Engineering, 2012, 4, 235-244
http://dx.doi.org/10.4236/eng.2012.45031 Published Online May 2012 (http://www.SciRP.org/journal/eng)

Design of Software for Maintenance Workforce Scheduling (A Case Study of Afam Power Station, Nigeria)

Ikuobase Emovon[1], Macaulay Thomas Lilly[2], Stephen Ogajiye Tamuno-Ojuemi Ogaji[3]
[1]Mechanical Engineering Department, Federal University of Petroleum Resources, Effurun, Nigeria
[2]Mechanical Engineering Department, Rives State University of Science and Technology, Port Harcourt, Nigeria
[3]School of Engineering, Cranfied University, Bedfordshire, United Kingdom
Email: ikuoy2k@yahoo.com

Received September 10, 2011; revised December 1, 2011; accepted December 8, 2011

ABSTRACT

This paper is concerned with the optimization of the workforce scheduling for solving maintenance problems. To achieve this aim an optimsation software for (5,7) problem was developed. The programme was written in Quick Basic. The software was designed to produce a seven day schedule for organization operating a seven day week. Hence organization operating a five day schedule wishing to change to a seven day schedule we find this software very useful. The Quick-Basic computer programme was based on Alfares [1,2] algorithm for solving (5,7) schedule problem. Data collected from Afam power station, Nigeria was used as input data. The test result shows the software is capable of determining workforce size and assigning workers to day-off pattern. The seven-day schedule produced savings of 11% maintenance labour cost when compared with the 5-day schedule currently being practiced by the Power station [3].

Keywords: Optimization; Workforce Scheduling; Software; Quickbasic

1. Introduction

For the past ten years, the federal government of Nigeria has embarked on aggressive rural electrification projects across the country. This laudable project is aimed at providing electricity to the rural dwellers where about 64% of the entire population lives. With this laudable project the federal government hope to minimize, the rural-urban migration thereby forcing young school leavers and graduates to remain in the village self-employed rather than moving to the urban area seeking white-collar jobs [4].

However the pace of this project has drastically slowed down of recent for obvious reasons; 1) Economic meltdown; 2) Drastic decline in the power generation in Nigeria, that is from 5924.7 MW which is the total installed capacity of our generating unit to about 2400 MW.

The available mega watt of electricity generated cannot even carter for the urban centers talk less of extending it to rural areas. The negative implication of this power problem in Nigeria has gone beyond rural-urban migration. Most youth that would have been self employed in Nigeria now go into prostitution and drug trafficking, in Asian, Europe and America.

However the major reason why power stations in Nigeria are generating far below installed capacity is due to maintenance problem. This has affected the availability and reliability of the power plants. In order to reduce these maintenance problems, there is need for effective maintenance scheduling. Scheduling is a crucial component of maintenance management. Maintenance requires three resources: Manpower, materials and equipment. Labour is usually the most important and expensive resource of any organization. Effective labour scheduling is a major determinant of the productivity of the workforce. Improved labour productivity results in efficient handling of maintenance activities. This will invariably result to improved availability and reliability of power plant. This will boost power generation in Nigeria.

Labour scheduling problems are classified into three types 1) shift, or time-of-day, scheduling, 2) days-off, or day-of-week, scheduling, and 3) tour scheduling, which combines the first two types [5]. For organization operating 7 days a week, such as Afam power station, Nigeria, employee days-off scheduling is a significant and relevant problem. The most usual type of days-off schedules is referred to as the (5,7) problem in which each work pattern includes 5 workdays and 2 consecutive off-days per week.

This paper is concerned with development of an Optimisation software for the (5,7) problem using Alfares [1,2] algorithm. The software is capable of determining workforce size and assigning workers to days-off patterns. This software is capable of solving the maintenance manpower scheduleing problem in Afam power station and other Power Station in Nigeria and invariably improving Power generation. However, it is not only applicable to power station in Nigeria, but applicable to other organization operating 7 day a week like restaurants, Hospitals and police stations.

The beauty of this software is that it does not require specialize training unlike integer programming software.

This paper is organized as follows: First a review of problem and background of Afam power station. Then Alfares algorithm for solving (5,7) problem is presented. This is followed by computer programme written in quick basic for solving (5,7) problem based on Alfares algorithm. Subsequently result of computer programme as applied to Afam power station maintenance labour scheduling problem. Finally conclusions.

2. Problem and Background

From the data collected by Emovon, [6] from Afam Power Station Nigeria it's on record that the first major gas turbine station built in Nigeria is the Afam Gas Turbine Power Station. The power station is located in the Niger Delta because of the large reserve of natural gas in the region. The first phase of the power station was commissioned in 1963 as Afam I, consisting of four generating units with total capacity of 54 MW. Consumption growth experienced in the seventies necessitated the construction of Afam II, which was commissioned in 1976. Afam II consists of four additional generating units with installed capacity totaling 92 MW. This was followed by Afam III, which was commissioned in 1978, consisting of four units with installed capacity totalling 108 MW. In 1982, six generating units were commissioned as Afam IV with total capacity of 450 MW. Finally, Afam V was commissioned in 2002, made up of two units with a total capacity of 138 MW. This brought the number of units in Afam power station to 20, with a total installed capacity of 980 MW. With only 413 MW currently being produced, the power generated is less than 43% of the installed capacity.

The station has been generating power far below installed capacity due to maintenance problems. These problems have affected the availability and reliability of the power plant. In order to minimize these maintenance problems, there is a need for effective maintenance scheduling, which is a major determinant of the productivity of the workforce. Improved labour productivity leads to more efficient handling of maintenance activities. Hence the purpose of this paper is the optimization of the workforce scheduling for solving maintenance problems. To achieve this aim, an optimization software for (5,7) problem was developed. The program is written in QuickBasic.

Afam power station maintenance line comprises of the following departments: Mechanical, Electrical, Instrumentation & Control, Workshop Services, and Planning. The maintenance management functions in Afam Power Station include both reactive (run-to-failure) and preventive maintenance (PM). The PM type in place is time based, which is not effectively carried out, not to talk of practicing the state-of-the-art, predictive maintenance.

The preventive maintenance procedure and intervals are not well defined. Poor plant history records make it difficult to retrieve PM history and reports of plant/ equipment, especially for old plants. The power plant maintenance in Afam power station is performed by combined effort of the various departments that make up the maintenance line. The maintenance crews of the various departments carry out daily plant maintenance checks. The checks involve taking plant parameters like load, compressor inlet temperature, and plant evaluated operating hours to mention but a few. Apart from daily plant checks, maintenance line crews carry out 3 types of inspections:

1) First minor inspection, also referred to as small or type A inspection;

2) Second minor inspection, also referred to as normal or type B inspection;

3) Major inspection, or type C inspection.

The first minor inspection of the plant is carried out every 4000 evaluated operating hours (EOH). The second minor inspection is carried out every 8000 EOH.

The major inspection is usually carried out every 16,000 EOH.

Eight mechanical workers are needed for type A inspection on a particular plant, 8 - 10 workers are needed for type B inspection, while 15 workers are needed for type C inspection. For daily plant checks, only one maintenance worker is needed to handle a plant. Information from plant daily checks, inspections, and other parameters are used by the planning department to develop an annual maintenance schedule.

This study is concerned with scheduling the Mechanical Department (MD) maintenance line, which is in charge of inspections and daily plant maintenance, checks of the mechanical components of the power plant. Based on inspection requirements and past experience, the maintenance engineer of the MD estimated the number of workers needed to satisfy the daily routine of maintenance activities for a typical week. As shown in **Table 1**, the plant maintenance needs up to 16 workers daily for the morning shift (8:00 am - 4:00 pm), and up to 7 workers daily for each of the afternoon shift (4:00 pm - 12:00 am) and the night shift (12:00 am - 8:00 am).

3. Models for Scheduling Days-Off

Alfares [1,2] developed algorithms to minimize the number of workers for (5,7) problem. Each worker must work for 5 days per week and only consecutive pairs of off days are allowed. This algorithms is applicable to

Table 1. Daily maintenance labour demands for the three shifts.

Day i	Mo - 1	Tu - 2	We - 3	Th - 4	Fr - 5	Sa - 6	Su - 7
Morning labour demand	16	16	16	16	16	8	7
Afternoon labour demand	6	6	7	7	7	6	6
Night labour demand	6	6	7	7	7	6	6

Afam power station with respect to providing an alternative workforce schedule as well as other organization operating a seven day a week. The integer linear programming (ILP) formulation is given as follows:

$$\text{Minimize } w = \sum_{i=1}^{7} c_j x_j \tag{1}$$

Subject to

$$\left(\sum_{j=1}^{7} x_j\right) - x_i - x_{i-1} \geq r_i, \quad i = 1, \cdots, 7 \tag{2}$$

$$x_j \geq 0 \text{ and an integer, } j = 1, \cdots, 7 \tag{3}$$

where:

x_j = number of workers, assigned to a days off pattern j *i.e.* number of workers off on days, j and j + 1.

c_j = weekly cost of days-off pattern j per employee, with premium pay for weekend workdays.

r_i = number of workers required on day i.

w = workforce size, that is total number of workers assigned to days-off patterns.

Alfares [1,2] further stated that since $\sum_{j=1}^{7} x_j$ is equal to w, Equation (2) can be written as:

$$x_i + x_{i-1} \leq b_i, \; i = 1, \cdots, 7 \tag{4}$$

whereas

$$b_i = w - r_i \tag{5}$$

= maximum number of workers off on day i.

Alfares [1] developed a simple equation that yields the minimum workforce size W for the (5,7) problem. The expression he obtained for minimum workforce size W is:

$$W = \sum_{j=1}^{7} x_j = \max\left\{ r_{max}, \left\lceil \frac{1}{5}\sum_{i=1}^{7} r_i \right\rceil, \left\lceil \frac{R_{max}}{3} \right\rceil \right\} \tag{6}$$

where $r_{max} = \max\{r_1, r_2, \cdots, r_7\}$

$R_{max} = \max\{R_1, R_2, \cdots, R_7\}$

$\lceil a \rceil$ = Smallest integer $\geq a$

$$R_i = \sum_{j \in S_i} r_j \quad i = 1, 2, \cdots, 7 \tag{7}$$

$$S_i = \text{Set of 4 subscripts} \subset \{1, 2, 3, 4, 5, 6, 7\}, \; i = 1, 2, \cdots, 7 \tag{8}$$

set S_1 to S_7 are shown in **Table 2**.

After determining the workforce size, he uses a simple algorithm to determine the number of workers assigned to each days-off pattern, $x_1, \cdots, x_7$. The result are shown in **Table 3**. Hence completing the optimum solution of the (5,7) problem without linear or interger programming software.

Algorithm Step [2]

1) Determine the minimum workforce size W using Equation (6) $W = \max\{r_{max}, \lceil \sum r_i/5 \rceil, \lceil R_{max}/3 \rceil\}$. In the case of ties go to step 3.

2) (a) If $\max\{r_{max}, \lceil \sum r_i/5 \rceil, \lceil R_{max}/3 \rceil\} = \lceil R_{max}/3 \rceil$, then:

- if $R_{max}/3$ is not integer, increment R_i = R_{max} by ($3W - R_{max}$) to make it a multiple of 3; among the four daily labor demands r_j, j E S_i, that can be increased, avoid whenever possible: 1) weekend, *i.e.*, r_6 and r_7; and 2) the maximum labor demand r_{max}.
- Calculate $b_1, b_2, \cdots, b_7$ using Equation (5), then apply system No. i, i =1,⋯,7, in **Table 3** to find $x_1, x_2, \cdots, x_7$. If there are ties for maximum value $R_i = R_{max}$, use any of the systems No. i corresponding to the applicable values of the index i arbitrarily.

(b) If $\max\{r_{max}, \lceil \sum r_i/5 \rceil, \lceil R_{max}/3 \rceil\} = \lceil \sum r_i/5 \rceil$, then:

- if $\sum r_i/5$ is not integer, increment $\sum r_i$ by $(5W - \sum r_i)$ in order to make it a multiple of 5; among all seven daily labor demands $r_1, \cdots, r_7$, chose the ones to be increased according to the criteria given above in step 2(a).
- calculate $b_1, b_2, \cdots, b_7$ using Equation (5), then apply system No. 8 in **Table 3** to find $x_1, x_2, \cdots, x_7$.

(c) If $\max\{r_{max}, \lceil \sum r_i/5 \rceil, \lceil R_{max}/3 \rceil\} = r_{max}$, then:

- calculate $b_1, b_2, \cdots, b_7$ using Equation (5), then apply system No. 9 in **Table 3** to find $x_1, x_2, \cdots, x_7$.

Table 2. Sets of subscripts define by Equation (8) and their complement [1].

i	S_i
1	1,2,4,6
2	2,3,5,7
3	3,4,6,1
4	4,5,7,2
5	5,6,1,3
6	6,7,2,4
7	7,1,3,5

Table 3. Values of Days off Assignment, $x_1,\cdots,x_7$ for all possible values of W. [2].

No	w	x_1	x_2	x_3	x_4	x_5	x_6	x_7
1	$\lceil R_1/3 \rceil$	0	b_2	$b_4 - x_4$	min $\{b_4, b_5 - x_5\}$	$b_6 - x_6$	min $\{b_6, b_7 - b_1\}$	b_1
2	$\lceil R_2/3 \rceil$	b_2	0	b_3	$b_5 - x_5$	min $\{b_5, b_6 - x_6\}$	min $\{b_7, R_2 - R_6\}$	$b_7 - x_6$
3	$\lceil R_3/3 \rceil$	$b_1 - x_7$	b_3	0	b_4	$b_6 - x_6$	min $\{b_6, R_3 - R_6\}$	min $\{b_1, b_7 - x_6\}$
4	$\lceil R_4/3 \rceil$	min $\{b_1, b_2\}$	$b_2 - x_1$	b_4	0	b_5	min $\{b_7, b_6 - b_5\}$	$b_7 - x_6$
5	$\lceil R_5/3 \rceil$	$b_1 - x_7$	min $\{b_2, b_3\}$	$b_3 - x_2$	b_5	0	b_6	min $\{b_1, b_7 - b_6\}$
6	$\lceil R_6/3 \rceil$	min $\{b_1 - b_7, b_2\}$	min $\{b_2, R_6 - R_2\}$	min $\{b_4, b_3 - x_2\}$	$b_4 - x_3$	b_6	0	b_7
7	$\lceil R_7/3 \rceil$	b_1	$b_3 - x_3$	min $\{b_3, b_4 - x_4\}$	$b_5 - x_5$	min $\{b_5, b_6 - b_7\}$	b_7	0
8	$\lceil \sum r_i/3 \rceil$	$w - b_3 - b_5 - b_7$	$b_2 - x_1$	$b_3 - x_2$	$b_4 - x_3$	$b_5 - x_4$	$b_6 - x_5$	$b_7 - x_6$
9	r_{max}	min $\{b_1 - x_7, b_2 - x_2, w - \sum_{i=2}^{7} x_i\}$	min $\{b_2, b_3 - x_3, w - \sum_{i=3}^{7} x_i\}$	min $\{b_3, b_4 - x_4, w - \sum_{i=4}^{7} x_i\}$	min $\{b_4, b_5 - x_5, w - \sum_{i=5}^{7} x_i\}$	min $\{b_5, b_6 - x_6, w - x_6\}$	min $\{b_6, b_7\}$	min $\{b_1, b_7 - x_6, w - x_5 - x_6\}$

3) In the case of tiers for r_{max}, $\lceil \sum r_i/5 \rceil$, $\lceil R_{max}/3 \rceil$, choose the argument that needs the minimum total increment and go to the corresponding step: 2(a) for $\lceil R_{max}/3 \rceil$, 2(b) for $\lceil \sum r_i/5 \rceil$, and 2(c) for r_{max}. While r_{max} does not need incrementing, the increment for $\lceil \sum r_i/5 \rceil$ is $(5W - \sum r_i)$, and the increment for $\lceil R_{max}/3 \rceil$ is $(3W - R_{max})$.

For easy execution of the above algorithm a simple computer programme is written in QUICK BASIC.

4. Computer Program for Solving (5,7) Problem Based on Alfares [1,2] Algorithm

```
DECLARE SUB DRIM ()
DECLARE SUB PRINTALL ()
DECLARE SUB XRI ()
DECLARE FUNCTION SRIMIN! (A!, B!)
DECLARE SUB XR ()
DECLARE FUNCTION RMIN! (A!, B!, C!)
DECLARE SUB DAILYREQUIREMENT ()
DECLARE SUB CALBS ()
DECLARE SUB WORKFORCE ()
CLS
DIM SHARED RI(7), R1(7), R2(7), R3(7), R4(7), R5(7), R6(7), R7(7), SUMR(7), B(7)
DIM SHARED WF, RiMAX, SUMRi5, SUMRMAX, IT$, SUMRi, X(7)
DIM SHARED FLAG AS STRING
PRINT "ENTER DAILY LABOUR REQUIREMENT"
INPUT "ENTER LABOUR REQUIREMENT FOR MON: ", RI(1)
INPUT "ENTER LABOUR REQUIREMENT FOR TUE: ", RI(2)
INPUT "ENTER LABOUR REQUIREMENT FOR WED: ", RI(3)
INPUT "ENTER LABOUR REQUIREMENT FOR THU: ", RI(4)
INPUT "ENTER LABOUR REQUIREMENT FOR FRI: ", RI(5)
INPUT "ENTER LABOUR REQUIREMENT FOR SAT: ", RI(6)
INPUT "ENTER LABOUR REQUIREMENT FOR SUN: ", RI(7)
CALL DAILYREQUIREMENT
RiMAX = 0
FOR I = 1 TO 7
    IF RI(I) > RiMAX THEN
        RiMAX = RI(I)
   END IF
NEXT I
SUMRi = 0
FOR I = 1 TO 7
    SUMRi = RI(I) + SUMRi
NEXT I
SUMRi5 = SUMRi / 5!
SUMRMAX = 0
FOR I = 1 TO 7
  IF SUMR(I) > SUMRMAX THEN
     SUMRMAX = SUMR(1)
     SELECT CASE I
      CASE 1
        IT$ = "R1"
     CASE 2
           IT$ = "R2"
        CASE 3
          IT$ = "R3"
       CASE 4
```

```
            IT$ = "R4"
        CASE 5
            IT$ = "R5"
        CASE 6
            IT$ = "R6"
        CASE 7
        IT$ = "R7"
        END SELECT
    END IF
  NEXT I
  SUMRMAX = SUMRMAX / 3
  CALL WORKFORCE
  END
  SUB CALBS
  FOR I = 1 TO 7
      B(I) = WF – RI(I)
  NEXT I
  END SUB

  SUB DAILYREQUIREMENT
  R1(1) = RI(1): R1(2) = RI(2): R1(4) = RI(4): R1(6) =
RI(6)
  R2(2) = RI(2): R2(3) = RI(3): R2(5) = RI(5): R2(7) =
RI(7)
  R3(3) = RI(3): R3(4) = RI(4): R3(6) = RI(6): R3(1) =
RI(1)
  R4(4) = RI(4): R4(5) = RI(5): R4(7) = RI(7): R4(2) =
RI(2)
  R5(5) = RI(5): R5(6) = RI(6): R5(1) = RI(1): R5(3) =
RI(3)
  R6(6) = RI(6): R6(7) = RI(7): R6(2) = RI(2): R6(4) =
RI(4)
  R7(7) = RI(7): R7(1) = RI(1): R7(3) = RI(3): R7(5) =
RI(5)
  SUMRi = 0: SUMR(1) = 0: SUMR(2) = 0: SUMR(3)
= 0: SUMR(4) = 0: SUMR(5) = 0
  SUMR(6) = 0: SUMR(7) = 0
  FOR K = 1 TO 7
      SUMRi = SUMRi + RI(K): SUMR(1) = SUMR(1)
+ R1(K): SUMR(2) = SUMR(2) + R2(K)
      SUMR(3) = SUMR(3) + R3(K): SUMR(4) =
SUMR(4) + R4(K): SUMR(5) = SUMR(5) + R5(K)
      SUMR(6) = SUMR(6) + R6(K): SUMR(7) =
SUMR(7) + R7(K)
  NEXT K

  END SUB

  SUB DETERMINEWF
      J = CINT(WF)
      IF J < WF OR J > WF THEN
          FLAG = "TRUE"
      ELSE
          FLAG = "FALSE"
      END IF
END SUB

SUB DRIM
IF IT$ = "R1" THEN
      X(7) = B(1)
      X(6) = SRIMIN(B(6), B(7) – B(1))
      X(5) = B(6) – X(6)
      X(4) = SRIMIN(B(4), B(5) – X(5))
      X(3) = B(4) – X(4)
      X(2) = B(2)
      X(1) = 0
ELSEIF IT$ = "R2" THEN
      X(1) = B(2)
      X(2) = 0
      X(3) = B(3)
      X(6) = SRIMIN(B(7), SUMR(2) – SUMR(6))
      X(5) = SRIMIN(B(5), B(6) – X(6))
      X(4) = B(5) – X(5)
      X(7) = B(7) – X(6)
ELSEIF IT$ = "R3" THEN
      X(2) = B(3)
      X(3) = 0
      X(4) = B(4)
      X(6) = SRIMIN(B(6), SUMR(3) – SUMR(6))
      X(7) = SRIMIN(B(1), B(7) – X(6))
      X(5) = B(6) – X(6)
      X(1) = B(1) – X(7)
ELSEIF IT$ = "R4" THEN
      X(1) = SRIMIN(B(1), B(2))
      X(2) = B(2) – X(1)
      X(3) = B(4)
      X(4) = 0
      X(5) = B(5)
      X(6) = SRIMIN(B(7), B(5) – B(6))
      X(7) = B(7) – X(6)
ELSEIF IT$ = "R5" THEN
      X(7) = SRIMIN(B(1), B(7) – B(6))
      X(2) = SRIMIN(B(2), B(3))
      X(1) = B(1) – X(7)
      X(3) = B(3) – X(2)
      X(5) = 0
      X(4) = B(5)
      X(6) = B(6)
ELSEIF IT$ = "R6" THEN
      X(2) = SRIMIN(B(2), SUMR(6) – SUMR(2))
      X(3) = SRIMIN(B(4), B(3) – X(2))
      X(4) = B(4) – X(3)
      X(1) = B(2) – X(1)
      X(6) = 0
      X(7) = B(7)
      X(5) = B(6)
ELSE
      X(5) = SRIMIN(B(5), B(6) – B(7))
```



```
        X(4) = B(5) – X(5)
        X(3) = SRIMIN(B(3), B(4) – X(4))
        X(2) = B(3) – X(3)
        X(1) = B(1)
        X(7) = 0
        X(6) = B(7)
    END IF



    END SUB

    SUB PRINTALL
    CLS
    PRINT " 1"; TAB(7); " 2"; TAB(14); " 3"; TAB(21); " 4"; TAB(28); " 5"; TAB(35); " 6"; TAB(42); " 7"; TAB(49); " SUM"
    PRINT "________________________________________________________"
    PRINT
    PRINT RI(1); TAB(7); RI(2); TAB(14); RI(3); TAB(21); RI(4); TAB(28); RI(5); TAB(35); RI(6); TAB(42); RI(7); TAB(49); SUMRi
    PRINT R1(1); TAB(7); R1(2); TAB(14); R1(3); TAB(21); R1(4); TAB(28); R1(5); TAB(35); R1(6); TAB(42); R1(7); TAB(49); SUMR(1)
    PRINT R2(1); TAB(7); R2(2); TAB(14); R2(3); TAB(21); R2(4); TAB(28); R2(5); TAB(35); R2(6); TAB(42); R2(7); TAB(49); SUMR(2)
    PRINT R3(1); TAB(7); R3(2); TAB(14); R3(3); TAB(21); R3(4); TAB(28); R3(5); TAB(35); R3(6); TAB(42); R3(7); TAB(49); SUMR(3)
    PRINT R4(1); TAB(7); R4(2); TAB(14); R4(3); TAB(21); R4(4); TAB(28); R4(5); TAB(35); R4(6); TAB(42); R4(7); TAB(49); SUMR(4)
    PRINT R5(1); TAB(7); R5(2); TAB(14); R5(3); TAB(21); R5(4); TAB(28); R5(5); TAB(35); R5(6); TAB(42); R5(7); TAB(49); SUMR(5)
    PRINT R6(1); TAB(7); R6(2); TAB(14); R6(3); TAB(21); R6(4); TAB(28); R6(5); TAB(35); R6(6); TAB(42); R6(7); TAB(49); SUMR(6)
    PRINT R7(1); TAB(7); R7(2); TAB(14); R7(3); TAB(21); R7(4); TAB(28); R7(5); TAB(35); R7(6); TAB(42); R7(7); TAB(49); SUMR(7)
    PRINT
    PRINT "WORKFORCE = ", WF
    PRINT
    PRINT "X1 =", X(1), "B1 = ", B(1)
    PRINT "X2 =", X(2), "B2 = ", B(2)
    PRINT "X3 =", X(3), "B3 = ", B(3)
    PRINT "X4 =", X(4), "B4 = ", B(4)
    PRINT "X5 =", X(5), "B5 = ", B(5)
    PRINT "X6 =", X(6), "B6 = ", B(6)
    PRINT "X7 =", X(7), "B7 = ", B(7)
    END SUB

    FUNCTION RMIN (A, B, C)
    IF A <= B AND A <= C THEN
        RMIN = A
    ELSEIF B < A AND B <= C THEN
        RMIN = B
    ELSE
        RMIN = C
    END IF
    END FUNCTION

    FUNCTION SRIMIN (A, B)
    IF A <= B THEN
        SRIMIN = A
    ELSE
        SRIMIN = B
    END IF
    END FUNCTION

    SUB WORKFORCE
    IF RiMAX >= SUMRi5 AND RiMAX >= SUMRMAX THEN
        WF = RiMAX
        CALL CALBS
        CALL XR
        CALL PRINTALL
    ELSEIF SUMRi5 > RiMAX AND SUMRi5 >= SUMRMAX THEN
        WF = SUMRi5
        FLAG = "FALSE"
        IF FLAG = "TRUE" THEN
            INCR = 5 * WF – SUMRi
            SUMRi = SUMRi + (5 * WF – SUMRi)
            WF = SUMRi / 5!
            FOR J = 1 TO 7
                IF (RI(J) + INCR) <= RiMAX THEN
                    RI(J) = RI(J) + INCR
                    CALL DAILYREQUIREMENT
                    EXIT FOR
                END IF
            NEXT J
        END IF
        CALL CALBS
        CALL XRI
        CALL PRINTALL
    ELSE
        WF = SUMRMAX
        FLAG = "FALSE"
        IF FLAG = "TRUE" THEN
            INCR = 3 * WF – SUMRMAX
            SUMRMAX = SUMRMAX + (3 * WF – SUMRMAX)
```

```
            WF = SUMRMAX / 3!

                IF IT$ = "R1" THEN
                FOR J = 1 TO 7
                    IF R1(J) <> 0 AND (R1(J) + INCR) <= RiMAX THEN
                        RI(J) = RI(J) + INCR
                        CALL DAILYREQUIREMENT
                        EXIT FOR
                    END IF
                NEXT J
                ELSEIF IT$ = "R2" THEN
                FOR J = 1 TO 7
                    IF R2(J) <> 0 AND (R2(J) + INCR) <= RiMAX THEN
                        RI(J) = RI(J) + INCR
                        CALL DAILYREQUIREMENT
                        EXIT FOR
                    END IF
                NEXT J
                ELSEIF IT$ = "R3" THEN
                FOR J = 1 TO 7
                    IF R3(J) <> 0 AND (R3(J) + INCR) <= RiMAX THEN
                        RI(J) = RI(J) + INCR
                        CALL DAILYREQUIREMENT
                        EXIT FOR
                    END IF
                NEXT J
                ELSEIF IT$ = "R4" THEN
                FOR J = 1 TO 7
                    IF R4(J) <> 0 AND (R4(J) + INCR) <= RiMAX THEN
                        RI(J) = RI(J) + INCR
                        CALL DAILYREQUIREMENT
                        EXIT FOR
                    END IF
                NEXT J
                ELSEIF IT$ = "R5" THEN
                FOR J = 1 TO 7
                    IF R5(J) <> 0 AND (R5(J) + INCR) <= RiMAX THEN
                        RI(J) = RI(J) + INCR
                        CALL DAILYREQUIREMENT
                        EXIT FOR
                    END IF
                NEXT J
                ELSEIF IT$ = "R6" THEN
                FOR J = 1 TO 7
                    IF R6(J) <> 0 AND (R6(J) + INCR) <= RiMAX THEN
                        RI(J) = RI(J) + INCR
                        CALL DAILYREQUIREMENT
                        EXIT FOR
                    END IF
                NEXT J
                ELSE
                FOR J = 1 TO 7
                    IF R7(J) <> 0 AND (R7(J) + INCR) <= RiMAX THEN
                        RI(J) = RI(J) + INCR
                        CALL DAILYREQUIREMENT
                        EXIT FOR
                    END IF
                NEXT J
               END IF
            END IF
        CALL CALBS
        CALL DRIM
        CALL PRINTALL
    END IF

    END SUB

    SUB XR
    IF B(6) <= B(7) THEN
        X(6) = B(6)
    ELSE
        X(6) = B(7)
    END IF
    X(5) = RMIN(B(5), B(6) – X(6), WF – X(6))
    X(7) = RMIN(B(1), B(7) – X(6), WF – X(5) – X(6))
    X(4) = RMIN(B(4), B(5) – X(5), WF – (X(5) + X(6) + X(7)))
    X(3) = RMIN(B(3), B(4) – X(4), WF – (X(4) + X(5) + X(6) + X(7)))
    X(2) = RMIN(B(2), B(3) – X(3), WF – (X(3) + X(4) + X(5) + X(6) + X(7)))
    X(1) = RMIN(B(1) – X(7), B(2) – X(2), WF – (X(3) + X(2) + X(4) + X(5) + X(6) + X(7)))
    END SUB

    SUB XRI
    X(1) = WF – B(3) – B(5) – B(7)
    X(2) = B(2) – X(1)
    X(3) = B(3) – X(2)
    X(4) = B(4) – X(3)
    X(5) = B(5) – X(4)
    X(6) = B(6) – X(5)
    X(7) = B(7) – X(6)
    END SUB
```

5. Result and Discussion

Afam power station Nigeria currently is operating a 5day schedule whereby scheduling workers on weekend (Saturday and Sunday) on overtime basis. To make it possible for her to change to a seven day schedule which is more efficient and less capital intensive, this software was developed. The software is still applicable to other organization operating seven days like airline, restaurant and police station.

The workforce size and days-off assignment for Afam power station Nigeria morning shift was determined using this software. The seven morning-time daily labour requirement shown in **Table 1** were used as input data. The result is shown in **Figure 1**.

Table 4 below is used to prove if the workforce size of 19 workers can satisfy morning shift daily labour demand in **Table 1**, **Table 4** also represent seven shifts *i.e* 10 workers are assigned to shift 6 (off on weekends, which is Saturday and Sunday).

However a seven-day schedule (workforce size and days-off assignment) for afternoon or night shifts was generated using the optimization software. The seven afternoon-time or night-time daily labour requirements shown in **Table 1** were used as input data. The result is also shown in **Figure 2**.

Table 5 below is use to prove if the workforce size of 9 workers can satisfy afternoon or night labour demand of **Table 1**.

6. Comparison between Current 5-Day Schedule and 7-Day Schedule [3]

A 7-day schedule in this paper is a schedule in which each worker must work five days per week, with two

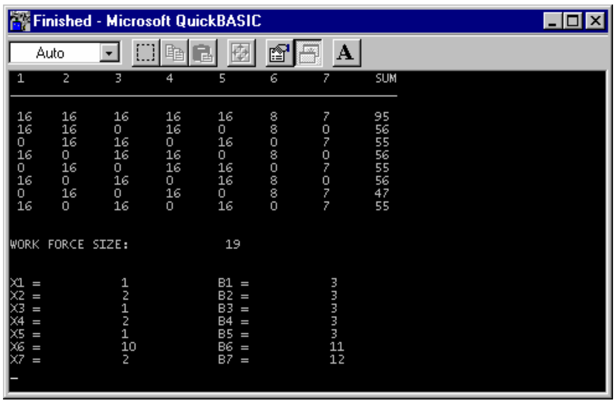

Figure 1. Results of analysis of 7-morning time shift.

Table 4. Total morning daily worker assignment.

Day	Morning Shifts							
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	Total
Monday	a	2	1	2	1	10	a	16
Tuesday	a	a	1	2	1	10	2	16
Wednesday	1	a	a	2	1	10	2	16
Thursday	1	2	a	a	1	10	2	16
Friday	1	2	1	a	a	10	2	16
Saturday	1	2	1	2	a	a	2	8
Sunday	1	2	1	2	1	a	a	7

Note: a = off day.

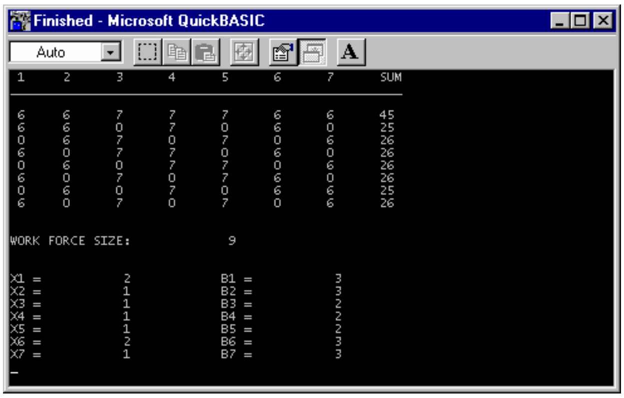

Figure 2. Results of analysis of 7-afternoon time shift.

Table 5. Total afternoon or night daily worker assignment.

Day	Afternoon or Night Shifts							
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	Total
Monday	a	1	1	1	1	2	a	6
Tuesday	a	a	1	1	1	2	1	6
Wednesday	2	a	a	1	1	2	1	7
Thursday	2	1	a	a	1	2	1	7
Friday	2	1	1	a	a	2	1	7
Saturday	2	1	1	1	a	a	1	6
Sunday	2	1	1	1	1	a	a	6

Note: a = off day.

consecutive days off. While 5 day schedule workers are schedule on weekend on overtime basis.

6.1. Current 5-Day Schedule

The maximum labour demand for the morning shift on weekdays is 16. Thus, 16 workers are assigned to the morning shift work on the five weekdays (Mo-Fr) on regular-time basis. Similarly, seven workers are assigned to each of the afternoon and night shifts on weekdays on regular time basis. Therefore, the total workforce is equal to 16 + 7 + 7 = 30 workers. Out of this workforce, 6 - 8 employees are assigned to weekend work on overtime basis.

The shift premium pay rate is 110% for the afternoon shift and 115% for the night shift. The weekend pay rate for any shift is 50% higher than the weekdays pay rate. The details of the pay hours calculations are shown in **Table 6**. The morning shift requires 820 weekly pay hours, while the afternoon and night shift require a total of 942 weekly pay hours. The total workforce size is 30 workers and the weekly pay hours are equal to 1762.

6.2. 7-Day Workweek Schedule

This alternative is used to satisfy labour demands with a seven-day workweek schedule for all three work shifts. However, a seven-day schedule is determined for the

Table 6. Calculation of the total pay hours per week for current 5-day schedule [3].

Work Times	No. of Workers (W)	Hours/day H = 8 × days	Hours/Week T = W × H	Pay Rate (%) R	Pay Hours P = T × R
Mo-Fr morning	16	40	640	100	640
Sa morning	8	8	64	150	96
Su morning	7	8	56	150	84
Mo-Fr afternoon	7	40	280	110	308
Sa afternoon	6	8	48	160	76.8
Su afternoon	6	8	48	160	76.8
Mo-Fr night	7	40	280	115	322
Sa night	6	8	48	165	79.2
Su night	6	8	48	165	79.2

morning-shift assignment, using the Quick Basic computer programme for solving (5,7) problem. The seven morning-time daily labour requirements shown in **Table 1** were used as input data. The optimum solution requires 19 maintenance workers (W = 19) to satisfy morning labour demands as shown below. Days-off assignments for the 19 morning shift workers are as follows:

$x_1 = 1, x_2 = 2, x_3 = 1, x_4 = 2, x_5 = 1, x_6 = 10, x_7 = 2.$

Each worker is paid for five workdays per week, at eight hours per day, all at regular time. Hence:

Morning-shift pay hours per week = 19 × 5 × 8 = 760.

We now consider a seven-day schedule for both the afternoon and night shift assignments. We also use the Quick Basic computer programme for solving (5,7) problem, where the seven afternoon- or night-time daily labour requirements shown in **Table 1** were used as input data. The optimum solution requires nine workers (W = 9) for each of the afternoon and the night shift, whose days-off assignments are as follows:

$x_1 = 2, x_2 = 1, x_3 = 1, x_4 = 1, x_5 = 1, x_6 = 2, x_7 = 1.$

In order to calculate the pay hours, it must be noted that the afternoon shift is paid at a premium rate of 110%, while the night shift is paid at a premium rate of 115%.

Afternoon shift pay hours per week = 9 × 5 × 8 × 1.10 = 396.

Night shift pay hours per week = 9 × 5 × 8 × 1.15 = 414.

Adding up the morning-shift assignment, requiring 19 employees and 760 weekly pay hours, we obtain:

Total pay hours per week = 760 + 396 + 414 = 1570.

Total number of employees = 19 + 9 + 9 = 37.

Comparison of the two alternative work schedules is shown in **Table 7** below. From **Table 7** it is obvious that a 7-day schedule, which leads to 11 percent labour cost saving, is the best alternative. Therefore, a 7-schedule was recommended to the management of Afam power station. Though the proposed schedule requires seven more maintenance workers, this is not a serious draw back because Afam power station has an abundant pool of underutilized mechanical maintenance workers. Hence with the current workforce, the proposed schedule can be implemented even without employing more maintenance workers.

Table 7. Comparison of work schedules [3].

Alternative work schedule	Pay hours per week	Savings in pay hours	Workforce size	Increase in workforce
Current 5-Day Schedule	1762	-	30	-
7-Day Workweek Schedule	1570	10.9%	37	23.3%

7. Conclusions

The traditional five-day work schedule currently being practiced in the Afam power-station Nigeria is neither efficient nor cost effective. The purpose of this paper is the optimization of the workforce scheduling for solving maintenance problem and improving productivity. In the light of these, this paper has presented a new, efficient optimization software for the cyclic (5,7) days-off scheduling problem.

The software is capable of determining workforce size and assigning workers to days-off pattern for Afam power station and other organization like airline, police station and restaurant operating a 7 day a week. The software does not require specialized training unlike integer programming software.

The software was tested with data from Afam power station Nigeria. Test result shows that the software is capable of generating a seven day schedule. The seven day Schedule is more efficient and cost effective than the five day work schedule. When comparison was made between the existing five-day schedule practiced by Afam power station Nigeria and the seven-day schedule generated by this Software. The seven-day schedule is expected to produce savings of 11% maintenance labour cost annually [3].

8. Acknowledgements

Emovon, Ikuobase would like to thank the managements of the Federal University of Petroleum Resources, Effurun, Nigeria for providing an enabling environment for carrying out this study.

REFERENCES

[1] H. K. Alfares, "An Efficient Two-Phase Algorithm for Cyclic Days-Off Scheduling," *Computers and Operations Research*, Vol. 25, No. 11, 1998, pp. 913-924. doi:10.1016/S0305-0548(98)00033-1

[2] H. K. Alfares, "Efficient Optimization of Cyclic Labour Days-Off Scheduling," *OR Spectrum*, Vol. 23, No. 2, 2001, pp. 283-294. doi:10.1007/PL00013353

[3] H. K. Alfares, M. T. Lilly and I. Emovon, "Maintenance Staff Scheduling at Afam Power Station," *Industrial Engineering and Management System*, Vol. 6, No. 1, 2007, pp. 22-27.

[4] R. L. Fagbenle, "Prospects and Problems of Solarizing Transport Technology," *Nigerian Journal of Renewable Energy*, Vol. 2, No. 1, 1991.

[5] H. K. Alfares, "Flexible 4-Day Workweek Scheduling with Weekend Work Frequency Constraints," *Computer and Industrial Engineering*, Vol. 44, No. 3, 2003, pp. 325-338. doi:10.1016/S0360-8352(02)00192-4

[6] I. Emovon, "Optimization of Workforce Scheduling for Solving Maintenance Problem in a Power Station (A Case Study of Afam Thermal Power Station)," M.Tech. Thesis, Rivers State University of Science and Technology, Nigeria, 2007.