

8-17-2020

## Improved Genetic Algorithm-based Network Game Path Selection and Simulation

Jianping Gu

*1. Lishui University, Lishui 323000, China;;*

Mingmin Zhang

*2. Zhejiang University, Hangzhou 310000, China;*

Meiliang Wang

*1. Lishui University, Lishui 323000, China;;*

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

---

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

---

# Improved Genetic Algorithm-based Network Game Path Selection and Simulation

## Abstract

**Abstract:** Traditional optimal path algorithm only sets the shortest path as the target, and it does not consider the network congestion and the number of users in game area for real-time situation, thus resulting in some limitations. According to the actual circumstance of network game, network game path selection model was proposed, and the improved genetic algorithm was employed for simulation. The method pre-processed the game map to get each road weighted length value for a real-time game map, and optimization solution was obtained through the genetic algorithm. *A network game path selection method based on improved genetic algorithm was proposed compared with the classical dijkstra algorithm, effectively improving the efficiency of the path search.*

## Keywords

genetic algorithm, online game, path choice, modeling and simulation

## Recommended Citation

Gu Jianping, Zhang Mingmin, Wang Meiliang. Improved Genetic Algorithm-based Network Game Path Selection and Simulation[J]. Journal of System Simulation, 2016, 28(8): 1805-1811.

# 基于改进遗传算法的路径选择算法及仿真实现

顾健萍<sup>1</sup>, 张明敏<sup>2</sup>, 王梅亮<sup>1</sup>

(1. 丽水学院, 丽水 323000; 2. 浙江大学, 杭州 310000)

**摘要:** 传统路径最优算法把路径最短作为目标, 不考虑网络拥堵和游戏区域人数等实时情况, 使得算法存在局限性。根据网络游戏实际情况, 提出网络游戏路径选择模型, 并用改进的遗传算法仿真实现。对游戏地图与游戏相关数据进行预处理, 得到游戏地图各路段实时加权长度值, 通过遗传算法进行寻优求解。提出一种基于改进遗传算法的网络游戏路径选择方法, 与经典算法 Dijkstra 相比, 有效提高了路径搜索效率。

**关键词:** 遗传算法; 网络游戏; 路径选择; 建模与仿真

中图分类号: TP391.9

文献标识码: A

文章编号: 1004-731X (2016) 08-1805-07

## Improved Genetic Algorithm-based Network Game Path Selection and Simulation

Gu Jianping<sup>1</sup>, Zhang Mingmin<sup>2</sup>, Wang Meiliang<sup>1</sup>

(1. Lishui University, Lishui 323000, China; 2. Zhejiang University, Hangzhou 310000, China)

**Abstract:** Traditional optimal path algorithm only sets the shortest path as the target, and it does not consider the network congestion and the number of users in game area for real-time situation, thus resulting in some limitations. According to the actual circumstance of network game, network game path selection model was proposed, and the improved genetic algorithm was employed for simulation. The method pre-processed the game map to get each road weighted length value for a real-time game map, and optimization solution was obtained through the genetic algorithm. A network game path selection method based on improved genetic algorithm was proposed compared with the classical dijkstra algorithm, effectively improving the efficiency of the path search.

**Keywords:** genetic algorithm; online game; path choice; modeling and simulation

## 引言

在网络游戏领域中有很多专家学者对游戏路径问题进行研究, 如潘志庚等提出了运动目标跟踪自动优化(TAO-MTP)算法<sup>[1]</sup>, Chong Jin Ong<sup>[2]</sup>、石祥滨、钟瑛<sup>[3-4]</sup>等都对网络游戏路径算法进行深入研究。求最短路径的算法有很多, 公认较好的算法

是 Dijkstra 标号法, 但实验结果表明, Dijkstra 标号法也有需要改进的地方, 其退出机制对不联通的有向图是无效的, 会陷入死循环<sup>[5]</sup>。而本研究通过对遗传算法的改进不存在不连通的路径问题。Dijkstra 算法的数据结构及其实现方法是以牺牲时间效率来换取空间节省, 存在计算时间较长, 性能低下的问题。单源多目标路径问题, 可以采用 Steiner 解决, 途径有二种, 一种是精确算法, 另一种是启发算法, 前者需要很长的时间, 而启发算法能在较短时间得到解, 但算法缺乏全局性, 很容易陷入局部最优解<sup>[6]</sup>。而网络游戏路径选择不仅仅是传统意义上的路径最短或时间最少的问题, 选择时



收稿日期: 2014-08-30      修回日期: 2015-02-01;  
基金项目: 2014 浙江省科技厅公益技术应用研究  
(2013C33G2260014);  
作者简介: 顾健萍(1970-), 浙江上虞, 硕士, 讲  
师, 研究方向为虚拟现实、数字媒体。

<http://www.china-simulation.com>

• 1805 •

还要考虑到网络拥堵, 游戏人数等因素。目前, 路径选择多采用智能路径优化算法, 如遗传算法与蚁群算法, 具有全局收敛和动态适应等优点, 并且容易与其他智能算法结合<sup>[7]</sup>, 对于蚁群算法当生成的蚁群较大, 算法初期又缺乏信息素时, 要在短时间内找出一条较好的路径就显得比较困难, 并且有搜索停滞和无效搜索问题存在<sup>[8]</sup>; 遗传算法使用概率机制进行迭代, 具有随机性, 可避免陷入局部最优解, 但搜索没有反馈机制, 搜索具有盲目性, 最优解的收敛速度也较慢, 求解效率低。

为了实现网络游戏最优路径搜索, 并提高搜索效率, 首先利用游戏系统, 获得游戏网络及游戏人数等相关数据, 建立游戏信息模型, 然后利用遗传算法全局搜索能力强的优点进行搜索, 并对遗传算法采用启发算法改进, 并且消除路径搜索过程中产生的“回路”与“死路”, 避免产生大量的无效染色体, 解决遗传算法收敛速度较慢, 求解效率低的问题, 最后通过仿真实验对其性能进行验证, 实践证明该算法具有较快的收敛速度与求解效率。

## 1 网络游戏路径选择系统数学模型

网络游戏路径选择系统分为二部分, 其一是建立网络游戏地图路径拓扑结构, 其二是建立网络游戏路径选择数据模型。每个游戏都有游戏地图, 通过游戏地图抽象出游戏地图路径拓扑结构, 以智慧白云山项目中漫游地图如图 1 为例, 其对应的路径拓扑结构如图 2, 其中节点标示交叉路口, 单实线表示道路路段。

本文采用游戏地图上路段实际长度与路段网速及网络拥堵情况相结合选择路段权值, 若二节点之间不连通, 则二节点间距离为 $\infty$ , 用一个很大的数来表示<sup>[9]</sup>, 本文采用 10 000 来表示。

网络游戏中二点间的行走时间计算以跑动为标准进行计算, 行走时间分为二部分: 第一是路段正常行走时间  $T_{ijr}(t)$ , 计算方法为路径长度除以人物正常网速下跑动速度,  $T_{ijr}(t)=L_{ij}/V$ , 其中  $L_{ij}$  为节点  $i$  至节点  $j$  间路段长度,  $V$  为单人游戏人物在节

点间跑动速度,  $i$  与  $j$  为邻节点。第二是受游戏人数影响而加速或延迟的时间  $T_{ijl}(t)$ , 可以在游戏数据库中获得相应数据后进行预处理。



图 1 网络游戏地图

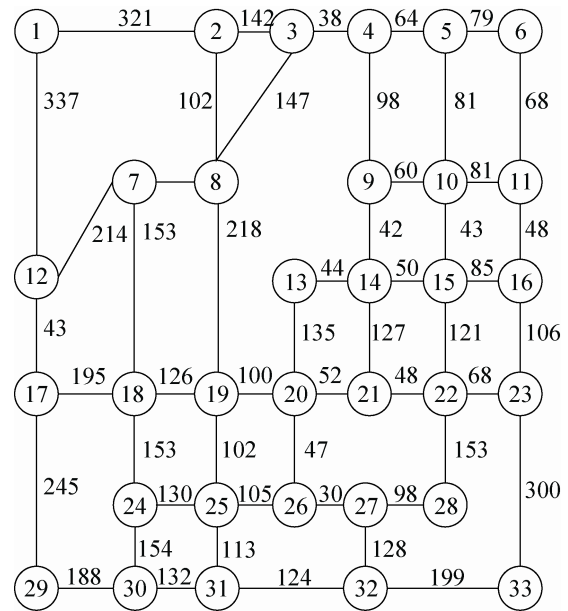


图 2 网络游戏节点加权长度拓扑结构

因此, 从出发节点到目标节点间的行走时间, 即整条路径的行走时间  $T_p(t)$  为:

$$T_p(t) = a_1 \times \sum_{j=\text{出发节点}}^{\text{目标节点}} T_{ijr}(t) + a_2 \times \sum_{i=\text{出发节点}}^{\text{目标节点}} T_{ijl}(t) \quad (1)$$

式中:  $a_1$  为路径长度,  $a_2$  游戏人数影响因素权重系数,  $a_1+a_2=1$ 。

上述函数在使用时还要注意单位的转换, 因为时间单位与路径单位不在同一级别上, 用权重系统去直接衡量时, 可能会有失重的现象产生, 若时间以毫秒为单位, 路径以厘米为单位, 则每毫秒相当于多少厘米, 应根据游戏人物跑动速度进行换算, 可使权重系数一致。假设  $\alpha_1=0.7$ ,  $\alpha_2=0.3$  时, 图 2 为某一时刻经过路径选择预处理后得到网络游戏地图各路段加权长度拓扑结构图。

## 2 改进遗传算法的网络游戏路径选择算法设计

遗传算法有 6 个主要要素: 参数的编码、初始种群的设定、适应度函数的设计、遗传操作、算法控制参数的设定与约束条件<sup>[10]</sup>。以下对本研究中各个要素进行详细设计。

### 2.1 个体编码

染色体编码(即个体编码)好坏将直接影响选择、交叉、变异等遗传操作。在本算法中, 将一条路径作为一个染色体, 将一个节点作为一个基因。为了方便个体编码, 采用等长编码的方法, 如图 2 中共有 33 个结点, 则染色体的长度为 33。为了方便编码及容易理解, 采用符号编码法, 用自然数来表示节点, 节点的顺序即按顺序通过各交通路口的顺序。设定在路径的出发节点到目标节点之间不回头路, 即某个节点在某条路径中最多只能出现一次, 且二个节点之间如果没有连通路段, 则这二个节点不能顺序出现, 因为顺序出现则该路径为无效无径, 目标节点之后的其他节点均用 0 替代, 表示没有行驶过的路段, 以保持每条路径的节点数为定长, 如该例中为 33 个节点, 从节点 1 出发到节点 33 结束, 其中一条路径编码为: 1 2 8 19 25 26 27 28 22 23 33 0。

### 2.2 初始种群与优化

为了避免在开始种群中产生大量无效路径, 采用随机选择节点与启发算法相结合的方法, 既可提高搜索效率又可得到全局最优解<sup>[11]</sup>。具体方法是:

将开始节点作为第一个基因, 在本例中第一个基为 V1, 然后扫描数据中与 V1 相连的所有节点, 随机选择其邻近的任一节点作为染色体中下一个基因如 V2, 再以 V2 为出发点随机选择一个相邻节点作为下一个基因如 V8, 依次类推, 直到目标节点 V33, 如 V1-V2-V8-V19-V25-V26-V27-V28-V22-V23-V33 即为用启发算法得到的一条染色体(即一条路径)。但用该启发算法时有可能出现二种例外情况: 第一是在某节点处选择相邻接点作为下一个基因时, 该相邻节点在该条染色体中已作为基因被选择过, 若再次被选择, 路径中就会出现回路的情况, 而大量回路的出现, 会导致算法效率大大降低。本算法中采用将所有 33 个节点作为一个向量存放, 路径选择时每选择一个节点就从该节点向量中除去该元素, 下一邻接点选择时只能从该向量中存在的元素中选择, 如此可避免出现回路; 第二是当路径中的节点按邻接点的方向选择, 且每个节点只能出现一次时, 会出现某条路径选择到某个节点时, 在节点向量中的元素里已经没有与该节点邻接的节点, 即该路径是一条“死路”, 在本规则的约束下, 不能到达目的节点处, 例如: V1-V2-V8-V7-V12-V17-V18-V19-V20-V13-V14-V15-V22-V21, 到 V21 后, 与该节点相连的节点只有 V20 与 V14, 而这二个节点已被该路径选择过, 解决此问题的方法, 是直接将该条染色体从初始种群中删除, 另外从新选择符合条件的路径作为染色体。如此循环选择路径, 直到需要的初始种群大小。

### 2.3 适应度函数

适应度函数所计算的适应度值应该是非负数, 且适应度越大其个体越优良<sup>[12]</sup>。而网络游戏路径选择中对于个体的路径加权长度值则越小越好, 故需要对适应度函数进行变换, 本算法中采用 1 减去每条路径的加权长度值与所有路径中最大路径长度值的比值作为适应度函数, 故个体的适应度函数  $Fit(x)$  定义为:

<http://www.china-simulation.com>

$$Fit(x) = (1 - P_i(x) / \max(P_i(X))) \quad (2)$$

式中： $P_i(x)$ 为某条路径的加权长度； $\max(P_i(x))$ 为种群中所有路径中最长路径。适应度函数值在 0~1 之间，函数值越大，适应度越好。

## 2.4 遗传操作

遗传操作有：选择、交叉重组、变异几个部分。

选择：本算法采用比例选择法，根据各个体适应度值的大小，择优选择适应度大的个体进入下一代<sup>[13]</sup>。

交叉重组：为避免节点随机交叉后，出现不存在的路段的状况即“死路”，本算法采用在父代二条染色体中任意抽出一段首尾相同节点进行交叉，并判断交叉后的染色体中是否出现“回路”，若有回路，则删除所有回路。

变异：为避免随机选择节点变异产生死路与回路的问题，本文采用重选路径变异算子方法。

## 2.5 算法控制参数的设定与约束条件

本算法的控制参数主要有初始种群的设定、遗传代数、变异系统。本算法中初始研究节点只有 33 个，路径总数不是很多，因此初始种群设定为 50，遗传代数暂定为 20 代，变异系统设为 1，根据实验结果可以适当调整这些参数以使算法发挥最大效率。

本算法的约束条件有 3 个：(1)是采用定长染色体，染色体中基因个数等于地图中节点总数；(2)是从出发节点到目标节点之间不能存在回路，若有回路也会破坏第一个约束条件；(3)是路径中不能存在死路。

## 3 仿真实验

### 3.1 实验环境

本算法的实验环境采用 P4 双核 3.6G CPU, 8G 内存，操作系统为 Windows 7，在 Matlab 软件上用改进的遗传算法进行仿真实现，并在同样的实验环境下与经典算法 Dijkstra 进行对比分析。

## 3.2 网络游戏最优化路径仿真实现

### (1) 初始化种群

设初始种群大小为 50，按上述启发式种群初始化方法，将图 2 中带权网络游戏道路拓扑结构所有路段用  $33 \times 33$  的矩阵  $V$  存放，其中每个元素  $V(i,j)$  存放节点  $i$  到节点  $j$  之间的路段值， $i$  和  $j$  都取 1 到 33 的自然数，当节点  $i$  与节点  $j$  不连通或  $i$  等于  $j$  时取 10000。 $V$  矩阵一部分如图 3 所示。

%	1	2	3	4	5	6	7	8
$V=$	10 000	321	10 000	10 000	10 000	10 000	10 000	10 000
	321	10 000	142	10 000	10 000	10 000	10 000	102
	10 000	142	10 000	38	10 000	10 000	10 000	147
	10 000	10 000	38	10 000	64	10 000	10 000	10 000
	10 000	10 000	10 000	64	10 000	79	10 000	10 000
	10 000	10 000	10 000	10 000	79	10 000	10 000	10 000
	10 000	10 000	10 000	10 000	10 000	10 000	10 000	146
	10 000	102	147	10 000	10 000	10 000	146	10 000
	10 000	10 000	10 000	98	10 000	10 000	10 000	10 000

图 3  $V$  矩阵部分元素

接着定义一个  $50 \times 33$  的零矩阵  $P$ ，用于存放 50 条随机选择的从出发节点到目标节点之间的路径作为初始种群，然后用三重循环来实现随机选取 50 条从 1~33 的路径，最内层用于查询所有与  $i$  相连并且还没有被当前路径选择过的邻节点，并从这些邻节点中随机选择一个节点作为与  $i$  相连的节点  $j$ ，修改  $P(i,j)$  元素值；第二层循环用于控制一条路径是否被正常选择，正常选择的标准是用邻节点随机选择方法，最终选择到终节点 33，若在选择过程中出现前述所述“死路”状况，则直接中断该循环，并将该路径删除，删除的方法是该路径上所有节点重新置 0，并返回重选该路径，最外层控制选择 50 条路径。其中某次运行结果如图 4 所示。从程序结果可以发现，路径中既不存在回路，也不存在死路。

### (2) 目标函数适应度函数

该研究中的目标函数就是计算每条路径的加权长度，方法是将图 4 所示的  $P$  矩阵中的每一行中从出发节点 1 到目标节点 33 之间，顺序二节点之间的路段  $V(P(i,j), P(i,j+1))$  累加，其中  $i$  是从 1~50

的行号,  $j$  是从节点 1 到节点 32 所对应的列号, 每条路径目标函数值存入  $PL$  矩阵。

1	2	8	7	18	19	20	13	14	21	22	28	27	26	25	31	32
1	2	3	8	19	25	26	20	13	14	21	22	28	27	32	33	0
1	2	3	4	5	10	11	16	23	33	0	0	0	0	0	0	0
1	2	3	4	5	10	15	14	13	20	21	22	23	33	0	0	0
1	2	3	4	5	6	11	16	23	22	28	27	32	33	0	0	0
1	12	7	18	19	25	24	30	31	32	33	0	0	0	0	0	0
1	12	7	8	2	3	4	5	10	9	14	13	20	26	27	32	33
1	12	7	18	19	20	21	22	23	33	0	0	0	0	0	0	0
1	12	7	8	2	3	4	9	14	15	16	23	33	0	0	0	0

图 4 初始种群路径一部分

由于适应度函数值一般是由 0~1 之间的一个值, 并且适应度值越大目标函数的适应度就越好, 而本算法中计算最短路径, 因此目标函数值越小, 其对应染色体越优秀, 所以采用式(1)方法, 将种群中每条路径值与最大路径值相除, 得到 0~1 之间的值, 但因为该值越大适应度应该越小, 所以本研究中再用 1 去减该值, 得到 0~1 之间的值, 其值越大适应度越好, 计算机结果存于向量  $FIX$  中。

(3) 选择算子

本研究采用比例选择法, 根据适应度值分配, 然后采用随机遍历采样的方法选择进入下一代的个体。其调用函数为:  $SelCh=select('sus', Chrom, Fix, GGAP)$ 。Select 函数中  $sus$  为底层随机遍历函数。Chrom 为种群,  $Fix$  为适应度值,  $GGAP$  为代沟, 本研究取 0.9。

(4) 交叉重组操作

交叉概率控制着交叉操作的频率, 该算子太大会使高适应值的结构很快破坏掉, 该算子太小, 则会使搜索停滞不前, 这里取 0.5。本算法的思路是, 随机抽取交叉样本数的一半, 每一个交叉样本都与未被选取样本却未交叉过的剩余个体匹配。交叉匹配后路径同样要解决“死路”、“回路”以及交叉后的个体与原父代的个体相同的问题, 采用 3.4 中的实现方法为: 将父代的二条染色体从第一个节点开始比较, 直到有不节点, 标志该节点位置  $flag1$ , 再将第一个个体中的  $flag1$  后每个节点与第二个个体中  $flag1$  后的所有节点匹配, 直到找到相同节点,

将这个相同节点的第一个个体节点位置标志为  $flag2$ , 第二个个体的节点位置标志为  $flag3$ , 再将第一个个体的  $flag1$  与  $flag2$  之间的路段与第二个个体的  $flag1$  与  $flag3$  之间的路段交叉, 得到二个新的子代个体, 若在  $flag1$  之后找不到相同节点, 则放弃第二个父代个体, 重新选择一个新的父代个体, 直到二个个体满足上述条件, 这样方法交叉后的个体不会存在死路以及交叉后子代与父代相同问题。上述交叉过程如下图 5 所示。

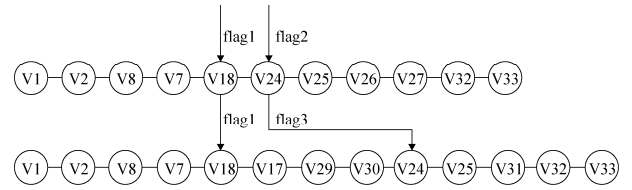


图 5 交叉重组

但交叉后的二个个体子代染色体可能会存在回路, 而且有可能存在多条回路, 回路的存在虽然不会影响最优路径的求解, 但会降低收敛速度, 而且因为回路的存在, 交叉后的子代染色体节点数也有可能超过原地图的节点总数, 本例中有可能超过 33, 导致种群矩阵列数增加, 因此必须删除所有回路, 例如: 如下二条父代染色体:

1-2-8-3-4-9-14-13-20-26-25-24-18-17-29-30-31-32-33

1-12-7-18-17-29-30-31-25-26-27-32-33

在第 2~10 点位置发生交叉(节点 1 与节点 25 首尾二个节点相同, 则交叉重组这部分节点), 交叉后的子代染色体如下:

1-12-7-18-17-29-30-31-25-26-25-24-18-17-29-30-31-32-33

1-2-8-3-4-9-14-13-20-26-27-32-33

其中子代第一条染色体的标号为 18 节点出现回路, 将回路删除后第一个子代染色体为:

1-12-7-18-17-29-30-31-32-33

删除回路后还有可能存在别的回路, 用同样的方法继续删除回路。并使种群的行列数恢复到原来大小。

## (5) 变异操作

变异操作采用重选路径变异操作, 采用方法为在路径中随机选择一个节点, 从这个节点出发, 重新随机选择一条与原路径不同的路径作为变异后的染色体, 重新选择路径的方法与选择操作相同, 不同点在于不是从 1 节点开始, 而是从随机选择的节点开始, 因此该方法同样可以避免回路与死路, 但有可能使收敛速度过快, 很容易陷入局部最优解, 因此适当的加大交叉重组与变异率可以避免收敛速度过快的问题, 此处变异率设为 0.1。

## (6) 实验结果以及与 Dijkstra 比对

采用上述方法, 经多次运行, 上例 33 个节点的网络游戏最优路径寻找, 采用遗传算法在 3 代以内就能得到最优解。上例最优路径值为 1166, 最优路径为: 1-2-3-4-5-6-11-16-23-33。得到最优解运行时间为 15 s, 运行结果如图 6。

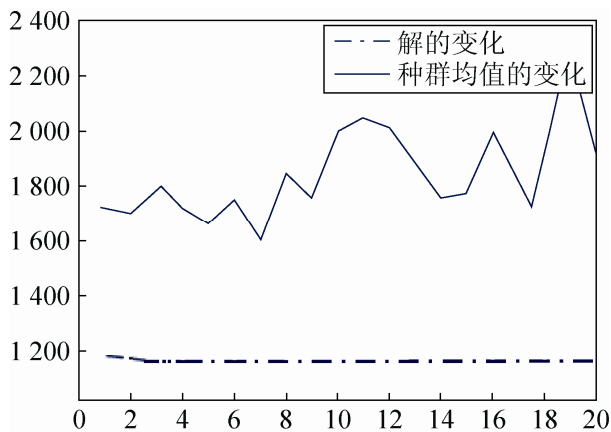


图 6 33 节点实验结果

由图可知, 上例有 33 个节点的网络游戏路径的仿真结果, 收敛速度极快, 效率极高。若将游戏地图扩大到 60 个节点的加权道路拓扑结构, 如图 7 所示。

当游戏地图扩展到 60 个节点时, 采用上述方法, 多次运行 5 代以内能得最优解, 其中一次运行结果如图 8, 最优路径值为 1 351, 最优路径为: 1-2-3-4-5-6-11-39-40-42-45-46-47-60, 得到最优解运行时间为 58 s, 收敛极快。

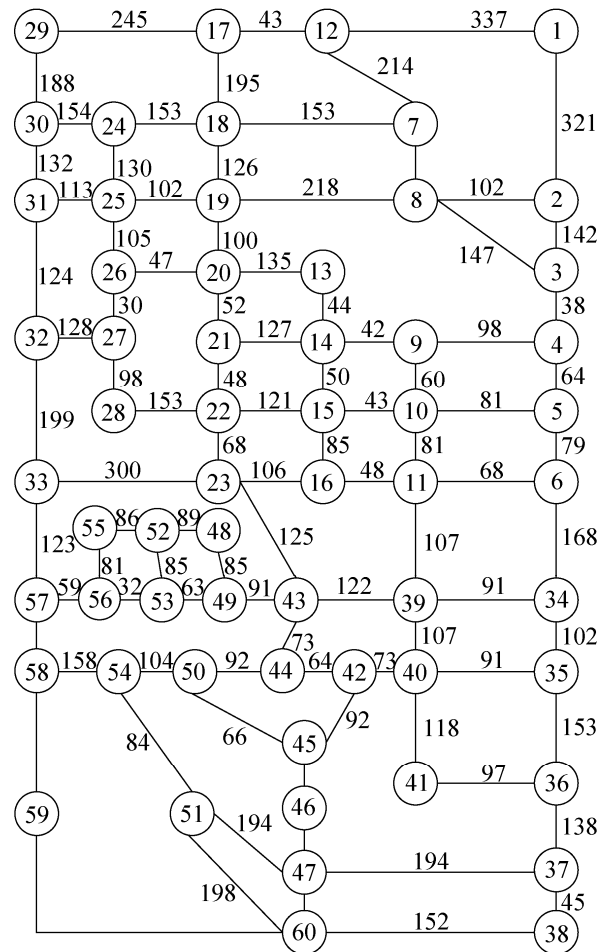


图 7 60 节点带权游戏地图拓扑结构

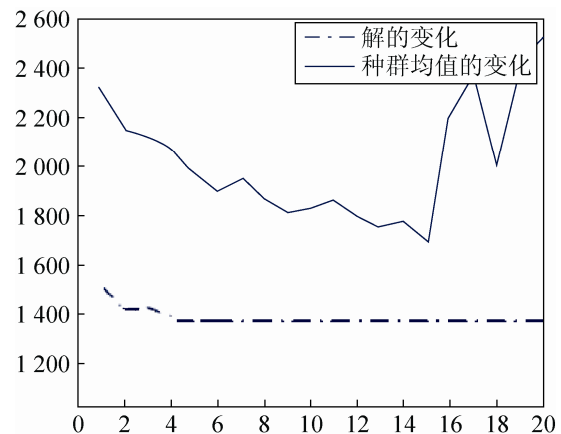


图 8 60 节点网络游戏自动寻径实验结果

求解单源最优路径的经典算法为 *dijkstra*, 上例有 33 个节点的网络游戏路网在同等条件下若采用经典 *dijkstra* 算法, 同样在 *matlab* 平台上运行, 得到最短路径实验结果也会 1 166, 路径为: 1-2-8-19-20-26-27-32-33, 运行时间为 28 s, 运行速



度比优化遗传算法慢约 46%。当地图扩展到 60 个节点时, 最短路径实验结果为: 1-351, 路径为: 1-2-3-4-5-6-11-39-40-42-45-46-47-60。运行时间为 74 s, 比本研究中遗传算法运行速度慢约 21%。

将优化遗传算法与经典 dijkstra 算法比较, 用 dijkstra 运行得到的一定是最优解, 但是当游戏地图很大, 节点数很多时, 采用多重循环耗用资源将会成指数级增加, 运行效率也会变的很差, 用遗传算法得到的可能只是局部最优解, 但随着游戏地图增大以及游戏节点数增加, 资源耗用将成线性增长, 还可以采用优化遗传算法的方法, 使遗传算避免陷入局部最优解, 同时提高算法的收敛速度, 从而使算法效率、准确性更高。

## 4 结论

本文提出一种基于优化的遗传算法的网络游戏最优路径查找模型, 并在 MATLAB 上采用优化遗传算法仿真实现, 并在同等条件下与经典求解最优路径的 dijkstra 方法相比较。从实验结果可看出在遗传算法中引入一种改进的交叉与变异方法, 在网络游戏自动寻径中能加快收敛速度, 从而快速有效地寻找到最优路径。与 Dijkstra 算法相比, 可以缩短寻径时间, 满足了网络游戏对实时性要求, 有效解决网络游戏自动寻径问题。

该算法还存在一定的不足之处, 从仿真实现中可以发现, 该算法收敛速度过快, 有可能会使本文遗传算法陷入局部最优解, 从本算法的多次实验结果来看, 33 个节点的算法中曾到过局部最优解, 虽然局部最优解不影响网络游戏寻径, 但会影响整体游戏运行速度, 所以还可以在遗传控制参数设定中做适当的调整, 适当加大交叉率与变异率, 使网络游戏在自动寻径中始终能得到全局最优解。另外, 本文迭代次数选择了 20 代, 从实验结果看, 60 个节点的网络拓扑结构, 5 代以内就可以得到最优解, 所以要提高网络游戏自动寻径速度, 还可以根据节点度的多少来调整迭代次数, 过多的迭代次

数会浪费整个算法的实现时间。最后, 本文在解决回路与死路的问题上花费较多的时间, 如何有效提高回路与死路的处理效率也是需要进一步研究的问题。

## 参考文献:

- [1] Mingliang Xu, Zhigeng Pan, Hongxing Lu, et al. Moving-Target Pursuit Algorithm Using Improved Tracking Strategy [J]. IEEE Transactions on Computational Intelligence and AI in Games (S1943-068X), 2010, 2(1): 27-39.
- [2] Chong Jin Ong. Distance Computation between Smooth Convex Objects [C]// Proceedings of IEEE International Conference on Robotics and Automation, Minnesota, USA. USA: IEEE, 1996: 785-790.
- [3] 石祥滨, 赵鑫, 刘芳, 等. MMOG 中 NPC 的路径搜索算法的研究 [J]. 微型计算机系统, 2006, 27(12): 2206-2211.
- [4] 钟瑛, 许明. 融合 A\*和 Bresenham 的游戏路径规划算法设计与实现 [J]. 电子技术与软件工程, 2013 (9): 64-66.
- [5] 王树西, 吴政学. 改进的 Dijkstra 最短路径算法及其应用研究 [J]. 计算机科学, 2012, 39(5): 223-228.
- [6] Mark L Psiaki. Block Acquisition of weak GPS signals in a software receiver [C]// The Institute of Navigation, Salt Lake City, USA. USA: IEEE, 2001: 2838-2850.
- [7] 侯燕. Steiner 树遗传蚁群算法在路径选择中的应用 [J]. 微电子学与计算机, 2013, 30(11): 88-92.
- [8] Tsui J B. Fundamentals of Global Positioning System Receivers: A Software Approach [M]. New York, USA: Wiley, 2000.
- [9] 王海雄, 郭剑毅, 张月红. 改进遗传算法求解交通最优路径的实现 [J]. 昆明理工大学学报(理工版), 2009, 34(4): 42-46.
- [10] Gunes M, Sorges U, Bouazizi I. ARA the Ant Colony based Routing Algorithm for MANETs [C]// International Conference on Parallel Processing Workshop (SICPPW02). Vancouver BC, Canada: IEEE, 2002: 79-85.
- [11] 张敏捷, 蔡延光, 宋康, 等. 基于改进遗传算法的路网路径优化方法 [J]. 微计算机信息, 2010, 26(2): 226-227.
- [12] 张本群. 基于启发式算法的路径规划 [J]. 计算机仿真, 2012, 29(10): 341-344.
- [13] 邱敏, 王公宝, 张松涛, 等. 一种求解最优路径的改进遗传算法 [J]. 计算机与现代化, 2010 (4): 6-8.