

# An Algorithm for Determining the Controllers of Supervised Entities at the First and Second Levels: A Case Study with the Brazilian Central Bank

Vinicius Guilherme Fracari Branco<sup>1</sup>, Li Weigang<sup>2</sup>, Maria Pilar Estrela Abad<sup>3</sup>,  
Jörg Denzinger<sup>4</sup>

<sup>1</sup> Deorf/Copec of the Brazilian Central Bank, Brasilia – DF, Brazil  
Vinicius.fracari@bcb.gov.br

<sup>2</sup> Department of Computer Science, University of Brasilia, C.P. 4466, CEP: 70919-970,  
Brasilia – DF, Brazil  
Weigang@unb.br

<sup>3</sup> Politec Informática Ltda., Brasilia – DF, Brazil  
Pilar@bsb.politec.com.br

<sup>4</sup> Department of Computer Science, University of Calgary, Canada  
Denzinge@cpsc.ucalgary.ca

**Abstract.** An algorithm was developed and implemented to find controllers from the stock shares for some financial institutions for the Brazilian Central Bank (BCB). The original problem is similar to a typical *Sum of Subset* problem that is suggested to be solved by a backtracking algorithm and the problem complexity is NP-complete. Usually BCB solves this problem manually which is time consuming and prone to errors. The heuristical approximation algorithm presented in this paper has polynomial complexity  $O(n^3)$  and is based on sub-routines for determining controllers at the first two levels. The paper describes the basic concepts and business rules currently employed in BCB, our algorithm and its major subroutines, it gives a brief complexity analysis and an example illustration at level 2. Our experimental results indicate the feasibility of an automation of the process of finding controllers. Though developed for BCB, our algorithm works equally well for other financial institutions.

## 1 Introduction

Although computer hardware development has increased the processing speed enormously over the last years, there are still a lot of real problems which cannot be solved optimally in reasonable time using modern computers, as computation theory has proven. To find controllers of an institution or company based on information about stock shares involving intermediate entities is one example of such a problem.

According to the rules of the Brazilian Central Bank (BCB), all financial institutions in Brazil, such as banks, investment and financial companies, credit societies, etc. should inform BCB about each ownership change. These institutions are called Supervised Entities (SE). After each stock movement, BCB needs to recalculate the potential controllers of an SE, which are defined as possible combinations of shareholders who have the privilege to vote directly and hold together 51% or above stock

shares of the SE. The importance of determining the potential controllers is significant. For example, if a SE bankrupted unreasonably, BCB needs to know its main shareholders in order to congeal immediately their properties for further juridical activities. In 2001 alone, there were 1100 financial institutions in whole Brazil and at least 500 SEs informed BCB about changes to their ownerships. This meant that BCB had to find out the controllers of all these institutions 500 times during 2001.

Usually the ownership of an SE is complicated, because of the transitive nature of the control property due to having stock shareholders. SE *A* may hold stocks of SE *B*, SE *B* may be the main controller of SE *C*. BCB divides the distribution of stock shares into levels. In this case, SE *C* is at level 1, SE *B* at level 2 and SE *A* at level 3. On the other hand, to determine the controllers is a difficult problem. In terms of mathematics, the problem is known as *Sum of Subset*, and the best known existing method to solve it optimally is by a backtracking algorithm with the complexity of  $O(p(n) 2^n)$  or  $O(p(n) n!)$ , where  $p(n)$  is a  $n^{\text{th}}$  order polynomial [2, 3, 4]. Here  $n$  corresponds to the number of individual shareholders. This means it is close to being impossible to determine the controllers from a huge number of stock shares for a big SE. To solve the problem of *Sum of Subset*, there have been some theoretical and empirical studies [5, 6, 7]. In the BCB case, the possible solution space is reduced quickly because of the great quantity of stock holders with a small amount of shares. This makes it possible to develop an approximation algorithm with polynomial complexity.

Even though all other management tasks of BCB are automatically processed, the determination of controllers is still manually checked. The main disadvantages of manually searching for controllers are: time consuming and human errors. For example, to find the controllers of a financial institute involved in a three-level SEs it can take a full day for two bank officers and sometimes, they may end up with a wrong result.

Recently, a project to automatically determine the controllers has been launched by BCB. The intention of it was the development of an algorithm and its implementation in a new system to establish the feasibility of an automation of the process of finding controllers. Though developed for BCB, the algorithm naturally works equally well for other financial institutions which have the same kinds of the applications.

This paper describes the basic concepts and business rules currently employed in BCB that have to be upheld by the algorithm, the subroutines for determining controllers at the first two levels (levels 1 and 2), the main algorithm, a brief complexity analysis of it and its application at the second level (for applications at first level, see [8]).

## 2 Concepts, Business Rules and Example Description

The basic concepts and business rules for dealing with controllers have been established by BCB. In the following, through some simple examples, we describe the problem of controller calculation at two levels. It is useful to understand them to develop an efficient model to resolve the problem [1].

## 2.1 Basic Concepts

A Supervised Entity (SE) is a company which is registered at BCB. At any moment, BCB would like to know who the controllers of that SE are based on ownership of the stock shares. To effect the calculation of the controllers, only the stocks or quotas held by shareholders with the right to vote are considered.

- *Types of shareholder controllers*
  - a) Natural Person (PF) - legal person represented by a single shareholder;
  - b) Artificial Person (AP) - legal person represented by a group of shareholders;
  - c) Individual Artificial Person (IAP, in Portuguese PJDD) - legal entities whose specific owners are not clearly defined.

The IAPs can be: cooperatives, associations and foundations, investment funds, entities of private providence, public companies headquartered in the exterior, financial institutions headquartered in the exterior, excepting solicitations of BCB.

An AP needs to be divided into PFs and/or IAPs who are the only entities to be considered as controllers. If an AP is found at a certain level, it surely indicates that there are upper levels of controllers at this SE. The society control of a SE is considered as definite when its final control is executed by one, and only one natural person (PF) or Individual Artificial Person (IAP) and indefinite when its final control is executed by more than one legal person.

- *Levels of the participation of the controllers in a Supervised Entity (SE)*
  - a) The first level: a SE is considered as on the first level when it contains only PFs and IAPs in the total composition of the stocks or quotas held by those with the right to vote.
  - b) The second, third and other levels: if a SE possesses some APs in their composition of voting shares, and for any AP it is possible to divide it into PFs and/or IAPs. Then his entity has more than one level of participation of the controllers. In our research, the algorithm was developed with the highest level being 3. This paper concentrates on the first two levels.

## 2.2 Business Rules

The business rules were established by BCB for a variety of situations. The following describes the rules for stock shares at levels 1 and 2.

- *At the level 1:*

The shareholders with the right of voting are considered to be the potential controllers in this work. If there is a large shareholder of a SE who holds more than fifty percent of the stock shares, he is already considered the unique controller of the company. If this condition does not exist, the calculation to find who the controllers are can use the following rules:

- a) assessing all the possible combinations of shareholders: if the total of the stocks from such a combination is 50% plus one of the voting shares they are considered the effective controllers of the SE;

- b) considering these combinations chosen in item "a", if we remove a shareholder with a determined percentage of stocks, and the total of the rest of the stocks remains larger than 50%, this removed shareholder will not be considered a controller of the SE.
- *At the level 2*
  - a) if a shareholder in level 2 is a unique controller of an AP, he is immediately considered as a unique shareholder PF of the AP at level 1 and, if he still possesses other shares in level 1, all his shares will be aggregated for calculating the controllers of the SE.
  - b) if there is no unique controller at level 2, all the controllers of this level are possible to be controllers. Therefore, no PF in an AP can sum his share within other APs and his share at level 1.
  - c) at level 1, if two or more APs are controlled by the same group of PFs, the shares of these APs will be summed together at level 1.
  - d) at level 1, if the APs are controlled by different groups of PFs at level 2, no stock shares of the same PF that constitutes within some APs and level 1 can be summed together.

### 2.3 Example Description

Suppose a Supervised Entity (SE), called BM Bank, with four stockholders: BM Automovios (AP1), Nelson da Silva (PF1), BM Insurance (AP2) and Andre de Barros (PF3), see Fig. 1. For BM Automovios (AP1), there are three stockholders: Nelson da Silva (PF1), Anderson Fernando (PF2), Marcelo Dantas (PF4). For GM Insurance (AP2), there are also three stockholders: Nelson da Silva (PF1), Anderson Fernando (PF2), Marcelo Dantas (PF4). All PFs in APs are considered at level 2. The first step of the analyses is to transfer APs to PFs and combine them into level 1. And then we have to find the controllers of the SE at level 1 (with only PF stockholders now). In section 4, we will present the solution for the example of BM Bank in Figure 1 using our algorithm.

## 3 Algorithm

To determine the controllers of a SE, three main subroutines are needed (resp. have been developed by us): a subroutine for determining controllers at level 1 (*sdcl1*), a subroutine for determining controllers at level 2 (*sdcl2*) and a subroutine for determining the possible combinations of the stock shares to form groups of controllers (*sdmlc*). *Sdmlc* arranges the most possible combinations of the stock shares for the group of controllers, but is still not proved that it can find all combinations. For every new combination, *sdcl1* is called to determine the controllers within *sdmlc*. When there is any AP to share the stock, *sdcl2* is used to analyze the stock shares of that AP and combine them to the level 1 as PFs. Figure 2 shows the general procedure to determine the controllers using these subroutines.

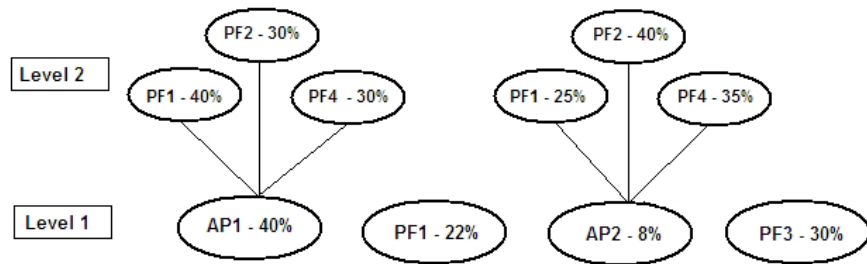


Fig. 1 The distribution of BM Bank stock shares

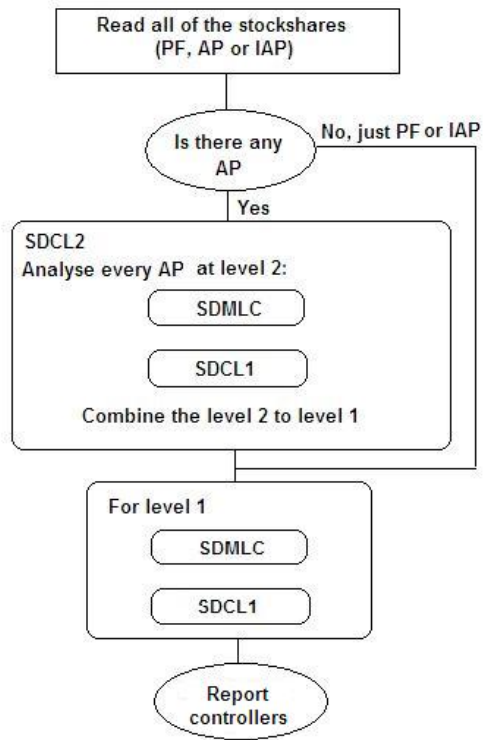


Fig. 2 The general procedure to determine the controllers

**3.1 The subroutine for determining controllers at level 1 (sdcl1)**

The task of this subroutine is to determine the controllers from a group of stock shares from a SE, if there are no APs. The stock shares of every shareholder are rep-

resented as percentages. The total of these percentages is 100%. The process of the algorithm is described in the following.

- 1) Rearrange the stock share percentages of the shareholders in decreasing order:  $A(1) > A(2) > \dots > A(k) > \dots > A(n)$ , where  $k = 1, 2, \dots, n$ .
- 2) Test the first shareholder. If his/her shares equal 50%, all shareholders of the SE are controllers and this finishes the process. If his/her shares are larger than 50%, the shareholder is the only controller of the SE and this finishes the process.
- 3) Create a vector  $Sum(k)$  to add one by one the stock shares from  $A(1), A(2), \dots$  until the  $A(k)$  with which  $Sum(k)$  arrives at 50% or higher. If  $Sum(k)$  equals 50%, all of the shareholders are controllers, and we can stop the program. If it is higher than 50%, all the shareholders whose stock is in  $Sum(k)$  are controllers,  $k = 1, 2, \dots, s$ .
- 4) Calculate the key variable  $Difference = 50\% - Sum(s)$ . Test the stock shares from  $k=s+1$ : if the stock share is larger than  $Difference$ , the shareholder is a controller. Suppose this test stop at  $A(d)$ ,  $s < d < n$ .
- 5) Create a vector  $Remainder\_Sum(k)$  which equals the sum of all remaining stock shares from  $k$  to  $n$ ,  $k = 1, 2, \dots, n$ . For  $k = d+1$ , do the following tests. If the  $Remainder\_Sum(k)$  is smaller than  $Difference$ , stop the program; if it equals  $Difference$ , all the remaining shareholders are controllers; if it is larger than  $Difference$ , the shareholder of  $A(k)$  is the controller. At the same time, a vector is generated i. e.  $V1(p) = V1(p) + A(k)$ ,  $k$  from  $d+1$ ,  $V1(0) = 0$ , if  $V1(p) > Difference$ ,  $V1(p) = V1(p) - A(k)$ . Suppose this test stops at  $A(h)$ ,  $d < h < n$ , so  $p = 0, 1, 2, \dots, h - d$ .
- 6) There are still some additional possible controllers. The test now begins from  $k = h + 1$ . if  $(Remainder\_Sum(k) + V1(p) = Difference)$  then all the remaining shareholders are controllers and we stop the program. If  $Remainder\_Sum(k) + V1(p) > Difference$ ,  $p = 0, 1, \dots, h - d$ , the shareholder of  $A(k)$  is a controller. Another vector is also generated i. e.  $V2(q) = V2(q) + A(k)$ ,  $k$  from  $h+1$ ,  $V2(0) = 0$ , if  $V2(q) > Difference$ ,  $V2(q) = V2(q) - A(k)$ . Suppose this test stops at  $A(z)$ ,  $h < z < n$ , so  $q = 0, 1, 2, \dots, z - h$ .
- 7) If  $Remainder\_Sum(k) + V1(p) < Difference$ ,  $p = 0, 1, \dots, h$ , if  $Remainder\_Sum(k) + V1(p) + V2(q) > Difference$  and  $V1(p) + V2(q) < Difference$ , then, the shareholder of  $A(k)$  is a controller. Suppose this test stops at  $A(z)$ ,  $h < z < n$ , so for  $V2(q)$ ,  $q = 0, 1, 2, \dots, z-h$ .  $Remainder\_Sum(k) + V1(p) + V2(q) < Difference$ , there is no more controller in this Supervised Entity. Stop. If  $z = n$ , it means all of the shareholders are controller, then stop the program.

### 3.2 The subroutine for determining the possible combinations (*sdmlc*)

The subroutine for determining the controllers at level 1 (*sdcll*) was just developed to find controllers from a combination of the shareholders. To find more controllers, almost all of the combinations should be considered. *Sdmlc* is designed to combine almost all possible shareholders to find the possible groups of the controllers.

Suppose  $I$  is an upper level loop to alternate the combinations of stock shares.  $J$  is a lower level loop to alternate the combinations of stock shares.  $I \times J$  are the total number of loops to generate the most combinations of shareholders  $A(k)$  to find possible controllers when using *sdcll*.

- 1) Rearrange the stock share percentages of the shareholders in decreasing order:  $A(1) > A(2) > \dots > A(k) > \dots > A(n)$ , where  $k = 1, 2, \dots, n$ .
- 2) Test the first shareholder. If his/her shares equal 50%, all shareholders of the SE are controllers and this finishes the process. If his/her shares are larger than 50%, the shareholder is the only controller of the SE and this finishes the process. In both cases,  $I$  equals 1 and  $J$  equals 1 too.
- 3) Define the value of  $I$ . To create a variable of  $Sum$  and add one by one of the stock share percentages from  $A(1)$  to  $Sum$  until it arrives 50%. If  $Sum$  equals 50%, all of the sharehold-

ers are controllers, and this finishes the process. If it is immediately minus 50%. All these shareholders are controllers,  $A(k)$ ,  $k = 1, 2, \dots, s$ .  $l = k - 1$ .

- 4) Create a vector *Remainder\_Sum* ( $k$ ) which equals the summary of all the rest of the stock share percentages from  $k = n-1, n-2, \dots$ , to 1.
- 5) For every  $i = 1, 2, \dots, l$  ( $l > 1$ ) to define  $J(i)$ . If  $i = 1$ ,  $j = 1$ , call directly the subroutine for determining controller in level 1 - *sdcl1*() to find controllers.
  - if  $i = 1$  then call *sdcl1*().
  - If  $i = 2$ , put the initial value  $j = 0$ , while  $(\text{Remainder\_Sum}((i-1) + (j+1)) - A((i-1) + (j+1))) \geq 50\%$ , do  $j = j + 1$ , temporarily eliminate  $A(i-1), \dots, A(j+1)$  to form a new group of stock shares and call *sdcl1*().
  - If  $i = 3$ , put the initial value  $j = 0$ , while  $(A(1) + \text{Remainder\_Sum}((i-1) + (j+1)) - A((i-1) + (j+1))) \geq 50\%$ , do  $j = j + 1$ , temporarily eliminate  $A(i-1), \dots, A(j+1)$  to form a new group of stock shares and call *sdcl1*().
  - If  $i > 3$ , put the initial value  $j = 0$ , while  $(A(1) + \dots + A((i-1)-1) + \text{Remainder\_Sum}((i-1) + (j+1)) - A((i-1) + (j+1))) \geq 50\%$ , do  $j = j + 1$ , temporarily eliminate  $A(i-1), \dots, A(j+1)$  to form a new group of stock shares and call *sdcl1*().

### 3.3 The subroutine for determining controllers at level 2 (*sdcl2*)

This subroutine was developed to extend the function of *sdcl1* if there are some APs in a SE. If  $n$  elements of the stock shares of the SE,  $A(k)$ ,  $k=1, \dots, n$ , are considered as at the level 1,  $m$  elements of an AP of the SE,  $Ap(k,i)$ ,  $i = 1, \dots, m$ , are at the level 2. In this paper, all  $Ap(.,i)$  are PFs, i.e. the algorithm covers up to level 2. The main objective of *sdcl2* is to analyze the elements at level 2 and combine them into level 1.

- 1) Read all elements of  $A(k)$  of a SE, where  $k = 1, \dots, n$ : if  $A(k)$  is an AP, then go to step 2; if there is no any AP, this means that all elements are PFs and there is no level 2, then do *sdmlc/sdcl1* for  $A(.,.)$ .
- 2) For every AP, create a matrix  $Ap(k,i)$ ,  $i = 1, \dots, m$ , where  $k$  is according to the  $k$  of  $A(k)$ ,  $m$  is the total number of PFs in  $A(k)$ . All of the elements of  $Ap(k,.)$  are at level 2. Suppose,  $K$  is the total number of APs in  $A(.,.)$ .
  - 2.1) For all  $k \in K$ , do *sdmlc/sdcl1* on  $Ap(k,i)$ ,  $i = 1, \dots, m$ :  
if for any  $Ap(k,i)$ , which is a only controller for this AP, then use this PF,  $Ap(k,i)$ , to substitute  $A(k)$ . Read  $A(k)$ ,  $k = 1, \dots, n$ , if there is another same PFs as  $Ap(k,i)$  in  $A(.,.)$ , sum it to  $A(k)$  and eliminate it from  $A(.,.)$ ;  
if there is no unique controller at  $Ap(k,.)$ , create a vector of  $Apc(k, j)$ ,  $j = 1, \dots, p$  for all  $k \in K$ , where  $p$  is the number of controllers of  $Ap(k,.)$ .
  - 2.2) Compare  $Apc(k,j)$ ,  $j = 1, \dots, p$ , with  $Apc(k', j)$ ,  $j = 1, \dots, p$ , for all  $k, k' \in K$  and  $k \neq k'$ ,  
if there are APs with same PFs, then sum the shares of  $Apc(k,.)$  and  $Apc(k',.)$ , and eliminate  $Apc(k',.)$ .
  - 2.3) After the possible elimination, a new vector of  $A'(.)$  is formed, where,  $k = 1, \dots, n'$  and  $n' < n$ .  
Do *sdmlc/sdcl1* on  $A'(.)$  and get a new vector  $A\_controller(k'')$ ,  $k'' = 1, \dots, n''$ ,  $n'' < n'$  and there are  $K''$  of APs in  $A\_controller(.,.)$ .
- 3) In  $A\_controller(.,.)$ , there are still some APs, but all of them are combined into the level1. The following steps are to used to substitute the APs by their PFs.
  - 3.1) For all  $k'' \in K''$ , using the elements of  $Apc(k'',.)$  to substitute  $A\_controller(k'')$ , where  $Apc(k'',.)$  is all PFs of AP of  $k''$ .
  - 3.2) In the  $A\_controller(.,.)$ , all elements are PFs. There are still some repeated PFs. For  $k'' = 1, \dots, n''$ , eliminate the repeated  $A\_controller(k'')$ . Put the new element in vector  $A\_controller\_final(.,.)$ .

## 4 Complexity Analysis

To determine the controllers from their stock shares is similar to the combinational problem *Sum of Subset*. A backtracking algorithm was developed with a possible search space which includes all of the possible solutions [2, 3]. Unfortunately, this algorithm, in the worst case, has the complexity  $O(p(n) 2^n)$  or  $O(p(n) n!)$ , where  $p(n)$  is a  $n^{\text{th}}$  order polynomial [2, 3]. Therefore, this algorithm is only of limited value to the practical problems of BCB.

The developed subroutines (*sdcl1*, *sdcl2* and *sdmlc*) take advantage of the features of the actual problem instance. For decreasing the number of combinations of stock shareholders, some criteria are established to reduce the dimensions of the search space significantly compared to the backtracking algorithm. For example, in *sdcl1*, if  $\text{Remainder\_Sum}(k) + V1(p) + V2(q) < \text{Difference}$ , then  $A(k)$  is not a controller and there are no more controllers in this group of stock shares. In *sdmlc*, if  $A(1) + \dots + A((i-1)-1) + \text{Remainder\_Sum}(((i-1) + (j+1)) - A((i-1) + (j+1))) < 50\%$  then, there are no more potential controllers.

- *Complexity analysis of sdcl1*

The complexity analysis of the main steps of *sdcl1* is illustrated in Table 1.

**Table 1.** Complexity analysis for *sdcl1*

Step	Method	Complexity	Condition
1	merge sort	$O(n \log n)$	
3	while iterative	$O(s)$	$1 < s < d$
4	while iterative	$O(d-s)$ ,	$s < d < h$
5	for iterative while iterative	$O(n)$ $O(h-d)$ ,	$d < h < z$
6-8	while and 2 for iterative	$O(p \times q \times (z-h))$	$h < z < n, 1 < p < (h-d), 1 < q < (z-h)$

From the step 6 to 8,  $O(p \times q \times (z-h)) < O(h \times z \times (z-h)) < O(z^3 - h^3) < O((z-h)^3)$ . After simplifying the results in Table 1, the complexity of *sdcl1* is  $\max(O(n \log n), O((z-h)^3))$ , where  $1 < h < z < n$ .

- *Complexity analysis of sdmlc*

The complexity analysis of the main steps of *sdmlc* is shown in Table 2.

From step 5, the complexity of *sdmlc* is  $\max(O(I \times J \times n \log n), O(I \times J \times (z-h)^3))$ , where  $1 < I < d < h < z < n$ , and  $1 < J < n$ .  $I$  and  $J$  are usually given parameters. Due to  $\max(O(I \times J \times n \log n), O(I \times J \times (z-h)^3)) < O(n^3)$ , the complexity of *sdmlc* is  $O(n^3)$ .

- *Complexity analysis of sdcl2*

The main operations of *sdcl2* are to call *sdcl1* and *sdmlc* for some APs of a SE. Other operations may involve the necessary comparison, etc. Basically, the complexity of *sdcl2* is similarly as *sdmlc*.

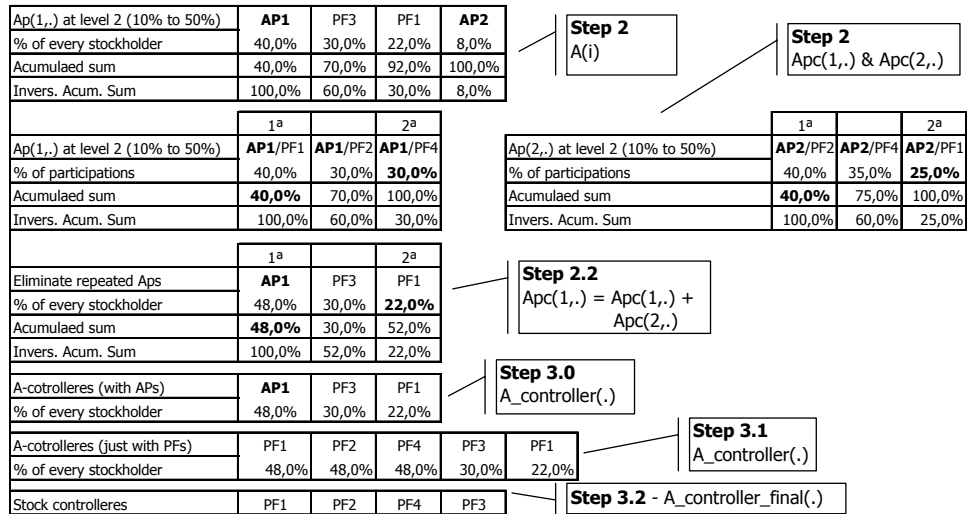


**Table 2.** Complexity analysis of *sdmlc*

Step	Method	Complexity	Condition
1	merge sort	$O(n \log n)$	
3	while iterative	$O(I)$	$1 < I < h$
4	for iterative	$O(n)$	
5	for iterative while iterative call <i>sdcll()</i>	$\max(O(I \times J \times (n \log n)), O(I \times J \times (z-h)^3))$	$1 < I < h$ $J = \max(J(i))$ $h < z < n$

## 5 An Example

The following example of the application of *sdcll* and *sdmlc* is taken from [8]. In this section, *sdcl2* was used to find the controllers for a SE, BM Bank, as in Fig. 1. In this example, there are two APs (BM Automovies and BM Insurance) and other two PFs that hold the stocks of BM Bank. To find the stock controllers of that bank, the process of applying *sdcl2* is described in Figure 3. The final stock controllers of BM Bank are PF1, PF2, PF3 and PF4. This example is just aimed at showing the problems of real instances of our problem.



**Fig. 3** The illustration of the application of algorithm: *sdcl2*

## 6 Conclusions

An algorithm for determining the stock controllers of some financial institutions for the Brazilian Central Bank (BCB) was developed and implemented. Initial analysis shows that the proposed algorithm has polynomial complexity  $O(n^3)$ . It can be used to find almost all possible controllers of a SE, though how near to a complete solution it comes has to be further evaluated. The paper described the three subroutines dealing with the first two levels, but in the real situation, to determine the controllers through stock shares for a SE, sometimes, may require up to 10 levels. The algorithm for the first two levels can be basically adapted for higher levels. But BCB asked us to just report the first two levels in this paper.

The developed algorithm was implemented by Politec Informatics Ltda, Brazil and is currently employed at Unica/BCB. *Natural* is a basic language which was used to implement. All the codes are adapted in the computing operation system of Unica (National Unique Registration) of Brazil and integrated with the database of BCB. The experimental results indicate the feasibility of automating the process of finding controllers using our approximation algorithm instead of the manually operation. What took two agents a full day of work has now been reduced to just minutes of computation in BCB's computer system. Though developed for BCB, it is expected to work equally well for other financial institutions which are controlled by stockholders. The future work will involve to combine methods of Artificial Intelligent and mathematics models to improve the efficiency and reduce human involvement in the process even more.

## References

1. Banco Central do Brasil: Regras para se calcular o controlador de uma entidade supervisionada pelo Banco Central do Brasil, (2001).
2. Cormen, T. H., Leiserson, C. E. and Rivest, R. L.: Introduction to Algorithms, The MIT Press, Combridge, USA, (1996).
3. Horowitz, E., Sahni, S. and Rajasekaran, S.: Computer Algorithms, Computer Science Press, New York, USA, (1998).
4. Garey, M. R., and Johnson, D. S.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company (1979).
5. Brickell, E. F.: Solving low density knapsacks, *Advances in Cryptology, Proceedings of Crypto '83*, Plenum Press, New York (1984), 25-37.
6. Lagarias, J. C., and Odlyzko, A. M.: Solving low-density subset sum problems, *J. Assoc. Comp. Mach.* 32(1) (1985), 229-246.
7. Radziszowski, S. and Kreher, D.: Solving subset sum problems with the  $L^3$  algorithm, *J. Combin. Math. Combin. Comput.* 3 (1988), 49-63.
8. Branco, V. G. F., Weigang, L., Ribeiro, A. F., Shibata, W., Abad, M. P. E., Torres, N. V., Dib, M. V. P. and de Andrade, V. M.: An Algorithm for Determining the Controllers of Supervised Entities at the First Level: Brazilian Central Bank Case Study, Proc. of FSKD'02 - 1st International Conference on Fuzzy Systems and Knowledge Discovery, pp. 240-244, Singapore, (2002).