

上海交通大学硕士学位论文

## 空空导弹分布式任务分配技术研究

硕 士 研 究 生：张 贇

学 号：118032910119

导 师：蔡云泽研究员

申 请 学 位：工程硕士

学 科：控制工程

所 在 单 位：自动化系

答 辩 日 期：2020 年 11 月 29 日

授予学位单位：上海交通大学



Dissertation Submitted to Shanghai Jiao Tong University  
for the Degree of Master

**RESEARCH ON DISTRIBUTED TASK  
ASSIGNMENT TECHNOLOGY OF  
AIR-TO-AIR MISSILES**

|                                       |                               |
|---------------------------------------|-------------------------------|
| <b>Candidate:</b>                     | Zhang Yun                     |
| <b>Student ID:</b>                    | 118032910119                  |
| <b>Supervisor:</b>                    | Prof. Cai Yunze               |
| <b>Academic Degree Applied for:</b>   | Master of Engineering         |
| <b>Speciality:</b>                    | Control Engineering           |
| <b>Affiliation:</b>                   | Department of Automation      |
| <b>Date of Defence:</b>               | November 29, 2020             |
| <b>Degree-Conferring-Institution:</b> | Shanghai Jiao Tong University |



# 上海交通大学

## 学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期：            年        月        日



## 上海交通大学 学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权上海交通大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保 密 ☐，在 \_\_\_\_\_ 年解密后适用本授权书。

本学位论文属于

不保密 ☒。

(请在以上方框内打✓)

学位论文作者签名： 某某

指导教师签名： 某某

日 期： 某 年 某 月 某 日

日 期： 某 年 某 月 某 日





## 空空导弹分布式任务分配技术研究

### 摘 要

现代空战中，空空导弹是一种杀伤力强、威慑度高的尖端武器，对夺取战场制空权和主动权具有重要的意义。在空战环境中，常常面对着多目标、多任务的场景，发展空空导弹对集群目标的攻击作战模式是一个重要方向。但随着大规模集群，如无人机集群投入实战的趋势，在空空导弹拦截大规模集群的场景中，由于集中式算法在求解大规模任务分配问题时面临通信限制和实时性问题，因此研究使用分布式算法求解任务分配问题具有重要的现实意义。

本文首先结合空空导弹的特点，系统介绍了空空导弹任务分配技术的研究现状及存在的主要问题。接着，本文依据空战的实际特点和需求，考虑空战中的约束条件，建立了整数规划模型。然后，针对空战强博弈的特点，本文利用博弈论的思想，建立了两种博弈论模型，研究并改进了在两种博弈论模型下的分布式任务分配算法，通过仿真与对比试验验证了其有效性，且在大规模问题实时性方面具有显著的优越性。最后，针对空战高动态的特点，本文研究了基于随机博弈模型下的动态任务分配算法，通过仿真试验研究了分配算法的动态适应性。

本文的主要研究贡献主要有：

- 1、针对空空导弹任务分配问题，以最短预计剩余攻击时间和最小导弹机动量为优化目标，建立了整数规划模型。利用博弈论中的势博弈模型，结合势博弈在多智能体系统控制中的应用，设计了无约束条件下导弹个体效用函数和协商协议，通过对比实验分析了该算法的有效性和实时性。然后针对整数规划问题的特点，提出了基于拉格朗日乘子法和基于改进美好生活效用（Wonderful Life Utility, WLU）的有约束条件下任务分配算法，通过理论分析和仿真对比验证了两种算法在求解有约束条件下的任务分配问题时保留了无约束优化时的有效性

和实时性，具有良好的实用性和可扩展性。

2、针对上述整数规划模型，利用博弈论中的享乐博弈模型，将任务分配问题转化为种群聚类问题，首先结合空战约束条件设计了享乐博弈模型下的关于种群数量的个体效用函数，然后基于 **Zeuthen** 策略设计了一种分布式协商协议，接着从理论方面分析了该算法的最坏优化值及迭代步数，并通过仿真对比实验验证了算法的有效性和实时性。

3、针对高动态场景下需要及时调整任务分配的问题，研究了基于随机博弈模型的动态任务分配问题。设计了导弹与目标对战随机博弈场景框架，在假定静态模型区间内利用势博弈模型建立并求解任务分配问题，同时在模型状态切换的时间点，设计滚动策略集，兼顾智能体决策的连续性和对新生事件的及时响应性。

**关键词：**空空导弹，任务分配，分布式，势博弈，享乐联盟博弈，随机博弈

# RESEARCH ON DISTRIBUTED TASK ASSIGNMENT TECHNOLOGY OF AIR-TO-AIR MISSILES

## ABSTRACT

In modern air warfare, air-to-air missiles are sophisticated weapons with strong lethality and high deterrence. They are of great significance to air dominance and initiative on the battlefield. An air combat environment is often faced with multi-target and multi-task scenarios, thus it is an important direction to develop an air-to-air missile attack combat mode against cluster targets. However, with the trend of large-scale UAV clusters being put into actual combat, in the scenario of air-to-air missiles intercepting large-scale clusters, centralized task assignment algorithms are limited to communication limitations and real-time problems. Therefore, the use of distributed algorithms to solve the task assignment problem have important practical significance.

This article first systematically introduces the related work and main problems of air-to-air missile task assignment technology, combined with the characteristics of air-to-air missiles. Then, according to the actual characteristics and requirements of air combat, this article takes the constraints of air combat in consideration, and establishes an integer programming model. In view of the characteristics of the strong game of air combat, this article introduces the idea of game theory to establish two game theory models, researches and improves the distributed task assignment algorithm under the two game theory models. These algorithms are verified effective and significantly advantageous in real-time large-scale problems through simulation and comparative experiments. Finally, in view of the high dynamic characteristics of air combat, this article studies the task reassignment algorithm

based on the two game theory models proposed in three scenarios, and studies the dynamic adaptability of the reassignment algorithm through simulation experiments.

The main research contributions of this article are as follows:

1. Aiming at the task allocation problem of air-to-air missiles, with the shortest expected remaining attack time and the minimum missile maneuver as the optimization objectives, an integer programming model is established. Using the potential game model in game theory, combined with the application of potential game in the control of multi-agent systems, the individual utility function and negotiation protocol of the missile are designed under unconstrained conditions, and the effectiveness and real-time performance of the algorithm are analyzed through comparative experiments. Then, in view of the characteristics of integer programming problems, task assignment algorithms under constraints are proposed respectively based on Lagrange multiplier method and based on the adjusted Wonderful Life Utility (WLU). Theoretical analysis and simulation comparison verify that the two algorithms retains the validity and real-time performance of unconstrained optimization, and has good practicability and scalability.

2. Aiming at the above integer programming model, using the hedonic game model in game theory, the task allocation problem is transformed into a population clustering problem. First, the individual utility function with respect to the number of population under the hedonic game model is introduced based on the air combat constraints, and then A distributed negotiation protocol is designed based on the Zeuthen negotiation strategy. The worst performance analysis of the algorithm are theoretically conducted, and the effectiveness and real-time performance of the algorithm are verified through simulation and comparison experiments.

3. Aiming at the problem of timely adjustment of task allocation in high dynamic scenarios, the dynamic task allocation problem based on stochastic

game model is studied. A stochastic game framework of missile and target battle is designed, and the potential game model is used to establish and solve the task allocation problem in the assumed static model interval. At the same time, at the time of state switching, a rolling strategy set is designed to take into account the continuity and matching of the agent's decision. Timely response to new events.

**KEY WORDS:** Air-to-Air Missile, Task Assignment, Distributed Algorithm, Potential Game, Hedonic Coalition Game, Stochastic Game



# 目 录

|                            |           |
|----------------------------|-----------|
| <b>第一章 绪论</b>              | <b>1</b>  |
| 1.1 课题研究背景及意义              | 1         |
| 1.2 国内外研究现状                | 1         |
| 1.2.1 空空导弹任务分配技术研究现状       | 1         |
| 1.2.2 博弈论研究现状及其在任务分配技术中的应用 | 1         |
| 1.3 主要研究内容                 | 1         |
| <b>第二章 空空导弹任务分配模型</b>      | <b>3</b>  |
| 2.1 引言                     | 3         |
| 2.2 弹目相对运动模型               | 3         |
| 2.3 剩余攻击时间估计               | 4         |
| 2.4 最优制导律                  | 6         |
| 2.5 任务分配模型建立               | 8         |
| 2.6 本章小结                   | 9         |
| <b>第三章 基于势博弈模型的分布式任务分配</b> | <b>11</b> |
| 3.1 引言                     | 11        |
| 3.2 势博弈模型                  | 11        |
| 3.3 智能体效用函数                | 12        |
| 3.4 约束条件处理方法               | 13        |
| 3.4.1 势博弈成立条件              | 14        |
| 3.4.2 纯纳什均衡可行性分析           | 15        |
| 3.4.3 纳什均衡最优性              | 16        |
| 3.5 均衡求解                   | 17        |
| 3.5.1 虚拟博弈算法               | 17        |
| 3.5.2 后悔匹配算法               | 19        |
| 3.5.3 空间自适应博弈算法            | 21        |
| 3.6 仿真分析与对比                | 21        |
| 3.7 本章小结                   | 21        |

|                               |           |
|-------------------------------|-----------|
| <b>第四章 基于享乐联盟博弈的分布式任务分配</b>   | <b>23</b> |
| 4.1 引言                        | 23        |
| 4.2 享乐联盟博弈模型                  | 23        |
| 4.3 基于享乐联盟博弈模型的任务分配算法         | 23        |
| 4.3.1 联盟效用函数                  | 23        |
| 4.3.2 协商协议                    | 23        |
| 4.3.3 算法收敛性分析                 | 23        |
| 4.4 仿真分析与对比                   | 23        |
| 4.4.1 算法性能对比                  | 23        |
| 4.5 本章小结                      | 23        |
| <b>第五章 基于随机博弈模型的分布式动态任务分配</b> | <b>25</b> |
| 5.1 引言                        | 25        |
| 5.2 随机博弈模型                    | 25        |
| 5.3 面向动态任务分配问题的随机博弈模型设计       | 25        |
| 5.3.1 效用函数估计                  | 25        |
| 5.3.2 滚动策略集                   | 25        |
| 5.3.3 分布式随机算法                 | 25        |
| 5.4 仿真对比与分析                   | 25        |
| 5.5 本章小结                      | 25        |
| <b>第六章 总结与展望</b>              | <b>27</b> |
| <b>附录 A Maxwell Equations</b> | <b>29</b> |
| <b>附录 B 绘制流程图</b>             | <b>31</b> |
| <b>致 谢</b>                    | <b>33</b> |
| <b>攻读学位期间发表（或录用）的学术论文</b>     | <b>35</b> |
| <b>攻读学位期间获得的科研成果</b>          | <b>37</b> |
| <b>个人简历</b>                   | <b>39</b> |



## 插图索引

|                      |    |
|----------------------|----|
| 图 2-1 弹目相对运动模型 ..... | 3  |
| 图 B-1 绘制流程图效果 .....  | 31 |



## 表格索引



## 算法索引

|                              |    |
|------------------------------|----|
| 算法 3-1 JSFP 算法流程 .....       | 19 |
| 算法 3-2 GRMFMI 算法流程 .....     | 20 |
| 算法 3-3 SAP 和 sSAP 算法流程 ..... | 22 |



## 主要符号对照表





## 第一章 绪论

### 1.1 课题研究背景及意义

### 1.2 国内外研究现状

#### 1.2.1 空空导弹任务分配技术研究现状

#### 1.2.2 博弈论研究现状及其在任务分配技术中的应用

### 1.3 主要研究内容



## 第二章 空空导弹任务分配模型

### 2.1 引言

在任务分配问题中，首先需要根据导弹自身传感器或其他支持平台收集得到的数据和信息，估计战场态势，建立多对多的任务分配模型。在空空导弹的作战场景下，本节选择了预计剩余攻击时间和当前导弹机动量是两个较为显著且易于获得的影响要素，首先分别介绍了两个要素的估计方法，之后以最小化总攻击时间和最小化导弹机动量为目标函数，建立多对多任务分配模型。

### 2.2 弹目相对运动模型

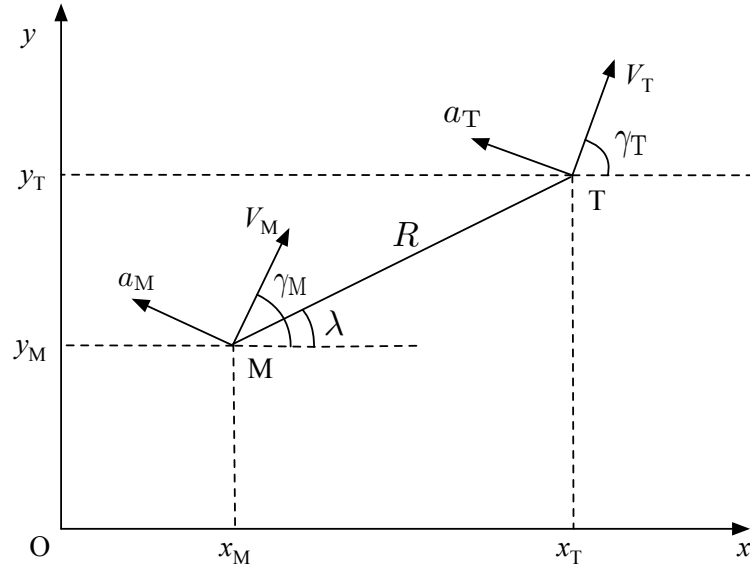


图 2-1 弹目相对运动模型

Figure 2-1 Relative motion model of missile and target

在建立多对多任务分配模型之前，需要先建立一对一的空战模型。本文所考虑的模型是在二维平面内，不考虑导弹和目标具体形状的影响，即将导弹和目标当作质点处理。图2-1展示的是本文所使用的导弹与目标相对运动模型，其中  $M$  表示导弹， $T$  表示目标， $[x_{(\cdot)}, y_{(\cdot)}]$ ,  $V_{(\cdot)}$ ,  $a_{(\cdot)}$ ,  $\gamma_{(\cdot)}$  分别表示导弹和目标的位置、速度、加速度和航向角。 $r$  为弹目距离， $\lambda$  表示弹目视线角。

为了建立弹目相对运动模型，需要作出以下假设：1、导弹与目标速度均为常值；2、目标的速度小于导弹的速度。由此分别建立导弹和目标的运动方程为：

$$\dot{x}_M = V_M \cos \gamma_M, \quad \dot{y}_M = V_M \sin \gamma_M, \quad \dot{\gamma}_M = \frac{a_M}{V_M}, \quad (2-1)$$

$$\dot{x}_T = V_T \cos \gamma_T, \quad \dot{y}_T = V_T \sin \gamma_T, \quad \dot{\gamma}_T = \frac{a_T}{V_T}. \quad (2-2)$$

令  $V_R, V_\lambda$  分别表示弹目相对速度平行于和垂直于视线角方向的分量， $V_R$  表示弹目相对接近速度，则弹目相对运动方程为：

$$V_R \equiv \dot{R} = V_T \cos \sigma_T - V_M \cos \sigma_M, \quad (2-3)$$

$$V_\lambda \equiv R\dot{\lambda} = V_T \sin \sigma_T - V_M \sin \sigma_M, \quad (2-4)$$

其中  $\sigma_{(\cdot)}, (\cdot) \in \{M, T\}$  分别表示是导弹和目标的前置角，其定义为：

$$\sigma_M = \gamma_M - \lambda, \quad \sigma_T = \gamma_T - \lambda. \quad (2-5)$$

由此可求得视线角变化率：

$$\dot{\lambda} = \frac{V_T \sin \sigma_T - V_M \sin \sigma_M}{R}. \quad (2-6)$$

### 2.3 剩余攻击时间估计

在导弹攻击过程中，为了使得预计击中时间最短，需要估计剩余攻击时间，本节采用对目标状态投影到弹目视线方向，将目标转化为相对静止状态的思想，实现对机动目标的剩余攻击时间估计。

首先假设目标静止，即  $V_T = 0$ 。此时由式 (2-1)、(2-5) 和 (2-6) 可得：

$$\dot{\sigma}_M = \frac{a_M}{V_M} + \frac{V_M \sin \sigma_M}{R}, \quad (2-7)$$

结合式 (2-3) 可得：

$$\frac{d\sigma_M}{dR} = -\frac{a_M}{V_M^2 \cos \sigma_M} - \frac{1}{R} \tan \sigma_M. \quad (2-8)$$

为方便叙述，令  $\rho = \sin \sigma_M$ ，代入式 (2-8) 可得：

$$\frac{d\rho}{dR} = -\frac{a_M}{V_M^2} - \frac{1}{R}\rho. \quad (2-9)$$

此处为方便推导，考虑导弹采用比例制导律，制导指令为：

$$a_M = NV_M\dot{\lambda}, \quad (2-10)$$

其中  $N$  为制导比例系数。将式 (2-10) 和 (2-6) 代入式 (2-9) 可得：

$$\frac{d\rho}{dR} = (N-1)\frac{\rho}{R}. \quad (2-11)$$

式 (2-11) 是一个一阶微分方程，其解形式为：

$$\rho = \rho_0 \left( \frac{R}{R_0} \right)^{N-1}, \quad (2-12)$$

其中  $\rho_0$  和  $R_0$  分别为初始时刻的对应变量值。将式 (2-3) 两边对  $R$  求积分，并进行泰勒展开得到：

$$\begin{aligned} t_f &= \frac{1}{V_M} \int_0^{R_0} \frac{1}{\sqrt{1-\rho^2}} dR \\ &= \frac{1}{V_M} \int_0^{R_0} \left( 1 + \sum_{i=1}^{\infty} \frac{2i-1}{2^i} \rho^{2i} \right) dR, \end{aligned} \quad (2-13)$$

其中  $t_f$  为初始时刻导弹已经飞行的时间。为简化问题，仅取式 (2-13) 泰勒展开的前两项代入式 (2-12) 求解，并使用小角度假设  $\rho \approx \sigma_M$  可得：

$$t_f = \frac{R_0}{V_M} \left( 1 + \frac{1}{2(2N-1)} \sigma_{M0}^2 \right). \quad (2-14)$$

因此，若在每个时刻均取该时刻视线角方向作为  $x$  轴，则可得目标静止时在每个时刻的剩余攻击时间的估计值为：

$$t_{go} = \frac{R}{V_M} \left( 1 + \frac{1}{2(2N-1)} \sigma_M^2 \right). \quad (2-15)$$

针对机动目标对式 (2-15) 作出进一步修正，采用虚拟目标的思路，将  $V_T$  和  $a_T$  对弹目相对运动的影响均投影到弹目视线角方向上，并修正由目标机动引起的航向角和视线角变化，将目标转化为相对静止状态。

首先得到目标机动引起的目标航向角和视线角的变化分别为：

$$\hat{\sigma}_T = \frac{a_T}{V_T} \cdot \hat{t}_{go}^p, \quad \hat{\lambda} = \frac{V_T \sin \sigma_T}{R} \cdot \hat{t}_{go}^p, \quad (2-16)$$

其中  $\hat{t}_{go}^p$  表示前一时刻对  $t_{go}$  的估计值，修正后的剩余攻击时间估计为：

$$\hat{t}_{go} = \frac{R}{V_{MT}} \left( 1 + \frac{1}{2(2N-1)} \sigma_M^2 \right) \quad (2-17)$$

其中  $V_{MT} = V_M - V_T \cos(\sigma_T + \hat{\sigma}_T - \lambda - \hat{\lambda})$ 。

## 2.4 最优制导律

除了预计攻击时间最小化以外，由于导弹在攻击过程中大部分时间处于无动力滑翔阶段，在机动追踪目标时会不断消耗能量，因此为了保证击中目标前导弹动能不会耗尽，需要追求导弹机动程度最小化。实际中导弹系统具有高阶动态特性和非线性，难以利用最优控制原理求得闭环解，但对于只考虑导弹低阶系统特性和二次型性能指标取极小的最优控制问题，可以求得闭环解。本节将在导弹在2.3小节估计剩余攻击时间的基础上，基于最优控制原理，设计使得导弹机动最小化的最优制导律。

首先假设目标不机动，即  $a_T = 0$  将2.2节建立的弹目相对运动模型改写为惯性系下的状态方程形式：

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad (2-18)$$

$$\mathbf{A} = \begin{bmatrix} 0 & \mathbf{I}_2 & 0 \\ 0 & 0 & \mathbf{I}_2 \\ 0 & 0 & -\frac{1}{\tau} \mathbf{I}_2 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{\tau} \mathbf{I}_2 \end{bmatrix}, \quad (2-19)$$

其中状态向量为  $\mathbf{x} = [\mathbf{R}^T, \dot{\mathbf{R}}^T, \mathbf{a}_M^T]^T$ ，各状态分量分别为对应变量的向量形式， $\mathbf{I}_2$  为二阶单位矩阵， $\tau$  为导弹的等效时间常数。

设二次型的性能指标为

$$\min J = \frac{1}{2} \int_{t_0}^{t_f} \mathbf{u}^T \mathbf{u} dt, \quad (2-20)$$

$$\text{s.t. } \mathbf{R}(t_f) = 0. \quad (2-21)$$

其中  $t_0$  和  $t_f$  分别表示制导开始和结束的时间, 式 (2-20) 可减小机动能量的消耗, 有利于保持弹道平直, 减小损耗增大射程。式 (2-21) 可约束末端脱靶量为 0。

上述最优控制问题可求解闭环最优控制率形式为

$$u = (\mathbf{B}^T \mathbf{P} k^{-1}) \mathbf{M}, \quad (2-22)$$

其中  $k$  是当  $u = 0$  时的预计末端脱靶量, 可由下式求得:

$$\mathbf{M} = \mathbf{P}^T \mathbf{X}. \quad (2-23)$$

由下列微分方程

$$\dot{\mathbf{P}} + \mathbf{A}^T \mathbf{P} = 0, \quad \mathbf{P}(t_f) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad (2-24)$$

可求得  $\mathbf{P}$  的闭环表达式为

$$\mathbf{P} = \begin{bmatrix} 1 \\ t_{go} \\ \tau^2(e^{-T} + T - 1) \end{bmatrix}, \quad (2-25)$$

其中  $t_{go}$  是预计剩余攻击时间,  $t_{go} = t_f - t$ ,  $T = t_{go}/\tau$ 。

由

$$k = - \int_t^{t_f} (\mathbf{P}^T \mathbf{G} \mathbf{G}^T \mathbf{P}) dt, \quad (2-26)$$

可求得  $k$  等于

$$k = \tau^3 \left( \frac{1}{2} e^{-2T} + 2T e^{-T} - \frac{T^3}{3} + T^2 - T - \frac{1}{2} \right). \quad (2-27)$$

将式 (2-25) 和式 (2-27) 代入式 (2-22) 和式 (2-23) 求得最优制导律闭合解为

$$u = \frac{N'}{t_{go}^2} [\mathbf{R} + \dot{\mathbf{R}} t_{go} - \tau^2 (e^{-T} - 1 + T) \mathbf{a}_M], \quad (2-28)$$

其中  $N'$  为扩展比例系数:

$$N' = T^2 (e^{-T} - \mathbf{I}_2 + T) \left( -\frac{1}{2} e^{-2T} - 2T e^{-T} + \frac{1}{3} T^3 - T^2 + T + \frac{1}{2} \mathbf{I}_2 \right)^{-1}. \quad (2-29)$$

在目标机动的情况下, 可对式 (2-28) 进行修正, 限于篇幅, 这里不再展开, 直接给出对于机动目标的最优制导律:

$$u = \frac{N'}{t_{\text{go}}^2} [\mathbf{R} + \dot{\mathbf{R}} t_{\text{go}} - \tau^2 (e^{-T} - 1 + T) \mathbf{a}_M - \frac{t_{\text{go}}^2}{2} \mathbf{a}_T]. \quad (2-30)$$

最优性能指标  $J^*$  为

$$J^* = \frac{1}{2} \mathbf{X}^T(t_0) \mathbf{P}(t_0) \mathbf{P}^T(t_0) \mathbf{X}(t_0). \quad (2-31)$$

## 2.5 任务分配模型建立

假设我方有  $n_m$  枚导弹, 其集合表示为  $\mathcal{M} := \{\mathcal{M}_1, \dots, \mathcal{M}_{n_m}\}$ ; 战场上有  $n_t$  个目标需要攻击, 其集合表示为  $\mathcal{T} := \{\mathcal{T}_0, \mathcal{T}_1, \dots, \mathcal{T}_{n_t}\}$ , 其中  $\mathcal{T}_0$  表示空目标。本文只考虑导弹数量不少于目标数量, 即  $n_m \geq n_t$  的情况。每枚导弹可根据自身能力、所处环境等因素选择部分目标作为可攻击对象, 设  $\mathcal{A}_i \subset \mathcal{T}$  表示第  $i$  枚导弹的可选目标集。特别地, 为后续叙述方便, 假设  $\mathcal{T}_0 \in \mathcal{A}_i, i = 1, \dots, n_m$ , 即所有导弹总是可以选择不攻击任何目标。<sup>①</sup> 导弹  $i$  从自己的可选目标集  $\mathcal{A}_i$  中选出自己的目标  $a_i$ , 则数组  $a = (a_1, \dots, a_{n_m}) \in \mathcal{A}$  组成了一个分配解。设  $S(a) = \{S_1, \dots, S_{n_t}\}$ , 其中  $S_j$  表示分配解  $a$  下选择目标  $j$  的导弹集合,  $s_j$  表示  $S_j$  中的导弹个数, 即  $s_j = |S_j|, j = 1, \dots, n_t$ 。

选取一个关于分配解  $a$  的函数  $U_g(a)$  作为全局效用函数, 建立任务分配模型:

$$\max_{a \in \mathcal{A}} U_g(a) = \sum_{\mathcal{T}_j \in \mathcal{T}} U_{\mathcal{T}_j}(a) \quad (2-32)$$

$$\text{s.t. } s_j \leq b_{\max}^{(j)}, j = 1, \dots, n_t \quad (2-33)$$

$$s_j \geq 1, j = 1, \dots, n_t, \quad (2-34)$$

其中  $b_{\max}^{(j)}$  表示选择目标  $j$  的导弹的最大数量和最小数量, 且有  $b_{\max}^{(j)} \geq 1$ 。因此式 (2-33) 表示目标  $j$  不得被超过  $b_{\max}^{(j)}$  枚导弹攻击, 式 (2-34) 代表每个目标必须有导弹攻击, 但由于实际中往往使用多于目标数量的导弹攻击, 因此在式 (2-34) 的约束下, 式 (2-34) 自然成立, 因此可将上述问题进一步简化为

<sup>①</sup> 显然, 根据实际场景以及约束条件可知导弹不攻击任何目标是不符合实际要求的, 本文在此作出的假设仅仅是为了后面章节叙述和论证需要, 无任何实际意义, 对问题的论述和求解也无任何影响。



$$\max_{a \in A} U_g(a) = \sum_{\mathcal{T}_j \in \mathcal{T}} U_{\mathcal{T}_j}(a) \quad (2-35)$$

$$\text{s.t. } s_j \leq b_{\max}^{(j)}, j = 1, \dots, n_t. \quad (2-36)$$

$U_{\mathcal{T}_j}(a)$  表示与目标  $\mathcal{T}_j$  有关的任务效用，为了体现预计剩余攻击时间最小和导弹机动能量消耗最小的目标，利用2.3小节和2.4小节的内容，将任务效用函数设计为：

$$U_{\mathcal{T}_j}(a) = \max\{0, r_j - w_1 \max_{\mathcal{M}_i \in S_j} t_{ij} - w_2 C_{\mathcal{T}_j}\}, \quad (2-37)$$

其中， $w_1$  和  $w_2$  为权重值； $r_j$  是攻击目标  $\mathcal{T}_j$  可获得的奖励值，本文假设该值仅与目标属性有关，与攻击的导弹无关； $t_{ij}$  表示导弹  $\mathcal{M}_i$  攻击目标  $\mathcal{T}_j$  的预计剩余攻击时间。 $C_{\mathcal{T}_j}$  表示导弹攻击目标  $\mathcal{T}_j$  的成本函数，定义为：

$$C_{\mathcal{T}_j} = \sum_{\mathcal{M}_i \in S_j} J_{ij}, \quad (2-38)$$

其中  $J_{ij}$  为导弹  $\mathcal{M}_i$  攻击目标  $\mathcal{T}_j$  的预计机动能量消耗可由式 (2-31) 得到。

## 2.6 本章小结

本章基于攻击时间最小化和导弹机动消耗最小化的原则建立了空空导弹任务分配模型。首先建立了弹目相对运动模型，对导弹和目标的运动方程进行建模。接着基于该相对运动模型估计剩余攻击时间，并设计导弹遵循最优制导律飞行已获得最小机动消耗。最后，本章定义了全局效用函数和任务效用函数的概念，建立了导弹与目标的任务分配模型，为后续求解空空导弹的任务分配问题打下基础。



### 第三章 基于势博弈模型的分布式任务分配

#### 3.1 引言

本章使用第二章中建立的空空导弹任务分配模型，设计了面向任务分配问题的势博弈模型（Potential Game, PG）框架。本章首先介绍了势博弈模型概念，并选择合适的函数作为智能体效用，初步建立了势博弈模型；接着针对任务分配中的约束问题，使用 Lagrange 乘子法对原始势博弈模型进行改进，并证明了模型的可行性与性能下界；然后，针对改进的势博弈模型，使用多种均衡求解算法进行模型求解；最后，通过仿真与对比实验验证了模型与算法的有效性。

#### 3.2 势博弈模型

在任务分配问题框架下，当所有导弹依据最大化自身效用的原则选择的目标不再发生变化时，则称所有导弹达到了均衡状态。博弈论中一个经典的均衡状态是纯纳什均衡，它表示一个任务分配解  $a^* = (a_1^*, \dots, a_{n_m}^*)$ ，满足任意一个导弹不能通过独自改变任务提高自身效用。在具体阐述纳什均衡和势博弈概念前，除第二章使用的概念外，仍需要引入以下概念和符号。设  $a_{-i}$  表示除了导弹  $\mathcal{M}_i$  以外的其他导弹的分配集合，即

$$a_{-i} = (a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_{n_m}),$$

利用该符号可将一组分配解表示为  $(a_i, a_{-i})$ 。此外，可类似地定义  $\mathcal{A}_{-i}$  为

$$\mathcal{A}_i := \mathcal{A}_1 \times \dots \times \mathcal{A}_{i-1} \times \mathcal{A}_{i+1} \times \dots \times \mathcal{A}_{n_m}.$$

设  $U_{\mathcal{M}_i}(a)$  或  $U_{\mathcal{M}_i}(a_i, a_{-i})$  表示导弹  $\mathcal{M}_i$  在分配  $a$  下的智能体效用，由此可得到纯纳什均衡的定义：

**定义 3.1 (纯纳什均衡)** 若一个分配解  $a^*$  满足

$$U_{\mathcal{M}_i}(a_i^*, a_{-i}^*) = \max_{a_i \in \mathcal{A}_i} U_{\mathcal{M}_i}(a_i, a_{-i}^*), \quad \forall \mathcal{M}_i \in \mathcal{M}. \quad (3-1)$$

则称  $a^*$  是一个纯纳什均衡解。

当所有导弹达到纯纳什均衡时，全局效用未必实现最大化；另一方面，在一些博弈场景下纯纳什均衡未必存在。因此，引入势博弈概念

**定义 3.2 (有序势博弈与势博弈)** 设一个博弈模型中的智能体效用函数为  $U_{\mathcal{M}_i}(a)$ ,  $\mathcal{M}_i \in \mathcal{M}$ , 如果存在一个全局势函数  $\phi(a) : \mathcal{A} \mapsto \mathbb{R}$ , 满足对任意智能体  $\mathcal{M}_i \in \mathcal{M}$ , 任意  $a_{-i} \in \mathcal{A}_{-i}$ , 智能体  $\mathcal{M}_i$  的任意两个分配  $a'_i, a''_i \in \mathcal{A}_i$ , 有

$$U_{\mathcal{M}_i}(a'_i, a_{-i}) - U_{\mathcal{M}_i}(a''_i, a_{-i}) > 0 \Leftrightarrow \phi(a'_i, a_{-i}) - \phi(a''_i, a_{-i}) > 0, \quad (3-2)$$

则该博弈模型是一个有序势博弈模型。在此基础上, 若该全局势函数进一步满足:

$$U_{\mathcal{M}_i}(a'_i, a_{-i}) - U_{\mathcal{M}_i}(a''_i, a_{-i}) = \phi(a'_i, a_{-i}) - \phi(a''_i, a_{-i}). \quad (3-3)$$

则称该博弈模型为势博弈模型。若策略集  $\mathcal{A}$  是有限的, 则称该势博弈为有限势博弈。

由势博弈的定义可知, 势博弈模型中的智能体效用变化会等地反映在全局效用, 而在有序势博弈上, 智能体效用与全局效用总是同向变化。易知势博弈是一种特殊的有序势博弈。有限势博弈模型有两点性质:

- (1) 有限势博弈至少存在一个纯纳什均衡;
- (2) 有限势博弈具有有限优化性质 (finite improvement property, FIP)。

性质 (1) 保证了纯纳什均衡的存在性, 因此对于任务分配问题, 至少可以得到一个确定的解; 性质 (2) 保证了任意智能体单方面提高自己效用的行为可以在有限时间内收敛到纯纳什均衡, 从而保证了优化算法可在有限时间内收敛。

### 3.3 智能体效用函数

使用势博弈模型解决任务分配问题对导弹的智能体效用函数提出的要求是, 满足

$$U_{\mathcal{M}_i}(a'_i, a_{-i}) - U_{\mathcal{M}_i}(a''_i, a_{-i}) = U_g(a'_i, a_{-i}) - U_g(a''_i, a_{-i}), \quad (3-4)$$

使得所有导弹形成一个势博弈模型, 其中  $U_g(a)$  即为式 (2-32) 中定义的全局效用函数, 因此在求解分配时所有导弹只需考虑最大化智能体效用便可实现全局效用最大化的目标。

一个显然而直接的思路是使用一致利益效用 (Identical Interest Utility, IIU), 其直接将全局效用作为智能体效用, 但 IIU 的缺点在于需要计算所有任务效用后得到全局效用, 因此本质上仍是集中式的方法。对 IIU 进行改进可以得到更分布化的有限区域效用 (Range-Restricted Utility, RRU), RRU 将智能体效用定义为所有该智能体参与的任务的效用之和, RRU 可以使得智能体组成势博弈模型, 但该

模型的纳什均衡未必是最优的。此外，IIU 和 RRU 有一个共同的问题在于如果存在大量的智能体同时参与大量的任务，由于 IIU 和 RRU 都是对一定数量的任务效用求和，此时一个智能体对任务的贡献微乎其微，即使它单方面改变决策，对该任务的效用改变可能很小，从而反映在智能体效用函数上的变化也很小。这对智能体的寻优和学习带来了很大难度。对此问题作出的改进是等份额效用（Equally Shared Utility, ESU），ESU 用智能体参与的任务效用除以参与该任务的智能体总数，可以有效消除智能体数量带来的效用稀释作用，但 ESU 的局限在于只有当智能体之间不存在区别时才可以组成势博弈模型。

为了克服 IIU 和 RRU 的缺陷，且不局限于 ESU 的应用场景，本节采用的是美好生活效用函数（Wonderful Life Utility, WLU），WLU 将智能体效用函数定义为每个智能体对全局效用的边际贡献程度，其具体定义为：

$$U_{\mathcal{M}_i}(a_i, a_{-i}) = U_g(a_i, a_{-i}) - U_g(a_0, a_{-i}). \quad (3-5)$$

由全局效用函数的定义式（2-32）可将式（3-5）改写为

$$U_{\mathcal{M}_i}(a_i, a_{-i}) = U_{\mathcal{T}_j}(a_i, a_{-i}) - U_{\mathcal{T}_j}(a_0, a_{-i}), \quad \text{if } a_i = \mathcal{T}_j. \quad (3-6)$$

由式（3-6）可知，WLU 假设其他智能体不改变目标，智能体效用只与智能体自身的决策有关，因此 WLU 排除了冗余信息，比 IIU 和 RRU 优化能力更强。另外，下面的命题表明，使用 WLU 的博弈模型一定是势博弈模型。

**命题 3.1 (WLU 可行性)** 智能体效用函数为式（3-6）的智能体集合组成的博弈模型是一个势博弈模型，且其势函数就是全局效用函数  $U_g(a)$ 。

**证明** 只需证明 WLU 满足式（3-4）即可。设  $a_i, a'_i \in \mathcal{A}_i$ ，则有

$$\begin{aligned} & U_{\mathcal{M}_i}(a_i, a_{-i}) - U_{\mathcal{M}_i}(a'_i, a_{-i}) \\ &= U_g(a_i, a_{-i}) - U_g(a_0, a_{-i}) - [U_g(a'_i, a_{-i}) - U_g(a_0, a_{-i})] \\ &= U_g(a_i, a_{-i}) - U_g(a'_i, a_{-i}) \end{aligned} \quad \square$$

### 3.4 约束条件处理方法

上述势博弈框架和智能体效用函数设计下，对于智能体选择任务并没有做出任何限制和约束，因此需要利用约束条件式（2-36）将势博弈的纳什均衡限制在可行解中。本节提出了基于 Lagrange 乘子法将有约束问题转化为无约束问题，并

证明了在满足一定的条件下，新的无约束问题下获得的纳什均衡解均为原问题的可行解。

**Lagrange** 乘子法是一种典型的处理含约束优化问题的方法，其将含约束问题转化为无约束问题，且可证明生成的新的无约束问题的最优解就是原问题的最优解。在势博弈模型下，结合全局效用函数的定义式 (2-35) 以及约束条件式 (2-33)，将式 (2-37) 任务效用函数  $U_{\mathcal{T}_i}(a)$  修正为：

$$\tilde{U}_{\mathcal{T}_j}(a) = U_{\mathcal{T}_j}(a) + \lambda [b_{\max}^{(j)} - s_j]^-, \quad (3-7)$$

其中  $[x]^+ = \min\{x, 0\}$ ， $s_j$  是第二章中定义的目标同为  $\mathcal{T}_j$  的导弹数量。由式 (2-32) 得到修正后的全局效用函数为

$$\tilde{U}_g(a) = U_g(a) + \lambda \sum_{j=1}^{n_t} [b_{\max}^{(j)} - s_j]^- \quad (3-8)$$

再利用式 (3-6) 提出的 WLU 效用函数计算修正后的智能体效用函数 (Modified WLU, MWLU) 为

$$\begin{aligned} \tilde{U}_{\mathcal{M}_i}(a_i, a_{-i}) &= \tilde{U}_g(a_i, a_{-i}) - \tilde{U}_g(a_0, a_{-i}) \\ &= \tilde{U}_{\mathcal{T}_j}(a_i, a_{-i}) - \tilde{U}_{\mathcal{T}_j}(a_0, a_{-i}), \quad \text{if } a_i = \mathcal{T}_j. \end{aligned} \quad (3-9)$$

下面需要验证 MWLU 的有效性，主要分为三部分：(1) 满足 MWLU 效用的智能体能否组成势博弈模型；(2) 该势博弈模型下得到的纳什均衡是否都满足原问题的约束；(3) 该势博弈模型下得到的纳什均衡解的最优性。

### 3.4.1 势博弈成立条件

为了验证 MWLU 函数的有效性，首先需要验证式 (3-9) 定义的智能体效用函数是否可以组成一个势博弈模型。设智能体效用函数为 MWLU 的智能体组成的组成的博弈模型为  $\mathcal{G} = \{\mathcal{M}, \mathcal{A}, U_{\mathcal{M}}(a)\}$ ，实际上，可以得到以下结论：

**命题 3.2 (势博弈成立条件)**  $\mathcal{G}$  是一个势博弈模型，其势函数为式 (3-8) 定义的全局效用函数。

**证明** 只需证明全局效用函数  $\tilde{U}_g(a)$  与式 (3-9) 定义的智能体效用函数  $\tilde{U}_{\mathcal{M}_i}(a_i, a_{-i})$  满足式 (3-4) 即可。对任意  $a'_i, a''_i \in \mathcal{A}_i$ ， $a' = (a'_i, a_{-i})$ ， $a'' = (a''_i, a_{-i})$ ，

且  $S', S''$  分别为  $a'$  和  $a''$  对应的智能体分配集合,  $s'_j$  和  $s''_j$  分别为  $S'$  和  $S''$  中目标为  $\mathcal{T}_j$  的智能体数量。假设  $a'_i = \mathcal{T}_j$ ,  $a''_i = \mathcal{T}_k$ ,  $j \neq k$ , 则有

$$\begin{aligned} & \tilde{U}_{\mathcal{M}_i}(a'_i, a_{-i}) - \tilde{U}_{\mathcal{M}_i}(a''_i, a_{-i}) \\ &= [\tilde{U}_g(a'_i, a_{-i}) - \tilde{U}_g(a_0, a_{-i})] - [\tilde{U}_g(a''_i, a_{-i}) - \tilde{U}_g(a_0, a_{-i})] \\ &= \tilde{U}_g(a'_i, a_{-i}) - \tilde{U}_g(a''_i, a_{-i}) \end{aligned} \quad (3-10)$$

所以满足 MWLU 效用的智能体可以组成一个势博弈模型, 且势函数为全局效用函数  $\tilde{U}_g(a)$ 。□

### 3.4.2 纯纳什均衡可行性分析

在验证势博弈模型成立之后, 下一步需要验证该势博弈模型所得到的纯纳什均衡是否是原问题的可行解, 此处有以下结论:

**命题 3.3 (纳什均衡可行性)** 当  $\lambda$  满足  $\lambda > \lambda_0$  时, 若分配策略  $a$  是  $\mathcal{G}$  的一个纯纳什均衡, 则  $a$  一定满足约束条件 (2-33)。其中

$$\lambda_0 = \max_{\substack{\mathcal{M}_i \in \mathcal{M}, \mathcal{T}_j \in \mathcal{T} \\ s_j > b_{\max}^{(j)}}} \left\{ \frac{U_{\mathcal{T}_j}(a_i, a_{-i}) - U_{\mathcal{T}_j}(a'_i, a_{-i})}{s_j - b_{\max}^{(j)}} \right\} \quad (3-11)$$

**证明** 设  $a^* = (a_i^*, a_{-i}^*)$  是势博弈模型  $\mathcal{G}$  的一个纯纳什均衡, 但不满足约束条件 (2-33), 即存在  $1 \leq j \leq n_t$ ,  $s_j^* > b_{\max}^{(j)}$ 。对于参与任务  $\mathcal{T}_j$  的智能体之一  $\mathcal{M}_k$ , 设  $a_k^* = \mathcal{T}_j$ ,  $a'_k = \mathcal{T}_l$ ,  $\forall l \neq j$ , 则有

$$\begin{aligned} & \tilde{U}_{\mathcal{M}_k}(a_k^*, a_{-k}^*) - \tilde{U}_{\mathcal{M}_k}(a'_k, a_{-k}^*) \\ &= \tilde{U}_{\mathcal{T}_j}(a_k^*, a_{-k}^*) - \tilde{U}_{\mathcal{T}_l}(a'_k, a_{-k}^*) \\ &= U_{\mathcal{T}_j}(a_k^*, a_{-k}^*) - \lambda(s_j^* - b_{\max}^{(j)}) - [U_{\mathcal{T}_l}(a'_k, a_{-k}^*) + \lambda(b_{\max}^{(l)} - s'_l)] \\ &\leq U_{\mathcal{T}_j}(a_k^*, a_{-k}^*) - U_{\mathcal{T}_l}(a'_k, a_{-k}^*) - \lambda(s_j^* - b_{\max}^{(j)}) \\ &\leq \max_{\mathcal{T}_j \in \mathcal{T}} \{U_{\mathcal{T}_j}(a_i^*, a_{-i}^*) - U_{\mathcal{T}_j}(a'_i, a_{-i}^*)\} - \lambda_0(s_j^* - b_{\max}^{(j)}) \end{aligned} \quad (3-12)$$

由  $\lambda_0$  的定义易知  $\tilde{U}_{\mathcal{M}_{k_0}}(a_{k_0}^*, a_{-k_0}^*) - \tilde{U}_{\mathcal{M}_{k_0}}(a'_{k_0}, a_{-k_0}^*) \leq 0$ , 这与  $a^*$  是纯纳什均衡的条件 (3-1) 矛盾, 因此  $a^*$  必然满足约束条件。□

### 3.4.3 纳什均衡最优性

在命题3.3下，势博弈模型  $\mathcal{G}$  的纳什均衡均为原问题的可行解，因此最后需要考察这些纳什均衡解的最优性。本节选择使用无秩序代价（Price of Anarchy, PoA）作为衡量纳什均衡性能的指标。设纯纳什均衡分配解为  $a^*$ ，全局最优分配解为  $a^{\text{opt}}$ ，则 PoA 的定义为全局最优效用和纳什均衡效用的最小值之比：

$$\text{PoA}(\mathcal{G}) = \frac{U_g(a^{\text{opt}})}{\min_{a^*} U_g(a^*)} \quad (3-13)$$

由 PoA 的定义可知，PoA 越大，代表最差情况下纯纳什均衡效用越小，即意味着该势博弈模型的性能越差。关于本节建立的势博弈模型  $\mathcal{G}$  的 PoA，我们可以用以下命题得到 PoA 的上界，从而保证了势博弈模型性能的下界。

**命题 3.4** 势博弈模型  $\mathcal{G}$  的 PoA 上界为

$$\text{PoA}(\mathcal{G}) \leq 1 + \eta \quad (3-14)$$

其中

$$\eta = \max_{\substack{\forall a, a' \in \mathcal{A} \\ \forall \mathcal{M}_i \in \mathcal{M}}} \left\{ \frac{\tilde{U}_g(a') - \tilde{U}_g(a'_i, a_{-i})}{\tilde{U}_g(a)} \right\} \quad (3-15)$$

**证明** 设  $a^{\text{opt}}$  是全局最优分配解， $a^*$  是一纯纳什均衡分配解。由纯纳什均衡的定义式 (3-1) 可得

$$\tilde{U}_{\mathcal{M}_i}(a_i^*, a_{-i}^*) \geq \tilde{U}_{\mathcal{M}_i}(a_i^{\text{opt}}, a_{-i}^*), \quad \forall \mathcal{M}_i \in \mathcal{M}. \quad (3-16)$$

将式 (3-4) 代入可得

$$\tilde{U}_g(a_i^*, a_{-i}^*) \geq \tilde{U}_g(a_i^{\text{opt}}, a_{-i}^*). \quad (3-17)$$

不等式右侧可以写为

$$\tilde{U}_g(a_i^{\text{opt}}, a_{-i}^*) = \tilde{U}_g(a_i^{\text{opt}}, a_{-i}^{\text{opt}}) - [\tilde{U}_g(a_i^{\text{opt}}, a_{-i}^{\text{opt}}) - \tilde{U}_g(a_i^{\text{opt}}, a_{-i}^*)], \quad (3-18)$$

所以有

$$\tilde{U}_g(a^*) \geq \tilde{U}_g(a^{\text{opt}}) - [\tilde{U}_g(a^{\text{opt}}) - \tilde{U}_g(a_i^{\text{opt}}, a_{-i}^*)], \quad (3-19)$$



两边同时除以  $\tilde{U}_g(a^*)$ ,

$$\begin{aligned} \frac{U_g(a^{\text{opt}})}{U_g(a^*)} &\leq 1 + \frac{\tilde{U}_g(a^{\text{opt}}) - \tilde{U}_g(a_i^{\text{opt}}, a_{-i}^*)}{\tilde{U}_g(a^*)} \\ &\leq 1 + \max_{\substack{\forall a, a' \in \mathcal{A} \\ \forall \mathcal{M}_i \in \mathcal{M}}} \left\{ \frac{\tilde{U}_g(a') - \tilde{U}_g(a_i', a_{-i})}{\tilde{U}_g(a)} \right\} \\ &\triangleq 1 + \eta \end{aligned} \quad (3-20)$$

因此

$$\text{PoA}(\mathcal{G}) \leq 1 + \eta \quad (3-21)$$

□

### 3.5 均衡求解

#### 3.5.1 虚拟博弈算法

虚拟博弈算法 (Fictitious Play, FP) 是一种经典的学习算法, 最初被用来求解零和博弈中的纳什均衡, 但也被提出可以用于多智能体系统的学习算法。在本节, 虚拟博弈算法被视为智能体选择目标的机制。在每个时刻  $k$ , 智能体  $i$  计算自身的经验频率分布  $q_i(k)$ , 并根据该分布选择下一步最优决策。智能体经验频率分布定义为

$$q_i^j(k) = \frac{1}{k} \sum_{t=0}^{k-1} I\{a_i = A_i^j\}, \quad A_i^j \in \mathcal{A}_i, j = 1, \dots, |\mathcal{A}_i|, \quad (3-22)$$

其中  $I\{\cdot\}$  为指示函数,  $|\mathcal{A}_i|$  表示目标集  $\mathcal{A}_i$  的势。

智能体  $i$  在 FP 下选择目标的策略是假设其他智能体会独立地根据各自的经验频率分布随机选择一个目标。在此假设下, 可计算出智能体  $i$  选择目标  $A_i^j$  的期望效用为

$$\begin{aligned} U_i(A_i^j, q_{-i}(k)) &= \mathbb{E}_{a_{-i}}[U_{\mathcal{M}_i}(A_i^j, a_{-i})] \\ &= \sum_{a_{-i} \in \mathcal{A}_{-i}} U_{\mathcal{M}_i}(A_i^j, a_{-i}) \prod_{a_j \in \mathcal{A}_{-i}} q_j^{a_j}(k), \end{aligned} \quad (3-23)$$

其中  $q_{-i}(k) := \{q_1(k), \dots, q_{i-1}(k), q_{i+1}(k), \dots, q_{n_m}(k)\}$ 。因此, 智能体  $i$  在  $k$  时刻选择的最优目标  $a_i(k)$  为

$$a_i(k) \in \arg \max_{\alpha \in \mathcal{A}_i} U_i(\alpha, q_{-i}(k)), \quad (3-24)$$

若同时存在多个最优目标，则在多个最优目标中任选一个。

FP 的优点在于除了极少数特殊情况，根据经验频率分布选出的目标分配解几乎都会收敛到纳什均衡。但 FP 的缺点也是显然的，每个智能体在作出决策时需要知道所有智能体的经验频率分布，且计算出所有组合情况下的效用的期望，因此 FP 的计算时间和计算复杂度会随着智能体数量的增加而急剧增加。

为了提高 FP 的可行性，联合策略虚拟博弈 (Joint Strategy Fictitious Play, JSFP) 在 FP 的基础上做出了改进。JSFP 最主要的改进在于在计算不同目标选择的效用时，取消了他智能体选择目标的独立性条件，从而在计算经验频率分布时，计算的是一个分配组合  $\mathbf{a}$  的经验频率而不是某个特定目标的频率。在 JSFP 下，联合经验频率  $z(\bar{\mathbf{a}}, k)$  的计算公式为

$$z(\bar{\mathbf{a}}, k) = \frac{1}{k} \sum_{i=0}^{k-1} I\{a(k) = \bar{\mathbf{a}}\}, \quad \bar{\mathbf{a}} \in \mathcal{A}, \quad (3-25)$$

类似地可定义出  $z_{-i}(\bar{\mathbf{a}}, k)$  的定义为

$$z_{-i}(\bar{\mathbf{a}}, k) = \frac{1}{k} \sum_{i=0}^{k-1} I\{a_{-i}(k) = \bar{\mathbf{a}}_{-i}\}, \quad \bar{\mathbf{a}} \in \mathcal{A}. \quad (3-26)$$

因此此时智能体  $i$  选择目标  $\bar{a}_i$  的期望效用是

$$U_i(\bar{a}_i, k) = \sum_{a_{-i} \in \mathcal{A}_{-i}} U_{\mathcal{M}_i}(\bar{a}_i, a_{-i}) z_{-i}(\bar{\mathbf{a}}_{-i}, k) \quad (3-27)$$

JSFP 第二个改进的地方在于，虽然式 (3-27) 似乎仍需要计算所有组合策略情况下的期望，但通过将式 (3-26) 代入式 (3-27) 可得

$$U_i(\bar{a}_i, z_{-i}(k)) = \frac{1}{k} \sum_{i=0}^{k-1} U_{\mathcal{M}_i}(\bar{a}_i, a_{-i}(k)). \quad (3-28)$$

上式表示在前  $k-1$  次迭代智能体  $i$  如果选择目标  $\bar{a}_i$ ，而其他智能体选择不变的情况下，智能体  $i$  可获得的平均效用。令  $\bar{U}_i^{\bar{a}_i}(k) = U_i(\bar{a}_i, z_{-i}(k))$ ，则式 (3-28) 可改写为迭代形式：

$$\bar{U}_i^{\bar{a}_i}(k+1) = \frac{k}{k+1} \bar{U}_i^{\bar{a}_i}(k) + \frac{1}{k+1} U_{\mathcal{M}_i}(\bar{a}_i, a_{-i}(k)). \quad (3-29)$$

JSFP 的算法流程如3-1所示。

---

**算法 3-1 JSFP 算法流程**


---

**Input:**  $\mathcal{M}, n_m, \mathcal{T}, n_t, \mathcal{A}$   
**Output:** 均衡解  $a^*$

- 1 随机初始化分配解  $a$ ;
- 2  $\bar{U}_i(0) \leftarrow \mathbf{0}_{1 \times n_t}, i = 1, \dots, n_t$ ;
- 3  $k \leftarrow 1$ ;
- 4 **while** *true* **do**
- 5     **for**  $i = 1 : n_m$  **do**
- 6         计算选择不同目标  $\bar{a}_i$  的  $U_{\mathcal{M}_i}(\bar{a}_i, a_{-i}(k))$ ;
- 7         使用式 (3-29) 更新  $\bar{U}_i(k)$ ;
- 8          $a(k) \leftarrow \arg \max_{a \in \mathcal{A}_i} \bar{U}_i(k)$ ;
- 9          $k \leftarrow k + 1$ ;
- 10     **end**
- 11 **end**

---

### 3.5.2 后悔匹配算法

后悔匹配算法 (Regret Matching, RM) 是借鉴了 JSFP 的思想, 与 JSFP 计算前  $k-1$  个时刻选择目标的平均效用不同, RM 计算的是前  $k-1$  次迭代不选择该目标将会损失的效用。沿用前面的符号, 在第  $k$  次迭代, 智能体  $i$  计算器对于目标  $\bar{a}_i^j$  平均后悔值为

$$R_{\mathcal{M}_i}^j(k) = \frac{1}{k-1} \sum_{t=1}^{k-1} [U_{\mathcal{M}_i}(\bar{a}_i^j, a_{-i}(t)) - U_{\mathcal{M}_i}(a(t))], j = 1, \dots, |\mathcal{A}_i|. \quad (3-30)$$

式 (3-30) 可改写为迭代形式

$$R_{\mathcal{M}_i}^j(k+1) = \frac{k-1}{k} R_{\mathcal{M}_i}^j(k) + \frac{1}{k} [U_{\mathcal{M}_i}(\bar{a}_i^j, a_{-i}(t)) - U_{\mathcal{M}_i}(a(t))], \quad k > 1. \quad (3-31)$$

在得到对每个目标的后悔值  $R_{\mathcal{M}_i}(k) = [R_{\mathcal{M}_i}^1(k), \dots, R_{\mathcal{M}_i}^{|\mathcal{A}_i|}(k)]$  后, 智能体计算当前选择目标的概率分布  $p_i(k)$

$$p_i(k) = \frac{[R_{\mathcal{M}_i}(k)]^+}{\mathbf{1}^T [R_{\mathcal{M}_i}(k)]^+}, \quad (3-32)$$

其中  $[x]^+ = \max\{x, 0\}$ ,  $\mathbf{1} = [1, 1, \dots, 1]$ 。

在 RM 的基础上进一步改进得到带有记忆和惰性的广义 RM 算法 (Generalized RM with Fading Memory and Inertia, GRMFMI)。将式 (3-31) 改写为

$$\tilde{R}_{\mathcal{M}_i}^j(k+1) = (1-\rho)\tilde{R}_{\mathcal{M}_i}^j(k) + \rho[U_{\mathcal{M}_i}(\bar{a}_i^j, a_{-i}(t)) - U_{\mathcal{M}_i}(a(t))], \quad j \in \{1, \dots, |\mathcal{A}_i|\}. \quad (3-33)$$

将惰性概念引入智能体  $i$  选择目标的概率分布, 智能体有  $1 - \alpha_i$  的概率会继续选择前一时刻的目标, 此时目标选择概率分布为

$$\tilde{p}_i(k) = \alpha_i P_i(\tilde{R}_{\mathcal{M}_i}(k)) + [1 - \alpha_i] \mathbf{v}^{a_i(k-1)}, \quad (3-34)$$

其中  $\mathbf{v}^{a_i(k-1)}$  表示第  $a_i(k-1)$  个元素为 1, 其余为 0 的  $|\mathcal{A}_i|$  维向量。  $P_i(x)$  是一个概率分布向量, 其每个分量的表达式为

$$P_i^l(x) = \begin{cases} \frac{e^{\frac{1}{\tau}x^l}}{\sum_{x^m > 0} e^{\frac{1}{\tau}x^m}} I\{x^l > 0\}, & \text{if } \mathbf{1}^T[x]^+ > 0, \\ \frac{1}{|\mathcal{A}_i|}, & \text{if } \mathbf{1}^T[x]^+ = 0. \end{cases}, \quad (3-35)$$

其中  $\tau > 0$  是一个参数, 当  $\tau$  较小时,  $\mathcal{M}_i$  会倾向于选择最大后悔值的目标, 当  $\tau$  较大时,  $\mathcal{M}_i$  会倾向于在有正后悔值的目标中随机选择一个。

综上所述, GRMFMI 的算法流程如算法3-2所示。

---

#### 算法 3-2 GRMFMI 算法流程

---

**Input:**  $\mathcal{M}, n_m, \mathcal{T}, n_t, \mathcal{A}$   
**Output:** 均衡解  $a^*$

- 1 初始化  $\tilde{R}_{\mathcal{M}_i}^j(0) = 0$ ;
- 2  $k \leftarrow 1$ ;
- 3 **while** true **do**
- 4     **for**  $i = 1 : n_m$  **do**
- 5         使用式 (3-31) 计算  $\tilde{R}_{\mathcal{M}_i}^j(k)$ ;
- 6         使用式 (3-34) 计算目标概率分布  $\tilde{p}_i(k)$ ;
- 7         根据  $\tilde{p}_i(k)$  随机选择目标  $a_i(k)$ ;
- 8          $k \leftarrow k + 1$ ;
- 9     **end**
- 10 **end**

---

### 3.5.3 空间自适应博弈算法

空间自适应博弈算法 (Spatial Adaptive Play, SAP) 原本是空间博弈中的一种自适应学习方法, 本节将针对任务分配问题的特点, 将其改造为适用于任务分配问题的 SAP 算法。

不同于之前几种算法, SAP 算法的特点是在每次迭代只等可能地随机选择一个智能体进行目标的选择, 其他智能体保持目标不变。被选中的智能体  $\mathcal{M}_i$  根据式计算其目标选择概率分布  $p_i(k)$

$$\max_{p_i(k) \in \Delta(|\mathcal{A}_i|)} p_i^T(k) \begin{bmatrix} U_{\mathcal{M}_i}(\bar{a}_i^{(1)}, a_{-i}(k-1)) \\ \vdots \\ U_{\mathcal{M}_i}(\bar{a}_i^{(|\mathcal{A}_i|)}, a_{-i}(k-1)) \end{bmatrix} + \tau \mathcal{H}(p_i(k)), \quad (3-36)$$

其中  $\mathcal{H}(\mathbf{x}) = -\mathbf{x}^T \log(\mathbf{x})$ ,  $x^l \neq 0, l = 1, \dots, |\mathcal{A}_i|$ 。

根据式 (3-36) 可求得  $p_i(k)$  的解析解形式为

$$p_i(k) = \sigma \left( \frac{1}{\tau} \begin{bmatrix} U_{\mathcal{M}_i}(\bar{a}_i^{(1)}, a_{-i}(k-1)) \\ \vdots \\ U_{\mathcal{M}_i}(\bar{a}_i^{(|\mathcal{A}_i|)}, a_{-i}(k-1)) \end{bmatrix} \right), \quad (3-37)$$

其中  $\sigma(\cdot)$  为 softmax 函数。

为了进一步提高 SAP 算法的效率, 还可对 SAP 算法进行改进得到部分选择 SAP (Selective SAP, sSAP) 算法。与 SAP 算法不同的是, sSAP 算法在计算  $p_i(k)$  时, 只在  $\mathcal{A}_i$  中挑选了  $n_i$  个目标 ( $1 \leq n_i < |\mathcal{A}_i|$ ) 作为下一步可选目标集, 其中包括了上一步选择的目标  $a_i(k-1)$ , 因此 sSAP 算法下  $p_i(k)$  的计算公式为

$$p_i(k) = \sigma \left( \frac{1}{\tau} \begin{bmatrix} U_{\mathcal{M}_i}(a_i(k-1)) \\ U_{\mathcal{M}_i}(\bar{a}_i^{(1)}, a_{-i}(k-1)) \\ \vdots \\ U_{\mathcal{M}_i}(\bar{a}_i^{(n_i)}, a_{-i}(k-1)) \end{bmatrix} \right), \quad (3-38)$$

SAP 算法和 sSAP 算法的流程如算法3-3所示。

## 3.6 仿真分析与对比

## 3.7 本章小结

---

算法 3-3 SAP 和 sSAP 算法流程

---

**Input:**  $\mathcal{M}, n_m, \mathcal{T}, n_t, \mathcal{A}$

**Output:** 均衡解  $a^*$

```
1  $k \leftarrow 1$ ;  
2 while true do  
3   随机选择一个智能体  $\mathcal{M}_i$ ;  
4    $n_i = |\mathcal{A}_i|$ ; // 使用 SAP 算法  
5   (或者) 确定  $n_i$  的值,  $1 \leq n_i < |\mathcal{A}_i|$ ; // 使用 sSAP 算法  
6   从  $\mathcal{A}_i$  中随机选择  $n_i$  个目标 (包括  $a_i(k-1)$ ) ;  
7   使用式 (3-37) 计算  $p_i(k)$ ;  
8   根据  $p_i(k)$  随机选择目标  $a_i(k)$ ;  
9    $k \leftarrow k + 1$ ;  
10 end
```

---

## 第四章 基于享乐联盟博弈的分布式任务分配

### 4.1 引言

使用享乐联盟博弈模型 (Hedonic Coalition Game, HCG) 解决任务分配问题的思路是将任务分配问题看作是智能体划分问题, 按照任务或目标的不同划分联盟, 每个智能体对不同联盟有着不同的偏好, 智能体会根据自己的偏好选择自己的联盟, 最终所有智能体在确定了自己的所属联盟后, 便得到了任务分配的解。本章将建立基于 HCG 的任务分配框架并提出对应求解算法, 并通过仿真对比实验验证其有效性、

### 4.2 享乐联盟博弈模型

在介绍 HCG 模型概念之前, 需要先引入几个概念和几点假设。

### 4.3 基于享乐联盟博弈模型的任务分配算法

#### 4.3.1 联盟效用函数

#### 4.3.2 协商协议

#### 4.3.3 算法收敛性分析

### 4.4 仿真分析与对比

#### 4.4.1 算法性能对比

### 4.5 本章小结





## **第五章 基于随机博弈模型的分布式动态任务分配**

### **5.1 引言**

### **5.2 随机博弈模型**

### **5.3 面向动态任务分配问题的随机博弈模型设计**

#### **5.3.1 效用函数估计**

#### **5.3.2 滚动策略集**

#### **5.3.3 分布式随机算法**

### **5.4 仿真对比与分析**

### **5.5 本章小结**



## 第六章 总结与展望



## 附录 A Maxwell Equations

选择二维情况，有如下的偏振矢量：

$$\mathbf{E} = E_z(r, \theta)\hat{\mathbf{z}}, \quad (\text{A-1a})$$

$$\mathbf{H} = H_r(r, \theta)\hat{\mathbf{r}} + H_\theta(r, \theta)\hat{\boldsymbol{\theta}}. \quad (\text{A-1b})$$

对上式求旋度：

$$\nabla \times \mathbf{E} = \frac{1}{r} \frac{\partial E_z}{\partial \theta} \hat{\mathbf{r}} - \frac{\partial E_z}{\partial r} \hat{\boldsymbol{\theta}}, \quad (\text{A-2a})$$

$$\nabla \times \mathbf{H} = \left[ \frac{1}{r} \frac{\partial}{\partial r} (r H_\theta) - \frac{1}{r} \frac{\partial H_r}{\partial \theta} \right] \hat{\mathbf{z}}. \quad (\text{A-2b})$$

因为在柱坐标系下， $\bar{\boldsymbol{\mu}}$  是对角的，所以 Maxwell 方程组中电场  $\mathbf{E}$  的旋度：

$$\nabla \times \mathbf{E} = i\omega \mathbf{B}, \quad (\text{A-3a})$$

$$\frac{1}{r} \frac{\partial E_z}{\partial \theta} \hat{\mathbf{r}} - \frac{\partial E_z}{\partial r} \hat{\boldsymbol{\theta}} = i\omega \mu_r H_r \hat{\mathbf{r}} + i\omega \mu_\theta H_\theta \hat{\boldsymbol{\theta}}. \quad (\text{A-3b})$$

所以  $\mathbf{H}$  的各个分量可以写为：

$$H_r = \frac{1}{i\omega \mu_r} \frac{1}{r} \frac{\partial E_z}{\partial \theta}, \quad (\text{A-4a})$$

$$H_\theta = -\frac{1}{i\omega \mu_\theta} \frac{\partial E_z}{\partial r}. \quad (\text{A-4b})$$

同样地，在柱坐标系下， $\bar{\boldsymbol{\epsilon}}$  是对角的，所以 Maxwell 方程组中磁场  $\mathbf{H}$  的旋度：

$$\nabla \times \mathbf{H} = -i\omega \mathbf{D}, \quad (\text{A-5a})$$

$$\left[ \frac{1}{r} \frac{\partial}{\partial r} (r H_\theta) - \frac{1}{r} \frac{\partial H_r}{\partial \theta} \right] \hat{\mathbf{z}} = -i\omega \bar{\boldsymbol{\epsilon}} \mathbf{E} = -i\omega \epsilon_z E_z \hat{\mathbf{z}}, \quad (\text{A-5b})$$

$$\frac{1}{r} \frac{\partial}{\partial r} (r H_\theta) - \frac{1}{r} \frac{\partial H_r}{\partial \theta} = -i\omega \epsilon_z E_z. \quad (\text{A-5c})$$

由此我们可以得到关于  $E_z$  的波函数方程：

$$\frac{1}{\mu_\theta \epsilon_z} \frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial E_z}{\partial r} \right) + \frac{1}{\mu_r \epsilon_z} \frac{1}{r^2} \frac{\partial^2 E_z}{\partial \theta^2} + \omega^2 E_z = 0. \quad (\text{A-6})$$



## 附录 B 绘制流程图

图 B-1 是一张流程图示意。使用 tikz 环境, 搭配四种预定义节点 (startstop、process、decision 和 io), 可以容易地绘制出流程图。

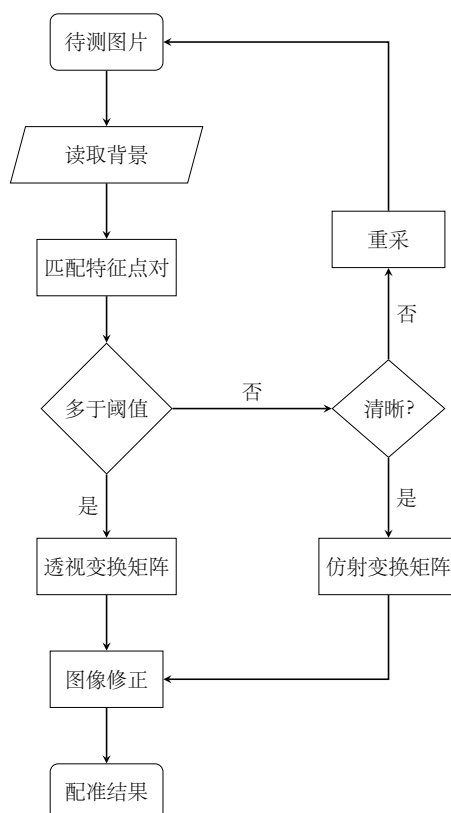


图 B-1 绘制流程图效果

Figure B-1 Flow chart





## 致 谢

感谢那位最先制作出博士学位论文 L<sup>A</sup>T<sub>E</sub>X 模板的交大物理系同学!

感谢 William Wang 同学对模板移植做出的巨大贡献!

感谢 @weijianwen 学长一直以来的开发和维护工作!

感谢 @sjtug 以及 @dyweb 对 0.9.5 之后版本的开发和维护工作!

感谢所有为模板贡献过代码的同学们, 以及所有测试和使用模板的各位同学!

感谢 L<sup>A</sup>T<sub>E</sub>X 和 SJTUThesis, 帮我节省了不少时间。



## 攻读学位期间发表（或录用）的学术论文

- [1] Chen H, Chan C T. Acoustic cloaking in three dimensions using acoustic metamaterials[J]. Applied Physics Letters, 2007, 91:183518.
- [2] Chen H, Wu B I, Zhang B, et al. Electromagnetic Wave Interactions with a Metamaterial Cloak[J]. Physical Review Letters, 2007, 99(6):63903.



## 攻读学位期间获得的科研成果

[1] 第一发明人,“永动机”,专利申请号 202510149890.0



## 个人简历

### 基本情况

某某，yyyy 年 mm 月生于 xxxx。

### 教育背景

- yyyy 年 mm 月至今，上海交通大学，博士研究生，xx 专业
- yyyy 年 mm 月至 yyyy 年 mm 月，上海交通大学，硕士研究生，xx 专业
- yyyy 年 mm 月至 yyyy 年 mm 月，上海交通大学，本科，xx 专业

### 研究兴趣

L<sup>A</sup>T<sub>E</sub>X 排版

### 联系方式

- 地址：上海市闵行区东川路 800 号，200240
- E-mail: xxx@sjtu.edu.cn