

基于聚类集成的蚁群优化与受限解空间的 TSP 算法

庞永明¹, 钟才明², 程 凯¹

(1. 宁波大学信息科学与工程学院, 浙江宁波 315210; 2. 宁波大学科学技术学院, 浙江宁波 315210)

摘要:蚁群算法是一种元启发式搜索算法, 能有效地解决 TSP 这类 NP 问题. 针对该算法的信息素更新机制易导致陷入局部最优的缺点, 提出了一种基于聚类集成的蚁群优化与受限解空间的 TSP 算法. 其主要思想如下: 先用三角形 TSP 算法生成初始 TSP 并构建蚁群的初始转移概率矩阵, 以减少蚂蚁选择的随机性; 然后运用 k -means 聚类集成生成关联矩阵, 作为扰动因子以优化蚂蚁对城市的选择概率, 即关联城市相互吸引, 不关联城市相互排斥, 以避免过早陷入局部最优; 最后提出一种重组受限解空间的边的方法再次优化蚁群的解. 实验结果表明, 与同类算法相比, 该算法具有较优的结果.

关键词:蚁群算法; TSP; 关联矩阵; 受限解空间

中图分类号: TP391 **文献标识码:** A doi: 10.3969/j.issn.0253-2778.2016.09.010

引用格式: 庞永明, 钟才明, 程凯. 基于聚类集成的蚁群优化与受限解空间的 TSP 算法[J]. 中国科学技术大学学报, 2016, 46(9): 780-787.

PANG Yongming, ZHONG Caiming, CHENG Kai. A TSP algorithm based on clustering ensemble ACO and restricted solution space[J]. Journal of University of Science and Technology of China, 2016, 46(9): 780-787.

A TSP algorithm based on clustering ensemble ACO and restricted solution space

PANG Yongming¹, ZHONG Caiming², CHENG Kai¹

(1. College of Information Science and Engineering, Ningbo University, Ningbo 315210, China;

2. College of Science & Technology, Ningbo University, Ningbo 315210, China)

Abstract: Ant colony algorithm (ACO) is a metaheuristic search method, which can solve efficiently NP-complete problems such as the famous traveling salesman problem (TSP). To alleviate, to some extent, ACO's drawback of getting stuck in a local optimum, a TSP algorithm based on clustering ensemble ACO and restricted solution space is proposed in this paper. The main idea is as follows: Firstly, a triangle algorithm is presented to generate initial TSPs, which are used to construct the primitive transfer probability matrix so that the randomness of trip of ants is reduced; Secondly, to avoid premature convergence, the k -means clustering method is employed repeatedly to produce co-association matrix (CM), which can be viewed as a perturbation factor and improve the selection of the next city for each ant, namely, city pairs with high values in CM are connected closely, or vice versa; Finally, a restricted

收稿日期: 2016-03-01; **修回日期:** 2016-09-17

基金项目: 国家自然科学基金(61175054)资助.

作者简介: 庞永明, 男, 1989 年出生, 硕士生. 研究方向: 机器学习与人工智能. E-mail: 891067464@qq.com

通讯作者: 钟才明, 博士/教授, E-mail: zhongcaiming@nbu.edu.cn

solution space, namely, the restricted connections, is reconsidered to improve the solutions produced by ACO. The experiments on 9 benchmarks from TSPLIB demonstrate that the proposed method outperforms some of the state-of-art TSP algorithms.

Key words: ant colony optimization; traveling salesman problem; co-association matrix; restricted solution space

0 引言

旅行商问题 (traveling salesman problem, TSP) 是一个经典的 NP 组合优化问题, 即一个旅行者从起点城市出发, 经过且只经过每个城市一次, 再回到起点. TSP 最优解就是旅行者所经历过的最短路径^[1]. 多年来, 研究人员不断地探究该问题, 以至文献中出现了大量的 TSP 算法, 其中智能算法具有一定的代表性. 这类智能算法大致包括: 遗传算法 (genetic algorithm, GA)、粒子群算法 (particle swarm optimization, PSO)、模拟退火 (simulated annealing algorithm, SA) 和蚁群算法 (ant colony optimization, ACO).

GA 使用交叉、变异操作, 合理运用基因片段求解 TSP^[2]. SA 是基于固体物质的退火过程与组合优化问题的相似性, 赋温度以一个初始值, 依照 Metropolis 准则不断优化^[3]. ACO 是模仿蚂蚁觅食过程中蚂蚁遗留信息素, 并通过其作为引导寻找食物源, 运用该原则及其内部的反馈机制不断的优化, 求解 TSP^[4-5]. 针对 ACO 信息素更新机制, Stutzle 等提出了最大-最小蚁群系统 (MMAS), 其将信息素增量限定在一个阈值内, 克服局部收敛缺陷, 并以此优化算法来求解 TSP^[6]. Dorigo 等在 1995 年提出 Ant-Q, 其改进 ACO 增加信息素的机制, 并应用于 TSP 的求解^[7]; 吴俊强和欧阳爱佳将蚁群算法混合 delete-cross 以求解 TSP^[8].

随着多种智能算法的发展, 针对 TSP 问题人们尝试将各种智能算法以及 k -opt^[9] 进行组合, 得到了较好的效果, 如 SA 结合 ACO、PSO 求解 TSP^[10]; 吉林大学黄兰于 2003 年提出用 2-opt 优化蚁群的解^[11]; DiCesare 等将蚁群算法、粒子群算法、3-opt 混合, 通过 PSO 来优化 ACO 中的参数 α 、 β , 再通过 3-opt 改进以上混合智能算法的解^[12]. 该方法存在如下的缺点: 求解 TSP 的过程中没有考虑到城市之间的具体分布和内在关联, 且 3-opt 将所有的边集作为候选边, 有较高冗余. 本文提出了一种混合算法, 不仅尽量避免蚂蚁寻找候选城市的盲目性, 并且

考虑了城市的分布和内在联系, 以及在受限解空间的条件下运用 k -opt, 进一步优化解.

1 蚁群算法求解 TSP 的算法过程

1.1 传统蚁群算法

蚁群算法 (ACO)^[4,5], 是意大利科学家 Dorigo 通过研究仿生学中蚂蚁觅食过程, 会遵循“信息素较浓的路线更近”的原则^[5,13], 所提出. 此算法是通过信息素不断迭代, 蚂蚁反复选择高浓度信息素的边以找到一条最优路径. 蚁群算法分为 (ant system)^[4] 和 (ant colony system)^[14], 其主要区别在于城市的选择和信息素更新策略. ACS 算法初始时, 蚂蚁对每个候选城市的选择概率是相同的, 即每条边的信息素浓度一致. 蚂蚁依据信息素浓度大小使用轮盘赌策略选择下一个城市 (从第 i 个城市到第 j 个城市), 而信息素浓度即为城市间的转移概率. 转移概率公式为:

$$P_{ij}^k = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{s \in \text{allow}_k} [\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}, & s \in \text{allow}_k \\ 0, & s \notin \text{allow}_k \end{cases} \quad (1)$$

式中, $\tau_{ij}(t)$ 表示 t 时刻蚂蚁在 ij 段上信息素浓度, k 表示第 k ($1 \leq k \leq M$) 只蚂蚁, M 为城市数, η_{ij} (启发函数) 等于 $1/d_{ij}$, 表示两城市之间的期望程度, $s \in \text{allow}_k$ 为第 k 只蚂蚁待访问集合, α 表示信息素重要度, β 表示启发式函数重要度. 每只蚂蚁旅行完所有城市之后, 会遗留信息素, 用此更新原有信息素矩阵. $\Delta\tau_{ij}^k$ 表示第 k 只蚂蚁在城市 i , 城市 j 上释放的信息素浓度. Dorigo 等将 $\Delta\tau_{ij}^k$ 归结为三种模型: 蚁量、蚁密、蚁周^[4-5]. 本文采用蚁周模型, 公式如下:

$$\Delta\tau_{ij}^k = \begin{cases} Q/L^k, & L \text{ 为蚂蚁旅行距离} \\ 0, & \text{其他} \end{cases} \quad (2)$$

L^k 代表的是第 k 只蚂蚁周游路径的长度, Q 为一个常数.

下面公式(3)表示, 蚁群算法中所有蚂蚁遗留的

信息素都将作为有效信息作用于全局,这也符合群智能中的群体性特点,而公式(4)表示信息素的实时更新, ρ 是一个挥发常数.

$$\Delta\tau_{ij} = \sum_{k=1}^n \Delta\tau_{ij}^k \quad (3)$$

$$\Delta\tau_{ij}(t+1) = (1-\rho)\Delta\tau_{ij}(t) + \Delta\tau_{ij} \quad (4)$$

1.2 优化蚁群算法

针对传统蚁群算法初始时蚂蚁寻找候选城市具有盲目随机性,提出 FM 矩阵以使蚂蚁具有选择性地走向下一城市. 传统蚁群中两城市之间信息素的覆盖原则是蚂蚁从贪婪搜索开始到蚂蚁找到信息素浓度最高的路径为止不断将信息素遗留累积叠加,从统计学角度引入 FM,既避免了蚂蚁初始寻向候选城市的贪婪搜索,又可以使蚂蚁在已有候选城市的情况下更有效地遗留信息素. 本文中提出 CM 矩阵,是改进传统蚁群容易陷入局部最优的陷阱的缺点,考虑整体布局,依最远城市间不可能相连的连接原则,相邻城市间具有更大可能的连接性,以此方式作为扰动因子增加蚂蚁对候选城市选择的精确性.

1.2.1 初始矩阵(信息素矩阵)的生成

文献[15]中的研究认为,蚁群算法具有较强的鲁棒性,用该方法求解 TSP 时,对初始路线的要求不高,即蚁群算法的求解结果不依赖于初始路线的选择,而且在搜索过程中不需要进行人工调整. 本文通过分析其初始矩阵发现,改变初始矩阵对最后路线是有影响的,如表 1、2 所示. 表 1 中反映蚂蚁寻找候选城市的概率是相等的,因而起始解具有随机性,使蚂蚁更容易陷入局部收敛的陷阱;若将蚂蚁的初始矩阵改进,如表 2 所示,蚂蚁初始选择更具有倾向性,并且用每次旅行后遗留的信息素对表 2 进行修改,则蚂蚁较易跳出局部最优.

为避免表 2 中相邻城市连接出现小概率事件,本文要求蚂蚁数等于城市数,并且多次统计初值以增加精确性.

表 1 蚂蚁寻找城市的初始矩阵

城市	1	2	3	4	5
1	0	1	1	1	1
2	1	0	1	1	1
3	1	1	0	1	1
4	1	1	1	0	1
5	1	1	1	1	0

表 2 更改过的蚂蚁搜索矩阵

Tab. 2 Improved matrix

城市	1	2	3	4	5
1	0	0.9	0.05	0.4	0.01
2	0.9	0	0.02	0.001	0.8
3	0.05	0.02	0	0.7	0.6
4	0.4	0.001	0.7	0	0.01
5	0.01	0.8	0.6	0.01	0

1.2.2 初始矩阵的生成过程

初始矩阵的生成方法,本文称为三角形 TSP 法,主要思想为:反复随机添加一个城市与已知的城市连接形成一个哈密顿回路^[16],使得添加该城市后,增加的两条路径与减去一条路径的总代价最小,直到该哈密顿回路包含所有的城市,如图 1 所示. 多次以三角形 TSP 法完成 Time 条回路,再统计每个城市的候选城市出现次数完成一个频次矩阵(该矩阵整体除以 Time),本文定义其为 FM,将其作为新的初始矩阵来代替原始矩阵. 具体实现如算法 1.1 所示.

算法 1.1 三角形 TSP 法以及初始矩阵的生成

输入:初始化城市个数 M,旅行路径 Tour

输出:Tour,FM

随机选取 3 个城市 a, b, c 设已选城市为 Idx

While length (Idx) < M

随机选取一个城市 d

For i = 1:length (Idx)

三种连接方式分别为 L_1, L_2, L_3 找出花费代价最小的,列举其中一种 $\Delta = \overline{ad} + \overline{cd} + \overline{ab} + \overline{bc} - \overline{ac}$

End

Idx ← [Idx, d];

Tour ← Idx 城市连接排序

End

%统计频次矩阵 FM

For j = 1:Time

统计每个城市的候选城市出现的频次

For m = 1:M

同一候选城市出现一次加 1

End

End

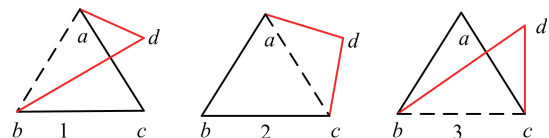


图 1 三角形法生成完整回路

Fig. 1 Triangle method for TSP

1.2.3 关联矩阵(CM)的由来以及生成

Feng 等基于 Fuzzy c-means 划分数据集^[17],对划分后的数据集分块进行路径规划,而本文用 k-means 划分数据集.图 2 中是 k-means 对 Eil101^[18]数据集的划分.其中划归类数 n 为 3,划分次数为 100(列举其中 3 次),本文中的 n 是 k-means 划分类数,从 $(1, \sqrt{n}]$, $(\sqrt{n}, 2\sqrt{n})$ 中选取正整数,该取值范围来自选取聚类算法 k-means 时通用经验值,通过实验反复评测比较算法能够取到的最优路径,选取其中最优的最优路径所对应的 n .从直观角度考虑,图 2(A)中,a 区域和 b 区域中城市没有关联,因此传统 ACS 只用欧式距离作为衡量两城市之间的关联度精确度不高.本文提出用 CM 作为一个扰动因子来提高城市之间关联性信息.图 2(C)中,e 区域中城市与非同一类的 f 区域中城市是有关联性的,但与 g 区域中城市没有关联,因此在本文中须同时将信息素增量添加在 CM 内,其添加大小,方式与该增量向 $\tau_{ij}(t)$ 中添加大小以及方式相同,见公式(3,4,6).这样进一步改进城市之间的关联度,使得该矩阵趋于合理化.如图 2(A)和 2(B)所示,c1 与 c2 区域中的城市是各类的交界城市,为了合理规划这部分城市就需要用多次 k-means 聚类方法来划分.文献[19]中提出关联矩阵 CM 的统计原则,简述为是将城市之间可能连接(或者说是划归为一类)的城市出现的频次统计,见图 3,具体统计方法如算法 1.2.本文以 CM 优化蚁群操作如下公式 5 所示,CM 在公式中被记为 τ , γ 为一个参数.

$$P_{ij}^k = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta [Tau(t)]^\gamma}{\sum_{s \in allow_k} [\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta [Tau(t)]^\gamma}, & s \in allow_k \\ 0, & s \notin allow_k \end{cases} \quad (5)$$

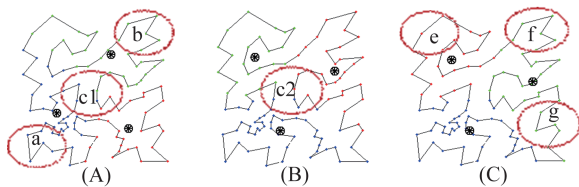


图 2 CM 矩阵的意义

Fig. 2 Significance of CM

算法 1.2 CM 矩阵的生成

输入:城市坐标 xy,类个数 n,划分次数 N
输出:CM 矩阵

```

For l = 1:N
  For i = 1:n
    Index ← Kmeans (xy, n)
  End
  For j = 1:N
    Index 中每两个城市划分在一起的次数 C
  End
  多次统计并叠加 C
End
CM ← C / N
    
```

	x1	x2	x3	x4	x5	x6
x1	2/2	2/2	1/2	0	0	0
x2	2/2	2/2	1/2	0	0	0
x3	1/2	1/2	2/2	1/2	0	0
x4	0	0	1/2	2/2		
x5	0	0	0	1/2	2/2	2/2
x6	0	0	0	1/2	2/2	2/2

图 3 关联矩阵^[20]

Fig. 3 Co-association matrix

1.2.4 优化蚁群算法的具体过程

本文提出优化蚁群算法,解决蚂蚁在一组城市 M 之间,寻找一条最短哈密顿回路问题,具体步骤见算法 1.3.

算法 1.3 优化蚁群算法

输入:初始矩阵为 FM,令 $\tau = FM$,统计关联矩阵 CM,令其为 τ ,城市数 M,信息素挥发因子 ρ ,迭代次数 iter,城市坐标 xy,赋值给 α, β, γ .

输出:最优的哈密顿回路

```

For i = 1:iter
  Idx ← Randperm (M)
  For j = 1:M
    For p = 1:M
      每只蚂蚁选择下一城市的转移概率依据:
      P_{ij} = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta [Tau(t)]^\gamma}{\sum_{s \in allow_k} [\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta [Tau(t)]^\gamma}, & s \in allow_k \\ 0, & s \notin allow_k \end{cases}
      轮盘赌选择候选城市
    End
  End (计算出 M 条路径,记为 l)
  依据公式(3),(4),(6)得出信息素增量 Δτ
  [FM, Tau] ← Δτ
End
    
```

由于蚂蚁会在旅行过的路径上留下信息素,用信息素增量衡量信息素遗留的多少,本文中将此增量同时叠加到信息素矩阵和 CM 中.信息素增量

$\Delta\tau_{ij}^k$ 的求法如下:

$$\Delta\tau_{ij}^k = \frac{Q}{l_k - \min(l)} \quad (6)$$

表示该轮所有蚂蚁行走路径与当前最优路径偏差的修正. l_k 表示第 k 只蚂蚁的旅行路径长度. $\min(l)$ 为当前一轮蚂蚁旅行路径的最小值,该公式的思想来源于粒子群,即从当前局部最优不断衍化出全局最优^[20], Q 为一个常数.

2 受限解空间以及对生成解的优化

假设解空间中的每条边都作为集合的解,完全无向图 $G = (V, E)$, $V = \{v_1, v_2, \dots, v_M\}$, $E = \{(r, s) | r, s \in V\}$ 中总共有 $M(M-1)/2$ 条边,本文自适应选择一个最优边数的阈值 NumEdge,将解空间压缩限制在一个范围内.具体过程如下:把初始矩阵 FM 中每个城市的候选城市依概率降序排列,将排序中出现概率以数量级减小处作为节点.将该节点设定为阈值.如图 4 所示(Eil101 数据集中城市 1 与候选城市的连接概率图);用受限解空间的边集和 k -opt 思想继续优化本文提出的优化蚁群算法的解,受限解空间用法示意图,如图 5.具体操作见算法 2.1.

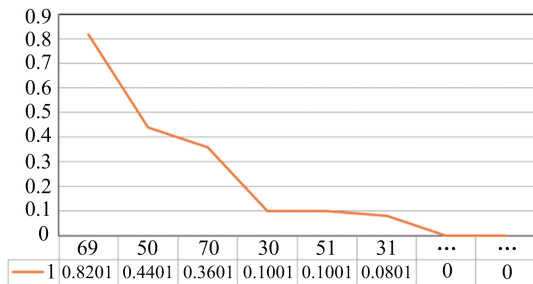


图 4 候选城市概率

Fig. 4 Probability of candidate city

算法 2.1 受限解空间求法及对路径的优化

输入:待被优化蚁群的解 Tour,城市数 M ,初始矩阵 FM

输出:优化后的回路

For $i = 1:M$

Set (i) ← FM 的第 i 个城市的候选城市以概率降序后选取节点

End

降序后的候选城市数只选取 NumEdge 个,构成受限解空间,记为 Edge.

For $j = 1:M$

将 Tour 中不同于 Edge(j)的边换掉,断成多个回路,

再以 Edge 中的边用 k -opt,以最小代价方式连接,如图 5.

End

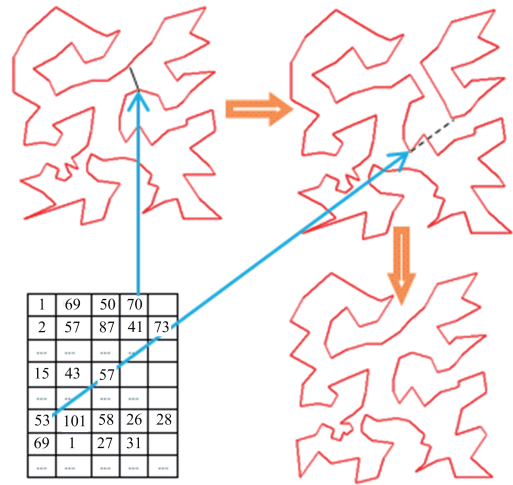


图 5 解空间的用法

Fig. 5 The function of restricted solution space

3 实验结果

本文在计算机主频 3.4HZ,处理器 i7,内存 16G,软件 matlab2014a 的运行环境下,测试 Eil51、Berlin52、St70、Eil76、Rat99、Kroa100、Lin105、Eil101、Ch150 这九个数据集(城市数分别为 51、52、70、76、99、100、105、105、101、150)的最短路径,并与已知最优值比较,该数据集下载来自 TSPLIB^[18],已知最优值来自文献[21],表 3 中分别列出数据集经过 50 次测试,统计出最大值(Worst)、最小值(Best)以及平均值(Average)、平均值与已知最优值的偏差(SD)和误差率(Error),误差率的计算如下:

$$\text{Error}(\%) = \frac{\text{平均值} - \text{已知最优值}}{\text{已知最优值}} \times 100 \quad (7)$$

测试数据集的所有参数指标见表 4,其中该算法中蚁群信息素的挥发率 ρ 为 0.1,常数 $Q = 300$, α, β, γ, k 经过多次实验确定. N 为 k -means 聚类的次数,Iter 为蚁群为了达到收敛所迭代总次数,Time 为初始三角形法形成回路数.表 5 比对几种情形的 ACS:传统 ACS;添加 FM 的 ACS;添加 FM、CM 的 ACS;以受限边集优化添加 FM、CM 的 ACS.经过数据分析比较发现,本文提出的算法中,当城市数目少时,CM 对数据结果有一定的影响,但随着城市数目的增加,这种影响会变得越来越明显,如表 6 中,随着城市数目的增多,带 CM 蚁群优化的数据结果更逼近已知的最优值.表 7 中是本文提出

的算法与已有算法^[10,11,13,22,23]的对比,数据表明本文算法在 5 个数据集 St70、Rat99、Kroa100、Eil101、Ch150 下,求解结果比其他算法更优,而对其他 4 个

数据集 Eil51、Berlin52、Eil76、Lin105 的求解结果与最优值的误差率也仅为 0.4、0.00、0.5、0.00.

表 3 数据集测试结果
Tab. 3 Result of test data

数据集	已知最优值 ^[18]	Best	Worst	Average	SD	Error(%)
Eil51	426	426	432	427.74	1.74	0.4
Berlin52	7 542	7 542	7 570	7 542.56	0.56	0.00
St70	675	675	679	675.86	0.86	0.12
Eil76	538	538	543	540.74	2.74	0.5
Rat99	1 211	1 211	1 229	1 215.66	4.66	0.38
Kroa100	21 282	21 282	21 379	21 293.45	11.45	0.00
Eil101	629	629	638	631.3	2.3	0.37
Lin105	14 379	14 379	14 401	14 380.69	1.69	0.00
Ch150	6 528	6 554	6 589	6 561.2	33.2	0.5

表 4 测试数据集所用参数
Tab. 4 The parameters for solving test data

数据集	α	β	γ	n/N	Time	NumEdge	Iter
Eil51	7.4	6	0.057	3/50	120	7	500
Berlin52	6.4	6	0.057	3/50	100	4	500
St70	9	7	0.08	3/50	200	10	1 000
Eil76	9.65	7.6	0.08	3/80	160	6	1 000
Rat99	15	15	0.01	3/100	200	10	1 000
Kroa100	15	15	0.01	3/100	200	6	1 000
Eil101	15	15	0.01	3/100	200	4	1 000
Lin105	15	15	0.06	3/100	200	5	1 000
Ch150	12	12	0.01	4/150	300	9	1 000

表 5 优化过程的数据比对(Average)
Tab. 5 The process of optimizing comparison (Average)

优化过程	Eil51	Berlin52	St70	Eil76	Rat99	Kroa100	Eil101	Lin105	Ch150
无初始 FM ^[13]	432.4	7 611.4	684	553.0	1 229	21 757.85	662	14 535	6 673.5
有 FM 无 CM	430.2	7 567.9	682.4	546.2	1 224.4	2 1361.21	640	14 495.38	6 662.3
有 FM 有 CM	429.8	7 564.1	680.9	544.0	1 221.5	21 358.22	638.44	14 475.67	6 659.5
有 FM、CM、受限解空间	427.7	7 542.5	675.9	540.7	1 215.6	21 293.45	631.3	14 380.69	6 561.2

表 6 CM 的作用结果

Tab. 6 Effects of CM

有无 CM	Eil51	Berlin52	St70	Eil76	Rat99	Kroa100	Eil101	Lin105	Ch150
无 CM(best)	426	7 542	677	543	1 216	21 308	632	14 456	6 657
有 CM(best)	426	7 542	676	540	1 212	21 282	630	14 379	6628

表 7 同算法比对(Average)

Tab. 7 Compare with different algorithms (Average)

算法	Eil51	Berlin52	St70	Eil76	Rat99	Kroa100	Eil101	Lin105	Ch150
ACO ^[13]	432.4	7 611.4	684	553.0	1 229	21 757.85	662	14 535	6 673.5
SA ACO PSO ^[10]	427.27	7 542.00	—	540.20	—	21 370.30	635.23	14 406.37	6 563.70
PSO-ACO-3OPT ^[11]	426.45	7 543.20	678.20	538.30	1 227.40	21 445.10	632.70	14 379.15	6 563.95
ACO with ABC ^[22]	443.39	7 544.37	700.58	557.98	—	22 435.31	683.39	—	6 677.12
ACO with 2-opt ^[23]	439.25	7 556.58	—	—	—	23 441.80	672.37	—	—
本文提出的算法	427.74	7 542.56	675.86	540.74	1 215.66	21 293.45	631.30	14 380.69	6 561.20

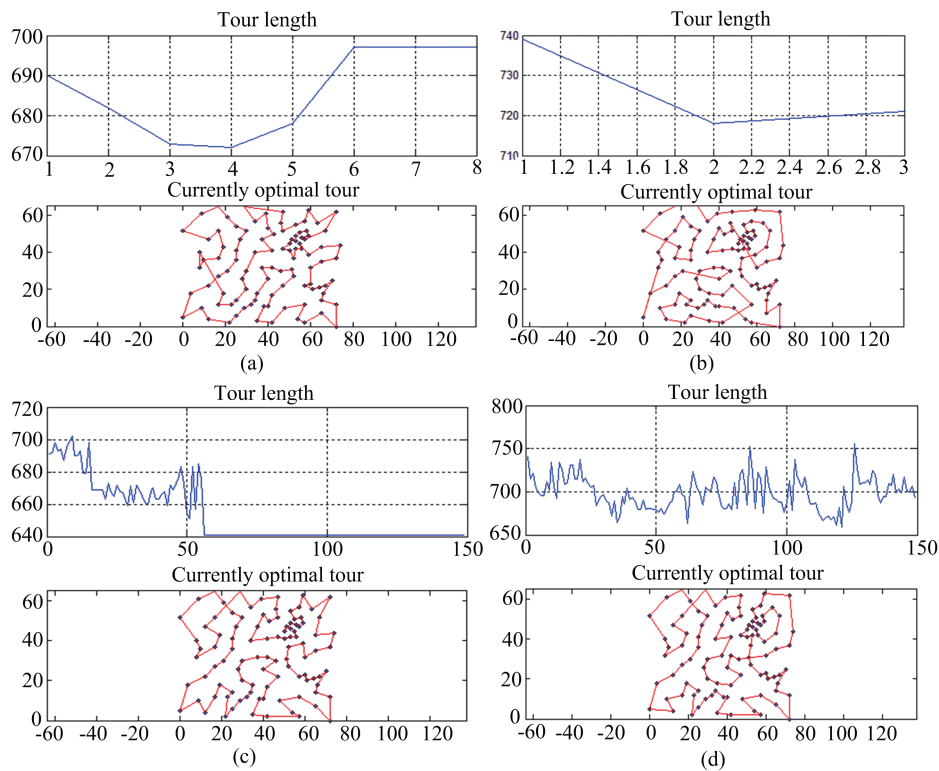


图 6 Eil101 数据结果分析(a, b, c, d)

Fig. 6 Analysis the data of Eil101 (a, b, c, d)

图 6 中, Eil101^[18] 数据集 a 和 c 分别表示有关联矩阵 CM. 通过对比图 6(a, b) 和图 6(c, d) 中的标可以发现, 有 CM 时起始的解会变小, 图 6(a) 中初值为 690, 而图 6(b) 中初值为 739, 平均值也变小了, 并且在同样迭代次数下有关联矩阵时最小值比没有关联矩阵时的更小, 如图 6(c, d) 中图 6(c) 更趋

近于最优值.

4 结论

本文提出基于 k -means 聚类优化的蚁群算法, 并在此基础上以受限解空间中的边进一步优化. 与文献中同类算法相比, 本文算法可以达到更优的效

果. 该算法的提出是考虑到传统蚁群易“早熟”, 通过分析数据及内部结构, 构建城市间关联度矩阵, 以此更好的解决该问题. 本文还在解集空间的分布上加以研究, 提出受限的解空间, 以该边结合 k -opt 思想作为优化, 达到预期较优的效果.

参考文献(References)

- [1] WIKIPEDIA. Travelling salesman problem[EB/OL]. http://en.wikipedia.org/wiki/Travelling_salesman_problem.
- [2] GREFENSTETTE J J, GOPAL R, ROSMAITA B J, et al. Genetic algorithms for the traveling salesman problem[C]// Proceedings of the First International Conference on Genetic Algorithms and their Applications. Pottsburg: IEEE Press, 1985: 160-168.
- [3] MEER K. Simulated annealing versus metropolis for a TSP instance [J]. Information Processing Letters, 2007, 104(6): 216-219.
- [4] DORIGO M, MANIEZZO V, COLORNI A. Ant system; Optimization by a colony of cooperating agents [J]. IEEE Transactions on Systems, Man and Cybernetics, Part B, 1996, 26 (1): 29-41.
- [5] DORIGO M, GAMBARDELLA L M. Ant colonies for the travelling salesman problem [J]. Bio Systems, 1997, 43(2):73-81.
- [6] STÜTZLE T, HOOS H H. Max-min ant System and local search for the traveling salesman problem [C]// Proceedings of the IEEE International Conference on Evolutionary Computation. Piscataway: IEEE Press, 1997: 309-314.
- [7] DORIGO M, BIRATTARI M, STÜTZLE T. Ant colony optimization; Artificial ants as a computational intelligence technique [J]. IEEE Computational Intelligence Magazine, 2006, 1(4): 28-39.
- [8] WU J Q, OUYANG A J. A hybrid algorithm of ACO and delete-cross method for TSP [C] //Proceedings of the International Conference on Industrial Control and Electronics Engineering. Xi'an: IEEE Press, 2012: 1694-1696.
- [9] HELSGAUN K. General k -opt submoves for the Lin-Kernighan TSP heuristic [J]. Mathematical Programming Computation, 2009, 1(2): 119-163.
- [10] CHEN S M, CHIEN C Y. Solving the traveling salesman problem based on the genetic simulated annealing ant colony system with particle swarm optimization techniques [J]. Expert Systems with Applications, 2011, 38(12): 14439-14450.
- [11] HUANG L, ZHOU C G, WANG K P. Hybrid ant colony algorithm for traveling salesman problem [J]. Progress in Natural Science, 2003, 13(4): 295-299.
- [12] MAHI M, BAYKAN Ö K, KODAZ H. A new hybrid method based on particle swarm optimization, ant colony optimization and 3-Opt algorithms for traveling salesman problem [J]. Applied Soft Computing, 2015, 30(C): 484-490.
- [13] WIKIPEDIA. Ant colony optimization algorithms[EB/OL]. https://en.wikipedia.org/wiki/Ant_colony_optimization_algorithms.
- [14] DORIGO M, GAMBARDELLA L M. Ant colony system: A cooperative learning approach to the traveling salesman problem [J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 53-66.
- [15] 百度百科. 蚁群算法[EB/OL]. [2016-12-01]<http://baike.baidu.com/link?url=nFMlbOVYTN1MEYrYUalwPCSx8pUreohumXgbIxmReFXX0rkyEdgnuyzOBK3xcaczfUiKQzHby7wrcaC-mOqCZA>.
- [16] WIKIPEDIA. Hamiltonian path[EB/OL]. [2016-12-01]https://en.wikipedia.org/wiki/Hamiltonian_path.
- [17] FENG H M, LIAO K L. Hybrid evolutionary fuzzy learning scheme in the applications of traveling salesman problems [J]. Information Sciences, 2014, 270: 204-225.
- [18] <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/tsp/>
- [19] ZHONG C M, YUE X D, ZHANG Z H, et al. A clustering ensemble: Two-level-refined co-association matrix with path-based transformation [J]. Pattern Recognition, 2015, 48(8): 2699-2709.
- [20] KENNEDY J, EBERHART R. Particle swarm optimization [C]// Proceedings of International Conference on Neural Networks. Perth, Australia: IEEE Press, 1995, 4: 1942-1948.
- [21] 数据堂. TSP 数据集[EB/OL]. [2016-12-01]<http://www.datatang.com/data/39337>.
- [22] GÜNDÜZ M, KIRAN M S, ÖZCEYLAN E. A hierarchic approach based on swarm intelligence to solve traveling salesman problem [J]. Turkish Journal of Electrical Engineering Computer Science, 2015, 23 (1): 103-107.
- [23] KAN J M, ZHANG Y. Application of an improved ant colony optimization on generalized traveling salesman problem [J]. Energy Procedia, Part A, 2012, 17(2): 319-325.