

多机器人路径规划软件

使用手册

目录

文 件 修 订 记 录	3
1. 总体功能描述	4
2. 运行环境	5
3. 软件编译环境	6
4. 软件安装说明	7
5. 软件启动	8
6. 软件用法	10
7. Dead-Cell 算法说明	12

文件修订记录

版本号	生成日期	作者	修订内容
V1.0	2022-3-10	郭帅	初始版本
V1.1	2022-5-6	郭帅	完善算法和初始界面
V1.5	2022-8-12	郭帅	加入全新 Dead-Cell 算法
V1.6	2022-8-31	郭帅	完成界面的全新布置
V1.7	2023-3-16	郭帅	引入迷宫和随机功能
V1.9	2023-4-15	郭帅	完善结果保存功能
V2.0	2023-4-25	郭帅	加入多机器人避碰算法

1. 总体功能描述

此软件是一个基于蚁群算法的 GUI 程序,可以帮助用户在自定义的环境中找到最短路径,支持多机器人在同一环境中的路径规划。并将结果保存为 .jpg 和 .gif 格式。

该软件具有清晰的用户界面,可以轻松重置起点和终点,可自定义蚁群算法相关参数。拥有全新原创算法“Dead-Cell”算法大幅提高搜索效率。

2. 运行环境

硬件要求

类 别	基本要求
客户端	4G 内存及以上；硬盘空间 40M 及以上

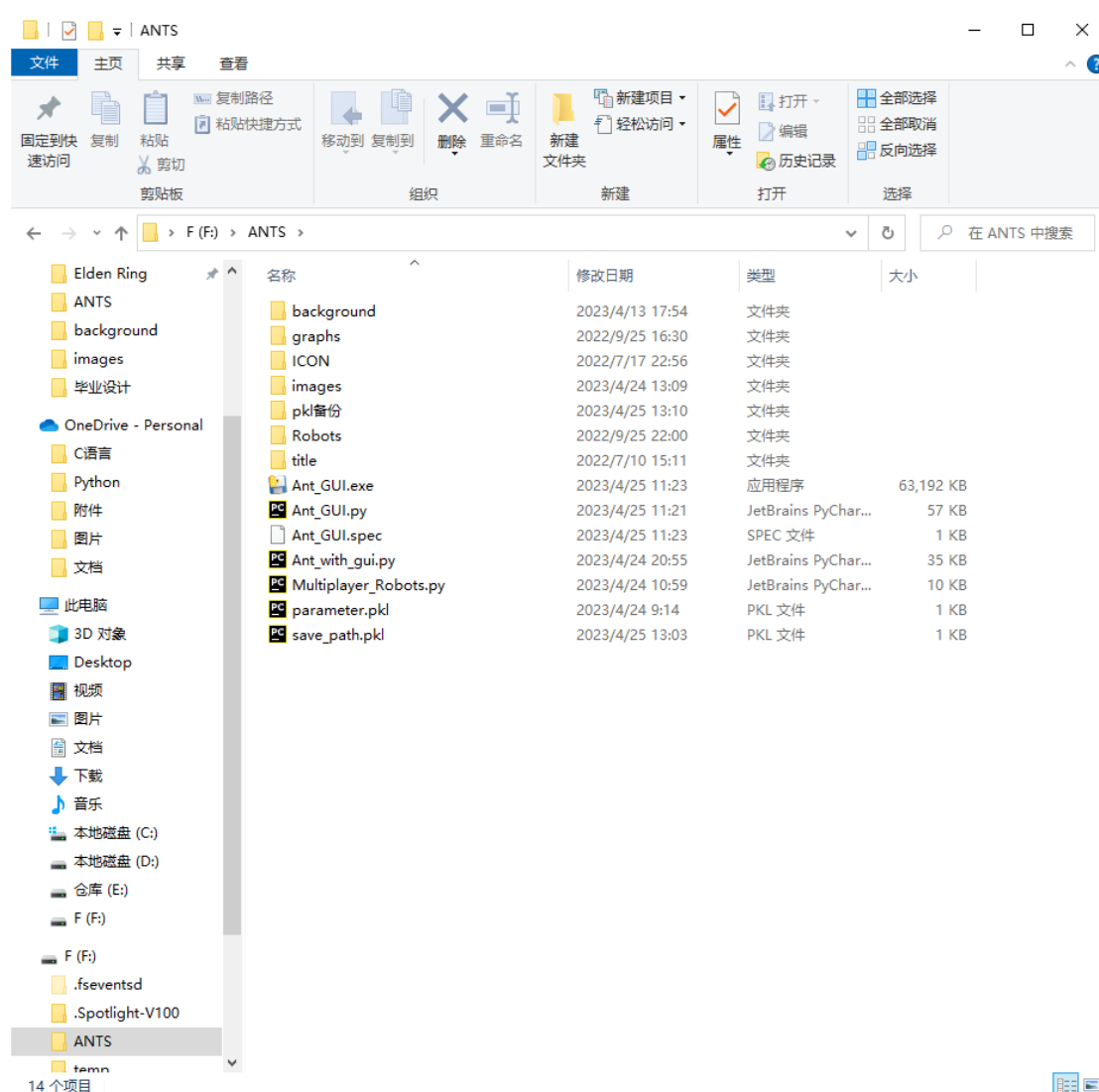
软件要求：

类 别	名 称	基本环境
客户端	操作系统	支持 Windows7 及更高 Windows 系统；

3. 软件编译环境

本软件使用 PyCharm Community Edition 2021.3.1 进行开发，需要使用相同软件进行开发编译。

需要按照源程序文档的三个模块分别建立三个.py 文件，最终保证以下文件在同一文件夹中：



4. 软件安装说明

在使用本软件之前，请确保您已经若你已配置好相关环境，并将所有文件下载到同一目录下。可以通过运行 `Ant GUI.py` 文件启动程序。

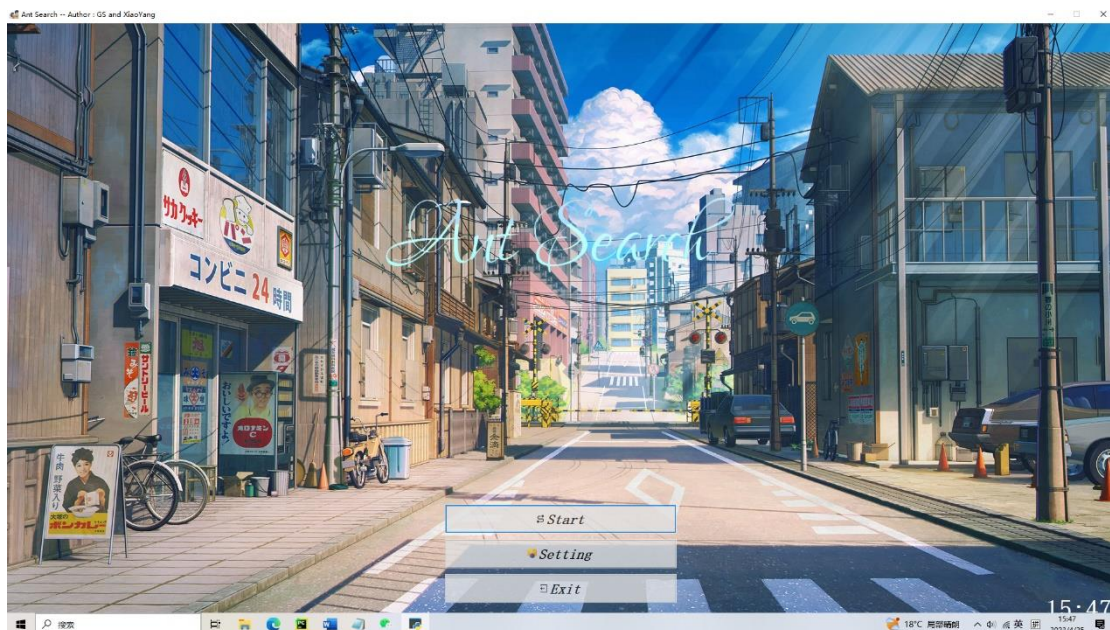
若没有配置相关环境，也可以直接运行 `Ant GUI.exe` 程序。

其中 `.pk1` 文件是运行必要的配置文件，整个项目可以在我的 Github 中下载。(<https://github.com/LiarGs/Ant-Colony-Algorithm-GUI>)

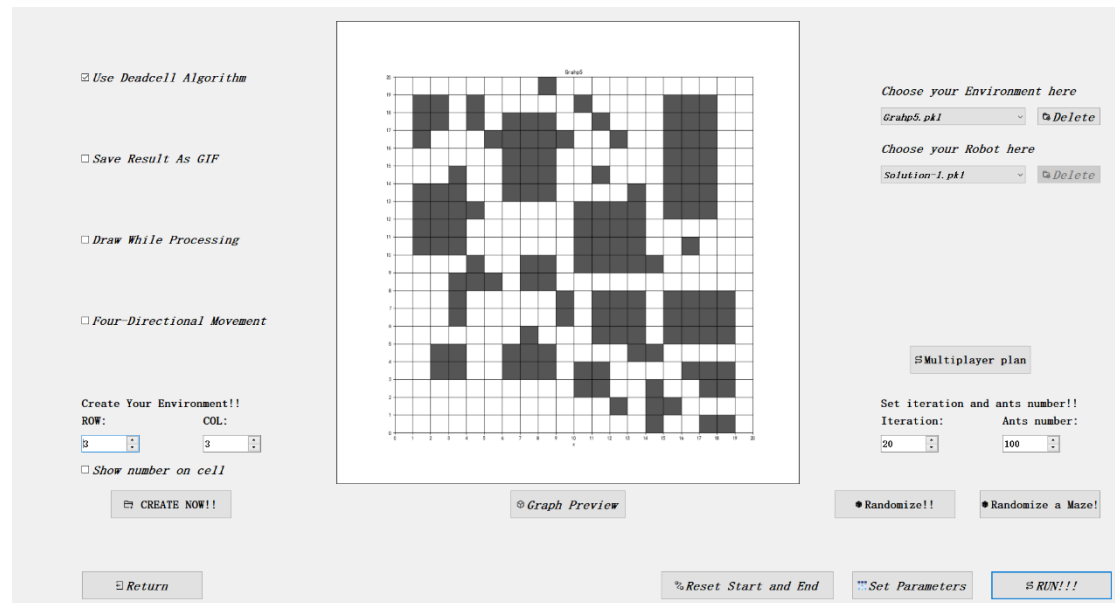
5. 软件启动

点击 Ant GUI.exe 后，等待软件加载完成，可以在主页选择进入，进行多机器人路径规划。

运行成功画面：



点击 Start 按钮正式进入多机器人路径规划界面：



6. 软件用法

创建环境

在程序界面中，您可以选择使用自己创建的环境，或者在界面右上方从预设的环境中选择一个进行测试。如果您选择创建自己的环境，可以通过在界面中点击“CREATE NOW!!”来直接绘制障碍物和起点终点等来创建环境。若您只是想创建一个临时环境，您可以在接下来的弹窗中选择“Don't Save”。您也可以单击右侧的“Randomize!!”和“Randomize a Maze!”来随机生成环境或迷宫。

寻找最短路径

一旦您选择好了环境，首先可以通过点击“Reset Start and End”按钮来部署机器人的起点和终点（左键改变障碍状态，右键设置起点或者终点），设置完起点和终点后，您可以通过单击“Run”按钮来启动蚁群算法，寻找最短路径。您可以通过单击“Set Parameters”按钮来调整蚁群算法的相关参数，如迭代次数、蚂蚁数量、信息素蒸发系数等，以便获得更好的结果。界面左边的四个选择分别代表是否使用“Dead-Cell”算法、是否将本次结果保存为动图、是否画出蚁群算法的中间过程、是否选择四向行动（适合求解迷宫）。

保存结果

当蚁群算法完成寻路任务后，会有一个弹窗，询问你是否要将本次寻路的环境应用到下次寻路中。此设定的存在是为了适应“Dead-Cell”算法。因为此算法的功能是在蚂蚁寻路过程中不断排除环境中的无效点。于是，若在弹窗中选择“yes”，则表明您不满意此次的寻路结果，您可以再次更改蚂蚁迭代次数和蚂蚁只数等相关参数，在已经被排除大部分环境中无效点的新环境中再次进行寻路，直到用户满意为止。因此，比较推荐的做法是先运行“20 次迭代，100 只蚂蚁”，排除掉环境中大部分无效点，然后再运行“80 次迭代，100 只蚂蚁”以获得更好的结果。若在弹窗中选择“No”，那么软件会认为用户接受此次的寻路结果，并将保存此

次的结果。您可以在 `images` 文件夹中找到你的结果。你也可以在主界面的 `Setting` 中更改你的保存路径。

其他功能

除了上述主要功能外，该程序还具有以下功能：

- “Dead-cell” 算法：一种新的算法，可以有效减少迭代次数，提高算法效率。
- 多机器人路径规划：支持在同一环境中同时规划多个机器人的最短路径。在得到数个机器人的最短路径后，单击 “Multiplayer plan” 按钮即可得到多个机器人之间实现完美的避碰处理后的路径规划，并得到保存在 `Robots` 下的 gif 结果动图

注意事项

- 请确保 `parameter.pkl` 和 `save_path.pkl` 文件与 `Ant GUI.exe` 或 `Ant GUI.py` 文件在同一目录下。
- 若出现更改保存路径后软件无法运行的情况，可将最初的两个 `pkl` 备份文件覆盖此时的两个 `pkl` 文件即可。
- 规划多机器人的路径时，其优先级是按照结果次序来的。例如：从 `Solutuion-1` 到 `Solutuion-9`，优先级依次递减。
- 在删除已有的规划好的机器人路径时，只能从最后一个机器人往前删除，因此在规划机器人寻路次序时务必小心。

7. Dead-Cell 算法说明

为了提高算法的运行效率，我创新性地提出了无效点理论。我通过改进蚁群的搜索规则，根据已有的环境信息，使用无效点理论的规则更新出新的环境信息，缩小蚁群的搜索空间，以加快算法的收敛速率。

被视为无效点的栅格在后续的循环过程中，将被视为障碍物存在的不可行栅格，既栅格地图中的黑色栅格。无效点理论提供了两个判断可行栅格是否为无效点的基本规则。这些规则将在以待判断栅格为中心的 3×3 大小的栅格环境中发挥作用。在下面表示判断环境的栅格地图中，黑色栅格为不可行点，白色栅格为可行点，红色（中间栅格）表示该点被判断为无效点。

（1）无效点理论判断规则一：

当在任意 3×3 栅格地图中出现了如图 3 所示情形，即待判断点三个方向皆为不可行栅格时，显然在路径规划中无法起到缩短路径长度的作用，故被判断为无效点。

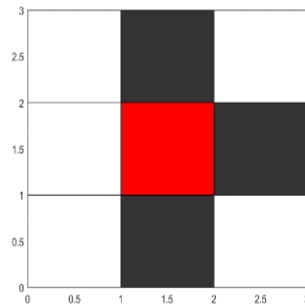
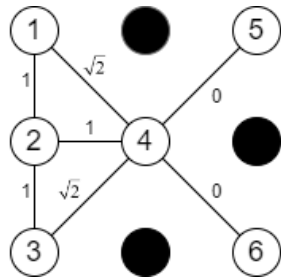


图 3 无效点理论规则一判断情形

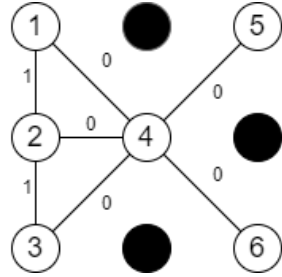
下面证明待判断栅格无法起到缩短路径长度的作用。将图中各个栅格点进行标号，并按照空间位置关系，画出其连通图，如图 4 所示。记录结点 1、2、3 各点之间最短路径长度的矩阵为 A_1 。矩阵中 a_{ij} 元素代表点 i 到点 j 的最短路径的长度。



$$A_1 = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{pmatrix}$$

图 4 无效点理论规则一连通图-判断前

而当待判断点 4，被判断为无效点时，其连通图如图 5 所示。此时，记录结点 1、2、3 各点之间的最短路径长度矩阵为 A_2



$$A_2 = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{pmatrix}$$

图 5 无效点理论规则一连通图-判断后

显然有 $A_1 = A_2$ 。由此可得，结点 4 可行与否并不影响其他结点之间的最短路径长度，即其未在缩短路径长度中起到作用，故可判断该栅格为无效点。

(2) 无效点理论判断规则二：

当任意 3×3 栅格地图中出现如图 6 所示情形，即待判断栅格附近存在三个不可行点，且其中两个不可行栅格斜向相邻，第三个不可行栅格只与这两栅格中的一个相邻。则该栅格为无效点，当且仅当图中的绿色栅格（左下角栅格）为可行栅格。

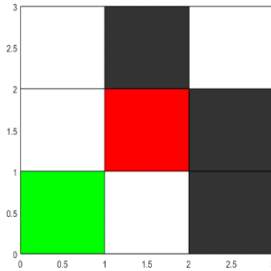


图 6 无效点理论规则二判断情形

下面证明待判断栅格无法起到缩短路径长度的作用。将途中各个栅格点进行标号，并按照空间位置关系，其连通图如图 7 所示。记录结点 1、2、3、4 各点之间最短路径长度的矩阵为 A_3 。

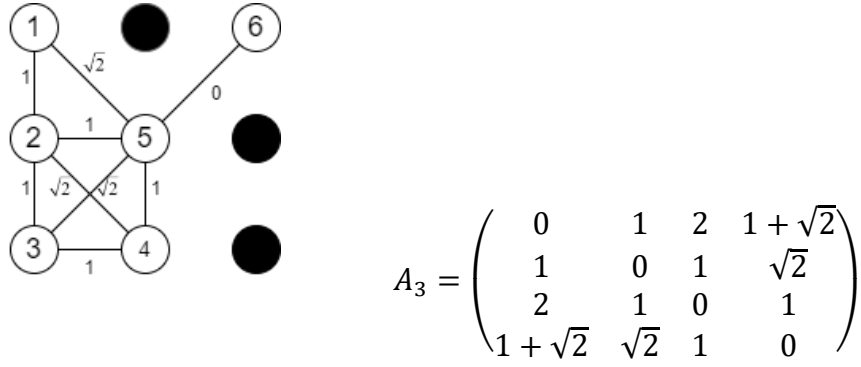


图 7 无效点理论规则二连通图-判断前

而当待判断点 5，被判断为无效点时，其连通图如图 8 所示。此时，记录结点 1、2、3、4 各点之间的最短路径长度矩阵为 A_4

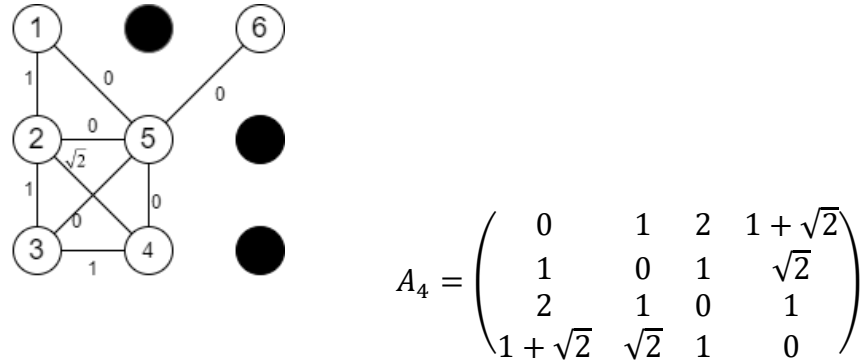


图 8 无效点理论规则二连通图-判断后

显然有 $A_3 = A_4$ 。由此可得，结点 5 可行与否并不影响其他结点之间的最短路径长度，即其未在缩短路径长度中起到作用，故可判断该栅格为无效点。

每当蚂蚁走到一个新栅格时，它都将读取以该栅格为中心的 3×3 大小的栅格环境，并通过上述的两种无效点判断规则，来判断当前栅格是否为无效点。若该栅格是无效点，则将该栅格进行记录，并继续下一步搜索。否则，蚂蚁正常进行下一步搜索。其中，被记录的栅格将在后续的迭代过程中被视为是不可行栅格。

通过一代又一代蚂蚁的不断搜索，栅格地图中的无效点将逐渐被看作是不可行栅格，使得栅格地图中的可行栅格数量逐渐减少。从而缩小蚂蚁的搜索空间和最短路径的解空间，使其能够更快地收敛到最短路径上。具体效果如图 9、图 10 所示。其中红色栅格表示被判断为无效点的可行栅格，显然在判断后的栅格环境中，可行点的数量减小了，并且将解空间限制在了合理范围内。除此之外，被判断为无效点的栅格点，将在以后的多机器人路径规划算法的退避策略中发挥作用。

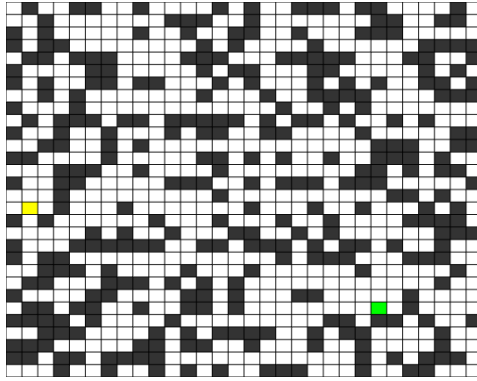


图 9 无效点判断前的栅格环境

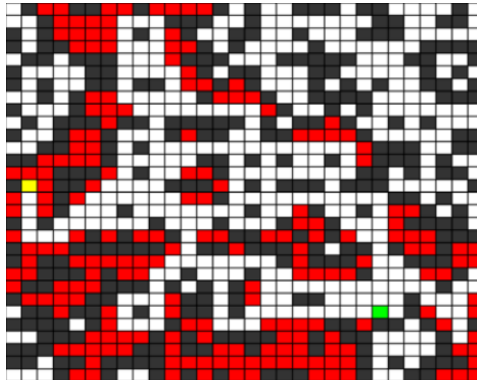


图 10 无效点判断后的栅格环境