

匹配算法 在搜索问题中的应用

浙江省杭州第十四中学 楼天城

`loutiancheng@sina.com`

前言

很多题目，如果我们可以建立数学模型，应该尽量用解析法来处理，因为简单的模型更清晰地反映了事物之间的关系。

但是，并不是所有的题目都可以建立简单的数学模型。我们这时必须使用搜索的方法，也就是枚举所有可能情况来寻找可行解或最优解。

由于搜索一般建立在枚举之上，所以搜索常常和低效是分不开的。

有时搜索的运算量非常大，实在是一件痛苦的事情。

于是我们需要利用很多技巧来提高效率：

可行性剪枝，

最优性剪枝，

调整搜索顺序，

等方法都很有用，在它们的帮助下，
我们可以大大提高搜索的效率。

而有些题目，这些常规的优化方法很难有用武之地。这时我们必须使用一些非常规的搜索方法。

本文中我们将讨论非常规搜索中的一种
部分搜索 + 匹配算法

引题：

N 个物品与 N 个位置，给定每个物品可能放的位置集合，要求寻找一一对应的关系。

但还给出物品位置之间的限制（例如：如果 1 放在 3 则 2 不能放在 1）。

求一组可行解，或给每一种对应关系一个权，求满足条件的最优解。

由于事物之间的限制关系非常复杂，很难建立简单的二分图关系，或者用网络流来解决。

面对这一系列类似的问题，我们一般只有搜索，如何搜索又如何优化呢？

简单分析：

如果我们枚举每一个物品的位置，然后判断。这样的时间复杂度为 $O(n!)$ 。好像似乎也只能这样。

进一步分析：

我们看一个例子， $n=6$ ：

其它限制有 4 条 (a, b, c, d) 表示如果 a 放在 b 则 c 不能放在 d

11 33 55 66

22 22 55 33

33 11 44 11

33 22 66 22

我们发现，如果我们一旦确定了 3 和 5 的位置，其它 4 个物品的位置之间已经没有限制关系了，这样其它 4 个物品的位置可以通过匹

配来解决。

时我们发现一个新的搜索方法：部分搜索 + 匹配

部分搜索 + 匹配：

搜索一部分变量，使得余下变量之间的关系简化，然后通过一些高效算法（匹配）完成余下问题。

就本题而言就是：先搜索一定数量（而不是全部）物品的位置，使问题内其它物品的关系简化为二分图关系，用二分图匹配来解决余下的物品。

例如上面的例子，如果我们先知道了 3 和 5 的位置后，不用匹配，其实我们是在用搜索来求匹配，效率当然不会高。

通过部分搜索为匹配算法提供条件（例如上面的例子创造二分图关系），而匹配算法代替搜索，高效地完成余下的任务。

部分搜索 + 匹配的方法充分发挥了搜索和匹配算法的双重优势。搜索的优势在于应用性广，可以克服复杂的情况，匹配算法的优势在于效率高。两者相互促进，同时也弥补对方的不足。这也是这个方法成功的关键。

部分搜索 + 匹配的方法已经在很多题目中得到了应用。

一个部分搜索 + 匹配算法的经典例子。

知破连环阵

题目简述 (NOI2003 二试第三题)

B 国的连环阵由 M 个武器组成。最初，1 号武器处于攻击状态，其他武器都处在无敌自卫状态。以后，一旦第 i ($1 \leq i < M$) 号武器被消灭，1 秒钟以后第 $i+1$ 号武器就自动从无敌自卫状态变成攻击状态。

A 国有 N 个炸弹，每个炸弹的作用半径均为 k ，且会持续爆炸 5 分钟。在这 5 分钟内，瞬间消灭离它直线距离不超过 k 的、处在攻击状态的 B 国武器，不会炸毁本国炸弹。

任务：

决定一个序列 a_1 、 a_2 、 $a_3 \cdots$ 使得在第 a_x 号炸弹引爆的时间内连环阵被摧毁。这里的 x 应当尽量小。

输入：

N, M 及武器和炸弹的坐标。

测试数据中的坐标是随机生成的。

初步分析：

A 国炸弹 i 可以炸到 B 国武器 j 的条件：

$$(u[i]-x[j])^2+(v[i]-y[j])^2 \leq R^2$$

结论：很难找到求最优解的多项式算法。
面对此类问题，一般只有搜索策略。

进一步分析：

每一颗炸弹必定炸掉 B 国武器中编号连续的一段。

5 分钟只是表明每一颗炸弹可以炸掉任意多个编号连续的 B 国武器。

普通的搜索方法：

每次寻找一个编号最小的没有被炸掉的 B 国武器，选择一颗没有使用过并能炸到此武器的 A 国炸弹，然后使用这颗炸弹炸掉 B 国武器连续的一段，继续深度优先搜索下一颗炸弹的编号，如果发现 B 国武器已经全部炸毁就可以回溯。

搜索的时间复杂度为 $O(n!)$ 。即使加上优化，程序效率也不是很高。

部分搜索：

此题使用部分搜索的算法需要一些转化：
如果已经将 B 国武器根据编号分为 x 段，其中第 i 段为 $[S_i, T_i]$ ($S_1=1, T_i \geq S_i, T_i+1=S_{i+1}$)。

然后的任务就是判断是否可以从 A 国的 N 颗炸弹中选出 x 颗，分别可以炸掉其中的一段

。

其实我们把搜索分为了两部分，

(1) 将 B 国武器根据编号分为 x 段。

(2) 判断是否可以从 A 国的 N 颗炸弹中选出 x 颗，分别可以炸掉其中的一段。

其实第二部分可以用匹配来解决。

建图：

$C[S][T][I]$ 表示 A 国炸弹 I 是否可以炸到 B 国武器 $S, S+1 \dots T-1, T$ 。

$$C[S][S][I] = ((u[I] - x[S])^2 + (v[I] - y[S])^2 \leq R^2)$$

$$C[S][T][I] = C[S][T-1][I] \ \&\& \ C[T][T][I] \quad (S < T)$$

求 C 的时间复杂度为 $O(n^3)$ 。

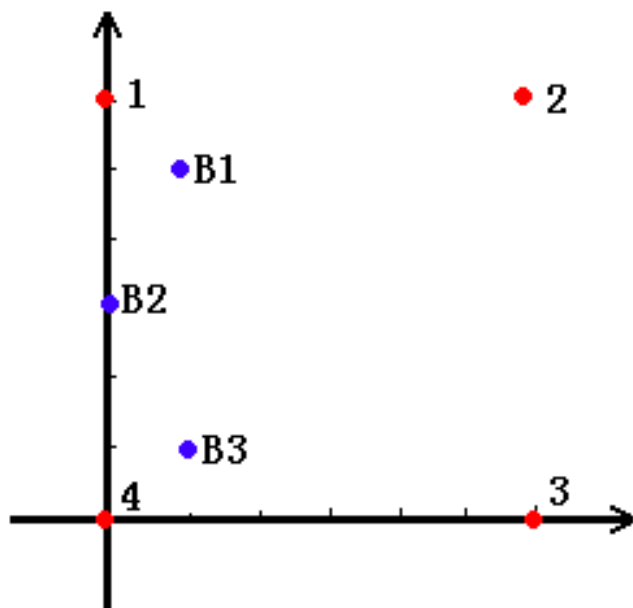
建图：左边 x 个点，表示 B 国武器根据编号分为的 x 段，右边 N 个点，表示 A 国的 N 颗炸弹。
左边第 i 个点到右边第 j 个点有边的条件即：

$$C[S_i][T_i][j]。$$

下面任务就是将 B 国武器根据编号划分为若干段 + 二分图匹配判断。

样例 1:

4	3	6				
0	6	6	6	6	0	0
1	5	0	3	1	1	

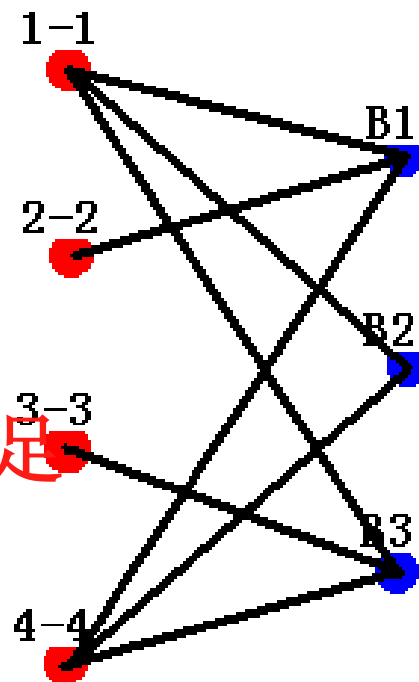


如果的划分方法是
 $x=4$,

4 段分别为

1-1, 2-2, 3-3, 4-4

最大匹配 $< x$, 不满足



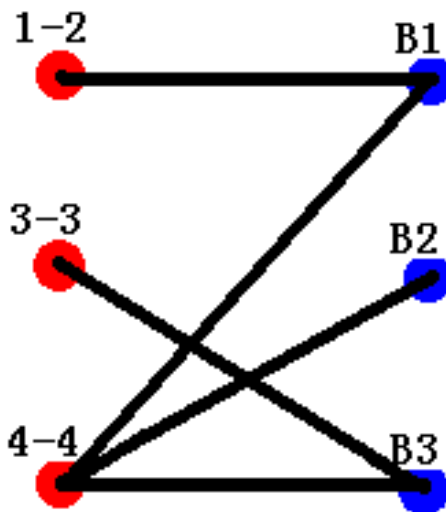
如果的划分方法是:

$x=3$,

3 段分别为

1-2, 3-3, 4-4

最大匹配 $= x$, 满足



性能分析（1）：

搜索的基本框架已经建立，虽然数据是随机生成的，但是 m 个 B 国武器的划分方案还是非常多的，有时可能高达 2^m 。时间上很难承受，如果使用卡时，正确性受到影响，效果不会很好。

只有 4 个数据可以在时限内出解，另外 6 个如果卡时，有 2 个也可以得到最优解。

优化：

优化可以通过可行性和最优性两方面分析。

优化一（最优性）：

如果 A 国炸弹可以重复使用，设：

$\text{Dist}[i]$ = 炸掉 B 国武器 $i - m$ 的最少使用炸弹数。

可以用动态规划计算 Dist 值，状态转移方程如下：

$$\text{Dist}[m+1] = 0,$$

$$\text{Dist}[i] = \min\{\text{Dist}[j] + 1 \mid C[i][j-1][k]\}$$

$(0 \leq k \leq n)$ 从而产生了一个最优性剪枝条件：

$\text{if } (1 \leq i \leq N) \quad (1 \leq j \leq N+1)$
 $\text{if } (\text{当前已经使用的炸弹数} + \text{Dist}[\text{当前已经炸掉的 B 国武器数} + 1] \geq \text{当前找到的最优解}) \text{ then 剪枝；}$
求 Dist 的时间复杂度为 $O(n^3)$ 。

优化二（可行性）：

部分搜索 + 匹配的方法一般都可以用两个效果很好的可行性优化：

(1) 提前判断是否可以匹配成功，避免多余的搜索。

(2) 每次匹配可以从以前的匹配开始扩展，不需要重新开始。如果当前的划分方法已经无法匹配成功，就没有搜索下去的必要了，只要每搜索新的一段时立即通过匹配判断即可。

每次求匹配只要从原来的基础上扩展就可以了。没有必要从头开始。

性能分析（2）：

通过上述两个优化，程序效率有了很大提高。

10 个测试数据中有 8 个可以在时限内出解，另外 2 个如果卡时，也可以得到最优解。

进一步优化：

优化二虽然排除了许多不必要的划分，但是在判断时浪费了不少时间。

因此，在枚举划分长度时，可以通过以前的划分和匹配情况（被匹配的边），用 $O(n^2)$ 的时间复杂度的宽度优先搜索计算出下一个划分的最大长度 maxL ，显然下一个划分的长度在 $[1, \text{maxL}]$ 都一定可以找到可行的匹配。

这样既节省了判断的时间，又可以使每次划分长度从长到短枚举，使程序尽快逼近最优解，从而同时增强剪枝条件一的效果。

这一部分的实现，首先需要求 MaxT 。

$\text{MaxT}[i][S]$ = 炸弹 i ，从 S 开始炸，可以炸到的最大编号
如果，炸弹 i 炸不到 S ，则 $\text{MaxT}[i][S] =$

$\text{MaxT}[i][S]$ 可以用动态规划的方法解决。

状态转移方程为：

$\text{MaxT}[i][S] =$ 炸弹 i 炸不到 S $S-1$
炸弹 I 炸得到 S $\text{MaxT}[I][S+1]$

$\text{MaxT}[I][m+1] = m$

MaxT 的时间复杂度为 $O(n^2)$ 。

具体实现方法：

考虑二分图右边的 n 个结点（ n 颗炸弹）

,

如果结点 i 未匹配，则 i 被认为可以使用。

如果结点 i 已匹配，假如从任何一个未匹配点出发存在一条到达 i 的交错路，并且 i 为外点，则 i 也被认为可以使用。

所以 $\max L = \max(\max T[i][S] \mid i \text{ 可以使用})$ ；

具体实现方法：

计算所有从未匹配点出发的交错路所能到达的已匹配点，从每一个未匹配点出发，宽度优先搜索，只要 $O(n^2)$ 的时间。

可以证明，从未匹配点出发的交错路上的已匹配点一定为外点。

注意判断重复（如果一个已匹配点已经被确定为可以使用，那么不需要对它再扩展一次，因为当把这个已匹配点确定为可以使用的结点的时候，已经从这个结点扩展过，如果再扩展必将产生无谓的重复）

如果已经求出了 MaxL ，可以先求一组长度为 MaxL 的匹配 A ，这样对于所有长度在 $1-\text{MaxL}$ 范围内的划分， A 都是一组可行匹配。扩展一次增广路的复杂度为 $O(n^2)$ 。
这样大大节省了优化二的时间。

性能分析（3）：

通过以上的优化，所有数据都是瞬间出解，并且所有结果都是最优解。

甚至对 $n=200$ 的随机数据，也可以在瞬间出解，可见程序的效率有了很大的提高。

	最简单的搜索	优化的搜索	进一步优化的搜索
1	0.00	0.01	0.01
2	0.00	0.01	0.01
3	0.50	0.10	0.02
4	TimeOver	TimeOver	0.03
5	0.65	0.21	0.00
6	TimeOver	0.26	0.02
7	TimeOver	TimeOver	0.02
8	TimeOver	0.26	0.01
9	TimeOver	0.10	0.02
10	TimeOver	0.40	0.02

总结：本文中的两个例子都可以应用部分搜索 + 匹配的方法高效解决。

它们在思想上有着明显的相同点。一般的思维过程如下：

很难想到多项式算法，简单常规的搜索方法无法解决问题

希望减少搜索量

选择搜索的变量

判断其它变量是否可以用高效算法解决

Y

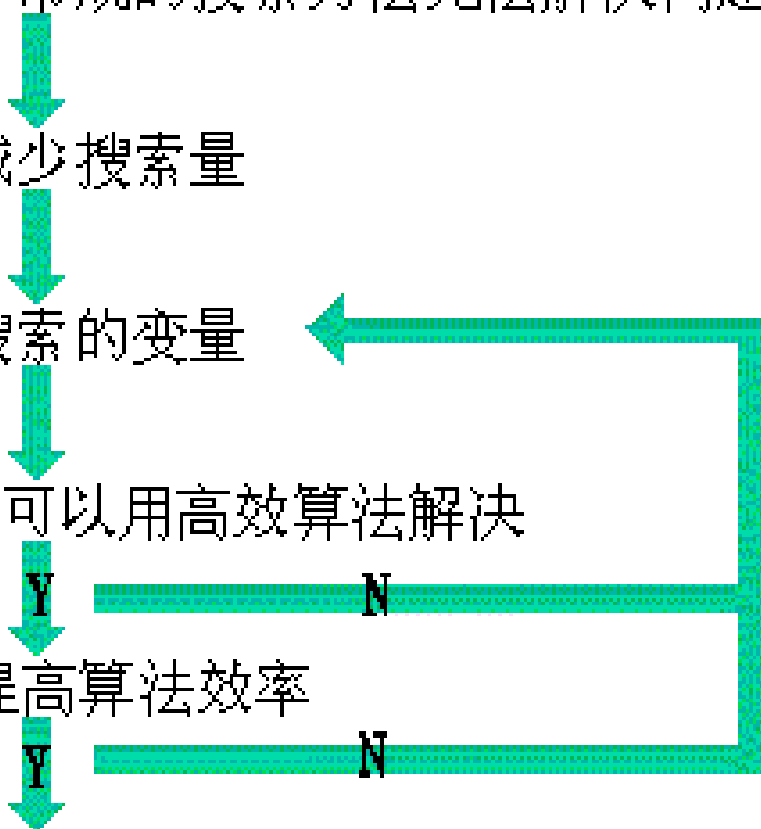
N

判断是否提高算法效率

Y

N

通过优化得到高效的搜索方法



一般的优化包括：

(1) 提前通过匹配判断，避免多余的搜索

(2) 判断时尽可能充分利用以前的结果，减少匹配的重复运算。

搜索同样可以和解方程、

贪心、
动态规划等高效算法结

总结

部分搜索 + 匹配算法体现了搜索与其他方法的有机结合，充分发挥两者的长处，相互弥补对方的不足，这就是其高效的主要原因所在。

因此，在搜索问题中灵活地应用部分搜索的方法，往往可以创造出奇效。

值得注意的是，部分搜索来解决搜索问题作为一种非常规的搜索方法。虽然在本文的例子中，部分搜索有着很多的过人之处，但是并不能认为常规方法一定不如非常规方法。大多数的搜索问题还是适合用常规的搜索方法的，所以只有充分把握部分搜索的特点，使之与常规的搜索融会贯通，才能真正得到高效的搜索算法。



谢谢大家！