

基于四叉树和改进蚁群算法的全局路径规划

赵百轶, 张利军, 贾鹤鸣

(哈尔滨工程大学 自动化学院 黑龙江 哈尔滨 150001)

摘 要: 为解决机器人在大范围二维平面区域内的路径规划问题, 提出一种四叉树和改进蚁群算法相结合的路径规划方法. 基于四叉树分解法, 对路径规划的二维区域进行环境建模. 在环境建模的基础上, 采用改进蚁群算法进行高效的路径规划. 四叉树在完整地记录环境信息的同时对环境信息进行了高效地压缩. 改进蚁群算法可以规划出与障碍物保持一定安全距离的路径, 提高了规划出的路径的实用性. 仿真实验表明, 提出的路径规划方法在执行效率和路径的实用性上取得了良好的平衡, 可以高效地对大区域进行路径规划.

关键词: 移动机器人; 全局路径规划; 蚁群算法; 四叉树

中图分类号: TP24

文献标志码: A

文章编号: 1009-671X(2011)10-0023-06

Global path planning based on quadtree and improved ant colony optimization algorithm

ZHAO Baiyi, ZHANG Lijun, JIA Heming

(College of Automation, Harbin Engineering University, Harbin 150001, China)

Abstract: In order to solve the path planning problem for a robot in a large two-dimensional plane, a new method based on the quadtree and improved ant colony algorithm is presented for path planning. A two-dimensional area model is built by the quadtree. An improved ant colony algorithm is used for high-efficiency path planning based on this model. The quadtree not only records all of the area information but also compresses area information efficiently. The improved ant colony algorithm can find a path that maintains a safe distance from obstacles, which improves the usefulness of the path. The results of simulation experiments show that the new method gets a good balance between efficiency and usefulness of the path and can find a path efficiently in the large area.

Keywords: mobile robot; global path planning; ant colony optimization algorithm; quadtree

全局路径规划是指在一个给定的区域内, 已知障碍物分布信息的基础上, 搜索一条从起点到终点的路径, 使得移动机器人与障碍物无碰撞地从起点运动到终点^[1].

目前, 国内外已对机器人路径规划问题进行了大量的研究^[2-6]. 传统算法和智能算法被应用于路径规划, 其中智能算法特别适用于解决路径规划等组合优化问题, 因此, 智能算法是路径规划问题的研究热点. 蚁群算法是智能算法的一种, 所以其被广泛地研究. 文献[2]采用栅格法对场景进行建模, 利用基本的蚁群算法实现路径规划. 文献[3]基于栅格法对环境建模, 将粒子群算法和蚁群算法相融合, 先

用粒子群算法初始化信息素, 再利用蚁群算法进行路径搜索. 文献[4]建立了以元胞自动机为系统框架和以蚁群算法为运行核心的离散模型, 适用于求解机器人路径.

上述文献多集中于对基本的蚁群算法改进, 达到提高算法执行效率及缩短规划出的路径的长度的目的, 但存在以下问题: 1) 算法不能高效地对大范围区域进行路径规划; 2) 规划出的路径不能保证与障碍物保持一定的安全距离, 路径的安全性及实用性不强. 文中采用改进的蚁群算法, 利用机器人与障碍物的距离及与终点的距离构造综合启发信息, 并引入惩罚因子概念, 对蚂蚁寻找出的路径施加惩罚

收稿日期: 2011-07-19.

基金项目: 教育部博士点基金资助项目(20102304110003).

作者简介: 赵百轶(1985-), 男, 硕士研究生, 主要研究方向: 先进控制理论. E-mail: zby41335@163.com.

因子,淘汰距离障碍物过近的路径,同时保留全局最优解,使得规划出的路径与障碍物保持安全距离的同时保证路径最短.为了提高算法在大范围区域内的搜索效率,利用四叉树对环境信息建模,高效地对环境信息进行压缩,降低路径规划问题的规模,提高算法搜索效率.

1 环境建模

四叉树种类很多,文中采用线性四叉树,其优点是只记录叶节点的编码,不记录中间节点的编码和层次关系,提高了存储空间的利用率^[7].

1.1 线性四叉树模型的建立

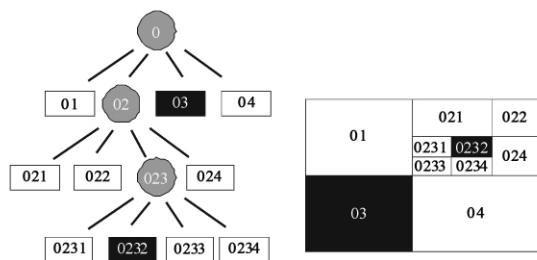
采用如下规则对二维区域进行非均匀分割来建立线性四叉树模型^[8].

1) 将整个区域作为根节点,编码为 0.

2) 将根节点区域沿 X 、 Y 方向分别进行二等分,根节点被均分成 4 个子节点,每个节点对应一块子区域;子节点按其方位,依据由西向东、由北向南的次序,分别将 1~4 追加到根节点的编码后形成各节点编码,即子节点的编码分别为 01~04. 子节点可以分为:障碍节点、自由节点和混合节点.障碍节点的空间内全为障碍物,自由节点的空间没有任何障碍物,既有障碍物又有自由空间的节点是混合节点.

3) 如果存在混合节点,对混合节点按规则 2) 方法继续分解,直到所有节点都为自由节点和障碍节点为止.

利用上述规则,图 1 展示了四叉树分解实例,其中白色区域代表自由节点,黑色区域代表障碍节点,灰色区域代表混合节点.



(a) 线性四叉树分解过程

(b) 最终的分解效果

图 1 四叉树分解实例

1.2 线性四叉树节点邻域的确定

生成的线性四叉树丢失了叶节点之间的连接关系,这种关系很难通过叶节点的编码用直观的方法确定,而其中自由叶节点间的关系在路径规划中又十分重要,因此,在路径规划前,应重新确立自由叶节点的连接关系,即确定每个自由叶节点的邻域.

本文只考虑节点边相邻的邻域,即水平方向和垂直方向的 4 个邻域,分别是北邻域、南邻域、西邻域和东邻域.边节点是 4 个邻域中至少有一个邻域不存在的节点;非边界点是 4 个邻域都存在的节点.相同尺寸邻域是指某一节点所求的邻域的尺寸与本节点的尺寸相同.反之,如果尺寸不相同则是不相同尺寸邻域^[9].

四叉树编码的基准体系可划分为如下 4 个集合:北集合 $N = \{1, 2\}$,南集合 $S = \{3, 4\}$,西集合 $W = \{1, 3\}$ 和东集合 $E = \{2, 4\}$.

对于某一节点 Q ,其编码为 $q_1 q_2 \dots q_n$,即 $Q = q_1 q_2 \dots q_n$.

1.2.1 非边界节点相同尺寸邻域的确定

如果 Q 是非边界节点,所求 4 个邻域按如下规则确定^[10]:

1) 如果 $q_n = 1$,东邻域为 $Q + 1$ (编码最后一位 q_n 加 1) 到的编码为所求邻域的编码,南邻域的编码为 $Q + 2$;对于北邻域,从编码的末尾开始向左扫描,直到找到第 1 个不属于北集合的编码位 q_i ,将编码最右边的 $n - i$ 个编码位均加 2, q_i 减 2,其余编码位的值不变,得到的新编码为所求邻域的编码;与北邻域类似,对于西邻域,从编码的末尾开始向左扫描,直到找到第 1 个不属于西集合的编码位 q_i ,将编码最右边的 $n - i$ 个编码位均加 1, q_i 减 1,其余编码位的值不变,得到的新编码为所求邻域的编码.

2) 如果 $q_n = 2$,西邻域的编码为 $Q - 1$,南邻域的编码为 $Q + 2$,北邻域的确定方法与 $q_n = 1$ 时的相同;对于东邻域,从编码的末尾开始向左扫描,直到找到第 1 个不属于北集合的编码位 q_i ,将编码最右边的 $n - i$ 个编码位均减 1, q_i 加 1,其余编码位的值不变,得到的新编码为所求邻域的编码.

3) 如果 $q_n = 3$,北邻域的编码为 $Q - 2$,东邻域

的编码为 $Q+1$,西邻域的确定方法与 $q_n=1$ 时的相同;对于南邻域,从编码的末尾开始向左扫描,直到找到第 1 个不属于南集合的编码位 q_i ,将编码最右边的 $n-i$ 个编码位均减 2、 q_i 加 2,其余编码位的值不变,得到的新编码为所求邻域的编码。

4) 如果 $q_n=4$,西邻域的编码为 $Q-1$,北邻域的编码为 $Q-2$,东邻域的确定方法与 $q_n=2$ 时的相同,南邻域的确定方法与 $q_n=3$ 时的相同。

1.2.2 非边界节点不相同尺寸邻域的确定

非边界节点不相同尺寸邻域的确定是以相同尺寸邻域确定为基础的,假设不相同尺寸邻域存在,鉴于节点邻域关系的对称性,不相同尺寸邻域按如下规则确定。

对于非边界节点 Q 采用 1.2.1 小节的方法得到相同尺寸邻域的编码 P ,寻找 P 的各级祖先的编码,即截短编码 P 。具体的规则为:移除编码 P 的最后一位得到新编码,用此编码遍历整个自由叶节点集合,查看是否存在此编码的叶节点,若存在,则找到了不相同尺寸邻域,否则用得到的新编码继续上述流程,直到编码长度变为 1 为止。

1.2.3 边界节点邻域的确定

对于节点 Q ,若其所有编码位的值都属于同一基准体系集合,则 Q 为边界节点,即对于 $q_i (i=1, 2, \dots, n)$,若 $q_i \in N$,则 Q 是边界节点,其没有北邻域,其他邻域的确定与 1.2.1 和 1.2.2 小节的方法相同;同理,对于 $q_i \in S, q_i \in W$ 和 $q_i \in E$ 的情况。

综上所述,可以得到某一自由叶节点的邻域,如图 1(b),叶节点 0233 应用规则可知,其北邻域为 0231、南邻域为 04、西邻域为 01、东邻域为 0234。

2 改进的蚁群算法

基本的蚁群算法由蚂蚁算法演化而来,其特点如下:1) 蚂蚁采用伪随机比例规则选择下一位置;2) 蚂蚁每运动一步都要进行局部信息素更新;3) 当所有蚂蚁从起点到达终点后,从蚂蚁搜索得到的路径中选择最优路径,并只对最优路径进行全局信息素更新^[10]。

基本的蚁群算法不能保证规划出安全的路径,

为了得到与障碍物保持一定安全距离且长度最短的路径,对基本的蚁群算法进行了改进,这主要体现在以下几个方面:利用人工势场概念构造新的启发式信息;引入惩罚因子概念并对蚂蚁搜索得到的路径施加惩罚因子;修改了全局信息素更新规则。

2.1 新的启发式信息

基本算法中的启发式信息计算公式为

$$\eta_{kd}(t) = \frac{1}{d_{kd}}. \quad (1)$$

式中: η_{kd} 表示位置 k 的启发式信息; d_{kd} 表示位置 k 到终点 d 的直线距离。启发式信息的值与位置到终点的距离成反比例关系,随着位置逐步靠近终点而变大,从而引导蚂蚁向终点运动。但其存在如下问题:1) 由于其不含位置与障碍物间的距离信息,因此,不能引导蚂蚁远离障碍物,搜索到安全路径;2) 其值变化不均匀,当位置与终点的距离大于 1 时,其值变化缓慢,而当距离小于 1 时,其值变化剧烈,导致启发式信息的引导效果不佳。

针对上述问题,借鉴人工势场中引力和斥力的概念,重新构造启发式信息^[11]:

$$\eta_{kd}(t) = \lambda V^{\text{att}}(k) + (1 - \lambda) V^{\text{safe}}(k). \quad (2)$$

式中: $V^{\text{att}}(k)$ 和 $V^{\text{safe}}(k)$ 分别表示位置 k 的引力启发式信息 and 安全性启发式信息, $\lambda (0 < \lambda < 1)$ 表示引力和安全性启发式信息的相对重要程度。

引力启发式信息表达式为

$$V^{\text{att}}(k) = 1 - \frac{d_{\text{goal}}(k)}{\max_{n \in P} |d_{\text{goal}}(n)|}. \quad (3)$$

式中: P 表示路径规划区域内的所有可选位置信息的集合, $d_{\text{goal}}(k)$ 和 $d_{\text{goal}}(n)$ 表示位置 k 和 n 到终点的距离,采用 $d_{\text{goal}}(k)$ 与全局最大值做比值的方法,归一化 $d_{\text{goal}}(k)$ 。

安全性启发式信息表达式为

$$V^{\text{safe}}(k) = 1 - \frac{R(k)}{\max_{n \in P} |R(n)|}. \quad (4)$$

式中: $R(k)$ 表示距离位置 k 最近的障碍物对其的斥力,类似于引力启发式信息的处理方式。归一化 $R(k)$ 得

$$R(k) = \begin{cases} (\frac{1}{d_{\text{obs}}(k)} - \frac{1}{d_0}) \frac{1}{d_{\text{obs}}^2(k)} d_{\text{obs}}(k) < d_0; \\ 0 & d_{\text{obs}}(k) \geq d_0. \end{cases} \quad (5)$$

式中: d_0 是一个给定值, 表示路径应距离障碍物的最小距离, 即安全距离; $d_{\text{obs}}(k)$ 表示位置 k 到障碍物的最短距离。

新的启发式信息由引力和安全性启发式信息按一定比例混合而成, 因而其可以引导蚂蚁避开障碍物过近的位置朝终点运动, 由于采用与最大值比值的方式对 $d_{\text{goal}}(k)$ 和 $R(k)$ 进行了归一化处理, 使得引力和安全性启发式信息不会剧烈变化, 最终使得新的启发式信息克服了原启发式信息变化不均匀的不足。

2.2 惩罚因子

即使采用新的启发式信息, 由于蚁群算法采用伪随机比例规则寻找路径, 因此蚂蚁可能搜索到不安全路径, 为了淘汰这些不安全路径, 提出惩罚因子概念, 惩罚因子是一个施加到路径长度的系数, 其表达式为

$$\delta = \begin{cases} \frac{d_0}{d_p} & d_p < d_0; \\ 1 & d_p \geq d_0. \end{cases} \quad (6)$$

式中: d_0 代表一个给定的安全距离, d_p 表示路径离障碍物的最短距离, δ 满足 $\delta \geq 1$ 。假设任意一只蚂蚁 k 寻找到的路径的长度为 LT_k , 考虑路径长度和路径离障碍物的最短距离, 得到综合路径距离 L_k 为

$$L_k = \delta LT_k. \quad (7)$$

可知, 如果 $d_p < d_0$, 则因路径距离障碍物过近, 其长度被增大; 如果 $d_p \geq d_0$, 则路径由于与障碍物保持安全距离而保持长度; 因此, 不安全路径因综合路径长度比安全路径大而被淘汰, 安全路径被保留。

2.3 新的全局信息素更新规则

在基本算法中, 所有蚂蚁同时从起点出发, 当蚂蚁都到达终点后, 依据蚂蚁搜索出的路径的长度, 选择最短的路径进行全局信息素更新, 而后所有蚂蚁重新从起点出发, 继续上述过程。该方案存在以下问题: 1) 有些蚂蚁可能提前到达终点, 但其必须等待

其他蚂蚁, 不能立刻重新开始新的搜索过程; 2) 只对每次中的最短路径进行全局信息素更新, 容易使算法陷入局部最优解^[12]。

为了解决上述问题, 进一步提高算法效率, 将全局信息素更新规则作如下更改: 当任意蚂蚁 k 到达终点后, 利用式(1)计算综合路径长度, 与此时的全局最优解做比较, 如果其优于最优解, 则用其更新最优解, 再采用式(8)更新蚂蚁搜索到的路径的信息素, 而后立刻从起点出发开始下一轮路径搜索, 不必等待其他蚂蚁都到达终点。

$$\tau_{ij}(k) = (1 - \rho) \tau_{ij}(k) + \rho \Delta \tau_{ij}(k). \quad (8)$$

式中: $\tau_{ij}(k)$ 表示蚂蚁 k 搜索到的路径 (i, j) 的信息素残留量; ρ 为信息素挥发系数, 满足 $0 < \rho < 1$; $\Delta \tau_{ij}(k)$ 表示信息素增量, 其计算公式为

$$\Delta \tau_{ij}(k) = \frac{L^{\text{elite}}}{L^k}. \quad (9)$$

式中: L^{elite} 表示全局最优路径的综合路径长度, L^k 表示蚂蚁 k 搜索到路径的综合路径长度。

采用异步搜索方式, 即蚂蚁到达终点后立刻进行下一轮搜索, 有效地提高了搜索效率。对每条路径按一定方式进行全局信息素更新, 避免了算法陷入局部最优解。

2.4 算法流程

改进算法的运行过程如下: 初始时刻蚂蚁全部位于起点节点, 而后蚂蚁不断地利用 2.3 节的改进式(8)搜索当前位置的临近位置集合, 并选择其中之一作为下一位置, 所有蚂蚁重复此过程并向终点并行搜索路径; 当任意一只蚂蚁到达终点节点后, 利用新的全局信息素更新规则更新信息素, 再将蚂蚁置于起点节点, 重新开始新的搜索, 重复上述过程直到满足算法终止条件, 算法终止条件为连续 M 次全局最优解没有得到更新。算法结束后, 全局最优解作为结果路径, 算法的流程图如图 2 所示。

3 仿真结果与分析

所有仿真实验的实现均采用 Java 语言, 进行仿真实验的计算机配置如下: Duo Core CPU T7100 @

1. 86GHz、1.00GB 内存、Window XP SP3 操作系统.

文中分别设计了 2 个仿真实验来验证算法在路径安全性和执行效率上的效果: 1) 在 $60\text{ m} \times 60\text{ m}$ 的区域内, 采用四叉树对区域进行建模, 通过改进蚁群算法与基本蚁群算法的比较实验, 验证改进蚁群算法规划出的路径的安全性; 2) 在 $1\,000\text{ m} \times 1\,000\text{ m}$ 的区域内, 通过栅格法和基本蚁群算法混合规划策略及四叉树和改进蚁群算法混合策略的对比实验, 验证四叉树和改进蚁群算法的结合在大尺寸区域进行路径规划时的效率.

如图 3 所示, 在 $60\text{ m} \times 60\text{ m}$ 的区域内, 展示了基本蚁群算法和改进蚁群算法规划出的路径. 图 4 展示了图 3 这 2 种算法规划的路径上各个位置距障碍物最短距离, 其中距障碍物安全距离设置为 5 m , 由图可知, 基本蚁群算法规划的路径部分位置距障碍物的距离小于安全距离, 而改进蚁群算法规划的路径能与障碍物保持安全距离, 因而, 其规划的路径具有安全性.

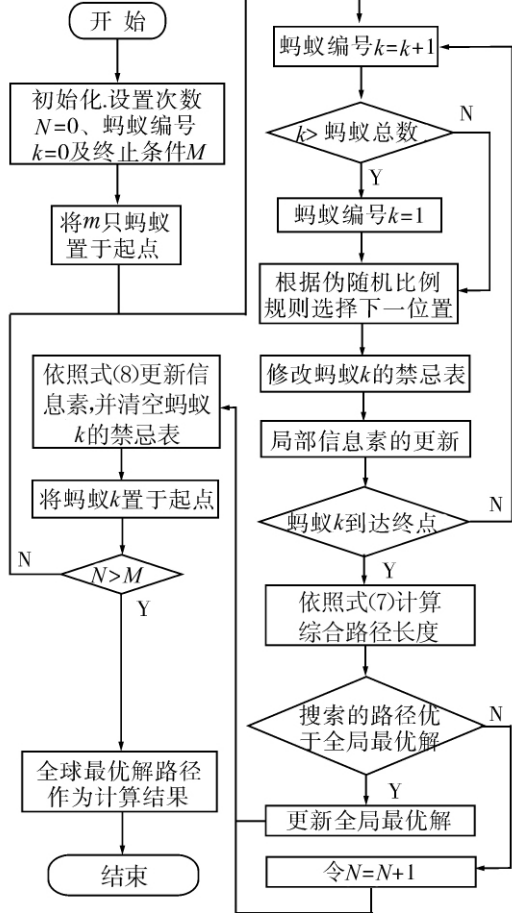


图 2 改进蚁群算法流程

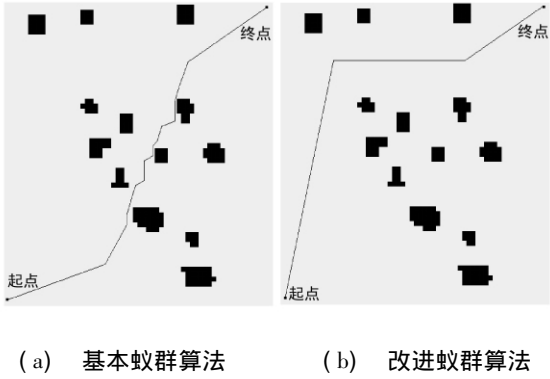


图 3 2 种蚁群算法规划的路径的效果

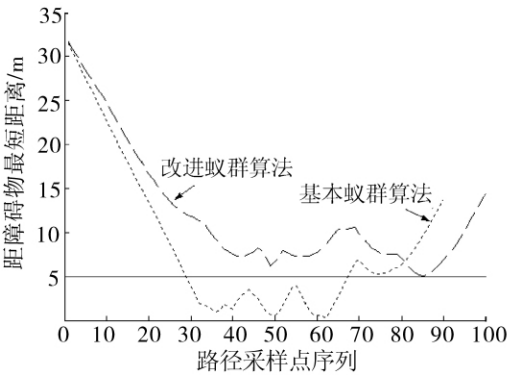


图 4 路径安全性比较曲线

图 5 展示了 2 种规划方法在 $1\,000\text{ m} \times 1\,000\text{ m}$ 的区域内路径规划的效果图, 由图(a) 和图(b) 的对比可知, 相对于基本蚁群算法, 改进的蚁群算法在大尺寸区域上规划的路径是安全的.

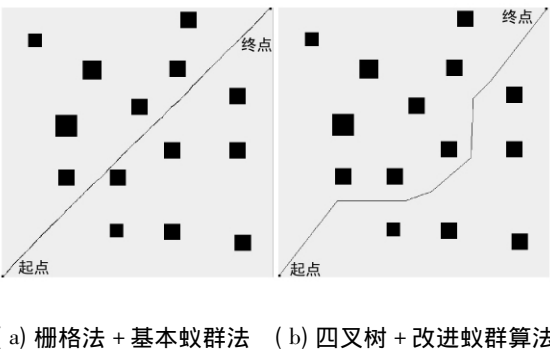


图 5 2 种方法规划的路径的效果

表 1 显示了不同环境建模方法对图 5 中 $1\,000\text{ m} \times 1\,000\text{ m}$ 大尺寸区域建模的结果, 可知, 四叉树法建模的节点数远小于栅格法的节点数, 四叉树法相对于栅格法, 可以对环境信息进行有效的压缩. 表 2 显示了图 5 中 2 种规划方法所用的时间, 可知四叉树

和改进蚁群算法混合规划策略的耗时约为栅格法和基本蚁群算法混合策略的 1.1% ,因此 ,四叉树和改进蚁群算法混合规划策略提高了对大尺寸区域路径规划的效率.

表 1 2 种环境建模方法比较

环境建模	自由节点数	障碍物节点数	总数
栅格法	946 400	53 600	1 000 000
四叉树	4 290	3 288	7 578

表 2 2 种路径规划方法执行效率比较

路径规划方法	耗时 /ms
栅格法 + 基本蚁群算法	113 438
四叉树 + 改进蚁群算法	1 296

4 结束语

提出了四叉树和改进蚁群算法相结合的路径规划方法.改进后的蚁群算法规划的路径可以与障碍物保持一定的安全距离 ,并且采用四叉树对环境进行建模 ,完整保存环境信息的同时极大地压缩了环境信息 ,将 2 种方法相结合不但可以保证规划的路径的安全性 ,还可以提高对大尺寸区域路径规划的效率.最后 ,通过仿真实验表明了新方法的有效性.

参考文献:

[1]毛宇峰. 智能水下机器人路径规划技术研究[D]. 哈尔滨: 哈尔滨工程大学 ,2006: 1-50.

[2] BRAND M , MASUDA M , WEHNER N. Ant colony optimization algorithm for robot path planning [C]// International Conference on Computer Design and Applications. Qinhuangdao , China , 2010: 436-440.

[3] 杨惠. 基于粒子群和蚁群融合算法的移动机器人路径规划研究[D]. 长沙: 长沙理工大学 ,2010: 1-35.

[4] 刘佑鑫 , 赵英凯. 元胞蚁群路径规划方法[J]. 机床与液压 , 2009 , 37(6) : 113-115.

[5] 薛英花 , 田国会 , 吴皓 , 等. 智能空间中的服务机器人路径规划[J]. 智能系统学报 , 2010 , 5(3) : 260-265.

[6] 印峰 , 王耀南 , 刘炜 , 等. 体速度差异的蚁群算法设计及仿真[J]. 智能系统学报 , 2009 , 4(6) : 528-533.

[7] 高涛. 基于八叉树模型的水下机器人三维全局路径规划研究[D]. 哈尔滨: 哈尔滨工程大学 , 2003: 27-33.

[8] 施斌. 虚拟场景下自动路径选择的研究[D]. 哈尔滨: 哈尔滨工程大学 , 2008: 5-16.

[9] 龚健雅. 一种基于自然数的线性四叉树编码[J]. 测绘学报 , 1992 , 21(2) : 90-98.

[10] 李士勇. 蚁群算法及应用[M]. 哈尔滨: 哈尔滨工业大学出版社 , 2004: 35-41.

[11] 肖乐斌 , 龚建华 , 谢传节. 线性四叉树和线性八叉树邻域寻找的一种新算法[J]. 测绘学报 , 1998 , 27(3) : 195-203.

[12] LEE J W , LEE J J. Novel ant colony optimization algorithm with path crossover and heterogeneous ants for path planning[C]//2010 IEEE International Conference on Industrial Technology (ICIT) . Vina del Mar: Chile , 2010: 559-564.