游戏引擎最短路径搜索优化遗传算法设计*

黎忠文[1]，覃志东[2†]，王全宇[2]，倪仲余[1,3]

(1. 成都大学 信息科学与技术学院，成都 610106；2. 东华大学 计算机科学与技术学院，上海 201620；3. 西华大学 数学与计算机学院，成都 610039)

摘　要： 为满足游戏地图中最短路径搜索求解，提出了一种优化的自适应遗传算法。该算法采用与游戏地图中节点数和弧段数相关联的节点复杂度算子，结合种群的整体情况和进化潜力来设定自适应遗传算法的交叉率和变异率。实验表明，该算法避免了搜索结果陷入局部最优解，确保最短路径的搜索成功率及提高搜索速度，在游戏引擎设计中具有一定的实用价值。

关键词： 遗传算法；最短路径；节点复杂度算子；交叉率；地图

中图分类号： TP301.6　　**文献标志码：** A　　**文章编号：** 1001-3695(2014)01-0076-04

doi:10.3969/j.issn.1001-3695.2014.01.017

Optimized genetic algorithm for shortest-path problem in game engine

LI Zhong-wen[1], QIN Zhi-dong[2†], WANG Quan-yu[2], NI Zhong-yu[1,3]

(1. *School of Information Science & Technology, Chengdu University, Chengdu* 610106, *China*; 2. *School of Computer Science & Technology, Donghua University, Shanghai* 201620, *China*; 3. *School of Mathematics & Computer Engineering, Xihua University, Chengdu* 610039, *China*)

Abstract: In order to meet the shortest-path searching in game map, this paper presented an optimized adaptive genetic algorithm. It used the environment operator which was associated with the number of nodes and arcs of the graph to set the crossover probability and the mutation probability considering the whole population distribution and evolutionary potential. Experiments show that the algorithm can avoid falling into local optimal solutions, ensure the search success rate, and improve the search speed. It provides some valuable work in game engine designing.

Key words: genetic algorithm; shortest path; node complexity operator; crossover probability; map

0　引言

在计算机游戏设计中，生成游戏地图中起点到目标点的最短路径是最基本的问题之一，游戏中很多的功能都需要在此基础上实现，如图 1 所示。最简单的如角色的自动寻路功能，假如给出的最短路径求解算法本身效率低且速度慢，或者该算法容易陷入局部最优解，或者该算法对于复杂的地图根本就无法求解，那么就会出现玩家等待时间过长、走错路、乱跑、绕圈子、短距离可以实现而长距离无法实现游戏角色自动寻路功能等奇怪现象。所以在游戏引擎设计时，需要对诸如最短路径求解提供兼顾搜索速度与搜索成功率的基本算法，甚至偏重搜索速度或者偏重搜索成功率的多种基本算法，以供游戏设计时视其具体功能选用[1]。

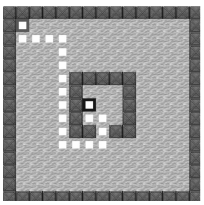

图 1　游戏设计中的最短路径问题示意图

针对最短路径求解问题，先后有深度优先、广度优先、Dijkstra、A* 等诸多算法被提出。其中，深度优先和广度优先算法属于状态空间搜索算法，其最大缺陷是搜索速度低，在状态空间很大或者不可预测的情况下无法给出最优解；Dijkstra 算法能够保证搜索成功率，但是由于需要遍历计算的节点过多，所以搜索速度慢；而 A* 算法属于启发式搜索算法，只需产生全部状态空间的部分节点及关系即可求解，因此，搜索效率较高，但由于该算法没有回溯，不能确保寻找到的路径一定是最优路径。

近年来，遗传算法因为其强大的并行搜索能力，能够解决复杂的和非线性的问题而被用来求解最短路径问题[2]。就搜索速度而言，遗传算法优于 Dijkstra 算法而劣于 A* 算法；就搜索成功率而言，遗传算法优于 A* 算法而劣于 Dijkstra 算法。游戏者的实时体验是一个重要的需求，所以，在游戏引擎设计中一般都给出了 A* 算法实现。但是采用 A* 算法实现的最短路径算法搜索成功率难以满足一些具体游戏功能的较高要求。文献[3]研究了遗传算法和 Dijkstra 算法在求解动态权值系统中最短路径时的性能问题，通过比较表明，遗传算法在每张地图上的得分数以及算法所用时间普遍高于 Dijkstra 算法，即遗传算法在求解动态权值系统中最短路径问题时稳定性和预期

收稿日期： 2013-04-17；**修回日期：** 2013-05-24　　**基金项目：** 国家自然科学基金资助项目(60903160)；中央高校基金资助项目(11D11209)；四川省科技支撑计划资助项目(2013GZ0016)；四川省教育厅科技资助项目(12ZB176,13ZA0296)

作者简介： 黎忠文(1970-)，女，教授，博士(后)，主要研究方向为计算机网络安全；覃志东(1974-)，男(通信作者)，副教授，博士，主要研究方向为嵌入式系统、可信计算(qinzhidong@gmail.com)；王全宇(1987-)，男，硕士研究生，主要研究方向为嵌入式系统；倪仲余(1989-)，男，硕士研究生，主要研究方向为计算机网络安全.

效果明显好于 Dijkstra 算法,但其时间复杂度较高。因此,在游戏引擎设计中有必要提供兼顾最短路径搜索速度和搜索成功率的优化遗传算法。

当前,一些学者对遗传算法的内部算子和参数进行了改善[4],以期在求解最短路径时获得较好的搜索速度与成功率。现有研究发现,遗传算法的搜索速度并不仅是单一算子或者参数的反映,而是多种算子、参数之间相互制约和平衡的结果[5]。Srinivas 等人[6]提出的自适应遗传算法,虽在一定程度上提高了算法性能,但由于没有考虑到多算子之间的平衡和制约,故在求解最短路径时搜索速度并不高。相对于其他应用领域,游戏地图最短路径问题求解有其自身特点,只有结合其特点才能设计出提高遗传算法效率的操作算子[7]。

本文为满足游戏引擎设计的需要,基于自然界中环境对生物体的影响因素,根据游戏地图的特点设计了节点复杂度算子,提出了一种优化自适应遗传算法。该算法结合节点复杂度算子对交叉率和变异率进行设定和调整,最终使得内部参数、算子之间相对平衡和制约,从而在确保搜索成功率的基础上提高了搜索速度。

1　自适应遗传算法

使用经典遗传算法求解游戏地图的最短路径问题可以描述为

$$\min F(X) \text{ subject to } X \in S \tag{1}$$

其中:$F(X)$是评价函数;X 是问题的解向量;S 是解向量的定义域。Srinivas 对经典遗传算法的改进之处在于,算法会根据每代个体适应度的改变来自适应地调整交叉率和变异率,这样能够在保护最优个体的基础上加快较差个体的淘汰速度,从而在一定程度上提高遗传算法的计算性能[4]。该算法中 p_c 和 p_m 定义为

$$p_c = \begin{cases} k_1 \dfrac{\text{fit}_{\max} - \text{fit}'}{\text{fit}_{\max} - \text{fit}_{\text{ave}}} & \text{fit}' \geqslant \text{fit}_{\text{ave}} \\ k_3 & \text{fit}' < \text{fit}_{\text{ave}} \end{cases} \tag{2}$$

$$p_m = \begin{cases} k_2 \dfrac{\text{fit}_{\max} - \text{fit}}{\text{fit}_{\max} - \text{fit}_{\text{ave}}} & \text{fit} \geqslant \text{fit}_{\text{ave}} \\ k_3 & \text{fit} < \text{fit}_{\text{ave}} \end{cases} \tag{3}$$

其中:fit_{ave}表示群体适应度的平均值;$\text{fit}_{\max}$表示群体适应度的最大值;fit′表示参与交叉的两个个体中较大的适应度;fit 表示变异个体的适应度。但是该算法存在计算经验性参数较多、负担较重,且在进化后期难以使种群摆脱最优解等缺点。为此,文献[8]提出通过判断群体适应度的集中程度来解决局部最优解问题。p_c 和 p_m 定义为

$$p_c = \begin{cases} p_c \dfrac{1}{1 - \dfrac{\text{fit}_{\min}}{\text{fit}_{\max}}} & \dfrac{\text{fit}_{\text{ave}}}{\text{fit}_{\max}} > a, \dfrac{\text{fit}_{\min}}{\text{fit}_{\max}} > b \\ p_c & \text{otherwise} \end{cases} \tag{4}$$

$$p_m = \begin{cases} p_m \dfrac{1}{1 - \dfrac{\text{fit}_{\min}}{\text{fit}_{\max}}} & \dfrac{\text{fit}_{\text{ave}}}{\text{fit}_{\max}} > a, \dfrac{\text{fit}_{\min}}{\text{fit}_{\max}} > b \\ p_m & \text{otherwise} \end{cases} \tag{5}$$

当 $\text{fit}_{\min}$与 $\text{fit}_{\max}$接近时,整个群体的集中程度就高;当 fit_{ave}与 $\text{fit}_{\max}$接近时,群体内部个体的集中程度就高。当判断出群体集中时,p_c 和 p_m 会自适应变化,避免搜索陷入局部最优解;反之,p_c 和 p_m 就会保持初值。在式(4)和(5)的约束条件中,a 的取值是(0.5,1),b 的取值是(0,1);a 越接近 0.5,b 越接近于 0 时,就表明群体越集中。这种自适应遗传算法避免了局部最优解问题,并且相对于 Srinivas 提出的自适应遗传算法,减少了计算量,一定程度上提高了算法的计算性能,但是由于没有结合游戏地图的特点设计 p_c 和 p_m,所以收敛速度仍然不是很快。

2　改进的优化自适应遗传算法

虽然遗传算法是一种随机优化算法,但是它同时也具有目的性,本文将利用节点复杂度算子指导设定交叉率和变异率的值,这样可以对遗传算法的进化过程进行更好的控制。

2.1　节点复杂度算子

在游戏程序中,载入的地图文件由地图编辑器产生,通过调入地图图片能够将游戏地图分成一定数量的单元格。游戏地图就是由单元格组成的节点集和度相等的弧段构成的。游戏地图的形式化定义如下:

$$\begin{cases} \text{graph} = (V, R) \\ V = \{ x \mid x \in \text{DataObject} \} \\ R = \{ VR \}, VR = \{ \langle x, y \rangle \mid P(x, y) \wedge (x, y) \in V \} \end{cases} \tag{6}$$

其中:DataObject 为一个集合,该集合中的所有元素具有相同的特性;V 中的数据元素通常称为节点;VR 是两个节点之间关系的集合;$P(x,y)$表示 x 与 y 之间有特定的关联属性 P。

游戏地图存储结构一般采用邻接矩阵表示法。邻接矩阵是具有如下性质的 $n \times n$ 矩阵 A:

$$A[i,j] = \begin{cases} w_{ij} & (v_i, v_j) \in VR \\ \infty & (v_i, v_j) \notin VR \end{cases} \tag{7}$$

其中:w_{ij}为两个节点间的权值,若两个节点不连通,则值为无穷。在最短路径问题中,目标函数为 $g(x) = \sum w_{ij}$,因为$g(x) > 0$且是求最小值问题,所以将适应度函数取为目标函数的倒数,即 $f(x) = 1/g(x)$,这样适应度越大的个体就会越容易被选中。

在自然界中,不同种类生物如果生活在条件相同的环境中,在同样选择的作用下,有可能产生功能相同或十分相似的形态结构以适应环境,该现象称为趋同进化[9],如澳大利亚的袋食蚁兽、非洲的土豚、亚洲的穿山甲、南美洲的食蚁兽,就具有相似的生活方式和适于捕食白蚁的相似生理结构;如果同种生物生活在不同环境下,它们的基因变异率会发生差异,如苏门答腊岛的猩猩和非洲的猩猩,基因的变异率就不相同。由此可见,自然界的环境对生物进化的交叉率和变异率的影响都是非常大的。

在游戏中地图环境的复杂度与多种因素相关,其中连通节点的复杂度是关键因素之一。在一定数量的节点中,当弧段越多,即非无穷值越多时,寻找路径的潜力越大,就越要求较高的交叉率和变异率;反之,非无穷值越少时,进化的方向就比较明确,交叉率和变异率就会降低。本文定义节点复杂度算子 env 为

$$\text{env} = \frac{m}{n^2} \tag{8}$$

其中:n 表示图节点的个数;m 值表示邻接矩阵中非无穷值的数量;env 的取值是[0,1)。为了克服算法在演化过程中出现停滞的问题,以及避免 fit_{ave}和 $\text{fit}_{\max}$相差太大导致的性能降低问题,首先调整曲线在 fit_{ave}处缓慢改变,这样可以提高适应度接近平均适应度个体的交叉率和变异率;其次保证较优个体仍具有一定的交叉率和变异率;最后 fit_{ave}和 $\text{fit}_{\max}$相差太大时不会趋

于直线。为此,本文引入 Sigmoid 函数[10],即

$$f(x)=\frac{1}{1+\exp(-x)} \tag{9}$$

该函数能够平衡线性行为和非线性行为,其函数曲线如图 2 所示。

所以定义经过调整后的节点复杂度算子 env 为

$$\text{env}'=\frac{1}{1+e^{5-10\times\text{env}}} \tag{10}$$

其中:env′的值域是[0,1)。

节点复杂度算子本质上就是对算法参数不确定性的一种度量,其值越小就越能明确算法的进化方向,反之则需要加大交叉率和变异率。基于这种内涵,本文将利用节点复杂度算子设计交叉率和变异率的计算公式。

2.2 交叉率的计算

遗传算法中的交叉操作是模拟自然界中有性繁殖的基因重组过程,目的是将参与交叉过程的两个个体中的优良基因遗传给下一代个体,生成基因更复杂的新个体[9]。交叉算子不仅能够增强种群的多样性,还具有实现算法的全局搜索能力。交叉算子由交叉率控制,交叉率越高,种群中引入新结构的速度越快,当然破坏已有的优良基因结构的可能性也就越大,交叉率太低又会造成搜索阻塞。大量的实验证明,交叉率的一般取值是[0.6,1.0)[11]。本文将交叉率定义为

$$p_c=0.6+\text{env}'\times 0.4 \tag{11}$$

交叉率函数曲线如图 3 所示。可知,节点复杂度算子的值越大,交叉率也就越大,产生新个体的速度也就越快,从而有助于提高搜索最短路径的速度。结合式(4),定义交叉率 p_c 为

$$p_c=\begin{cases}\dfrac{0.6+\text{env}'\times 0.4}{1-\dfrac{\text{fit}_{\min}}{\text{fit}_{\max}}} & \dfrac{\text{fit}_{\text{ave}}}{\text{fit}_{\max}}>a,\dfrac{\text{fit}_{\min}}{\text{fit}_{\max}}>b\\ 0.6+\text{env}'\times 0.4 & \text{otherwise}\end{cases} \tag{12}$$

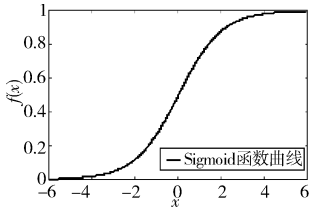

图 2 Sigmoid 函数曲线

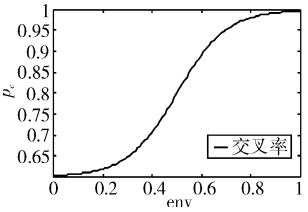

图 3 交叉率取值变化

2.3 变异率的计算

遗传算法的变异操作是模拟自然界中生物进化中染色体上某位基因发生突变的现象,从而改变染色体的结构和物理性状[10]。变异操作是保持群体多样性的有效手段。交叉操作结束后,染色体上的基因按变异率随机改变,这样可以起到产生新个体和抑制早熟的作用。变异率的值过大会使遗传搜索变成随机搜索,太小又可能会使某些基因过早丢失,从而造成信息的丢失。p_m 一般的取值是 0.005 ~ 0.01[11]。本文将 p_m 定义为

$$p_m=0.005+\text{env}'\times 0.005 \tag{13}$$

变异率函数曲线如图 4 所示。可知,节点复杂度算子的值越大,变异率值也就越大,从而有助于维持种群的多样性,避免陷入局部解。再将变异率与式(5)结合,定义变异率 p_m 为

$$p_m=\begin{cases}\dfrac{0.005+\text{env}'\times 0.005}{1-\dfrac{\text{fit}_{\min}}{\text{fit}_{\max}}} & \dfrac{\text{fit}_{\text{ave}}}{\text{fit}_{\max}}>a,\dfrac{\text{fit}_{\min}}{\text{fit}_{\max}}>b\\ 0.005+\text{env}'\times 0.005 & \text{otherwise}\end{cases} \tag{14}$$

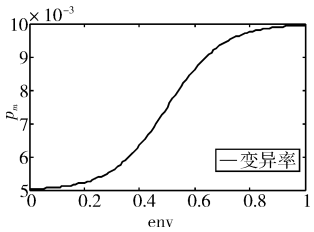

图 4 变异率取值变化

3 数值仿真实验及分析

为了验证本文所设计的自适应遗传算法在求解游戏地图最短路径时的性能,进行了三组数值仿真实验。第一组实验使用标准遗传算法(SGA)、Srinivas 提出的自适应遗传算法(AGA)和本文优化的自适应遗传算法(IAGA)对比测试算法的搜索速度和搜索成功率;第二组实验将分别验证节点复杂度算子相关因子,即节点数、弧段数的影响作用,由式(8)可知,节点复杂度算子值与节点数成反比,与弧段数成正比;第三组实验给出 A* 和遗传算法实现实例的对比。

3.1 算法性能对比测试

测试的游戏地图中,节点数为 20,弧段数为 100,因为是无向图,所以邻接矩阵图中有 200 个非无穷值,种群规模为 90。标准遗传算法的相关参数设置:交叉率为 0.9,变异率为 0.0075;自适应遗传算法的相关参数设置:$k_1=k_3=0.9$,变异率初始值 $k_2=k_4=0.0075$;本文设计的自适应遗传算法的初始设置:$a=0.7$,$b=0.2$。进行了 100 次独立实验。在 100 次实验中,游戏地图的路径会随机发生改变,但是路径的权值和弧段的数量不会改变,最后将 100 次实验的平均值作为检验指标。第一组实验的统计结果如表 1 所示。

表 1 三种遗传算法测试结果

算法种类	平均运行时间/s	搜索成功率/%	平均进化代数	最大进化代数
SGA	0.043 23	78	17.05	31
AGA	0.046 02	94	25.20	52
IAGA	0.045 39	95	23.82	50

由表 1 可知,改进的自适应遗传算法运算速度虽然不如标准遗传算法快,但是搜索成功率得到了非常大的提高,降低了陷入局部最优解的概率,从而提高了遗传算法的质量;Srinivas 提出的自适应遗传算法虽然也提高了搜索成功率,但是由于计算量较大,所以搜索速度不是很好。对比三种遗传算法的测试结果,本文提出的优化自适应遗传算法在保证搜索成功率的基础上提高了搜索效率。

3.2 节点数、弧段数对算法效果的测试及分析

1)节点数对算法效果的测试

本实验设定游戏地图的节点数分别为 10、15、20、25 和 30;弧段数均为 40,即邻接矩阵中有 80 个非无穷值,其余的实验条件与第一组实验相同。实验结果如表 2 所示。表中所列的交叉率和变异率是初始值,不是计算过程中的自适应调整值。

表 2 节点数对改进的自适应遗传算法的影响

节点数	初始交叉率	初始变异率	运行时间/s
10	0.981 0	0.009 8	0.044 02
15	0.676 3	0.006 0	0.045 65
20	0.619 0	0.005 2	0.045 93
25	0.609 5	0.005 1	0.046 28
30	0.606 5	0.005 1	0.046 48

2)弧段数对算法效果的测试

本实验设定游戏地图的节点数为 20,弧段数分别为 20、60、100、140 和 180。其余的实验条件与第一组实验相同。实验结果如表 3 所示。表中所列的交叉率和变异率也只是通过节点复杂度算子计算得出的初始值。

表 3　弧段数对改进的自适应遗传算法的影响

弧段数	初始交叉率	初始变异率	运行时间/s
20	0.607 2	0.005 1	0.046 98
60	0.647 7	0.005 6	0.046 77
100	0.800 0	0.007 5	0.045 40
140	0.952 3	0.009 4	0.044 78
180	0.992 8	0.009 9	0.044 12

3)测试结果分析

由表 2、3 的结果可知,本文设计的优化自适应遗传算法中的节点复杂度算子与游戏地图的节点数和弧段数有关。节点数一定,弧段数越多,交叉率和变异率的初始值也就越高;弧段数一定,节点数越多,交叉率和变异率的初始值会越低。交叉率加大,搜索最短路径的搜索速度会加快;变异率升高,就越不会陷入局部最优解,反之亦然。对越来越复杂的游戏地图而言,本算法能有效地确保其搜索成功率并改善搜索速度。

3.3　遗传算法和 A* 算法应用实例对比

目前,在游戏设计中已广泛采用 A* 算法进行最短路径搜索。虽然 A* 算法搜索速度比较快,但是由于 A* 算法不能回溯,搜索成功率不是很高。而本文提供的优化自适应遗传算法既能满足游戏设计中较高的搜索成功率要求,又能确保玩家的实时响应要求(即较高的搜索速度)。图 5 和 6 分别是用 A* 算法和本文优化的自适应遗传算法所实现的对 S 与 T 两点间的最短路径搜索实例,可见,后者得到的路径更加优化,而游戏者基本感觉不到响应时间的差别。

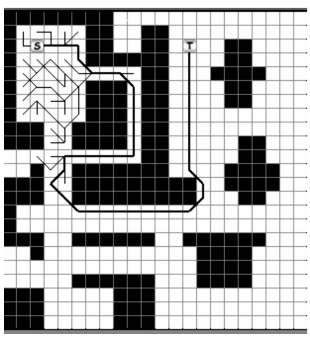

图 5　A* 算法实现最短路径搜索

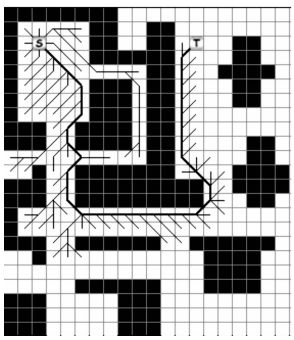

图 6　本文改进的遗传算法实现最短路径搜索

4　结束语

本文基于自然界环境对生物体的影响因素,结合游戏地图的特点设计了基于 Sigmoid 函数节点复杂度算子的优化自适应遗传算法。通过对数值仿真实验结果的分析,表明该算法在进行游戏地图最短路径搜索时,能够在确保搜索成功率的基础上提高搜索速度。本文提供的算法可在游戏引擎设计时予以实现,以满足对搜索成功率要求较高的具体游戏功能设计,具有良好的实用价值。

参考文献:

[1] GEN M, CHENG Run-wei, WANG Ding-wei. Genetic algorithms for solving shortest path problems[C]//Proc of IEEE International Conference on Evolutionary Computation. New York: IEEE Press, 1997: 401-406.

[2] AHN C W, RAMAKRISHNA R S. A genetic algorithm for shortest path routing problem and the sizing of population[J]. IEEE Trans on Evolutionary Computation, 2002, 6(6):566-579.

[3] 马超. 遗传算法和 Dijkstra 算法在动态权值系统中的比较[J]. 计算机技术与发展, 2012,22(9):21-24.

[4] JUHA M V. Game engines in game programming education: experiences from use of the CAGE game engine[C]//Proc of the 11th Koli Calling International Conference on Computing Education Research. New York: ACM Press, 2011:118-119.

[5] 庄健,杨清宇,杜海峰,等. 一种高效的复杂系统遗传算法[J]. 软件学报, 2010,21(11):2790-2801.

[6] SRINIVAS M, PATNAIK L M. Adaptive probability of crossover and mutation in genetic algorithms[J]. IEEE Trans on Systems, Man and Cybernetics, 1994, 24(4):656-677.

[7] PHILLIPA A, SUSHIL L. Coevolving influence maps for spatial team tactics in a RTS game[C]//Proc of the 12th Annual Conference on Genetic and Evolutionary Computation. New York: IEEE Press, 2010:783-790

[8] 王蕾,沈庭芝,招扬. 一种改进的自适应遗传算法[J]. 系统工程与电子技术, 2002:24(5):75-78.

[9] HOLLAND J H. Adaptation in natural artificial systems [M]. Cambridge:MIT Press, 1975.

[10] MENNON A, MEHROTRA K, MOHAN C K. Characterization of a class of sigmoid functions with applications to neural networks[J]. Neural Networks, 1996,9(5): 819-835.

[11] 韩瑞峰. 遗传算法原理与应用实例[M]. 北京:兵器工业出版社, 2010.

(上接第 75 页)

[22] OLUFSEN M S, WHITTINGTON M A, CAMPERI M, *et al*. New roles for the Gamma rhythm: population tuning and preprocessing for the beta rhythm [J]. Computational Neuroscience, 2003, 14(1):33-54.

[23] BORGERS C, EPSTEIN S, KOPELL N. Background Gamma rhythmicity and attention in cortical local circuits: a computational study [J]. Proceedings of the National Academy of Sciences, 2005, 102(19):7002-7007.

[24] YU Kai, WANG Jiang, DENG Bin, *et al*. Synchronization of neuron population subject to steady DC electric field induced by magnetic stimulation[J]. Cognitive Neurodynamics, 2013, 7(3):237-252.

[25] RADMAN T, RAMOS R L, BRUMBERG J C, *et al*. Role of cortical cell type and morphology in sub and suprathreshold uniform electric field stimulation [J]. Brain Stimulation, 2009, 2(4):215-228.

[26] BEDARD C, KROGER H, DESTEXHE A. Model of low-pass filtering of local field potentials in brain tissue [J]. Physical Review E, 2006, 73(1):051911.

[27] GIANNI M, LIBERTI M, APOLLONIO F, *et al*. Modeling electromagnetic fields detectability in a HH-like neuronal system: stochastic resonance and window behavior [J]. Biological Cybernetics, 2006, 94(2):118-127.

[28] PERC M. Spatial decoherence induced by small-world connectivity in excitable media [J]. New Journal of Physics, 2005, 7(1):252.

[29] STAM C J, JONES B F, NOLTE G, *et al*. Small-world networks and functional connectivity in Alzheimer's disease [J]. Cerebral Cortex, 2007, 17(1):92-99.

[30] YENER G G, BASAR E. Sensory evoked and event related oscillations in Alzheimer's disease: a short review [J]. Cognitive Neurodynamics, 2010, 4(4):263-274.