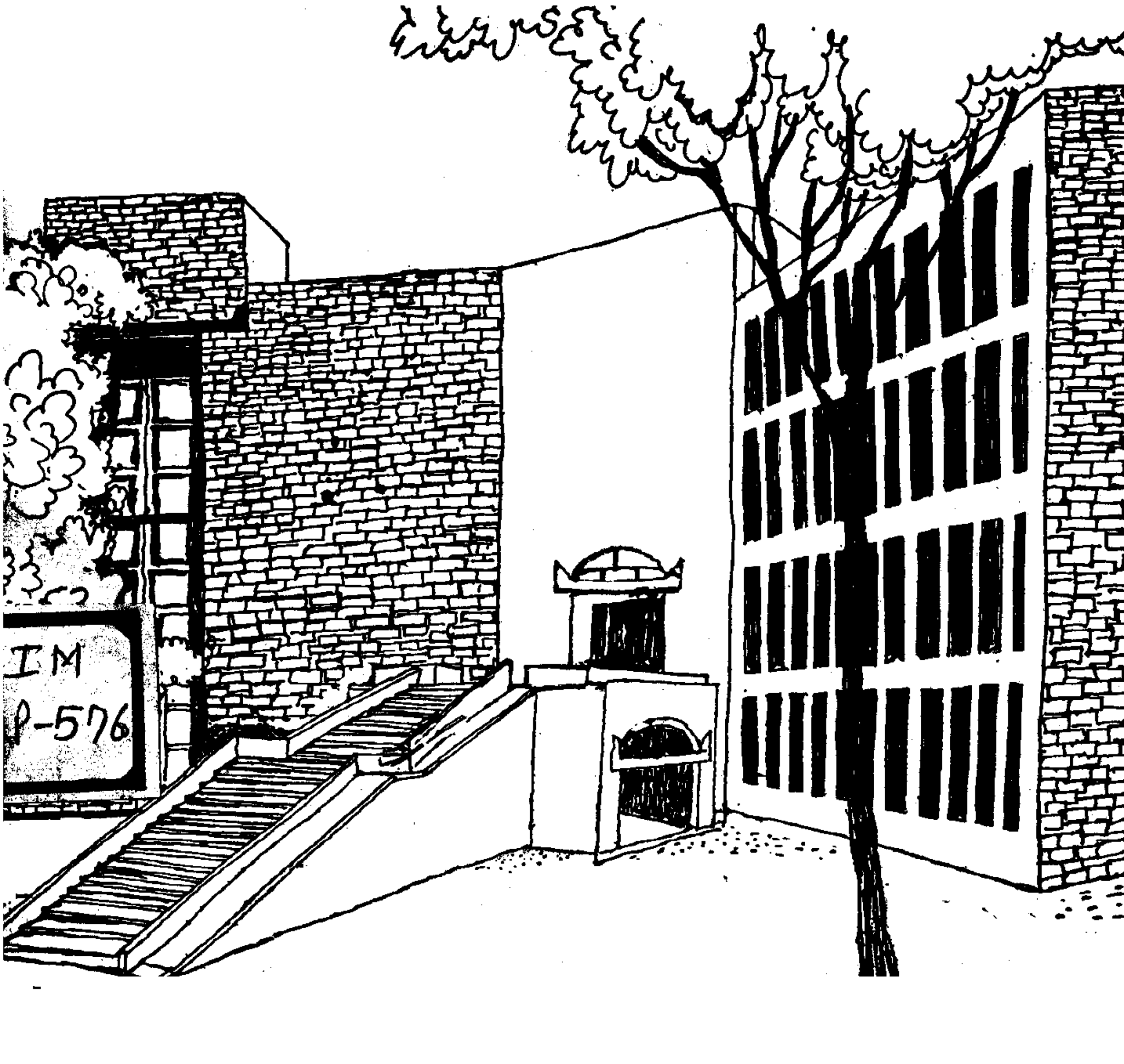
विद्याविनियोगाद्विकासः

IIM

AHMEDABAD

W. P.
576

Working Paper



THE DEFECTIVE COIN PROBLEM: AN ALGORITHMIC ANALYSIS

By

Suresh Ankolekar
Arindam Das Gupta
&
G. Srinivasan

W P No. 576
August 1985

The main objective of the working paper series of the IIMA is to help faculty members to test out their research findings at the pre-publication stage.

INDIAN INSTITUTE OF MANAGEMENT
AHMEDABAD-380015
INDIA

The Defective Coin Problem: An Algorithmic Analysis

Suresh Ankolekar

Arindam Das Gupta

G. Srinivasan

Abstract

The defective coin problem involves identification of defective coin, if any, and ascertain the nature of the defect (heavier/lighter) from a set of coins containing at the most one defective coin, using an equal-arm pan-balance. This paper gives algorithmic analysis of the problem. The solution strategy to minimise number of weighings required to detect the defective coin is based on problem reduction approach involving successive decomposition of the problem into subproblems until it is trivially solved. One of the two types of subproblems is visualised as combination of pair of antithetic problems, leading to an optimal solution procedure which is simply a term by term merger of corresponding antithetic procedures. Also, the algorithm is capable of generating all possible optimal solutions.

The Defective Coin Problem: An Algorithmic Analysis

Suresh Ankolekar
Arindam Das Gupta
G. Srinivasan

Consider a set S of identical-looking coins one of which may be defective, being lighter/heavier compared to others. The problem is to identify the defective coin, if any, and ascertain the nature of the defect (lighter/heavier), using only an equal-arm-pan-balance, such that number of weighings required for the purpose are minimised.

Algorithm Development

Intuitively, one of the solution strategies would be to successively reduce the problem-size until the reduced problem can be trivially solved. In other words, we successively partition the set of soins into smaller subsets, and pursue the subset containing the defective coin for further decomposition until no more decomposition is required. Obviously, the weighing process would have to indicate the subset to be pursued for further decomposition. Therefore, the correct interpretation of weighing outcomes would be crucial to optimum decomposition.

There are three possible outcomes in the weighing process, namely, both the pans balanced, left-pan up & right-pan down, and left-pan down & right-pan up, respectively. Consequently, at any stage we can decompose a problem by partitioning the set of coins into at <u>most three</u> subsets, so that each of the outcomes uniquely indicates one subset to be pursued for further decomposition.

Suppose we partition the given set S into

S_0 : set of coins kept aside, not participating directly in weighing

S_1 : set of coins in left-pan

S_2 : set of coins in right-pan

Let

W(X) : weight of pan containing set X of coins

Note that with the limited apparatus that we have, we would not be in a position to know actual value of W(X). We shall use the notion of weight only in a comparative sense, to express the outcomes of weighing process, e.g.

$$W(S_1) = W(S_2), \; W(S_1) < W(S_2) \text{ or } W(S_1) > W(S_2).$$

Let the number of coins in a set be denoted by the lower case letter corresponding to the set notation in upper case letter, e.g. number of coins in set X would be x.

We shall express the interpretation and algorithm development in terms of algorithmic language using standard program structures in structured English such as

procedure	case	if ..
..	..	then ..
endprocedure	endcase	else ..
		endif

Interpretation of Weighing Outcomes

The weighing process and its interpretation can be expressed as follows:

```
Partition S into S0, S1, S2 such that s1 = s2

weigh S1 and S2

case

  W(S1) = W(S2)   :  "the defective coin, if any, is in S0"
                     "S1 U S2 is a set of good or standard
                                                     coins"

  W(S1) < W(S2)   :  if the defective coin is lighter
                        then
                            "it is in S1".
                        else
                            "it is heavier and in S2"
                     endif
                     "S0 is a set of good coins"

  W(S1) > W(S2)   :  if the defective coin is heavier
                        then
                            "it is in S1"
                        else
                            "it is lighter and in S2"
                     endif
                     "S0 is a set of good coins"

endcase
```

The first weighing changes the complexion of the problem both in terms of size and structure as follows:

i) we are able to trap the defective coin, if any, in either S_0 or $S_1 \cup S_2$.

ii) if the pan-balance is balanced then the defective coin, if any, has escaped in S_0, and we can safely declare $S_1 \cup S_2$ as set of good coins.

iii) if the pan-balance is unbalanced then the defective coin has been trapped in $S_1 \cup S_2$, and we can safely declare S_0 as set of good coins.

iv) if the defective coin is in $S_1 \cup S_2$, we can make statement about it being specifically in S_1 or S_2 conditional to nature of its defect. For example, $W(S_1) < W(S_2)$ can be interpreted as consequence of the lighter defective coin being in S_1 or heavier defective coin being in S_2, but not both, since there is only one defective coin.

v) in addition to pan-balance, we now have a set of good coins, which were not available before the first weighing. That means, in future, while decomposing a problem of F coins by partitioning into three subsets, namely, F_0, F_1 and F_2, we need not force the condition $f_1 = f_2$ unlike in the first weighing. Because, we can equalise the number of coins in each pan by appropriately adding good coins either to F_1, or to F_2, depending upon which of them is a smaller subset. Henceforth, the equalising of coins in two pans is to be taken for granted, and accordingly $W(X)$ is to be interpreted as

W(X) : weight of the pan containing (among other good coins, if any), the set X of coins

rather than weight of the set X of coins, per se.

The above interpretation throws up two related problems distinctly different from the original problem we started with. Let us express all the problems precisely and represent them symbolically.

Before weighing, we had,

P(S) : Given a set S of identical-looking coins containing at most one defective coin, identify the defective coin and nature of its defect (lighter/heavier), if any, using only a pan-balance such that number of weighings required for the purpose is minimised.

After first weighing we have,

$PA(S_0)$: Given a set S_0 of identical-looking coins containing at most one defective coin, identify the defective coin and nature of its defect, if any, using a pan-balance and a set of good coins such that number of weighings required for the purpose is minimised.

$PB(S_1, S_2)$: Suppose there are two sets S_1 and S_2 of identical-looking coins, one of them containing exactly one defective coin such that the defective coin if lighter is contained in S_1, and if heavier in S_2. The problem is to identify the defective coin and nature of its defect. Using a pan-balance and a set of good coins such that number of weighings required for the purpose is minimised.

We note that $P(S)$ and $PA(S_0)$ are structurally somewhat similar except that $PA(S_0)$ has a set of good coins available for use in weighing process. $PB(S_1,S_2)$ is a two parameter problem, and may give a false impression that nature of defect is known. In $PB(S_1,S_2)$ problem, we are only making a statement about defective coin belonging to S_1 or S_2 conditional to nature of defect which is unknown, and finding it is part of the problem.

Having recognised the problem $P(S)$, $PA(S_0)$ and $PB(S_1,S_2)$, we may tentatively express a possible solution procedure to solve $P(S)$ as follows:

```
procedure P(S)
   partition S into subsets S0,S1 and S2 such that s1 = s2
   if W(S1) = W(S2)
      then
          solve PA(S0)
      else
          if W(S1) < W(S2)
             then
                 solve PB(S1,S2)
             else
                 solve PB(S2,S1)
          endif
    endif
 endprocedure - P(S)
```


At this stage we shall not address directly to the issue of optimising on number of weighings. Given the above structure of procedure $P(S)$, if the optimal number of weighings to solve $P(S)$ is m^*, then we will have to partition the set S in such a manner as to ensure solution of $PA(S_0)$, $PB(S_1,S_2)$ and $PB(S_2,S_1)$ in (m^*-1) weighings.

In the solution procedure envisaged by us, the problems $PA(S_0)$, $PB(S_1,S_2)$ or $PB(S_2,S_1)$ will be solved by further decomposition of them into related problems until they become trivially simple capable of being solved without any more weighings.

The PA(X) Problem

Suppose we partition X into three subsets X_0, X_1, X_2 as in P(S) problem. If we subject X_1 and X_2 to weighing process, the outcomes may be interpreted as follows:

i) if $W(X_1) = W(X_2)$ then the defective coin, if any has escaped in X_0, and we will have to solve a $PA(X_0)$ problem to pursue it further. This is a recursive situation, and we can go on and on until PA(X) is trivially solved requiring no more weighing, which is possible only when we reach $PA(\emptyset)$ in the process of successive decomposition. Interpretation of reaching $PA(\emptyset)$ problem is that there is no defective coin in the original set, since we pursue PA(X) problem only when we fail to trap the defective coin in the subsets directly participating in the weighing process.

ii) if $W(X_1) \neq W(X_2)$ then we have been successful in trapping the defective coin in the subsets directly participating in the weighing process, and so we have a case of $PB(X_1,X_2)$ or $PB(X_2,X_1)$ problem depending on whether $W(X_1) < W(X_2)$ or $W(X_1) > W(X_2)$ respectively.

Expressing the procedure formally, we have

```
procedure PA(X)
   if x = Ø
      then "assert that there is no defective coin in the
                                         original set"
      else
         partition X into X0, X1, X2
         if W(X1) = W(X2)
            then solve PA(X0)
            else
               if W(X1) < W(X2)
                 then solve PB(X1,X2)
                 else solve PB(X2,X1)
               endif
         endif
    endif
  endprocedure
```

As in P(S) problem, the optimality in PA(X) problem would depend on the partitioning of X into X_0, X_1, X_2, in the sense that if PA(X) optimally requires K* weighings, then the partitioning must ensure that $PA(X_0)$, $PB(X_1,X_2)$ and $PB(X_2,X_1)$ do not require more than K*-1 weighings. Both P(S) and PA(X) problems require solution of PB(Y,Z) type of problem, and therefore we will be able to analyse the number of weighings required to solve them, only after doing so for PB(Y,Z) problem.

The PB(Y,Z) problem

In PB(Y,Z) problem, there is exactly one defective coin either in Y or in Z depending on whether the defective coin is lighter or heavier respectively, which itself is unknown.

The problem PB(Y,Ø) and PB(Ø,Z) appear to be somewhat related to PB(Y,Z). By definition, in PB(Y,Ø) problem, there is exactly one defective coin in Y, which is lighter, and similarly in PB(Ø,Z) problem, there is exactly one defective coin in Z, which is heavier. Therefore, PB(Y,Ø) and PB(Ø,Z) are not valid decomposition of PB(Y,Z), since together they imply two defective coins, one in each category of defect, whereas PB(Y,Z) contains exactly one defective coin. Therefore, it is not possible to pursue one of PB(Y,Ø) and PB(Ø,Z) unless we have prior knowledge of nature of defect or we are prepared to sacrifice one more weighing just to ascertain the nature of defect which would be seemingly suboptimal. On the other hand, PB(Y,Ø) and PB(Ø,Z) are potentially easier to solve compared to PB(Y,Z) since nature of defect is conceptually known in those problems.

Then, the interesting question at this stage would be, can we pursue both PB(Y,Ø) and PB(Ø,Z) simultaneously, avoiding the separate weighing required only to ascertain nature of the defect?

Another related question would be, can we derive the procedure to solve PB(Y,Z) on the basis of procedures to solve PB(Y,Ø) and PB(Ø,Z)?

The problems PB(Y,Ø) and PB(Ø,Z) are structurally antithetic, and their procedure are likely to form mirror image with respect to each other. If we "merge" these procedures step-bystep, would the effect be as if we pursue these problems simultaneously, and hence leading to the solution of PB(Y,Z)?

Let us perform the merger of the procedures for PB(Y,Ø) and PB(Ø,Z) and analyse its validity vis-a-vis PB(Y,Z) problem. We shall pursue the same "divide-and-rule" policy of decomposition to solve PB(Y,Ø) and PB(Ø,Z) as in case of PA(X).

Examining the definition PB(Y,Z) problem closely, we observe that PB(Y,Ø : y=1) stands trivially solved requiring no further weighing, since the definition asserts that Y is the defective coin and is lighter, and similar is the case of PB(Ø,Z : z=1).

Procedures to solve PB(Y,Ø) and PB(Ø,Z) can be developed along the lines of procedure PA(X). Situation is again recursive with above trivial problems terminating the recursion.

We shall juxtapose the procedures for PB(Y,Ø) and PB(Ø,Z) so that we can merge them step-by-step.

```
procedure PB(Y,Ø)
  if y = 1
    then "assert Y is defective
           and lighter"
    else
      partition Y into Y0,Y1,Y2
        if W(Y1) = W(Y2)
          then solve PB(Y0,Ø)
          else
            if W(Y1) < W(Y2)
              then solve PB(Y1,Ø)
              else solve PB(Y2,Ø)
            endif
        endif
    endif
endprocedure - PB(Y,Ø)
```

```
procedure PB(Ø,Z)
  if z = 1
    then "assert Z is defective
           and heavier"
    else
      partition Z into Z0,Z1,Z2
        if W(Z1) = W(Z2)
          then solve PB(Ø,Z0)
          else
            if W(Z1) < W(Z2)
              then solve PB(Ø,Z2)
              else solve PB(Ø,Z1)
            endif
        endif
    endif
endprocedure - PB(Ø,Z)
```

Let us blindly merge these procedures step-by-step and within a step, term-by-term replacing reference to individual subsets by their union. We shall use

case
...
endcase

structure instead of complex nested if-then-else-endif to take care of multiple choice situation.

The merged procedures would look like,

procedure PB(Y,Z)
  case
    $y=1$ and $z=\emptyset$: "assert Y is defective and lighter"
    $y=\emptyset$ and $z=1$: "assert Z is defective and heavier"
  otherwise:
      partition Y into Y_0, Y_1, Y_2
      partition Z into Z_0, Z_1, Z_2 and combine them to form $Y_0 \cup Z_0$, $Y_1 \cup Z_1$, $Y_2 \cup Z_2$
      if $W(Y_1 \cup Z_1) = W(Y_2 \cup Z_2)$
        then solve $PB(Y_0, Z_0)$
        else
          if $W(Y_1 \cup Z_1) < W(Y_2 \cup Z_2)$
            then solve $PB(Y_1, Z_2)$
            else solve $PB(Y_2, Z_1)$
          endif
      endif
   endcase
 endprocedure - PB(Y,Z)

As in previous problems, the optimality in PB(Y,Z) problem would be determined by the partitioning of Y and Z into Y_0, Y_1, Y_2 and Z_0, Z_1, Z_2 respectively. In other words, if PB(Y,Z) is optimally solvable within J^* weighing, then the partitioning must ensure that $PB(Y_0,Z_0)$, $PB(Y_1,Z_2)$ and $PB(Y_2,Z_1)$ do not require more than J^*-1 weighings.

Having blindly merged the two procedures $PB(Y,\emptyset)$ and $PB(\emptyset,Z)$ to derive the procedure $PB(Y,Z)$, let us analyse its validity by interpreting each step as follows:

1) The two assertions, namely, "assert Y is defective and lighter" for $PB(Y,\emptyset: y=1)$, and "assert Z is defective and heavier" for $PB(\emptyset,Z : z=1)$ are by definition obvious.
2) if $W(Y_1 \cup Z_1) = W(Y_2 \cup Z_2)$ then obviously we have failed to trap the defective coin in $Y_1 \cup Z_1 \cup Y_2 \cup Z_2$, and it has escaped in $Y_0 \cup Z_0$, specifically in Y_0 if lighter and in Z_0 if heavier, requiring us to pursue it through $PB(Y_0,Z_0)$.
3) The condition $W(Y_1 \cup Z_1) < W(Y_2 \cup Z_2)$ could have developed due to a lighter coin in $Y_1 \cup Z_1$ or a heavier coin in $Y_2 \cup Z_2$. Since Z_1 cannot inherit a defective lighter coin from Z, as also Y_2 cannot inherit a defective heavier coin from Y, the condition $W(Y_1 \cup Z_1) < W(Y_2 \cup Z_2)$ could have developed only due to a lighter coin in Y_1 or a heavier coin in Z_2, and we must pursue the defective coin through $PB(Y_1,Z_2)$. Similar argument will require us to pursue $PB(Y_2,Z_1)$ in case the condition $W(Y_1 \cup Z_1) > W(Y_2 \cup Z_2)$ holds.

The Optimality/Feasibility Considerations

The decomposition during the solution of $P(S)$, $PA(X)$ and $PB(Y,Z)$ has to be guided by the following considerations:

1) Number of subproblems into which a given problem may be decomposed,
2) Number of coins associated with each subproblem,
3) Composition of coins corresponding to two categories in $PB(Y,Z)$ problem, and

4) number of good coins available for equalising the coins in two pans at any stage.

The decomposition of a problem is clearly limited to three subproblems, since it is guided by the fact that the subproblem to be pursued next is to be uniquely indicated by the weighing outcomes are possible with a pan-balance.

The decomposition of a problem results into subproblems that are reduced in size, and may be of different type, e.g. P(S) is decomposed into PA(S_0), PB(S_1,S_2) and PB(S_2,S_1) where S_0,S_1,S_2 are partitions of S.The decomposition must ensure that if the present problem is solvable within P weighing steps, then each of the subproblems must be solvable within P-1 steps. The number of coins associated with a problem would clearly be a major factor determining number of weighing steps required to solve it, especially in the case of single parameter problems such as P(S) and PA(X). In the case of PB(Y,Z), being two parameter problem, the number of weighing steps may seem to be influenced by not only the size of the problem (y+z), but also by the composition, (Y,Z). We shall show that the number of steps are independent of the composition in a PB(Y,Z) problem.

We shall investigate the optimality aspects of problems in the order of PB(Y,Z), PA(X) and P(S) for obvious reasons that analysis of former problems is required in the analysis of the latter problems.

Let

b_k : the maximum size of the PB(Y,Z) problem that can be solved in k steps

a_k : the maximum size of the PA(X) problem that can be solved in k steps

c_k : the maximum size of the P(S) problem that can be solved in k steps.

$PB^*_k(Y,Z)$: the saturated PB problem, being the largest sized problem that can be solved in k steps, e.g. $PB(Y,Z\colon y+z = b_k)$

$PA^*_k(X)$: the saturated PA problem, e.g. $PA(X\colon x = a_k)$

$P^*_k(S)$: the saturated P problem, e.g. $P(S\colon s = c_k)$

The $PB^*_k(Y,Z)$ problem

By definition of PB(Y,Z) it is obvious that $b_0 = 1$, since $PB(Y,Z\colon y+z = 1)$ is trivially solved without requiring any weighings.

A PB(Y,Z) problem can be solved within one weighing if and only if all the three resulting subproblems can be trivially solved. In other words, $PB^*_1(Y,Z)$ can be seen as the aggregation of three PB^*_0 problems, namely,

$$PB^*_0(Y_0,Z_0) = PB(Y_0,Z_0\colon y_0+z_0 = 1)$$

$$PB^*_0(Y_1,Z_2) = PB(Y_1,Z_2\colon y_1+z_2 = 1)$$

$$PB^*_0(Y_2,Z_1) = PB(Y_2,Z_1\colon y_2+z_1 = 1)$$

Thus,

$$PB^*_1(Y,Z) = PB(Y_0 \cup Y_1 \cup Y_2,\ Z_0 \cup Z_1 \cup Z_2\colon y_0+y_1+y_2 + z_0+z_1+z_2=3)$$

$$= PB(Y,Z : y+z = 3)$$

giving us

$$b_1 = 3$$

We observe that the saturation of a PB(Y,Z) problem requiring one weighing is <u>independent</u> of its composition, and is determined only by the total number of coins associated with the problem. Thus,

$$PB_1^*(Y,Z) = PB(Y,Z:\ \underline{y=3\ \&\ z=0}\ \text{OR}\ \underline{y=2\ \&\ z=1}\ \text{OR}\ \underline{y=1\ \&\ z=2}\ \text{OR}\ \underline{y=0\&z=3})$$

since decomposition of any of the combination would result in three trivial (saturated) subproblems, each of which is either $PB(Y,Z:\ y=1\ \&\ z=0)$ or $PB(Y,Z:\ y=0\ \&z=1)$.

In general $PB_k^*(Y,Z)$ can be seen as aggregation of following saturated subproblems.

$$PB_{k-1}^*(Y_0,Z_0) = PB(Y_0,Z_0:\ y_0 + z_0 = b_{k-1})$$

$$PB_{k-1}^*(Y_1,Z_2) = PB(Y_1,Z_2:\ y_1 + z_2 = b_{k-1})$$

and $$PB_{k-1}^*(Y_2,Z_1) = PB(Y_2,Z_1:\ y_2 + z_1 = b_{k-1})$$

leading to

$$PB_k^*(Y,Z) = PB(Y,Z\ :\ y + z = 3\ b_{k-1})$$

giving

$$b_k = 3\ b_{k-1}$$

which together with $b_0 = 1$

gives

$$b_k = 3^k$$

It is fairly simple to prove that $\left\{PB(Y,Z:\ 3^{p-1} < y+z \leq 3^p)\right\}$ can be optimally solved in p weighing steps. In other words, optimal value of maximum number of steps required to solve a PB(Y,Z) problem is given by

$$p = \lceil \log_3 (y+z) \rceil$$ where $\lceil x \rceil$ is smallest integer greater than or equal to x

The $PA^*_k(X)$ problem

By definition of PA(X), it is obvious that $a_0 = \emptyset$, since PA(X: $x=\emptyset$) is trivially solvable without requiring any weighing.

A PA(X) problem can be solved within one weighing if and only if all the three resulting subproblems can be trivially solved, and $PA^*_1(X)$ can be seen as aggregation of following problems, namely,

$$PA^*_0(X_0) = PA(X_0: x_0 = \emptyset)$$

and one of the following:

$$PB^*_0(X_1,X_2) = PB(X_1,X_2: x_1 + x_2 = 1)$$

$$PB^*_0(X_2,X_1) = PB(X_2,X_1: x_2 + x_1 = 1)$$

leading to

$$PA^*_1(X) = PA(X: x=1)$$

and $a_1 = 1$

In general, for $PA^*_k(X)$,

$$PA^*_{k-1}(X_0) = PA(X_0: x_0 = a_{k-1})$$

and one of the following:

$$PB^*_{k-1}(X_1,X_2) = PB(X_1,X_2: x_1 + x_2 = b_{k-1})$$

$$PB^*_{k-1}(X_2,X_1) = PB(X_2,X_1: x_2+x_1 = b_{k-1})$$

leading to

$$PA^*_k(X) = PA(X: x = a_{k-1} + b_{k-1})$$

$$\text{and } a_k = a_{k-1} + b_{k-1}$$

solution of which results in

$$a_k = a_0 + \sum_{i=0}^{k-1} b_i$$

$$= \sum_{i=0}^{k-1} 3^i = \frac{1}{2}(3^k - 1)$$

As in case of PB* problem, it is simple to prove that

$$\left\{PA(X)\ \tfrac{1}{2}(3^{p-1}-1) < x \leq \tfrac{1}{2}(3^{p}-1)\right\}$$

can be optimally solved in p weighing steps. In other words, optimal value of maximum number of steps required to solve a PA(X) problem is given by $p = \lceil \log_3 (2x+1) \rceil$

The $P^*_k(S)$ problem

As in case of PB* and PA* problems, the $P^*_k(S)$ can be visualised as aggregation of,

$$PA^*_{k-1}(S_0) = PA(S_0\colon s_0 = a_{k-1})$$

and one of the following:

$$PB^*_{k-1}(S_1,S_2) = PB(S_1,S_2\colon s_1 + s_2 = b_{k-1})$$

$$PB^*_{k-1}(S_2,S_1) = PB(S_2,S_1\colon s_2 + s_1 = b_{k-1})$$

However, since $s_1 + s_2 = b_{k-1} = 3^{k-1}$ being odd number, it is not possible to ensure $s_1 = s_2$ as required in the first weighing. Therefore, the largest PB problem that can be feasibly handled during first weighing will have to have at least one coin less than the corresponding saturated PB* problem. Thus, $P^*_k(S)$ would be the aggregation of

$$PA^*_{k-1}(S_0) = PA(S_0\colon s_0 = a_{k-1})$$

and one of the following:

$$PB(S_1,S_2) = PB(S_1,S_2\colon s_1 + s_2 = b_{k-1} - 1)$$

$$PB(S_2,S_1) = PB(S_2,S_1\colon s_2 + s_1 = b_{k-1} - 1)$$

leading to

$$P^*_k(S) = P(S\colon s = a_{k-1} + b_{k-1} - 1)$$

and $c_k = a_{k-1} + b_{k-1} - 1$

solution of which results in

$$\begin{aligned} c_k &= a_k - 1 \\ &= \frac{1}{2}(3^k - 1) - 1 \\ &= \frac{1}{2}(3^k - 3) \end{aligned}$$

As in case of previous problems, it is a simple matter to prove that

$$\left\{ P(S): \frac{1}{2}(3^{p-1} - 3) < s \leq \frac{1}{2}(3^p - 3) \right\}$$

can be optimally solved in p weighing steps. Hence optimal value of maximum number of steps required to solve a P(S) problem is given by $p = \lceil \log_3 (2s + 3) \rceil$

The observation that the saturation of PB(Y,Z) problem is independent of the composition, gives us freedom to choose the composition without affecting the optimality. This has implications to the feasibility considerations, where any inequality between y and z must be equalised using good coins. It can be easily shown that there can never be shortage of good coins after the first weighing to solve the P(S) problem optimally, since in the worst case s/3 good coinds are generated during the first weighing, and further decompositions can be optimally carried out even with one good coin. Though out of large number of possible optimum solutions, some may turnout to be infeasible due to paucity of good coins at the initial stages, the number of optimum feasible solutions will still be combinatorially very large.

We now express the complete algorithm to <u>optimally</u> solve the P(S) problem. In the algorithm we shall keep track of the good coins so that the algorithm is capable of generating all possible optimal solutions.

Let

G : set of good coins

τ : number of steps required to solve the problem.

```
Initialize G <- Ø
 procedure P(S)
    find p such that 1/2 (3^(p-1) -3) < s <= 1/2 (3^p -3)
    partition S into S_0,S_1,S_2 such that
       s_0 <= 1/2 (3^(p-1) -1)
       s_1 + s_2 <= 3^(p-1)
        s_1  =  s_2
       τ <- 1
      if W(S_1) = W(S_2)
         then
              G <- S_1 U S_2
              solve PA(S_0)
         else
              G <- S_0
              if W(S_1) < W(S_2)
                 then
                     solve PB(S_1,S_2)
                 else
                     solve PB(S_2,S_1)
              endif
      endif
endprocedure - P(S)
```

Where: $\frac{1}{2}(3^{p-1}-3) < s \leqslant \frac{1}{2}(3^{p}-3)$, $s_0 \leqslant \frac{1}{2}(3^{p-1}-1)$, $s_1 + s_2 \leqslant 3^{p-1}$.

```
procedure PA(X)
    if x = 0
       then
          "assert that there is no defective coin"
       else
          find p such that 1/2 (3^(p-1) -1) < x <= 1/2 (3^p -1)
          partition X into {X_0, X_1, X_2} such that
                X_0 <= 1/2 (3^(p-1) -1)
                x_1 + x_2 <= 3^(p-1)
                |x_1 - x_2| <= g
          γ <- γ + 1
             if W(X_1) = W(X_2)
                then
                    G <- G U X_1 U X_2
                    solve PA(X_0)
                else
                    G <- G U X_0
                   if W(X_1) < W(X_2)
                      then
                          solve PB(X_1,X_2)
                      else
                          solve PB(X_2,X_1)
                   endif
             endif
    endif
endprocedure - PA(X)

procedure PB(Y,Z)
   case
      y = 1 and z=0 : "assert that Y is defective and lighter"
      y =0 and z =1 : "assert that Z is defective and heavier"
```

```
otherwise:
      find p such that 3^(p-1) < y + z ≤ 3^p
      partition Y and Z into Y0,Y1,Y2 and Z0,Z1,Z2 such that
            y0 + z0 ≤ 3^(p-1)
            y1 + z2 ≤ 3^(p-1)
            y2 + z1 ≤ 3^(p-1)
            y1 + z1 - y2 - z2 ≤ g
           γ ← γ + 1
           if W(Y1 U Z1) = W(Y2 U Z2)
             then
                  G ← G U Y1 U Z1 U Y2 U Z2
                  solve PB(Y0,Z0)
             else
                  G ← G U Y0 U Z0
                if W(Y1 U Z1) < W(Y2 U Z2)
                   then
                       solve PB(Y1,Z2)
                   else
                       solve PB(Y2,Z1)
                endif
           endif
      endcase
endprocedure - PB(Y,Z)
input "the set of coins", S
solve P(S)
print "number of steps", γ
end.
```

Some Related Problems

We have tackled P(S) and other related problems with the objective of minimising the maximum number of weighing steps required to detect the defective coin, if any, in a set of coins. Clearly, for saturated problems this will also be equal to the minimum number of weighing steps, since each of the subproblems also is saturated, if we are not prepared to deteriorate maximum number of steps. However, for unsaturated problems, at least one of the subproblems will be either unsaturated or saturated at a lower level requiring less than (p-1) weighings for a parent problem requiring p weighings. In such cases, the algorithm may terminate at less number of weighing steps than that is indicated by the upper bound of a corresponding saturated problem depending upon whether we have been able to trap the defective coin in the smaller subproblem. Therefore, in general, associated with any solution strategy, the number of weighing steps would follow a probability distribution, leading to interesting issues like,

i) behaviour of expected number of weighing steps with respect to number of coins, in our solution strategy,

and ii) a solution strategy seeking to minimise the expected number of weighing steps.

Another interesting situation would be, to have more than one defective coins in the given set of coins. The general situation is likely to be combinatorially uninteresting or degenerating into general sorting kind of situation. However, the problems like, at the most two defective coins with identical and known nature of defect may still be amenable to interesting algorithmic analysis.

Finally, there may be some interesting ramifications of our observation that merging of two antithetic procedures PB(Y,Ø) and PB(Ø,Z) leads to a valid procedure for PB(Y,Z), giving the effect of simultaneous tackling of PB(Y,Ø) and PB(Ø,Z) for solving PB(Y,Z).