

Proje Tanımı ve Amacı

Bir telekomünikasyon şirketi A ve B şehirlerinde baz istasyonlarının yerleşimini kapsama alanlarına göre optimize etmek istemektedir. Ancak A ve B şehirlerinin aşağıda belirtilen özel durumları bulunmaktadır:

- A şehirde baz istasyonu yerleşimine izin verilen potansiyel noktalar önceden belirlenmiş ve sabittir. Şirket bu potansiyel noktalardan seçtikleri üzerine baz istasyonu kurabilmektedir.
- B şehirde baz istasyonu yerleşim noktaları değişebilmekte olup önceden sınırlandırılmamıştır. Şirket herhangi bir noktaya karar verip baz istasyonunu kurabilmektedir.

Hem müşteriler hem de baz istasyonları 2 boyutlu Öklid düzleminde birer nokta olarak temsil edilmektedir. Bir müşterinin bir baz istasyonu tarafından kapsanıp kapsanmadığına ise aralarındaki uzaklık ile karar verilmektedir. Eğer iki nokta arasındaki Öklid uzaklığı kapsama büyüklüğü olan r değerinden küçük ya da ona eşitse müşteri kapsanmış sayılmaktadır.

İÇERİK

Bölüm 1

Amaç : (0,0) noktası sol üst köşe olmak üzere kullanıcı tarafından parametre olarak verilen genişlik (x eksen boyu) ve uzunluk (y eksen boyu) sınırları içerisinde m adet müşteri ve n adet baz istasyonu yerleşimi noktasını rastgele üreten ve gerekli veri yapılarında saklayan metodu yazınız. Metot ürettiği tüm noktaları konsol çıktısı olarak listelemelidir.

Bu bölümün çözümü **Coordinate.java** sınıfında yapılmıştır. Müşterilerin ve baz istasyonlarının koordinatlarını tutmak için iki boyutlu iki tane dizi tanımlanmıştır. Dizilerin birinci boyutunda random olarak üretilmiş koordinatlar tutuluyor, ikinci boyutunda ise x ve y değişkenleri tutuluyor.

randomGenerator metodu parametre olarak koordinatların üretilebileceği uzunluğu, yüksekliği ve üretilecek müşteri sayısı ile baz istasyonu sayısını parametre olarak alıyor. Verilen parametrelere göre random olarak x ve y değişkeni üretip ayrı döngülerde müşteri ve baz istasyonu dizisine atıyor.

Main.java sınıfında **Coordinate.java** sınıfından bir nesne üretip metodu çağırdığımızda müşterilere ve baz istasyonlarına ait koordinat noktaları elde ediyoruz.

```
Coordinate c = new Coordinate();  
c.randomGenerator(200, 300, 10, 10);
```

Output

```
Baz Yerleşim 0 (123.6, 191.8)  
Baz Yerleşim 1 (40.8, 246.1)  
Baz Yerleşim 2 (92.1, 146.9)  
Baz Yerleşim 3 (160.2, 70.8)  
Baz Yerleşim 4 (116.0, 40.7)  
Baz Yerleşim 5 (42.6, 84.2)  
Baz Yerleşim 6 (73.3, 40.5)  
Baz Yerleşim 7 (36.1, 11.4)  
Baz Yerleşim 8 (154.6, 253.6)  
Baz Yerleşim 9 (143.1, 264.3)  
  
Müşteri 0 (148.5, 84.2)  
Müşteri 1 (139.0, 102.3)  
Müşteri 2 (186.7, 123.7)  
Müşteri 3 (190.4, 31.7)  
Müşteri 4 (63.5, 131.6)  
Müşteri 5 (175.0, 284.7)  
Müşteri 6 (66.6, 185.7)  
Müşteri 7 (38.2, 189.2)  
Müşteri 8 (133.5, 271.9)  
Müşteri 9 (195.4, 18.9)
```

Bölüm 2

Amaç : A şehri için, genişlik=400, uzunluk=400, $m=250$, $n=100$ olmak üzere rastgele yerleşim noktaları üretiniz. Kapsama büyüklüğünü $r = 30$ kabul ederek, kapsanan müşteri sayısını maksimize edecek şekilde 50 adet baz istasyonunu Genetik Algoritma kullanarak yerleştiriniz.

i) Üretilen rastgele noktalara (müşteri ve potansiyel baz istasyonu) ait bilgiler

c.randomGenerator(400, 400, 250, 100); ile verilen parametrelere göre 250 adet müşteri ve 100 adet baz istasyonu için rastgele yerleşim noktaları üretilmiştir.

```
Baz Yerleşim 0 (132.0, 82.6)
Baz Yerleşim 1 (74.0, 62.4)
Baz Yerleşim 2 (304.4, 8.9)
Baz Yerleşim 3 (386.2, 152.1)
Baz Yerleşim 4 (225.3, 10.6)
Baz Yerleşim 5 (383.4, 240.6)
Baz Yerleşim 6 (132.2, 190.3)
Baz Yerleşim 7 (336.5, 252.2)
Baz Yerleşim 8 (291.5, 161.0)
Baz Yerleşim 9 (384.5, 362.3)
Baz Yerleşim 10 (276.7, 19.5)
Baz Yerleşim 11 (183.1, 362.7)
Baz Yerleşim 12 (44.2, 283.1)
Baz Yerleşim 13 (32.0, 156.9)
Baz Yerleşim 14 (87.4, 52.6)
Baz Yerleşim 15 (77.0, 345.5)
Baz Yerleşim 16 (357.9, 360.2)
Baz Yerleşim 17 (193.0, 214.1)
Baz Yerleşim 18 (158.8, 373.3)
Baz Yerleşim 19 (283.2, 3.6)
Baz Yerleşim 20 (249.4, 245.3)
Baz Yerleşim 21 (164.5, 319.6)
Baz Yerleşim 22 (343.0, 339.8)
Baz Yerleşim 23 (360.4, 330.3)
Baz Yerleşim 24 (37.0, 302.2)
Baz Yerleşim 25 (387.5, 4.6)
```

```
Baz Yerleşim 25 (387.5, 4.6)
Baz Yerleşim 26 (183.3, 212.6)
Baz Yerleşim 27 (293.3, 59.3)
Baz Yerleşim 28 (389.3, 225.8)
Baz Yerleşim 29 (295.1, 110.8)
Baz Yerleşim 30 (332.5, 288.2)
Baz Yerleşim 31 (366.6, 370.1)
Baz Yerleşim 32 (279.8, 121.1)
Baz Yerleşim 33 (134.6, 272.3)
Baz Yerleşim 34 (21.3, 66.2)
Baz Yerleşim 35 (270.5, 297.0)
Baz Yerleşim 36 (159.3, 15.0)
Baz Yerleşim 37 (103.1, 116.2)
Baz Yerleşim 38 (139.0, 75.6)
Baz Yerleşim 39 (25.3, 296.2)
Baz Yerleşim 40 (361.5, 79.1)
Baz Yerleşim 41 (234.2, 83.3)
Baz Yerleşim 42 (392.3, 287.9)
Baz Yerleşim 43 (215.8, 320.8)
Baz Yerleşim 44 (55.9, 339.0)
Baz Yerleşim 45 (176.6, 270.5)
Baz Yerleşim 46 (305.6, 342.0)
Baz Yerleşim 47 (322.9, 56.0)
Baz Yerleşim 48 (161.0, 347.3)
Baz Yerleşim 49 (101.8, 196.4)
Baz Yerleşim 50 (326.0, 109.5)
```

```
Baz Yerleşim 76 (48.4, 139.4)
Baz Yerleşim 77 (153.7, 314.7)
Baz Yerleşim 78 (89.6, 322.4)
Baz Yerleşim 79 (192.1, 88.3)
Baz Yerleşim 80 (260.4, 235.7)
Baz Yerleşim 81 (305.4, 11.2)
Baz Yerleşim 82 (112.3, 96.0)
Baz Yerleşim 83 (162.7, 244.9)
Baz Yerleşim 84 (259.5, 383.1)
Baz Yerleşim 85 (167.2, 56.3)
Baz Yerleşim 86 (173.5, 388.4)
Baz Yerleşim 87 (347.5, 61.5)
Baz Yerleşim 88 (163.8, 331.4)
Baz Yerleşim 89 (249.9, 34.0)
Baz Yerleşim 90 (165.3, 187.4)
Baz Yerleşim 91 (206.3, 261.2)
Baz Yerleşim 92 (234.1, 148.1)
Baz Yerleşim 93 (237.0, 54.6)
Baz Yerleşim 94 (48.0, 14.5)
Baz Yerleşim 95 (248.7, 267.7)
Baz Yerleşim 96 (16.1, 225.7)
Baz Yerleşim 97 (101.7, 321.8)
Baz Yerleşim 98 (17.7, 32.4)
Baz Yerleşim 99 (145.7, 312.6)
```

Baz Yerleşim 76 (48.4, 139.4)
Baz Yerleşim 77 (153.7, 314.7)
Baz Yerleşim 78 (89.6, 322.4)
Baz Yerleşim 79 (192.1, 88.3)
Baz Yerleşim 80 (260.4, 235.7)
Baz Yerleşim 81 (305.4, 11.2)
Baz Yerleşim 82 (112.3, 96.0)
Baz Yerleşim 83 (162.7, 244.9)
Baz Yerleşim 84 (259.5, 383.1)
Baz Yerleşim 85 (167.2, 56.3)
Baz Yerleşim 86 (173.5, 388.4)
Baz Yerleşim 87 (347.5, 61.5)
Baz Yerleşim 88 (163.8, 331.4)
Baz Yerleşim 89 (249.9, 34.0)
Baz Yerleşim 90 (165.3, 187.4)
Baz Yerleşim 91 (206.3, 261.2)
Baz Yerleşim 92 (234.1, 148.1)
Baz Yerleşim 93 (237.0, 54.6)
Baz Yerleşim 94 (48.0, 14.5)
Baz Yerleşim 95 (248.7, 267.7)
Baz Yerleşim 96 (16.1, 225.7)
Baz Yerleşim 97 (101.7, 321.8)
Baz Yerleşim 98 (17.7, 32.4)
Baz Yerleşim 99 (145.7, 312.6)

Müşteri 0 (126.0, 146.9)
Müşteri 1 (360.2, 43.0)
Müşteri 2 (113.5, 38.2)
Müşteri 3 (86.7, 124.1)
Müşteri 4 (294.2, 285.8)
Müşteri 5 (103.9, 384.6)
Müşteri 6 (126.3, 66.5)
Müşteri 7 (69.9, 149.6)
Müşteri 8 (180.4, 121.0)
Müşteri 9 (346.2, 274.3)
Müşteri 10 (373.9, 31.3)
Müşteri 11 (316.1, 162.8)
Müşteri 12 (375.3, 137.3)
Müşteri 13 (325.2, 351.6)
Müşteri 14 (7.3, 131.5)
Müşteri 15 (94.8, 207.9)
Müşteri 16 (301.8, 291.7)
Müşteri 17 (99.9, 67.8)
Müşteri 18 (11.1, 247.3)
Müşteri 19 (350.6, 131.5)
Müşteri 20 (230.6, 122.0)
Müşteri 21 (80.6, 238.0)
Müşteri 22 (162.2, 254.6)
Müşteri 23 (21.6, 54.2)
Müşteri 24 (323.9, 73.3)

Müşteri 25 (230.4, 245.4)
Müşteri 26 (111.9, 82.4)
Müşteri 27 (34.8, 233.4)
Müşteri 28 (121.0, 118.6)
Müşteri 29 (179.0, 400.7)
Müşteri 30 (54.4, 41.9)
Müşteri 31 (272.3, 69.4)
Müşteri 32 (380.9, 251.9)
Müşteri 33 (386.6, 56.9)
Müşteri 34 (145.9, 51.3)
Müşteri 35 (93.0, 105.3)
Müşteri 36 (94.4, 327.9)
Müşteri 37 (197.0, 182.8)
Müşteri 38 (338.8, 85.7)
Müşteri 39 (301.6, 172.9)
Müşteri 40 (380.9, 27.4)
Müşteri 41 (319.7, 394.2)
Müşteri 42 (395.0, 276.3)
Müşteri 43 (153.9, 136.1)
Müşteri 44 (284.2, 222.3)
Müşteri 45 (179.3, 147.6)
Müşteri 46 (341.6, 386.3)
Müşteri 47 (285.0, 184.0)
Müşteri 48 (379.8, 319.6)
Müşteri 49 (271.7, 294.6)
Müşteri 50 (134.7, 91.0)

Müşteri 51 (152.8, 123.3)
Müşteri 52 (209.9, 258.6)
Müşteri 53 (166.1, 101.4)
Müşteri 54 (192.7, 387.0)
Müşteri 55 (283.7, 24.0)
Müşteri 56 (57.9, 338.8)
Müşteri 57 (91.2, 133.1)
Müşteri 58 (166.8, 158.0)
Müşteri 59 (109.6, 268.7)
Müşteri 60 (78.8, 155.8)
Müşteri 61 (146.3, 305.4)
Müşteri 62 (208.7, 24.2)
Müşteri 63 (286.6, 356.3)
Müşteri 64 (297.2, 20.0)
Müşteri 65 (74.5, 104.3)
Müşteri 66 (303.6, 100.4)
Müşteri 67 (83.4, 313.8)
Müşteri 68 (289.4, 336.4)
Müşteri 69 (238.1, 76.7)
Müşteri 70 (128.5, 391.1)
Müşteri 71 (128.1, 265.7)
Müşteri 72 (261.0, 121.3)
Müşteri 73 (135.0, 192.3)
Müşteri 74 (3.9, 247.6)
Müşteri 75 (104.1, 213.6)
Müşteri 76 (136.6, 331.9)
Müşteri 77 (95.4, 378.6)

Müşteri 78 (399.4, 215.0)
Müşteri 79 (282.4, 177.5)
Müşteri 80 (174.1, 186.6)
Müşteri 81 (297.5, 230.9)
Müşteri 82 (360.2, 94.9)
Müşteri 83 (280.3, 149.8)
Müşteri 84 (317.9, 352.0)
Müşteri 85 (141.4, 199.1)
Müşteri 86 (115.9, 349.7)
Müşteri 87 (135.4, 214.7)
Müşteri 88 (5.7, 172.1)
Müşteri 89 (320.3, 251.7)
Müşteri 90 (38.0, 77.5)
Müşteri 91 (83.8, 82.9)
Müşteri 92 (242.0, 218.7)
Müşteri 93 (28.2, 119.8)
Müşteri 94 (208.9, 80.8)
Müşteri 95 (331.8, 194.8)
Müşteri 96 (253.1, 336.9)
Müşteri 97 (242.8, 122.0)
Müşteri 98 (166.1, 58.5)
Müşteri 99 (210.4, 243.2)
Müşteri 100 (337.8, 263.9)
Müşteri 101 (123.4, 72.5)
Müşteri 102 (87.3, 246.0)

Müşteri 103 (200.1, 265.8)
Müşteri 104 (226.9, 397.6)
Müşteri 105 (61.1, 271.9)
Müşteri 106 (378.5, 330.7)
Müşteri 107 (17.9, 361.6)
Müşteri 108 (366.4, 390.6)
Müşteri 109 (223.1, 286.0)
Müşteri 110 (330.9, 169.1)
Müşteri 111 (52.7, 380.5)
Müşteri 112 (38.5, 147.3)
Müşteri 113 (328.1, 109.7)
Müşteri 114 (208.7, 132.6)
Müşteri 115 (339.9, 238.6)
Müşteri 116 (333.5, 299.6)
Müşteri 117 (91.7, 21.3)
Müşteri 118 (34.7, 207.8)
Müşteri 119 (163.9, 10.9)
Müşteri 120 (88.2, 138.4)
Müşteri 121 (282.5, 326.8)
Müşteri 122 (34.3, 278.6)
Müşteri 123 (276.9, 252.7)
Müşteri 124 (203.6, 278.3)
Müşteri 125 (27.2, 295.8)
Müşteri 126 (205.2, 116.0)
Müşteri 127 (224.0, 300.9)
Müşteri 128 (255.9, 229.3)

Müşteri 129 (102.5, 91.9)
Müşteri 130 (174.1, 103.0)
Müşteri 131 (163.3, 7.8)
Müşteri 132 (216.4, 355.5)
Müşteri 133 (55.0, 397.2)
Müşteri 134 (207.4, 220.1)
Müşteri 135 (269.5, 58.1)
Müşteri 136 (192.3, 195.3)
Müşteri 137 (286.9, 134.1)
Müşteri 138 (336.4, 118.7)
Müşteri 139 (111.3, 120.3)
Müşteri 140 (3.9, 19.1)
Müşteri 141 (346.4, 385.6)
Müşteri 142 (264.7, 9.0)
Müşteri 143 (90.3, 185.5)
Müşteri 144 (112.3, 203.9)
Müşteri 145 (362.8, 314.6)
Müşteri 146 (18.9, 188.6)
Müşteri 147 (331.4, 143.9)
Müşteri 148 (309.7, 297.8)
Müşteri 149 (264.5, 185.5)
Müşteri 150 (208.8, 338.5)
Müşteri 151 (113.6, 350.9)
Müşteri 152 (290.0, 153.8)
Müşteri 153 (319.4, 288.1)
Müşteri 154 (60.0, 212.8)

Müşteri 155 (229.6, 34.7)
Müşteri 156 (309.4, 7.7)
Müşteri 157 (236.8, 266.3)
Müşteri 158 (282.8, 249.7)
Müşteri 159 (364.3, 96.5)
Müşteri 160 (255.1, 77.8)
Müşteri 161 (169.2, 253.0)
Müşteri 162 (342.4, 328.5)
Müşteri 163 (80.5, 155.0)
Müşteri 164 (333.0, 362.1)
Müşteri 165 (254.2, 112.1)
Müşteri 166 (194.1, 201.5)
Müşteri 167 (101.3, 187.1)
Müşteri 168 (290.7, 210.3)
Müşteri 169 (234.5, 390.5)
Müşteri 170 (247.0, 24.7)
Müşteri 171 (277.6, 15.8)
Müşteri 172 (188.4, 146.2)
Müşteri 173 (396.0, 280.2)
Müşteri 174 (220.6, 316.9)
Müşteri 175 (331.2, 397.4)
Müşteri 176 (232.8, 118.4)
Müşteri 177 (361.7, 393.1)
Müşteri 178 (59.4, 370.6)
Müşteri 179 (360.2, 359.0)
Müşteri 180 (87.1, 54.7)

Müşteri 181 (382.5, 249.5)
Müşteri 182 (167.2, 243.8)
Müşteri 183 (198.4, 338.4)
Müşteri 184 (171.7, 297.2)
Müşteri 185 (261.5, 82.2)
Müşteri 186 (327.8, 358.2)
Müşteri 187 (16.2, 377.9)
Müşteri 188 (370.4, 394.8)
Müşteri 189 (353.4, 257.2)
Müşteri 190 (374.2, 29.5)
Müşteri 191 (251.6, 322.8)
Müşteri 192 (54.2, 29.3)
Müşteri 193 (375.8, 113.9)
Müşteri 194 (171.9, 358.9)
Müşteri 195 (262.3, 277.4)
Müşteri 196 (238.7, 352.2)
Müşteri 197 (273.1, 124.4)
Müşteri 198 (178.0, 209.3)
Müşteri 199 (386.3, 69.3)
Müşteri 200 (388.2, 108.1)
Müşteri 201 (271.1, 175.3)
Müşteri 202 (344.0, 94.7)
Müşteri 203 (200.2, 173.2)
Müşteri 204 (80.7, 221.8)
Müşteri 205 (162.1, 71.9)
Müşteri 206 (22.0, 109.3)

Müşteri 207 (265.3, 246.7)
Müşteri 208 (248.2, 225.5)
Müşteri 209 (358.7, 68.4)
Müşteri 210 (41.6, 122.0)
Müşteri 211 (349.4, 329.4)
Müşteri 212 (5.5, 2.3)
Müşteri 213 (155.9, 123.5)
Müşteri 214 (220.2, 163.4)
Müşteri 215 (14.2, 256.0)
Müşteri 216 (382.8, 23.4)
Müşteri 217 (341.0, 284.4)
Müşteri 218 (70.8, 231.5)
Müşteri 219 (356.9, 136.2)
Müşteri 220 (375.3, 154.5)
Müşteri 221 (189.2, 300.0)
Müşteri 222 (171.8, 43.8)
Müşteri 223 (194.8, 52.4)
Müşteri 224 (23.7, 9.2)
Müşteri 225 (185.0, 265.7)
Müşteri 226 (162.3, 160.4)
Müşteri 227 (83.1, 26.0)
Müşteri 228 (140.5, 324.3)
Müşteri 229 (308.0, 200.6)
Müşteri 230 (227.7, 6.4)
Müşteri 231 (53.7, 205.9)
Müşteri 232 (395.7, 135.7)

Müşteri 233 (222.0, 158.3)
Müşteri 234 (238.6, 19.5)
Müşteri 235 (264.6, 128.3)
Müşteri 236 (375.6, 229.2)
Müşteri 237 (207.1, 174.8)
Müşteri 238 (8.2, 49.3)
Müşteri 239 (248.9, 1.2)
Müşteri 240 (329.4, 244.7)
Müşteri 241 (173.3, 252.3)
Müşteri 242 (232.0, 234.6)
Müşteri 243 (152.9, 264.5)
Müşteri 244 (17.2, 109.2)
Müşteri 245 (194.9, 282.8)
Müşteri 246 (172.6, 286.3)
Müşteri 247 (251.6, 176.7)
Müşteri 248 (174.0, 175.1)
Müşteri 249 (104.0, 311.9)

ii) Problem için en uygun kromozom (çözüm adayı) yapısı tasarımı

A şehrinde 100 adet yerleşim yerine 50 baz istasyonunun yerleştirilmesi istenmektedir. Bu da problemin binary string (1'ler ve 0'lar) şeklinde tanımlanabilmesine olanak tanımaktadır. Böylece kromozomları 100 uzunluklu bit dizisi olarak tanımlayabiliriz. 1 olan yerleşim yerine baz istasyonu kurulmuş, 0 olan yere ise kurulmamış sayılacaktır.

Örnek bir kromozom yapısı

```
000010011100001000010011000111110100000001001011100100011110101111101011111101110010001111010000
```

iii) Uygunluk (fitness) değerini hesaplayan metot

Uygunluk değerini hesaplamak için öklid uzaklık formülü kullanıldı. Örneğin $(x_1=10.5, y_1=20.7)$ noktası müşterinin konumunu, $(x_2=3.1, y_2=19.9)$ noktası ise baz istasyonu konumunu veriyor olsun. İki nokta arasındaki Öklid uzaklığı şu şekilde hesaplanabilmektedir:

$$uzaklık = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \cong 7.44$$

- Kromozom yapısında 1 olan bitlerin yerine baz istasyonu kurulmuş olduğunu varsaymıştık. Öklid formülünden yola çıkarak kromozomda 1 olan bitlerde bulunan baz istasyonları ile müşteri yerleri arasındaki uzaklığı hesaplıyoruz. Eğer bu uzaklık kapsama alanı olarak verilen r değişkeninden küçükse müşteri baz istasyonu tarafından kapsanıyor demektir ve bu durumda uygunluğu tuttuğumuz **totalScore** değişkenini artırmalıyız.
- Bir müşteri birden fazla baz istasyonu tarafından kapsanabilmektedir. Bunun uygunluk değerini ve optimizasyonu etkilememesi açısından totalScore'un artırıldığı kısımdan break komutu ile çıkmalıyız.
- Hazır Watchmaker kütüphanesi kullanıldığı için uygunluk (fitness) değerini **FitnessEvaluator** sınıfından **BitString** tipinde extend edilmiş **BitStringEvaluator** sınıfında uygunluk değerini hesaplayan metodu yazıyoruz.
- Buradaki **getFitness** fonksiyonuna BitString tipinde kromozom alan **candidate** bit dizisi ve populasyon parametre olarak veriliyor.
- Biz burada 50 adet baz istasyonu optimize edeceğimiz için 100 uzunluklu bit dizisi olan kromozomlarımızdaki 1 bitlerinin sayısı hep 50 olmalıdır. Bunu kontrol altına alabilmek için uygunluk fonksiyonu hep 1 bit sayısı 50 olan bireylerde hesaplanıyor.
- Soruda önceden verilen r kapsama büyüklüğü değerini double tipinde range değişkenine atıp müşteri ile baz istasyonu arasında bulunan uzak bu değer ile kıyaslanıyor.

isNatural metodu ise uygunluk değerin minimize ya da maksimize edileceğini belirler. Eğer **true** döndürülürse maksimize, **false** döndürülürse minimizasyon yapılacağı anlaşılmaktadır. Bizim bu problemde amacımız kapsanan müşteri sayısını maksimize etmek olduğu için true döndürülecek.

iv) Genetik Algoritma Tasarımı - Watchmaker

- Genetik algoritma için **Watchmaker** kütüphanesi kullanılmıştır. Burada kullandığımız sınıf ve metotlar hazır olarak eklenmiştir.
- Integer tipinde **length** değişkenine kromozom için bit dizisinin uzunluğu veriliyor, bu problem boyutunu belirtir.
- **EvolutionaryOperator** kendisine verilen bir popülasyonu değişime uğratarak geri döndürme fonksiyonunu üstlenir. Bu tipte bir **operators listesi** oluşturup operators listesine kütüphane içerisinde hazır olan **BitStringCrossover** ve **BitStringMutation** ekliyoruz. BitStringCrossover'a parametre olarak kaç noktadan çaprazlama yapılacağını ve olasılığını veriyoruz. BitStringMutation'a da parametre olarak olasılık veriliyor.
- Operators listesini pipeline nesnesine aktarıyoruz. **EvolutionPipeline** operatörlerin sırasını belirtir. Yani her seferinde önce çaprazlama sonra mutasyon yapılmasını sağlayacak. Bu pipeline **GenerationalEvolutionEngine** oluşturulurken yapılandırıcı metoduna veriliyor.
- **EvolutionEngine** nesnesini oluşturmak için **GenerationalEvolutionEngine** kullanıyoruz.
- **GenerationalEvolutionEngine**'de tüm çocuk bireyler tüm ata bireylerden oluşturulup jenerasyon bittikten sonra çocuk bireyler ile ata bireyler yer değiştiriyor.
- **BitStringFactory** çözümleri rastgele oluşturmak için kullanılıyor. Parametre olarak length değişkenini alır. O uzunlukta rastgele bit dizisi oluşturur.
- Uygunluk değerini hesapladığımız **BitStringEvaluator** sınıfından **fitnessFunction** nesnesi oluşturarak **GenerationalEvolutionEngine**'e veriyoruz.
- Diğer bir parametre olarak **SelectionStrategy** sınıfından türetilmiş olan parent seçim mekanizması sınıflarından birini veriyoruz. (InteractiveSelection, RankSelection, RouletteWheelSelection, SigmaScaling, StochasticUniversalSampling, TournamentSelection, TruncationSelection).
- Son olarak da random sayı üretici ekliyoruz. **Random** fonksiyonuna belli bir sayıda çalıştırıp ortalama değerini alabilmek için run parametresi veriyoruz. Aynı zamanda SEED görevi görüp program her çalıştığında random sayıların aynı üretilmesini sağlıyoruz.
- **EvolutionEngine** nesnesinin **evolve** metoduna popülasyon sayısı, elitizm ve sonlanma koşulu parametrelerini vererek optimizasyon yapılmasını sağlıyoruz. **Elitizm** değerine 1 değeri vererek popülasyondaki en iyi bireyin bir sonraki jenerasyona aktarılmasını sağlıyoruz. Sonlanma koşulu olarak **GenerationCount** ile maksimum iterasyon sayısını veriyoruz.

v) Deneysel Çalışmalar ve İstatistiksel Yorum

Parametre Seti 1 : populationSize = 50, Selection = RouletteWheelSelection

Parametre Seti 2 : populationSize = 100, Selection = RouletteWheelSelection

Parametre Seti 3 : populationSize = 200, Selection = RouletteWheelSelection

Parametre Seti 4 : populationSize = 50, Selection = RankSelection

Parametre Seti 5 : populationSize = 100, Selection = RankSelection

Parametre Seti 6 : populationSize = 50, Selection = StochasticUniversalSampling

Parametre Seti 7 : populationSize = 100, Selection = StochasticUniversalSampling

	Parametre Seti 1	Parametre Seti 2	Parametre Seti 3	Parametre Seti 4	Parametre Seti 5	Parametre Seti 6	Parametre Seti 7
run 1	146.0	168.0	172.0	167.0	166.0	175.0	160.0
run 2	150.0	148.0	163.0	161.0	155.0	159.0	156.0
run 3	162.0	164.0	176.0	178.0	158.0	162.0	166.0
run 4	146.0	160.0	164.0	162.0	165.0	169.0	157.0
run 5	160.0	172.0	166.0	176.0	166.0	173.0	170.0
run 6	154.0	173.0	169.0	171.0	161.0	164.0	170.0
run 7	160.0	166.0	166.0	175.0	161.0	168.0	163.0
run 8	154.0	172.0	171.0	174.0	164.0	170.0	157.0
run 9	159.0	164.0	165.0	180.0	158.0	163.0	155.0
run 10	143.0	166.0	168.0	168.0	160.0	144.0	158.0

run 11	153.0	175.0	162.0	179.0	164.0	152.0	162.0
run 12	162.0	173.0	172.0	170.0	156.0	162.0	168.0
run 13	156.0	173.0	169.0	173.0	165.0	162.0	159.0
run 14	160.0	177.0	176.0	190.0	161.0	165.0	156.0
run 15	159.0	175.0	169.0	182.0	154.0	171.0	163.0
run 16	155.0	162.0	158.0	175.0	163.0	171.0	156.0
run 17	156.0	164.0	168.0	169.0	155.0	172.0	159.0
run 18	152.0	165.0	171.0	177.0	171.0	158.0	162.0
run 19	150.0	167.0	171.0	167.0	164.0	157.0	163.0
run 20	165.0	168.0	171.0	169.0	164.0	161.0	159.0
run 21	144.0	160.0	163.0	157.0	160.0	159.0	150.0
run 22	160.0	166.0	161.0	166.0	160.0	172.0	167.0
run 23	155.0	166.0	169.0	168.0	166.0	164.0	162.0
run 24	152.0	174.0	170.0	173.0	154.0	161.0	167.0
run 25	149.0	170.0	172.0	169.0	165.0	169.0	157.0
Ortalama	154.48	167.52	168.08	171.84	161.44	164.12	160.88
Standart Sapma	5.872	6.191	4.417	7.023	4.355	7.044	4.998
En iyi	165.0	177.0	176.0	190.0	171.0	175.0	170.0
En kötü	143.0	148.0	158.0	157.0	154.0	144.0	150.0
Medyan	155.0	167.0	169.0	171.0	161.0	164.0	160.0

Bu deneylerin hepsinde maksimum iterasyon sayısı 10000 verilmiştir. Çözüm 25 kez çalıştırılıp çıktı değerlerinin ortalaması alınmıştır. Çözümlerin ortalaması üzerinden yorumlamak rastgele değerler arasındaki farklılıklar sebebiyle daha sağlıklı bir yöntemdir. Seçim yöntemlerinden RouletteWheelSelection'a baktığımızda popülasyon büyüklüğündeki artışın çözümü olumlu etkilediği görülüyor. RankSelection'a baktığımızda popülasyon büyüklüğünün artması çözümü olumsuz etkiliyor. Popülasyon büyüklüğünü hepsinde 50 olarak verdiğimizde ise seçim yöntemleri arasında en iyi uygunluk ortalaması veren yöntemin RankSelection olduğu görülüyor. Fakat popülasyon büyüklüğünü 100 olacak şekilde arttırdığımızda RouletteWheelSelection yönteminin diğerlerine göre daha iyi olduğu görülüyor.

Bölüm 3

Amaç: B şehri için, genişlik=500, uzunluk=500, $m=300$, $n=0$ olmak üzere rastgele müşteri noktaları üretiniz (B şehrinde önceden tanımlı baz istasyonu yerleşim noktaları üretilmeyecektir, çünkü istenildiği yere koyulabilme seçeneği vardır. Bu nedenle $n=0$ alınmıştır.). Kapsama büyüklüğünü $r = 35$ kabul ederek, kapsanan müşteri sayısını maksimize edecek şekilde 50 adet baz istasyonunu seçtiğiniz bir sürekli optimizasyon (reel parametre optimizasyonu) algoritması ile en uygun konumlara yerleştiriniz.

i) Üretilen rastgele müşteri noktalarına ait bilgiler

`c.randomGenerator(500, 500, 300, 0);` ile verilen parametrelere göre 300 adet müşteri için rastgele yerleşim noktaları üretilmiştir.

```
Müşteri 0 (277.4, 67.1)
Müşteri 1 (89.2, 248.1)
Müşteri 2 (80.5, 174.8)
Müşteri 3 (376.5, 418.1)
Müşteri 4 (412.6, 428.3)
Müşteri 5 (183.1, 139.0)
Müşteri 6 (249.5, 50.0)
Müşteri 7 (338.5, 157.4)
Müşteri 8 (37.0, 402.4)
Müşteri 9 (266.0, 306.8)
Müşteri 10 (318.3, 247.7)
Müşteri 11 (327.7, 435.7)
Müşteri 12 (415.2, 228.8)
Müşteri 13 (124.9, 221.1)
Müşteri 14 (321.8, 254.1)
Müşteri 15 (316.3, 151.0)
Müşteri 16 (190.7, 238.3)
Müşteri 17 (123.3, 427.2)
Müşteri 18 (401.9, 437.2)
Müşteri 19 (429.6, 265.5)
Müşteri 20 (202.8, 416.8)
Müşteri 21 (270.3, 74.3)
Müşteri 22 (301.7, 11.4)
Müşteri 23 (24.9, 248.2)
Müşteri 24 (76.7, 460.5)
```

```
Müşteri 25 (297.9, 344.1)
Müşteri 26 (235.4, 261.2)
Müşteri 27 (271.3, 202.0)
Müşteri 28 (323.1, 136.3)
Müşteri 29 (263.4, 127.2)
Müşteri 30 (432.5, 398.7)
Müşteri 31 (310.6, 82.0)
Müşteri 32 (308.2, 387.1)
Müşteri 33 (319.8, 417.6)
Müşteri 34 (280.3, 57.9)
Müşteri 35 (203.9, 40.1)
Müşteri 36 (80.2, 246.7)
Müşteri 37 (422.1, 20.8)
Müşteri 38 (113.2, 52.0)
Müşteri 39 (495.7, 51.7)
Müşteri 40 (165.4, 253.1)
Müşteri 41 (361.2, 296.6)
Müşteri 42 (323.1, 146.6)
Müşteri 43 (347.9, 110.6)
Müşteri 44 (67.4, 175.2)
Müşteri 45 (144.4, 281.6)
Müşteri 46 (89.0, 225.3)
Müşteri 47 (325.6, 397.8)
Müşteri 48 (333.3, 492.7)
Müşteri 49 (363.2, 163.0)
```

```
Müşteri 50 (150.6, 349.0)
Müşteri 51 (40.3, 458.4)
Müşteri 52 (171.6, 192.5)
Müşteri 53 (317.7, 199.1)
Müşteri 54 (174.0, 337.3)
Müşteri 55 (478.4, 45.6)
Müşteri 56 (234.3, 72.0)
Müşteri 57 (289.0, 306.3)
Müşteri 58 (383.5, 122.5)
Müşteri 59 (471.8, 413.1)
Müşteri 60 (297.9, 363.4)
Müşteri 61 (478.4, 159.7)
Müşteri 62 (258.3, 254.0)
Müşteri 63 (390.7, 466.2)
Müşteri 64 (207.2, 118.8)
Müşteri 65 (254.1, 277.4)
Müşteri 66 (218.2, 54.7)
Müşteri 67 (262.2, 302.9)
Müşteri 68 (65.1, 420.0)
Müşteri 69 (475.1, 183.8)
Müşteri 70 (347.1, 57.0)
Müşteri 71 (146.9, 304.7)
Müşteri 72 (282.2, 102.7)
Müşteri 73 (283.4, 373.4)
Müşteri 74 (327.0, 469.0)
Müşteri 75 (226.2, 59.9)
```

Müşteri 76 (221.6, 309.8)
Müşteri 77 (382.7, 419.8)
Müşteri 78 (383.9, 49.2)
Müşteri 79 (183.1, 26.4)
Müşteri 80 (300.3, 312.9)
Müşteri 81 (445.9, 202.0)
Müşteri 82 (166.1, 391.5)
Müşteri 83 (78.4, 469.4)
Müşteri 84 (32.8, 166.7)
Müşteri 85 (419.1, 341.8)
Müşteri 86 (106.7, 246.0)
Müşteri 87 (55.7, 330.1)
Müşteri 88 (130.0, 385.6)
Müşteri 89 (115.2, 75.7)
Müşteri 90 (442.3, 129.8)
Müşteri 91 (327.6, 310.3)
Müşteri 92 (388.1, 456.8)
Müşteri 93 (113.2, 403.2)
Müşteri 94 (187.1, 102.9)
Müşteri 95 (24.5, 79.6)
Müşteri 96 (228.9, 129.9)
Müşteri 97 (107.8, 486.9)
Müşteri 98 (487.4, 212.3)
Müşteri 99 (110.9, 139.5)
Müşteri 100 (379.4, 47.0)
Müşteri 101 (475.3, 395.1)

Müşteri 102 (402.8, 230.1)
Müşteri 103 (199.4, 2.3)
Müşteri 104 (300.4, 402.5)
Müşteri 105 (448.5, 412.4)
Müşteri 106 (291.3, 439.4)
Müşteri 107 (216.8, 467.6)
Müşteri 108 (100.3, 448.2)
Müşteri 109 (419.5, 259.7)
Müşteri 110 (93.4, 342.6)
Müşteri 111 (2.9, 405.4)
Müşteri 112 (297.6, 281.4)
Müşteri 113 (11.6, 198.0)
Müşteri 114 (442.8, 296.1)
Müşteri 115 (47.9, 78.0)
Müşteri 116 (415.3, 482.2)
Müşteri 117 (188.0, 108.9)
Müşteri 118 (149.2, 197.1)
Müşteri 119 (196.0, 160.0)
Müşteri 120 (21.5, 288.2)
Müşteri 121 (290.7, 273.8)
Müşteri 122 (229.3, 475.1)
Müşteri 123 (273.6, 238.5)
Müşteri 124 (450.3, 121.9)
Müşteri 125 (26.7, 198.0)
Müşteri 126 (283.5, 265.0)
Müşteri 127 (219.0, 202.7)

Müşteri 128 (65.8, 409.4)
Müşteri 129 (43.8, 448.6)
Müşteri 130 (143.5, 262.6)
Müşteri 131 (421.4, 55.1)
Müşteri 132 (54.2, 255.9)
Müşteri 133 (364.0, 104.3)
Müşteri 134 (337.7, 392.0)
Müşteri 135 (96.7, 304.0)
Müşteri 136 (356.9, 166.6)
Müşteri 137 (66.9, 35.4)
Müşteri 138 (50.0, 34.0)
Müşteri 139 (157.9, 218.6)
Müşteri 140 (479.8, 127.1)
Müşteri 141 (240.5, 361.3)
Müşteri 142 (369.9, 194.0)
Müşteri 143 (157.8, 356.0)
Müşteri 144 (39.9, 485.1)
Müşteri 145 (343.0, 56.1)
Müşteri 146 (258.3, 111.8)
Müşteri 147 (10.1, 393.8)
Müşteri 148 (36.9, 194.2)
Müşteri 149 (241.1, 296.9)
Müşteri 150 (276.7, 438.6)
Müşteri 151 (319.4, 425.5)
Müşteri 152 (459.9, 112.3)
Müşteri 153 (319.7, 277.0)

Müşteri 153 (319.7, 277.0)
Müşteri 154 (255.4, 234.8)
Müşteri 155 (274.5, 1.5)
Müşteri 156 (465.5, 167.8)
Müşteri 157 (178.9, 328.0)
Müşteri 158 (197.9, 164.3)
Müşteri 159 (185.1, 326.5)
Müşteri 160 (489.8, 333.3)
Müşteri 161 (62.1, 323.7)
Müşteri 162 (346.1, 388.2)
Müşteri 163 (367.4, 142.5)
Müşteri 164 (279.3, 470.5)
Müşteri 165 (281.3, 459.1)
Müşteri 166 (455.0, 284.6)
Müşteri 167 (208.2, 459.5)
Müşteri 168 (452.8, 237.2)
Müşteri 169 (34.1, 126.9)
Müşteri 170 (451.1, 245.4)
Müşteri 171 (363.4, 414.5)
Müşteri 172 (330.5, 369.8)
Müşteri 173 (127.7, 236.7)
Müşteri 174 (272.1, 429.5)
Müşteri 175 (169.2, 168.1)
Müşteri 176 (391.4, 280.7)
Müşteri 177 (255.4, 447.4)
Müşteri 178 (313.9, 470.9)

Müşteri 179 (474.0, 371.8)
Müşteri 180 (345.2, 36.0)
Müşteri 181 (89.5, 408.5)
Müşteri 182 (30.7, 272.7)
Müşteri 183 (217.5, 364.0)
Müşteri 184 (431.4, 401.6)
Müşteri 185 (86.7, 231.6)
Müşteri 186 (365.7, 134.6)
Müşteri 187 (215.4, 377.5)
Müşteri 188 (179.2, 273.4)
Müşteri 189 (21.6, 37.2)
Müşteri 190 (307.4, 95.4)
Müşteri 191 (227.6, 168.6)
Müşteri 192 (492.6, 430.3)
Müşteri 193 (390.9, 47.4)
Müşteri 194 (498.7, 487.7)
Müşteri 195 (15.9, 97.7)
Müşteri 196 (44.8, 174.9)
Müşteri 197 (456.7, 320.7)
Müşteri 198 (444.8, 336.4)
Müşteri 199 (220.5, 323.2)
Müşteri 200 (94.2, 50.8)
Müşteri 201 (321.3, 349.5)
Müşteri 202 (496.7, 349.2)
Müşteri 203 (12.4, 389.0)
Müşteri 204 (458.2, 451.5)

Müşteri 205 (50.2, 365.6)
Müşteri 206 (405.6, 378.4)
Müşteri 207 (98.3, 115.8)
Müşteri 208 (6.3, 60.2)
Müşteri 209 (487.4, 325.1)
Müşteri 210 (375.5, 354.7)
Müşteri 211 (287.0, 328.1)
Müşteri 212 (157.7, 184.5)
Müşteri 213 (28.7, 167.5)
Müşteri 214 (160.0, 320.8)
Müşteri 215 (444.2, 76.2)
Müşteri 216 (497.1, 176.7)
Müşteri 217 (247.4, 70.2)
Müşteri 218 (379.8, 315.9)
Müşteri 219 (292.5, 134.7)
Müşteri 220 (242.2, 252.1)
Müşteri 221 (388.9, 45.0)
Müşteri 222 (356.3, 488.8)
Müşteri 223 (136.6, 188.4)
Müşteri 224 (426.2, 147.3)
Müşteri 225 (377.0, 344.8)
Müşteri 226 (389.6, 441.2)
Müşteri 227 (217.4, 235.8)
Müşteri 228 (281.0, 180.1)
Müşteri 229 (99.5, 279.7)
Müşteri 230 (24.6, 428.5)

```
Müşteri 231 (27.6, 370.7)
Müşteri 232 (432.6, 403.6)
Müşteri 233 (366.0, 159.3)
Müşteri 234 (484.1, 168.7)
Müşteri 235 (40.7, 223.6)
Müşteri 236 (144.1, 165.8)
Müşteri 237 (54.8, 210.8)
Müşteri 238 (63.6, 138.9)
Müşteri 239 (436.4, 472.8)
Müşteri 240 (469.5, 16.6)
Müşteri 241 (91.8, 440.2)
Müşteri 242 (414.1, 358.9)
Müşteri 243 (25.7, 455.6)
Müşteri 244 (302.7, 25.8)
Müşteri 245 (29.7, 35.0)
Müşteri 246 (261.0, 290.0)
Müşteri 247 (206.4, 41.4)
Müşteri 248 (434.1, 200.5)
Müşteri 249 (118.8, 26.8)
Müşteri 250 (232.4, 100.7)
Müşteri 251 (403.8, 47.0)
Müşteri 252 (351.2, 318.5)
Müşteri 253 (387.7, 61.5)
Müşteri 254 (96.8, 176.7)
Müşteri 255 (77.3, 103.1)
Müşteri 256 (143.5, 115.0)
```

```
Müşteri 257 (376.5, 246.5)
Müşteri 258 (332.6, 376.6)
Müşteri 259 (497.1, 6.5)
Müşteri 260 (499.4, 389.0)
Müşteri 261 (289.9, 171.5)
Müşteri 262 (79.8, 160.0)
Müşteri 263 (121.7, 214.0)
Müşteri 264 (405.7, 216.9)
Müşteri 265 (462.9, 322.0)
Müşteri 266 (146.0, 341.2)
Müşteri 267 (31.1, 263.6)
Müşteri 268 (285.5, 89.3)
Müşteri 269 (156.4, 233.8)
Müşteri 270 (367.3, 343.7)
Müşteri 271 (495.7, 28.3)
Müşteri 272 (437.3, 410.0)
Müşteri 273 (325.7, 356.8)
Müşteri 274 (246.5, 459.5)
Müşteri 275 (386.2, 339.4)
Müşteri 276 (259.9, 421.0)
Müşteri 277 (129.4, 224.9)
Müşteri 278 (418.3, 237.9)
Müşteri 279 (254.3, 133.4)
Müşteri 280 (376.0, 244.3)
Müşteri 281 (70.5, 152.4)
Müşteri 282 (114.5, 257.7)
```

```
Müşteri 283 (39.4, 409.4)
Müşteri 284 (389.6, 353.0)
Müşteri 285 (288.9, 303.6)
Müşteri 286 (429.4, 158.6)
Müşteri 287 (481.5, 402.9)
Müşteri 288 (139.2, 413.7)
Müşteri 289 (145.7, 104.1)
Müşteri 290 (307.6, 434.3)
Müşteri 291 (444.8, 9.3)
Müşteri 292 (433.1, 62.1)
Müşteri 293 (369.3, 92.1)
Müşteri 294 (304.4, 113.7)
Müşteri 295 (483.8, 117.4)
Müşteri 296 (378.1, 302.9)
Müşteri 297 (154.0, 428.2)
Müşteri 298 (291.3, 83.2)
Müşteri 299 (103.9, 132.9)
Müşteri 300 (36.9, 347.9)
```

ii) Problem için en uygun kromozom (çözüm adayı) yapısı tasarımı

Bu problemde önceden belirlenmiş bir noktayı seçmek yerine baz istasyonu konumlarına sürekli (reel sayı) uzayda kendimiz karar veriyoruz. Bizden istenen çözüm $(x_1, y_1), (x_2, y_2), \dots, (x_{50}, y_{50})$ şeklinde 50 adet koordinat çifti (100'lük reel sayı vektörü) biçimindedir. Dolayısıyla çözüm adaylarımız reel sayı vektörü şeklinde olacak.

iii) Uygunluk (fitness) değerini hesaplayan metot

- Uygunluk fonksiyonu 2. bölümde kullanılan uygunluk fonksiyonu ile temelde aynıdır. Yine uygunluk fonksiyonu hesaplamak için Öklid uzaklık formülü kullanılmıştır.
- CMAES (Covariance Matrix Adaptation Evolution Strategy) kütüphanesi kullanıyoruz.
- **MultivariateFunction** arayüzünden implement edilmiş olan **DistanceFunction.java** sınıfında uygunluk fonksiyonu tanımlanmıştır. **value metodu** double dönüş tipindedir ve input olarak double dizi tipinde bir vektör (**point**) alır.
- Fonksiyon içerisinde problemde verilen parametrelere göre **Coordinate.java** sınıfından bir nesne oluşturularak müşterilerin koordinatları rastgele olarak belirlenmiştir. Amacımız baz istasyonlarının optimum yerlerine ulaşmak olduğu için bunu üretmiyoruz.

- Çözüm sonunda bir reel sayı vektörü elde edeceğiz. Bu vektör optimize edilen baz istasyonu koordinatlarını içeren 100 uzunluklu bir reel sayı vektörüdür. Bu aynı zamanda 50 adet koordinat çifti demektir.
- Uygunluk fonksiyonunda bu vektördeki koordinat çiftleri ile müşterilerin koordinatları arasındaki uzunluklar bulunmuştur. Eğer bu uzaklık problemde daha önceden belirtilen kapsama alanı uzaklığından küçükse müşteri baz istasyonu tarafından kapsanıyor demektir. Bu durumda uygunluk değerini tuttuğumuz **total** değişkeni artırılmalıdır.

iv) Optimizasyon Algoritması Tasarımı - CMAES

- Apache Commons Math kütüphanesi içerisinde Evrimsel Strateji algoritmasının gelişmiş bir versiyonu olan ‘Covariance Matrix Adaptation Evolution Strategy (CMA-ES) yöntemi bulunmaktadır.
- CMAES ile **reel sayı parametreleri** şeklinde temsil edilen optimizasyon problemleri etkin bir şekilde çözülebilmektedir.
- Apache commons math içerisinde **CMAESOptimizer** adı altında CMAES algoritmasının hazır bir gerçekleştirimi bulunmaktadır. Bu sınıfın bir nesnesi oluşturulup **optimize** metodunun çağırılması gerekmektedir.
- Parametrelerinden **maxIterations** algoritmanın kaç iterasyon boyunca çalışacağını belirler. **stopFitness** değerine eğer 0 verirse kullanmaz, başka bir değer verirse çalışma sırasında bu değere rastlanırsa algoritmayı yarıda keser. **MersenneTwister** rastgele sayı üretici olarak kullanılır. Bunların sonunda **es objesi** oluşuyor.
- es objesinin optimize metodunu çalıştırdığımız zaman problemi çözüp bulduğu optimum sonucu **result** değişkenine atıyor.
- **optimize** metot çağırım parametreleri:
 - **MaxEval:** Uygunluk fonksiyonunu kaç kere çalıştıracağını belirtir
 - **CMAESOptimizer.PopulationSize:** Popülasyon büyüklüğü
 - **CMAESOptimizer.Sigma:** Genellikle rastgele oluşturulan sigma değerleri
 - **InitialGuess:** Genellikle rastgele oluşturulan başlangıç çözümü tahminleri
 - **SimpleBounds:** Aramanın alt ve üst limitleri (optimize edilecek x değerleri alt limitin altında ve üst limitin üstünde olamaz)
 - **ObjectiveFunction:** Optimize edilecek olan uygunluk fonksiyonu
 - **GoalType:** minimizasyon ya da maksimizasyon
- **result** objesindeki en iyi çözüm noktalarını bulmak için de onun **getPoint()** metodu çağırıldığında optimize edilmiş reel sayı vektörü çözümü elde edilir.

v) Deneysel Çalışmalar ve İstatistiksel Yorum

Parametre Seti 1 : populationSize = 10, maxIteration = 1000

Parametre Seti 2 : populationSize = 20, maxIteration = 1000

Parametre Seti 3 : populationSize = 30, maxIteration = 1000

Parametre Seti 4 : populationSize = 40, maxIteration = 1000

Parametre Seti 5 : populationSize = 100, maxIteration = 1000

Parametre Seti 6 : populationSize = 10, maxIteration = 5000

	Parametre Seti 1	Parametre Seti 2	Parametre Seti 3	Parametre Seti 4	Parametre Seti 5	Parametre Seti 6
run 1	162.0	164.0	144.0	167.0	150.0	146.0
run 2	148.0	177.0	160.0	166.0	158.0	148.0
run 3	147.0	152.0	155.0	160.0	165.0	143.0
run 4	149.0	145.0	164.0	151.0	142.0	161.0
run 5	147.0	146.0	155.0	149.0	151.0	136.0
run 6	163.0	137.0	143.0	149.0	147.0	191.0
run 7	166.0	153.0	157.0	166.0	158.0	145.0
run 8	163.0	137.0	156.0	135.0	160.0	149.0
run 9	154.0	157.0	151.0	144.0	155.0	143.0
run 10	148.0	161.0	141.0	143.0	145.0	139.0
run 11	146.0	162.0	145.0	157.0	145.0	155.0
run 12	174.0	159.0	160.0	151.0	152.0	161.0
run 13	169.0	159.0	169.0	167.0	163.0	144.0
run 14	152.0	170.0	165.0	160.0	178.0	147.0

run 15	139.0	160.0	147.0	161.0	162.0	160.0
run 16	167.0	142.0	152.0	158.0	143.0	154.0
run 17	145.0	155.0	168.0	169.0	156.0	151.0
run 18	161.0	140.0	138.0	141.0	133.0	142.0
run 19	140.0	154.0	164.0	139.0	153.0	147.0
run 20	133.0	148.0	148.0	139.0	155.0	155.0
run 21	148.0	141.0	157.0	157.0	153.0	149.0
run 22	164.0	158.0	152.0	163.0	167.0	152.0
run 23	168.0	145.0	155.0	154.0	132.0	165.0
run 24	163.0	152.0	150.0	141.0	155.0	164.0
run 25	146.0	166.0	165.0	172.0	163.0	155.0
Ortalama	154.48	153.6	154.44	154.36	153.64	152.08
Standart Sapma	10.752	10.213	8.485	10.669	10.189	10.976
En iyi	174.0	177.0	169.0	172.0	178.0	191.0
En kötü	133.0	137.0	138.0	135.0	132.0	136.0
Medyan	152.0	154.0	155.0	157.0	155.0	149.0

Bu deneysel çalışmada popülasyon büyüklüğünün optimum çözüme etkisi gözlenmek istenmiştir. Çözüm 25 kez çalıştırılıp çıktı değerlerinin ortalaması alınmıştır. Çözümlerin ortalaması üzerinden yorumlamak rastgele değerler arasındaki farklılıklar sebebiyle daha sağlıklı bir yöntemdir. İlk 5 parametre seti için ortalamalara baktığımızda aralarında çok büyük fark olmadığı görülmektedir. Popülasyon büyüklüğü 10 olan deney ortalamasının popülasyon büyüklüğü 100 olan deneyin ortalamasından daha büyük olduğu görülmektedir. Burdan çıkarım yaparak maksimizasyon problemleri için büyük boyutlu popülasyonlar yerine küçük boyutlu popülasyonlar tercih etmeliyiz diyebiliriz. İterasyon sayısı ile alakalı değişiklikleri gözlemek için ise 6. parametre seti oluşturulmuştur ve iterasyon sayısı artırılmıştır. Her ne kadar değerlerin daha yukarı çıktığı görülse de ortalamaya baktığımızda aralarındaki en düşük ortalamanın bu deneye ait olduğu görülüyor ve yüksek iterasyon sayısının optimum çözüme katkı sağlamadığı anlaşıyor.