

# 基于差分进化和稀疏 A\* 算法的 游戏地图智能寻径方法

齐 燕<sup>1</sup>, 常晋义<sup>2</sup>

(1. 苏州托普信息职业技术学院 信息技术学院, 江苏 昆山 215311;

2. 常熟理工学院 计算机科学与工程学院, 江苏 常熟 215500)

**摘 要:** 提出了一种基于差分进化和稀疏 A\* 算法的游戏地图智能寻径方法, 通过精准过滤非正常状态从而达到较高的收敛速度进而实现快速路径搜寻. 仿真结果表明, 该方法相较于现有方法有着更快的搜寻速度和更优的路径选择, 对游戏地图智能寻径是可行有效的.

**关键词:** 游戏地图; A\* 算法; 差分进化; 智能寻径

**中图分类号:** TP301.6 **文献标识码:** A **文章编号:** 1671-6876(2021)02-0132-07

**DOI:** 10.16119/j.cnki.issn1671-6876.2021.02.007

## 0 引言

随着人工智能(AI)技术的发展,越来越多的产业与人工智能相结合产生了较好的应用效果,例如:路径规划、计算机视觉、服务式机器人等<sup>[1-3]</sup>. 其中,智能寻径方法<sup>[4]</sup>因人工智能算法的成熟而受到了越来越多的关注,并成为目前的研究热点.

近年来,游戏行业的发展也将寻径方法带入了游戏产业,Non-Player Character(NPC)的出现也造成了游戏地图中的智能寻径方法的广泛应用. 若想使得 NPC 角色更加灵活与智能化,游戏设计研究者必须开发一个较好的寻径方法来搜寻最佳路径. 较为经典的 Dijkstra 算法<sup>[5]</sup>可以较好地完成寻找路径这一任务,但是存在最终路径并非最优路径且算法稳定性较差等问题. Greedy 算法<sup>[6]</sup>也是寻径算法中的一种,该算法通过导出局部最大值或者最小值来寻找最优路径,但 Greedy 算法只取局部最优,并不需要考虑当前的决定对之后决策的影响,所以尽管 Greedy 算法给出了接近于最优的结果,但是还不能给出最优路径. 目前在游戏地图寻径问题上应用最广泛的是 A\* 算法<sup>[7]</sup>,是采用一种直接搜索的方法来求得静态网络中的最优路径. A\* 算法通过比较当前路径栅格的 8 个邻居的启发式函数值 F 来逐步确定下一个路径栅格,但是无法计算出来存在多个最小值的情况,即不能出现最优解. 同时, A\* 算法当面临地图较大且复杂的时候运行速度较慢,并不能应用到实时场景当中.

为了解决 A\* 算法在游戏地图寻径当中面临的问题,本文提出了一种基于差分进化和稀疏 A\* 算法的游戏地图智能寻径算法,该算法通过对 A\* 算法进行剪枝处理<sup>[8-9]</sup>去除冗余操作,从而实现实时要求并且通过差分进化算法来提高总体的全局寻优能力.

## 1 预备知识

### 1.1 差分进化算法

差分进化算法是一种有效的全局优化方法,主要是基于群体式搜索,然后通过交叉、变异、选择等操作来求得最终的最优解的一种方法. 具体如算法 1<sup>[10]</sup>所示.

**算法 1** Differential Evolution Algorithm, DE

**Input:** 种群:  $M$ ; 交叉因子:  $D$ ; 迭代次数  $T$

收稿日期: 2020-12-18

基金项目: 江苏省高校自然科学基金项目(15KJB520001)

通信作者: 齐燕, 讲师, 硕士, 研究方向为人工智能及软件开发应用.

```

    t ← 1
    for i = 1 to M do
        for j = 1 to D do
             $x_{i,t}^j = x_{\min}^j + rand(0,1) * (x_{\max}^j - x_{\min}^j);$ 
        end
    end
    while ( $|f(\Delta)| \geq \epsilon$ ) or ( $t \leq T$ ) do
        for i = 1 to M do
             $\Rightarrow$ (Mutation and Crossover)
            for j = 1 to D do
                 $v_{i,t}^j = \text{Mutation}(x_{i,t}^j);$ 
                 $u_{i,t}^j = \text{Crossover}(x_{i,t}^j, v_{i,t}^j);$ 
            end
             $\Rightarrow$ (Selection)
            if  $f(u_{i,t}) < f(x_{i,t})$  then
                 $x_{i,t} \leftarrow u_{i,t};$ 
                if  $f(x_{i,t}) < f(\Delta)$  then
                     $\Delta \leftarrow x_{i,t};$ 
                end
            else
                 $x_{i,t} \leftarrow x_{i,t};$ 
            end
        end
        t = t + 1
    end
    return the best  $\Delta$ 

```

## 1.2 A\* 算法

A\* 算法<sup>[11]</sup>是一种应用在游戏地图中寻找最短路径的较有效的方法,其使用直接搜索方式,并且采用启发式函数来计算终点距离起点的最小代价<sup>[12]</sup>. A\* 算法不仅仅在游戏地图寻径当中应用,目前还在智能机器人的最短路径规划以及封闭式场景货车最佳路线的规划中使用<sup>[13-15]</sup>. 代价函数是 A\* 算法中重要的一步.

定义 1  $V = \{v_i \mid i \in 1, 2, \dots\}$  表示在游戏地图中起点距终点所经过的路径节点集合,  $v_i$  表示在游戏地图中起点距终点中的某一节点.

定义 2 对于节点  $v_n$ , 其代价函数  $f(v_n)$  可定义如下:

$$f(v_n) = \sigma(v_n) + \rho(v_n) \quad (1)$$

其中,  $\sigma(v_n)$  为起点距离当前节点的实际代价, 而  $\rho(v_n)$  为当前节点距终点的预估代价.

根据定义 2 可知, 代价函数  $f(v_n)$  也称为距离函数, 是启发式函数的一种, 在游戏地图寻径中, 代价即距离. 预估代价  $\rho(v_n)$  越接近于真实值, 其代价函数功能就越强. 当  $\rho(v_n) = \sigma(v_n)$  时, A\* 算法即变为 Dijkstra 算法, 即通过增加横向节点数目来提高算法的寻径能力, 同时, 随着横向节点的增多, 其运行速度也同样会变慢, 使得算法的整体效率变慢. 所以如何确定实际代价  $\sigma(v_n)$  与预估代价  $\rho(v_n)$  在总体代价函数  $f(v_n)$  中的权重是非常重要的.

$\rho(v_n)$  必须满足代价真实性原则与代价一致性原则. 代价真实性即  $\rho(v_n)$  的取值不能超过当前节点距目标节点的真实代价. 而代价一致性原则要求  $\rho(v_n)$  满足式(2)所示条件:

$$\rho(v_{i+1}) + \kappa(v_i, v_{i+1}) \geq \rho(v_i) \quad (2)$$

其中,  $\kappa(v_i, v_{i+1})$  为节点  $v_i$  距下一节点  $v_{i+1}$  的真实代价, 而  $\rho(v_{i+1})$  为下一节点  $v_{i+1}$  距最终节点的预估代价.

定理 1 如果  $\rho(v_i) \leq \rho(v_{i+1}) + \kappa(v_i, v_{i+1})$ , 那么距最终节点所有路径  $f(v_n)$  全部为单调费递增函数.

证明 首先假设  $v_{i+1}$  为  $v_i$  的下一个目标节点, 则

$$f(v_{i+1}) = \sigma(v_{i+1}) + \rho(v_{i+1}) = \sigma(v_i) + \sigma(v_i, v_{i+1}) + \rho(v_{i+1}) \geq \sigma(v_n) + \rho(v_n) \quad (3)$$

A\* 算法是一个较为经典的寻径问题的解决方案, 但是当其应用在游戏地图寻径问题上时, 因为游戏地图寻径需要较低的时延, 所以其效率与准确率产生了一定的冲突, 并不能通过有效的手段来控制较低的时延下产生较好的路径规划. 本文提出了一种基于差分进化和稀疏 A\* 算法的游戏地图 AI 智能寻径算法来应对此问题.

## 2 主要结果

在 A\* 算法研究的基础上, 使用差分进化方法结合稀疏 A\* 算法来生成最佳游戏地图路径, 主要是因为在实际应用场景中, 游戏地图中的 NPC 是实时变化的, 角色在线不能完全按照初始路径规划方法来确定唯一路径, 需要试试检测当时所在路径点距离终点的动态最佳路径. 所以, 建立了一种重规划算法来确定游戏最佳路径, 同时考虑了路径的最优性与路径重规划的实时性. 由于 A\* 算法耗时较长并不能作为实时的路径重规划算法, 为了避免算法搜索空间中的无用中间步骤与节点, 对 A\* 算法进行剪枝处理, 称为稀疏 A\* 算法. 首先, 设计了以下稀疏 A\* 算法的代价函数, 稀疏 A\* 算法与 A\* 算法相类似, 需要同时对实际代价与预估代价设计. 具体如下所示.

$\rho(v_n)$  为角色  $P$  在游戏地图中节点  $v_n$  处的预估代价

$$\rho(v_n) = w_1 * \Gamma(P) \quad (4)$$

其中,  $w_1$  为角色  $P$  实际代价的权重系数, 而  $\Gamma$  为相邻路径距离之和,  $\Gamma$  可由以下计算得到

$$\Gamma = \sum_{i=0}^{N-1} S_{i,i+1} \quad (5)$$

其中,  $S_{i,i+1}$  为两个相邻节点的距离.

$\sigma(v_n)$  为角色  $P$  在游戏地图中节点  $v_n$  处的实际代价, 将其设置为当前节点  $v_n$  与目标节点  $v_m$  的欧氏距离, 则为

$$\sigma(v_n) = \sqrt{(x_n - x_m)^2 + (y_n - y_m)^2} \quad (6)$$

其中,  $(x_n, y_n)$  为当前节点  $v_n$  的横纵坐标, 而  $(x_m, y_m)$  为目标节点  $v_m$  的横纵坐标.

所以, 稀疏 A\* 算法的总体代价函数  $f(v_n)$  为

$$f(v_n) = w_1 * \Gamma(P) + w_2 * \sigma(v_n) \quad (7)$$

其中,  $w_2$  为实际代价函数的权重系数.

稀疏 A\* 算法的算法流程如图 1 所示.

根据稀疏 A\* 算法的流程, 其具体稀疏 A\* 算法如算法 2 所示.

算法 2 A\* Algorithm

Input: Game map data

Set the start and end points  $v_i$  and  $v_j$  of this Path;

Put the inaccessible points in the Close table, and the start points in Open table;

While Current node is not a target point

Calculate cost of all feasible successors

$$\sigma(v_n) = \sqrt{(x_n - x_m)^2 + (y_n - y_m)^2}$$

$$f(v_n) = w_1 * \Gamma(P) + w_2 * \sigma(v_n)$$

For  $i = 1$  to total number of reachable grids

If Nodes are in the Open table and cost less

Update Open table

End

```
If node  $v_i$  not within Open table
Put  $v_i$  join Open table
End
End
Sorting the nodes in the Open table by cost;
If Open table  $\neq \emptyset$ 
Put node with smallest cost into Close table
Else
No accessible trail
End
End
Output: the best path
```

稀疏 A\* 算法可以较好地寻找到最优路径,并可以在面临任何突发情况下可以更好地进行重规划,但是稀疏 A\* 算法在游戏地图寻径当中可能陷入局部最优的情况,从而较难发现游戏地图中的全局最优路径. 因此,引入差分进化算法来对稀疏 A\* 算法进行优化,通过差分进化和稀疏 A\* 算法相结合,克服了差分进化方法寻径中不能重规划的缺点,同时也加强了稀疏 A\* 算法在寻找全局最优路径上的能力. 本文提出的基于差分进化和稀疏 A\* 算法的游戏地图智能寻径算法如算法 3 所示.

算法 3 Game Map Path-Finding Method Based on Differential Evolution and Sparse A\* Algorithm

```
Input: Game map data
Use Differential Evolution to plan an optimal path  $\langle v_i, \dots, v_j \rangle$  for an NPC from the start point  $v_i$  to the goal point  $v_j$ ;
The NPC advances along  $\langle v_i, \dots, v_j \rangle$ , constantly detecting whether NPC has reached the target point as it does so;
If NPC not reach the end
Reprogramming using the sparse A* algorithm;
Else
Statistical optimal path;
Output: the best path
```

3 仿真实验结果及分析

为了验证本文提出的游戏地图寻径方法的有效性和可应用性,选取山脉地形与城市建筑群地形对本文算法与原稀疏 A\* 算法进行仿真试验,实验所采用的硬件平台与软件环境具体为:CPU: Intel-Core i7-9700K; 显卡: GTX 2060; 内存: 16 G; 操作系统: Windows 10; 仿真软件: Matlab.

在山脉地形上对所提出的算法进行仿真试验. 首先在实验中设置 NPC 的起点与终点,其中,在游戏地图中的起点为 (0,0,0),然后将游戏地图寻径的终点设置为 (100,100,100). 在实验中,假设该 NPC 为匀速前进,其速度为 3 m/s,图 2 和图 3 是稀疏 A\* 算法在游戏地图寻径上的表现,分别为 2 维平面上的展示和 3 维实际平面上的展示.

接下来,将稀疏 A\* 算法在山脉地形上进行仿真试验,同样的,地图中的起点为 (0,0,0),终点同样设置为 (100,100,100). 图 4 和图 5 是本文算法寻径的表现,分别为 2 维平面上的展示和 3 维实际平面上的展示.

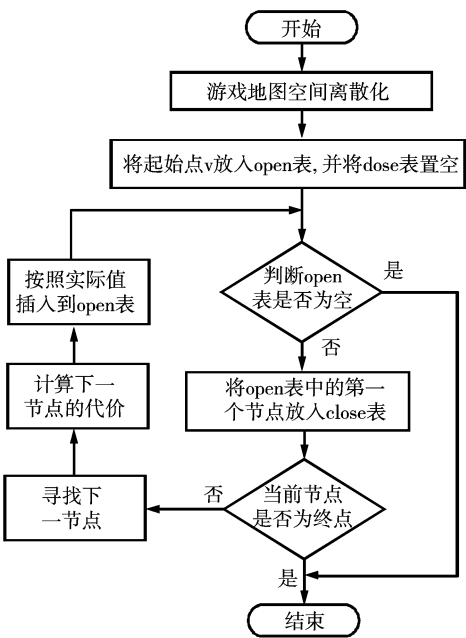


图 1 稀疏 A\* 算法的算法流程

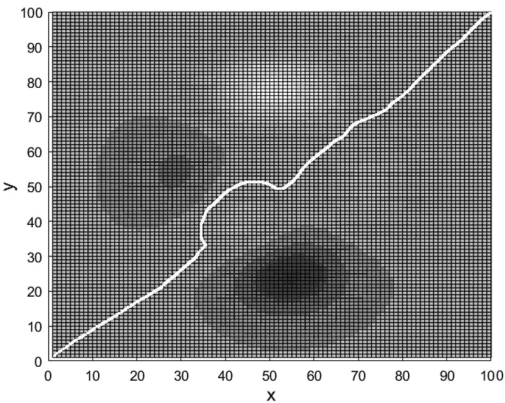


图 2 稀疏 A\* 算法在山脉地形上的表现(2 维)

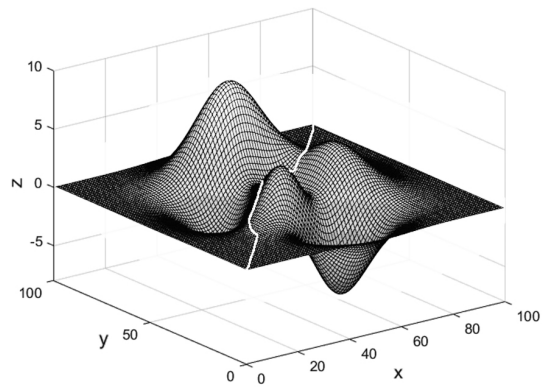


图 3 稀疏 A\* 算法在山脉地形上的表现(3 维)

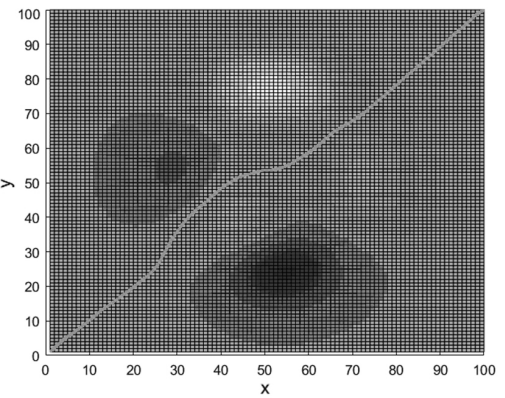


图 4 本文方法在山脉地形上的表现(2 维)

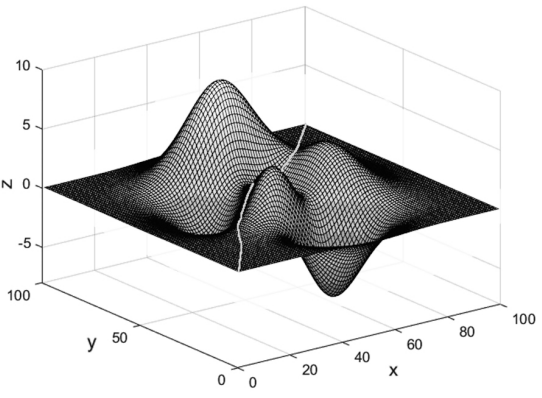


图 5 本文方法在山脉地形上的表现(3 维)

由图 2—5 可以看出,本文方法相较于稀疏 A\* 算法有着较好的路径规划能力且在遇到山脉等情况时会进行重规划而进一步选取更优的路径。

最后将稀疏 A\* 算法在城市建筑群地形上进行仿真试验。图 6—9 为稀疏 A\* 算法与本文算法寻径的表现,分别为 2 维平面上的展示和 3 维实际平面上的展示。

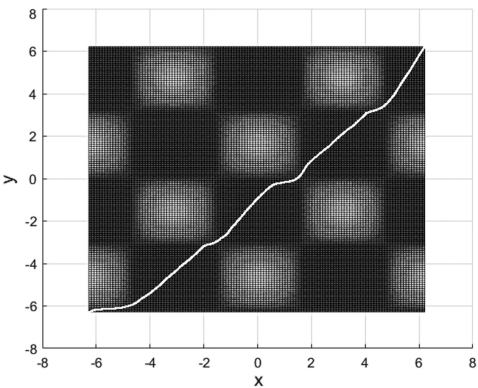


图 6 稀疏 A\* 算法在城市建筑群地形上的表现(2 维)

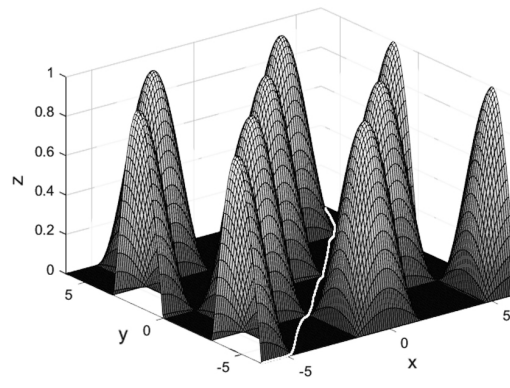


图 7 稀疏 A\* 算法在城市建筑群地形上的表现(3 维)

表 1 本文方法与稀疏 A\* 算法的性能对比

算法	运行时间/s	路径长度	总体代价
稀疏 A* 算法	21.97	197.67	247.39
本文算法	22.78	156.52	201.44

由图 6—9 可以看出,本文方法相较于稀疏 A\* 算法在面临城市建筑较多的情况下可以更早地规避城市建筑,具有更优的路径规划。

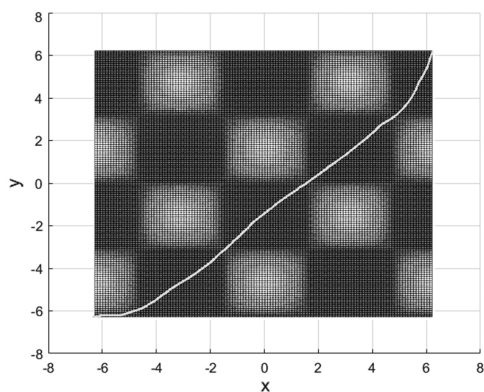


图8 本文方法在城市建筑群地形上的表现(2维)

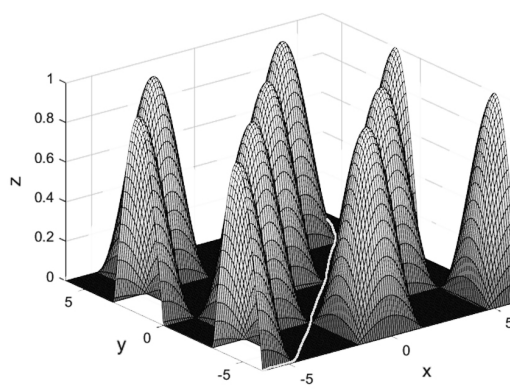


图9 本文方法在城市建筑群地形上的表现(3维)

本文方法是稀疏A\*算法的改进算法,因此与稀疏A\*算法在性能上进行比较是有必要的.接下来对所提出的算法与稀疏A\*算法在运行时间、路径长度和总体代价上进行了对比,表1为本文方法与稀疏A\*算法的性能比较,可以看出本文方法所规划出来的路径距离更短,总代价更小.

#### 4 结语

人工智能技术迅速发展,已经在各方面取得了重要的成就,本文针对A\*算法在游戏地图寻径中容易陷入局部最优导致不能找到最优路径,且由于差分进化算法不能进行重规划的问题,提出了一种新的游戏地图AI技术寻径方法.该方法基于差分进化和稀疏A\*方法相结合,在游戏地图寻径上取得了较好的结果.仿真实验证明了本文算法相较于稀疏A\*算法具有更低的总体代价、更短的路径长度,能够较好地适用于游戏地图的寻径研究.

#### 参考文献:

- [1] WEINAN E, VANDEN E. Transition-path theory and path-finding algorithms for the study of rare events[J]. Annual Review of Physical Chemistry, 2010, 61(1):391-420.
- [2] LIU Y, JIN L. Deep Matching Prior Network: Toward Tighter Multi-oriented Text Detection[EB/J]. [https://openaccess.thecvf.com/content\\_cvpr\\_2017/papers/Liu\\_Deep\\_Matching\\_Prior\\_CVPR\\_2017\\_paper.pdf](https://openaccess.thecvf.com/content_cvpr_2017/papers/Liu_Deep_Matching_Prior_CVPR_2017_paper.pdf).
- [3] 宋健, 张铁中, 徐丽明, 等. 果蔬采摘机器人研究进展与展望[J]. 农业机械学报, 2006(5):158-162.
- [4] LI J, HARABOR D, STUCKEY P J, et al. Symmetry-Breaking Constraints for Grid-Based Multi-Agent Path Finding[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2019(33):6087-6095.
- [5] ALYASIN A, ABBAS E I, HASAN S D. An Efficient Optimal Path Finding for Mobile Robot Based on Dijkstra Method[C]. 2019 4th Scientific International Conference Najaf (SICN), 2019:189-197.
- [6] 方红, 杨海蓉. 贪婪算法与压缩感知理论[J]. 自动化学报, 2011(12):1413-1421.
- [7] LIU X, WEI G, SUN J W, et al. Nonlinear multifunctional sensor signal reconstruction based on least squares support vector machines and total least squares algorithm[J]. 浙江大学学报(A卷英文版), 2009, 10(4):497-503.
- [8] ZHAO H, KIT C. A Simple and Efficient Model Pruning Method for Conditional Random Fields[C]. Computer Processing of Oriental Languages Language Technology for the Knowledge-based Economy, International Conference, Iccpol, Hong Kong, March. DBLP, 2009:1109-1116.
- [9] HE L, ZHANG J. Snowflakes Removal for Single Image Based on Model Pruning and Generative Adversarial Network[C]. 2019 IEEE 4th International Conference on Image, Vision and Computing (ICIVC) IEEE, 2019.
- [10] STORN R, PRICE K. Differential Evolution: A Simple and Efficient Heuristic for global Optimization over Continuous Spaces[J]. Journal of Global Optimization, 1997, 11(4):341-359.
- [11] KEVIN D, JEANNE D. Exploring the Conformational Space of Growth-Hormone-Releasing Hormone Analogues Using Dopant Assisted Trapped Ion Mobility Spectrometry-Mass Spectrometry[J]. The journal of physical chemistry, 2019, 123(29):6169-6177.
- [12] KOLDA T G, TORCAON L V. Optimization by Direct Search: New Perspectives on Some Classical and Modern Methods[J]. SIAM Review, 2003, 45(3):385-389.
- [13] SILMAN L A, BARZILY Z, PASSY U. Planning the route system for urban buses[J]. Computers & Operations

Research, 1974, 1(2):201-211.

- [14] SMILANSKY Z. Miniature autonomous agents for scene interpretation[J]. Electrotehnica Electronica Automatica, 2004,12(3):67-75.
- [15] ZHAO X, YANG J, LIN L I, et al. Path Tracking Control for Autonomous Underground Mining Articulated Dump Truck[J]. Electrotehnica Electronica Automatica, 2015,8(2):46-53.

## A Game Map Path-Finding Method based on Differential Evolution and Sparse A\* Algorithm

QI Yan<sup>1</sup>, CHANG Jin-yi<sup>2</sup>

(1. Suzhou Top Institute of Information Technology Institute of Information Technology, Kunshan Jiangsu 215311, China)

(2. School of Computer Science and Engineering, Changshu Institute of Technology, Changshu Jiangsu 215500, China)

**Abstract:** In this paper, A game map path finding method based on differential evolution and sparse A\* algorithm was propose, which achieves fast path finding by accurately filtering non-normal states to achieve high convergence speed. The experimental results show that the proposed method has faster search speed and better path selection than the existing methods, and that the AI intelligent path finding method based on differential evolution and sparse A\* algorithm was feasible and effective.

**Key words:** Artificial intelligence; A\* algorithm; differential evolution; intelligent pathfinding

[责任编辑:蒋海龙]

(上接第 124 页)

## English Teaching Quality Evaluation based on Decision Tree and Rule Association Analysis

CHEN Xiao-hong<sup>1</sup>, AN Shuo-yu<sup>2</sup>, QI Hui<sup>3</sup>

(1. School of Humanity and Education, Guangzhou Nanyang Polytechnic College, Guangzhou Guangdong 510925, China)

(2. School of Marxism, Shanghai Jiao Tong University, Shanghai 200240, China)

(3. Department of Computer, Taiyuan Normal University, Jinzhong Shanxi 030619, China)

**Abstract:** In order to improve the effectiveness of English teaching quality evaluation, decision tree was used to establish the classification rules of teaching quality evaluation indicators, and association rule analysis was used to verify the evaluation results. Firstly, the effective indicators of English teaching quality evaluation were extracted, and the attributes of the indicators were quantified and generalized; then the entropy gain of each indicator attribute was calculated according to the samples to be evaluated, and the entropy gain rate was sorted in descending order to obtain the root node and branch node respectively, and the number of branches was obtained combined with the attribute value of the node, and then a complete decision tree of teaching quality evaluation was constructed; finally, the association rules were used. Then the support and confidence of each attribute were calculated by the analysis method, and the effectiveness of the decision tree evaluation was verified according to the results of association analysis. Experiments showed that the key factors and internal relations that affect the quality of English teaching can be better evaluated and analyzed by setting the evaluation index parameters scientifically and selecting the evaluation index attribute segmentation interval reasonably.

**Key words:** teaching quality evaluation; decision tree; entropy gain; association rules; *confidence*

[责任编辑:李春红]