

基于回溯蚁群-粒子群混合算法的多点路径规划

刘丽珏, 罗舒宁, 高琰, 陈美妃

(中南大学信息科学与工程学院, 湖南 长沙 410083)

摘要: 景区多点路径规划问题是一个 NP-hard 问题, 相当于寻找经过起始点和特定节点的最短路径。针对多点路径规划问题, 提出了回溯蚁群-粒子群混合算法, 该算法运用弗洛伊德 (Floyd-Warshall) 算法将图进行转换并且结合了蚁群算法和粒子群算法寻找最短路径。实验结果表明, 此算法可以在小规模数据下快速找到精确解, 同时, 在较大规模数据量下, 可以得到比最大最小蚁群算法和遗传算法更好的结果。

关键词: NP-hard 问题; 最大最小蚁群系统; 弗洛伊德算法; 粒子群算法

中图分类号: TP399

文献标识码: A

doi: 10.11959/j.issn.1000-436x.2019039

Multi-point path planning based on the algorithm of colony-particle swarm optimization

LIU Lijue, LUO Shuning, GAO Yan, CHEN Meifei

School of Information Science and Engineering, Central South University, Changsha 410083, China

Abstract: The problem of multi-point path planning is a NP-hard problem, which is equivalent to finding the shortest path of a starting point and some specific node. Aiming at the problem of multi-point path planning, a retrospective ant colony-particle swarm optimization algorithm was proposed. This algorithm used Floyd-Warshall to transform the graph and combined ant colony algorithm and particle swarm algorithm to find the shortest path. The experimental results show that this algorithm can find the precise solution under small data, at the same time, under a large amount of data, can be better than the maximum minimum ant colony algorithm and genetic algorithm.

Key words: NP-hard problem, max-min ant system, Floyd-Warshall, particle swarm algorithm

1 引言

随着智慧旅游的兴起, 为了向游客提供更精准的导游服务, 不少景区推出了官方 App。单一基于道路网的导航方式已经难以满足游客对精准化、个性化导航寻径的需求。针对旅游景区个性化导航需求, 不少论文提出了路径规划算法。文献[1]通过改进 A* 算法, 融合多维动态环境信息, 提出了景点间动态路径规划的解决方案。文献[2]以鼓浪屿景区为例, 改进 Dijkstra 算法, 实现了对该景区旅游线路的规划。文献[3]考虑了景区路径的复杂性, 将景区

路径分为全景区图和子景区图, 改进蚁群算法实现了景区路径规划。文献[4]运用混合密度聚类方法识别景点热度, 改进传统蚁群算法, 提出了一种新型的热门游览景点路径规划方法。这些算法和模型仅仅解决了从起点到终点之间的最优路径规划问题。但在实际的旅游需求中, 游客或旅游大巴进行路径规划时, 更希望能得到一条从起点出发, 经过某些必去的景点 (必经点) 的最短规划路线, 以便游客无遗漏且更省时间地进行游览。

鉴于此问题, 本文参考 TSP (traveling salesman problem) 问题解决方法, 提出了一种回溯蚁群-粒

收稿日期: 2018-01-24; 修回日期: 2018-10-23

基金项目: 国家科技重大专项基金资助项目 (No.2013FY110800); 长沙计划科技基金资助项目 (No.kq1606004)

Foundation Items: National Science and Technology Major Project (No.2013FY110800), Planned Science and Technology Project of Changsha (No.kq1606004)

子群混合算法 (BAPM, backtracking ant colony-particle swarm method), 可用于解决景区中从某个旅游景点开始经过一些特定旅游点的多点路径规划问题, 该问题的解决对旅游景区多点路径规划具有一定的参考价值。

本文将景区景点抽象为点, 将景区路径抽象为有向线段, 将景区图抽象为非完全有向图, 从而对该图模型进行描述, 并且利用该图模型, 实现本文提出的 BAPM 算法。

2 基本知识

2.1 最大最小蚁群系统

最大最小蚁群系统(MMAS, max-min ant system)^[5]是目前众多蚁群算法中效率较高的算法之一, 它模拟了蚂蚁在觅食过程中寻找路径的场景, 对当前路径不断地进行优化, 从而找到最短觅食路线。该算法中, 节点 i 到节点 j 路径上的启发因子记作 η_{ij} ; 在第 t 次迭代中, 节点 i 到节点 j 路径上的信息素记作 $\tau_{ij}(t)$; η_{ij} 与 $\tau_{ij}(t)$ 的占比将会成为第 t 次迭代中蚂蚁 k 站在节点 i 选择下一个节点 j 所走路径的重要因素, 记作 $p_{ij}^k(t)$, 该因素值越大, 选择该值对应的节点的概率就越高。反复迭代该算法, 最终可以得到一条近似最短路径。在迭代过程中, 信息素 $\tau_{ij}(t)$ 根据蚂蚁寻找到的最优权值变化进行更新。信息素 $\tau_{ij}(t)$ 将会被限制在最大最小信息素范围之间, 使该方法不会在短时间内快速收敛陷入局部最优。

2.2 粒子群优化算法

粒子群优化 (PSO, particle swarm optimization) 算法^[6]模拟鸟群或鱼群的觅食方法, 不断地更新当前位置及下一时刻的方向及速度, 使其不断逼近目标。在 PSO 算法中, 每个粒子通过不断的自我学习与向迭代最优粒子学习来改进自身的速度及位置, 从而达到一个最优结果。

MMAS 算法相对于传统的蚁群算法来说, 极大地避免了局部最优情况的发生, 并且增加了该模型找到最优值的概率, 但是在节点数量非常多的情况下, MMAS 仍然容易陷入局部最优, 它的最大最小值的限制在避免算法快速陷入局部最优的同时, 也降低了其寻找到最优路径的可能性, 即忽略了信息素中的优越因子。为了增强算法寻找最优路径的能力, 本文结合 PSO 算法和 MMAS 算法, 对信息素 $\tau_{ij}(t)$ 进行局部及全局更新, 突出了对优越路

段的标记, 实现了对多点路径规划问题的有效处理。

3 回溯蚁群-粒子群混合算法

本文提出运用回溯蚁群-粒子群混合算法 (BAPM) 解决旅游景区中的多点路径规划问题。该算法结合了 MMAS 算法和 PSO 算法, 增强了路径寻优能力, 避免了算法过早陷入局部最优, 并改善了算法寻优能力弱的问题, 在对比实验部分, 也达到了较为理想的测试结果。该算法如算法 1 所示。

算法 1 回溯蚁群-粒子群混合算法

- 1) 使用 Floyd-Warshall 算法将图 G 转换成 \tilde{G} , 更新权值, 用 city 表示 \tilde{G} 中节点个数
- 2) 建立 \tilde{G} 的权值矩阵 \tilde{c} , 如果两点之间没有路径, 则 $\tilde{c}_{ij} = -1$
- 3) 设置迭代次数 num 和蚂蚁数量 m
- 4) 初始化信息素 τ 、启发因子 η 、信息素和启发因子的相关重要性参数 α 与 β 、学习因子中的可调参数 ω 、衰减因子 ρ 、最小信息素可调参数 p_{best} 、迭代代数 $i=1$ 、全局最优路径 $path_{gb}$ 、局部最优路径 $path_{lb}$ 及蚂蚁所处当前节点 startpoint
- 5) while $i \leq \text{num}$
- 6) 初始化 path 记录每只蚂蚁走过的路径
- 7) for $j=1:m$
- 8) startpoint = s
- 9) 设置 $\text{city} \times \text{city}$ 的矩阵 **tabu_arc**, 如果 $\tilde{c}_{ij} = -1$, 那么 $\text{tabu_arc}(i, j) = -1$, 设置邻接表 **tabu_node** 用于存放蚂蚁当前到达点的可行路段, 设置不定长数组 **node** 记录 $path_j$ 中经过的所有不重复节点
- 10) while $\text{count}(\text{node}) < \text{count}(\tilde{V})$
- 11) if $\text{count}(\text{tabu_node}(\text{startpoint})) > 0$
- 12) 计算 $\text{tabu_node}(\text{startpoint})$ 中的所有路段的被选择概率
- 13) 运用轮盘赌的概率选择方法选择下一步路段 $\tilde{A}(\text{startpoint}, b)$, $path_j = (path_j, \tilde{A}(\text{startpoint}, b))$, $\text{tabu_arc}(\text{startpoint}, b) = -1$, 将 $\text{tabu_node}(\text{startpoint})$ 中的路段 $\tilde{A}(\text{startpoint}, b)$ 标记为已用路段
- 14) 更新变量, $\text{startpoint} = b$ 、if $b \notin \text{node}$, $\text{node} = \{\text{node}, b\}$
- 15) else
- 16) 回溯到 **tabu_arc** 的上个状态, 删除 $path_j$

中的最后一条路段,更新 startpoint、node,并在 tabu_node 中对当前不可行路段进行标记

- 17) end if
- 18) end while
- 19) 对局部最优路径 $path_{lb}^i$ 进行更新
- 20) 用改进 PSO 算法局部更新信息素 τ
- 21) end for
- 22) 对全局最优路径 $path_{gb}$ 进行更新
- 23) 对信息素 τ 进行全局更新
- 24) $i = i + 1$
- 25) end while

本文所提 BAPM 算法中关键的几个部分为:原始图 G 转换成完全图 \tilde{G} ; 寻找可行路径,包括路径的编码、计算路径的被选择概率;信息素的局部更新和全局更新。下面分别对这几个部分进行详细说明。

3.1 图模型转换

从景区地图中抽象获得的图 G 是一个非完全有向图,它存在以下 2 种特殊情况: 1) 2 个节点之间没有一条直接连接的路径, 2) 直接连接路径的权值比间接路径的权值大。因此在处理图 G 之前,本文重新构建图模型,以便更好地计算出多点最短路径。

假设 N 为非完全连通图 $G=(V,A)$ 中所有节点的个数,其中有 $n(n \leq N-1)$ 个节点为必经节点,从节点 s 出发,寻找一条经过这 n 个必经节点一次且仅一次,再返回起始节点 s 的最短路径。将起始点 s 看作固定点, $s \in V$, 必经点集为 $M \subseteq V - \{s\}$ 。

本文通过以下步骤实现问题的转换。

步骤 1 生成新的顶点集合 $\tilde{V}=M \cup \{s\}$, 显然 \tilde{V} 是 V 的子集。

步骤 2 生成新的边集合 $\tilde{A}=\{(i,j)|i,j \in \tilde{V}, \text{且 } i,j \text{ 在 } G \text{ 中存在路径}\}$, 显然,由于 G 是一个连通图, G 中任意两点间都有路径,则 $\tilde{G}=(\tilde{V},\tilde{A})$ 是一个完全图。

步骤 3 计算图 \tilde{G} 中每条边的长度,令 \tilde{V} 中的任意两点 i,j 之间的距离 \tilde{c}_{ij} 为图 G 中 i,j 之间最短路径的长度,由于 G 是连通图,图中任意两点间都有路径,即该长度不会等于无穷大。在图 G 中求 i,j 两点之间的最短距离使用了 Floyd-Warshall^[7]算法。

很明显,通过以上 3 个步骤建立了一个完全图 \tilde{G} ,其点集合只包含 G 中的起始点 s 和必经点集合

M , 边的长度则对应边的 2 个端点在 G 中的最短路径长度,则原问题转换为在完全图 \tilde{G} 中找寻一个最短 Hamilton 圈的 TSP 问题。

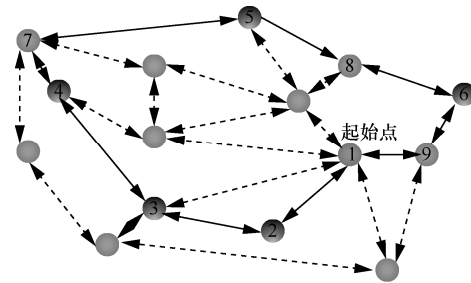


图 1 多点路径规划原问题

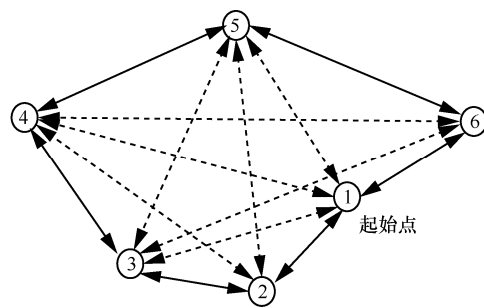


图 2 多点路径规划转换后

图 1 所示为非完全连通图。需要从点 1 出发,前往点 2、点 3、点 4、点 5、点 6 进行游览,之后回归点 1,起始点 $s=1$, $M = \{2,3,4,5,6\}$ 。将图 1 中必经点提取出来构成图 2,运用 Floyd 算法计算集合 M 中每两点之间最短路径长度,将其作为图 2 中的边长。

斯坦纳旅行商问题^[8] (STSP, Steiner traveling salesman problem) 与本文所述问题相似,都是解决经过必经点的路径规划问题,并且必经点可以重复使用。不同之处在于 STSP 有固定的起始点和终点,而本文所述问题仅需要固定起始点。由于必经点可以重复使用,为了让该问题在定义上更加准确,引入引理 1^[8]。

引理 1 在 STSP 最佳解决方案中,任何一条有向边仅能遍历一次。

也就是说,在图 \tilde{G} 的最优路径中,点集 \tilde{V} 中的节点可以重复使用,但边集 \tilde{A} 中的有向边不能重复使用。本文基于此规则实现了 BAPM 算法对最小权值路径的搜索。

3.2 寻找可行路径

3.2.1 路径编码

将景区地图按照 3.1 节步骤 1~步骤 3 进行转换后,问题转变为从转换后的图中给定的起始点 s 出

发，需要找到一条经过所有必经点的可行路径。

由 3.1 节可知， \tilde{A} 为转换图 \tilde{G} 的有向边集合。从节点 i 出发到达节点 j 的有向边记作 $\tilde{A}(i, j)$ 。本文用图中的边构成的位串对路径进行编码，将蚂蚁 a 寻找到的路径 $path_a$ 表示为

$$path_a = (\tilde{A}(i_1, i_2), \tilde{A}(i_2, i_3) \cdots \tilde{A}(i_{k-1}, i_k))$$

其中， $i_1, i_2, \dots, i_k \in \tilde{V}$ ， $path = (path_a, path_b, \dots)$ 记录了所有蚂蚁的寻找路径。

将路径 $path_a$ 中包含的所有不重复节点的集合记作 $node$ ；使用禁忌表 $tabu_arc$ 对有向边集合 \tilde{A} 进行管理，保证在 $path_a$ 中的有向边不会重复出现；使用邻接表 $tabu_node$ 来记录路径 $path_a$ 当前到达节点的可选路段； $startpoint$ 记录路径当前到达的节点，当路径 $path_a$ 经过 $\tilde{A}(i, j)$ 时，可以说当前路径 $path_a$ 到达了节点 j ，此时 $startpoint = j$ 。

3.2.2 可行路径的寻找与路段的选择

根据图 2，初始化的权值矩阵 \tilde{c} 和禁忌表 $tabu_arc$ 为

$$\tilde{c} = \begin{bmatrix} -1 & 10 & 50 & 20 & 30 \\ -1 & -1 & 40 & 10 & 20 \\ -1 & -1 & -1 & 5 & 15 \\ -1 & -1 & 5 & -1 & 10 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

$$tabu_arc = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 \\ -1 & -1 & -1 & 0 & 0 \\ -1 & -1 & 0 & -1 & 0 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

其中， \tilde{c} 存放路段的权值， $tabu_arc$ 表示路径段的使用状态，“0”表示该路径可行，“-1”表示该路径不可行。

寻找一条可行路径的方法如下。蚂蚁 a 从起始点 $startpoint = s$ 开始寻找路径，根据 $tabu_node(startpoint)$ 得到下一时刻的所有可选路段，选择其中一条路段 $\tilde{A}(s, j)$ ，在 $tabu_arc$ 中将已走过的路段标记为“-1”，同时，将路段 $\tilde{A}(s, j)$ 存入 $path_a$ 中，并且 $startpoint = j$ 。为了减少算法的时间复杂度， $tabu_node$ 使用散列表进行存储。当执行到某个没有可行边的节点，即可行路段 $tabu_node(startpoint)$ 为空或者均已被标记时，证明该路

径不可行，则使用回溯方法，最终找到一条可行路径。

上述寻找可行路径的过程可简化表示为图 3。

path: {}	tabu_node:	(1,2)(1,3)(1,4)(1,5)
path: {(1,3)}	tabu_node:	(3,4)(3,5)
path: {(1,3)(3,5)}	tabu_node:	
path: {(1,3)(3,4)}	tabu_node:	(4,5)
path: {(1,2)}	tabu_node:	(2,3)(2,4)(2,5)
path: {(1,2)(2,3)}	tabu_node:	(3,4)(3,5)
path: {(1,2)(2,3)(3,4)}	tabu_node:	(4,5)

图 3 寻找可行路径的迭代过程

在非完全有向图中，运用回溯方法查询路径可以避免产生非可行性路径，提高选择路径的效率。

$Tabu_node$ 表中，确定可选路段是该算法中的一个重要环节。在第 k 次迭代中，蚂蚁 a 在寻找路径时，会根据 $startpoint$ 对应的 $tabu_node(startpoint)$ 中每条路段的可行概率 $p_{ij}^a(k), i \in \tilde{V}, j \in \tilde{V}, i \neq j$ ，选择概率大的路段，并根据已选择的路段到达下个节点 d ， $startpoint = d$ 。若 $tabu_node$ 不存在可行路段，即 $count(tabu_node(startpoint))=0$ ，则删除 $path_a$ 中 $startpoint$ 所对应路段，将蚂蚁到达的当前节点赋值给 $startpoint$ ，释放 $tabu_arc$ 中该路段的标记，继续寻找下一个可行路段，以此类推，直到所有必经点都包含在该蚂蚁的所行路径当中，即 $node = \tilde{V}$ ，此路径 $path_a$ 则为一条可行路径。这种寻路方式适应本文模型要求，使寻找路径更加灵活并且保证了每条路径的可行性，避免了非可行解的产生。

选择路段的概率计算式如式(1)所示。

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)][\eta_{ij}]}{\sum_{l \in tabu_node^*(i)} [\tau_{il}(t)]^\alpha [\eta_{il}]^\beta} \quad (1)$$

其中， $p_{ij}^k(t)$ 表示在第 t 次迭代中，蚂蚁 k 选择有向边 $\tilde{A}(i, j)$ 的概率； α 和 β 分别描述了信息素 $\tau_{ij}(t)$ 和启发因子 η_{ij} 的相关重要性； $tabu_node^*(i)$ 表示蚂蚁 k 从节点 i 出发可以到达的可行节点集合。

3.3 更新信息素

3.3.1 信息素的全局更新

根据上述寻路方法，可以得到所有蚂蚁的可行路径 $path$ ，找出其中权值最小的路径，记作 $path_{gb}$ ，并且记录每只蚂蚁的可行路径，记作 $path_{ib}$ 。在迭代过程中，找出权值最小的路径并且将其权值与 $path_{gb}$ 的权值进行比较，将 $path_{gb}$ 替换成权值最小

的路径, 本文将 path_{gb} 称作全局最优路径, 同理, 将 path_{lb} 替换成每只蚂蚁寻找到的权值最小的路径, 称作局部最优路径。

当所有蚂蚁均寻找到一条可行路径后, 需要更新图中每个路段对应的信息素。信息素 $\tau_{ij}(t)$ 描述了蚂蚁经过各个路段所留下的痕迹, 本文将 $\tau_{ij}(t)$ 值较大的路段称为优越路段, 信息素更新如式(2)所示。

$$\tau_{ij}(t+1) = \rho\tau_{ij}(t) + \Delta\tau_{ij}^{\text{best}} \quad (2)$$

其中, 参数 $\rho(0 \leq \rho \leq 1)$ 为衰减因子, 用于冲淡信息素的浓度; $\Delta\tau_{ij}^{\text{best}}(t)$ 为全局最优路径 path_{gb} 的信息素增量, 计算式如式(3)所示。

$$\Delta\tau_{ij}^{\text{best}}(t) = \begin{cases} \frac{1}{f(s^{\text{best}})}, & \tilde{A}(i, j) \text{ 在最优路径中} \\ 0, & \text{其他} \end{cases} \quad (3)$$

其中, $f(s^{\text{best}})$ 表示 path_{gb} 路径权值。

为了避免信息素过快地陷入局部最优, 设置最大信息素 τ_{max} 和最小信息素 τ_{min} , 如式(4)和式(5)所示, 则信息素更新范围为 $\tau_{\text{min}} \leq \tau_{ij}(t) \leq \tau_{\text{max}}$ 。

$$\tau_{\text{max}}(t) = \frac{1}{1 - \rho} \frac{1}{f(s^{\text{best}})} \quad (4)$$

如果 $\tau_{ij}(t) \geq \tau_{\text{max}}(t)$, 则 $\tau_{ij}(t) = \tau_{\text{max}}(t)$ 。

$$\tau_{\text{min}} = \frac{\tau_{\text{max}}(1 - \sqrt[\text{city}]{p_{\text{best}}})}{(\text{avg} - 1)\sqrt[\text{city}]{p_{\text{best}}}} \quad (5)$$

其中, city 表示点集 \tilde{V} 的大小; p_{best} 表示一个可调参数, 本文取值为 0.005。

3.3.2 信息素的局部更新

设置最大最小信息素后, 随着迭代次数的增多, 优越路段得不到有效的突出, 使算法容易陷入局部最优, 造成最终得到的最短路径与期望路径存在较大差距。本文结合改进后的 PSO 算法, 突出路径中较优路段的信息素, 以保证算法能产生更为接近最短路径的结果。

基本的 PSO 算法通常用于连续函数寻找最优解的问题, 粒子从随机的初始位置出发, 不断改变自身的速度和位置, 每次迭代均飞向自身最优和全局最优位置的加权中心, 以达到自我认知和信息共享的目的。本文建立了所讨论问题和 PSO 算法之间的映射关系, 将问题空间映射到粒子空间, 将问题

离散化。

将每只蚂蚁 a 的可行路径 path_a 看作一个粒子, 将当前的信息素 τ 看作粒子的当前位置, 将信息素的增量看作 PSO 算法中的速度变化量。

本文定义了 2 个变量——全局最优路径片段 p_{gb} 和局部最优路径片段 p_{lb} , 用于计算粒子的运动速度。全局最优路径片段为蚂蚁当前找到的路径与全局最优路径中相同的路径片段集合, 局部最优路径片段为蚂蚁当前找到的路径与其自身曾找到的最优路径中相同的路径片段集合。若蚂蚁 a 找到当前路径为 path_a , 其计算式为

$$\text{path}_a = (p_1, \dots, p_n), p_i \in \tilde{A}, 1 \leq i \leq n$$

全局最优路径为

$$\text{path}_{\text{gb}} = (p_1, \dots, p_m), p_i \in \tilde{A}, 1 \leq i \leq m$$

局部最优路径为

$$\text{path}_{\text{lb}} = (p_1, \dots, p_k), p_i \in \tilde{A}, 1 \leq i \leq k$$

则全局最优路径片段为

$$p_{\text{gb}} = \{p_i \mid p_i \text{ 为边构成的位串, 并且 } p_i \text{ 同时存在于 } \text{path}_a \text{ 和 } \text{path}_{\text{gb}} \text{ 中}\}$$

局部最优路径片段为

$$p_{\text{lb}} = \{p_i \mid p_i \text{ 为边构成的位串, 并且 } p_i \text{ 同时存在于 } \text{path}_a \text{ 和 } \text{path}_{\text{lb}} \text{ 中}\}$$

假设蚂蚁 a 在本次迭代中寻找到的可行的路径为 $\text{path}_a = \{\tilde{A}(1,3), \tilde{A}(3,2), \tilde{A}(2,5), \tilde{A}(5,4), \tilde{A}(4,9), \tilde{A}(9,10), \tilde{A}(10,6), \tilde{A}(6,7), \tilde{A}(7,8)\}$, 而该蚂蚁在迭代过程中寻找到的权值最短路径为 $\text{path}_{\text{lb}}^a = \{\tilde{A}(1,3), \tilde{A}(3,6), \tilde{A}(6,7), \tilde{A}(7,8), \tilde{A}(8,2), \tilde{A}(2,10), \tilde{A}(10,5), \tilde{A}(5,9), \tilde{A}(9,4)\}$, 将这只蚂蚁的当前可行路径与 path_{lb} 作对比, 找到 2 条路径之间的相同的路段 $\{\tilde{A}(1,3)\}$ 和 $\{\tilde{A}(6,7), \tilde{A}(7,8)\}$, $p_{\text{lb}} = \{\tilde{A}(1,3), \tilde{A}(6,7), \tilde{A}(7,8)\}$, 同理, 将当前可行路径与 path_{gb} 作对比, 可以得到全局最优路径片段 p_{gb} 。

在蚂蚁 a 寻找到可行路径 path_a 后, 得到 p_{gb} 和 p_{lb} , 增加 p_{gb} 、 p_{lb} 对应路段的信息素, 信息素增长式如式(6)~式(9)所示。

$$v = c_1 \text{rand}() p_{\text{lb}}^* + c_2 \text{rand}() p_{\text{gb}}^* \quad (6)$$

$$p_{lb}^*(i, j) = \begin{cases} 1, & \text{如果与当前路径存在交集} \\ 0, & \text{其他} \end{cases} \quad (7)$$

$$p_{gb}^*(i, j) = \begin{cases} 1, & \text{如果与当前路径存在交集} \\ 0, & \text{其他} \end{cases} \quad (8)$$

$$\tau = \tau + \nu \quad (9)$$

其中, p_{lb}^* 表示局部最优路径片段的 0-1 矩阵, p_{gb}^* 表示全局最优路径片段的 0-1 矩阵, 学习因子 $c_1=c_2 = \frac{\omega}{city}$, $rand()$ 表示[0, 1]之间的随机数。

在迭代过程中, 不断地更新 $path_{lb}$ 和 $path_{gb}$, 从而更新 p_{lb} 和 p_{gb} 对应的信息素, 使信息素不断地向有益于获取最优路径的方向增长, 从而使算法能够更加快速有效地得到图中的最短路径。

4 实验结果与分析

本文的实验主要分为 4 个部分。第一部分将本文所提 BAPM 算法与 MMAS 算法、遗传算法(GA, genetic algorithm) 进行比较, 展示 BAPM 算法在精确度上的优越性, 本文选择了 TSPLIB 数据库中的数据进行实验。第二部分为仿真实验, 运用 BAPM 算法寻找包含多点的最短路径, 然后运用分支定界法^[9-10]找出这个图中的精确最短路径, 验证 BAPM

算法寻找最短路径的正确性。第三部分为景区地图实际规划结果, 展示了 BAPM 算法在一个实际的景区地图数据上的实验结果。第四部分对上述实验进行总结。

4.1 TSPLIS 数据库实验

为了验证 BAPM 算法有效地改进了局部最优问题并且能找到更为接近的最短路径, 本文使用 MMAS 算法和 GA 算法作为对比算法, 3 种算法均在 MATLAB(R2014a)上进行实现, 且迭代次数相同。BAPM 算法的参数设置如表 1 所示。

表 1 BAPM 算法参数

名称	参数值
蚁群大小	$m = city$
起始信息素	$\tau_0 = \tau_{min}$
信息素相关参数	$\alpha = 1$
启发因子相关参数	$\beta = 5$
信息素削减因子	$\rho = 0.98$
用于 τ_{min} 的参数	$p_{best} = 0.005$
参数 ω	$\omega = 0.0001$

分别运用 BAPM 算法、MMAS 算法和 GA 算法对 TSPLIS 数据库中提供的 14 组数据进行实验, 每种算法各运行 10 次, 获取每组数据在不同算法中运行得到的最大路径长度、最小路径长度以及 10 组数据值与精确解之间的平均偏差百分比, 如表 2 所示。

表 2 TSPLIB 数据库对比实验

数据	各算法间比较								
	BAPM 算法			MMAS 算法			GA 算法		
	平均偏差	max	min	平均偏差	max	min	平均偏差	max	min
ft17	0.0%	39	39	0.0%	39	39	0.0%	39	39
ft53	1.9%	7 344	6 905	3.2%	7 344	6 905	4.1%	7 372	6 095
ft70	1.0%	39 102	38 673	2.9%	40 109	38 673	3.3%	40 109	38 673
ftv33	0.0%	1 286	1 286	0.0%	1 286	1 286	0.0%	1 286	1 286
ftv35	0.0%	1 473	1 473	0.02%	1 477	1 473	0.0%	1 473	1 473
ftv38	0.0%	1 530	1 530	0.05%	1 549	1 530	0.0%	1 530	1 530
ftv44	0.0%	1 613	1 613	0.01%	1 615	1 613	0.0%	1 613	1 613
ftv47	0.4%	1 791	1 776	0.8%	1 791	1 791	0.4%	1 791	1 776
ftv55	0.1%	1 613	1 608	1.0%	1 643	1 608	0.7%	1 631	1 608
ftv64	0.2%	1 844	1 839	1.6%	1 893	1 844	0.2%	1 844	1 844
ftv70	1.9%	2 078	1 950	7.5%	2 114	1 950	6.2%	2 078	2 078
kro124	3.4%	37 693	37 287	3.5%	37 739	37 287	4.2%	37 804	37 739
p43	0.0%	5 620	5 620	0.1%	5 647	5 620	0.2%	5 647	5 620
ry48p	0.4%	14 507	14 459	1.1%	14 603	14 570	0.8%	14 603	14 459

4.2 仿真实验

为验证 BAPM 算法的正确性, 本文建立了一个仿真系统。该系统可随机构建不同的图。这些图都可以看作景区地图的抽象图结构, 图中的顶点对应景区的景点, 边则表示景点之间的路径。假设必经节点数量为 9, 寻找图中从某一起始点出发经过这些点的最短路径, 就对应为从景区的当前位置出发, 规划一条包含 9 个游览景点的最短路径问题。第一种情况, 随机构建一个包括 100 个随机生成的节点的图, 任意两节点之间均有路径。在产生的路径中随机删除 4 000 条路径, 将其转换为一个稀疏图, 从而模拟景区路线网络的布局, 然后在图中选择 10 个必经点 (包含起点), 如图 4 所示。第二种情况, 随机产生 10 个点及其之间的路径, 构成一个有向非完全图, 随机选择一个起始点, 寻找经过这 10 个点的最短路径, 如图 5 所示。

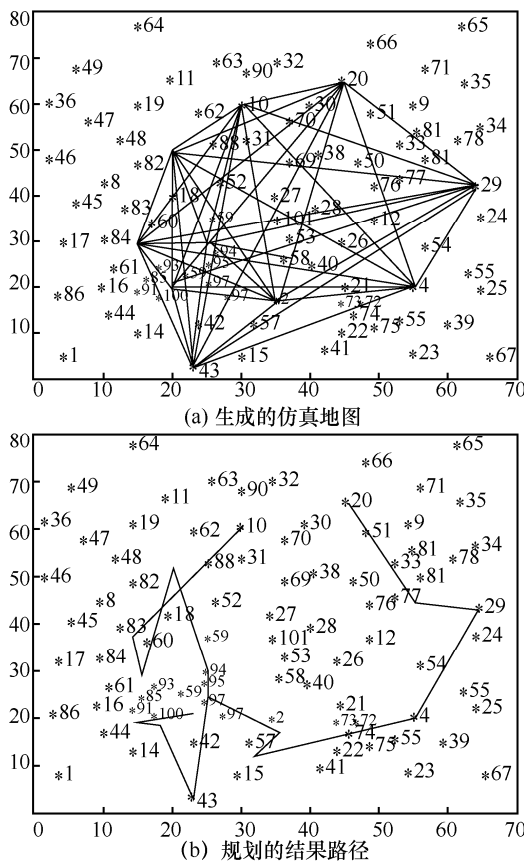


图 4 第一种情况

图 4(a)展示了通过 Floyd-Warshall 算法变换后的路线图, 由选定的这 10 个点构成的图, 图 4(b)是运用 BAPM 算法经过这 10 个点寻找的最短路径。

图 5(a)为由 Floyd-Warshall 算法进行重组的有向图, 图 5(b)表示由 BAPM 算法产生的最短路径。

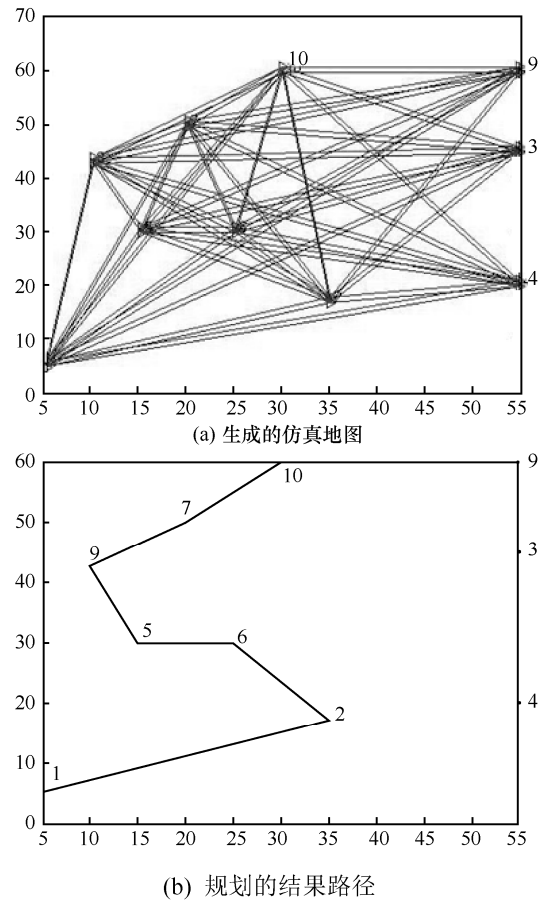


图 5 第二种情况

以上 2 种情况下本文方法得到的结果与分支定界法找到的路径相同。

4.3 橘洲景区游览路线规划

橘洲景区是长沙的著名旅游景区, 本文算法被应用于橘洲景区智能导览系统中。该系统中的橘洲地图是借助 GPS 设备、GOOGLE 卫星地图等, 通过实地打点, 遍历景区道路并测量记录后自行建立的电子地图, 橘洲景区景点的位置信息如表 3 所示。

以表 3 中标注*的 6 个景点为实验 1 必经景点, 采用 BAPM 算法规划出的最佳路径为: 枕江观光车售票处—沁园春诗词碑—毛泽东青年艺术雕像—望江亭—指点江山石碑—问天台, 全程 645 m, 为最短游览路径。

以表 3 中标注#的 10 个景点为实验 2 必经景点, 采用 BAPM 算法规划出的最佳路径为: 地铁站西站口—原长沙海关旧址—婚庆园—橘子洲古董钢琴博物馆—沁园春·长沙橘洲 1925 —橘子洲

人行入口—美孚洋行旧址—财神古殿—游客服务中心西站点—橘子洲西大门，全程 1 218 m，为最短游览路径。

表 3 橘子洲景区景点信息

景点序号	景点名称	经度	纬度
1	问天台*	112.966 631°	28.171 271°
2	指点江山石碑*	112.966 793°	28.171 637°
3	望江亭 *	112.966 811°	28.171 796°
4	毛泽东青年艺术雕像*	112.966 802°	28.173 412°
5	沁园春诗词碑*	112.968 028°	28.187 983°
6	长株潭两型社会展览馆	112.966 938°	28.178 382°
7	竹园	112.967 779°	28.179 878°
8	朱张渡	112.966 456°	28.180 764°
9	唐生智公馆旧址站点	112.966 919°	28.183 873°
10	潇湘名人旧址	112.967 835°	28.183 717°
11	江神庙	112.969 802°	28.204 733°
12	桃园	112.967 844°	28.184 935°
13	桂园	112.968 208°	28.186 169°
14	桂园诗词碑	112.968 033°	28.187 964°
15	九州龙骧码	112.967 482°	28.189 632°
16	百米高喷	112.967 913°	28.190 228°
17	婚庆园#	112.968 614°	28.199 179°
18	美孚洋行旧址#	112.968 347°	28.193 838°
19	湘江亭观光车售票处*#	112.968 865°	28.196 559°
20	财神古殿#	112.967 534°	28.196 197°
22	橘子洲古董钢琴博物馆#	112.968 69°	28.196 961°
23	橘子洲入口	112.969 148°	28.197 757°
24	橘子洲人行入口#	112.968 542°	28.197 793°
25	橘子洲西大门#	112.967 535°	28.198 004°
26	原长沙海关旧址#	112.968 614°	28.199 179°
27	地铁站西站口#	112.967 998°	28.200 396°

注：*表示实验 1 必经景点，#表示实验 2 必经景点。

实验 1 与实验 2 均找到了包含必经景点的最短游览路径。

4.4 实验总结

本文使用了 3 组实验，验证了提出的 BAPM 算法的有效性及其优越性。在第一组实验中，选用

了 TSPLIB 数据库中的 TSP 数据，将本文提出的 BAPM 算法与 MMAS 算法及 GA 算法进行了比较，结果显示 BAPM 算法可以在数据量较大的条件下，找出更接近最小路径的结果。在第二组实验中，随机生成点和路径，模拟景区地图和景点，该组实验的必经点数设置较小（10 个必经点），验证了在点数较少的情况下，BAPM 算法可以有效地找到准确的最短路径。第三组实验对实际的景区景点进行规划，所有景点位置和景点间路径长度均为实测数据，验证了算法在景点路径规划中的有效性。

通过实验验证了 BAPM 算法的有效性。实验结果表明，BAPM 算法可以处理非完全图问题和完全图问题，并且在必经点个数较小的情况下，可以准确并且快速地寻找到最短路径。

5 结束语

本文针对旅游景区场景下，旅游者对多景点游览路径的规划需求，提出了一种基于回溯蚁群-粒子群混合算法的旅游景点路径规划模型，用于解决从起始点出发，经过若干必经点的景区景点路线规划的问题。通过实验，证明了 BAPM 算法可解决上述多点景点路径规划问题。由于 BAPM 算法实际上是一种多点路径规划算法，在复杂网络分析与设计、机器人路径规划^[11]、路由复杂网络^[12]等方面同样具有一定的实用价值。

参考文献：

- [1] 朱庆,王焯萍,张骏骁,等. 综合导航网格模型及其在智慧旅游寻径中的应用[J]. 西南交通大学学报, 2017, 52(1):195-201.
ZHU Q, WANG Y P, ZHANG J X, et al. Integrated navigation grid model and its applications in smart tourism routing[J] Journal of Southwest Jiaotong University, 2017, 52(1):195-201.
- [2] 吕琼艺. 基于改进的 Dijkstra 算法的旅游规划线路研究与实践 [J]. 柳州职业技术学院学报,2017,17(2):32-36.
LV Q Y. Research and practice of tourism planning line based on improved Dijkstra algorithm[J]. Journal of Liuzhou Vocational & Technical College, 2017,17(2):32-36.
- [3] 胡军国, 祁亨年, 董峰, 等. 一种改进蚁群算法研究和旅游景区路径规划问题求解[J]. 计算机应用研究, 2011, 28(5):1647-1650.
HU J G, QI H N, DONG F, et al. Improved ant colony algorithm for path planning of tourist scenic are[J]. Application Research of Computers, 2011, 28(5):1647-1650.
- [4] 蒋仲安,王明,陈雅. 基于地理坐标和轨迹数据的路径推荐方法[J].

通信学报,2017,38(5):165-171.

JIANG Z A, WANG M, CHEN Y. Path recommendation based on geographic coordinates and trajectory data[J]. Journal on Communications,2017,38(5):165-171.

- [5] STÜTZLE T, HOOS H H. Max-min ant system[J]. Future Generation Computer Systems, 2000, 16(08): 889-914
- [6] EBERHART R C, KENNEDY J. A new optimizer using particle swarm theory[C]// The 6th International Symposium on Micro Machine And Human Science, 1995: 39-43.
- [7] FLOYD R W. Algorithm 97: shortest path[J]. Communications of the ACM, 1962, 5(6): 345.
- [8] LETCHFORD A N, NASIRI S D, THEIS D O. Compact formulations of the Steiner traveling salesman problem and related problems[J]. European Journal of Operational Research, 2013, 228(01): 83-92
- [9] JEPSEN M K, PETERSEN B, SPOORENDONK S. A branch-and-cut algorithm for the capacitated profitable tour problem[J]. Discrete Optimization, 2014, 14(C): 78-96.
- [10] ROSTAMI B, MALUCELLI F, BELOTTI P. Lower bounding procedure for the asymmetric quadratic traveling salesman problem[J]. European Journal of Operational Research, 2016, 253(03): 584-592
- [11] 普兴成, 李俊杰, 吴慧超, 等. 基于改进粒子群算法的移动机器人多目标点路径规划[J]. 智能系统学报, 2017, 12(3):301-309.
PU X C, LI J J, WU H C, et al. Mobile robot multi-goal path planning using improved particle swarm optimization[J]. CAAI Transactions on Intelligent Systems, 2017, 12(3):301-309.
- [12] 郑继明, 杨坤, 柳慧鹏, 等. 基于 0-1 线性规划的多点路由规划模型研究[J]. 通信技术, 2017, 50(7):1443-1446.
ZHENG J M, YANG K, LIU H P, et al. Multicast routing model based on 0-1 linear programming[J]. Communications Technology, 2017, 50(07): 1443-1446.

[作者简介]



刘丽珏(1973-), 女, 湖南长沙人, 博士, 中南大学副教授, 主要研究方向为智能计算、人工智能、机器学习。



罗舒宁(1993-), 女, 山东泰安人, 中南大学硕士生, 主要研究方向为多点路径规划。



高琰(1973-), 女, 湖南长沙人, 博士, 中南大学副教授, 主要研究方向为智能信息处理。



陈美妃(1998-), 女, 广西壮族自治区, 主要研究方向为大数据、云计算。