784, 518, 70, 41, 868, 890, 929, 435, 218, 757, 732, 881, 814, 501, 240, 406, 977, 141, 748, 122, 995, 158, 829, 952, 376, 177, 865, 817, 819, 809, 527, 678, 787, 304, 699, 125, 87, 373, 643, 443, 167, 59, 627, 824, 495, 839, 296, 831, 353, 345, 680, 285, 657, 192, 151, 31, 943, 866, 904, 519, 602, 267, 891, 50, 128, 674, 72, 595, 467, 96, 241, 526, 253, 533, 12, 124, 198, 312, 668, 816, 386, 171, 869, 598, 350, 323, 763, 468, 575, 695, 242, 555, 610, 681, 746, 843, 388, 352, 439, 80, 283, 413, 299, 347, 321, 13, 275, 4, 130, 840, 833, 803, 544, 844, 425, 514, 23, 531, 738, 895, 180, 182, 289, 538, 793, 876, 992, 719, 416, 251, 1001, 867, 457, 54, 82, 53, 268, 536, 528, 133, 161, 247, 127, 97, 714, 235, 108, 679, 8, 314, 654, 785, 751, 908, 282, 294, 486, 815, 94, 145, 319, 896, 55, 630, 28, 290, 209, 700, 707, 927, 903, 989, 309, 301, 278, 726, 99, 571, 385, 62, 52, 387, 616, 999, 842, 870, 256] 5664637.600222828 output_12_1.png

[ ]:

[20]:
```python
import matplotlib.pyplot as plt
import bisect
def reverse(l):
    return float(1) / float(l)

def genentic(iterations, population_number, mutation_rate, elite_size):
    population = first_population(population_number, n)
    progress = []
    stop = True
    stop_cnt = 0
    best = 3000000
    pre = 100000000
    i = 0
    while stop:

        if best < 3000*1000:
            stop = False

        population.sort(key = fitness)
        best = fitness(population[0])

        if (i%100 == 0):
            print(i,mutation_rate, best, fitness(population[1]), end = '\r', flush=True)
        i += 1

        progress.append(best)

        child = []
        # moving elites to next generation
        for j in range(elite_size):
            child.append(population[j])

        # selection based on fitness(normalize fitness)
        select = []
        normalized_fitness = [(reverse(fitness(j))) for j in population]
```

```
        reversed(population)
        sum_fitness = sum(normalized_fitness)
        prefix_sum = [normalized_fitness[0]/sum_fitness]
        for k in range(1, len(population)):
            prefix_sum.append(prefix_sum[k-1] + float(normalized_fitness[k]) /␣
↪float(sum_fitness))

        while(len(select) < population_number - elite_size):
            x = random.random()
            select.append(population[bisect.bisect(prefix_sum, x)])

        # making new child from selected parents
        j = 0
        population.sort(key = fitness)
        while (len(child) < population_number):
            child.append(cross_over(select[j], select[len(select) - 1 - j]))
            j = (j + 1) % len(select)
        # mutate next generation
        for j in range(elite_size, population_number):
            if (random.random() < mutation_rate):
                child[j] = mutation(child[j], mutation_rate)
        population = child

    population.sort(key = fitness)
    print(population[0], fitness(population[0]))

    plt.plot(progress)
    plt.ylabel('Distance')
    plt.xlabel('Generation')
    plt.show()

genentic(500, 10, 0.001, 2)
```

```
[597, 249, 583, 162, 582, 221, 458, 427, 413, 816, 559, 506, 481, 507, 827, 786,
210, 467, 468, 261, 258, 273, 282, 309, 310, 83, 42, 302, 354, 289, 313, 29, 70,
388, 673, 988, 721, 747, 914, 920, 934, 999, 782, 411, 391, 263, 178, 215, 497,
587, 230, 490, 548, 781, 813, 452, 635, 939, 963, 879, 931, 785, 619, 756, 933,
880, 869, 861, 915, 849, 854, 860, 873, 898, 793, 743, 749, 787, 877, 896, 802,
549, 614, 445, 435, 584, 911, 872, 900, 950, 717, 768, 940, 903, 897, 891, 777,
899, 865, 924, 878, 832, 799, 763, 766, 790, 758, 930, 937, 698, 784, 760, 909,
484, 92, 71, 341, 20, 11, 63, 67, 117, 116, 409, 114, 576, 594, 858, 844, 838,
902, 971, 684, 695, 961, 730, 746, 752, 1000, 997, 457, 499, 538, 531, 479, 489,
224, 265, 995, 156, 159, 272, 146, 260, 183, 188, 173, 255, 54, 26, 17, 332, 47,
39, 24, 58, 40, 50, 315, 165, 245, 163, 147, 118, 101, 73, 204, 242, 496, 449,
438, 420, 377, 390, 620, 941, 862, 491, 510, 524, 894, 867, 923, 840, 213, 254,
168, 145, 140, 167, 179, 225, 207, 248, 141, 264, 833, 515, 605, 389, 608, 794,
625, 735, 893, 525, 600, 761, 938, 754, 980, 985, 962, 734, 736, 472, 268, 456,
```

239, 550, 487, 493, 270, 21, 7, 9, 246, 528, 514, 522, 824, 535, 592, 831, 871,
859, 835, 523, 174, 125, 134, 100, 509, 191, 175, 214, 180, 170, 164, 186, 143,
139, 415, 789, 957, 705, 765, 771, 792, 935, 955, 977, 864, 269, 226, 494, 357,
386, 288, 442, 547, 846, 539, 830, 818, 593, 563, 627, 293, 318, 314, 352, 329,
339, 663, 797, 195, 234, 271, 196, 253, 32, 105, 109, 37, 280, 305, 89, 35, 66,
429, 231, 958, 731, 922, 919, 992, 690, 623, 428, 372, 362, 664, 85, 991, 346,
400, 661, 954, 701, 843, 223, 243, 113, 69, 33, 96, 57, 22, 295, 462, 521, 1001,
561, 577, 233, 150, 137, 46, 250, 478, 495, 475, 436, 407, 297, 91, 148, 149,
203, 553, 887, 926, 932, 693, 376, 41, 64, 31, 62, 368, 638, 16, 353, 419, 399,
355, 649, 647, 626, 395, 437, 292, 277, 281, 306, 88, 634, 380, 422, 745, 733,
616, 689, 944, 682, 884, 881, 511, 266, 209, 251, 176, 240, 181, 161, 169, 241,
136, 87, 93, 199, 171, 201, 244, 575, 579, 633, 645, 441, 99, 119, 28, 217, 929,
667, 707, 753, 262, 216, 182, 994, 848, 847, 927, 444, 726, 729, 651, 618, 367,
392, 617, 711, 901, 951, 968, 969, 779, 801, 796, 960, 942, 644, 739, 542, 571,
469, 287, 387, 307, 610, 629, 416, 274, 990, 238, 228, 423, 432, 556, 555, 543,
545, 505, 513, 518, 578, 425, 403, 632, 738, 657, 364, 697, 762, 450, 454, 574,
122, 104, 107, 160, 152, 972, 979, 702, 928, 853, 589, 532, 820, 434, 814, 236,
158, 187, 502, 516, 829, 569, 519, 585, 906, 870, 856, 599, 798, 907, 912, 953,
767, 750, 876, 925, 883, 795, 52, 23, 691, 335, 334, 256, 267, 126, 157, 177,
211, 205, 501, 257, 324, 421, 433, 724, 668, 984, 946, 601, 544, 541, 365, 723,
51, 194, 498, 482, 504, 527, 34, 356, 636, 371, 290, 75, 6, 108, 80, 474, 440,
483, 581, 590, 692, 987, 679, 687, 952, 666, 709, 986, 983, 774, 699, 943, 982,
708, 363, 393, 662, 374, 405, 59, 338, 461, 466, 72, 3, 396, 447, 737, 360, 643,
540, 892, 596, 564, 568, 486, 476, 477, 573, 572, 227, 247, 470, 124, 103, 218,
185, 184, 198, 439, 888, 857, 886, 744, 658, 291, 465, 49, 15, 45, 401, 275,
488, 503, 580, 588, 819, 471, 90, 299, 296, 361, 316, 464, 520, 586, 189, 192,
208, 190, 123, 25, 81, 86, 133, 259, 142, 212, 508, 821, 443, 106, 76, 132, 326,
671, 716, 728, 778, 910, 908, 775, 889, 905, 451, 603, 826, 895, 770, 591, 921,
936, 804, 455, 624, 678, 945, 841, 823, 810, 811, 788, 741, 769, 918, 807, 404,
300, 631, 414, 418, 38, 27, 317, 343, 611, 755, 369, 974, 10, 4, 131, 473, 283,
382, 385, 780, 947, 727, 713, 989, 675, 670, 672, 719, 916, 822, 850, 890, 845,
837, 558, 229, 55, 402, 345, 347, 659, 656, 646, 602, 373, 609, 323, 397, 336,
412, 480, 235, 459, 82, 56, 84, 304, 121, 193, 153, 642, 384, 110, 98, 0, 60,
68, 327, 298, 446, 222, 512, 500, 536, 2, 5, 36, 74, 303, 312, 308, 998, 154,
155, 138, 129, 322, 337, 791, 975, 612, 836, 855, 851, 875, 533, 817, 842, 742,
970, 639, 621, 718, 725, 956, 949, 674, 973, 696, 773, 967, 606, 359, 333, 319,
426, 351, 43, 112, 485, 800, 700, 740, 683, 978, 751, 703, 783, 660, 653, 378,
430, 812, 654, 676, 622, 615, 966, 828, 534, 815, 759, 981, 965, 732, 655, 311,
330, 410, 379, 637, 350, 366, 321, 398, 348, 375, 652, 714, 694, 604, 570, 529,
130, 61, 128, 993, 78, 166, 530, 202, 252, 95, 237, 144, 197, 151, 65, 331, 381,
294, 613, 97, 102, 127, 219, 200, 220, 232, 94, 48, 1, 135, 77, 111, 301, 53,
79, 320, 14, 340, 342, 44, 328, 757, 776, 964, 358, 344, 772, 839, 566, 424,
640, 383, 996, 460, 417, 325, 284, 370, 279, 12, 8, 286, 278, 13, 19, 30, 18,
120, 595, 852, 809, 803, 834, 492, 565, 567, 537, 349, 276, 285, 628, 607, 748,
710, 463, 115, 406, 431, 172, 206, 526, 868, 866, 904, 917, 874, 825, 882, 805,
885, 557, 560, 546, 562, 517, 551, 764, 863, 554, 720, 976, 959, 913, 704, 715,
408, 648, 650, 630, 394, 680, 686, 669, 712, 688, 677, 665, 948, 681, 685, 641,
722, 706, 453, 806, 808, 448, 598, 552] 2998957.6620706366

output_14_1.png