Alexander Dowell

CS461 Program 2

Sample Output:

gen: 1 best: 8.9 average: 3.8008 percent gain in averages: 100.0 %

gen: 2 best: 9.3 average: 6.7133 percent gain in averages: 43.4 %

gen: 3 best: 9.7 average: 7.5908 percent gain in averages: 11.6 %

gen: 4 best: 9.8 average: 8.0977 percent gain in averages: 6.3 %

gen: 5 best: 10.5 average: 8.4824 percent gain in averages: 4.5 %

gen: 6 best: 10.5 average: 8.841 percent gain in averages: 4.1 %

gen: 7 best: 10.5 average: 9.0875 percent gain in averages: 2.7 %

gen: 8 best: 10.8 average: 9.2604 percent gain in averages: 1.9 %

gen: 9 best: 10.8 average: 9.531 percent gain in averages: 2.8 %

gen: 10 best: 10.8 average: 9.7092 percent gain in averages: 1.8 %

gen: 11 best: 10.8 average: 9.9331 percent gain in averages: 2.3 %

gen: 12 best: 10.8 average: 10.0918 percent gain in averages: 1.6 %

There is less than 1% difference in the averages between generation 12 and generation 13

gen: 13 best: 11.0 average: 10.1764 percent gain in averages: 0.8 %

gen: 14 best: 11.0 average: 10.2352 percent gain in averages: 0.6 %

gen: 15 best: 11.0 average: 10.308 percent gain in averages: 0.7 %

gen: 16 best: 11.5 average: 10.329 percent gain in averages: 0.2 %

gen: 17 best: 11.0 average: 10.4608 percent gain in averages: 1.3 %

gen: 18 best: 11.5 average: 10.4936 percent gain in averages: 0.3 %

gen: 19 best: 11.5 average: 10.5306 percent gain in averages: 0.4 %

gen: 20 best: 11.5 average: 10.5242 percent gain in averages: -0.1 %

gen: 21 best: 11.5 average: 10.573 percent gain in averages: 0.5 %

gen: 22 best: 11.5 average: 10.6494 percent gain in averages: 0.7 %

gen: 23 best: 11.5 average: 10.6669 percent gain in averages: 0.2 %

gen: 24 best: 11.5 average: 10.6492 percent gain in averages: -0.2 %

gen: 25 best: 11.5 average: 10.7348 percent gain in averages: 0.8 %

gen: 26 best: 11.5 average: 10.8184 percent gain in averages: 0.8 %

gen: 27 best: 11.5 average: 10.8412 percent gain in averages: 0.2 %

gen: 28 best: 11.5 average: 10.9038 percent gain in averages: 0.6 %

gen: 29 best: 11.5 average: 11.0166 percent gain in averages: 1.0 %

gen: 30 best: 11.5 average: 11.0724 percent gain in averages: 0.5 %

gen: 31 best: 11.5 average: 11.175 percent gain in averages: 0.9 %

gen: 32 best: 11.5 average: 11.2402 percent gain in averages: 0.6 %

gen: 33 best: 11.5 average: 11.2684 percent gain in averages: 0.3 %

gen: 34 best: 11.5 average: 11.2782 percent gain in averages: 0.1 %

gen: 35 best: 11.5 average: 11.2624 percent gain in averages: -0.1 %

gen: 36 best: 11.5 average: 11.3421 percent gain in averages: 0.7 %

gen: 37 best: 11.5 average: 11.4026 percent gain in averages: 0.5 %

gen: 38 best: 11.5 average: 11.3943 percent gain in averages: -0.1 %

gen: 39 best: 11.5 average: 11.4018 percent gain in averages: 0.1 %

gen: 40 best: 11.5 average: 11.4332 percent gain in averages: 0.3 %

gen: 41 best: 11.5 average: 11.433 percent gain in averages: -0.0 %

gen: 42 best: 11.5 average: 11.4033 percent gain in averages: -0.3 %

gen: 43 best: 11.5 average: 11.382 percent gain in averages: -0.2 %

gen: 44 best: 11.5 average: 11.3874 percent gain in averages: 0.0 %

gen: 45 best: 11.5 average: 11.3833 percent gain in averages: -0.0 %

gen: 46 best: 11.5 average: 11.3829 percent gain in averages: -0.0 %

gen: 47 best: 11.5 average: 11.376 percent gain in averages: -0.1 %

gen: 48 best: 11.5 average: 11.354 percent gain in averages: -0.2 %

gen: 49 best: 11.5 average: 11.3852 percent gain in averages: 0.3 %

gen: 50 best: 11.5 average: 11.4122 percent gain in averages: 0.2 %

gen: 51 best: 11.5 average: 11.3702 percent gain in averages: -0.4 %

gen: 52 best: 11.5 average: 11.3526 percent gain in averages: -0.2 %

gen: 53 best: 11.5 average: 11.3846 percent gain in averages: 0.3 %

gen: 54 best: 11.5 average: 11.3826 percent gain in averages: -0.0 %

gen: 55 best: 11.5 average: 11.3832 percent gain in averages: 0.0 %

gen: 56 best: 11.5 average: 11.3948 percent gain in averages: 0.1 %

gen: 57 best: 11.5 average: 11.3754 percent gain in averages: -0.2 %

gen: 58 best: 11.5 average: 11.3482 percent gain in averages: -0.2 %

gen: 59 best: 11.5 average: 11.367 percent gain in averages: 0.2 %

gen: 60 best: 11.5 average: 11.3954 percent gain in averages: 0.2 %

gen: 61 best: 11.5 average: 11.4032 percent gain in averages: 0.1 %

gen: 62 best: 11.5 average: 11.3895 percent gain in averages: -0.1 %

gen: 63 best: 11.5 average: 11.376 percent gain in averages: -0.1 %

gen: 64 best: 11.5 average: 11.399 percent gain in averages: 0.2 %

gen: 65 best: 11.5 average: 11.43 percent gain in averages: 0.3 %

gen: 66 best: 11.5 average: 11.4056 percent gain in averages: -0.2 %

gen: 67 best: 11.5 average: 11.406 percent gain in averages: 0.0 %

gen: 68 best: 11.5 average: 11.4124 percent gain in averages: 0.1 %

gen: 69 best: 11.5 average: 11.4462 percent gain in averages: 0.3 %

gen: 70 best: 11.5 average: 11.407 percent gain in averages: -0.3 %

gen: 71 best: 11.5 average: 11.4212 percent gain in averages: 0.1 %

gen: 72 best: 11.5 average: 11.4044 percent gain in averages: -0.1 %

gen: 73 best: 11.5 average: 11.4276 percent gain in averages: 0.2 %

gen: 74 best: 11.5 average: 11.4078 percent gain in averages: -0.2 %

gen: 75 best: 11.5 average: 11.3608 percent gain in averages: -0.4 %

gen: 76 best: 11.5 average: 11.3906 percent gain in averages: 0.3 %

gen: 77 best: 11.5 average: 11.3974 percent gain in averages: 0.1 %

gen: 78 best: 11.5 average: 11.4068 percent gain in averages: 0.1 %

gen: 79 best: 11.5 average: 11.4234 percent gain in averages: 0.1 %

gen: 80 best: 11.5 average: 11.3941 percent gain in averages: -0.3 %

gen: 81 best: 11.5 average: 11.3968 percent gain in averages: 0.0 %

Alexander Dowell

gen: 82 best: 11.5 average: 11.4277 percent gain in averages: 0.3 %

gen: 83 best: 11.5 average: 11.3722 percent gain in averages: -0.5 %

gen: 84 best: 11.5 average: 11.3828 percent gain in averages: 0.1 %

gen: 85 best: 11.5 average: 11.3638 percent gain in averages: -0.2 %

gen: 86 best: 11.5 average: 11.3754 percent gain in averages: 0.1 %

gen: 87 best: 11.5 average: 11.3684 percent gain in averages: -0.1 %

gen: 88 best: 11.5 average: 11.359 percent gain in averages: -0.1 %

gen: 89 best: 11.5 average: 11.3443 percent gain in averages: -0.1 %

gen: 90 best: 11.5 average: 11.3337 percent gain in averages: -0.1 %

gen: 91 best: 11.5 average: 11.3866 percent gain in averages: 0.5 %

gen: 92 best: 11.5 average: 11.4032 percent gain in averages: 0.1 %

gen: 93 best: 11.5 average: 11.4148 percent gain in averages: 0.1 %

gen: 94 best: 11.5 average: 11.4318 percent gain in averages: 0.1 %

gen: 95 best: 11.5 average: 11.3948 percent gain in averages: -0.3 %

gen: 96 best: 11.5 average: 11.385 percent gain in averages: -0.1 %

gen: 97 best: 11.5 average: 11.3774 percent gain in averages: -0.1 %

gen: 98 best: 11.5 average: 11.3537 percent gain in averages: -0.2 %

gen: 99 best: 11.5 average: 11.3688 percent gain in averages: 0.1 %

gen: 100 best: 11.5 average: 11.4193 percent gain in averages: 0.4 %

Alexander Dowell

schedule fitness score: 11.5

course	time	building/room	instructor
CS101A	11 AM	Royall 201	Hare
CS101B	11 AM	FH 310	Zein el Din
CS191A	10 AM	Haag 301	Hare
CS191B	12 PM	Haag 301	Zein el Din
CS201	01 PM	Bloch 119	Zein el Din
CS291	10 AM	Haag 201	Zein el Din
CS303	01 PM	FH 310	Hare
CS304	02 PM	FH 216	Hare
CS394	12 PM	FH 216	Xu
CS449	02 PM	Bloch 119	Xu

What were the challenges in writing the program? Or did it seem to go smoothly from the beginning?

There were a few sticking points while writing program 2. The first issue that I had encountered was interpreting the rules that needed to be implemented for determining the fitness for each schedule. After writing each rule in the way I thought was the most logical and getting clarification from the professor that interpretation was up to the programmer this problem was resolved. I also originally had the rules inter-dependent so I had to decouple them so I could unit test each of them independently. I had found some logical errors in some of my rules so I'm glad I did this. After getting some clarification on how the mutation function should be functioning I needed to alter my mutation function. The largest

challenge I encountered was after I had what I thought was going to be a functioning genetic algorithm but the average fitness score between generations was not increasing. It took some time but I finally located the bug in my tournament selection function. I just needed to add " p = " to

" parentI = np.random.choice(list_of_chromosome_index, 1, p = probabilities) "

What do you think of the schedule your program produced? Does it have anything that still looks odd or out of place?

My schedule optimizes for the given rules but some of the rules are odd. The fact that it assigns an multiple instructors to 4 classes is odd.

How would you improve the program, or change the fitness function?

I would start by adding a penalty for teaching 3 or 4 classes. I think this would prevent a teacher from teaching 3 or 4 classes and allow for all or most of the instructors an assignment. I also might change some of the weights for some of the rules.

Anything else you feel like discussing, asking about, bragging about, etc.?

I don't think that my algorithm is the most efficient and would always appreciate any feedback to make it better/faster. I did dial back the mutation rate to .007 but too much lower than that increased the time to convergence greatly. Usually between generation 10 and 15 the 1% gain in the difference between consecutive generation averages occurs. After multiple trials fitness score would typically converge between generation 20 and 25. The fitness score would converge to a range between 10.6 and 11.5. I would also like to understand how I can avoid local maximums and consistently converge to the global maximum or at least decrease the range.