A Novel Approach to Volume Determination

S. K. Pushpa
Associate Professor
Dept. of ISE
BMSIT&M, Bangalore

Shankar Kantharaj
Dept.of ISE
BMSIT&M, Bangalore

Neeraj V.
Dept. of ISE
BMSIT&M, Bangalore

ABSTRACT
With an assumption of having two measuring flasks of known in this problem, there are two bottles wherein one has a volume having values one greater and one lesser than the larger capacity than the other. The transfers between them required quantity, in a chemical industry, the problem is to take place in such a way that any time the liquid is poured find the required volume of chemical.Thesetype ofto the bottle from another source other than those two problems may also be found in beverages industry were the bottles, itmust be completely filled and anytime the customers inserting coins, and then they get the required bottle is poured back to the source it must be completely volume ofbeverage. Theproblemproposed is called the emptied and the transfer that takes place between those volume determination problem and a series of algorithms two bottles is done till either of the bottles are empty or have been designed to solve this problem. A mathematical filled to the brim. By making these transfers, the desired approach hasalsobeenpresented to solve the problem quantity that lies in between the capacities of the two mathematically ratherthancomputationallyalong with respective bottles is obtained with minimum number of the series of steps involved. Thetime complexities have also transfers possible, thus allowing us to find the minimum been derived and shown accordingly, number of transfers and the minimum number of fills.

General Terms
Algorithms, Volume Determination, Algorithm Analysis, Volume Measurements

Keywords
Required to obtain this desired quantity. This provides usVolume Determination problem, Larger, Smaller, with an indirect approach to obtaining volumes using Mathematical,Computationally,Series,Complexitiesalgorithms to show applications in measuring volume.

1. INTRODUCTION
This problem is used to find a specific quantity using two known quantities without making actual measurements. The various areas where this could be useful are stated below:

- In a chemistry lab, a specific quantity of the chemical can be obtained using two measuring flasks of known volume having values one greater and one lesser than the required quantity.
- In the case of petrol pumps volume of petrol can be pre-defined as opposed to the Conventional way of measuring value using the flow rate
- A soda machine can give out customized volume of the beverage in accordance to the coins inserted. This would in turn benefit both the consumer and the company as the sales of the company increases and even the consumer is satisfied

2. ANALYSIS
This is a mathematical approach to solve the problem theoretically. The steps to be followed have been shown accordingly.

Step 1:
Find the pass no 'I' by using B to A transfers and A to B transfers and consider the I which is lesser of the two. In each case the value for 'I' is found for which the equation holds. Then the smaller of the 'I' values is chosen for both directions of transfers.

B→A:

$[n-k+[(off)(I_{B\rightarrow A}-1)](modm)](modm)=0$

n=capacity of bottle B

m=capacity of bottle A

k=required capacity

off=n(modm)

$I_{B\rightarrow A}$=The pass number for B to A transfers

A→B:

$[k+[(off)(I_{A\rightarrow B}-1)](modm)](modm)=0$
$I_{A\rightarrow B}$=The pass number for A to B transfers

Step 2:

Find the offset sequence for the given problem. The offset sequence is the sequence of offset values for each pass of a given problem. To find the offset sequence, first find the value ofn(modm). Then put each value of the offset sequence which will be a sum of the offset value with the previous offset value, till the offset obtained is m which will be designated as 0 in the offset sequence.

Example:

If n=10,m=3,k=5

n(modm)=1

B→A Offset Sequence:

1→2→0

A→B Offset Sequence:

0→2→1

Choose the offset sequence according to the direction of transfers obtained from the previous step.

Step 3:

Find the transfer number 'J' according to transfer direction using the following formulae.

For B→A:

J=[n-k+[(off)($I_{B\rightarrow A}$-1)](modm)]/m

For A→B:

J=[k+[(off)($I_{A\rightarrow B}$-1)](modm)]/m

Then find the offset term to which this belongs to by using the formula,

(k(modm))

Step 4:

Find the 'J' value for each offset term before the (k(modm)) term in the sequence and add 1 for each term.

Step 5:

Add up all the J values obtained in step 3 and step 4 and this will be the final number of transfers required. The number of fills required is given by the pass number 'I'.

3. METHODOLOGY

Let the two bottles be bottle A and bottle B. For this example let the capacity of bottle A be 3 lts and the capacity of bottle B be 10 lts. bottle B must be filled up completely and each time the contents of bottle B has to be transferred to bottle A. Whenever the bottle A fills up the bottle must be emptied in order the make another transfer. Thus the sequence of transfers for each pass where each pass involves all the transfers that are possible when bottle B is completely filled with 10 lts.For the sake of claritythe transfers which give rise to a negative value and the transfers which go beyond the total capacity of the bottle B have not been shown.The last value of each pass is designated as the offset of the next pass. Let (total capacity of B – offset of pass) for any pass be called the reverse offset of that pass.

Example:

A-3lts

B-10lts

Transfers from B to A(10ltr bottle to 3 ltr bottle)

Pass1:

10-(1)(3)=7;

10-(2)(3)=4;

10-(3)(3)=1;

Pass 2:	*Pass 3:*
10-(1)(3)+1=8;	10-(1)(3)+2=9;
10-(2)(3)+1=5;	10-(2)(3)+2=6;
10-(3)(3)+1=2;	10-(3)(3)+2=3;
	10-(4)(3)+2=0;

For transfers from B to A the offset of any pass must be added to transfers of that pass. Theoffset for the first pass is set to an initial value of 0. After each pass bottle A is completely emptied again.

Transfers from A to B(3 ltr bottle to 10 ltr bottle)

Pass1:

(0)(3)+3=3;

(1)(3)+3=6;

(2)(3)+3=9;

Pass 2:	*Pass 3:*
(0)(3)+3-1=2;	(0)(3)+3-2=1;
(1)(3)+3-1=5;	(1)(3)+3-2=4;
(2)(3)+3-1=8;	(2)(3)+3-2=7;
	(3)(3)+3-2=10;

For transfers from A to B, the reverse offset of any pass must be subtracted from each transfer of that pass. The reverse offset for the first pass is set to an initial value of 0. After each pass bottle B is completely emptied again

- Transfers from A to B and B to A give the same values in reverse order of each other, thus showing that only one direction of transfers must be used to get a specific value
- For transfers from B to A, the transfers of each pass are in descending order
- For transfers from A to B, the transfers of each pass are in ascending order
- For transfers from B to A corresponding transfer numbers from each pass are in ascending order
- For transfers from A to B corresponding transfer numbers from each pass are in descending order

Example:
A-2lts
B-8lts

Transfers from B to A(8 ltr bottle to 2 ltr bottle)

Pass 1:
8-(1)(2)=6;
8-(2)(2)=4;
8-(3)(2)=2;
8-(4)(2)=0;

Transfers from A to B(2 ltr bottle to 8 ltr bottle)

Pass 1:

(0)(2)+2=2;
(1)(2)+2=4;
(2)(2)+2=6;
(3)(2)+2=8;

Example:
A-4
B-18

Transfers from B to A(18 ltr bottle to 4 ltr bottle)

Pass 1:	*Pass 2:*
18-(1)(4)=14;	18-(1)(4)+2=16;
18-(2)(4)=10;	18-(2)(4)+2=12;
18-(3)(4)=6;	18-(3)(4)+2=8;
18-(4)(4)=2;	18-(4)(4)+2=4;
18-(5)(4)+2=0;	

Table.1: Table showing feasibility of obtaining values in between for various bottle capacities

Capacity of bottle A(C_A)	Capacity of bottle B(C_B)	Feasibility for all values in between
even	odd	Yes
odd	even	Yes
odd	odd	Yes
even	even	No

Transfers from A to B(4 ltr bottle to 18 ltr bottle)

Pass 1:

(0)(4)+4=4;
(1)(4)+4=8;
(2)(4)+4=12;
(3)(4)+4=16;
(4)(4)+4-2=18;

Pass 2:

(0)(4)+4-2=2;
(1)(4)+4-2=6;
(2)(4)+4-2=10;
(3)(4)+4-2=14;

Example:
A-3

B-11

Transfers from B to A(11 ltr bottle to 3 ltr bottle)

Pass1:

11-(1)(3)=8;

11-(2)(3)=5;

11-(3)(3)=2;

Pass2:

11-(1)(3)+2=10;

11-(2)(3)+2=7;

11-(3)(3)+2=4;

11-(4)(3)+2=1;

Pass 3:

11-(1)(3)+1=9;

11-(2)(3)+1=6;

11-(3)(3)+1=3;

11-(4)(3)+1=0;

Transfers from A to B(3 ltr bottle to 11 ltr bottle)

Pass1:

(0)(3)+3=3;

(1)(3)+3=6;

(2)(3)+3=9;

Pass2:

(0)(3)+3-2=1;
(1)(3)+3-2=4;
(2)(3)+3-2=7;
(3)(3)+3-2=10;

Pass 3:

(0)(3)+3-1=2;
(1)(3)+3-1=5;
(2)(3)+3-1=8;
(3)(3)+3-1=11;

- If the capacity of one of the bottles is a multiple of the capacity of the other, only the quantities which are multiples of the smaller capacity bottle are obtained.
- If both the bottles have an even capacity, we will only be able to get the even quantities in between

Algorithms:
Algorithms to evaluate transfer number for transfers from B to A and A to B:

LARGE_TO_SMALL(n,m,k)

//The smaller bottle is kept empty at the beginning and the larger bottle full.

//The larger bottle is then emptied to the smaller bottle and refilled repeatedly till the desired value is obtained.

//This method allows us to find the minimum number of transfers and the minimum number of fills

//To obtain a desired quantity by making transfers only from larger bottle B to smaller bottle A

//Input: The capacity of larger bottle B is 'n',the capacity of smaller bottle A is 'm', the required quantity is 'k'

//Output:The minimum number of transfers required to obtain the desired quantity and the minimum number of fills required to do so

Step1:T←0; //'T' is the total number of

//transfers required

Step2:L←0;//'L' is the remaining quantity

//for every pass

Step3:off←n; //'off' is quantiy obtained on each

//transfer

Step4:For i←0 to n-1 do

Step5:j←0;

Step6:While off-m>=0 do

Step7:j←j+1

Step8:off←n-(m*j)+L

Step9:If off=k do

Step10:T←T+j

Step11:Display "number of transfers" ,T

Step12:Display "number of fills",i+1

End If Statement

End While Loop

Step13:j←j+1; //increment by 1 to transfer leftover

//quantity for each iteration

Step14:T←T+j;

Step15:L←off;

Step16:off←n-m+L;

End For Loop

SMALL_TO_LARGE(n,m,k)

//In the beginning the larger bottle is kept empty and the smaller bottle full

//Fill the larger bottle by emptying the contents of the smaller bottle to it.

//Empty the larger bottle when full and fill the smaller bottle when emptyand refill repeatedly till the desired value is obtained

//This method allows us to find the minimum number of transfers and the minimum number of fills

//to obtain a desired quantity by making transfers only from smaller bottle A to larger bottle B

//Input: The capacity of larger bottle B is 'n',the capacity of smaller bottle A is 'm', the required quantity //is 'k'

//Output: The minimum number of transfers required to obtain the desired quantity and the minimum

//number of fills required to do so

Step1:off←m;

Step2:L←0;

Step3:T←0;

Step4:For i←0 to i←n-1 do

Step5:j←0;

Step6:While off+m<=n do

Step7:off←(m*j)+m-L

Step8:j←j+1;

Step9:If off=k then

Step10:T←T+j;

Step11:Display "number of transfers" ,T

Step12:Display "number of fills",i+1

End If Statement

End While Loop

Step13:j←j+1;

Step14:T←T+j;

Step15:L←n-off;

Step16:off←m-L;

End For Loop

Decision(n,m,k)
//Used to determine which direction of transfers must be performed in order to get the required quantity in the least number of steps, by choosing either LARGE_TO_SMALL or SMALL_TO_LARGE

//Input: The capacity of larger bottle B is 'n',the capacity of smaller bottle A is 'm', the required quantity //is 'k'

//Output: A call to the function LARGE_TO_SMALL or SMALL_TO_LARGE

Step1:If n(modm) is 0 and k(modm) is not 0 then

Step2:Display "Not possible"

End If Statement

Step3:Else if n(mod2) is 0 and m(mod2) is 0 then

Step4:Ifk(mod2)is not 0 then

Step5:Display "Not possible"

Step6: Exit program

End If Statement

Step7: Else

Step8: Ifk(modm) = (n-k)(modm) then

Step9: If k<n/2 then

Step10: Call SMALL_TO_LARGE(n,m,k);

End If Statement

Step11: Else

Step12: Call LARGE_TO_SMALL(n,m,k);

End Else Statement

End If Statement

Step13: Else

Step14: For i←0 to i←n(modm)-1 do

Step15:If [(k(modm)+(m*i))(mod(n(modm))]=0 then

Step16: Exit for loop

End If Statement

End For Loop

Step17: If k(modm)=0 then

Step18: CallSMALL_TO_LARGE(n,m,k);

End If Statement

Step19: Else if n(modm)=1 then

Step20: If k(modm)<=m/2 then

Step21: CallLARGE_TO_SMALL(n,m,k)

End If Statement

Step22: Else

Step23: CallSMALL_TO_LARGE(n,m,k)

End Else Statement

End Else If Statement

Step24:Elseif ((k(modm)+(m*i))<=((n(modm)*m)/2 then

Step25: CallLARGE_TO_SMALL(n,m,k)

End Else If Statement

Step26: Else

Step27: CallSMALL_TO_LARGE(n,m,k)

End Else Statement

End Else Statement
End Else Statement
End Else If Statement

Step28: Else

Step29: If k(modm)=(n-k)(modm) then

Step30: If k<n/2 then

Step31: Call SMALL_TO_LARGE(n,m,k)

End If Statement

Step32: Else

Step33: Call LARGE_TO_SMALL(n,m,k)

End Else Statement

End If Statement

Step34: Else

Step35: For i←0 to i←(n(modm))-1 do

Step36: If((k(modm))+(m*i))(mod(n(modm)))==0 then

Step37:Exit for loop;

End If Statement

End For Loop

Step38:If k(modm)=0 then

Step39:Call SMALL_TO_LARGE(n,m,k)

End If Statement

Step40:Else if n(modm)=1 then

Step41: If k(modm)<=m/2 then

Step42: CallLARGE_TO_SMALL(n,m,k)

End If Statement

Step43: Else

Step44: CallSMALL_TO_LARGE(n,m,k)

End Else Statement

End Else If Statement

Step45: Else if ((k(modm))+(m*i))<=(((n(modm))*m)/2)

then

Step46:Call LARGE_TO_SMALL(n,m,k)

End If Statement

Step47: Else

Step48: Call SMALL_TO_LARGE(n,m,k)

End Else Statement

End Else Statement

End Else Statement

4. TIME ANALYSIS

For computing the time complexity of the problem the time efficiency of each algorithm is computed.

4.1 Time Efficiency of Decision Function

In the decision function all steps take a constant time to execute, except for the for loop steps. There are two such for loop steps in the function at two different parts of the function both executing for a maximum of 'm' number of times. Thus the efficiency in the worst case of the decision function is m.

The average case would therefore be roughly around m/2 and the best case would be about 1. Thus,

If the time complexity of decision function is D,

For worst case,

$D(t) \in O(m)$

For average case,

$D(t) \in \theta(m/2)$

For best case,

$D(t) \in \Omega(1)$

4.2 Time Efficiency Of Method Functions

In both the method functions, the basic operation is finding the offset for each iteration. Thus taking the for loop and while loop in a summation formula,

$$\sum_{i=0}^{n-1} \sum_{j=0}^{n-m} 1$$

$(n-m)\sum_{i=0}^{n-1} 1$

$=(n-m)(n-1)$

Letting the time efficiency for method functions be M,

$M(t) \in O(nm)$

Since the program will club decision function with either of large_to_small or small_to_large, but since these two functions have the same time complexity, the overall time complexity is obtained as follows,

Best case:

$\Omega(1) + \Omega(nm) = \Omega(nm)$

average case:

$\theta(m/2) + \theta(nm) = \theta(nm)$

Worst case:

$O(m) + O(nm) = O(nm)$

Thus the overall time complexity of the volume determination problem is O(nm).

5. RESULTS & DISCUSSIONS

The number of computations for this problem varies on the type of inputs given i.e. the quantities of the two bottles A and B and the required quantity. The growth of the complexity with input size for various types of inputs havebeen shown below

1. **Larger Quantity Is A Multiple Of The Smaller Quantity**

Ex)10,100

Table.2:Table of number of computations required for various values of 'k'

Required quantity 'k'	Number of computations
10	0
20	2
30	3
40	4
50	5

60	4
70	3
80	2
90	1
100	0

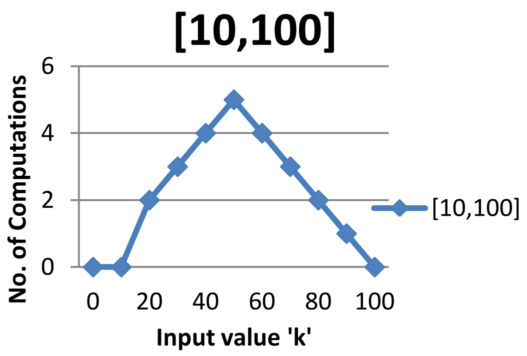

Figure.1: Larger quantity is a multiple of smaller quantity. Plot of required quantity 'k' versus no. of computations

For type 1 problems, the no. of computations increase linearly upto a certain point and then decreases linearly.

2. Larger And Smaller Quantities Are Even Numbers

Ex)4,30

Table.3:Table of number of computations required for various values of 'k'

Required quantity 'k'	Number of computations
4	0
6	6
8	2
10	5
12	3
14	4
16	4
18	3
20	5
22	2
24	6
26	1
28	7
30	0

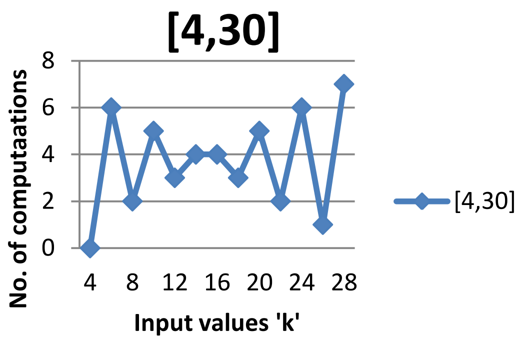

Figure.2: Both larger and smaller quantities are even numbers. Plot of required quantity 'k' versus no. of computations

For type 2 problems, the no. of computations Is symmetric around the central input value

3. One Of The Quantities Are Even And The Other Is Odd

Ex)2,15

Table.4:Table of number of computations required for various values of 'k'

Required quantity 'k'	Number of computations
2	0
3	6
4	2
5	5
6	3
7	4
8	4
9	3
10	5
11	2
12	6
13	1
14	7
15	0

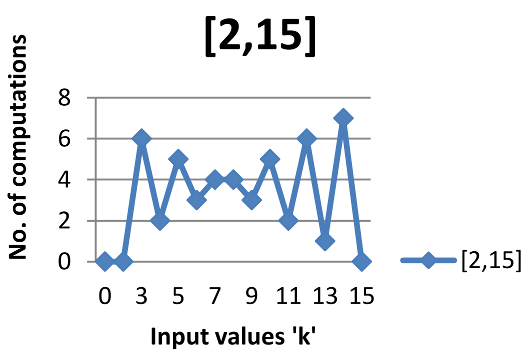

Figure.3: Larger one is odd and smaller quantity is even. Plot of required quantity 'k' versus no. of computations

For type 3 problems, the no. of computations fluctuates but this fluctuation decreases as the input size reaches a middle value and again increases with further increase in input size

4. Both Larger And Smaller Quantities Are Odd
Ex)5,27

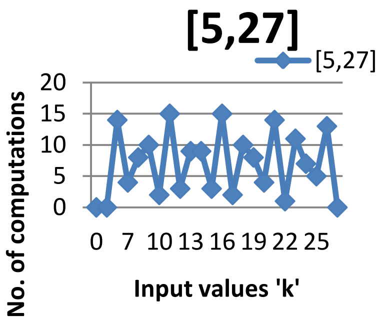

Figure.4: Both larger and smaller quantities are odd. Plot of required quantity 'k' versus no. of computations

For type 4 problems, the no. of computations have a variable tendency but between variations maintain a fixed value

Table.5:Table of number of computations required for various values of 'k'

Required quantity 'k'	Number of computations
5	0
6	14
7	4
8	8
9	10
10	2
11	15
12	3
13	9
14	9
15	3
16	15
17	2
18	10
19	8
20	4
21	14
22	1
23	11
24	7
25	5
26	13
27	0

6. CONCLUSIONS

The volume determination problem is a situation in which there are two bottles having two different capacities and are used to find a quantity in between. In this paper a method is presented to find the minimum number of computations that would be required to get a specific quantity. The time complexity of the problem was shown to be Θ (nm). From the graphs shown above it can beconcluded that the number of computations performed to obtain a specific quantity radically vary with different types of capacity values of the bottles and drastically depend on the sizeof input.

7. REFERENCES

[1] www.wikipedia.org/wiki/fuel_dispenser-a reference to how the fuel dispenser works.

[2] www.scientificpsychic.com/mind/aqua1-Exampe of the simiar problem is cited in the webpage.

[3] Levitin,Anany.The design and analysis of algorithms. New Jearsey:Pearson Education Inc.,2003

[4] C Programming: A Modern Approach, 2nd Edition By K. N. King

[5] C Programming Language (2nd Edition By B. W. Kernighan & D. M. Ritchie).

[6] www.studytonight.com/data-structures/time-complexity-of-algorithms-deducinggtime complexity of algorithms

[7] examples.yourdictionary.com/examples-of-ways-to-measure-volume.html-methods of measuring volume

[8] www.aaamath.com/mea69_x3.html-for conversion of units

[9] www.investopedia.com/university/technical/techanalysis 5.asp-importance of volume

[10] www.investopedia.com/articles/technical/02/010702.asp-improvmentin trading

[11] www.cell.com/cell/abstract/S0092-8674(16)30555-4-automated volume anaysis

[12] www.accaglobal.com/in/en/student/exam-support resources/.../CVP-analysis.html-cost-volumefprofit analysis

[13] https://www.cg.tuwien.ac.at/research/publications/.../haidacher-2011-VAM-Paper.pdf-volume analysis using cost volume multi similarity

[14] www.team-bhp.com › Team-BHP › BHP India › The Indian Car Scene-petrol story of hp