

# 算法赛道赛题说明与参赛指南

## 智慧仓库机器人调度：多智能体路径规划

### 一、 背景介绍

在智慧物流系统运作过程中，仓储机器人（见图 1）可自主完成从货架存储区域到拣货工作站的货物搬运作业。在此模式下，工作人员仅需在拣货工作站内，按照指令精准拣选对应品类与数量的商品即可。

相较于传统物流场景中“人到货”的作业模式，智慧仓储的这一运作方式能大幅提升商品拣选的效率，同时有效降低拣选误差，显著提高作业准确性。



图 1、仓储智能机器人示意

在智慧仓储系统的实际运作过程中，**如何在确保多台机器人有效避障的前提下**，以最短时间协同完成既定任务，已成为该领域的核心技术难点。

本届大赛算法赛道聚焦于“多智能体路径规划 (Multi-Agent Path Planning, MAPF)”

问题，通过对该问题的抽象建模，广泛汇聚行业技术专家与算法领域爱好者，共同探索并研发最优路径规划算法。**该算法的核心目标是 최소화所有机器人从起点抵达各自目标点的系统总运行时间。**

## 二、 比赛安排

算法赛道比赛分为初赛、复赛和决赛三个阶段，具体流程如下：

### （一）初赛

1. 参赛团队完成报名后，需下载指定数据与仿真器，在本地环境中运行模拟器并调试算法。
2. 赛题包含 4 个问题场景（地图），分别应对 10 台、50 台、100 台及 200 台仓储机器人；启动仿真 GUI 界面后，可通过右侧的“地图选择”栏查看 4 个场景的平面布局。
3. 初赛阶段，评审会不直接执行参赛团队设计的算法。选手需提交每个场景下所有时间步的机器人路由指令集合，并保存为 json 格式文件。
4. 初赛截止时间结束后，评审会将通过指令文件复现结果，计算成绩并完成排名。

### （二）复赛

1. 初赛成绩排名前 10 的参赛队伍将获得晋级决赛资格。
2. 晋级队伍需提交完整算法文件及 Docker 环境镜像，镜像需包含算法运行所需的依赖及运行环境。

3. 大赛组委会将在 10 个全新测试场景（矩形仓库平面，长宽范围为 50-100，机器人数量范围为 200-1000）中运行参赛团队提交的算法。
4. 测试用服务器的硬件配置为：不含 GPU 计算资源。
5. 算法实际执行过程中，**第 0 步响应时长需控制在 15 分钟以内（含），单步响应时长（从第 1 步开始）需控制在 0.5 秒以内（含）**；若超出该时长，该算法在对应测试场景下的成绩将判定为无效。

### （三）决赛

决赛以现场陈述为主，参赛队伍需提交陈述材料（PPT 格式，16:9 比例），展示程序运行过程、输出结果及结果分析，重点阐述算法的创新性与对企业实际问题的解决价值。演示时长不超过 10 分钟。由评委现场打分。

### （四）评分规则

最终评分 = 复赛分数\*60% + 决赛答辩分数\*40%

### （五）补充说明

1. 目前，比赛仅支持 Python 编程语言。
2. 排名规则中，若两支队伍的算法在系统总运行时间相同，则将进一步比较机器人总行驶距离，总行驶距离较短的队伍排名更优。

## 三、 算法赛题说明

初赛阶段设置 4 张不同尺寸的矩形地图，每张地图配置若干台仓储机器人，每台机器人都设有固定的起点坐标、起始方向与目标位置。

(一) 任务目标

在明确地图参数和任务信息的前提下，为所有机器人规划可行路径。

(二) 核心概念定义

概念	定义
模拟器	负责场景模拟及机器人状态更新
GUI	仓库平面与机器人行为的可视化 web 界面
地图	矩阵平面，由单元格组成
单元格	地图矩阵的最小构成要素，由二维坐标定位，一个单元格只能同时容纳一台机器人
机器人	可移动智能体，具有碰撞体积，在仓库平面中可沿上下左右 4 个方向移动，静止后可原地 360 度转向
预留队列	为每台机器人预设的未经过的路径单元格集合，模拟器将参考算法提供的预留队列具体执行机器人的运动

四、 模拟器与算法接口说明

(一) 控制系统

模拟器采用基于预留机制（reservation-based）的控制系统，其运作流程与核心逻辑

如下：

1. 模拟器在第 0 步提供仓库平面信息（不含任务信息与机器人信息，第 1 步开始提供相关信息）。
2. 每台机器人需向模拟器提交自身的运动路径申请。
3. 模拟器接收申请后，将依据各机器人基于运动路径生成机器人的预留队列。
4. 系统会确保不同机器人的预留队列无交集，该设计的核心目的是杜绝机器人间的冲突碰撞，有效避免因算法问题导致的机器人路径重叠情况。
5. **机器人的实际运动状态由其对应的预留队列决定**，具体原理可参考配套示意图（见图 2）。

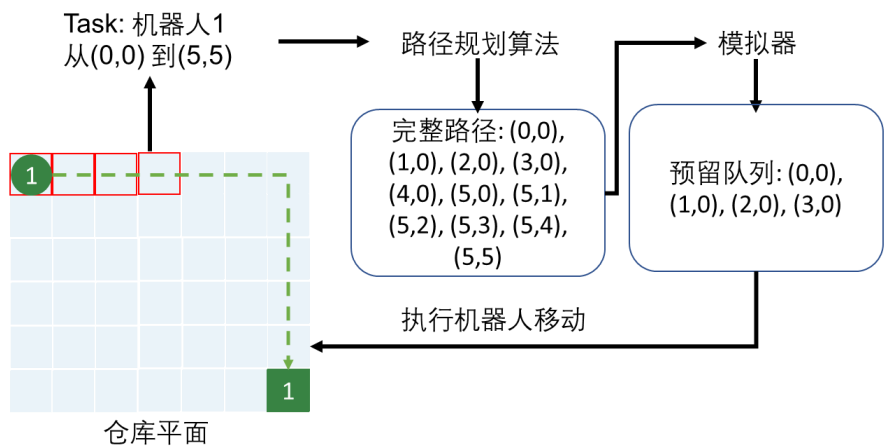


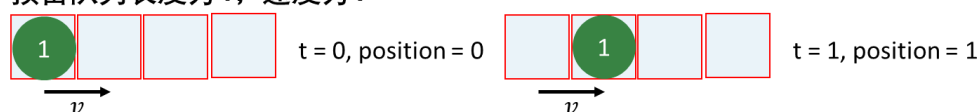
图 2、路径规划与预留队列的关系

## (二) 机器人行为规则

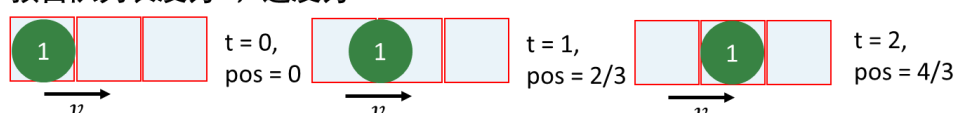
1. 机器人通过预设路径实现移动，路径由连续单元格构成。其移动速度受预留队列长度及转弯操作影响。
2. 向模拟器提交路径申请后，系统将为机器人生成预留队列（最大长度为 4）。**队列一旦确定即不可修改**，机器人需按序完成队列中的所有移动，直至抵达队列终点单元格。

3. 机器人的最大速度为 1，计算公式为：（预留队列中转弯点之前的列表长度 - 1）  
 $\div$ （预留队列最大长度 - 1），（见图 3）。
4. 机器人移动至下一个单元格的判定依据为 position 值：当 position 值大于等于 1 时，判定为已移动至下一个单元格，此时 position 值将重置为 0。该值通过当前机器人的速度累加计算得出。
5. 机器人移动过程中若需转弯，完成转弯动作需消耗 2 个时间片断，转弯完成后方可继续移动。
6. 机器人抵达目标点后，若当前方向与目标方向不一致，可执行原地转向调整。原地转向动作固定消耗 2 个时间片断，与转向角度无关。
7. 机器人方向值的计算规则：设当前单元格坐标为  $(x_1, y_1)$ ，下一步目标单元格坐标为  $(x_2, y_2)$ ：
  - 若  $x_1 = x_2$  且  $y_2 - y_1 = -1$ ，方向值为 1；
  - 若  $x_1 = x_2$  且  $y_2 - y_1 = 1$ ，方向值为 3；
  - 若  $y_1 = y_2$  且  $x_2 - x_1 = -1$ ，方向值为 2；
  - 若  $y_1 = y_2$  且  $x_2 - x_1 = 1$ ，方向值为 4。

**预留队列长度为4，速度为1**



**预留队列长度为3，速度为0.667**



**预留队列长度为2，速度为0.333**

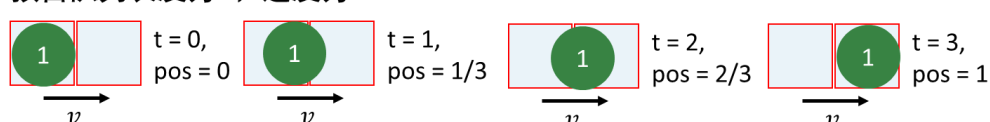


图 3、机器人速度与预留队列长度的关系

(三) 模拟器约束

- 1. 每台机器人的**预留队列存在最大长度限制，最大值为 4**。若提交的路径申请超出该长度，超出部分将被系统自动丢弃。
- 2. 机器人新申请路径的第一个单元格需满足特定条件：若已存在预留队列，需与队列最后一个单元格保持一致；若预留队列为空，则需与机器人当前所在单元格保持一致。
- 3. 机器人仅支持向相邻单元格移动，禁止斜向行驶；同时，不可向存在障碍的单元格行驶，其他机器人占据的单元格视为障碍。
- 4. 原地转向操作存在执行前提，**仅当机器人处于停止状态时可触发**。停止状态判定标准为预留队列长度为 0，或预留队列内仅包含机器人当前所在单元格；其他状态下提交的原地转向命令将被系统忽略。
- 5. **机器人抵达终点单元格后不会从系统中消失，其所在单元格将被视为障碍，其他机器人不可进入。**

(四) 接口说明

- 1. 算法接收输入的接口（hephaestus\_pb2.RobotEnvironmentContext）

字段名	字段说明
timestamp	当前模拟器运行的时间片断值
width	地图的宽度
height	地图的高度
robots	地图中所有机器人(<Robot>)的集合
mapCells	地图中所有单元格(<MapCell>)的集合

tasks	所有的任务(<Task>)信息
isInit	是否为初次执行，主要用于算法初始化

2. 算法返回输出的接口 (hephaestus\_pb2.RobotAlgorithmResult)

字段名	子字段	字段说明
robotAlgorithmUpdates		RobotAlgorithmUpdates 的集合
RobotAlgorithmUpdates		为单台机器人设置路径
	robotId	机器人 ID
	cells	机器人新的路径所包含单元格的集合
	direction	机器人在路径结束时最终的方向

3. 基础字段说明

字段名	子字段	字段说明
Robot	robotId	机器人 ID, 唯一
	locationIndex	机器人当前位置
	nextIndex	机器人前进的位置，当 locationIndex 和 nextIndex 相同时表示机器人在点上，不同时表示机器人在两个格子中间
	assignedPath	预留队列列表



	position	机器人在两个格子中间的状态取值在 0-1 之间
MapCell	cellCode	地图节点 code, 唯一
	index	地图节点在整个地图中的索引位置, 索引起始点(0,0)为地图的左下角
	location	地图节点在实际仓库中的位置信息
	directionCost	负载状态下该单元格可以行驶的方向
Task	robotId	机器人 ID
	Index	机器人起始点坐标
	dest	机器人目标点坐标
	direction	机器人初始方向
	endDirection	机器人最终方向

4. 单元格类型说明

类型编号	类型 code	说明
0	NULL	未分配功能的单元格
1	SHELF_CELL	放置货架的单元格, 可以向四个方向移动
22	BLOCKED_CELL	阻塞的单元格 (不可通行)

五、 运行与调试说明

(一) 环境依赖

软件

Java	jre8
Python	3.8

#### Python 包

grpcio	1.28.1
protobuf	3.11.3
websockets	8.1
PyInquirer	1.0.3
xmldict	0.12.0

## (二) 算法启动流程

- 1) 下载项目并解压;
- 2) 在模拟器目录 (.\hephaestus) 下启动命令

```
java -cp hephaestus-rpc.jar com.geekplus.hephaestus.rpc.SimulatorServer
```

- 3) GUI 程序启动命令

在 bundle 目录内执行

```
python multi_robot.py
```

- 4) 算法启动命令

在 .\algorithm\multi\_robot 目录下启动命令

```
python __main__.py
```

- 5) 在 GUI 界面点击 “开始” 键启动算法

\*GUI 界面可视化在大规模问题下运行更新非常慢，此时建议无 GUI 验证算法：

- 1) 在模拟器目录 (.\hephaestus) 下启动命令

```
java -cp hephaestus-rpc.jar com.geekplus.hephaestus.rpc.SimulatorServer
```

- 2) 算法启动命令

python 算法在 .\algorithm\multi\_robot\python\_demo 目录下启动命令

```
python __main__.py
```

- 3) 在项目根目录内执行

```
python -m bundle.resource.serve.multi_robot.cli
```

### (三) 保存算法中间过程为 json

- 1) 在模拟器目录 (.\hephaestus) 下启动命令

```
java -cp hephaestus-rpc.jar com.geekplus.hephaestus.rpc.SimulatorServer
```

- 2) 算法启动命令

python 算法在 .\algorithm\multi\_robot\python\_demo 目录下启动命令

```
python __main__.py
```

- 3) 在项目根目录内执行

```
python -m bundle.resource.serve.multi_robot.cli --record
```

\*此时会在根目录生成 result\_x.json 的文件，其中保存了算法在每一步调用时的结果，可以用记事本打开查看。

#### (四) 基于 json 文件评估算法

- 1) 在模拟器目录 (.\hephaestus) 下启动命令

```
java -cp hephaestus-rpc.jar com.geekplus.hephaestus.rpc.SimulatorServer
```

- 2) 在项目根目录内执行 (可以跳过算法启动命令)

```
python -m bundle.resource.serve.multi_robot.cli --replay
```

\*此时会读取根目录的 result\_x.json 的文件, 直接基于其中每一步的指令向模拟器发送指令, 跳过算法的实际执行环节。

##### 1. 初赛算法评分方式

针对每张地图 (记为地图  $i$ ) , 评分规则如下:

- 1) 若所有任务执行完成, 以实际执行步数  $T_i$  为基准, 该地图得分为  $(300 - T_i)$ 。
- 2) 若出现算法超时 (超过 300 步)、算法错误, 或模拟达到时间上限时仍有机器人未抵达终点的情况, 该地图得分为 0。
- 3) 参赛团队的最终成绩为所有地图得分的**平均值**。

##### 2. 算法结果提交说明

初赛阶段仅需提交算法每一步的返回结果, 具体要求如下:

- 1) **结果应为 json 格式文件**, 4 个文件需分别命名为 result\_1.json、result\_2.json、result\_3.json 和 result\_4.json, 对应 4 个不同场景地图。

- 2) 需将上述 4 个文件放入名为“results”的文件夹，压缩为一个 zip 格式压缩包，zip 文件命名为：参赛队伍名称，（如 “IE Award group 1.zip”），压缩包大小不得超过 2MB。
- 3) 完成压缩后，将压缩包按指定路径上传至赛事系统即可。

## 六、 联系方式

赛事官方邮箱：[ie\\_award@tsinghua.edu.cn](mailto:ie_award@tsinghua.edu.cn)，为赛事消息发布的唯一指定联系方式。

如参赛团队在参赛中遇到任何问题，请按照以下格式发送至官方邮箱：

标题：【所选赛道】+【队伍名称】+【简述问题】

正文：（描述您遇到的问题）

我们将通过官方邮箱对您的问题回复解答。