**Experiment report 1：unconstrained benchmark problems**

1. **Objectives**
2. Review the intelligent optimization methods, mathematical model, and unconstrained optimization problems.
3. Learn to code the intelligent optimization algorithms, mathematical model, and unconstrained optimization problems with MATLAB.
4. **Tasks**
5. Learn to build mathematical model of unconstrained optimization problems.
6. Learn to code one of intelligent optimization algorithms.
7. Learn to tune the parameters or operators of intelligent optimization algorithms.
8. **Contents**

Try to use one of the intelligent optimization algorithms to solve the following unconstrained benchmark problems. 25 independent runs should be conducted for each benchmark functions with its maximum fitness evaluations. The mean value and standard deviation should be recorded in the table. Please output the best solution(individual) you can find for each function.

1. Maximum number of fitness evaluations is 1500,000.



1. Maximum number of fitness evaluations is 1500,000.



1. Maximum number of fitness evaluations is 9000,000.



1. **Steps**
2. Build the mathematical model of the benchmark functions with MATLAB/python/C++ et al.
3. Code the classical intelligent optimization algorithm.
4. Present the control parameters setting.
5. Output the experimental results of the classical intelligent optimization problems.
6. Try to improve the algorithm and output the outcome of the improved one.
7. Use statistical analysis and figures (convergence curves) to analyze the results of the two algorithms.
8. **Experiment report**
9. Experiment result

|  |  |
| --- | --- |
| **Function 1** | |
| **Times** | **Optimal value** |
| 1 | 8.18169E-18 |
| 2 | 8.7118E-18 |
| 3 | 1.48814E-16 |
| 4 | 2.99122E-17 |
| 5 | 7.76355E-18 |
| 6 | 1.31247E-16 |
| 7 | 2.04576E-17 |
| 8 | 7.05463E-17 |
| 9 | 7.61011E-18 |
| 10 | 4.15136E-16 |
| 11 | 7.80995E-17 |
| 12 | 7.23426E-16 |
| 13 | 2.91124E-17 |
| 14 | 2.34105E-16 |
| 15 | 1.85334E-16 |
| 16 | 2.92968E-17 |
| 17 | 1.72599E-16 |
| 18 | 1.4948E-17 |
| 19 | 9.13743E-17 |
| 20 | 7.12405E-17 |
| 21 | 5.91224E-17 |
| 22 | 3.72798E-16 |
| 23 | 1.71153E-16 |
| 24 | 3.26898E-18 |
| 25 | 3.37053E-17 |

Mean optimal value = 1.24718E-16

Standard deviation = 1.63224E-16

|  |  |
| --- | --- |
| **Function 2** | |
| **Times** | **Optimal value** |
| 1 | 1.78935E-09 |
| 2 | 4.47851E-09 |
| 3 | 1.18036E-09 |
| 4 | 3.96948E-09 |
| 5 | 2.88245E-09 |
| 6 | 1.24338E-09 |
| 7 | 2.56448E-09 |
| 8 | 1.80976E-09 |
| 9 | 1.90907E-09 |
| 10 | 1.43298E-09 |
| 11 | 3.40193E-09 |
| 12 | 4.11993E-09 |
| 13 | 9.79292E-10 |
| 14 | 2.08516E-10 |
| 15 | 1.08902E-09 |
| 16 | 1.3928E-09 |
| 17 | 2.28522E-10 |
| 18 | 5.29911E-09 |
| 19 | 2.27466E-09 |
| 20 | 1.55006E-09 |
| 21 | 1.87543E-09 |
| 22 | 2.10242E-09 |
| 23 | 1.80254E-09 |
| 24 | 1.83461E-09 |
| 25 | 4.84797E-09 |

Mean optimal value = 2.25067E-09

Standard deviation = 1.35079E-09

|  |  |
| --- | --- |
| **Function 3** | |
| **Times** | **Optimal value** |
| 1 | -11969.91455 |
| 2 | -12569.48662 |
| 3 | -11370.34253 |
| 4 | -11969.91457 |
| 5 | -12569.48662 |
| 6 | -12569.48662 |
| 7 | -12569.48662 |
| 8 | -11969.91457 |
| 9 | -11969.91457 |
| 10 | -11969.91457 |
| 11 | -12569.48662 |
| 12 | -12569.48662 |
| 13 | -11969.91457 |
| 14 | -11969.91457 |
| 15 | -12569.48662 |
| 16 | -12569.48662 |
| 17 | -12569.48662 |
| 18 | -11370.34253 |
| 19 | -12569.48662 |
| 20 | -12569.48662 |
| 21 | -11969.91457 |
| 22 | -11969.91457 |
| 23 | -11969.91457 |
| 24 | -11370.34253 |
| 25 | -12569.48662 |

Mean optimal value = -12185.76051

Standard deviation = 411.2207618

1. Chart

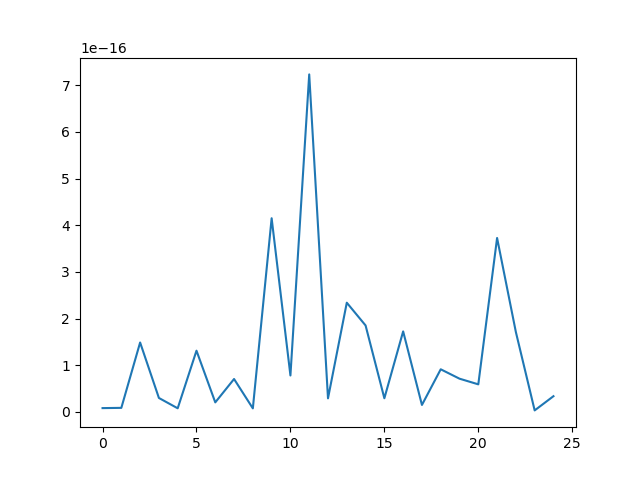


图 1 optimal value of function 1(25 times)

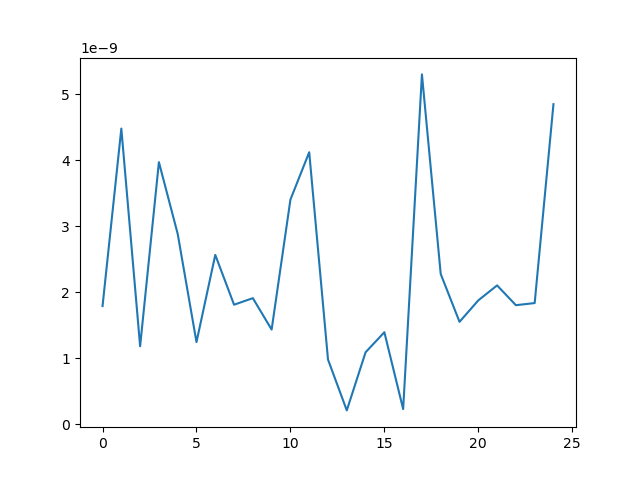


图 2 optimal value of function 2(25 times)

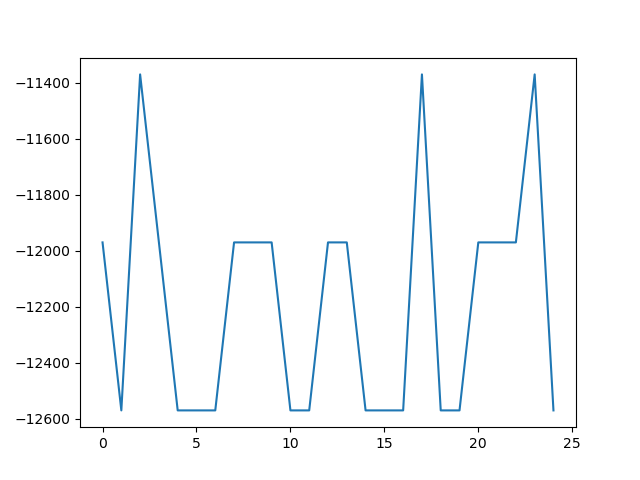


图 3 optimal value of function 3(25 times)

1. Improvement
2. In function 3, the PSO can not ensure that the decision variables are in the value range when the particles travel. If you cannot restrict the range of values, then you cannot find the optimal value. In this experiment, the strategy of assigning special value is adopted if the range is out of it.
3. Conclusion

For function 1 and function 2,25 independent runs result in 25 optimal values, each of which is on the order of , and the average optimal value is very close to 0, so we consider the optimal value of the first function to be 0.

For function 3, we get an optimal value of -11370.34253.

1. Experience and suggestions
2. Particle swarm optimization algorithm cannot guarantee that the values of particles are within the required value range, so it can only estimate the solution range first. If it is included in the required value range, then use particle swarm optimization algorithm to solve the specific value.
3. Particle swarm optimization algorithm in solving, if the number of particles is small, it is difficult to find the global optimal solution, if the number of particles is large, then it has a greater chance to traverse to the global optimal solution, because the traveling area of a single particle is limited, the more the number of particles, the larger the traveling area, the more chance to find the optimal solution.
4. If the PSO exceeds the required value range, the methods of re-randomizing, bouncing particles and assigning specific values can be adopted. In the "assign specific value" method, an artificially estimated approximate optimal value can be assigned. For example, in problem 3 of Experiment 1, if the method is out of range, re-randomization is adopted. After several iterations, it is found that when is between 400 and 500, the fitness function is larger. Therefore, I guess that , which produces the optimal solution, is between 400 and 500. Instead, I assign a specific value to if it is beyond the range, and set it to a certain number between 400 and 500 to make the particle move more focused in this range and enhance the local search ability.
5. Code

See the attachment PSO experiment 1 problem1.py, PSO experiment 1 problem2.py, PSO experiment 1 problem3.py.

**Experiment report 2：unconstrained benchmark problems**

1. **Objectives**
2. Review the intelligent optimization methods, mathematical model, and TSP.
3. Learn to code the intelligent optimization algorithms, mathematical model, and TSP.
4. **Tasks**
5. Learn to build mathematical model of TSP.
6. Learn to code one of intelligent optimization algorithms.
7. Learn to tune the parameters or operators of intelligent optimization algorithms.
8. **Contents**

Try to use one of the intelligent optimization algorithms to solve the following TSP. 5 independent runs should be conducted for TSP. The value of best route and city visiting sequence should be recorded in the table. Please output the best solution(individual) you can find for each run.

**TSP: 48 Cities data：**

1 6734 1453

2 2233 10

3 5530 1424

4 401 841

5 3082 1644

6 7608 4458

7 7573 3716

8 7265 1268

9 6898 1885

10 1112 2049

11 5468 2606

12 5989 2873

13 4706 2674

14 4612 2035

15 6347 2683

16 6107 669

17 7611 5184

18 7462 3590

19 7732 4723

20 5900 3561

21 4483 3369

22 6101 1110

23 5199 2182

24 1633 2809

25 4307 2322

26 675 1006

27 7555 4819

28 7541 3981

29 3177 756

30 7352 4506

31 7545 2801

32 3245 3305

33 6426 3173

34 4608 1198

35 23 2216

36 7248 3779

37 7762 4595

38 7392 2244

39 3484 2829

40 6271 2135

41 4985 140

42 1916 1569

43 7280 4899

44 7509 3239

45 10 2676

46 6807 2993

47 5185 3258

48 3023 1942

1. **Steps**
2. Build the mathematical model of the TSP .
3. Code the classical intelligent optimization algorithm (eg: Tabu Search).
4. Present the control parameters setting.
5. Output the experimental results of the classical intelligent optimization problems.
6. Try to improve the algorithm and output the outcome of the improved one.
7. Use statistical analysis and figures (convergence curves) to analyze the results of the two algorithms.
8. **Experiment report**
9. Experiment result

|  |  |  |
| --- | --- | --- |
| **Times** | **Optimal value** | **Route** |
| 1 | 36089.79373 | [34, 44, 23, 9, 41, 4, 47, 38, 31, 20, 46, 19, 29, 42, 16, 26, 18, 36, 5, 27, 45, 39, 2, 21, 15, 0, 7, 8, 37, 30, 43, 17, 6, 35, 32, 14, 11, 10, 22, 12, 24, 13, 33, 40, 28, 1, 25, 3] |
| 2 | 34599.87656 | [4, 47, 31, 38, 24, 12, 20, 46, 19, 32, 14, 39, 8, 0, 7, 37, 30, 43, 17, 6, 29, 42, 16, 26, 18, 36, 5, 27, 35, 45, 11, 10, 22, 13, 33, 2, 21, 15, 40, 28, 1, 25, 3, 34, 44, 9, 23, 41] |
| 3 | 34795.75296 | [11, 14, 37, 30, 43, 17, 6, 27, 29, 42, 16, 26, 18, 36, 5, 35, 45, 32, 19, 46, 20, 31, 38, 47, 4, 41, 9, 23, 44, 34, 3, 25, 1, 28, 40, 33, 13, 24, 12, 22, 2, 15, 21, 0, 7, 8, 39, 10] |
| 4 | 34351.31059 | [11, 10, 22, 2, 21, 15, 40, 33, 28, 1, 25, 3, 34, 44, 23, 9, 41, 4, 47, 31, 38, 24, 13, 12, 20, 46, 19, 14, 39, 0, 7, 8, 37, 30, 43, 35, 29, 42, 16, 26, 18, 36, 5, 27, 6, 17, 45, 32] |
| 5 | 36476.97759 | [43, 6, 27, 5, 36, 18, 26, 16, 42, 29, 19, 46, 20, 12, 13, 24, 38, 31, 23, 9, 44, 34, 3, 25, 41, 47, 4, 28, 1, 40, 33, 22, 10, 11, 32, 35, 17, 45, 14, 39, 2, 21, 15, 0, 7, 8, 37, 30] |

Mean optimal value = 35262.74229

Standard deviation = 853.9975643

1. Chart

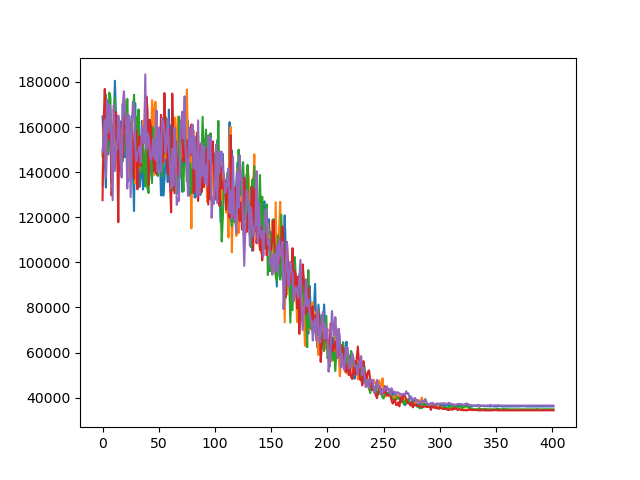


图 4 fitness fluctuations

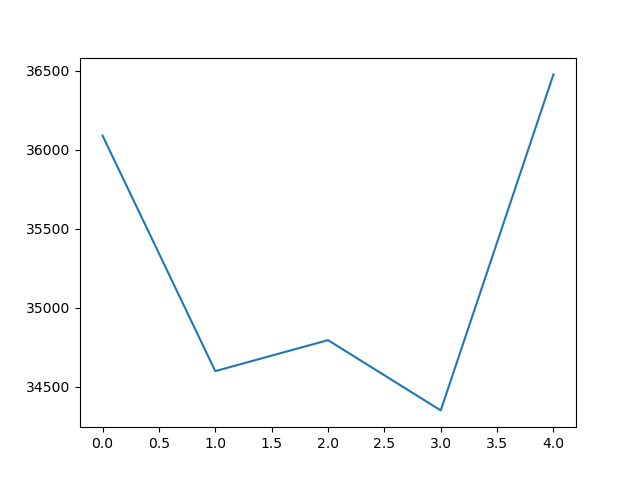


图 5 optimal value fluctuations

1. Improvement

Considering that the perturbation function designed by the simulated annealing algorithm to solve the TSP problem is to randomly select two points on the travel path for exchange, two neighboring perturbations may exchange the same pair of waypoints when random exchange points are generated, which reduces the efficiency. The improved simulated annealing algorithm introduces tabu list of length 3 into the disturbance function to avoid the above problems. The results of five independent runs after improvement are as follows.

|  |  |  |
| --- | --- | --- |
| **Times** | **Optimal value** | **Route** |
| 1 | 34307.24264 | [40, 15, 21, 0, 7, 8, 37, 30, 43, 35, 29, 42, 16, 26, 18, 36, 5, 27, 6, 17, 45, 32, 14, 39, 11, 19, 46, 20, 38, 31, 23, 9, 44, 34, 3, 25, 41, 1, 28, 4, 47, 24, 13, 12, 10, 22, 2, 33] |
| 2 | 34139.53825 | [36, 5, 35, 45, 32, 19, 46, 20, 12, 24, 47, 4, 28, 1, 41, 25, 3, 34, 44, 9, 23, 31, 38, 13, 33, 40, 15, 21, 2, 22, 10, 11, 14, 39, 8, 0, 7, 37, 30, 43, 17, 6, 27, 29, 42, 16, 26, 18] |
| 3 | 36380.27373 | [40, 28, 1, 25, 3, 34, 44, 9, 23, 41, 4, 47, 38, 31, 20, 19, 35, 5, 36, 18, 26, 16, 42, 29, 27, 6, 17, 43, 30, 37, 11, 10, 22, 13, 24, 12, 46, 32, 45, 14, 39, 8, 7, 0, 15, 21, 2, 33] |
| 4 | 36824.72078 | [40, 15, 21, 0, 7, 8, 37, 30, 43, 17, 6, 35, 32, 11, 10, 12, 13, 22, 2, 39, 14, 45, 27, 5, 36, 18, 26, 16, 42, 29, 19, 46, 20, 24, 38, 31, 23, 9, 44, 34, 3, 25, 41, 1, 28, 4, 47, 33] |
| 5 | 34580.98939 | [16, 26, 18, 36, 6, 17, 43, 30, 37, 8, 7, 0, 21, 15, 2, 40, 33, 47, 4, 28, 1, 41, 25, 3, 34, 44, 9, 23, 31, 38, 20, 12, 24, 13, 22, 10, 46, 19, 11, 39, 14, 32, 45, 35, 27, 5, 29, 42] |

Mean optimal value = 35246.55296

Standard deviation = 1124.873038

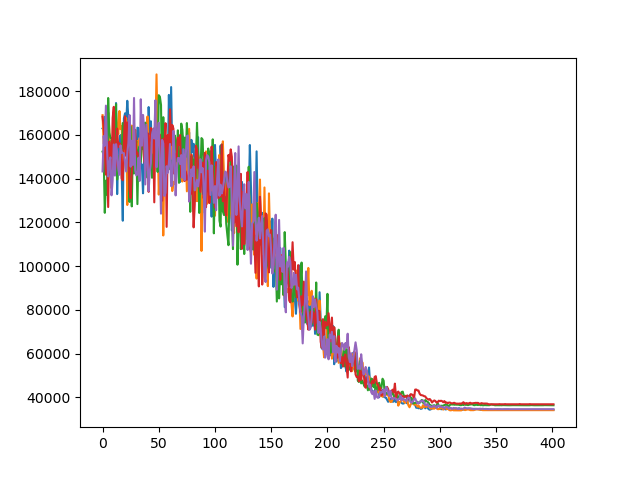


图 6 improved fitness fluctuations

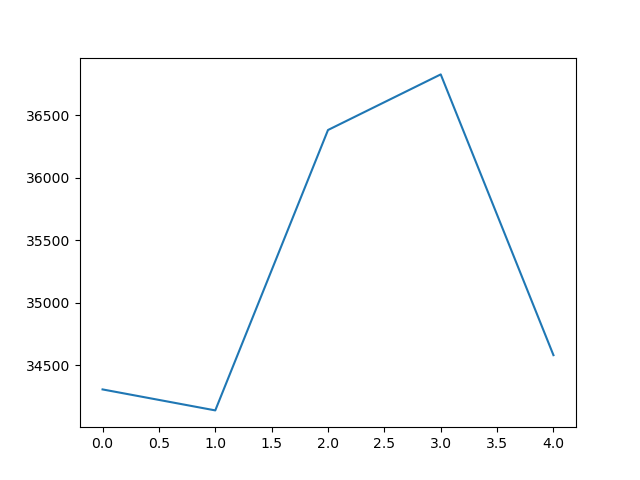


图 7 improved optimal value fluctuations

1. Conclusion

As can be seen from the figure, the average shortest path after the improvement is lower than that before the improvement, but the variance of the optimal value obtained from five independent runs is increased. Among them, The improved simulated annealing algorithm found the shortest path in this experiment [36, 5, 35, 45, 32, 19, 46, 20, 12, 24, 47, 4, 28, 1, 41, 25, 3, 34, 44, 9, 23, 31, 38, 13, 33, 40, 15, 21, 2, 22, 10, 11, 14, 39, 8, 0, 7, 37, 30, 43, 17, 6, 27, 29, 42, 16, 26, 18] and the length of the path is 34139.53825.

Because the improved results have a larger standard deviation, we believe that the improved simulated annealing algorithm has a stronger global search ability and can converge to multiple local optimal solutions in multiple independent runs. This is also in line with the original intention of introducing tabu search, which is to traverse as many candidate solutions as possible in a limited number of iterations.

1. Experience and suggestions

SA uses a single solution to iterate, and has a poor global search ability compared with GA and PSO. Because GA and PSO can use a group of solutions for iteration, each independent run can obtain more local optimal solutions, and have a greater probability of finding the global optimal solution.

When the intelligent optimization algorithm is used to solve the unconstrained problem, the GA or PSO can be used for global search to obtain some local optimal solutions, and then the SA can be used to search in the field of these local optimal solutions.

1. Code

See the attachment SA experiment 2.py, SA + tabu experiment 2.py.