CS547: Advanced Topics in Software Engineering

coursework #1 (part 1)

BY

SUNNY RAI

REGISTRATION NUMBER

202464889

Genetic Algorithm

Genetic algorithm is based on real biological evolution. It solve complex problems also it focus on optimization .

Implementation of Genetic Algorithm:

1. Step) Selected initial population
2. Step) Calculate fitness function.
3. Step) Selection of parent on the basis of their fitness function.
4. Step) Implemented crossover for selecting offspring from parent.
5. Step) Implemented mutation and did little change in offspring.
6. Step) Reach to the final solution

Applied Tournament Selection:

For choosing quality parent class, I applied tournament selection on basis of best fitness score.

REPRESENTATION:

Steps involved:

1. Initialize an empty population list:

   A list population is created to store the initial individuals.

2. Iterate to create individuals:

   For each individual in the population:

   - A list of integers individual is created, representing the indices of the test cases.
   - The random.shuffle() function is used to randomly shuffle the elements of the individual list, creating a random permutation of test case indices.

        - The shuffled individual is appended to the population list.

3. Return the population:

    The function returns the generated population, which is a list of lists, where each inner list represents a potential test order.

FOR CROSSOVER IMPLIMENTAION:

1. **Probability Check:**
    - A random number between 0 and 1 is generated.
    - If this random number is less than the crossover rate (default is 0.8), the crossover operation proceeds. Otherwise, the parents are directly copied to the offspring.
2. **Selecting Crossover Point:**
    - A random crossover point is chosen within the length of the parent individuals. This point determines where the genetic material will be exchanged.
3. **Creating Offspring:**
    - The first part of the first child is taken from the first parent up to the crossover point.
    - The second part of the first child is taken from the second parent starting from the crossover point.
    - The second child is created in a similar way, but the parent roles are reversed.

FOR MUTATION IMPLIMENTAION:

1. Probability Check:
   - A random number between 0 and 1 is generated.
   - If this random number is less than the mutation rate (default is 0.01), the mutation operation proceeds. Otherwise, the individual remains unchanged.
2. Random Selection of Indices:
   - Two random indices, i and j, are selected from the range of the individual's length. These indices represent the positions of two elements in the individual's chromosome.
3. Swapping Elements:
   - The elements at positions i and j in the individual are swapped. This creates a new individual with a slightly different genetic.

HILL CLIMBING ALGORITHM

It is local search algorithm when it will get best move it will work. If best move stop it won't work.

- It always get stuck local optima.

Implementation of hill climbing algorithm.

1. Calculate it fitness function of given fault matrix.
2. Generated its neighbour

```
i, j = random.sample(range(len(solution)), 2)

neighbor[i], neighbor[j] = neighbor[j], neighbor[i]

    return neighbor
```

3. Applied HC algorithm.

Comparison of Hill climbing and Genetic algorithm on the basis of their fitness function and solution in small fault matrix.

IN SMALL FAULT MATRIX after multiple iteration

Hill climbing algorithm	Genetic algorithm
Best Fitness function - 9	Best Fitness function - 10
BEST SOLUTION ITERATION ON .	**BEST SOLUTION ITERATION ON .**
Best Solution: [54 64 55 76 27 88 62 56 43 17 5 51 8 48 11 16 57 13 52 77 83 42 9 3 34 47 89 20 86 70 33 35 25 84 32 82 69 6 7 22 46 78 58 44 1 53 0 14 28 81 40 50 26 72 36 74 38 15 49 61 21 37 23 45 12 80 65 79 24 19 39 31 30 29 75 59 2 71 87 73 66 60 68 41 18 63 4 10 67 85]	Best order: [34, 32, 68, 48, 44, 35, 77, 17, 81, 38, 58, 63, 50, 16, 8, 36, 80, 66, 65, 12, 7, 75, 85, 3, 41, 28, 31, 59, 72, 62, 76, 1, 49, 42, 33, 83, 30, 64, 10, 13, 37, 46, 47, 14, 51, 24, 15, 40, 54, 89, 79, 61, 2, 82, 53, 25, 78, 39, 19, 11, 0, 4, 70, 67, 56, 84, 9, 69, 52, 21, 5, 86, 73, 88, 55, 60, 18, 45, 29, 74, 43, 71, 26, 57, 6, 22, 87, 20, 27, 23]

As compare to hill climbing algorithm on the iteration of 1000 ,100…, it’s taking longer time as compare to genetic algorithm.

IN BIG FAULT MATRIX

Hill climbing algorithm	Genetic algorithm
Best Fitness function - 35	Best Fitness function - 38
BEST SOLUTION ITERATION ON .	**BEST SOLUTION ITERATION ON .**
Best Solution: [350 149 238 95 146 196 48 189 119 307 115 280 240 229 77 148 69 283 53 183 311 10 215 330 103 158 56 198 26 377 108 230 165 306 264 343 61 43 190 121 336 297 79 44 101 21 262 129 261 231 252 265 359 75 91 278 104 232 344 136 185 27 117 341 370 293 112 285 321 50 184 213 118 226 97 277 88 259 54 82 334 28 345 305 191 110 85 51 304 127 216 246 219 89 16 322 193 288 17 206 212 374 317 33 100 176 47 67 22 354 326 287 324 258 310 255 55 6 282 373 168 308 299 203 286 272 49 234 340 269 114 133 339 202 270 157 291 78 195 279 245 147 5 372 154 25 19 273 274 313 201 135 302 251 295 84 356 281 160 166 182 11 331 80 145 106 289 338 124 207 109 161 73 319 12 64 197 126 159 169 209 211 1 60 309 349 355 177 233 346 180 276 35 32 235 94 256 328 284 164 186 153 150 375 125 335 224 9 366 187 131 62 162 134 360 23 253 122 221 188 0 337 111 263 194 90 18 327 361 243 244 323 152 267 222 268 367 68 200 74 351 294 31 116 81 301 199 257 376 172 218 130 296 142 175 132 365 208 76 192 40 144 220 163 260 378 173 239 36 45 92 312 254 318 123 42 105 39 86 170 178 174 13 248 352 98 70 181 141 38 314 236 59 66 37 167 3 113 364 329 14 250 120 333 227 347 52 58 140 57 325 139 29 217 4 357 107 128 63 138 87 171 362 320 348 205 292 41 353 303 290 315 143 342 99 275 96 204 363 72 155 298 247 249 228 214 7 102 15 83 371 93 266 20 156 223 71 225 368 30 65 242 241 8 358 179 369 137 151 237 24 46 271 210 300 34 332 316 2]	**Best order: [180, 64, 107, 279, 301, 221, 47, 251, 115, 3, 320, 123, 256, 34, 204, 170, 231, 60, 335, 128, 295, 125, 285, 71, 253, 358, 179, 62, 244, 176, 86, 58, 37, 302, 19, 70, 342, 63, 316, 13, 236, 174, 177, 214, 220, 147, 343, 263, 258, 89, 61, 245, 307, 161, 213, 347, 366, 149, 293, 206, 30, 357, 232, 311, 318, 325, 327, 120, 308, 25, 212, 223, 276, 94, 154, 148, 158, 41, 122, 51, 119, 273, 126, 346, 337, 137, 289, 254, 124, 181, 184, 292, 229, 297, 163, 2, 356, 7, 145, 321, 219, 33, 373, 38, 266, 72, 52, 341, 339, 26, 74, 156, 146, 250, 298, 35, 84, 243, 27, 365, 56, 286, 328, 306, 160, 91, 73, 190, 282, 372, 121, 355, 185, 106, 143, 76, 332, 165, 20, 260, 45, 164, 104, 291, 239, 12, 269, 224, 133, 88, 310, 248, 334, 326, 215, 344, 5, 16, 175, 205, 139, 272, 48, 57, 40, 17, 129, 95, 109, 265, 363, 90, 98, 118, 24, 290, 367, 228, 281, 182, 202, 275, 280, 261, 4, 349, 278, 92, 68, 54, 338, 1, 284, 21, 80, 36, 201, 169, 353, 103, 114, 43, 210, 14, 59, 274, 136, 294, 140, 108, 376, 304, 87, 369, 10, 194, 230, 255, 330, 351, 132, 138, 361, 130, 277, 173, 233, 171, 317, 242, 199, 162, 296, 134, 186, 141, 195, 305, 85, 0, 340, 368, 209, 150, 225, 6, 226, 79, 207, 227, 15, 183, 300, 259, 28, 319, 336, 249, 287, 348, 371, 127, 352, 208, 135, 157, 188, 131, 235, 105, 257, 252, 312, 216, 65, 46, 313, 264, 364, 323, 78, 55, 360, 200, 314, 309, 237, 238, 299, 117, 303, 75, 116, 50, 192, 11, 39, 49, 111, 218, 324, 42, 345, 240, 329, 322, 44, 370, 172, 187, 23, 83, 350, 375, 77, 53, 9, 32, 101, 191, 270, 315, 159, 67, 113, 374, 112, 152, 377, 100, 222, 196, 267, 18, 283, 142, 211, 167, 31, 247, 333, 217, 198, 151, 153, 268, 203, 193, 178, 189, 97, 144, 93, 378, 241, 288, 81, 331, 66, 99, 29, 354, 8, 102, 262, 166, 155, 246, 96, 22, 359, 82, 234, 362, 110, 69, 168, 197, 271]**