中山大學

SUN YAT-SEN UNIVERSITY

实验报告

实验三：搜索算法

姓　　名　　元朗曦

学　　号　　23336294

班　　级　　计算机八班

专　　业　　计算机科学与技术

学　　院　　计算机学院

一、算法原理

启发函数设计：

- 曼哈顿距离：计算每个非零数字当前坐标与其目标坐标之间水平和垂直距离之和。
- 线性冲突：在同一行或列中，如果两个目标位于同一行（或列）的数字顺序错误，则它们必然需要至少多移动两步，因而额外增加代价。

两者叠加，使得启发函数更准确地估计当前状态到目标状态的移动步数。

A* 算法

利用启发函数为每个状态估计总代价（路径代价＋启发代价），并使用一个优先队列（堆）来保证总是扩展总代价最小的状态。 随着状态的扩展，算法不断更新状态的最佳路径代价，直到找到目标状态或耗尽所有状态。

IDA* 算法：

采用迭代加深的方式，使用深度优先搜索在一个动态增加的阈值内探索状态空间。 每次迭代阈值根据当前搜索中最小超过阈值的总代价更新，从而不断逼近目标状态。 这两种方法都利用了精心设计的启发函数，以减少状态空间搜索的数量，提高求解效率。

遗传算法求解 TSP

- 初始种群生成：随机生成多个城市排列（路径），作为初始种群，每个路径都是一个候选解。
- 适应度评估：通过计算路径的总长度（使用欧几里得距离求和，并构成一个环路）来评估每个候选解的好坏。路径越短适应度越高。
- 选择操作：使用锦标赛选择（tournament selection），从种群中随机选取几个候选解，然后选取路径长度最短的个体作为父代。
- 交叉操作：采用顺序交叉（order crossover），确保生成的子代保持合法的城市排列，继承父母部分基因的顺序信息，从而保证每个城市只出现一次。
- 变异操作：对子代应用交换变异（swap mutation），以一定的概率随机交换两个城市的位置，增加种群的多样性，防止局部最优。
- 迭代更新：重复选择、交叉、变异过程，经过多代迭代，不断更新种群以寻找更短的巡游路径。

最终，最佳个体即所求解的最优路径和对应的路径长度会被输出。

二、代码展示

见附件 `week5-2.py` 和 `week5-4.py` 文件。

三、实验结果

`week5-2.py` 运行输出如下：

```
Test case 0:
A* solved in 1 step(s), took 0.000101 second(s): R
IDA* solved in 1 step(s), took 0.000099 second(s): R
----------------------------------------
Test case 1:
A* solved in 22 step(s), took 0.004575 second(s): LDRUURDLURULDLDDLURRRD
IDA* solved in 22 step(s), took 0.001475 second(s): URDLLDRUURULDLDDLURRRD
----------------------------------------
Test case 2:
```

```
A* solved in 49 step(s), took 9.833782 second(s):
LLDLURDRULDDLUURDDDLURRRUULLDDDRULDLUURRRDLLURDDR
IDA* solved in 49 step(s), took 16.643323 second(s):
LLDLURDRULDDLUURDDDLURRRUULLDDDRULDLUURRRDLLURDDR
----------------------------------------
Test case 3:
A* solved in 15 step(s), took 0.000513 second(s): RRRUULLLURDDDRR
IDA* solved in 15 step(s), took 0.000292 second(s): RRRUULLLURDDDRR
----------------------------------------
Test case 4:
A* solved in 48 step(s), took 21.193891 second(s):
DLLURRURDDLLURURULDRDLULLDDRRULUULDDDRUUURRDLDRD
IDA* solved in 48 step(s), took 25.810287 second(s):
DLLURRURDDLLURURULDRDLULLDDRRULUULDDDRUUURRDLDRD
----------------------------------------
Test case 5:
A* solved in 56 step(s), took 237.629372 second(s):
ULULDDRRULLLDRRULLDRUULURRDLLURRDRDDLURUULDLLDRRULURDRDD
IDA* solved in 56 step(s), took 239.861461 second(s):
ULULDDRRULLLDRRULLDRUULURRDLLURRDRDDLURUULDLLDRRULURDRDD
----------------------------------------
Test case 6:
A* solved in 62 step(s), took 578.837128 second(s):
DRDLURULDRDDLUURRDRDLLURURDDLUUURDDDLUUULLDDRDRUUURDDDLULLDRRR
IDA* solved in 62 step(s), took 901.344829 second(s):
DRDLURULDRDDLUURRDRDLLURURDDLUUURDDDLUUULLDDRDRUUURDDDLULLDRRR
----------------------------------------
```

week5-4.py 运行输出如下：

```
Generation 0: Best length = 298417.63
Generation 50: Best length = 282435.44
Generation 100: Best length = 282435.44
Generation 150: Best length = 280264.07
Generation 200: Best length = 280264.07
Generation 250: Best length = 276758.26
Generation 300: Best length = 276758.26
Generation 350: Best length = 276758.26
Generation 400: Best length = 276758.26
Generation 450: Best length = 276758.26
Best tour: [593, 677, 407, 590, 137, 812, 886, 672, 138, 328, 742, 109, 384, 296, 87,
13, 229, 638, 120, 907, 863, 116, 651, 913, 400, 663, 765, 708, 668, 541, 354, 438,
269, 55, 557, 165, 174, 707, 383, 211, 43, 74, 869, 819, 893, 477, 794, 838, 325,
255, 633, 284, 124, 883, 31, 763, 842, 233, 872, 298, 568, 234, 836, 182, 940, 273,
743, 714, 552, 313, 613, 501, 614, 625, 352, 64, 935, 291, 792, 473, 959, 618, 347,
683, 170, 277, 48, 81, 162, 852, 621, 68, 801, 446, 114, 931, 943, 837, 191, 737,
243, 224, 817, 260, 723, 167, 583, 14, 640, 755, 117, 933, 608, 133, 317, 587, 648,
979, 553, 606, 905, 690, 611, 408, 582, 436, 125, 240, 515, 268, 253, 623, 960, 919,
232, 533, 95, 37, 439, 107, 236, 578, 486, 330, 899, 126, 67, 537, 375, 455, 345, 84,
357, 161, 29, 511, 307, 76, 241, 666, 464, 156, 145, 102, 657, 726, 487, 286, 906,
152, 676, 564, 18, 17, 570, 826, 584, 859, 731, 185, 505, 818, 822, 644, 50, 747,
685, 517, 82, 810, 239, 305, 528, 15, 451, 739, 843, 829, 481, 321, 977, 267, 1, 776,
799, 720, 350, 104, 282, 204, 732, 143, 281, 265, 760, 735, 758, 485, 920, 546, 889,
403, 96, 563, 63, 348, 713, 297, 952, 163, 150, 467, 225, 449, 442, 797, 695, 850,
861, 805, 832, 33, 519, 360, 489, 401, 751, 562, 823, 756, 514, 507, 216, 437, 892,
402, 278, 914, 550, 721, 480, 329, 53, 808, 509, 659, 609, 478, 97, 704, 110, 927, 3,
316, 483, 271, 405, 77, 599, 556, 11, 956, 59, 915, 655, 928, 111, 740, 292, 25, 266,
```

744, 670, 434, 696, 362, 361, 687, 130, 881, 414, 142, 99, 667, 675, 658, 288, 231,
215, 197, 759, 147, 420, 831, 78, 602, 198, 847, 752, 476, 969, 646, 615, 66, 601,
974, 532, 512, 622, 368, 136, 452, 865, 399, 60, 525, 144, 39, 411, 242, 594, 19,
610, 961, 753, 417, 524, 571, 538, 597, 416, 252, 729, 227, 508, 901, 343, 338, 944,
139, 41, 706, 777, 340, 168, 300, 768, 572, 381, 728, 395, 718, 834, 447, 382, 630,
925, 460, 35, 6, 559, 746, 678, 279, 970, 171, 276, 468, 890, 421, 679, 951, 558,
440, 181, 425, 22, 418, 895, 878, 34, 314, 934, 365, 493, 577, 921, 569, 958, 178,
586, 499, 787, 214, 866, 898, 129, 24, 177, 790, 692, 764, 100, 738, 108, 911, 710,
283, 46, 71, 121, 169, 334, 862, 722, 72, 900, 781, 393, 88, 949, 748, 238, 355, 153,
49, 674, 629, 942, 190, 310, 331, 761, 113, 194, 406, 802, 140, 327, 516, 870, 665,
813, 9, 855, 904, 814, 908, 853, 207, 304, 649, 626, 246, 289, 27, 275, 719, 968,
335, 887, 506, 917, 184, 413, 4, 902, 717, 656, 470, 788, 941, 769, 844, 796, 308,
631, 937, 388, 496, 80, 733, 922, 379, 851, 767, 542, 337, 535, 223, 534, 342, 603,
322, 523, 469, 757, 415, 16, 540, 588, 770, 426, 318, 245, 458, 749, 830, 456, 378,
272, 634, 598, 967, 453, 346, 529, 119, 148, 444, 397, 784, 380, 359, 885, 574, 301,
319, 448, 482, 122, 932, 450, 580, 741, 475, 712, 619, 856, 89, 410, 581, 730, 412,
8, 916, 230, 549, 390, 118, 54, 705, 311, 445, 637, 661, 820, 607, 459, 373, 689,
585, 783, 871, 828, 518, 187, 44, 189, 554, 573, 65, 565, 576, 257, 466, 877, 28,
235, 141, 471, 664, 688, 93, 457, 555, 12, 793, 30, 612, 264, 188, 86, 398, 228, 880,
643, 146, 641, 363, 367, 299, 320, 709, 91, 431, 645, 785, 70, 806, 858, 948, 500,
427, 929, 510, 924, 290, 716, 222, 963, 396, 544, 262, 404, 803, 202, 209, 938, 561,
652, 45, 180, 7, 839, 807, 220, 567, 543, 653, 536, 213, 52, 349, 250, 490, 123, 779,
261, 21, 780, 686, 647, 433, 159, 285, 660, 94, 766, 344, 975, 849, 333, 551, 868,
391, 800, 669, 201, 824, 909, 875, 423, 195, 491, 654, 295, 930, 134, 791, 827, 978,
443, 620, 192, 157, 624, 259, 750, 964, 158, 864, 966, 495, 882, 804, 155, 92, 639,
183, 386, 270, 419, 840, 736, 199, 531, 884, 504, 604, 778, 700, 888, 374, 548, 953,
85, 962, 698, 351, 218, 694, 786, 69, 502, 208, 210, 782, 324, 945, 965, 798, 484,
472, 294, 205, 315, 595, 642, 51, 616, 846, 896, 691, 494, 809, 40, 248, 682, 474,
526, 175, 263, 545, 684, 101, 681, 217, 302, 23, 435, 857, 703, 939, 635, 955, 73,
891, 498, 149, 503, 332, 387, 702, 61, 789, 727, 306, 62, 795, 833, 591, 341, 115,
186, 923, 539, 821, 530, 258, 389, 287, 976, 627, 79, 353, 196, 867, 513, 848, 309,
479, 280, 547, 592, 135, 193, 650, 176, 605, 173, 771, 754, 636, 860, 38, 256, 422,
680, 950, 42, 774, 212, 371, 520, 522, 303, 429, 20, 825, 589, 32, 358, 897, 841, 98,
879, 773, 600, 628, 462, 200, 409, 693, 673, 326, 432, 566, 715, 461, 815, 131, 105,
947, 428, 164, 274, 874, 632, 454, 151, 745, 617, 366, 206, 441, 112, 323, 247, 910,
10, 811, 47, 179, 377, 711, 364, 127, 912, 249, 372, 873, 946, 203, 724, 772, 90,
973, 336, 376, 954, 385, 339, 172, 128, 926, 254, 370, 701, 957, 560, 527, 36, 430,
854, 424, 0, 57, 579, 56, 762, 521, 699, 596, 293, 369, 2, 392, 356, 775, 237, 160,
244, 497, 971, 26, 816, 221, 845, 132, 575, 903, 83, 876, 463, 936, 103, 166, 75, 58,
394, 465, 972, 894, 488, 734, 492, 251, 226, 671, 219, 725, 5, 662, 918, 312, 697,
106, 835, 154]
Tour length: 276758.26
Elapsed time: 527.03 seconds