游戏场景中基于势场的交互寻路方法

余 帅[1] 李 艳[1] 王熙照[1] 赵鹤玲[2]
(河北大学数学与计算机学院机器学习与计算智能重点实验室 保定 071002)[1]
(中国人民解放军95866部队基础教研室 保定 071051)[2]

摘 要 在即时战略游戏中,路径规划是一种重要且常见的任务。游戏的实时性要求玩家能够快速寻找一条进攻的路径,而且游戏单元之间普遍存在的交互作用对寻路质量有着重要的影响。传统的寻路算法如Dijkstra算法虽然能够找到最优路径,但是耗时较多,而且未体现真实游戏中的交互。为此选取RTS游戏中一种典型的攻防场景,提出基于人工势场的快速高效动态寻路方法,同时为了体现RTS中游戏单元之间的交互性,将模糊测度引入到势场寻路中。实验结果表明,采用人工势场法寻路较Dijkstra算法耗时少、路径平滑;而引入模糊测度体现了真实游戏中单元之间的交互影响作用,与真实的游戏场景更为接近。

关键词 即时战略游戏,Dijkstra算法,A* 算法,人工势场法,模糊测度,模糊积分
中图法分类号 TP181 **文献标识码** A

Interactive Path-planning Method Based on Artificial Potential Field in Game Scenarios

YU Shuai[1] LI Yan[1] WANG Xi-zhao[1] ZHAO He-ling[2]
(Key Lab. in Machine Learning and Computational Intelligence, College of Mathematics and Computer Science, Hebei University, Baoding 071002, China)[1]
(Teaching and Research Section of Basic Theory, Troop 95866, Chinese People's Liberation Army, Baoding 071051, China)[2]

Abstract In real-time strategy (RTS) games, path planning is one of the typical and important tasks for game players. To meet the requirement of real-time response, the game players need to find an offensive path quickly. Besides, there are often interactions among game units which will greatly influence the quality of path planning. Dijkstra algorithm is a traditional and widely used algorithm which can find an optimal path. However, this algorithm cannot meet the strict time limit in RTS games and does not consider the unit interactions. This paper selected a typical RTS game attack-defense scenario, and presented a fast and dynamic path-planning method based on artificial potential field. We also introduced the concept of fuzzy measure to describe the interaction of units. The experiment results show that the proposed method is more efficient and makes the selected game scenario closer to the real games.

Keywords Real-time strategy game, Dijkstra algorithm, A* algorithm, Artificial potential field, Fuzzy measure, Fuzzy integral

1 引言

即时战略游戏(Real-time Strategy game,简称RTS game)是一种含有资源采集、基地建造、科技发展等元素的即时进行的战略游戏。RTS游戏具有实时性和交互性的特点,玩家在游戏中的谋略和战斗都是即时进行的,交互性体现在多个单元组合的效果不一定等于单个效果的线性累加。在RTS游戏中,非玩家角色(Non-Player Character,简称NPC)是不可缺少的重要组成部分之一,是指一切游戏中不受玩家控制的角色,也是玩家们在游戏中打交道最多的对象。如何产生更加智能化的NPC一直是计算机游戏中的重要课题。

为了设计能够自学习的NPC和提高在线过程的效率,刘天白等[1]将机器学习中的BP神经网络和支持向量机技术运用到游戏场景中,增加了游戏的趣味性,同时赋予了NPC学习和推理的能力。杨佩等[2]将Agent技术应用到第一人称射击游戏中,设计了HUNTBot作为游戏中的NPC,而具有混合式结构的Agent能适应动态、复杂、实时的游戏场景,使得NPC的智能行为与玩家更加接近。Ben Niu等[3]选取了代表性攻防场景,建立了基于遗传算法与神经网络的学习模型,从而提高了即时战略游戏的路径搜索效率,优化了游戏中随机地图的寻路方案。之后Ben Niu等[4]针对之前算法的缺陷,将粒子群算法引入到神经进化模型中,提高了游戏处理速度,

到稿日期:2013-05-20 返修日期:2013-08-06 本文受国家自然科学基金项目(60903088, 61170040),河北省自然科学基金(F2012201023),河北省第二批百名优秀人才支持计划(CPRC002)资助。
余 帅(1988-),男,硕士生,主要研究方向为机器学习、计算智能,E-mail:yushuai_2006@sina.com;李 艳(1976-),女,博士,副教授,硕士生导师,CCF会员,主要研究方向为机器学习、Rough集理论、计算智能;王熙照(1963-),男,教授,博士生导师,主要研究方向为机器学习、计算智能。

也更加符合游戏设计要求。佟晓磊[5]将遗传算法、蚁群算法、神经网络相结合，为解决即时战略游戏中武器资源的分配和防守位置的分布提供了自适应的快速学习方法。Peter H. F. Ng 等[6]将势场和模糊积分相结合，在 RTS 游戏的单元优化配置中考虑单元之间的交互作用，并将该方法运用到 Warcraft III 游戏中同原 AI bot 进行比较，使 NPC 获得了较高的决策精确性。文献[7]建立了一种基于极端学习机(Extreme Learning Machine)的单元组合评估模型，它具有良好的泛化能力和快速的在线反应。

路径规划是 RTS 游戏中常见和重要的任务之一，NPC 玩家需要快速高效地寻找一条从起点到达目标点的路径。目前关于路径规划有大量的相关研究，但大部分没有考虑交互性，且不能满足实时性需求。早在 1968 年，P. E. Hart 提出了一种基于启发式搜索的 A* 算法[8]，用于最优路径求解和一些策略设计中。Dijkstra 算法[9]是一种典型的最短路径算法，用于计算图中某个特定顶点到其他所有顶点的最短路径。Khatib[10]提出了基于抽象人造引力场的人工势场法，该方法结构简单，寻路耗时小，在游戏实时避障、平滑轨迹控制以及机器人寻路方面得到了广泛的应用。2008 年，Johan Hagelböck[11]将 Multi-agent 势场运用到即时战略游戏中，该方法能有效处理复杂的动态的游戏场景，解决资源采集、基地建设等问题。A* 算法在小地图中的搜索效率较高，但在地图较大时搜索效率较低，而 Dijkstra 算法虽然能寻找最优路径，但是耗时较大，尤其不能对动态地形进行快速路径规划。

本文选取 RTS 游戏中一种典型的攻防场景，提出基于人工势场的快速高效寻路方法来代替传统 A* 和 Dijkstra 算法，同时为了体现 RTS 中游戏单元之间的交互性，将模糊测度引入到势场寻路中。实验结果表明，势场寻路方法与 Dijkstra 算法相比耗时较小，寻路效率明显提高，寻得的路径较平滑；采用模糊测度的交互势场寻路时，单元之间的交互值越大，对寻路质量的影响也越大。

2 游戏中场景任务

通过对多种游戏场景的对比分析，我们抽取了具有基本 RTS 游戏要素的攻防场景。具体的任务如下：

(1)在平台的攻防两端有两支队伍，一支作为进攻方，一支作为防守方。攻防场景的中间区域是障碍物腹地，为攻防双方不可通过区域。

(2)对守方来说，需要在非障碍物区域布置 N 个火力防守点，在攻方到达其攻击范围内对攻方进行射击，在攻方到达防守方腹地前消灭攻方。

(3)对攻方来说，需要在非障碍物区域寻找一条攻击线路以快速到达守方防守腹地。在保证不被消灭的情况下，寻路的速度越快，则效果越好。

本文从攻方的角度考虑，在守方动态变换火力位置的情况下，为其提供一种基于势场的快速寻路方法，且引入模糊测度和积分来体现游戏单元之间的交互。在模拟的游戏场景中，用 Dijkstra 和势场算法分别进行攻击路线的寻找，并对两种方法消耗的时间和攻方遭受的伤害值进行比较。另外，采用不同的模糊测度来体现不同交互值对寻路的影响程度，使得场景任务更加接近真实游戏。

3 基于势场和模糊测度的交互寻路方法

人工势场法是由 Khatib 提出的一种机器人路径规划算法[10]。该算法将目标和障碍物分别看作对机器人有引力和斥力的物体，机器人沿引力与斥力的合力来进行运动，引入该方法是为了提高动态环境下的寻路效率。另一方面，模糊测度的引入则是为了体现游戏单元之间的交互作用。经典测度具有可加性，然而可加性在许多含有交互因素的实际情况中无法得到满足。日本学者 Sugeno 首次提出用比较弱的单调性代替可加性的一类集函数，称之为模糊测度[12]。

本文采用了文献[3,5]中的场景，但其主要工作是守方火力点位置的优化，攻方寻路采用 Dijkstra 算法进行，每次寻路完成后用遗传算法进行火力位置的迭代，最后用神经网络方法进行加速。Dijkstra 算法能寻找到最优路径，攻方寻路过程中遭受伤害值也相对较小，但该算法寻路耗费时间较多，而且遗传算法每次迭代都需要寻路，不能较好地满足 RTS 游戏的实时性；而防守方在摆放火力的过程中，只是将各个大炮对攻方的伤害进行简单叠加，并未考虑真实游戏中各单元之间的交互影响作用。

3.1 人工势场法的引入

人工势场法基本模型如下[13]：

设：$Uatt(X)$为引力场，为目标点对机器人的吸引作用；$Urep(X)$为斥力场，表示障碍物对机器人的排斥作用；$U(X)$为总力场，是引力场 $Uatt(X)$与斥力场 $Urep(X)$的和。$Fatt(X)$表示引力，$Frep(X)$为斥力。

这里考虑真实的游戏场景，寻路的主体为攻方 NPC，目标点是守方的腹地，并希望成功躲避守方的火力。寻路过程中的障碍物即守方布置的火力点有多个，所以斥力有多个。此时的 $Frep(X)$为各个分斥力的矢量和，$F(X)$为总合力，代表引力 $Fatt(X)$和斥力 $Frep(X)$的矢量和。用 k 表示引力系数，η 表示斥力系数，k，η 在具体的实验中为相应的正比例位置增益常数。X，X_g，X_o 分别代表游戏中的攻方 NPC、目标以及障碍物在空间中的位置。$\rho(X,X_o)=|X-X_o|$表示 NPC 在空间的位置与障碍物之间的距离，$\rho(X,X_g)=|X-X_g|$表示攻方 NPC 在空间的位置与目标之间的距离。常数 ρ_0 代表障碍物的影响距离，需根据障碍物和目标点的具体情况而定。

引力势场函数为：

$$Uatt(X)=\frac{1}{2}k\rho^2(X,X_g) \tag{1}$$

斥力势场函数为：

$$Urep(X)=\begin{cases}\frac{1}{2}\eta\left[\frac{1}{\rho(X,X_o)}-\frac{1}{\rho_0}\right]^2, & \rho(X,X_o)\leqslant\rho_0\\ 0, & \rho(X,X_o)>\rho_0\end{cases} \tag{2}$$

总势场函数为：

$$U(X)=Uatt(X)+Urep(X) \tag{3}$$

根据力函数是势场函数的负梯度，我们可以得出 NPC 所受引力和斥力的计算公式。

引力公式为：

$$Fatt(X)=k\rho(X,X_g) \tag{4}$$

斥力公式为：

$$Frep(X)=\begin{cases}\eta\left[\frac{1}{\rho(X,X_o)}-\frac{1}{\rho_0}\right]\frac{1}{\rho^2(X,X_o)}, & \rho(X,X_o)\leqslant\rho_0\\ 0, & \rho(X,X_o)>\rho_0\end{cases} \tag{5}$$

合力为：

$$F(X)=Fatt(X)+Frep(X) \tag{6}$$

则攻方 NPC 在合力的作用下前进，如图 1 所示。

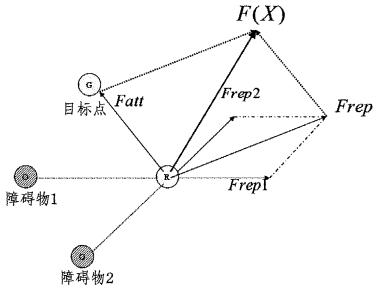

图 1　人工势场法基本模型

具体地，我们把守方的火力（如大炮）抽象为障碍物 1，地形障碍抽象为障碍物 2，它们都对攻方 NPC 起到排斥作用。防守方腹地抽象为目标点，也就是引力点的位置，对攻方 NPC 起到吸引的作用，势场法寻得的路径即为攻方 NPC 寻路的结果。引力斥力参数通过实验获取：引力系数 k 取 5，斥力系数 η 取 1，地形障碍的影响距离 ρ_0 取 2，火力障碍的影响距离为大炮的杀伤半径。

3.2　模糊测度和模糊积分的引入

RTS 游戏中，单元之间的交互影响主要体现在多个单元的组合效果不一定等于单个效果的线性累加，称为非可加性。例如防守方多个兵种对进攻方的伤害值大于或者小于单个兵种独立对进攻方的伤害值之和。本文中的单元交互影响体现在：防守单元对进攻单元的组合杀伤效果相比独立单元的杀伤效果有加强或者削弱，防守方总的防守效率大于或者小于单个防守单元防守效率的累加和。而现有的势场方法都是对独立障碍产生的斥力进行线性叠加，这势必会影响寻路的质量，可能造成攻方不能很好地避开组合火力较大的位置，而遭受较大的伤害值。模糊测度和模糊积分[12]的主要特征是非可加性，是描述因素之间交互作用的不可替代的数学工具。游戏单元之间的加强和削弱可以用模糊测度的超可加性和次可加性来表示。

设 X 为非空集，Γ 为 X 的幂集，μ 为 Γ 上的模糊测度。模糊测度的非可加性体现为：

超可加性：

$$\mu(A\cup B)>\mu(A)+\mu(B),A\cap B=\emptyset \tag{7}$$

次可加性：

$$\mu(A\cup B)<\mu(A)+\mu(B),A\cap B=\emptyset \tag{8}$$

如果 X 中含有 n 个元素，则需要确定 2^n-1 个参数才能定义 Γ 上的模糊测度。即需要给出$[\mu(x_1),\cdots,\mu(x_n),\mu(x_1,x_2),\mu(x_1,x_n),\cdots,\mu(x_1,x_2,x_3,\cdots,x_n)]$的值。这些值一般由专家给出，也可以通过学习得到，但所需工作量都很大。本文中，我们采用 λ-模糊测度，只需设定 n 个单点集上的 λ-模糊测度来体现防守单元间的交互作用。

λ-模糊测度定义如下：

模糊测度 μ 满足下面条件时称为 λ-模糊测度。存在常数 $\lambda,\lambda>-1$，使得

$$\mu(A\cup B)=\mu(A)+\mu(B)+\lambda\mu(A)\mu(B) \tag{9}$$

其中，$A\in\Gamma,B\in\Gamma,A\cap B=\emptyset$。对 $\forall S\subseteq X,\mu(S)\subseteq X$ 可以解释为属性集 S 的权重或者重要程度。$\lambda=0$ 说明属性间相互独立；$-1<\lambda<0$ 说明属性间存在冗余关联作用；$\lambda>0$ 说明属性间存在补充关联作用。

在引入模糊测度体现单元间的交互影响后，我们采用最常见的 Choquet 模糊积分来求防守方对进攻方的总的伤害值。

设 $f:X\to[0,+\infty),\mu$ 是定义在 X 上的一个 σ-代数 Γ 上的模糊测度，f 关于 μ 的 Choquet 模糊积分为：

$$(c)\int f\mathrm{d}\mu=\int_0^{\infty}\mu(F_\alpha)\mathrm{d}\alpha \tag{10}$$

其中，$F_\alpha=\{x\mid f(x)\geqslant\alpha,x\in X\}$。

当 X 是一个有限集合，记为 $X=\{x_1,x_2,\cdots,x_n\}$，且 $f:X\to[0,1]$时，Choquet 积分的计算公式相应地变为[14]：

$$(c)\int f(x)\circ\mu(X)=\sum_{i=1}^{n}(\alpha_i-\alpha_{i-1})\cdot\mu(x\mid f(x)\geqslant\alpha_i) \tag{11}$$

其中，α_i 为对应的 x_i 分类比例的值，$\alpha_0=0$ 且 $\alpha_1\leqslant\alpha_2\leqslant\cdots\leqslant\alpha_n$。

在考虑不同火力（大炮）之间的交互作用时，我们抽象出 $n=3$ 种不同杀伤力的大炮 A、B、C，数量各为 5，即火力障碍物总数为 15。采用 λ-模糊测度来计算三者之间的交互程度，并将其运用到势场寻路中去。不同的交互程度对我们势场法寻路的影响不同。实验中攻方所受伤害值总和通过模糊积分求出，交互值越大，伤害值越大。

4　实验与分析

仿真实验中采用的机器平台配置为：CPU 主频为 3.09GHz，2.98GB 内存，操作系统为 windows XP。具体的仿真工具采用的是 Matlab R2010b。

我们共做了 3 组实验。一组是基于 Dijkstra 算法寻路和人工势场法寻路的效率及伤害值的比较；一组是基于 A* 算法和人工势场法寻路的结果的比较；另一组是人工势场法寻路中采用不同模糊测度值对势场寻路的影响的比较。

4.1　Dijkstra 算法寻路和人工势场法寻路比较

选取 50 * 50 像素的仿真图来模拟攻防双方的场景。攻方初始血量为 100hp，大炮攻击半径为 4，攻方移动速度为 0.5，大炮杀伤效率为 1hp/s，遗传算法的评估函数为攻方寻路过程中遭受的伤害值。图 2 是采用 Dijkstra 算法和本文的人工势场法寻路的结果。

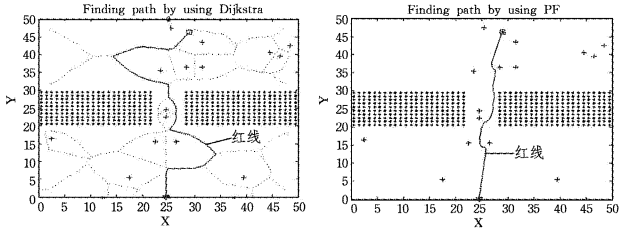

图 2　Dijkstra 算法寻路（左图）和人工势场法寻路（右图）

图中，红线部分为攻击方所选取的攻击线路，蓝色“+”表示防守方的防御位置，也就是防守方 15 门大炮的摆放位置，中间黑色点表示的区域为障碍物，也就是攻击方需要绕过的区域。地图底端有 3 个随机初始点，攻击方能够随机从这 3 个点开始进攻，顶部圆圈代表攻击腹地，攻击方必须从攻击方寻找路径到达腹地完成进攻任务。而防守方的主要任务则是通过武器的摆放试图在攻方到达腹地前消灭攻方。此过程也就是我们模拟 RTS 游戏攻防回合的仿真模式。

从图 2 中可以看出，势场法寻得的路径较 Dijkstra 算法寻得的路径更为平滑，NPC 只有在进入守方的火力及地形障碍的影响距离范围内才受到斥力影响。由于势场法在参数既定的情况下寻得的路径只与障碍物及目标点的位置有关，因此该方法在动态的地形环境中能被广泛采用。

为了比较两种算法的寻路效率和质量，我们在一个给定的地图中，通过随机生成 100 门大炮的摆放位置来求得 100 条用 Dijkstra 算法寻得的路径和人工势场法寻得的路径。实验结果如表 1 所列。

表 1　Dijkstra 算法和势场算法寻路受到的伤害值和时间耗费

组数	Dijkstra		PF	
	伤害值	时间(秒)	伤害值	时间(秒)
1	55	0.0981	84	0.0017
2	46	0.0921	58	0.0017
3	43	0.0912	36	0.0027
4	50	0.0888	28	0.0041
5	63	0.0862	91	0.0028
		……		
96	45	0.0917	62	0.0019
97	61	0.0956	38	0.0017
98	59	0.1007	55	0.0018
99	51	0.0939	50	0.0034
100	58	0.0946	77	0.0016

表 2 为 100 组实验的平均耗费时间和伤害值。可以发现，势场算法的平均时间大约为 Dijkstra 算法的 1/53，平均伤害值约为 Dijkstra 算法的 1.18 倍。势场算法由于只需要给定的障碍物坐标和起始点坐标就可求出路径，因此所耗时间较短。而 Dijkstra 是寻求最优路径的算法，时间耗费较大，但因为找到的是最优路径，所以其寻路过程中所受伤害值较小。势场法中，攻方寻路过程中只有攻方在障碍物的影响距离内才计算斥力，而此时攻方 NPC 有可能已进入火力攻击范围，所以势场法在寻路中遭受的伤害值较 Dijkstra 较高。总体来看，当游戏实时性要求非常严格时，采用势场寻路的效率有明显优势。

表 2　Dijkstra 算法和势场算法寻路平均伤害值和寻路时间比较

平均时间(秒)			
Dijkstra	PF	时间比例	100 次迭代所需时间
0.094877	0.001796	52.82684(1/53)	
平均伤害值			
Dijkstra	PF	伤害比例	148s
52.63	62.14	0.846958(1.18/1)	

4.2　A* 算法寻路和人工势场法寻路比较

选取 50 * 50 像素的仿真图来模拟攻防双方的场景。在势场寻路中，防守单元的数目为 40 个，攻方移动速度为 0.5，故而将 A* 算法的地图设置为 100 * 100，每个单元抽象为 0.5 * 0.5 大小的小格，A* 算法里移动一步代表移动 0.5 的距离。图 3 为 A* 算法寻路和人工势场法寻路的结果。

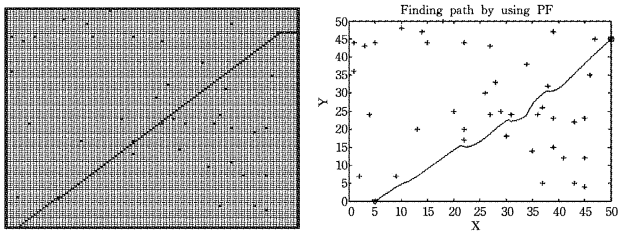

图 3　A* 算法寻路(左图)和人工势场法寻路(右图)

为了比较两种算法的寻路效率和质量，我们在一个给定的地图中，通过随机生成 50 门大炮的摆放位置来求得 50 条用 A* 算法寻得的路径和人工势场法寻得的路径。实验结果如表 3 所列。

表 4 为 50 组实验的平均耗费时间和伤害值。可以发现，人工势场算法的平均时间大约为 A* 算法的 0.38 倍，平均伤害值约为 A* 算法的 1.12 倍。A* 算法是一种静态路网中求解最短路最有效的方法，在障碍比较少、地图比较简单的情况下寻路的效率比较高，但在障碍物多、转角较多、构造较为复杂的地图中，基于启发式的 A* 算法寻路容易受到地形环境的限制，此时 A* 会扩展大量的结点，从而消耗较长时间。而人工势场法在动态的障碍物较多的环境中也能快速高效地规划路径，因而在复杂的大型游戏中人工势场法具有优势。

表 3　A* 算法和势场算法寻路受到的伤害值和时间耗费

组数	A*		PF	
	伤害值	时间(秒)	伤害值	时间(秒)
1	108	0.0042	141	0.0020
2	114	0.0043	120	0.0018
		……		
49	86	0.0057	91	0.0023
50	110	0.0075	113	0.0026

表 4　A* 算法和势场算法寻路比较

平均时间(秒)		
A*	PF	时间比例
0.0056	0.0021	2.67∶1(1∶0.375)
平均伤害值		
A*	PF	伤害比例
98.4	110.6	0.8897∶1(1∶1.12)

4.3　势场法中采用不同模糊积分模式的比较

我们单独在一张模拟地图上摆放 3 个种类的大炮 A(蓝色“+”)、B(绿色“x”)、C(红色“*”)，每种大炮为 5 门，杀伤力分别为 2、3、4，杀伤半径均为 2，攻方运行的速度为 1，势场的其他参数与前面相同。基于第 3 节提出的势场寻路方法，在游戏环境中找到一条路径，如图 4 所示。对这样一条固定的路径，下面通过不同的模糊测度来模拟游戏中单元之间的不同的交互影响。

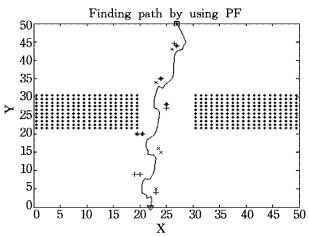

图 4　存在交互作用下的势场寻路

这里，我们通过给出不同的 λ 值(>0，$=0$，<0)定义 3 种不同的模糊测度，即超可加、可加和次可加测度，来表示火力单位之间的相互加强、无交互和相互减弱 3 种模式。图 3 中的路径是不考虑交互的情况下得到的，即在 $\lambda=0$ 时的情况。但如果单元间存在相互影响，那么攻方 NPC 仍然走这条路径时受到的伤害值会发生变化。表 5 是令 $\lambda=0.3$，0，-0.3 时，利用 Choquet 积分计算出的相应测度下攻方所受的伤害值。

从实验结果可以看出，火力单元间的交互作用对于寻路质量有着明显的影响。如果单元间存在超可加的交互作用，而攻方 NPC 仍然采用原来的攻击路线，那么攻方会因没有避

开较强的组合火力而受到较大的伤害值。一般来说，模糊测度越大，交互效果越明显，大炮之间的攻击效率较无交互的越高，攻方在寻路中遭受的伤害较原来直接叠加的伤害也越大。

表5 不同模糊测度值结果比较

组别	超可加	可加	次可加
模糊测度	λ=0.3	λ=0	λ=-0.3
	μ(A)=2	μ(A)=2	μ(A)=2
	μ(B)=3	μ(B)=3	μ(B)=3
	μ(C)=4	μ(C)=4	μ(C)=4
	μ(A∪B)=6.8	μ(A∪B)=5	μ(A∪B)=3.2
	μ(A∪C)=8.4	μ(A∪C)=6	μ(A∪C)=3.6
	μ(B∪C)=10.6	μ(B∪C)=7	μ(B∪C)=3.4
	μ(A∪B∪C)=16.8	μ(A∪B∪C)=9	μ(A∪B∪C)=1.2
伤害值	293	218	143

结束语 本文选取RTS游戏中一种典型的攻防场景，从攻方的角度考虑，在守方动态变换火力位置的情况下，对比人工势场法寻路、A* 算法和Dijkstra算法寻路的耗时和攻方所受伤害值，并在势场法寻路中引入模糊测度，考虑到防守单元之间的交互作用，计算不同模糊测度下势场寻路的伤害值。从实验结果看，Dijkstra算法寻得最短路径，寻路所受伤害值相对较小，但寻路所耗时间较大；A* 算法在障碍物较少、转角少的地图中寻路较快，但不适用于复杂的多障碍的地图；人工势场法寻路较Dijkstra算法寻路效率高，时间明显缩短，寻得的路径较平滑，但势场法只考虑障碍点坐标，只有进入大炮射击范围内才计算斥力，因此在寻路过程中遭受的伤害值比前者稍高。在势场法寻路中引入模糊测度，模糊测度值越大，伤害值越高，因此势场法寻路与真实的游戏场景更为接近。未来的工作是将单元之间的交互作用直接在势场的斥力公式中表示出来，以在各种交互模式中高效地完成动态寻路任务。

参 考 文 献

[1] 刘天白，张舒白. 机器学习技术在游戏中的应用研究——解决鼠标轨迹识别问题[J]. 电脑知识与技术，2011，7(13)：3100-3102

[2] 杨佩，王皓，罗文杰. 等. HUNTBot—第一人称射击游戏中NPC的结构设计[J]. 计算机科学，2008，35(11)：290-292

[3] Huo P，Shiu S C-K，Wang H，et al. A neural evolutionary model for case-based planning in real time strategy games[C]// Next-Generation Applied Intelligence 2009. Berlin Heidelberg，2009：291-300

[4] Huo P，Shiu S C-K，Wang H，et al. Application and comparison of particle swarm optimization and genetic algorithm in strategy defense games[C]// Fifth International Conference on Natural Computation. IEEE，2009：387-392

[5] Tong Xiao-lei，Li Yan，Li Wen-liang. Optimization Methods For Resources Allocation In Real-Time Strategy Games[C]// Proceedings of the 2011 International Conference on Machine Learning and Cybernetics. Guilin，2011，2：507-513

[6] Peter H F N，Li Y J，Shiu S C K. Unit Formation Planning in RTS game by using Potential Field and Fuzzy Integral[C]// 2011 IEEE International Conference on Fuzzy Systems (FUZZ-Y). Taipei，2011：178-184

[7] Li Ying-jie，Li Yan，Simon C K S，et al. RTS game strategy evaluation using extreme learning machine[J]. Soft Computing，2012，16(9)：1627-1637

[8] Hart P E，Nilsson N J，Raphael B. A formal basis for the heuristic determination of minimum cost paths in graphs[J]. IEEE Trans. Syst. Sci. and Cybernetics，SSC，1968，4(2)：100-107

[9] Dijkstra E W. A note on two problems in connexion with graphs[J]. Numerische Mathematik，1959，1：269-271

[10] Khatib O. Real-Time Obstacle Avoidance For Manipulators and Mobile Robots[J]. Int. J. Robotics Res，1986，5(1)：90-98

[11] Hagelback J，Johansson S. Using multi-agent potential fields in real-time strategy games[C]// Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems. 2008：631-638

[12] Sugeno M. Theory of fuzzy integrals and its applications[D]. Tokyo：Tokyo Institute of Technology，1974

[13] 胡志华. 基于离散人工势场的移动机器人动态路径规划研究[D]. 长沙：中南大学，2009

[14] Peters J，Han L，Ramanna S. The Choquet integral in a rough software cost decision system[M]// Fuzzy Measures and Integrals：Theory and Applications，Studies in fuzziness and soft computing，1999

(上接第130页)

[4] Dean J，Ghemawat S. MapReduce：simplified data processing on large clusters[J]. Communications of the ACM，2008，51(1)：107-113

[5] Chang E Y. Foundations of Large-Scale Multimedia Information Management and Retrieval[M]. Berlin，Heidelberg：Springer，2011：213-230

[6] Zhao J，Liang Z，Yang Y. Parallelized incremental support vector machines based on MapReduce and Bagging technique[C]// Proceedings of the International Conference on Information Science and Technology，2012. New York：IEEE，2012：297-301

[7] Zhao Z. Parallel Pegasos for Mahout [OL]. http://wenku.baidu.com/view/0e8d6d639b6648d7c1c74661.html，2012

[8] Lanckriet G R，Ghaoui L E，Bhattacharyya C，et al. A Robust Minimax Approach to Classification[J]. Journal of Machine Learning Research，2002，3(3)：555-582

[9] Huang K，Yang H，King I，et al. Learning large margin classifiers locally and globally[C]// Proceedings of the twenty-first international conference on Machine learning，2004. New York：ACM，2004：51

[10] Yeung D S，Wang D，Ng W W，et al. Structured Large Margin Machines：Sensitive to Data Distributions[J]. Machine Learning，2007，68：171-200

[11] Xue H，Chen S，Yang Q. Structural support vector machine[C]// Proceedings of the fifth International Symposium on Neural Networks，2008. Berlin，Heidelberg：Springer，2008：501-511

[12] Shalev-Shwartz S，Singer Y，Srebro N. Pegasos：Primal Estimated sub-GrAdient SOlver for SVM[C]// Proceedings of the 24th International Conference on Machine Learning，2007. New York：ACM，2007：807-814