

Decision-Making Tree Generation Algorithm for Weapon Resource Assignment

Jiangbo SUI, Ling WU, Changfeng XING

Electronic Engineering College, Naval University of Engineering, Wuhan 430033, China

E-mail: jiangbo_sui@163.com

Abstract: By analyzing the operational process of weapon platform intercepting air targets, the models of target intercepting time and firing time are established. Insisting on the principle that ownership has more chances to intercept targets, it presents a decision-making tree generation algorithm for weapon resource assignment. This algorithm synchronously solves the problem of weapon resource planning and weapon resource scheduling. A practical example is provided to show the effectiveness of this algorithm. It can be used for reference researches in this area.

Keywords: weapon resource assignment; decision-making tree; backtracking

武器资源分配问题的方案决策树生成算法

隋江波, 吴玲, 邢昌凤

海军工程大学电子工程学院, 湖北 武汉 430033

Email: jiangbo_sui@163.com

摘要: 通过对武器平台防空作战过程分析, 建立了武器目标拦截时间和武器开火时间模型, 以对目标进行更多次抗击为原则, 提出了一种方案决策树生成算法。该算法从武器目标拦截时间和武器开火时间节点进行回溯搜索, 得到从武器最晚拦截时间到武器最早拦截时间之间的可行武器资源分配方案, 同时解决了资源分配规划与调度两个问题。仿真验证了算法的实用性, 对相关领域的研究有一定借鉴意义。

关键词: 武器资源分配; 决策树; 回溯搜索

1 引言

武器资源分配问题应包括两部分内容: 资源分配规划 (resource allocation planning, RAP) 和资源分配调度 (resource allocation scheduling, RAS)。RAP 即军事运筹领域经常提到的武器目标分配问题 (weapon target assignment, WTA), 是在武器与目标之间进行优化排列组合, 确定武器作战任务。RAS 主要用于确定武器具体行动方案, 包括方案的起始时间、终止时间和武器开火时间等, 目前解决这一问题的方法是在确定武器作战任务的基础上, 再根据任务目标信息和武器的相关战技指标制定出方案的具体行动计划。自上世纪 80 年代美国麻省理工学院的 Hosein 等人对一般性的 WTA 问题做出较为系统的研究后[1], 涌现出大量应用于解决 WTA 问题的算法, 如遗传算法[2]、蚁群算法[3]、粒子群算法[4]、模拟退火法[5]、人工

神经网络[6]、禁忌搜索算法等[7]。这些算法各有所长, 然而当这些算法与实际应用相结合时, 总是出现这样或那样的问题, 实际应用价值不高, 究其原因, 主要是着重于算法本身的优化设计, 而忽略了实际交战的一些影响因素。

本文从武器防御作战流程出发, 以实际交战中的各个重要时间节点为依据, 设计了一种方案决策树生成算法。该算法将武器开火时间和武器目标拦截时间引入武器资源分配过程中, 同时解决了 RAP 与 RAS 两个问题, 使作战系统的反应更快。

2 武器防御作战流程与方案决策树分析

对于一次防御交战来说, 应包含以下几方面的条件: 一种防御武器、一个目标跟踪指示器、一个威胁目标和一段具有时序性的四步作战行动, 这四步作战行动是: 发现并锁定目标、武器开火、目标拦截和目

标杀伤评估[8]。于是某一武器 W_i 对某一目标 T_j 的拦截交战在时间上可表示为 $I_{ij} = [t_{\min}, t_{\max}]_{ij}$ ，其中 t_{\min} 为武器 W_i 拦截目标 T_j 的最早时间， t_{\max} 为武器 W_i 拦截目标 T_j 的最晚时间。所有武器对每个目标的可行拦截时间间隔表示为矩阵 E 。

$$E = \begin{bmatrix} I_{11} & I_{12} & \cdots & I_{1n} \\ I_{21} & I_{22} & \cdots & I_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ I_{m1} & I_{m2} & \cdots & I_{mn} \end{bmatrix}$$

其中 m 为武器数量， n 为目标数量。

在许多试探性算法特别是人工智能算法中，每进一步都面临着若干种选择，选择策略和前进“路径”的最直观表示法就是树。无论是想搜索一个解、所有解或最优解，最重要的问题是要能确信检查到了各种可能或在某种限制条件下的各种可能。最常见的树状结构是二叉树结构，如在常见的故障树分析中，根据出现的故障情况判断出可能导致故障产生的原因，不同的分叉代表不同的结果。

方案决策树是指，按照各个武器对目标开火的时间顺序，对武器目标全航路分配方案所进行的树状决策网络分划，并据此对作战方案进行选择、优化。

3 交战中关键时间变量的计算

算法中涉及的重要变量有威胁目标攻击作战平台的时间 t_{hit} 、武器拦截最早时间 t_{\min} 、武器拦截最晚时间 t_{\max} 、武器开火时间 t_{fire} 、交战效果评估时间 t_{KA} 等。

目标攻击时间计算公式为

$$t_{hit} = R_T / v_T \quad (1)$$

其中 R_T 为目标到平台的距离， v_T 为目标速度。

武器拦截最早和最晚时间与武器雷达锁定目标时间、武器射界有关，其计算过程如下^[9]：

首先判断武器与目标相遇时，目标是否进入武器的射界内，即拦截点是否位于武器射界内。

令 $d = (R_T - v_T \cdot t_{lock}) \cdot v_W / (v_T + v_W)$

其中 t_{lock} 为武器雷达锁定目标时间， v_W 为武器弹体飞行速度。则有

(1) 若 $d > R_{\max}$ ，此时拦截点未进入武器的最大拦截距离，则最早和最晚拦截时间的计算公式为

$$t_{\min} = (R_T - R_{\max}) / v_T$$

$$t_{\max} = (R_T - R_{\min}) / v_T$$

其中 R_{\max} 为武器的最大拦截距离， R_{\min} 为武器的最小拦截距离。

(2) 若 $R_{\min} \leq d \leq R_{\max}$ ，此时拦截点进入武器的射界内，则最早和最晚拦截时间的计算公式为

$$t_{\min} = (R_T - v_T \cdot t_{lock}) / (v_T + v_W) + t_{lock}$$

$$t_{\max} = (R_T - R_{\min}) / v_T$$

(3) 若 $d < R_{\min}$ ，此时拦截点进入武器最小拦截距离内，则该武器不能用于拦截目标。

在防空反导作战中，只要目标进入近程防御武器系统 CIWS 的射击范围，CIWS 立即开火，其最早开火时间的计算公式为

$$CIWS_t_{fire} = CIWS_t_{\min} - CIWS_t_{air} \quad (2)$$

其中 $CIWS_t_{air}$ 为武器滞空时间。

为避免舰炮弹丸与 CIWS 弹丸发生火力冲突，舰炮的开火时间将受到 CIWS 开火时间的影响，其开火时间为

$$Gun_t_{fire} = CIWS_t_{fire} - CIWS_t_{space} \quad (3)$$

其中 $CIWS_t_{space}$ 为舰炮提前 CIWS 射击的时间。

在满足 CIWS 的射击时间限制条件下，舰炮的射击原则为：只要目标位于舰炮射击区域内，就进行射击，每次射击的时间间隔为 Gun_t_{inter} 。

舰空导弹最晚开火时间为

$$SAM_t_{fire} = SAM_t_{\max} - SAM_t_{air} \quad (4)$$

目标拦截时间落入时间间隔 $[t_{\min}, t_{\max}]$ 内的武器的开火时间为

$$t_{fire} = (t_{interception} \cdot (v_W + v_T) - R_T) / v_W \quad (5)$$

其中 $t_{interception}$ 为武器目标拦截时间。

舰空导弹和舰炮的交战效果评估时间为

$$t_{KA} = t_{interception} + t_{killAss} \quad (6)$$

其中， $t_{killAss}$ 为武器杀伤评估时间常量。

近程防御武器系统的交战效果评估时间即为目标攻击舰艇的时间，有

$$CIWS_t_{KA} = t_{hit}$$

4 算法设计

通常情况下，随着目标与作战平台之间距离的减小，作战平台将获得以下优势：武器对目标的杀伤力将不断增大；目标能够被可靠稳定地跟踪；目标信息的获取更加准确；平台防御反应时间相对变长。因而选择武器在目标最晚拦截时间与其交战作为初始武器目标分配方案，并以此为最终方案决策树的主干部分。方案决策树主干上每一个初始分配方案都作为一个节点，然后按时间标记依次对每个节点进行可行方案的

回溯搜索重分配，添加新的交战方案，以保证作战平台对目标的再交战次数最大。算法的步骤如下。

第一步：生成一个由所有武器对每个目标最晚拦截时间为截止点的交战方案树。程序设计如下：

```

while E ≠ 0 then
    选择(Wi, Tj) 使 Ii,j = τ(E);
    for each τ(Ii,j) > fi,j(τ(E)) do
        插入方案(τ(Ii,j), Wi, Tj);
        E' = Reduce(E, Wi, Tj);
    end for
    E = E'
end
    
```

其中， $\tau(I_{ij}) \leftarrow \max(I_{ij})$ 表示 I_{ij} 中最晚拦截时间，即武器 W_i 对目标 T_j 的最晚拦截时间； $\tau(E) \leftarrow \max_{i,j} \tau(I_{ij})$ 表示所有 I_{ij} 中最晚拦截时间； $f_{ij}(\tau(I_{ij}))$ 表示武器 W_i 在时刻 $\tau(I_{ij})$ 拦截目标 T_j 的开火时间； $(\tau(I_{i,j}), W_i, T_j)$ 表示一个交战方案，该方案表示武器 W_i 在时间 $\tau(I_{i,j})$ 拦截目标 T_j ；“Reduce”程序表示删减 E 中的 I_{ij} 生成新的矩阵 E' 。

第二步：在第一步得到的交战方案树的基础上，以武器目标最晚拦截时间所确定的方案为节点，重新向前回溯搜索新的武器对目标的可行交战方案，以提高武器使用效率和目标的拦截次数。设 A 是由第一步算法得到的交战方案集； A_v 是节点 v 的可行方案子集； t_v 是节点 v 的时间标记； r 为目标威胁等级标记。程序设计如下：

```

while 存在 v 使 Av ≠ 0 do
    选择具有最早时间标记 tv 的有效节点 v;
    r ← 1;
    搜索 Av;
    if 搜到一个交战方案 then
        在规划中插入该方案并且更新可行交战士子集;
    else
        r ← r + 1;
    end if
    if r ≤ n (n 为目标个数) then
        return “搜索 Av .....”;
    end if
end while
    
```

5 实例仿真

雷达探测到两个目标向我方舰艇袭来，标识为目标 T1 和目标 T2，其距舰艇的距离分别为 28km 和 8km。假设目标速度为 850m/s，由式 (1) 可得目标攻击舰艇的时间为

$$T1_{-}t_{hit} = 32.94s$$

$$T2_{-}t_{hit} = 9.41s$$

舰艇拦截目标的主要硬武器为反舰导弹 SAM、舰炮 Gun 和近程防御系统 CIWS，具体参数如表 1 所示^[3]。

Table 1. Relative parameters of weapon systems
表 1. 各武器系统相关参数

	Speed	minRange	maxRange
SAM	0.9Ma	2.2km	16km
Gun	850m/s	0.9km	5km
CIWS	1200m/s	—	2.5km

以 SAM 为例，说明如何求得武器对目标 T1、目标 T2 的拦截时间间隔。设导弹与舰炮共享一个火控雷达，雷达锁定时间为 3s。

对于目标 T1 有

$$dSAM = \frac{28 - 0.85 \times 3}{0.85 + 0.34 \times 0.9} \times 0.34 \times 0.9 = 6.74km$$

由判断条件 (2)，此时拦截点位于武器射界内，可得

$$SAM_{-}t_{min} = \frac{28 - 0.85 \times 3}{0.85 + 0.34 \times 0.9} + 3 = 25.02s$$

$$SAM_{-}t_{max} = \frac{28 - 2.2}{0.85} = 30.35s$$

对于目标 T2 有

$$dSAM = \frac{8 - 0.85 \times 3}{0.85 + 0.34 \times 0.9} \times 0.34 \times 0.9 = 1.44km$$

由判断条件 (3)，此时拦截点进入武器最小拦截距离内，因而 SAM 不能拦截目标 T2。同理可得其他武器对目标 T1、目标 T2 的拦截时间间隔，见表 2。

Table 2. Intercepting time intervals of weapons
表 2. 各武器拦截时间间隔

	min/maxTime(T1)	min/maxTime(T2)
SAM	25.02s/30.35s	—
Gun	27.06s/31.88s	6.21s/8.35s
CIWS	30.00s/32.94s	6.47s/9.41s

于是可得

$$E = \begin{bmatrix} (25.02, 30.35) & (0, 0) \\ (27.06, 31.88) & (6.21, 8.35) \\ (30.00, 32.94) & (6.47, 9.41) \end{bmatrix}$$

设 CIWS 火控雷达锁定时间为 1s，舰炮提前 CIWS 射击的时间为 0.3s，SAM 与舰炮的评估时间常量分别为 0.09s 和 0.06s，运行算法第一步得到一组以目标最晚拦截时间为基准的武器资源分配方案决策树如图 1 所示。

各武器的最晚开火时间和效能评估时间，见表 3。

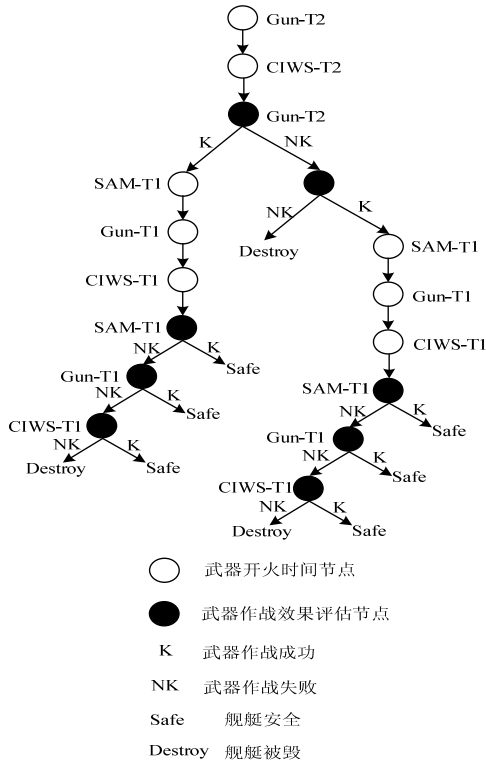


Figure 1. Weapon resource allocation scheme

图 1. 武器资源分配方案

Table 3. Fire and estimation time of weapons
表 3. 各武器开火与评估时间

	T1(fire)	T1(KA)	T2(fire)	T2(KA)
SAM	23.15s	30.44s	—	—
CIWS	27.92s	32.94s	4.39s	9.41s
Gun	27.62s	30.34s	4.09s	6.81s

运行算法第二步，选择具有最早交战时间的节点“Gun-T2”，为避免对 CIWS 的影响，舰炮对目标 T2 的实际拦截时间区间为 [6.21,6.75]，设舰炮齐射时间间隔为 1.5s，则向前回溯的可能方案开火时间为 2.59s，其对应的拦截时间为 6s，超过了舰炮最早拦截时间，因而对于节点“Gun-T2”可行方案子集为零。

目标一旦进入 CIWS 作战范围，CIWS 将持续射击直到击毁目标，因而对于节点“CIWS-T2”来说，可行方案子集为零。

设 SAM 的射击原则为观察到前一枚导弹拦截失败，立即发射下一枚。对于节点“SAM-T1”，其最晚发射时间为 23.15s，则上一枚导弹的评估时间为 23.15s，由式 (6) 得上一枚导弹的拦截时间为 23.06s，显然这超出了 SAM 的最早拦截时间 25.02s，因而对于节点“SAM-T1”来说，可行方案子集为零。

同理可以得到节点“Gun-T1”和“CIWS-T1”的方案子集，最终形成的武器资源分配方案如表 4 所示。

Table 4. Weapon resource allocation scheme
表 4 武器资源分配方案

	开火(T1)	拦截(T1)	评估(T1)	开火(T2)	拦截(T2)	评估(T2)
SAM	23.15s	33.35s	30.44s	—	—	—
Gun-1	21.62s	27.28s	27.34s	4.09s	6.75s	6.81s
Gun-2	23.12s	28.03s	28.09s	—	—	—
Gun-3	24.62s	28.78s	28.84s	—	—	—
Gun-4	26.12s	29.53s	29.59s	—	—	—
Gun-5	27.62s	30.28s	30.34s	—	—	—
CIWS	27.92s	30.00s	32.94s	4.39s	6.47s	9.41s

6 结 论

本文所设计的用于解决武器资源分配问题的决策树生成算法，是通过对武器目标拦截时间和开火时间进行回溯搜索迭代，从而得到由武器最早拦截时间到武器最晚拦截时间的武器资源分配方案，保证了对目标进行更多次的抗击。从最终的方案决策树来看，每一分支的交战评估结果都将影响到武器资源的下一步分配，所有的分支代表了实际交战中可能发生的某种战场态势，算法给出的武器资源分配方案，能够为指挥员迅速判明战场态势、定下决心提供辅助决策。

References (参考文献)

- [1] H. P. Cai, Y. W. Chen, L. N. Xing. Research on Dynamic Weapon Target Assignment Problem Based on SVNTS Algorithm. Computer Engineering and Applications, 2006 (31): 7-10.
蔡怀平, 陈英武, 邢立宁. SVNTS 算法的动态武器目标分配问题研究[J]. 计算机工程与应用, 2006(31): 7-10.
- [2] Z. J. Lee, S. F. Su, C. Y. Lee. Efficiently solving general weapon-target assignment problem by genetic algorithms with greedy eugenics. IEEE Transactions on Systems Man and Cybernetics Part B, 2003, 33(1): 113-121.
- [3] Z. J. Lee, C.Y. Lee, S. F. Su. An immunity-based ant colony optimization algorithm for solving weapon-target assignment problem. Applied Soft Computing, 2002, 2(1):39-47.
- [4] S. Gao, J. Y. Yang. Solving Weapon-target Assignment Problem by Particle Swarm Optimization Algorithm. System Engineering and Electronics, 2005, 27(1): 79-80.
高尚, 杨静宇. 武器-目标分配问题的粒子群优化算法[J]. 系统工程与电子技术, 2005, 27 (1): 79-80.
- [5] P. Wu, Q. Liang. Simulated Annealing Algorithm for Weapon-Target Assignment Problem. Computer Engineering and Applications, 2006(4):87-90.
吴平, 梁青. 武器-目标分配问题的模拟退火算法[J]. 计算机工程与应用, 2006 (4): 87-90.
- [6] E. Wacholder. A neural network-Based optimization algorithm for the static weapon-target assignment problem. ORSA Journal on Computing, 1989, 4:232-246.
- [7] Y. R. Huang. Intelligent Optimization Algorithm and Its Application, National Defense Industry Press, Beijing, China, 2008.
黄友锐. 智能优化算法及其应用[M]. 北京: 国防工业出版社, 2008.
- [8] A. Benaskeur, E. Bosse, D. Blodgett. Combat resource allocation planning in naval engagements. Defense R&D Canada-Valcartier Technical Report, 2007:23-27.
- [9] H. Y. Wang, S. J. Wang, P. Li, Principles of Warship Fire Control System, National Defense Industry Press, Beijing, China, 2006.
王航宇, 王士杰, 李鹏. 舰载火控原理[M]. 北京: 国防工业出版社, 2006.