THE ALGORITHMS OF STRATEGIC FINANCIAL MANAGEMENT

Manana Chumburidze, Nana Shonia*

Akaki Tsereteli State University, Kutaisi
Georgia

* Corresponding author, e-mail: manana.chumburidze@atsu.edu.ge

Abstract: This article discusses the task of strategic financial management of investments portfolio. The information model of investment contributions in tree-diagrams is constrected. Efficient algorithms for managing investment financial transactions are built. the tools aplyd in this development based on the generalizing breadth-first search algorithms and forward approch of dynamical programing method. Relevant pseudocodes have been developed. For software implementation priority queuing aproches with a stacks owerflows are duscessed.

Key words: decision business tree, priority queue.

1. INTRODUCTION

The Financial Planning and Analysis (FPA) role is gaining greater importance today, as it helps bring out strategic analysis on business performance. Financial planning encompasses a wide variety of different financial matters including debt management, budgeting, investing, saving and retirement planning Investors divide their money between investment tools rather than placing all their resources in one investment tool. Placing money in different investment tools increases the chances of getting benefits and minimizing the negative results of each. Spreading investments across industries and geographic market sectors can reduce the overall impact to a portfolio, especially when surprising events radically change a certain industry or area[1].

The purpose of the work is to create a decision making algorithm for FPA and demonstration of opportunity to explore plan and predict several possible outcomes of financials decisions and regardless to strategy of management.

Financial planning apply to programs that require complex analysis and computations and have numerous operations [2]. FPA traditionally use interconnected offline spreadsheets with Microsoft Excel. The **C++** programming language has a rich tools to accomplish any task and developing complex projects. Its efficiency is due to the tight data structures and programming. In this work

financial modeling by using the **C++** is involved. This approach differs from traditional Excel-based models and provide detailed analysis of financial performance and decision guidance to the business, executives and others.

In this paper the algorithms of Financial Analysis by managing various reporting processes is developed. Investment Management problem to responsibled of financial planning and modeling to guidance on investment and project decisions is solved. The algorithm consider the credit score of each business and how to get started with financial planning [3].

We provide an overview of priority queueing approaches to emprowe incomes forecasting around returns at the fund and accounts level to surface simple fund visuals to address business sectors.

In this paper advanced BFS algorithms [4]techniques to describe alternative plans of investment and dynamical programing methods tools of forward approaches to assess step by step incomes of portfolio are discussed. The correlations to optimize business planning process according to some local criterion is discovered. the priority queueing approaches to explore of neighbors business of discovered business sector in the given stage of algorithm is implemented.This article describes several of the most common storage models.

Interactive tree-plot to modeling hierarchical related to one another of business sectors is designed. The branches of tree are presenting of enquiry finances. The internal nodes used for store relevant outcomes of investments in the given stage of algorithm .The leaves are contented of sums of financial plans outcomes of corresponding branches of tree. They are viewed by using data structure stack as a contents stack of strategic investment portfolio.

This algorithm produces a so-called strategic investments tree[5].

2. NOTATIONS AND DEFINITIVE CONSEPTS

In this section effective solution of decision making problem by examining alternative choices and deciding on the best route to take is required [6].

The algorithms to describe investment management action into several sequential stages of business sectors are constructed.

For investigation the graphical-model of decision business tree is considered. The root represents such as a top-level node of ultimate objective investment portfolio. Branches represent different options of investment action that are available when making a particular decision. The nodes represent possible outcomes for each action to storing the sum of incomes in the planning stage. The leaf nodes predict several possible outcomes of investment decisions.

Using a step-by-step approach of dynamical programing method [7] is an efficient way to make strategic plan of investment management to track income from alternative business projects that have a positive impact for organization’s short- and long-term goals.

Let us introduce the following notations: n-number of business sectors; $\mathbf{G=(g_{ij})_{nxn}}$ Boolean matrix to represent the edges (adjacent business) of tree whose vertices are numbered 1 to **n**. $\mathbf{Income=(\ income)_{nxn}}$ matrix to save enquiry invests for currently business sector $\mathbf{Win=(win_{ij})_{nxn}}$ matrix to keep sum of income invests from beginning to current stage (see Figure 1). The **parent**[i] to access the nodes in a path of tree. An forward approach of dynamical programing method to build of a business tree for this problem would be described as a business tree-diagram, where memorizing is started from the root to the leaves of tree, there leaves will give us the maximum sum of the node values from root to any of the leaves. The elements of matrices are defined by the following formula [7]:

$$win[j][i] = income[j][i] + win[parent[i]][j] \quad (1)$$

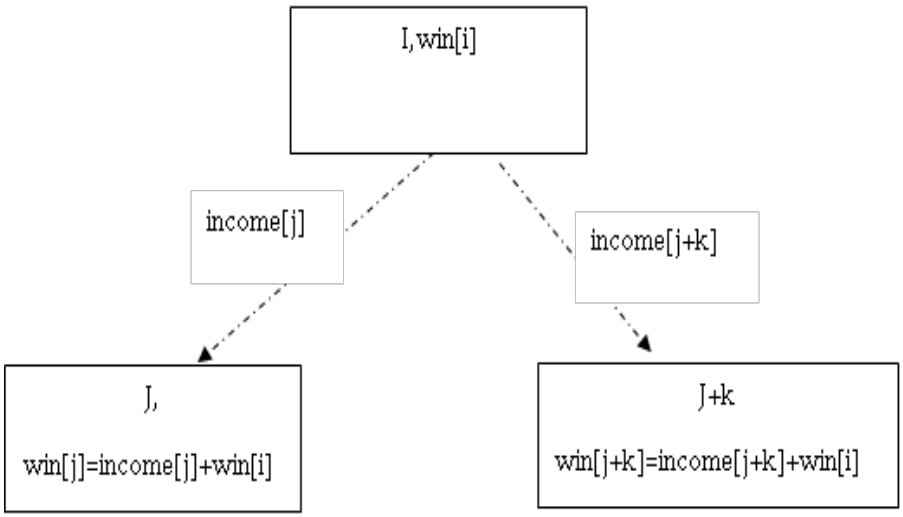

Figure 1. Subtree of decision business tree

3. ALGORITHM SOLUTION

In this section problem-solving algorithm is constructed. The software design described in pseudocodes[8]. For Software implementation the following data structures are created:

- The audiences nod matrix_for represent neighboring businesses
- The enquiry invests matrix
- The queue to save of discovered neighboring businesses and their exploring
- The matrix to save a incomes invest
- The predecessors nodes array to save path of invest transaction
- The stack overflows of incomes results of invest transaction

Let us consider of trees describe pseudo code:

```
int a,x,y,mon;
  cin>>a;
   while(a--){
      cin>>x>>y>>mon;
      graph[x][y]=1; // represents a adjences business matrix
      income[x][y]=mon; // represents of enquiry invests matrix
   }
```

The list for Breadth First Search adjacent business to save current incomes is constructed:

```
//Unchanged data packing
   for(int i=1;i<=20;i++){
   if(graph[1][i]!=0){
  List.push_back(i); // Mark the current node as visited and insert it in list
 win[1][i]=income[1][i]; // Create a matrix of current incomes
                }
        }
```

Let us introduce a variable **pred** to remember of ancestral business number for each sectors and to check each time whether it is a leaf (terminal) of tree , i.e. with a business that no longer has neighboring.

These operations must continue until queue is empty, if the list is empty it means we no longer have neighboring businesses, which means we have already reviewed all the businesses:

```
int pred=1,maxwin=0,cnt=2;
        //As long as there is a neighboring businesses
while(!Empty()){//
       int counter=0;
       if(!ifempty(pred)){
          pred++;
       }
       parents[List[0]]=pred;
```

where

```
bool Empty(){
   if(List.size()==0)return true;
   else return false;
```

```
}// To check if an List has elements used
bool ifempty(int pred){
   int g=0;
   for(int i=1;i<=20;i++){
      if(graph[pred][i]==1)return true;
   }
   return false;
}// To check if an node is a leaf of tree
```

The **parent** attribute of each node is useful for accessing the nodes in path of investment, for example by backtracking from the destination node up to the starting node, once the predecessor's nodes had been seated.

The pseudo code to count of investment way:

```
       int cnt1=cnt;
        while(parents[cnt1]!=0){
            if(counter>=way.size())
        way.push_back(cnt1);
            else way[counter]=cnt1;
            cnt1=parents[cnt1];
            counter++;
        }
        if(counter>=way.size())
         way.push_back(1);
        else way[counter]=1;
        if(!check())  // checks the end of a tree branch
{
      ans.push(win[pred][List[0]]);  // Create a stack-of  investment
            if(ans.top()>=maxwin){
                    maxwin=ans.top();
                    cnt=List[0];
            }
        }
```

where

```
       bool check(){
    int g=0;
    for(int i=1;i<=20;i++){
        if(graph[List[0]][i]==1)return true;
    }
    return false;
}// declaring a Boolean variable with true value of existence
adjacencies business
```

The pseudo code to filling of the queue and recording of win- (two-dimensional array) get to the form:

```
      for(int i=1;i<=20;i++){
           if(graph[List[0]][i]!=0){
            List.push_back(i);//create  list of neighboring businesses
        win[List[0]][i]=income[List[0]][i]+win[pred][List[0]];//create
                                                              incomes array
               if(win[List[0]][i]>maxwin){
                   maxwin=win[List[0]][i];
                   cnt=List[0];
               }
           }
       }
       for(int i=1;i<=20;i++){
           if(graph[pred][i]==1){
               graph[pred][i]=0;
               break;
           }
       }
```

In the next step the discussed item are removed from the list and assigned in -0. Function to remove item from list:

```
     del(){
    for(int i=0;i<List.size()-1;i++){
        List[i]=List[i+1];
    }
    List.pop_back();
}
```

In finally stage, each branch of winnings are printed and appeared clearly:

```
    int ansmax=0;
    while(ans.size()!=0){
        if(ans.top()>ansmax)
   ansmax=ans.top();
        ans.pop();
    }// Get all   deleted adjacent vertices
    cout<<"The best profit = "<<ansmax;
    cout<<endl;
    cout<<"The best way to win = ";
     for(int i=0;i<way.size();i++){
        cout<<way[i]<<" ";
     }
```

For simplest case, we have considered the particular example a business tree (see Figure 2)

Let us consider the scenarios of particular tree (see fig 2.) The following results of data structures evaluated investments' scenarios for the particular case of business tree (Figure 2) is delivered

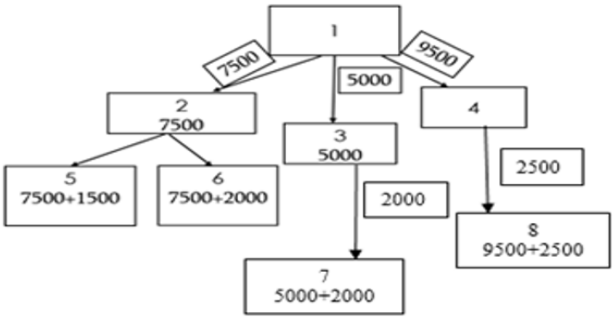

Figure 2. Decision business tree

Table 1. Vector of predecessor's business

1	***2***	***3***	***4***	***5***	***6***	**7**	**8**
0	*1*	*1*	*1*	*2*	*2*	*3*	*4*

Table 2. Matrix of incomes

	1	***2***	***3***	***4***	***5***	***6***	*7*	*8*
1		*7500*	*5000*	*9500*				
2					*9000*	*9500*	*7000*	*12000*

There are results of data structures evaluated of all investments' scenarios for business tree (see Figure 2):

Table 3. List of results to choose a suitable way

1	***2***	***3***	***4***
9000	*9500*	*7000*	*12000*

Table 4. List of Parent

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
0	*1*	*1*	*1*	*2*	*2*	*3*	*4*

This algorithm offer how to do track business income and expenses.

The development propose of efficiently approach to the decision making process. It help to collect information, probably identify several possible paths of action, or alternatives, to choose among alternatives the best or to consideration the combination of alternatives.

4. CONCLUSION

Thus in this paper strategic financial management problem is investigated. Decision making problem by mapping out different courses of business action with a potential outcomes has been decided. The algorithm to observe of different plan of investment action is constructed. The tools to accomplish tasks in this development flexibly reduces the complexity of algorithms and help teams streamline, automate, and increase effectiveness of their strategic efforts. This implementation has a many advantages:

- Easy to understand and implement

- Identifies deliverables and milestones
- Forecasting and analyzing which plan of investment action has the success and safeguard of manager's decisions undesirable outcomes.
- Efficient in terms of time complexity

REFERENCES

[1] Valenčík, R., Červenka, J. Analysis tools of connecting investment opportunities and investment means in the area of small and medium-sized. *Enterprises European Research Studies,* Vol. XIX, No. 4, 2016 pp. 130- 139.

[2] Gritsyuk, M.K., Gritsyuk I.V. Technologies and tools used for prediction of characteristics of activity of industrial enterprises. *International Journal on Information Technologies and Security*, Vol. 12, No. 4, 2020, pp. 47-62.

[3] Skiena, St. Sorting and searching. *The Algorithm Design Manual.* Springer, 2008, p. 480. doi*:*10.1007/978-1-84800-070-4_4. ISBN 978-1-84800-069-8.

[4] Cheryshov, O., Choporov, N., Preobrazhenskiy, A.P., Kravets, O.Ja. The development of optimization model and algorithm for support of resources management in organizational system**.** *International Journal on Information Technologies and Security*, vol. 12, No. 2, 2020, pp. 25-36.

[5] Chumburidze, Manana, Mzia Kiknadze, Nino Topuria, and Elza Bitsadze. "About the Algorithms of Strategic Management." In *International Conference on Applied Mathematics, Modeling and Computational Science*, pp. 647-654. Springer, Cham, 2019, DOI:https://doi.org/10.1007/978-3-030-63591-6_59.

[6] Chumburidze, M. et al. Delivery management algorithms and their applications. *International Journal of Advanced Research*, Vol. 9, No. 5, 2021, pp. 596-601.

[7] Chumburidze, M., Beradze, T. Cash flows management algorithm and their applications. *International Journal on Information Technologies & Security*, Vol. 12, No. 2, 2020, pp. 15-24.

[8] Bennett, N. Introduction to algorithms and pseudo code. Working paper in Project *Exploring Modelling and Computation*, August 2015 (19 p.), DOI: 10.13140/RG.2.2.28657.28008

Information about the authors:

Manana Chumburidze – PhD of Mathematics Sciences. Professor at the Akaki Tsereteli State University of Department Computer Technology. Area of scientific research: Computer Sciences. Membership of ASME (The American Society of Mechanical Engineering). Member of the Scientific and Technical Committee of Editorial Review Board on computational sciences in the WASET organization of USA.

Nana Shonia – Professor at Business Administration Department; Dean of Faculty of Business, Law and Social Sciences at Akaki Tsereteli State University Area of scientific research: Business, Information Technology**.**

Manuscript received on 4 November 2021