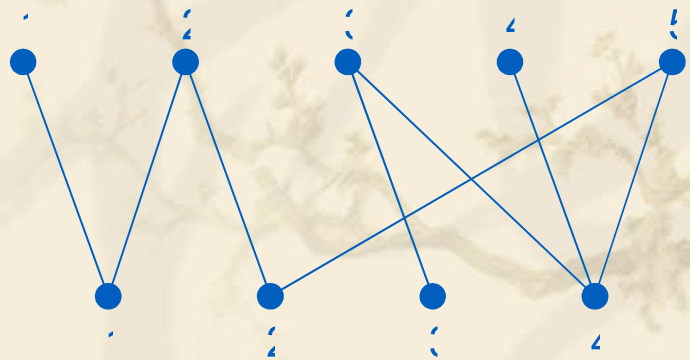
二分图匹配

匈牙利算法和 **KM** 算法简介

二分图的概念

- 二分图又称作二部图，是图论中的一种特殊模型。
- 设 G=(V,{R}) 是一个无向图。如顶点集 V 可分割为两个互不相交的子集，并且图中每条边依附的两个顶点都分属两个不同的子集。则称图 G 为二分图。



最大匹配

- 给定一个二分图 G，在 G 的一个子图 M 中，M 的边集 {E} 中的任意两条边都不依附于同一个顶点，则称 M 是一个匹配。
- 选择这样的边数最大的子集称为图的最大匹配问题 (maximal matching problem)
- 如果一个匹配中，图中的每个顶点都和图中某条边相关联，则称此匹配为完全匹配，也称作完备匹配。

匈牙利算法

- 求最大匹配的一种显而易见的算法是：先找出全部匹配，然后保留匹配数最多的。但是这个算法的复杂度为边数的指数级函数。因此，需要寻求一种更加高效的算法。
- 增广路的定义 (也称增广轨或交错轨) :
- 若 P 是图 G 中一条连通两个未匹配顶点的路径，并且属 M 的边和不属 M 的边 (即已匹配和待匹配的边) 在 P 上交替出现，则称 P 为相对于 M 的一条增广路径。

匈牙利算法

- 由增广路的定义可以推出下述三个结论：
- 1 － P 的路径长度必定为奇数，第一条边和最后一条边都不属于 M 。
- 2 － P 经过取反操作可以得到一个更大的匹配 M’ 。
- 3 － M 为 G 的最大匹配当且仅当不存在相对于 M 的增广路径。

匈牙利算法

- 用增广路求最大匹配 (称作匈牙利算法，匈牙利数学家 Edmonds 于 1965 年提出)
- 算法轮廓：
- (1) 置 M 为空
- (2) 找出一条增广路径 P ，通过取反操作获得更大的匹配 M’ 代替 M
- (3) 重复 (2) 操作直到找不出增广路径为止

匈牙利算法

- 程序清单：
- Function find(k:integer):integer;
- var st,sf,i,j,t:integer;
-     queue,father:array[1..100] of integer;
- begin
- queue[1] := k; st := 1; sf := 1;
- fillchar(father,sizeof(father),0);
- repeat

匈牙利算法

```
for i:=1 to n do
    if (father[i]=0)and(a[queue[st],i]=1) then
     begin
       if match2[i]<>0 then
        begin
         inc(sf);
         queue[sf] := match2[i];
         father[i] := queue[st];
        end else
```

匈牙利算法

```
begin
        j := queue[st];
        while true do
         begin
          t := match1[j];
          match1[j] := i;
          match2[i] := j;
          if t = 0 then break;
          i := t; j := father[t];
```

匈牙利算法

```
end;
        find := 1;
        exit;
       end;
     end;
  inc(st);
 until st>sf;
 find := 0;
end;
```

匈牙利算法

- 在主程序中调用下面的程序即可得出最大匹配数。
- Bmatch := 0;
- For I:=1 to n do Bmatch := Bmatch + find(i);
- Writeln(Bmatch);
- 一个关于二分图的性质：
- 最大匹配数＋最大独立集＝ X ＋ Y

最佳匹配

- 如果边上带权的话，找出权和最大的匹配叫做求最佳匹配。
- 实际模型：某公司有职员 $x_1,x_2,\ldots,x_n$, 他们去做工作 $y_1,y_2,\ldots,y_n$, 每个职员做各项工作的效益未必一致，需要制定一个分工方案，使得人尽其才，让公司获得的总效益最大。
- 数学模型： G 是加权完全二分图，求总权值最大的完备匹配。

KM 算法

- 穷举的效率－ n ！，我们需要更加优秀的算法。
- 定理:
- 设 M 是一个带权完全二分图 G 的一个完备匹配，给每个顶点一个可行顶标 (第 i 个 x 顶点的可行标用 lx[i] 表示，第 j 个 y 顶点的可行标用 ly[j] 表示)，如果对所有的边 (i,j) in G, 都有 lx[i]+ly[j]>=w[i,j] 成立 (w[i,j] 表示边的权)，且对所有的边 (i,j) in M, 都有 lx[i]+ly[j]=w[i,j] 成立，则 M 是图 G 的一个最佳匹配。证明很容易。

KM 算法

- 对于任意的 G 和 M，可行顶标都是存在的：
- $l(x) = \max w(x,y)$
- $l(y) = 0$
- 欲求完全二分图的最佳匹配，只要用匈牙利算法求其相等子图的完备匹配；问题是当标号之后的 G_l 无完备匹配时怎么办？ 1957 年（居然比匈牙利算法早？？？），Kuhn 和 Munkras 给出了一个解决该问题的有效算法，用逐次修改可行顶标 $l(v)$ 的办法使对应的相等子图之最大匹配逐次增广，最后出现完备匹配。

KM 算法

- 修改方法如下：
- 先将一个未被匹配的顶点 u(u in {x}) 做一次增广路，记下哪些结点被访问那些结点没有被访问。求出 d=min{lx[i]+ly[j]-w[i,j]} 其中 i 结点被访问， j 结点没有被访问。然后调整 lx 和 ly ：对于访问过的 x 顶点，将它的可行标减去 d ，对于所有访问过的 y 顶点，将它的可行标增加 d 。修改后的顶标仍是可行顶标，原来的匹配 M 仍然存在，相等子图中至少出现了一条不属于 M 的边，所以造成 M 的逐渐增广。

KM 算法

- 上述算法的证明也很容易
- Kuhn － Munkras 算法流程：
- (1) 初始化可行顶标的值
- (2) 用匈牙利算法寻找完备匹配
- (3) 若未找到完备匹配则修改可行顶标的值
- (4) 重复 (2)(3) 直到找到相等子图的完备匹配为止

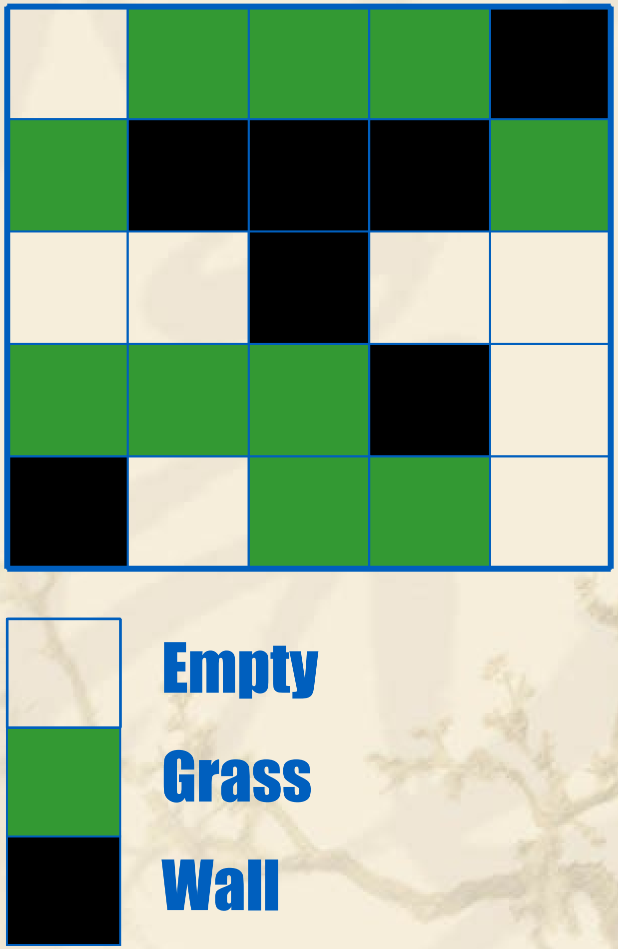
参考文献

- 王树禾《离散数学引论》
- 吴文虎 王建德《图论算法与程序设计》
- 刘汝佳 黄亮《算法艺术与信息学竞赛》
- 2002 年冬令营论文—孙方成《偶图的算法及应用》
- 2004 年冬令营论文—黄源河《浅谈图论模型的建立与应用》

例题 1 Place the Robots （Z0J1654）

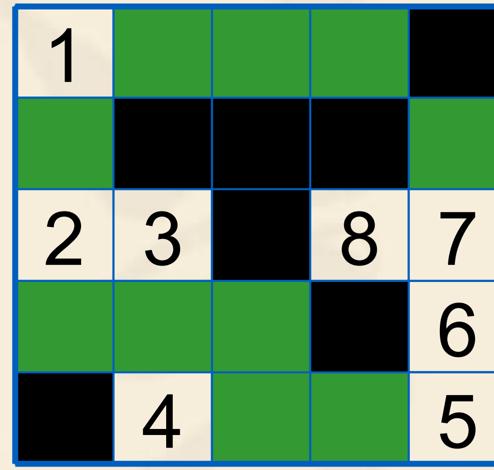
问题描述

有一个 N*M(N,M<=50) 的棋盘，棋盘的每一格是三种类型之一：空地、草地、墙。机器人只能放在空地上。在同一行或同一列的两个机器人，若它们之间没有墙，则它们可以互相攻击。问给定的棋盘，最多可以放置多少个机器人，使它们不能互相攻击。



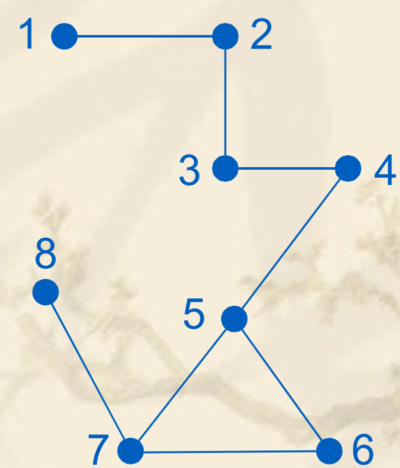
例题 1 Place the Robots （ZOJ）

模型一



在问题的原型中，草地，墙这些信息不是我们所关心的，我们关心的只是空地和空地之间的联系。因此，我们很自然想到了下面这种简单的模型：

以空地为顶点，有冲突的空地间连边，我们可以得到右边的这个图：

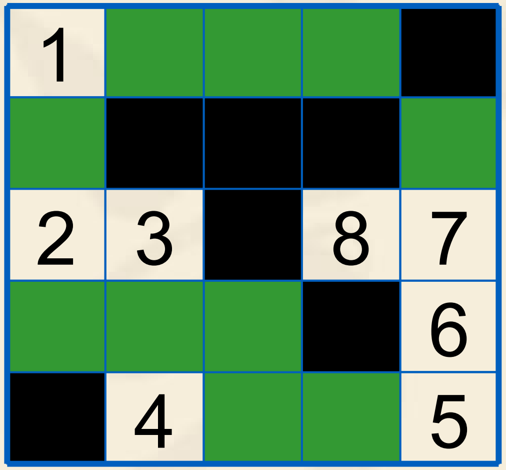


于是，问题转化为求图的最大独立集问题。

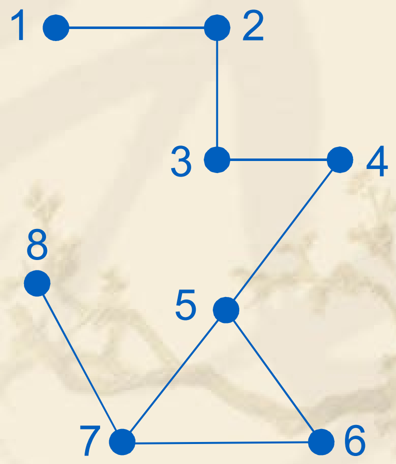
例题 1 Place the Robots （ZOJ）

模型一

在问题的原型中，草地，墙这些信息不是我们所关心的，我们关心的只是空地和空地之间的联系。因此，我们很自然想到了下面这种简单的模型：



以空地为顶点，有冲突的空地间连边，我们可以得到右边的这个图：

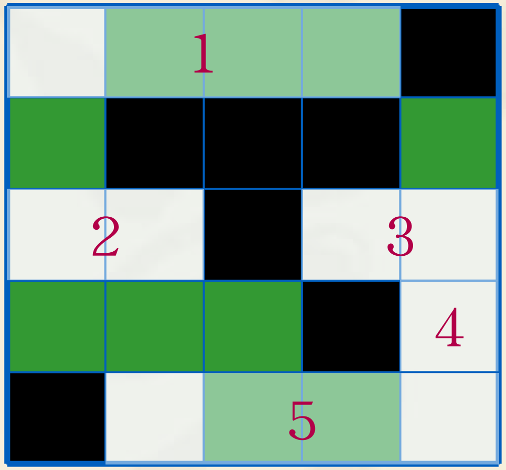


这是 NP 问题！

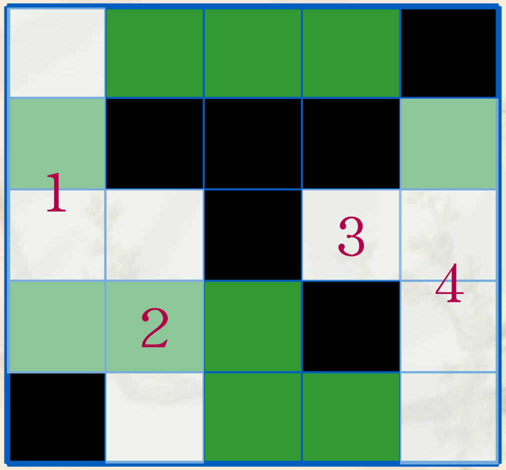
例题 1 Place the Robots（ZOJ）

模型二

我们将每一行，每一列被墙隔开，且包含空地的连续区域称作“块”。显然，在一个块之中，最多只能放一个机器人。我们把这些块编上号。



同样，把竖直方向的块也编上号。

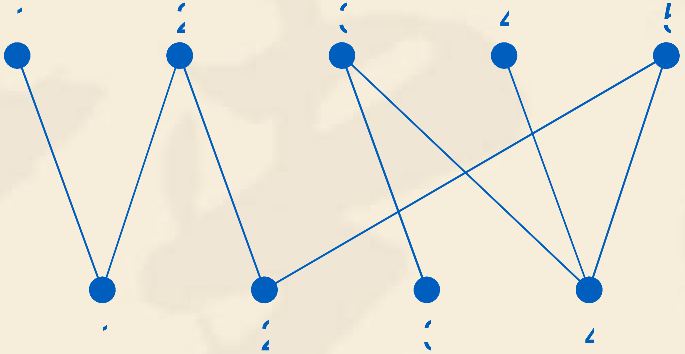


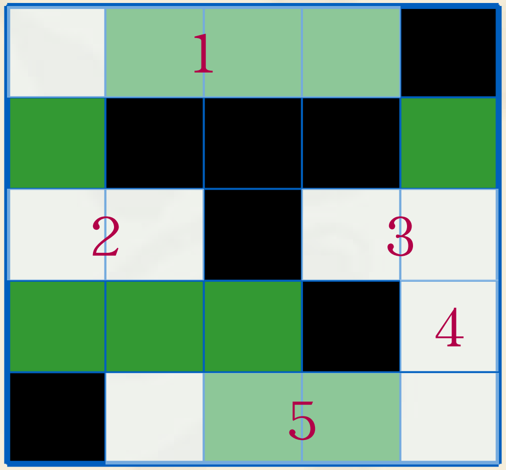
例题 1 Place the Robots （ZOJ）

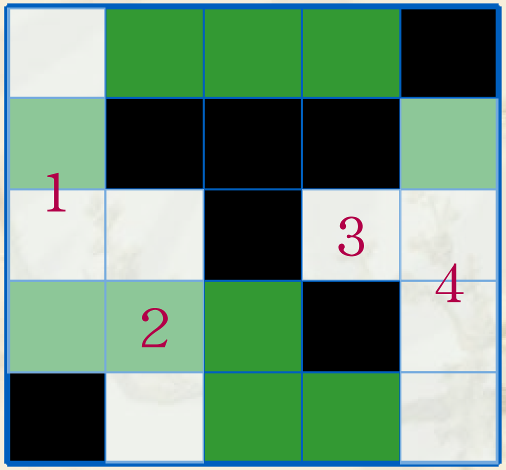
模型二

把每个横向块看作 X 部的点，竖向块看作 Y 部的点，若两个块有公共的空地，则在它们之间连边

于是，问题转化成这样的一个二部图：

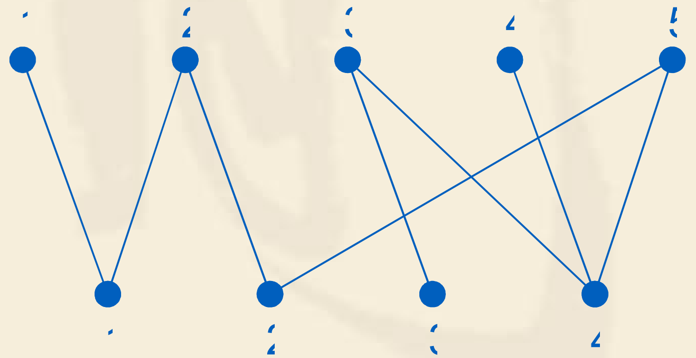




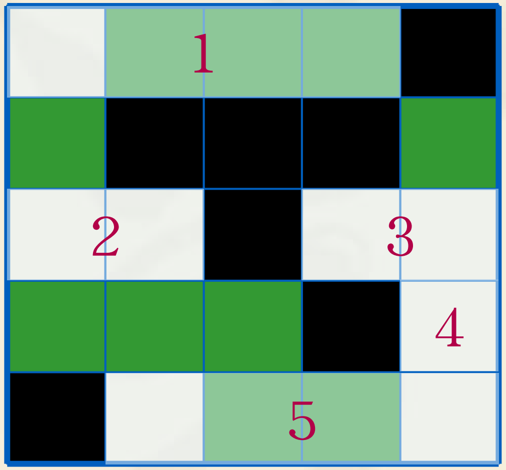


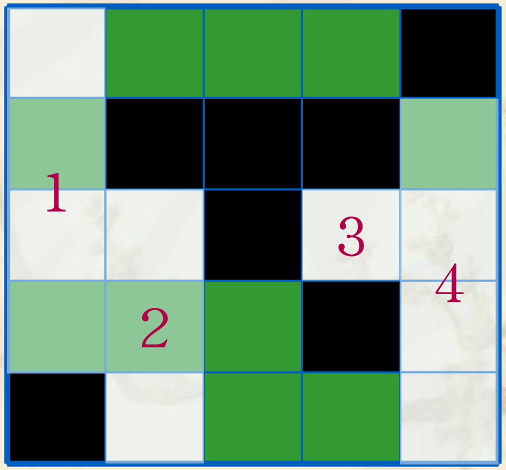
例题 1 Place the Robots （ZOJ）

模型二



由于每条边表示一个空地，有冲突的空地之间必有公共顶点，所以问题转化为二部图的最大匹配问题。





例题 1 Place the Robots（ZOJ）

小结

比较前面的两个模型：模型一过于简单，没有给问题的求解带来任何便利；模型二则充分抓住了问题的内在联系，巧妙地建立了二部图模型。为什么会产生这种截然不同的结果呢?其一是由于对问题分析的角度不同：模型一以空地为点，模型二以空地为边；其二是由于对原型中要素的选取有差异：模型一对要素的选取不充分，模型二则保留了原型中“棋盘”这个重要的性质。由此可见，对要素的选取，是图论建模中至关重要的一步。

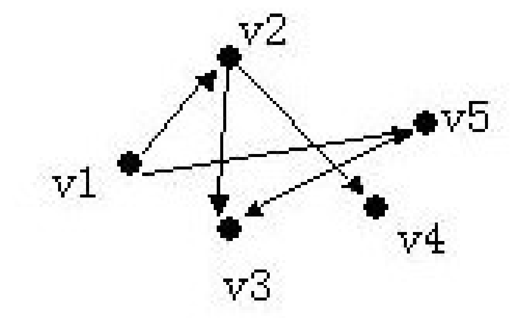
例题 2 救护伤员（T0J1148）

- 无情的海啸夺取了无数人的生命．很多的医疗队被派往灾区拯救伤员．就在此时，医疗队突然发现自己带的药品已经不够用了，只剩下了 N 种。(1 < n <= 20)，随着病人病情的发展，每种药在每天能获得的效果是不一样的。同时，每天病人只能服用一种药。也就是说，这些药还够支持 N 天。现在，给出你每种药在每天使用的效果，请你判断当每种药都用完后所有药达到的效果之和最大可以是多少。

例题 3　打猎

- 猎人要在 n*n 的格子里打鸟，他可以在某一行中打一枪，这样此行中的所有鸟都被打掉，也可以在某一列中打，这样此列中的所有鸟都打掉。问至少打几枪，才能打光所有的鸟？
- 建图：二分图的 X 部为每一行， Y 部为每一列，如果 (i,j) 有一只鸟，那么连接 X 部的 i 与 Y 部的 j 。
- 该二分图的最大匹配数则是最少要打的枪数。

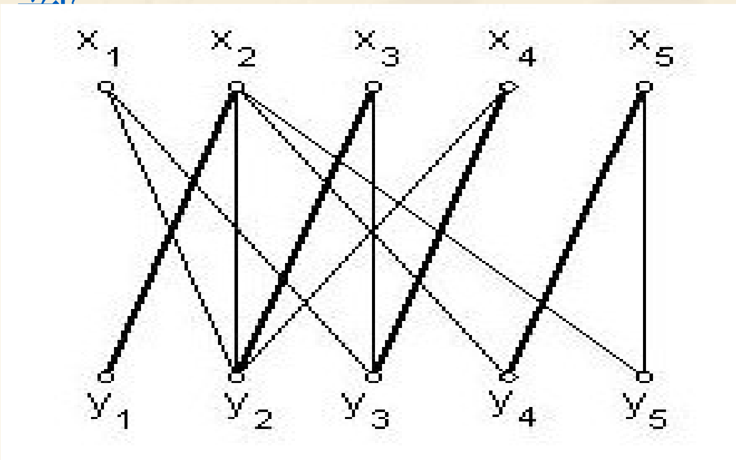
例题 4 最小路径覆盖



- 一个不含圈的有向图 G 中， G 的一个路径覆盖是一个其结点不相交的路径集合 P ，图中的每一个结点仅包含于 P 中的某一条路径。路径可以从任意结点开始和结束，且长度也为任意值，包括 0 。请你求任意一个不含圈的有向图 G 的最小路径覆盖数。
- 理清一个关系：最小路径覆盖数＝ G 的定点数－最小路径覆盖中的边数

例题 4 最小路径覆盖

- 试想我们应该使得最小路径覆盖中的边数尽量多，但是又不能让两条边在同一个顶点相交。
- 拆点：将每一个顶点 i 拆成两个顶点 Xi 和 Yi 。然后根据原图中边的信息，从 X 部往 Y 部引边。所有边的方向都是由 X 部到 Y 部



例题 4 最小路径覆盖

- 因此，所转化出的二分图的最大匹配数则是原图 G 中最小路径覆盖上的边数。因此由最小路径覆盖数＝原图 G 的顶点数－二分图的最大匹配数便可以得解。