THE UNIVERSITY of ADELAIDE

Evolutionary Computation
Assignment 1 report

By:

a1788848	Hao Zhang
a1781122	Hanyue Zhang
a1763050	Yiyuan Tang
a1804817	Xiaoman Li
a1779176	Xiaocen Guo
a1767860	Zibo Lin

Exercise 1:

We allocated the six exercises to six teammates and we chose Hao Zhang as our team leader to manage the task completing progress.

Hao zhang(Exercise 2):

- Followed the instructions in exercise 2 we wrote a class called TSPProblem in the file :TspProblem.py
- Structure design and implete
- Bugs fixing
- Integrating codes together

Zibo Lin (Exercise 3):

- Implement a possible solution to the TSP as a permutation of given cities under the class “Individual”
- Implement a class “Population” for representing a population which is a set of individuals.
- Implement functions to evaluate the quality of a solution

Xiaoman Li (Exercise 4):

- Do the research of different approaches to mutation and crossover.
- Implement functions of mutation operators (insert, swap, inversion, scramble) for permutations.
- Implement functions of crossover operators include Order Crossover, PMX Crossover, Cycle Crossover, and Edge Recombination for permutations.
- Edit the parameter and return of the functions to satisfy the requirements of the teammate who did the integration.

Yiyuan Tang (Exercise 5):

- Implement functions of selection methods including fitness-proportional, tournament selection, and elitism.
- Implement functions to calculate the fitness of the individual, which is part of the Class Population.

Hanyue Zhang(Exercise 6):

- Design three evolutionary algorithms.
- Prepare and report the results in exercise 6 (Experiment 1 and 2) and exercise 7.

Xiaocen Guo(Exercise 7):

Do research about GuoTao algorithm in the report
Cooperate with Hao Zhang to implement the algorithm
Learn how to draw the route graph and trend chart in python
Prepare for assignment 1 report

Exercise 6 (Design three evolutionary algorithms and justify the design choices)

From the perspective of time complexity, we get rid of insert and scramble mutation operators. Then we test all the combinations of mutation and crossover operators with the same population size and generations three times. The average result shows that inversion with edge recombination ranked first, inversion with order crossover ranked second, inversion with cycle crossover ranked third. These three algorithms perform best, the best algorithm nearly reaches the optimum result in experiment 1. We choose elitism as the selection method because it performs best.

	Algorithm 1	Algorithm 2	Algorithm 3
Mutation method	Inversion	inversion	Inversion
Mutation rate	0.5	0.6	0.7
Crossover method	Order crossover	Cycle crossover	edge recombination

Crossover rate	0.5	0.7	0.6
Selection method	Elitism	Elitism	Elitism

Exercise 6 (experiment 1)

			Algorithm 1(inversion, order crossover)			Algorithm 2(inversion, cycle crossover)			Algorithm 3(inversion, edge recombination)		
files	Population size	Optimum result	5000 generation	10000	20000	5000	10000	20000	5000	10000	20000
eil51	10	429.982	455.674	455.1750	443.2426	448.9347	449.8734	445.2531	451.3946	450.1342	450.2942
	20		461.320	456.0051	453.2167	470.3583	470.6584	469.4213	461.4598	461.0335	460.1365
	50		441.7001	441.9751	443.4701	468.0832	471.9083	468.2319	452.3259	450.2475	450.7421
	100		455.7176	457.7134	449.4963	471.2364	468.3291	469.2312	452.7684	454.2301	449.2497
eil76	10	545.387	591.8350	595.5649	582.0766	599.8177	603.6354	600.9862	591.6302	599.5349	593.7202
	20		593.3220	598.5491	605.1218	605.6842	607.3567	601.2410	574.2370	588.9832	600.3587
	50		593.2882	590.6419	589.3679	607.6929	605.2342	600.3198	576.6843	574.9553	573.9753
	100		592.8128	590.2388	589.0371	607.0211	601.4392	600.2851	582.8589	580.3132	579.0932
eil101	10	642.315	761.5288	733.9799	722.7949	763.1825	767.2343	766.9887	737.4648	730.8723	694.5713

	20		723.2923	722.7128	722.9789	759.8865	762.0124	760.9786	703.6841	700.3129	685.8391
	50		707.5955	710.7127	709.9789	755.0217	759.9886	760.2304	709.8358	703.8719	684.312
	100		706.2211	709.2876	707.0981	765.0134	766.3460	768.3142	695.2191	691.2123	686.7131
st70	10	678.597	729.2717	710.0803	706.123	724.7722	727.1832	722.9153	700.6313	702.2543	700.2126
	20		700.5051	702.6326	704.0150	720.2291	722.9871	721.0863	699.2542	700.0868	706.0585
	50		786.0234	714.0274	705.8712	743.9916	729.8549	722.9227	752.0078	698.257	689.0624
	100		714.9009	709.9102	705.7153	719.3315	722.6122	720.9689	687.0928	691.4753	688.9212
kroa100	10	21285.443	26915.9094	24119.8808	23802.8863	27544.5123	27431.7215	24894.2638	24573.7413	24869.3396	22120.6103
	20		23673.6786	23647.6901	23623.9410	25210.5192	26100.4182	25383.7290	23945.0354	23533.6012	22993.8171
	50		23841.4423	23932.4975	23821.9985	25112.9152	25398.3097	25120.3299	23741.4423	23586.2195	22321.8603
	100		23788.2527	23831.9651	23652.7492	24872.0912	24856.2395	24667.3892	22101.8862	22388.2854	22184.0642
kroc100	10	20750.762	27614.9388	26093.1125	24706.8834	29646.468	29754.347	27881.091	24706.9997	23648.9268	22362.8433
	20		24779.5701	24116.7072	24065.112	27982.1017	27564.0769	26903.7392	23723.8487	22451.3989	22161.9313
	50		23343.520	23397.762	23116.7072	25570.062	25981.986	25237.821	23561.901	23875.213	22690.981

			9	5		3	2	9	7	6	3
lin105	10	1438 2.99	1848 3.964 3	1655 2.281 7	1573 6.165 2	2130 2.081 7	2122 9.701 9	2098 7.632 9	1640 4.133 9	1625 6.452 3	1564 6.152 5
	20		1674 0.862 3	1635 2.327	1669 8.712	1889 2.072 5	1902 3.786 5	1865 4.986 0	1643 3.760 8	1596 6.239 7	1573 2.140 3
pcb44 2	10	5078 3.547 5	2070 76.23 42	1799 13.83 2	11040 7.669 0	2435 82.10 94	2438 76.70 57	2395 61.56 36	1819 28.56 62	1328 84.97 34	9257 6.750 21
pr239 2	10	3780 62.82 6	8014 876.2 355	8934 352.9 842	8023 741.2 468	1009 8234. 2482	1042 5256. 3569	9910 463.3 985	7341 582.1 245	7598 342.3 414	7324 751.4 535
usa13 509	10	-	-	-	-	-	-	-	-	-	-

Exercise 6 (experiment 2)
Best algorithm:algorithm 3
Mutation method: Inversion
Mutation rate:0.7
Crossover method: Edge recombination
Crossover rate:0.6
Selection method: Elitism

50 population size 20000 generation

files	1st result	2nd result	3rd result	4th result	Average cost	Standard deviation
eil51	450.7421	453.1220	449.8761	458.2390	452.9948	3.2526
eil76	573.9753	586.2384	579.3183	585.6321	581.2910	5.0182
eil101	684.312	698.0735	695.3288	685.3981	690.7781	6.0142

st70	689.0624	710.2681	698.6179	688.2096	696.5395	8.9175
kroa100	23813.3462	22661.3597	22321.8603	22893.3125	22922.4696	553.0418
kroc100	22942.6162	22690.9813	23252.2419	22809.5407	22923.8450	209.4563
lin105	15609.1032	16043.9873	15462.7245	15689.5407	15701.3389	213.8874
pcb442	92134.8730	95598.6690	93126.1245	93985.3900	93711.2641	1271.4609
pr2392	7324751.4535	7529834.0988	7387452.3582	7609284.2418	7462830.538	112567.6109
usa13509	-	-	-	-	-	-

Exercise 7 Inver-over

Population size 50,generation size 20000

	eil51	eil76	eil101	st70	kroa100	kroc100	lin105	pcb442	pr2392(1 hour limit)	usa13509
1	446.8716	667.9320	1076.6803	701.24350	43071.4110	42871.9391	31920.7474	480625.3978	14600043.1357	
2	448.1403	670.8675	1079.4812	875.0912	44210.2471	43494.8901	31303.6728	481523.6374	14607469.3	
3	447.1498	687.7402	1085.6390	854.5033	42645.0825	42361.0414	30939.3151	475571.0353	14559996.1	
4	435.1244	677.2985	1082.6043	863.3910	41290.2570	42981.333	31207.8322	471721.8069	14654229.3	
5	436.6430	689.1697	1075.2534	817.1924	42183.2592	43644.2326	31312.5914	472722.9834	14585883.4	
6	445.4355	691.1703	1096.1760	850.5364	41975.2089	42887.3998	32505.3384	473453.4325	14607469.266	
7	447.6912	642.1949	1060.3765	844.3422	42389.8408	42675.3845	31427.4958	480291.5887	14559996.14	

									92	
8	445.7582	703.9780	1058.3488	798.9063	42123.8530	43103.9874	30989.4788	471995.3492	14578507.8054	
9	452.0805	665.2239	1060.7568	863.2379	43001.4832	42357.0564	31406.3875	473862.4925	14688736.3424	
10	442.2041	707.6906	1076.3461	871.7236	42586.0842	42769.4573	31748.3259	479542.5685	14548834.5923	
11	448.2575	688.6328	1080.3477	832.8943	42886.8391	42324.2263	30639.5939			
12	444.8708	711.6357	1078.2389	842.5711	42761.7545	43550.3348	31571.7785			
13	445.9635	686.3572	1075.9826	800.1202	42355.3487	42579.3667	30758.0864			
14	447.2341	690.2341	1083.4588	821.4594	42618.5381	43520.6751	31592.4691			
15	448.9043	687.3310	1088.6081	886.2741	42819.4278	42311.6781	31348.2574			
16	448.2794	701.3248	1079.2456	846.5644	41706.4521	42578.3459	31698.4064			
17	446.9527	702.3458	1084.8943	833.9903	41261.4788	43499.2681	31712.6709			
18	445.3577	689.6194	1073.2954	810.2817	42854.4676	42497.0251	30825.6508			
19	445.1608	707.1236	1077.2418	883.1952	41588.3567	43417.3297	30910.6254			
20	449.0425	688.7017	1089.4345	799.0887	43011.0187	42312.4561	31943.5253			
21	450.2539	682.5931	1065.3783	845.3628	42299.5673	42886.3401	31873.2459			
22	447.2445	691.2763	1078.2349	867.3421	42801.2958	42655.2976	31873.2547			
23	448.5442	711.6377	1083.2636	812.6424	43350.4496	42318.6723	31567.0732			
24	446.2475	703.6732	1087.2367	843.1573	43152.6044	43012.4595	31662.3361			
25	447.3529	700.7042	1069.3426	850.8844	42957.5494	43224.9835	30883.4561			

26	450.9201	692.4759	1060.4577	842.4961	42866.3578	42719.8410	30882.4982			
27	445.7891	687.4526	1080.3457	827.4329	42356.9876	43397.5378	31093.2884			
28	448.2307	688.0921	1076.3453	844.9887	42147.2357	42831.7369	32015.4321			
29	447.7908	702.0223	1072.4250	810.2472	42811.8762	43512.4589	31849.3327			
30	453.6106	679.7539	1059.2385	848.0987	41226.2389	42597.3488	31639.3591			
Average cost	446.7702	689.8751	1076.4892	836.3086	42510.3524	42896.4701	31436.7175	476131.0292	14599116.54	
Standard deviation	3.7161	15.0707	9.7391	35.3046	662.9616	443.4734	452.2604	3930.1918	43967.7211	

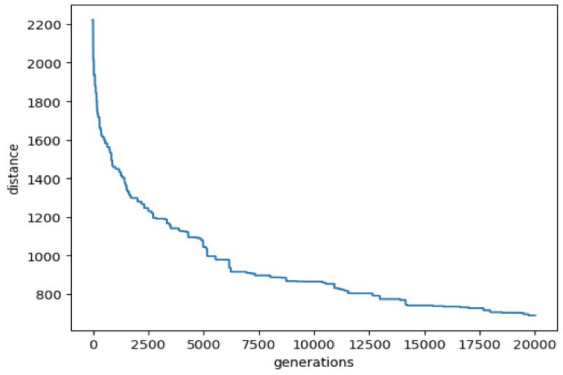
Reflection:

During the group assignment process, our final goal was to design and implement the algorithm completely to get the optimal solution. However, in the testing period, we noticed that the running time for a 20000 generations' result will take 60000 minutes. We considered it as a normal situation and we decided to minimise the generation to get results as soon as possible. Two days before the deadline, we found that it is reasonable to improve the algorithm performance to reduce the running time, but it is a bit late to calculate all the required results. The remaining time is only enough for us to run once for each file. Learned from this experience, the assignment is supposed to start early to leave more time for the testing period.

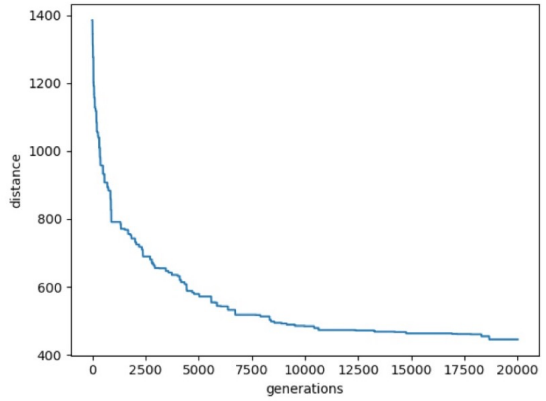
PLOTS

Exercise 7 plots:

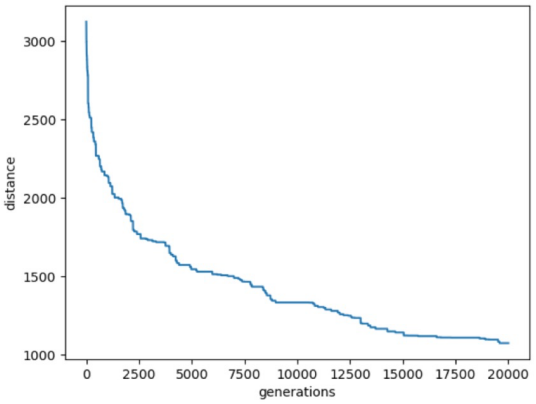
guotao-eil76



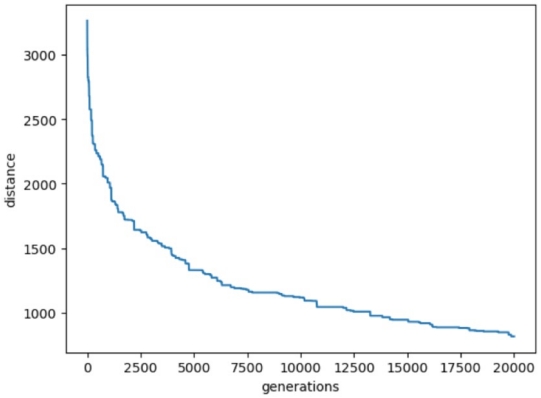
guotao-eil50



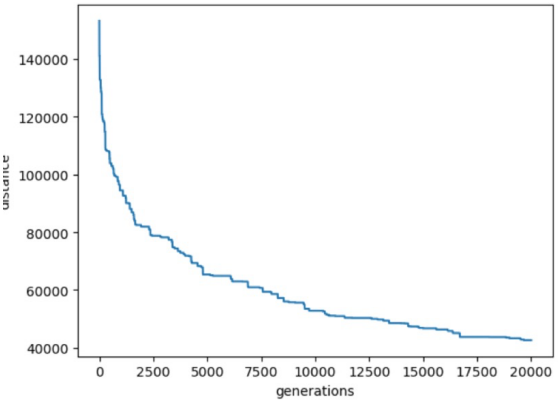
guotao-eil101



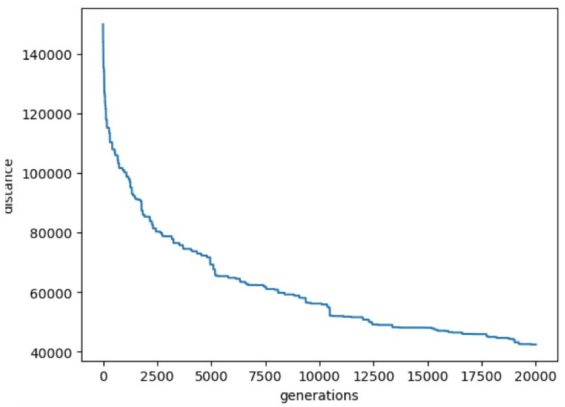
guotao-st70



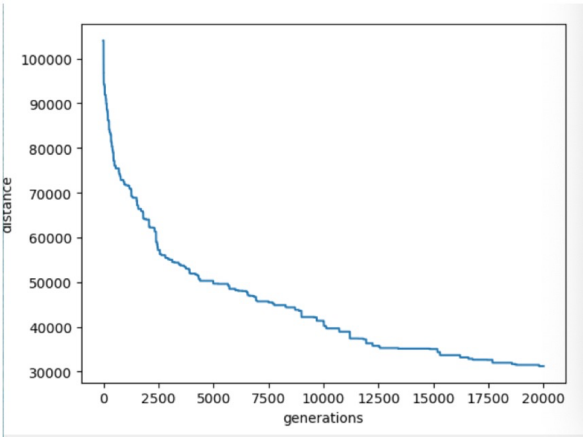
guotao-kroa100



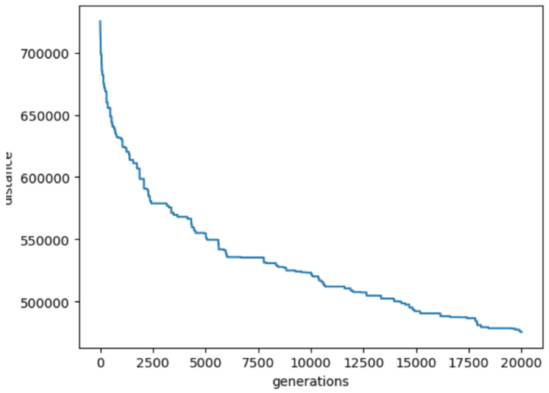
guotao-kroc100



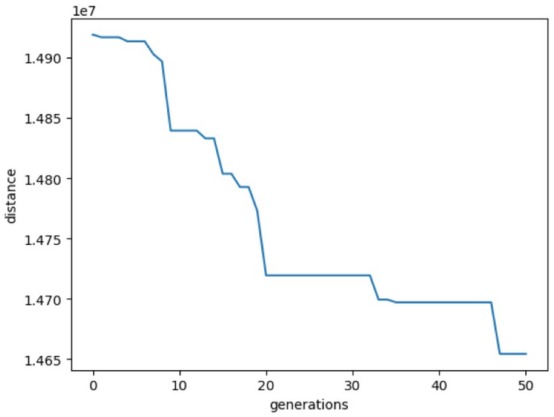
guotao-lin105



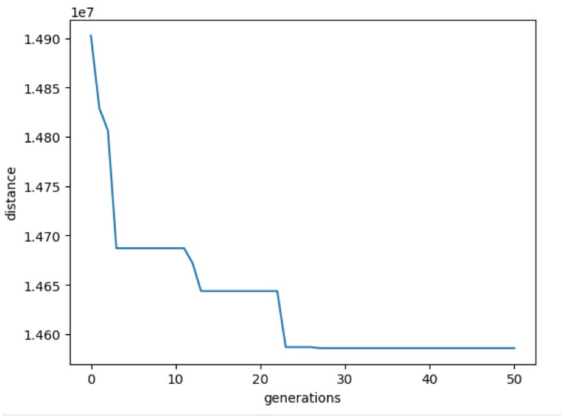
guotao-pcb442



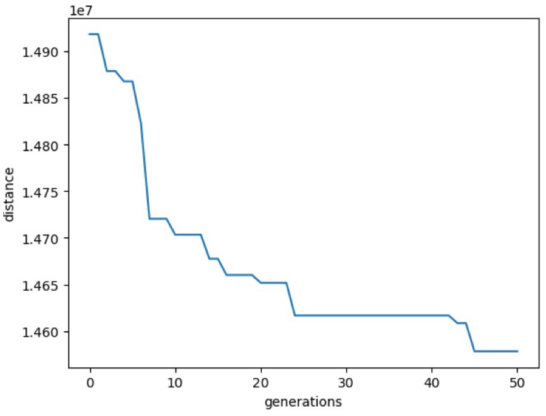
guotao-pr2392



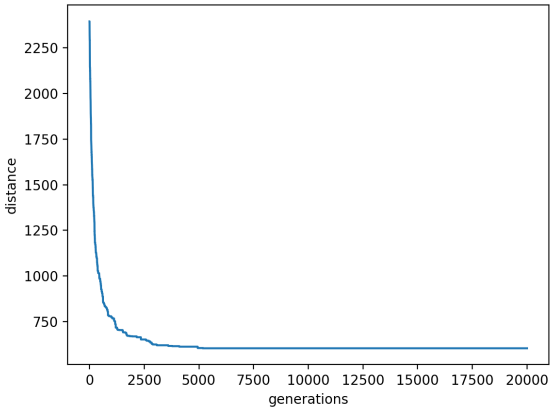
guotao-pr2392



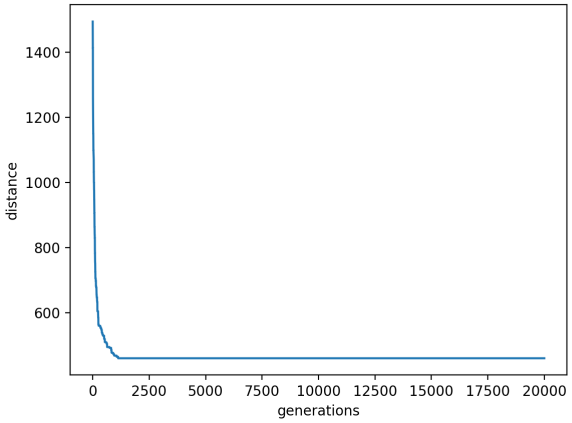
guotao-pr2392



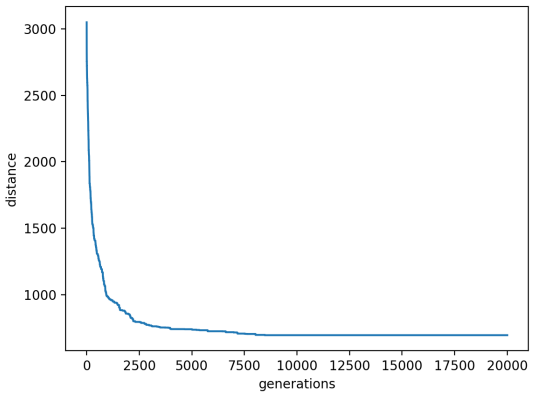
Exercise 6 Best algorithm plots



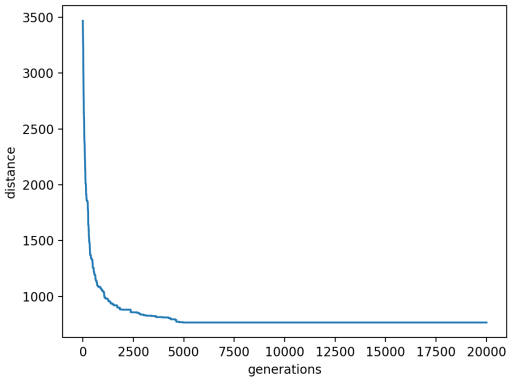
eil76



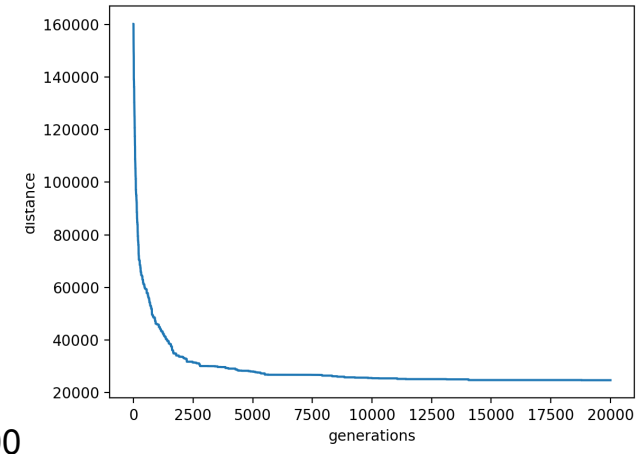
eil51



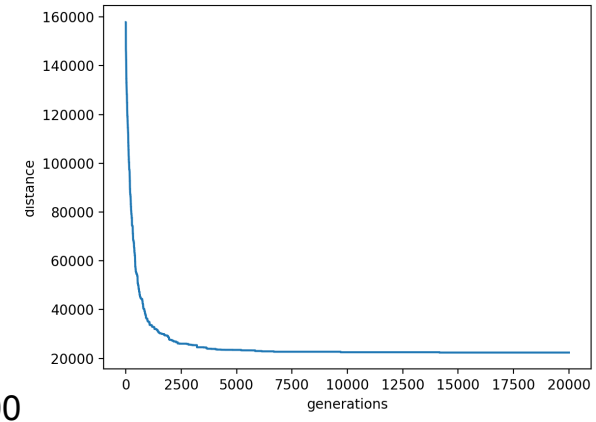
eil101



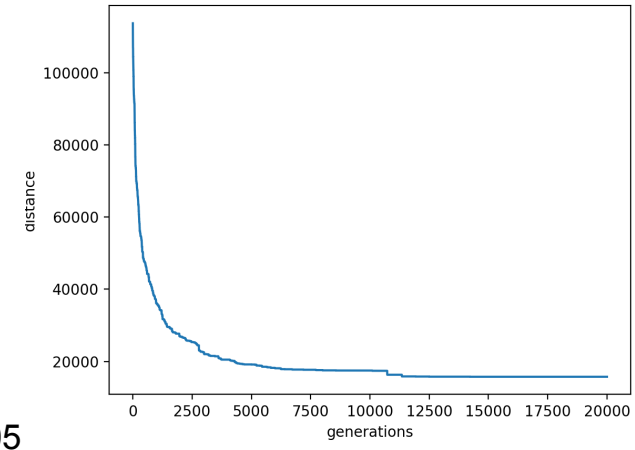
st70



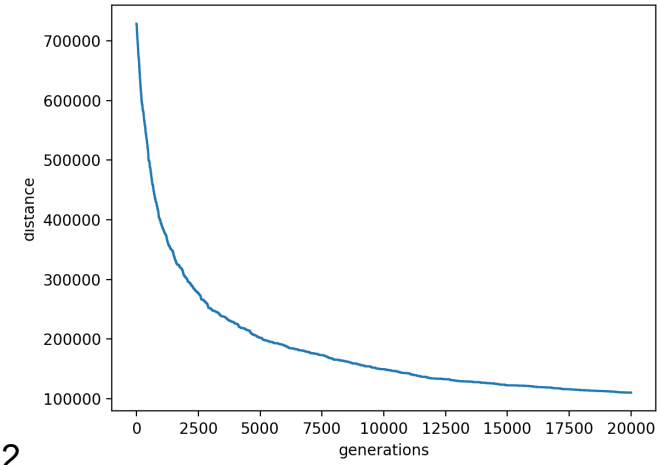
kroa100



kroc100



lin105



pcb442