

Tom And Jerry 解题报告

绍兴市第一中学 王文涛

试题来源

CodeChef MARCH12 TOMJERRY

试题大意

Tom和Jerry（猫和老鼠）在一个 $N \times M$ 方格地图上博弈。初始时，告诉你Jerry的位置，有一些格子是门，有一些格子是障碍。每一轮，Tom先操作：选择某个空位放上一个障碍；接着Jerry会这样移动：找到最短的一条通往一个门的路，沿路径的第一步移动（如果有多条这样的路那么按照某个上下左右的优先级）。现在Tom想围死Jerry，也就是使得Jerry无法到达任何一个门。你需要为Tom设计这样的方案。当然，放的障碍数越少越好。题目设计的评分方式会使得失败的方案（最终Jerry走到某个门）的得分远差于成功的方案的得分，同时成功的方案步数越少分数越好。

数据范围： $N, M \leq 50$

时间限制：2s

算法介绍

为了方便，定义预计路径为当前情况根据Jerry的策略预计要走的路径。

首先必须注意到最重要的是要围死Jerry，因为失败的话分数就会非常差。其次才是最小化放的障碍数。

想法1：搜索

最简单的爆搜就是枚举放的位置，然后让Jerry走一步，然后再枚举放的位置……

但这样显然是不能承受的。我们要想办法让Tom更聪明，不是盲目地放障碍。

想法2：跟着Jerry放障碍

我们可以在Jerry四周放障碍。但这样往往效果不好，因为Jerry会溜走。所以最好留有余地，比较好的是在距离不超过2、3的范围内放障碍。但实际在比较空旷的地图上效果并不好。

想法3：在Jerry预计路径上放障碍

这样的障碍会改变Jerry的预计路径，从而拖延时间，甚至堵死Jerry。我们可以枚举当前Jerry的预计路径上的格子放上障碍。

想法4：把门堵死

对于门比较少的情况，把门堵死是很好地策略。这样，一个门最多只要4个障碍，还可以避免和Jerry直接周旋。实际实现时，我们要优先堵住当前最有可能的方向。我们可以优先填上Jerry当前预计路径的最后一个空格。

但对于门较多的情况，这样做效果就不好了：把每个门堵住就需要耗费很多的障碍，需要消耗很多时间，Jerry就更有机会在很多门里选择。

想法5：给每个格子一个估价值

我们可以给每个格子一个估价值，估计堵住这个格子对Jerry未来可能的路径的影响大小，再找一个估价值最大的格子堵上。估价值的设计要考虑很多方面：距离不能太近，要优先堵上当前最紧迫的方向，要堵在要塞处，等等。估价值的设计比较困难，只能作为一个大概的判断，不能处理很精确的问题，难以应对各种各样的环境。

想法6: 设计陷阱伏击Jerry

我们仔细想就会发现, Jerry的策略并不是很高明: 他只知道往当前最近的路上走, 却没有考虑中途被堵死的危险性。与其想着如何把Jerry到门的路堵住, 不如想办法引诱Jerry, 直接把Jerry抓住。没错, 这就是本题一个很重要的想法: 利用地形设计一个陷阱, 埋伏在Jerry的路上, 等到Jerry进入陷阱后趁机围住Jerry。这种做法在地形空旷的情况下效果拔群, 因为Jerry往往没走几步就进了圈套。

对于陷阱的设计, 如果是空旷地带, 没有什么有利地形, 那么就可以设计这样的陷阱:

```
.###.
->..
.###.
```

其中→表示Jerry进入的方向。当Jerry从图中(2,2)到达图中(2,3)位置后, 就立即在(2,4)位置放上障碍。此时Jerry往往会往回走, 想逃出陷阱, 从而回到(2,2)。此时只要在(2,1)处放上障碍, 就可以把Jerry抓住了。

```
.###.
#*.#
.###.
```

构造一个陷阱需要在Jerry到来之前留有一定的时间放置障碍。为此, 我们可以引入空余障碍的概念: 如果Tom暂时不确定这一步的障碍放在哪里, 那么就把这一步的障碍称为空余障碍。在不影响Jerry的预计路径的前提下, 在后来的步骤中可以在有需要的地方放置空余障碍, 也就是决定之前的那一步障碍放在哪里。由于不能影响Jerry的预计路径, 所以当前这一步产生的空余障碍不能放在当前到这个空余障碍被决定放置之前所有Jerry的预计路径上。

利用空余障碍, 当Jerry走到某个位置时, 我们就可以在条件允许的情况下构造一个陷阱。

看起来要用代码实现这样的陷阱比较困难, 而且这样的模式不依赖原有的地形。

为了让我们的陷阱聪明地利用原有地形, 在要塞处放置障碍, 从而减少所需的障碍数, 我们可以用最小割。注意到一个陷阱必然是在Jerry的路径上找2个

连续的空格使得当Jerry试图退出陷阱时留有余地。当我们找到这样合适的2格后，就找一个最小割分隔开这2格和其他所有的门。可以这样构图：

对于每个格子建2个点 $in_{x,y}, out_{x,y}$ 。 S 表示源， T 表示汇。连边： (u, v, w) 表示点 u 到 v 有一条流量限制为 w 的边。

对于每个空格： $(in_{x,y}, out_{x,y}, 1)$

对于两个相邻的非障碍格子： $(out_{x1,y1}, in_{x2,y2}, \infty)$

对于2个特殊格子： $(S, out_{x,y}, \infty)$

对于每个门： $(in_{x,y}, T, \infty)$

在这张图上跑最大流。最后得到一组最小割只需从源在参量网络上DFS一下，所有跨越被DFS到的点和没被DFS到的点的边构成最小割。最小割必然在流量为1的边上，相应的格子就是我们的陷阱的组成部分。

放障碍的时候要考虑某些空余障碍不能放在某个位置上。对于一前一后2个对Jerry的预计路径有影响的障碍留到最后依次放置。

最终算法

首先，用搜索让Jerry先走几步，走的方向需要找出预计路径（可以用BFS求出）。然后让Tom利用想法2、3的方式枚举格子放障碍或者暂时不定这个障碍的位置，把它作为空余障碍留到以后需要时再确定放哪里。同时也可以尝试利用最近的2步设置一个陷阱把Jerry直接抓住。（想法4其实已经包含在这个算法里：先迫使Jerry改变路径拖延时间，再把门旁边剩下的那些缺口看成一个巨大的陷阱）由于搜索的状态太多，我们可以使用卡时、迭代加深等方法在有限的时间内得到尽可能优的解。

当然其他的策略还有很多，解法不唯一。

总结

本题作为一类challenge问题，考察的是选手分析问题，考虑多种不同的情况，找到问题的关键，并利用合适的算法或数据结构提出相应的解决方法，把模糊的思路逐渐变得清晰，最后用代码实现的能力，具有很强的综合性。

对于AI等博弈问题，要善于寻找对手策略的薄弱点，并用相应的策略攻击。

对于网格图的围堵问题，很多情况下可以考虑网络流/最小割的模型。