Introducing a new enhanced method in peripheral maintenance simulation and a software based approach to solve the model

H.Attaran [1], M.Teymoortashloo [2] ,A.Barazandeh Shirvan [1] ,H.Orooji [1]

[1]Khavaran Higher Education Institute, Mashhad, Iran; [2]

Presenter: Houbakht Attaran

Abstract

Having adequate information and long term scheduling, will result in less cost and maximum benefit in industrial systems. A known method for this purpose is dynamic programming in peripheral maintenance simulation term. This simulation method has been less considered due to its low speed of execution and high order of iterations. This paper at first will focus on existing models and their execution methods. Then a new enhanced model is introduced with minimum execution order. Same sample data is used to test the models and the proof is given to show that the newly introduced model contains a higher capability than existing models.

Keywords: Simulation, Peripheral maintenance, Dynamic programming, Cost optimization

Introduction

Dynamic programming is an efficient method for querying and optimization problems in programming and mathematical science [1][3]. In spite of linear programming, there is no standard format to formulize dynamic programming problems and in each case, special formulas must be represented that fully meet the needs. Dynamic modelling methods including recursive technique, omitting never used values, dynamic programming and finally the creative combinational method, are previously taken into consideration by writers and a sample model for peripheral maintenance problem was solved in all methods and the most efficient method was introduced[5]. Due to the efficiency of this new creative method, the presented model is enhanced in this article and implemented using real world data.

Further, in section 2, existing models and obtained results will be discussed briefly; section 3 presents the new enhanced model and implementation algorithm and execution results, and finally, section 4 contains conclusion and some suggestions for further studies.

2- Existing models and tested methods

Industrial systems use several types of machines. All these machines need to be revised in a period of time which is called a simulation year here *[1][3]*. This revision can be one of the following choices:

Changing

Overhauling

Keeping

Islamic Azad University, Bojnoord, Iran

The goal is finding a sequence that minimizes the costs in a definite time which is called system life time.

The primary model is described as follow in which p shows the price of a brand new machine, k shows the age of machine, t(k) shows the selling price of a k years old machine only if you buy a new one, c(k) shows the total cost of keeping a k years old machine for whole simulation year, n shows the year, g(k) shows the selling price of a k years old machine and N shows the end of system life time.*[1][3]*

$$s(n,k)\begin{cases} p - t(k) + C(0) + s(n+1,1) \\ C(k,m) + S(n+1,k+1,m+1) \end{cases}$$

$$s(N,k)=-g(k)$$

$$S(0,0)$$

Formula 1: The preliminary model

This very basic model was improved by adding the capability of overhauling the machine[1][3] and letting the operator to buy a second hand machine which can be y years old[5]. The improved model is shown as follow in which m shows how many years before the machine was overhauled and this parameter is applied to other prices. Note that mp shows the same subject for newly changed machine[5].

$$S(n,k,m) = min\begin{cases} p(y,mp) - t(k,m) + c(y,mp) + S(n+1,y+1,mp + \\ o(k,m) + c(k,0) + S(n+1,k+1,1) \\ c(k,m) + S(n+1,k+1,m+1) \end{cases}$$

$$S(N,k,m)=-g(k,m)$$

$$S(0,y,m)$$

Formula 2: The improved model

Pseudo code 1 shows the implementation by recursion [2][4].

```
Function  S(int n,int k ,int m)
{
If (n==N) return –g[k,m]
Else
For each y and mp
Calculate R
Calculate minimum in R's
Calculate O,K
Calculate minimum between R,O,K;
Return minimum
}
```

Pseudo Code1: Recursive

The operational and placement orders are too high in this method and function calls can be too much. Note that all models are tested using an AMD 64 X2 Dual computer with 2 MB RAM. Results are shown in table 1. Figure 1 shows the function calls in this method[2][4].

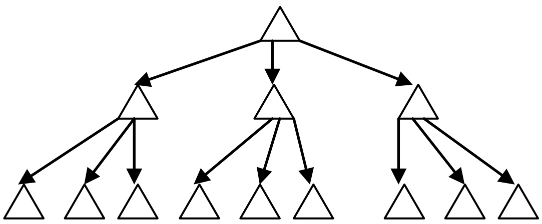

Fig 1: Function calls schematic having N=3

Each node of this tree is actually a pyramid. Based on different types of machine, the cell quantity is increased. Figure 2 shows this fact.

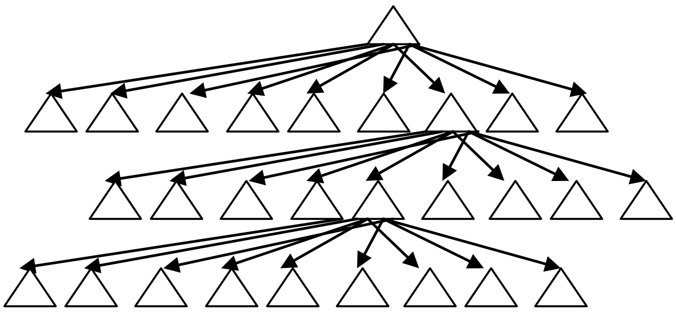

Fig 2: Pyramid scematic of each node

If we consider that b is available number of machines and system life time is N, total nodes on Nth level would be b^N (Root level is 0) and the amount of total nodes can be calculated by formula 3 [2][4].

$$b0 + b1 + b2\ b3 + \ldots. + b^N = b^{N+1} - 1 \approx O(L \quad ($$

In omitting never used values method, all values of should not be considered. Only those values which are based on the condition y>=mp are valid that can be seen colored in figure 3. The main diameter of matrix shows those machines which are never overhauled.

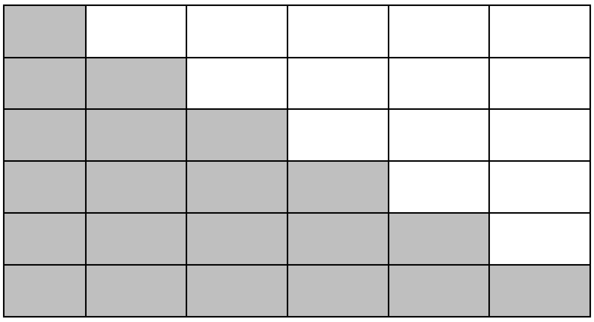

Fig 3: Valid values shown colored

This fact was applied to model and a new program was implemented using the same data. Pseudo Code 2 shows the implementation and results are shown in table 1.

```
Function  S(int n,int k ,int m)
{
If (n==N) return –g[k,m]
Else
For each y and mp when mp<=y
Calculate R
Calculate minimum in R's
Calculate O,K
Calculate minimum between R,O,K;
Return minimum
}
```

Pseudo Code2: Implementation by omitting never used values

The main algorithm has been changed to fit this method. Changes can be seen in Pseudo Code 3 and results are shown in table 1.

```
Function  S(int n,int k ,int m)
{
If (n==N) return –g[k,m]
Else
If s(n,k,m) was calculated
Return s(n,k,m)
else
For each y and mp
Calculate R
Calculate minimum in R's
Calculate O,K
Calculate minimum between R,O,K;
Return minimum
}
```

Pseudo Code3: Dynamic programming

Combinational model uses a combination of both omitting never used values model and dynamic programming model. Pseudo Code 4 shows the implementation method of this model.

```
Function  S(int n,int k ,int m)
{
If (n==N) return –g[k,m]
Else
If s(n,k,m) was calculated
Return s(n,k,m)
else
For each y and mp when mp<=y
Calculate R
Calculate minimum in R's
Calculate O,K
Calculate minimum between R,O,K;
Return minimum
}
```

Pseudo Code4: Combinational model

All mentioned models were tested with the same data set. Tables 1 shows the number of function calls in different methods.

Total years	*Combina model*	*Dynamic Program*	*Omitting r* Used values	*Recursi*
2	**11**	**17**	**13**	*28*
3	**36**	**60**	**157**	*757*
4	**62**	**101**	**1885**	*20440*
5	**96**	**153**	**22621**	*551881*
6	**139**	**217**	**271453**	*149007*
7	**192**	**294**	**3257437**	*402321*
8	**256**	**385**	**39089245**	∞
9	**332**	**491**	**469070941**	∞

Table 1: Comparison on number of function calls.

Table 2 shows the execution time taken by each method in milliseconds.

Ti ye	*Combinati model*	*Dynamic Program*	*Omitting* Used values	*Recur*
2	*Less then*	**2**	**7**	*16*
3	*Less then*	**3**	**16**	*21*
4	*Less then*	**5**	**18**	*46*
5	*Less then*	**14**	**41**	*291*
6	**1**	**15**	**515**	*4234*
7	**1.2**	**16**	**6353**	*10001*
8	**1.4**	**17**	**62063**	∞
9	**1.7**	**19**	**928344**	∞

Table 2: Comparison on execution time in milliseconds.

These tables prove the efficiency of new creative combinational method [5]. This method can be used for new suggested model represented in section 3.

3- Suggested model plus implementation

As it can be seen in existing models, the number of overhauls applied to machines is not considered. This factor can affect all prices according to type of machine. Let f show the total overhaul times applied to a machine. This parameter should be applied to all prices including new machine price, overhaul, keeping costs and also selling prices. Formula 4 shows the new model.

$$S(n,k,m,f) = min\begin{cases} p(y,mp,fp) - t(k,m,f) + c(y,mp,fp) \\ \quad +S(n+1,y+1,mp+1,fp) \\ o(k,m,f) + c(k,0,f+1) + \\ \quad S(n+1,k+1,1,f+1) \\ c(k,m,f) + S(n+1,k+1,m+1,f) \end{cases} \qquad (4)$$

$S(N,k,m,f)=-g(k,m,f)$

$S(0,y,m,f)$

Formula 4: The new enhanced suggested model

In this model, p(y,mp,fp) shows the price of a y years old machine that m years earlier was overhauled for fp times, k shows the age of machine, t(k,m,f) shows the selling price of a k years old machine that m years earlier overhauled f times only if you buy a new one, c(k,m,f) shows the total cost of keeping a k years old machine for whole simulation year, n shows the year, g(k,m,f) shows the selling price of a k years old machine and N shows the end of system life time. Pseudo Code 5 shows the implementation of this model.

```
Function  S(int n,int k ,int m, int f)
{
If (n==N) return –g[k,m,f]
Else
If s(n,k,m,f) was calculated
Return s(n,k,m,f)
else
For each y and mp and fp when
mp<=y
Calculate R
Calculate minimum in R's
Calculate O,K
Calculate minimum between R,O,K;
Return minimum
}
```

Pseudo Code4: New model implementation using combinational method

Table 3 shows the number of function calls and execution times in milliseconds of former implementation.

Total y	*Function*	*Execution*
2	*17*	**11**
3	*53*	**12**
4	*120*	**13**
5	*225*	**16**
6	*381*	**21**
7	*603*	**36**
8	*907*	**40**
9	*1311*	**49**

Table 3: Results of implementation of new model.

4- Conclusion

The traditional model and a new enhanced model was described and introduced during this paper. Due to the shortages of traditional model, a new enhanced model has been suggested by writers and implemented by combinational method which is again introduced earlier by writers [5]. This model and the presented method seem to be efficient and can be upgraded again by adding quality values using fuzzy logic in future.

5- References

David J Leonberger, Dr Mahdavi Amiri(2000), "Dynamic Programming", Sharif University press .

Richard A Neopolitan , Kioomars Naeemipoor(2006), "Algorithm design using C++ Pseudo Codes", Oloom rayaneh press.

Dr Mohammad javad Asgharpoor (1999), "Dynamic Programming", Tehran university Press.

Dr Mahmood Naghibzadeh (2007), "Algorithm design", Behnashr Press.

H. Attaran, M. Teimoortashloo, A. Barazandeh, H. Orooji,(2010) "A Comparison On Peripheral Maintenance Implementation Methods", ITPF Conference.