十二硬币问题的进一步求解

朱福喜　卓　识

（武汉大学数学与计算机学院/国家多媒体软件工程技术研究中心，武汉 430072）

E-mail:fxzhu@whu.edu.cn

摘要　文章在用AO★算法求解一个智力难题——十二硬币问题的基础上，进一步就如何提高算法效率，如：降低内存消耗，提高计算速度、减少节点生成、防止重复搜索等方面做了一些努力，并把该问题推广到十二个硬币以上的情况，计算出较理想的结果。通过对计算结果的分析，找出了该问题的一般规律，并对结论给出了证明。

关键词　人工智能　AO★算法　十二硬币问题

文章编号 1002-8331-(2001)21-0132-04　**文献标识码** A　**中图分类号** TP301.6

The Further Solution of Twelve Coins Problem

Zhu Fuxi　Zhuo Shi

(College of Mathematics and Computer Science /National Engineering Center for Multimedia Software, Wuhan University, Wuhan 430072)

Abstract: After using AO* algorithm to solve the twelve coins problem and get all the solution of this problem, this paper takes further effort to enhance the efficient of the algorithm.For example, try to reduce the memory consume, speed up the computing, lessen the generating of the nodes, prevent the repeat search etc.Then generalize the problem into more than twelve coins and compute an ideal result.By analysis the results, It finds the general law of the problem and then proves the law.

Keywords: Artificial Intelligence, OA* algorithm, Twelve coins problem

1　问题描述

有12个硬币，已知有一个是不标准的（不能确定是轻是重），要求使用天平最多3次，找出那个不标准的硬币，并确定是轻的还是重的。该智力难题如果仅从智力游戏方面去解，比较困难。如果硬币数推广到39个硬币在4次内找出假币，则更加困难。

2　算法设计

2.1　状态表示及问题转化

该问题的状态空间可用五元组(lhs,ls,hs,s,t)表示，其中lhs表示状态或轻或重或是标准型的硬币个数，s、ls、hs分别为已确定为标准、轻标准、重标准型硬币的个数，t表示已称的次数。对于12硬币问题其初始状态为(12,0,0,0,0)，目标状态为(0,1,0,11,3)或(0,0,1,11,3)。

2.2　转换规则

根据当前状态(lhs,ls,hs,s,t)，选取$(lhs1,ls1,hs1,s1)$、$(lhs2,ls2,hs2,s2)$分别放在天平的左右两边，根据天平的偏向确定新的状态(lhs',ls',hs',s',t')分别有天平左偏规则、天平平衡规则、天平右偏规则(见参考文献[1])。

2.3　启发式函数

$F(lhs,ls,hs,s,t)=lhs*2+ls+hs-1$

3　算法的实现

3.1　有关术语定义

算法的基本思想是使用AI中AO*算法，其求解过程可以描述为一个与/或图(见图1)，在这个图中，结点$a1$、$b1$、$c1$构成一个与结点群$G1$，结点$a2$、$b2$、$c2$构成另一个与结点群$G2$，与结点群$G1$、$G2$构成一个或结点链，而$a1$、$a2$分别为与结点群$G1$、$G2$的代表结点。

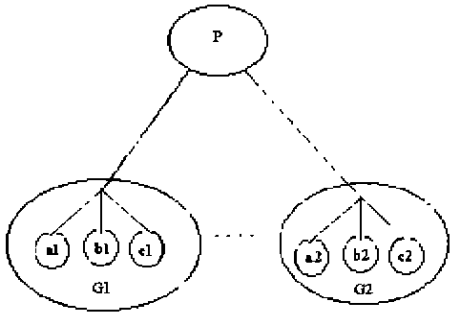

图1

3.2　数据结构及有关全局变量

a.结点表示

```
typedef struct cnode1{
  cnode1(int,int,int,int,int);//构造函数
  cnode1();//缺省构造函数
  int lhs,hs,ls,s,t;//状态数据
  struct cnode1 *brother;//指向与结点群中的兄弟结点，构成一循环链表
  struct cnode1 *exbrother,*exbrotherprev;//指向或结点链中的
```

作者简介：朱福喜，副教授，在职博士生，主要研究领域为人工智能、知识工程及分布计算。卓识，硕士生，主要研究领域为人工智能。

前后结点

```
struct cnode1 *next,*prev;//指向在 G 表中的前后结点 (或散列表中后结点)
struct cnode1 *parent;//指向父结点
struct cnode1 *pick;//指向选硬币方案
struct cnode1 *excellent;//指向最优的子结点群
char signal,inferor;//标志及权值
int nsussor;//子结点总数
} cnode;
```

b.散列表 SuccTab、FailTab、分别存放已成功的结点、已确定为失败的结点.

c.G 表,尚未处理结点链表,是一个按权值递增次序排列的有表头的双向链表。

d.每个结点的子结点所形成的或结点链,也是个按权值递增次序排列的双向链表。

3.3 主算法

STEP1.初始化散列表、在 G 表中加入起始结点 proot(12、0、0、0、0)

STEP2.循环、直到 G 表为空,或者根结点已标志为可解.

STEP2.1 从 G 表中抽取起第一个结点 (即权值最小的结点),设为 pnode

STEP2.2 如果 pnode 已可解,则跳至 STEP2.5。

STEP2.3 产生 pnode 的后继结点链。

STEP2.4 如果 pnode 不能产生有效的后继结点, 该 pnode 标志为"失败",并加到散列表 FailTab 中。在调用 DeleteBack (pnode)回溯删除祖先结点(必要时)后、跳至 STEP2。

STEP2.5 调用 BackWard(pnode)修改祖先结点的权值。

STEP3 若 G 表为空,则打印"不能找到"信息,否则打印出该树。

STEP4 删除 G 表、散列表的结点、回收空间。

3.4 产生结点 pnode 的后继或结点链的算法

该算法所采取的步骤有:

step1.pnode 的每一个后继结点 pnode1、pnode2、pnode3 一产生、便在散列表 SuccTable、FailTable 中查找其是否属于可解结点集,或属于不可解结点集。

step2. 如果一个与结点群 G1 (pnode1、pnode2、pnode3)可解,则删除 Lp 中的其它与结点群,使 Lp 仅由 G1 组成。

step3.在生成选择方案时、为避免对称情况的出现、在循环中始终要求

lhs1<=lhs2、lhs1+ls1<=lhs2+ls2、lhs1+ls1+hs1<lhs2+ls2+hs2,以减少生成结点数。

其主循环为:

```
for(lhs1=0;lhs1<=LHS/2;lhs1++)
 for(ls1=0;ls1<=LS;ls1++)
  for(hs1=0;hs1<=HS;hs1++)
   if(lhs1+ls1+hs1+S>=TotalCoin/4)
     //确保每次放在天平上的硬币个
     //数至少达到待验总数的一半
    for(lhs2=lhs1;lhs2<=LHS-lhs1;lhs2++)
     for(ls2=MAX(lhs1+ls1-lhs2,0);ls2<=LS-ls1;ls2++)
      for (hs2=MAX (lhs1+ls1+hs1-lhs2-ls2,0);hs2<=HS-hs1;
hs2++)do
```

{(1)如果左右两边的个数差异大于 S,则忽略本次选择方案、否则、根据差异数确定 S1 或 S2

(2)如果(lhs1+ls1+hs1+lhs2+ls2+hs2)==0,则忽略本次选择方案。

(3)根据选择方案及规则产生三个结点 pnode1、pnode2、pnode3、形成一个与结 点群 G1、并在该结点群中保存选择方案。

(4)如果在三个结点中有一个结点已在 FailTab 中(已被确定为失败结点),则删除与结点群 G1、并跳至 for 循环继续找 pnode 的下一个与结点群。

(5)调用函数 Link 把该与结点群 G1 连接到 Lp 中、

(6)如果 pnode1,pnode2、pnode3 都在 SuccTable 表中,即三个结点都有解,则

(6.1)调用 BackWard(pnode1)函数,删除 Lp 中的其他与结点群、使其仅由 G1 组成;并回溯修改 pnode 结点的祖先结点的权值。

(6.2)结束本算法,返回。

(7)把 pnode1,pnode2,pnode3 三个结点加入表 G 中、继续 for 循环、

}

4 结果及分析

对于 12 硬币问题,用上述算法求解 12 硬币问题,已求得该问题的全部解为 410 种, 然后改变硬币数及限定查找次数、得到结果如下表:

硬币数	限查次数	所需时间(秒)	实际生成不同结点个数	可生成不同结点总个数	修剪率 %	备注
12	3	0.050	33	455	7.25	成功
13	3	1.150	54	560	9.64	不可解
14	3	2.140	63	680	9.26	不可解
14	4	0.001	45	680	6.61	成功
24	4	0.270	96	2925	3.28	"
26	4	0.280	116	3654	3.17	"
28	4	0.940	137	4495	3.04	"
30	4	1.200	140	5456	2.56	"
32	4	2.030	163	6545	2.49	"
33	4	2.960	164	7140	2.29	"
34	4	4.500	189	7770	2.43	"
35	4	3.060	190	8436	2.25	"
36	4	4.890	191	9139	2.08	"
37	4	7.800	217	9880	2.19	"
38	4	9.660	219	10660	2.05	"
39	4	9.010	220	11480	1.91	"
40	4	*	*	*	*	不可解
40	5	4.843	248	12341	2.00	成功

从该表中,可得出以下几点结论:

(1)随着硬币数的增加,运行时间、结点个数呈几何级数增长,

(2)在上表中比较硬币数都为 14 个两种情况,在不可解的情况下运行了 2.140 秒,在可解情况下,运行时间小于 1 毫秒。比较硬币数都为 40 个两种情况, 在不可解的情况下 (限定 4 次)运行了数小时,毫无结果。在可解情况下(限定 4 次),运行时间小于 5 秒。这也表明了该算法通过及时删除结点、散列表等形式,使程序执行效率得到了提高。

5 问题的推广

在反复的计算过程中观察到称硬币的次数与硬币数之间的关系有:

称 3 次可在$(3^3-3)/2=12$ 个硬币中找出假币;

称 4 次可在$(3^4-3)/2=39$ 个硬币中找出假币;

称 5 次可在$(3^5-3)/2=120$ 个硬币中找出假币。

从上述结果,推出如下结论:

定理 1 使用天平 k 次可从$(3^k-3)/2$ 个硬币中找出假币。

证明:当 $n=2$ 时,$(3^2-3)/2=3$,显然成立,因为使用 2 次天平可在 3 个硬币中找出假币。

假设当 $n=k$ 时成立,即使用天平 k 次可在$(3^k-3)/2$ 个硬币

中找出假币。要证使用天平 $k+1$ 次可在 $(3^{k+1}-3)/2$ 个硬币中找出假币。

该结论的证明过程为：先将 $(3^{k+1}-3)/2$ 个硬币分为三部分，每部分各为 $(3^k-1)/2$，并将其中的两部分放到天平上。这样会出现三种情况，但实质上只有两种情况：天平平衡和天平倾斜。下面予以分别讨论。

5.1 天平平衡

此时要在 k 次内，在 $(3^k-1)/2$ 个硬币中找出假币。这似乎与本定理的结论不符，但实际上可用这样的方法处理，因为已有 (3^k-1) 个标准硬币存在，可以分别取 $(3^{k-1}+1)/2$ 和 $(3^{k-1}-1)/2+1(s)$ 放到左、右天平，其中 $1(s)$ 表示 1 个标准硬币。剩下有 $(3^{k-1}-1)/2$ 个硬币。于是，该问题转化为：

(1)如何在不平衡的情况下，在 $k-1$ 次内，在 3^{k-1} 个硬币中找出假币，这 3^{k-1} 个硬币都已知为 *ls* 或 *hs* 状态；

(2)如何在平衡的情况下，在 $k-1$ 次内，在 $(3^{k-1}-1)/2$ 个硬币中找出假币，$(3^{k-1}-1)/2$ 个硬币的状态都为 *ths*，但已知至少已有 $(3^{k-1}-1)$ 个标准硬币存在。

对于(1)，可由引理 1 证明。对于(2)显然是本定理在天平平衡情况下的一个递归问题，这里可以按照同样的方法分解下去，当递推到 $k=2$ 时，$(3^2-1)/2=4$，即 4 个硬币要在 2 次称出假币，此时可将左天平放 2 个未知币，右天平放 1 个未知币和一个标准币，这样无论那种情况都可以再称一次后找出假币。当递推倒 $k=1$ 时，$(3^1-1)/2=1$，只要将这个未知币与标准币比较即可找出假币。

5.2 天平不平衡

此时要在 k 次内，在 3^k-1 个硬币中找出假币，因为已经称过一次，这 3^k-1 个硬币都已知为 *ls* 或 *hs* 状态，这个要求与(1)类似，且弱于(1)，所以也可以归结为(1)的证明。

对于上述(1)这种情况，可以通过一个有趣的例子加以说明，通过前面的表格可以看出硬币数为 13 时，用 3 次天平无法找出假币，但在求解 39 个硬币问题时，将 39 个硬币分为 3 份，其中的两份放到天平称过且平衡后，剩下的 13 个却可以在 3 次找出假币。其称法为：PICKUP([4,0,0,1],[5,0,0,0])，即左天平放 4 个完全未知币和 1 个标准币，右天平放 5 个完全未知币其详细解法见附录。

引理 1：当 3^k 个未知硬币的状态为 ls 或者为 hs 时，则可以在 k 次称出假币。

证明：将 3^k 个硬币分 3 份，3^{k-1} 个放到左天平，3^{k-1} 个放到右天平，放置时将 ls 和 hs 状态的硬币对称放置，即天平一边放 ls 的硬币 3^{k-2} 个，放 hs 的硬币 2.3^{k-2} 个，另一边则放 ls 的硬币 2.3^{k-2} 个，放 hs 的硬币 3^{k-2} 个，这种取法可描述为：

PICKUP([0,3^{k-2},2.3^{k-2},0],[0,2.3^{k-2},3^{k-2},0])

这样无论天平是左倾还是右倾，按照计算规则，在天平上总有 $3^{k-2}+2.3^{k-2}=3^{k-1}$ 个硬币为标准的，此时已知未放到天平的 3^{k-1} 个硬币是标准的，所以只会剩下 3^{k-1} 个硬币是未知的，但它们的状态仍 ls 或者为 hs。如此递推下去，对于状态的总数为 3^{k-1}，可以在 k-1 次称出假币。当 k=1 时，不难看出 3^1 个硬币其状态为 ls 或 hs，可以 1 次称出假币。

下面的定理说明，在 $(3^k-3)/2$ 之上再加一个，就不能在 k 次称出假币。

定理 2：使用天平 k 次不可能从 $(3^k-1)/2$ 个硬币中找出假币。

证明：要使使用天平的次数最少，必须使取出到两个天平的硬币与剩下的硬币之间均匀分配，否则会使某一部分剩余的未知硬币太多，而不能在规定的次数内找出假币。下面考察三种平均或接近平均的情况：

(1)首先看两个天平各取 $3^{k-1}/2$ 个硬币，则剩下有 $(3^{k-1}-1)/2$ 个硬币。但是这是行不通的，因为 3^k 总是一个奇数，$3^{k-1}/2$ 不是一个整数。

(2)然后再看两个天平各取 $(3^{k-1}+1)/2$ 个硬币，则剩下有 $(3^{k-1}-3)/2$ 个硬币。这也是行不通的，因为天平上的 $(3^{k-1}+1)$ 个硬币被称一次后，有 $3^{k-1}+1$ 个状态要区分，例如天平左倾，则要区分每个在左天平的硬币是否为重的假币，同样要区分每个在右天平的硬币是否为轻的假币，但称天平一次最多只能区分三种状态，称天平 k-1 次最多只能区分 3^{k-1} 种状态。

(3)其次再看两个天平各取 $(3^{k-1}-1)/2$ 个硬币，则剩下有 $(3^{k-1}+1)/2$。这还是行不通的，因为天平平衡时，剩下的 $(3^{k-1}+1)/2$ 个硬币都为轻重标准型，因此有 $3^{k-1}+1$ 种状态要区分，这在 k-1 次内是不能够完成的。

以上三种情况都说明无法使用天平 k 次从 $(3^k-1)/2$ 个硬币中找出假币

值得注意的是对于上述第(1)种情况，似乎与引理 1 的结论相悖，因为它的状态数 $2(3^k-1)/2=3^{k-1}$，没有超过 3^{k-1}，但要注意这是第一次称硬币，没有标准硬币"帮忙"。若是有标准硬币的话，就可以在天平的一方放置 $(3^{k-1}+1)/2$ 个硬币，另一方放置 $(3^{k-1}-1)/2+1$ 个硬币，这样就可以有效地在规定次数内找到假币。

附录：39 个硬币解法。

(39,0,0,0)天平两边放硬币：(13,0,0,0)(13,0,0,0)

1.1(0,13,13,13)天平两边放硬币：(0,6,3,0)(0,6,3,0)

1.1.1(0,3,6,30)天平两边放硬币：(0,0,0,6)(0,3,3,0)

1.1.1.1 (0,0,3,36) 天平两边放硬币：(0,1,0,0)(0,1,0,0)

1.1.1.1.1(0,0,1,38)

1.1.1.1.2(0,0,1,38)

1.1.1.1.3(0,0,1,38)

1.1.1.2(0,0,3,36)，略，见 1.1.1.1

1.1.1.3(0,3,0,36)，天平两边放硬币：(0,0,1,0)(0,0,1,0)

1.1.1.3.1(0,1,0,38)

1.1.1.3.2(0,1,0,38)

1.1.1.3.3(0,1,0,38)

1.1.2(0,7,1,31)天平两边放硬币：(0,0,1,3)(0,1,3,0)

1.1.2.1 (0,1,1,37) 天平两边放硬币：(0,0,0,1)(0,0,1,0)

1.1.2.1.1(0,0,0,39)此种情况不可能

1.1.2.1.2(0,0,1,38)

1.1.2.1.3(0,1,0,38)

1.1.2.2(0,3,0,36)略，见 1.1.1.3

1.1.2.3(0,3,0,36)略，见 1.1.1.3

1.1.3(0,3,6,30)略，见 1.1.1

1.2(13,0,0,26)天平两边放硬币：(4,0,0,1)(5,0,0,0)

1.2.1(0,4,5,30)天平两边放硬币：(0,0,0,6)(0,3,3,0)

1.2.1.1(0,0,3,36)略，见 1.1.1.1

1.2.1.2 (0,1,2,36) 天平两边放硬币：(0,0,0,2)(0,1,1,0)

1.2.1.2.1(0,0,1,38)

1.2.1.2.2(0,0,1,38)

1.2.1.2.3(0,1,0,38)

1.2.1.3(0,3,0,36)略,见 1.1.1.3

1.2.2(4,0,0,35)天平两边放硬币:(0,0,0,3)(3,0,0,0)

1.2.2.1(0,0,3,36)略,见 1.1.1.1

1.2.2.2 (1,0,0,38) 天平两边放硬币:(0,0,0,1)(1,0,0,0)

1.2.2.2.1(0,0,1,38)

1.2.2.2.2(0,0,0,39)此种情况不可能

1.2.2.2.3(0,1,0,38)

1.2.2.3(0,3,0,36)略,见 1.1.1.3

1.2.3(0,5,4,30)天平两边放硬币:(0,0,1,4)(0,2,3,0)

1.2.3.1(0,1,2,36)略,见 1.2.1.2

1.2.3.2(0,1,2,36)略,见 1.2.1.2

1.2.3.3(0,3,0,36)略,见 1.1.1.3

1.3(0,13,13,13)略,见 1.1(收稿日期:2000 年 9 月)

参考文献

1.朱福喜,余亮.用 AO* 算法求解一个智力难题[J].计算机工程与应用,2001;37(3):69-70

2.周祥和,戴大为.麦卓文编译自动推理引论及其应用[M].武汉大学出版社,1987.12

3.L Wos,L Henschen.Automated theorem proving 1965-1970 in The Automation of Reasoning:Collected papers from 1957-1970[M].Jorg Sickmann,Graham Wrightsoned.Springer-Verlag,New York,1983

4.Nils J Nilsson.Artificial Intelligence:A New Synthesis[M].Morgan Kaufann,1998

(上接 131 页)

一些三维用户界面仍然采用二维光标。但是,二维光标在三维视觉空间中很不自然。而且由于二维光标只有两个自由度,用它来完成三维空间中六自由度的交互操作不仅不自然,而且十分复杂,因此,三维用户界面的研究者和开发者普遍认为三维光标在三维用户界面中是十分必要的。三维光标还有一个好处,是可以使各种各样的三维交互设备在用户界面中有统一的表示形式。

(2)三维 widgets

另一种被广泛用的三维交互方式是通过三维 widgets 进行交互操作。三维 widgets 是从 X-Windows 中的 widgets(如菜单、按钮等)概念引伸而来的,即三维画面中的一些小工具。用户右以通过直接控制它们使画面或画面中的三维对象发生改变,这就好像工人拿着螺丝刀拧螺丝一样。现有的一些 widgets 包括在三维空间中漂浮的菜单、用于点取物体的手形图标、平移和旋转指示器、透视墙等等。

三维 widget 现仍在探索阶段,许多三维用户界面的研究者正在设计和试验各种不同的三维 widget。人们希望将来能够建立一系列标准的三维 widget,就像二维图形用户界面中的窗口、按钮、菜单等。

目前,三维交互技术的研究已经在国际图形学界引起广泛的重视。人们普遍认为三维图形交互技术将成为新一代用户界面的关键技术之一。但是,三维交互技术的研究还很不成熟,已经开发出来的三维用户界面大部分离实用产品还有距离,许多重要问题仍在研究之中,如:三维交互原语、三维交互模型,三维交互用户界面的软件结构等等。因此,为了使人们可以充分利用三维交互的强大功能,用户界面和交互技术的研究者们还需要付出大量的努力。

必须指出,人机交互技术是一种综合性的集成技术,除文章所介绍的几个方面外,还涉及到语音识别、表情识别、手写识别等多项技术。(收稿日期:2000 年 10 月)

参考文献

1.Andries V D.VR as a Forcing Function:Software Implication of a New Paradmg Advanced Virtual Reality Applications[C].Course Notes 2 of SIGGraph'94,1994

2.Bellik Yacine.Modality Integration:Speech and Gesture[M].In:Cole R A,et al.(ed.),Survey of the State of the Art in Human Language Technology,Ch9.4,1996

3.Bernse Niels O.Modality Theory:Supporting Multimodal Interface Design[M].Proc.of the ERCIM Workshop on Multimodal Human-computer Interaction,1993:13-23

4.Bernse Niels O.Foundations of Multimodal Representations:A Taxonomy of Representational Modalities Interactintg with Computers.1994;6(4):347-371

5.Billinghurst M,Savage J,Oppenheimer P et al.The Expert Surgical Assistant:*An Intelligent Virtual Environment with Multimodal Input*. Medicine Meets Virtual Reality IV.,1995

6.Cohen P R.The Role of Natural Language in a Multimodal Interface [C].Proc of the 5th Annual ACM Symposium on User Interface Software and Technology-UIST'92,New York:ACM Press,1992:15-18

7.Encarnacao J,Foley J D et al.Research Issues in Perception and User Interface[J].IEEE Computer Graphics and Applications,1994.3:67-69

8.Feiner S K,McKeown K R.Automating the Generation of Coordinated Multimedia Explanations[M].In:Maybury M.T.(ed.),Intelligent Multimedia Interfaces,AAAI Press,1993:117-138

9.Hartson H R,Siochi A C,Hix D.The UAN:A User-Oriented Representation for Direct Manipulation User Interfaces[J].ACM Transactions on information Systems,1990;8(3):181-203

10.Jacob R J K.Survey and Examples of Specification Techniques for User-Computer Interfaces[R].NRL Report 8948,1986.4

11 Jacob R J K.A Specification Language for Direct Manipulation User *Interface[J].ACM Transactions on Graphics*,1986;5(4):283-317

12.Jacob R J K et al.Integrity and Separabilty of Input Devices[J].ACM Transactions on Computer-Human Interaction,1994.3:3-26

13.Jacob R J K.Eye Tradhing in Advanced Interfaces Design[M].In:Barfield W.and Furness T.(ed.),Advanced Interface Design and Virtual Environments,Oxford,1995:258-288