RoboCup 中基于动作序列模型的动作决策*

丁晨阳
(扬州职业大学,江苏 扬州 225009)

摘 要: RoboCup 机器人足球比赛是当前人工智能领域的一个研究热点,其目的在于为多智能体系统提供一个标准的研究平台。为了让 RoboCup 仿真比赛中球员智能体实时地作出合理的动作决策,提出一种基于动作序列模型的决策机制,通过对球员智能体的动作空间分解、建立动作序列并对动作序列进行评价,而让智能体选择出当前最优的动作执行。仿真结果表明应用这种决策机制提高了智能体对环境的适应性。

关键词: 动作决策;RoboCup;动作序列模型;评价
中图分类号: TP242.6 **文献标识码**: A **文章编号**: 1674-7720(2013)07-0089-03

Action decision based on action sequence model in RoboCup

Ding Chenyang
(Yangzhou Polytechnic College, Yangzhou 225009, China)

Abstract: Robocup is a research hotspot in artificial intelligence field, which aims to provide a standard research platform for multiagent system. An action sequence model based decision mechanism is proposed in this paper for agents making reasonable action decision. By decomposing the action space, establishing action sequence and evaluating action sequence, current optimal action can be chosen to execute. Simulation results indicate that with application of applying this decision mechanism, environment adaptability of agents is enhanced.

Key words: action decision; RoboCup; action sequence model; evaluation

RoboCup 机器人足球世界杯赛是多智能体系统 MAS(Multi-Agent System)和分布式人工智能的一个重要研究平台。它提供了一个标准的多智能体环境,在这样一个动态的、连续的、不可预知的环境中,进行多智能体决策是一个难题。

RoboCup 仿真比赛中球员智能体快速地采取合理的行动选择对于球队的表现有着决定性的影响。作为整队策略的设计者,要做的工作就是决定智能体在比赛的每一个仿真周期选择什么动作发送给服务器。传统的方法是使用决策树[1],其动作选择过程分为两个步骤:首先确定当前的状态模式,然后根据相应的状态模式产生一个恰当的动作命令。这个方法的优点是实现简单,而且计算量小,反应比较快。然而要确定当前的状态模式,就必须在设计时将场上状态离散化成一个有限的集合,而在 RoboCup 中,由于智能体处于一个动态、连续的环境中,场上状态的变化是微妙的,这使得状态的离散化变得很难,而且由于状态的数量是趋于无穷的,在程序中无法将所有的状态都考虑到,所以这种方法的效果是有限的。虽然场上状态是连续、无限的,但球员可选择的动作却是有限且易于区分的。

本文提出动作序列模型将球员智能体的动作空间离散化,并对动作进行评价得到一个评价值,智能体根据评价值挑选出最优的动作执行。动作序列模型的设计主要在于动作空间的划分和动作序列的评价。合理的动作划分使得智能体具有更多、更合理的动作序列,在当前形势下能够进行更加周全的分析和决策。对动作序列的评价是动作序列模型设计中的关键环节,通过将智能体当前情况下执行某一动作能得到的收益进行量化来使得动作是可选的。通过对动作序列的离散化并进行评价,可以实现一个适应性强、易于扩展的整队策略。

*基金项目:扬州职业大学教研项目(编号:07202)

1 动作序列模型的设计

动作序列模型中所有动作都具有三个相同的操作：判断动作是否能被执行、评价函数和取得执行动作所需的服务器命令。为了提供一个统一的接口，可以抽取这些公有操作，设计一个抽象类 Action，定义如下：

```
class Action
{
virtual bool IsPreconditionReady()=0;
virtual estimateT getEstimateAfterAction()=0;
virtual SoccerCommand getCurrentCommand()=0;
}
```

其中 IsPreconditionReady()判断动作的前提条件是否满足，这是在评价动作前必须要做的，若前提条件不满足，比如无人可以传球，那评价动作也就没有意义了；getEstimateAfterAction()对执行动作的结果进行评价，返回评价值；getCurrentCommand()取得当前执行此动作所对应的命令。这三个操作应该按顺序调用。

由 Action 可派生出各个具体动作，如图 1 所示有传球 passBall、带球 dribble、射门 shoot 等，在这些具体动作中实现其父类的纯虚函数。

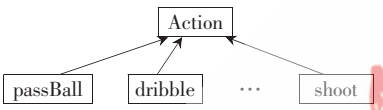


图 1 动作的类层次

1.1 动作空间的划分

本节描述的动作属于高层决策问题的一部分，是利用仿真比赛服务器提供的底层基本技能（dash、kick、turn 等）组合而成的高层复杂动作（passBall、dribble、shoot 等）。图 2 给出了个人基本技能以及动作序列在智能体结构中所处的层次。

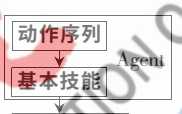


图 2 动作序列在智能体结构中所处的层次

动作序列模型中主要包括以下动作：(1) 传球 passBall：在传球者和各个可能的队友之间计算出一条合适的路线传球。(2) 带球 dribble：选择合适的带球角度以及速度移动球，并将球控制在一定距离内。(3) 射门 shoot：将球踢向对方球门线上的最佳射门点。(4) 过人 outplayOpponent：将球传到对方防守队员背面的位置。(5) 截球 intercept：当智能体是离球最快的队员时，依据截球周期、截球点等参数执行此动作。(6) 清球 clearBall：大力将球踢尽量远。(7) 跑位 moveToPos：根据协作需要搜索一个空闲位置移动过去。(8) 盯人 mark：选择盯人对象以及盯人位置，阻止对方截球或者接近球门。(9) 找球 searchBall：当球不在视觉范围内时，转身找球。

动作序列模型里的动作序列可以被逐渐扩充，在初期的设计时可以足够简单，易于测试和实现，而在后期可以通过添加动作将动作序列逐渐完善。

如果让智能体遍历所有动作挑选出最优，有可能会无法满足比赛实时性的要求，因此不同角色的球员对应的动作序列是不同的，决策时，球员智能体只需要考虑与自己相关的动作序列，例如一个充当守门员角色的智能体永远也不会进入对方的禁区，就不需要把这个状态所对应的动作考虑在内。这样按球员角色分配动作序列，可以缩小每个智能体动作序列的求解空间，智能体的行为选择越少，将越容易进行动作决策。

1.2 动作序列的评价

动作序列模型还要求对动作进行评价，依据评价值可给出各动作的优先级。序列中的某些动作被排除在评价的范围之外，这是由于它们具有比较明确的前提条件，一旦前提条件满足，则动作必须执行。比如截球动作，只要队员是最快到达球的位置就去截球。

对动作进行评价的最终结果是得到一个评价值，评价值的计算方法由式(1)给出，其中 estimate_value 是评价值，benefit 是执行动作所得到的效益，chance 是执行此次动作的成功率。

$$estimate\_value = benefit \times chance \tag{1}$$

1.2.1 效益值

效益值 benefit 是度量执行动作对实现多智能体系统的最终目标能产生多大效果的标准。在 RoboCup 中全队的最终目标是进球，所以执行动作后球越靠近对方球门，越靠近边路效益越大。为了同时体现 x 和 y 坐标在效益中的作用，同时减少效益值计算时产生的振荡，本文参考了 UvA Trilearn 曾提出过的分区的思想[2]，对球场进行了分区来辅助效益值的计算。球场被分为 7 个区，如图 3 所示。

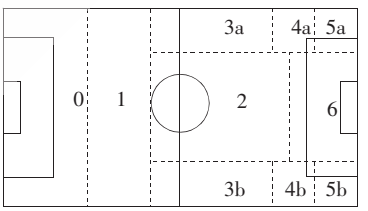


图 3 球场分区情况

area_0，area_1，area_2，area_3，area_4，area_5，area_6 分别表示这 7 个区，对每个区都规定了效益的最小值和最大值。分区效益的最大值定义如表 1 所示，当且仅当在分区的最大 x 坐标处取得最大值，0 区的最小值为 0.0，其余分区的效益最小值为其相邻上一个分区的最大值（area_6 除外）。

表 1 球场 7 个分区分别对应的效益最大值

区域类型	效益最大值
area_0	0.1
area_1	0.2
area_2	0.4
area_3	0.7
area_4	0.9
area_5	1.0
area_6	1.8

给定一个位置 pos(x,y),计算其效益值如式(2)和式(3)。

若 pos 处于 area_N,$N\in\{0,1,2,3,4,5\}$:

benefit=min_benefit_area_N+(max_benefit_area_N−min_benefit_area_N)×(pos.getX()−min_areaN_x)/(max_areaN_x−min_areaN_x) (2)

若 pos 处于 area_6:

benefit=2×min_benefit_area_6+(max_benefit_area_6−min_benefit_area_6)×(pos.getX()−MAX_AREA2_X)/(PITCH_LENGTH/2−MAX_AREA2_X) (3)

其中 getX(),getY()分别为取得 pos 的 x,y 坐标值的方法,由于 area_6 比较特殊,处于敌方球门的正前方,对我方十分有利,所以对它进行了特殊的处理。

1.2.2 成功率

成功率 chance 代表执行此动作成功的概率,对于不同的动作,成功率的计算是不一样的,下面以传球为例说明成功率的计算过程。

为计算传球成功率,使用了一个分布函数:$s(x)=1/(1+e^{-x})$。它是一个 S 形的函数,且 $s(x)\in[0,1]$;$s(0)=0.5$;当 $x<<0$ 时,$s(x)\sim0$;当 $x>>1$ 时,$s(x)\sim1$。所以,$s(x)$ 提供了一个在(0,1)上的分布。

如图 4 所示,球的当前位置 posBall 和传球点 posPassTo 的连线为传球路,对于此传球路线只考虑一个最危险的敌人,计算得到它的截球点 posInter,DistBall 为 posBall 到 posInter 的距离,DistOpp 为最危险的敌人到 posInter 的距离,则传球成功率的计算如式(4):

chance=s(1.5×DistOpp − DistBall) (4)

图 4 传球成功率的计算

按照 $s(x)$ 函数的分布,当 1.5×DistOpp=DistBall 时,chance=0;当 1.5×DistOpp<<DistBall 时,chance→0,即球通过 posInter 的几率很小;当 1.5×DistOpp>>DistBall 时,chance→1,即球通过 posInter 的几率很大。在速度一定的情况下,距离越远,需要的时间越长,由于球的速度通常都比球员的速度大,所以将 DistOpp 放大了 1.5 倍作为修正。

当效益值和成功率都被计算以后,动作的评价也就结束了,评价值就是这两者的乘积。

2 动作决策过程

动作序列模型提供动作序列以及对动作进行评价的方法,球员智能体在进行动作决策时通过扫描动作序列并比较其评价值来选出当前状态下的最优动作。

动作序列被分为两部分:可评价动作和非评价动作。对于可评价动作,只需要调用动作序列模型提供的接口取得评价值,比较评价值的大小即可得到当前的最优动作。对于非评价动作,需要为它们确定优先级,这以它们在序列中的位置来表示,即位置越靠前,优先级越高,高优先级的动作将首先被扫描,然后是较低优先级的动作。非评价动作的优先级是高于可评价动作的,即一旦有非评价动作前提条件满足,就立刻放弃对动作序列的扫描。整个决策流程如图 5 所示。

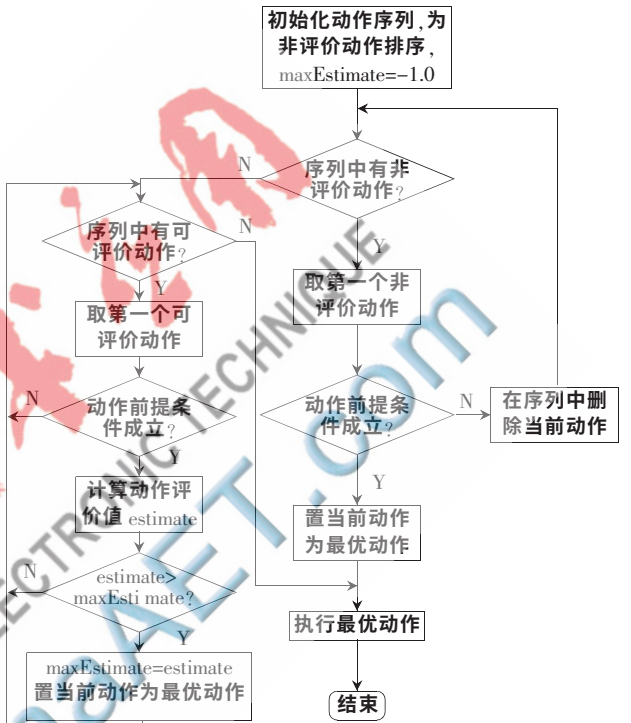


图 5 动作决策流程

动作序列模型根据球员的角色生成相应的动作序列,首先判断是否有非评价动作满足条件,如果有则执行此动作,不再继续判断;如果没有则对可评价函数序列进行扫描,调用每个动作的 IsPreconditionReady()计算动作执行的前提满足的可能性,如果可行,再使用 getEstimateAfterAction()方法取得动作的评价值,通过比较评价值的大小即可得到当前的最优动作。最后调用最优动作的 getCurrentCommand() 取得服务器命令发送给动作执行模块。

3 仿真结果

下面是应用基于动作序列模型进行动作决策的实例(如图 6 中的 a-d)。

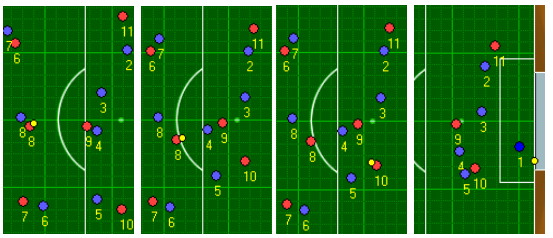
(1) 对方 3 号球员传球给对方 8 号球员的过程中,我方 8 号队员判断自己是能最快到达球的队员,因此选择执行截球动作,如图(a);

(2)截球后观察到适合协作传球的队友周围都有较危险敌人,成功率不高,而自己前方区域较空闲,因此选择执行带球;我方 8 号队员带球的同时,10 号队友跑位至

对方球员较少的区域为和 8 号队员协作做准备，如图(b)；

(3) 在 10 号队友执行跑位动作的形势下 8 号队员做出的动作选择就是传球给 10 号队友，将球推进至对方球门附近，如图(c)；

(4)10 号队员接到球后，由于前方其他的队友都被敌方球员盯紧，10 号队员的最佳策略就是选择一条路径射门，如图(d)。



(a) (b) (c) (d)

:我方球员 :对方球员 :球

图 6 基于动作序列模型进行决策的示意图

经过对球员在比赛中表现的观察，采用该动作决策机制的效果还是显而易见的。

在 Robocup 中，执行动作是智能体影响周围环境的唯一途径，因此智能体是否能挑选出当前周期的最优动作是 RoboCup 仿真球队设计的重点。

本文提出一种基于动作序列模型的决策机制，所构建的动作序列模型包括两个主要部分：建立动作序列和对动作序列进行评价。通过离散化动作空间预先定义好动作集合，在比赛时可以根据动作集合动态地生成动作序列；对动作的评价是算法的核心部分，评价值是高层决策对动作的重要标准。决策机制对评价值进行比较并挑选出最优动作，克服了传统的决策树策略人为设定优先级的不灵活的缺点，对环境具有较好的适应性。此外，这种决策机制简单而又清晰，其复杂性被封装在动作序列模型中，其扩展性也由动作序列中动作的添加而得到很好的支持。

参考文献

[1] KONUR S, FERREIN A, LAKEMEYER G. Learning Decision Trees for Action Selection in Soccer Agents[C]. In Proceedings of Workshop on Agents in Dynamic and Real-time Environments, 16th European Conference on Artificial Intelligence, Valencia, Spain, 2004.

[2] BOER R D, KOK J. The Incremental Development of a SyntheticMulti -Agent System: The UvA Trilearn 2001 Robotic Soccer Simula -tion Team [D]. Master's thesis, University of Amsterdam, 2002.

[3] 彭军，刘亚，吴敏等.基于状态预测的多智能体动态协作算法[J].系统仿真学报，2008，20(20).

[4] 王骋，王浩，方宝富.使用基于值规则的协作图实现多 agent 的动作选择[J].计算机工程与应用，2004，40(19)：61-65.

(收稿日期：2013-01-06)

作者简介：

丁晨阳，女，1981 年生，主要研究方向：人工智能与多智能体系统，无线网络。