

# 海空兵棋中基于行为树的智能决策方法

田 佩<sup>1,2</sup>, 臧兆祥<sup>1,2</sup>, 郭鸿村<sup>1,2</sup>, 张 震<sup>1,2</sup>(1. 三峡大学 水电工程智能视觉监测湖北省重点实验室;  
2. 三峡大学 计算机与信息学院, 湖北 宜昌 443002)

**摘要:** 在兵棋推演过程中, 面对策略的不确定性和决策情况的复杂性, 人类专家制定的策略往往需要经过计算推演的验证, 计算结果也可能会和预想结果大相径庭, 而智能决策辅助推演可以极大地提升推演过程中策略制定的准确性。针对传统有限状态机决策机制的不灵活, 模糊状态机又缺乏适应性的问题, 提出了一种基于行为树的智能决策方法来实现兵棋的实时决策过程, 并基于 Behavior3 Editor 平台实现行为树的策略生成过程, 最后将基于行为树的智能决策方法用于海空兵棋推演, 实现了作战策略的生成, 验证了该方法的可行性和有效性。

**关键词:** 作战决策; 行为树; 兵棋推演; Behavior3 Editor

中图分类号: TP391.9

文献标识码: A

文章编号: 2096-9759(2022)04-0122-05

## Intelligent Decision-Making Method based on Behavior Tree in Sea-Air Wargames

TIAN Pei<sup>1,2</sup>, ZANG Zhaoxiang<sup>1,2</sup>, GUO Hongcun<sup>1,2</sup>, ZHANG Zhen<sup>1,2</sup>(1. Hubei Key Laboratory of Intelligent Vision Based Monitoring for Hydropower Engineering, China Three Gorges University;  
2. College of Computer and Information Technology, China Three Gorges University, Yichang Hubei, 443002, China)

**Abstract:** In the process of wargaming, with the uncertainty of the strategy and the complexity of the decision-making situation, strategies formulated by human experts often need to be verified by computational deduction, and the calculation results may be very different from the expected results, intelligent decision-making can greatly improve the accuracy of strategy formulation during the wargaming process. To overcome the inflexibility of traditional finite state machine decision-making mechanism and the lack of adaptability of fuzzy state machine, an intelligent decision-making method based on behavior tree is proposed to realize the real-time decision-making in wargames, and the strategy of behavior tree is realized based on the platform of Behavior3 Editor. This method is applied to sea-air wargaming to give combat strategy, and the feasibility and effectiveness of the method have been verified.

**Key words:** Operational Decision; Behavior Tree; Wargaming; Behavior3 Editor

## 0 引言

兵棋推演最早起源于军事领域, 现在也已经成为军事教学的必备科目, 甚至已经被应用于军事领域之外的经济、政治等对抗活动之中。兵棋系统则作为推演计算的工具, 用来研究如何合理组织明确目标的作战活动, 帮助作战指挥人员快速进行决策分析和制定, 从而获取战争优势, 这对真实的军事部署行动具有重大意义<sup>[1]</sup>。

用行为树构建兵棋推演中的策略生成模块, 就是将人类专家的决策知识和方法用行为树表示出来。经典的兵棋推演技术主要运用统计学、概率论和博弈论等方法实现。刘满<sup>[2]</sup>等人借助棋子的历史位置概率, 使用多属性综合评估算法来决定棋子的未来位置。夏阳升<sup>[3]</sup>等人使用 0-1 整数规划的方法对决策过程进行建模, 应用于无人机侦察。邹焯翰<sup>[4]</sup>等人则对推演规则进行归纳总结, 借助运筹学对推演过程的复杂性和随机性进行研究, 认为能够提升作战推演系统的性能, 进而提升决策能力。

本文提出的基于行为树的智能决策方法, 主要用于实现海空兵棋推演平台中红、蓝双方的作战策略生成, 该方法首先对该兵棋推演平台中的对抗规则进行分析, 构建行为树的整体框架, 针对规则智能体的作战细节方案, 对行为树的决策方

式做出相应调整。实验结果表明, 基于行为树的智能决策方法能够有效地实现作战策略的生成。

## 1 行为树

### 1.1 行为树的概念与结构

行为树 (Behavior Tree, BT) 是一种包含逻辑节点和行为节点的树形层次结构<sup>[5-6]</sup>, 用一个二元组表示为:  $BT = \langle V, E \rangle$ , 二元中  $V$  是行为树的节点集合, 并且  $V = A \cup C \cup N \cup \tau$ ,  $A$  是行为节点的集合,  $C$  是条件节点的集合,  $N$  是逻辑节点的集合,  $\tau$  是行为树的根节点。在行为树的所有节点中, 只有行为节点和条件节点是叶子节点, 而逻辑节点是非叶子节点。 $E$  是行为树的边的集合, 当有任意的  $\langle v_i, v_j \rangle \in E$ , 且  $v_i, v_j \in V$ , 则可以称  $v_i$  是  $v_j$  的父节点,  $v_j$  是  $v_i$  的子节点。行为树中所有子节点的执行结果由该子节点的父节点管控<sup>[7]</sup>。当需要找到一个行为时, 从树的根节点出发, 遍历行为树的各个子节点, 然后找到第一个和当前数据相符合的节点予以执行。行为树最早被应用于游戏行业, 并且被广泛应用于大型游戏中的非玩家角色 (NPC) 的行为策略控制<sup>[8]</sup>。

行为树包含了不同的节点类型, 分别是行为节点 (Action Node)、条件节点 (Condition Node)、逻辑节点 (Logical Node) 以及根节点 (Root)。其中逻辑节点是非叶子节点, 在

收稿日期: 2022-03-29

基金项目: 国家自然科学基金 (No.61502274)、三峡大学水电工程智能视觉监测湖北省重点实验室开放基金 (No.2015KLA08) 资助项目。

作者简介: 田佩 (1995-), 女, 硕士研究生, 主要研究方向: 强化学习; 郭鸿村 (1996-), 男, 硕士研究生, 主要研究方向: 机器学习; 张震 (1995-), 男, 硕士研究生, 主要研究方向: 深度强化学习。

通讯作者: 臧兆祥 (1985-), 男, 副教授, 硕士生导师, 主要研究方向: 决策智能、计算机游戏智能。



其相关指令做了简单介绍,如表 1 和表 2 所示。

表 1 红方兵力设置

兵力	数量	初始位置	任务
轰炸机	16 架	机场	空中突击
预警机	1 架	红方岛屿附近空域	空海探测
无人机	3 架	红方半场靠近中线空域	空中侦察
干扰机	1 架	机场	干扰压制
歼击机	20 架	机场	掩护护航
护卫舰	2 艘	红方岛屿附近海面	舰艇防空
雷达	1 部	红方岛屿	对空探测
机场	1 个	/	支援保障

表 2 蓝方兵力设置

兵力	数量	初始位置	任务
轰炸机	8 架	南岛机场	空中突击
预警机	1 架	南岛附近空域	空海探测
雷达	1 部	每岛 1 部	对空探测
歼击机	12 架	南岛机场	空中拦截
护卫舰	1 艘	北岛附近海面	舰艇防空
地防车	3 部	北岛 1 部南岛 2 部	地面防空
机场	1 个	南部岛屿	支援保障
指挥所	2 个	每岛 1 个	保卫目标

兵棋推演平台中涉及了多种作战单位,它们的功能和作战方式都有一定的区别,对作战单位的相关属性说明介绍如表 3、表 4 和表 5 所示。

表 3 行动单位指令说明

兵力	行动指令
轰炸机	(起飞)目标突击、(起飞)区域突击、返航
预警机	(起飞)区域巡逻、(起飞)航线巡逻、返航
无人机	区域、航线巡逻、侦察任务取消、返航
干扰机	区域、航线干扰,关闭、结束干扰、返航
歼击机	拦截、(起飞)护航、返航
护卫舰	添加/移除指定目标
雷达	雷达开关机指令
地防车	添加/移除指定目标

表 4 空中武器性能装备参数信息

类型	速度 (km/h)	高度 (m)	探测/ 干扰角度 (度)	探测/ 干扰距离 (km)	攻击 距离 (km)	弹药数
轰炸机	600-800	[,18000]	120	100	80	2
歼击机	900-1000	[,18000]	120	100	80	6
预警机	600-800	[,10000]	360	250	/	/
干扰机	600-800	[,10000]	360	60	/	/
无人机	180-350	[,18000]	360	40	/	/

表 5 地面、水面武器装备性能参数信息

类型	速度 (km/h)	探测方位/ 距离	攻击方位/ 距离 (km)	弹药数	火力 通道
雷达	0	360 度/180km	/	/	/
地防	0-20	120 度/100km	120 度/100km	12	3
舰船	0-20	360 度/180km	360 度/100km	36	4

2.2 构建行为树

经过分析兵棋推演平台的规则,行为树的决策任务就是利用己方兵力进行战争部署,以获得战争对抗的胜利。又因为平台内部已经实现了底层的决策控制逻辑,行为树的决策就只需要调用相关的底层行动命令接口,即可实现相应的作战决策。使用行为树实现兵棋推演的作战决策生成的过程,可以理解为是战争指挥人员对部下发布任务,这些命令任务可以是同步下达的,任务的执行也可以是同步的。那么在行为树决策过程中,可以使用平行节点来实现多任务指令的同步进行。接下来以红方为例就决策任务对行为树的构建进行分析。

首先对红方的部分作战单位进行初始战争部署(如图 3),预警机、无人机开始执行空中预警侦察任务,护卫舰到达指定海域进行海上巡逻,干扰机到达指定区域进行干扰等。初始部署完成后,预警机获取敌方平台信息,对其位置信息进行核实之后,轰炸机、歼击机编队开始进行目标打击,如图 4 所示。

接下来将战争初始部署部分的框架转化为行为树结构(如图 5)。此后根据预警机获取到的实时态势信息,行为树会在推演时间内做出相应的决策指令。

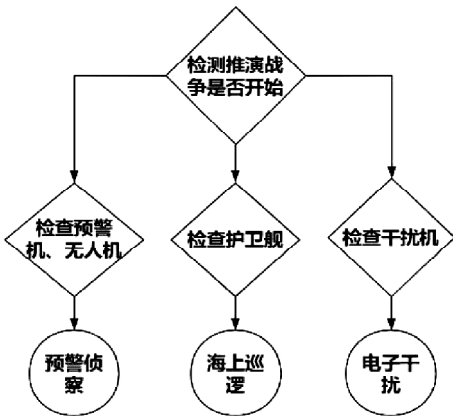


图 3 初始部署

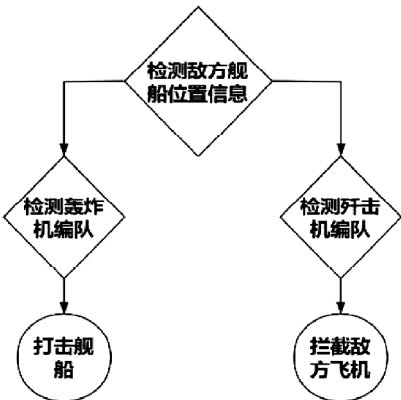


图 4 突击敌方护卫舰

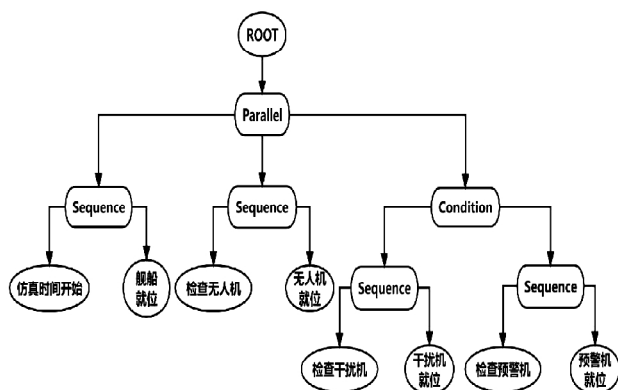


图5 对应的行为树示意图

### 2.3 兵棋推演平台调度行为树

首先在平台的对战调度和推演运行入口处接入行为树框架,如表6中的配置参数config,在config中接入行为树框架。分别将红蓝双方的行为树决策模型,将行为树的类名添加到“agent”中,然后选择推演对抗平台的态势显示终端的版本信息“image\_name”。当启动推演对抗时,触发行为树决策机制,然后按照流程图实现行为树进行作战决策。实现行为树流程如图6所示。

表6 平台接入行为树框架代码参数配置

config 参数配置接入行为树框架
<pre> config = {     'server_port': 6100,     'config': {         'scene_name': '/home/Join_Operation_scenario.mtdt', # 容器里面想定文件绝对路径         'prefix': '/', # 容器管理的脚本 manage_client 所在路径(这里给的相对路径)         'image_name': 'combatmodserver:v1.4', # 镜像名         'volume_list': [],         'max_game_len': 350 # 最大决策次数     },     'agents': {         'red_name': { # 战队名             'class': RedRuleBooster, # 智能体类名 RedRuleAgent RedRuleBooster             'side': 'red' # 智能体所属军别(不可更改!)         },         'blue_name': { # 战队名             'class': BlueRuleBooster, # 智能体类名 BlueRuleAgent BlueRuleBooster             'side': 'blue' # 智能体所属军别(不可更改!)         }     },     'replay': { # 记录回放相关设置         'save_replay': False, # 是否记录 False         'replay_dir': './replays' # 回放保存路径     },     'wargaming': { # 记录推演数据         'save_wargaming': False, # 是否记录 False         'wargaming_dir': './wargaming' # 推演数据保存路径     } } </pre>

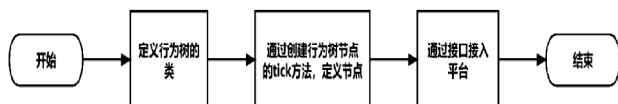


图6 实现行为树流程图

### 2.4 基于 behavior3 实现行为树

在第一节中介绍了行为树的构建,其中涉及到多种行为树节点,本小节将介绍在 behavior3 editor 编辑器中实现各节点功能。作战双方行动单位的指令复杂多样,这里以行为节点的创建为例说明行为树的创建过程。节点伪代码如表7所示。

表7 行为节点伪代码

伪代码创建行为节点函数

**Class** 创建行为节点的类

```

{
    继承行为树类的属性和方法
    创建使能信号 tick() 方法
    对节点条件进行判断
    执行指令
    节点返回状态
}

```

表8 行为节点代码

行为节点代码

```

class Takeoff_TargetHunt(b3.Action):
    def __init__(self, **kwargs):
        super(Takeoff_TargetHunt, self).__init__(**kwargs)
    def tick(self, tick):
        if self.nickname in tick.target.nickname_team_dic.keys():
            return b3.ERROR
        else:
            t_id = None
            t_id2 = None
            if isinstance(self.target, int):
                t_id = self.target
            elif self.target in tick.target.nickname_enemy_dic:
                t_id = tick.target.nickname_enemy_dic[self.target]
            if t_id is not None:
                temp_cmd = EnvCmd.make_takeoff_targethunt(self.airport_id, self.fly_num, t_id,
                                                            self.direction, self.range, self.speed)
                tick.target.cmd_list.append(temp_cmd)
                temp = tick.blackboard.get('name_cmd_dic')
                temp.update({self.nickname: temp_cmd})
                tick.blackboard.set('name_cmd_dic', temp)
            return b3.SUCCESS

```

表8是伪代码所对应的节点代码,首先创建了一个节点的类“”,并继承了行为树类的相关属性和方法,接着写实现节点功能的tick()方法,在使能信号中根据判定条件执行指令,最后返回节点的执行状态。根据双方作战单位的执行任务以及平台推演规则,以类似的方法创建行为树的其他各类节点,行为树构建的部分示意图如图7所示。

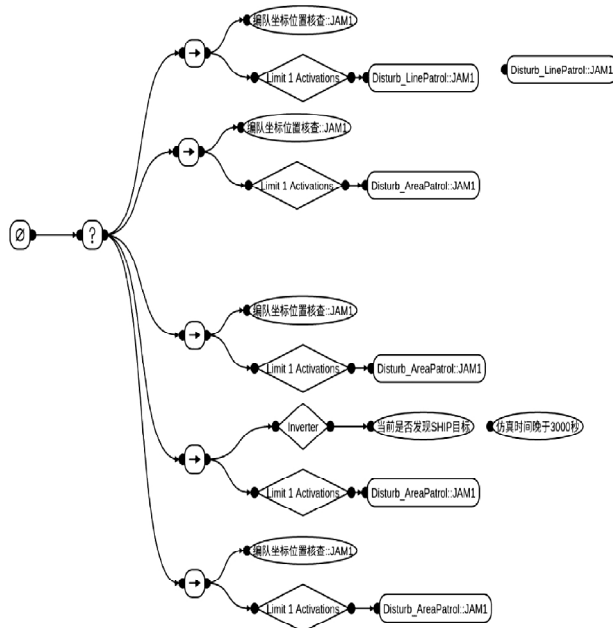


图7 行为树构建示意图

图7中所示的是在 behavior3 editor 编辑器中创建的部分行为树结构图,智能体从根节点开始执行任务,经过选择节点

(问号表示)时,选择其中一支进入到顺序节点(箭头表示)开始按从上到下的顺序执行不同的任务,当条件节点(椭圆表示)的执行条件为真时,继续执行任务,直至行为树运行完毕。

### 3 推演实现与分析

本文以智能博弈对抗赛为平台背景,红方作为攻击方,综合运用海空突击支援力量,突破蓝方防空体系,摧毁蓝方指挥所两个关键目标;蓝方作为防守方,依托地、海、空立体防空火力,守住本岛两个指挥所的重点目标。红蓝双方的决策过程根据前文所述完成的行为树决策模型实现。

当开启战争推演时,触发加载行为树的脚本事件,按照图6的流程实现模拟作战决策,仿真推演过程的信息输出,包括推演进行的实时时间、行为树节点的返回状态、执行指令等(如表9)。图8、图9中红色和蓝色(扇形区域种岛屿附近)编号标识的行动单位均为行为树所做出的决策,包括侦察任务、打击任务、护航等任务。图8中左下角的“事件”即为行为树的决策说明,图9中左下角的“单元”是所选定的作战单位的当前状态说明。随着推演时间的进行,战场态势信息的变化,行为树会做出不同的决策以应对局势变化,图8和图9是同一场战争推演过程中的局势变化示意图,可以发现所构建的行为树决策模型不仅实现了相关的侦察、巡逻、打击等任务,还可以进行压制敌方防空的打击等任务,具有较好的自主决策能力。

表9 仿真推演输出信息

仿真推演输出信息
1 64.2 根据 被攻击数目余额,更新: { 根据 被攻击数目余额,更新: { In NicknameCheck condition node, nickname JAM1 is not found! FAILURE. {'info': '红方干扰机起飞'} In EnemyTargetFound condition node, it did not find more than 0 SHIP targets! FAILURE. In EnemyTargetFound condition node, it did not find more than 2 GROUND targets! FAILURE. In EnemyNicknameCheck condition node, nickname E_SHIP1 is not found! FAILURE. In EnemyNicknameCheck condition node, nickname E_SHIP1 is not found! FAILURE. In EnemyNicknameCheck condition node, nickname E_SHIP1 is not found! FAILURE. In EnemyNicknameCheck condition node, nickname E_SHIP1 is not found! FAILURE. [{'maintype': 'takeoffareaprotol', 'airport_id': 30001, 'fly_num': 1, 'fly_type': 13, 'point_x': 113000, 'point_y': 0, 'point_z': 8500, 'direction': 270, 'length': 8000, 'width': 8000, 'speed': 230, 'patrol_time': 7200}] BehaviorTree status = 3[1:SUCCESS,2:FAILURE,3:RUNNING,4:ERROR] 根据 被攻击数目余额,更新: { 根据 被攻击数目余额,更新: { [{'maintype': 'takeoffareaprotol', 'airport_id': 20001, 'fly_num': 1, 'fly_type': 11, 'point_x': -55000, 'point_y': 35000, 'point_z': 7000, 'direction': 90, 'length': 30000, 'width': 30000, 'speed': 280, 'patrol_time': 7200}] BehaviorTree status = 1[1:SUCCESS,2:FAILURE,3:RUNNING,4:ERROR]



图8 仿真推演态势显示



图9 仿真推演态势显示

### 4 结语

本文在分析了行为树的结构基础上,结合海空兵棋推演系统平台,提出了基于行为树的智能决策模型,实现了在战场信息瞬息万变的情况下,行为树可以实时地根据态势信息更改或者重新做出决策,获得推演战争的胜利。本文虽然通过建立行为树模型实现了推演作战的策略生成过程,但是在推演过程中,作战双方的胜负局面在一定程度上是不稳定的,随机性比较强,且行为树的效果基本依赖于根据专家经验分析作战规则所构建的决策模型,很可能会在一些细节决策上产生失误导致战争的失败,因此,在行为树中加入机器学习技术,以实现提升智能体的全方位决策精确性将是下一步的思考问题。

### 参考文献:

- [1] 鲁大剑.面向作战推演的博弈与决策模型及应用研究[D].南京:南京理工大学.2013.
- [2] 刘满,张宏军,郝文宁,等.战术级兵棋实体作战行动智能决策方法[J].控制与决策,2020,35(12):2977-2985.
- [3] 夏阳升,石建迈,陈超,等.车机协同多区域覆盖侦察路径规划方法[J].指挥与控制学报,2020,6(4):372-380.
- [4] 邹焯翰,冯阳赫,程光权,等.面向军事条令条例的本体构建技术[J].指挥与控制学报,2019,5(1):47-54.
- [5] Alex. J. Champandard. Behavior Tree for Next Gen AI.http://aigamedev.com/insider/presentations/behavior-tree/2008-11-28.
- [6] Alex.J.Champandard.Understanding the Second-Generation of Behavior Tree.http://aigamedev.com/insider/tutorial/second-generation-bt,2012-02-26.
- [7] 董倩,纪梦琪,朱一凡,等.空中作战决策行为树建模与仿真[J].指挥控制与仿真,2019,041(001):12-19.
- [8] Puga G F, Gómez-Martínez M A, Dağz-Agudo B, et al. Dynamic Expansion of Behavior Trees [C], Artificial Intelligence and Interactive Digital Entertainment Conference, October 22-24,2008, Stanford, California, Usa.DBLP,2008.
- [9] S. Yi, S. Worrall and E. Nebot, Integrating Vision, Lidar and GPS Localization in a Behavior Tree Framework for Urban Autonomous Driving[C]. 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), 2021, pp. 3774-3780.
- [10] 智能博弈挑战赛.https://encourage.dcjingsai.com/WarGame.html.
- [11] http://pu-datacastle.oss-cn-qingdao.aliyuncs.com/master.t hird.source/other/competition/联合作战智能博弈挑战赛\_白皮书/联合作战智能博弈挑战赛\_白皮书 .html#\_Toc1993.