COMPUTER AIDED MAINTENANCE PLANNING

by

REMZİ TÜRECİ

B.S. in M.E., Boğaziçi University, 1983

Submitted to the Institute for Graduate Studies in Science and Engineering in partial fulfillment of the requirements for the degree of

Master of Science

in

Industrial Engineering

Boğaziçi University

1985

ACKNOWLEDGEMENTS

I wish to express my sincere gratitude to my thesis advisor, Doç.Dr. Ali Rıza Kaylan, for his valuable guidance and his continuous support, encouragement and understanding throughout all phases of this study.

I also sincerely wish to thank Doç. Dr. Ahmet Kuzucu and Doç. Dr. İlhan Or for their valuable comments and suggestions.

I also would like to thank Dr. Ömer Cerid for his valuable computer assistance.

I also would like to thank Mr. Uran Tiryakioğlu, Technical Vice President of Bekoteknik San. A.Ş., and to Mr. Bülent Öztürk, Manager of Industrial Engineering Department of the same company, for their continuous support, understanding and patience throughout this study.

ABSTRACT

Recently, in developing countries, the importance of maintenance planning is being well understood by the management of big companies, especially in industries where losses caused by sudden failures are very high.

For every company, there exists a trade off between the costs incurred in unexpected failures and costs invested in the planned maintenance activities of the company.

On the one hand, sudden failures lead into significant production time losses, which, in turn, usually have very high associated costs. On the other hand, for a planned maintenance system, the number of necessary maintenance personnel is higher, and a safety spare parts stock should be kept available at hand, which both have a considerable cost, but since the replacement time is much shorter, the down time costs will be negligable when compared with failure replacement policy.

In this thesis it is aimed to devise an interactive computer program to aid the decision maker in planning maintenance activities. The program will help the maintenance planner in developing efficient scheduling, as well as identifying correct maintenance periods and determining minimum safety spare parts stock levels.

Bekoteknik San. A.Ş., a producer of consumer electronics, is selected as the pilot factory, but the model is designed in such a way to enable easy adaptation to other factories, provided past data about the current machinery is available.

ÖZET

Son yıllarda, gelişmekte olan ülkelerde, özellikle ani durmalardan doğan maliyetlerin çok yüksek olduğu büyük şirketlerde, Bakım Planlama gittikçe daha büyük ilgi görmektedir.

Ancak planlı bakım belli maliyetler getirmektedir. Belirli büyüklükte bir bakım ekibinin ve yedek parça stoğunun sürekli işletme bünyesinde bulundurulması gerekmektedir. Her şirkette, bu planlı bakım maliyetleri ile ani durmalardan doğan maliyetlerin bir karşılaştırması yapılmalıdır.

Bir taraftan, ani makina durmaları, maliyeti çok yüksek olan üretim zamanı kayıplarına sebep olmakta, diğer taraftan, planlı bakım masrafları, daha düşük süreli makina durmalarını sağlayacağından, kendi maliyetini fazlasıyla geri ödemektedir.

Bu tezde, karar vericiye bakım planlamasında yardımcı olacak etkileşimli bir kompüter programı geliştirilmesi amaçlanmıştır. Program bakım planlamacıya hızlı ve sağlıklı bir şekilde makina bakım zamanları belirlenmesinde, doğru bakım periodlarının, sipariş miktarı ve minimum stok seviyelerinin saptanmasında yardımcı olacak şekilde düşünülmüştür.

Bekoteknik San. A.Ş., bir elektronik mamuller üreticisi, bu çalışmada örnek işletme olarak alınmıştır. Program bu firma ihtiyaçları doğrultusunda hazırlanmış olmakla birlikte mevcut makinalar için geçmiş bilgileri olan her türlü fabrikada kullanılabilecek şekildedir.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

1.1 Objectives of the Study

This study mainly comprises the following subjects. First, the model acts as a data input, data output, information accumulating media. Every action that has been undertaken the maintenance activities can be stored into the corresponding files.

Data about the past maintenance activities can be analysed, and optimal periodic replacement times of parts can be identified. For all planned maintenance activities of a factory, annual manhour requirement can be calculated.

Periodic reports of maintenance activities can be generated to be utilized in the performance evaluation of the maintenance department.

Finally, reorder points and economic lot size order quantities of parts can be calculated. Stock level of parts are continuously controlled and the user is immediately warned as a spare part drops to its reorder quantity level.

The computer program developed will aid the maintenance planner in developing fast and accurate scheduling, as well as identifying correct maintenance periods and determining minimum safety spare parts stock levels.

At any time, interruptions or changes in the maintenance plans are permitted and the decision maker has the flexibility to revise the schedules for each particular period.

Each preventive maintenance, repair or replacement activity is entered into the program. Hence data is accumalated in the file of each particular unit to allow the submodels to be run when desired to update maintenance periods or safety spare parts stock levels. This is portrayed in Figure 1.1.

This program can be utilized as a vehicle to measure performance of the maintenance department. By keeping track of the maintenance activities on the computer, this tool reflects how effectively the management goals are met.

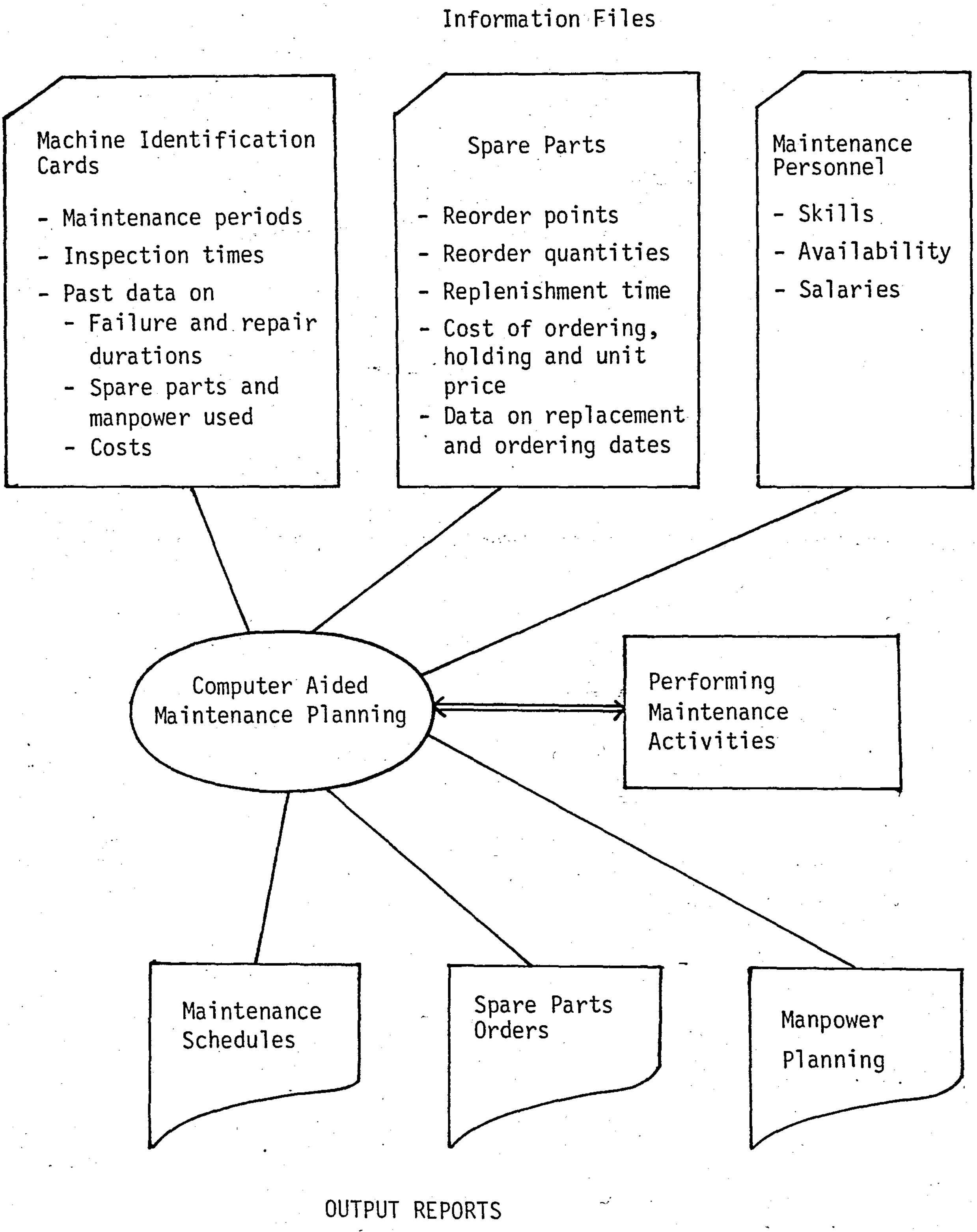


Figure 1.1 Data Input and Information Output of the Designed Computer Aided Maintenance Planning

1.2 Literature Survey on Decision Support Systems

The model developed in this study will help the decision-maker in preparing fast and accurate scheduling of maintenance activities, finding out correct replacement periods and determining minimum safety spare parts stock levels. A decision support system is defined by Micheal Scott Morton as: "Any interactive, computer based system that helps decision-makers utilize data and models to solve unstructured problems" (Sprague, 19).

Within the last decade, fast developments in data handling systems such as personal computers, computer networks, large data bases, color graphics and computer-based models lead to an increasing interest in the use of computers to support decision-making. Such uses are being called Decision Support Systems (DSS).

Since there must be a close relationship between the decision-makers and the system in order to be effective, DSS differ from the traditional recordkeeping. For this close relationship to occur, the decision-maker must understand what the DSS can do.

The first concepts involved in DSS are presented by Michael S. Scott Morton as "Management Decision Systems", in 1970's. He sought the possibilities for improving the management problem solving through the use of a visual display device, and made the aforementioned definition of DSS.

This definition, however, is thought as a restrictive one, and later, a more broad definition as 'any system that makes some contribution to decision-making' or 'any system that supports a decision' is proposed.

Usually, managers have informal models or processes that they use to solve problems as they perceive them. In this form, problem solving becomes an intuitive judgemental kind of process. It seems desirable to help the decision maker build as firm a base from the facts as possible, and only then have him apply his judgement to this base. This is done by providing him necessary materials such as highly developed computers, operations research methods, mathematical models etc. to analyse data an draw effective conclusions by making use of formal models.

Decision support systems bay be recognized by their following characteristics:

- They tend to be aimed at the less well structured, under-specified problems that upper-level managers typically face.
- They attempt to combine the use of models or analytic techniques with traditional data access and retrieval functions.
- They specifically focus on features that make them easy to use by noncomputer people in an interactive mode; they are user initiated and controlled.
- They emphasize flexibility and adaptability to accomodate changes in the environment and in the decision-making approach of the user.

The decision making which occurs at several levels must often be coordinated. Therefore an important dimension of decision support is the *communucation and coordination between decision makers across organizational levels as well as at the same level.*

DSS differ from the Management Information Systems (MIS) and Transaction Processing Systems (TPS) in the sense that they use the data handled at TPS and reports or inquiries developed by MIS, combine the results and using some models or methods help the user take implementable decisions.

Capabilities that good DSS should possess can be analyzed through some performance objectives. DSS are usually very much dependent on the task, organizational environment and decision maker(s) involved. Therefore a specific decision support system can not be required to satisfy all of the performance measures listed below:

1. DSS should support semi-structured and unstructured decisions as well as structured ones.

   A decision may be unstructured because of its constraints, lack of knowledge, large search space, multidimensionality, compleq interrelationships, nonquantifiable data, dynamic environment and unstructured decisions are of greatest concern to decision makers. (Gorry and Scott, 6).

2. DSS should be developed in such a way to be used at all levels assisting in integration between the levels whenever appropriate, *decision levels* can be classified as the following (Anthony, 2):

   i. Strategic Planning; related to setting policies, choosing objectives and selecting resources.

   ii. Management Control; related to assuring effectiveness in acquisition and use of resources.

iii. Operational Control; related to assuring effectiveness in performing operations.

iv. Operational Performance; related to performing the operations.

3. DSS should support the communication between decision-makers so as to support interdepending decision-making.

Decision types are classified as (Hackathorn and Keen, 8):

i. independent: a decision maker has full responsibility and authority to make a complete implementable decision,

ii. sequential interdependent: he makes part of a decision which is passed on to someone else,

iii. pooled interdependent: the decision must result from negotiation and interaction among decision-makers.

4. DSS should support all phases of the decision-making processing.

A popular model of decision-making with four steps follows (Simon,18):

i. Intelligence: Finding out the situations where a decision is required. Past data and necessary information is analysed to identify problems.

ii. Design: Creating possible alternatives by clearly identifying problems; developing solution methods and testing the results for validity and feasibility.

iii. Choice: From the alternatives above, selecting the best suitable one to the analysed situation.

iv. Implementation: Investigating for the decision made above.

5. DSS should support a variety of decision-making processes.

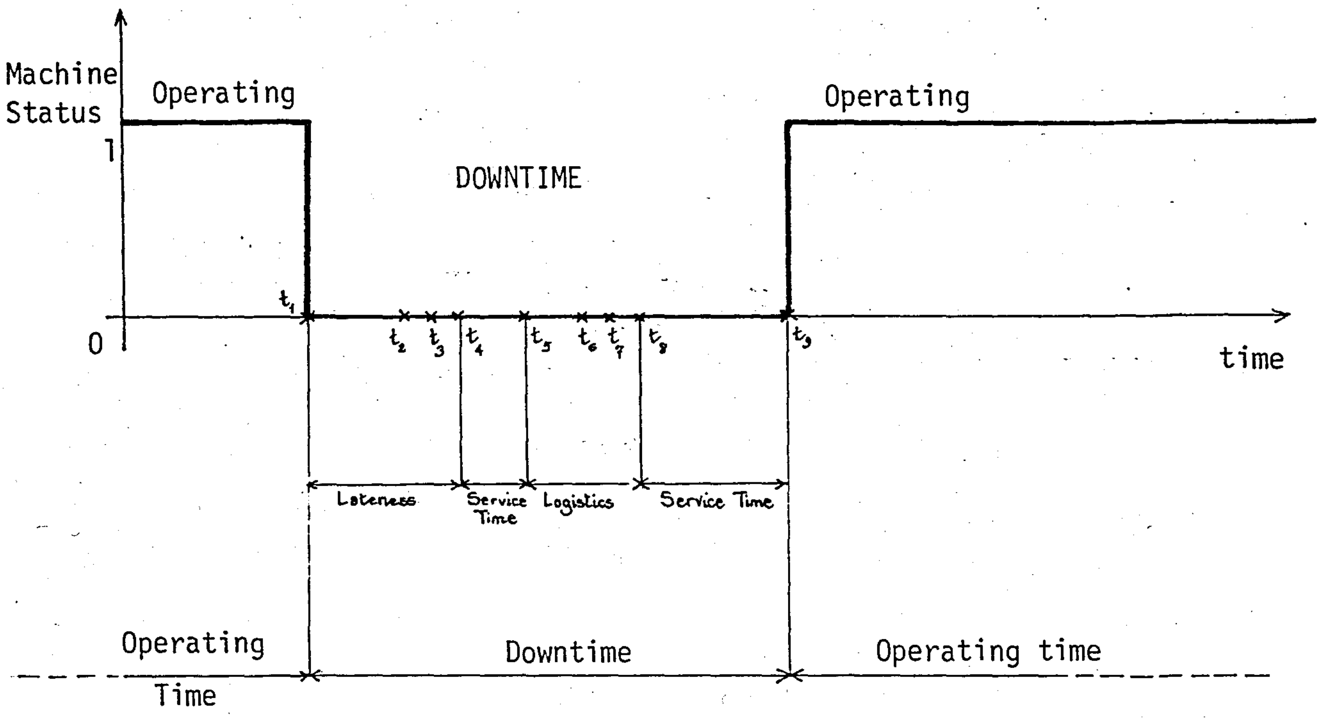
   It should be kept in mind that a specific decision may be of a different type, in different organizations, at different times, for different decision-makers. Therefore DSS should be flexible, friendly and provide a variety of options, so that the user's style should become irrelevant.

6. DSS should be easy to use and user controlled, i.e.: decision-makers can enter and use DSS at any point in the decision process. DSS should be constructed in a way to be used by people who may often be unfamiliar with computers.

The above listed objectives, as a whole, represent a set of capabilities that determine the value of the DSS concept from the user's point of view. Of course, depending upon the task required from it, a decision support system may not satisfy all of the performance measures listed above.

1.3 Preliminary Concepts

A machine status, in any production line, can be represented as in Figure 1.2.



1 = Machine up

0 = Machine down

t_1 - Failure occurence, Machine down

t_2 - Report to Maintenance Department

t_3 - Assignment of Repair Crew

t_4 - Trouble Shooting Started

t_5 - Failure diagnosis made

t_6 - Procurement of Spare Parts, ordered if necessary

t_7 - Order replenishment

t_8 - Repair restarted.

t_9 - Repair completed, Machine is up again

Figure 1.2 Description of machine status when an unexpected failure is encountered.

In Figure 1.2, the first interval t_1-t_2, represents the time elapsed since the failure of the machine to the time when the maintenance department is informed about the failure. The second interval stands for the time that the maintenance department take action, form the repair team and schedule the repair or replacement. Interval t_4-t_5 shows the time to find out the reason or failed part. Interval t_5-t_8 represents the time for spare parts procurement or finding it from the stock. In instances when the failure does not need a spare for repair, interval t_5- t_8 is said to be zero. Finally, interval t_8-t_9 is the actual repair time of the machine, which includes any tests or adjustments time if necessary.

The time elapsed from the beginning of failure, to the informing time of maintenance department is called the lateness, and the interval from that time to reoperation time, is called as the service time.

Lateness and service time together, constitute the downtime of the machine, whose relative length, directly influences the effectiveness of the maintenance department.

We can now define the machine availability as:

$$\text{Availability} = \frac{\text{Operating time}}{\text{Operating time} + \text{Downtime}}$$

The effectiveness can be improved by increasing availability of each particular machine, hence decreasing the downtime.

In Figure 1.2, it is worth to note that the downtime is mainly composed of four time segments, namely lateness, service time to find out failure reason, logistics and actual repair time.

If we consider a planned replacement policy, lateness and logistics times will be zero, and the downtime will only be composed of a single service time. Therefore the availability of the machine will be higher, because the numerator in the above equation will be close to the denominator.

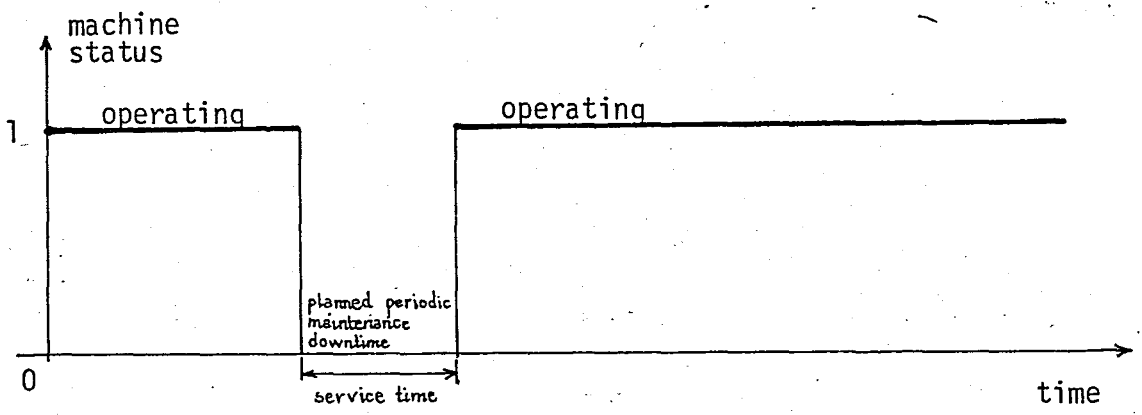


Figure 1.3 Description of machine status in case of planned maintenance activity.

In planned replacement models, a unit is replaced either at some prespecified time, or at time of failure, whichever occurs first.

It is especially meaningful to try a planned replacement policy, when the items in the system have the aging characteristics. This means that the hazard rate, or instanteneous failure rate, shows an increasing failure probability as the time increases.

In production systems, minimizing the production time losses due to unexpected failures, and realization of repair in case of failure in minimum amount of time and most economically are directly related to how effective the maintenance planning is done. This planning aims to reach the desired availability limits of production units with minimum cost and in a most reliable way.

As mentioned earlier, the availability of a machine is affected mostly by two factors:

i. reliability function of the machine, i.e. how often the failures occur?

ii. Delay of machine repair due to unexpected failures.

The main aim of maintenance planning is to increase the availability of machines which, in turn, improves the efficiency of the production system. Correct decisions are to be taken for the following subjects in order to minimize costs and to increase the efficiency and profitability. Maintenance manager encounters decision problems related to

a) planned maintenance of machinery before a failure occurs,

b) in case of failure, realization of repair effectively, in minimum time,

c) keeping the necessary maintenance personnel,

d) supply of necessary spare parts when needed.

In a modelling study of a maintenance system, it is important that the objective function is a correct representation of the real system. Furthermore the assumptions should be made carefully.

Usually, the total cost is the sum of two cost functions slopes have different sign with respect to the planned replacement time:

i. Cost of downtime due to failure,

ii. Cost of planned maintenance

as shown in Figure 1.4.

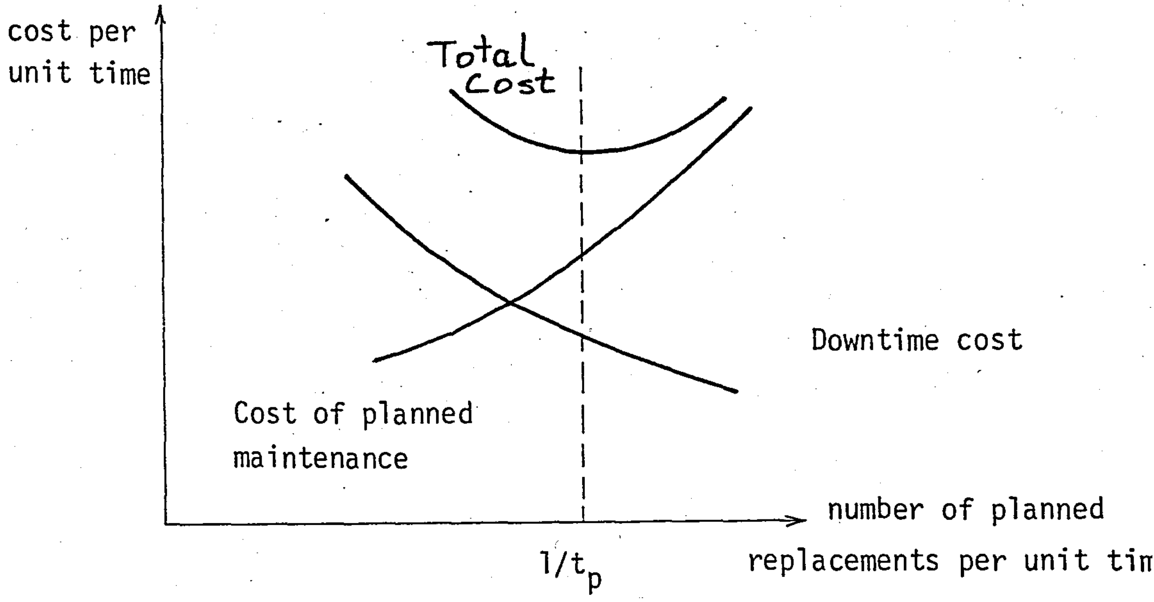


Figure 1.4 Total maintenance cost function

Note in Figure 1.4 that $1/t_p$ represents the optimal number of planned replacements per unit time.

An important statistical characteristic of equipment used in replacement studies should also be mentioned here, before proceeding any further. It is related to the failure rate. Let us suppose that a test where a large number of identical components are put into operation and the time to failure of each component is noted. An estimate of the failure rate of a component at any point in time may be thought of as the ratio of a number of items which were operational at the start of the interval. Thus the failure rate of component at time t is the probability that the component will fail in the next time interval given that it is good at the start of the interval, i.e. it is a conditional probability.

Specifically, letting $h(t)\Delta t$ be the probability that an item fails during a short interval Δt, given that it has survived upto time t, then this may be written as

P(A/B) = probability of A occuring once it is known that B has occured

$= h(t)\Delta t$

where

A is the event "failure occurs in interval Δt".

B is the event "no failure has occured up to time t".

The conditional probability can be expressed as

$$P(A/B) = P(A \text{ and } B)/P(B)$$

$$= \int_t^{t+\Delta t} f(t) / \int_t^{\infty} f(t)dt$$

where f(t) denotes the probability density function. If the cumulative distribution function is designated as F(t).

Then, the failure rate in interval Δt is

$$h(t)\Delta t = \frac{\int_t^{t+\Delta t} f(t)dt}{\int_t^{\infty} f(t)dt} - \frac{F(t+\Delta t) - F(t)}{1 - F(t)}$$

Dividing both numerator and denominator of the above equation by t, and letting t go to zero, yields

$$h(t) = \frac{f(t)}{1 - F(t)}$$

which is the instantaneous failure rate.

For some lifetime distributions like hyper exponential, the failure rate decreases with respect to time. This may be interpreted as an improvement in the equipment with time and may be the case with equipment which requires small adjustments after an overhaul or replacement to get it completely operational. Such situations are called Decreasing Failure Rate (DFR). When the failure rate increases, such as for the normal distribution, this indicates an ageing or wear-out effect. Such situations are called Increasing Failure Rate (IFR). With the negative exponential distribution the failure rate is constant (Constant Failure Rate). It is to be noted that a periodic replacement activity should only be carried out when the failure rate is increasing. Figure 1.5 distinguishes the three failure rates mentioned.

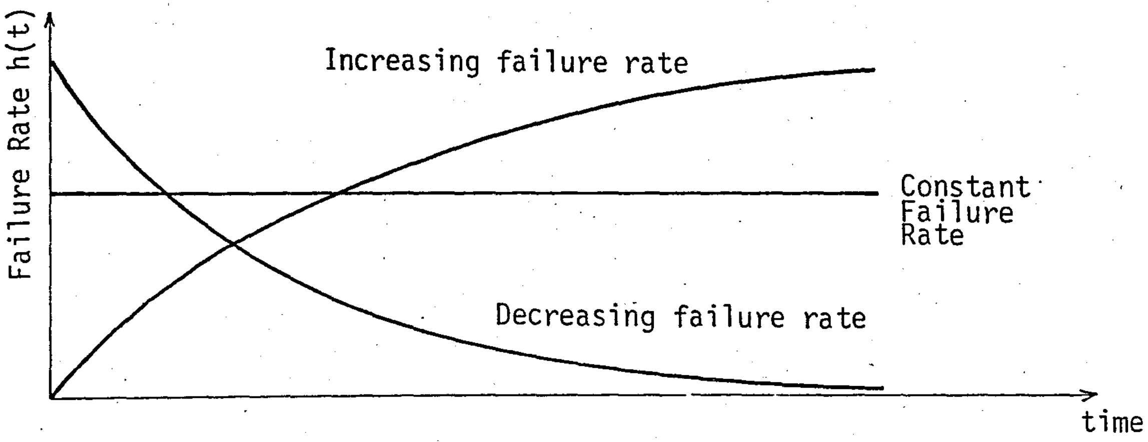


Figure 1.5 Representation of failure rate function

Especially in factories where the output rate is high, the repair cost incurred at failure, is usually much larger than the planned replacement cost. Therefore, preventive replacement policies are suitable for such systems whenever the components have the aging characteristics, i.e. whenever the failure rate h(t) is IFR.

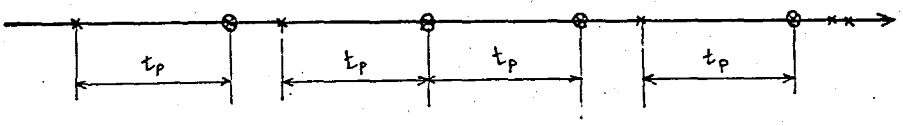
Two major replacement policies will be discussed in the remaining of this section. These are age replacement and block replacement policies. Because of machine characteristics of Bekoteknik for which this study is undertaken, the age replacement policy seems to be more appropriate. The main reasons for selecting this policy can be listed as:

- Existence of few identical machines,
- significant differences in periodic replacement times of components,
- existence of very few machines which have more than one identical components in them.

1.4 Overview of Preventive Replacement Policies

1.4.1 Age Replacement Policy

A unit is either replaced at a specific age t_p or on failure, whichever comes earlier. This is shown in Figure 1.6.



× : Failure Replacement

⊗ : Planned Replacement

Figure 1.6 Age replacement policy

For any period t_p, the expected total replacement cost per unit time can be expressed as:

$$E[\text{Unit time cost}] = \frac{E[\text{Cost during replacement period } t_p]}{E[\text{Replacement time}]}$$

Employing the

$f(t)$: probability density function (pdf)

$F(t)$: cumulative distribution function (cdf)

$R(t)$: reliability function, $R(t) = 1 - F(t)$

$h(t)$: hazard rate (or instantaneous failure rate), $h(t)=f(t)/R(t)$

C_p : cost of planned replacement

C_f : cost of failure replacement

t_p : replacement period

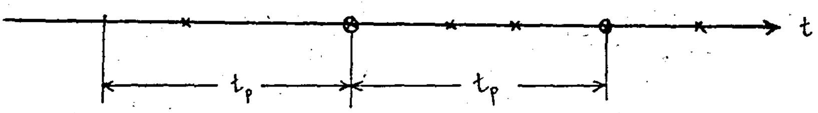
the expected cost per unit time can be defined as

$$\frac{C_f\ F(t_p) + C_p\ R(t_p)}{t_p\ R(t_p) + \int_0^{\tau} f(\tau)d}$$

To find the optimal value of t_p, derivative of $E[C]$ is set equal to zero and is solved for t_p.

1.4.2 Block Replacement Policy

A unit is replaced at fixed interval of times t_p, whatever its status at that time. In other words, even if the unit is replaced just before its replacement time t_p, it is replaced again, at time t_p. This replacement policy of fixed time interval is



× : failure replacement

⊗ : planned replacement

Figure 1.7 Block replacement policy

In this case

$$E[\text{Unit time cost}] = \frac{C_p + C_f\ F(t_p)}{t_p}$$

and we should minimize this, to find the optimal replacement time of part, t_p.

1.5 Scope and Organization

In this study it is mainly aimed to devise a computer program to aid maintenance planner in taking decisions about the maintenance activities. Upto 100 machines and 500 spare parts can be catalogued and all information about them can be stored in these files. Submodels built in the program serve to the following purposes:

a) Scheduling of machines whose maintenance is to be done.

b) Finding optimal replacement times of parts by analysing past data about them.

c) Ordering of spare pards, reorder quantities and minimum stock levels.

d) Preparing periodic reports about the maintenance activities of maintenance department.

Details of the above submodels are explained in Chapter II. The organization of the interactive computer program will be illustrated in Chapter III. The conclusions of the study and the suggestions for further study will be presented in the subsequent chapter. Description of the production system and current maintenance practices are reviewed in the following sections.

1.6 Description of the Production System

Bekoteknik is a producer of consumer electronics which manufacture more than 300,000 TV sets, 300,000 electronic cash registers, 50,000 video, 200,000 radio and music sets per year. Although approximately 27 different models of these products are manufactured, the production technology does not significantly differ among them.

An electronic product, can be thought of as a group of modules which work consistently with each other. Although these components serve different functions, the production stages they go through are similar. A module is composed of a printed circuit board (PCB) and all electronic components installed on it. The pcb plays a role of carrying the electronic components and electrically connecting them one to another. Usually the main module is called as the chasis. At the simple audio products all groups are condensed on the main chasis.

Stages of the production process can be illustrated as follows:

1. Input quality control,
2. Pre-assembly preparation of components, preparation of semi-finished products,

3. Preparation of printed circuit boards,
4. Insertion of components on the pcb's, foam fluxer, pre-heating and wave soldering applications,
5. Functional control of modules, adjustments, allignments.

All modules and seperately prepared mechanical components are combined at the final assembly line and a product is thus produced. The details of TV production line is portrayed in Figure 1.8. PCB manufacturing is also depicted in Figure 1.9.

1.7 Description of the Current Maintenance System

The maintenance activities in the selected factory are conducted by the maintenance engineer, who works under the supervision of the production manager. Each month, he selects the machines whose maintenance activities are to be done. He assigns the maintenance personnel to perform the maintenance.

The machines in the factory are classified into four categories, with respect to their maintenance periods as 12, 6, 2 and 1 month respectively. These maintenance periods are determined in the following manner:

Decisions on periodic maintenance are undertaken in the light of a scoring mechanism. For every machine, different people involved in the production system give subjective points over ten, considering the criticality of the machine in the overall production. These points are given by: production manager, industrial engineering department, maintenance engineer and the chief of the shop where the machine is. If the mean of these points is 7 or over, it is decided that it is rational to do periodic maintenance for that machine, otherwise repair

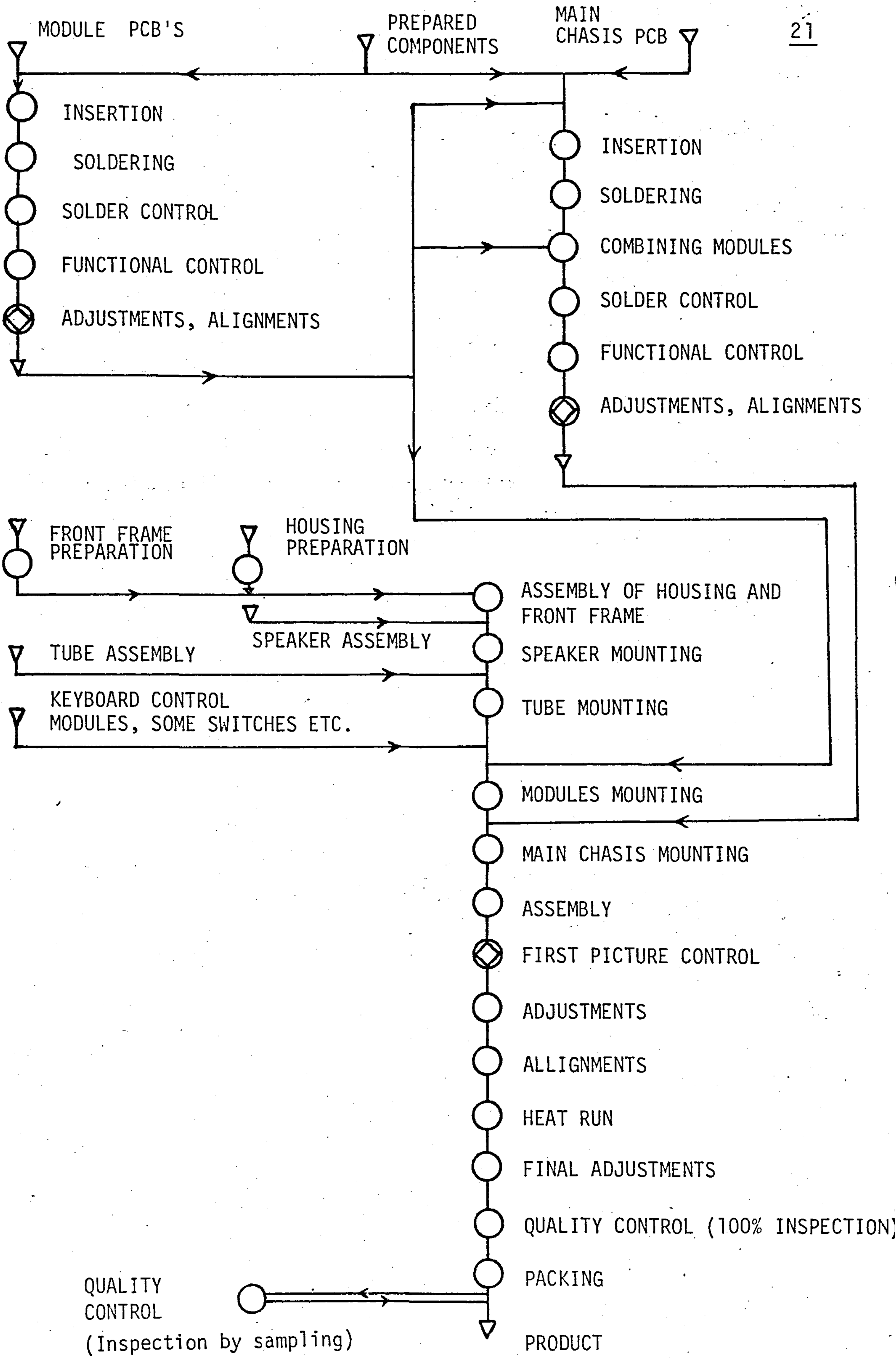


Figure 1.8 TV Production Flowchart

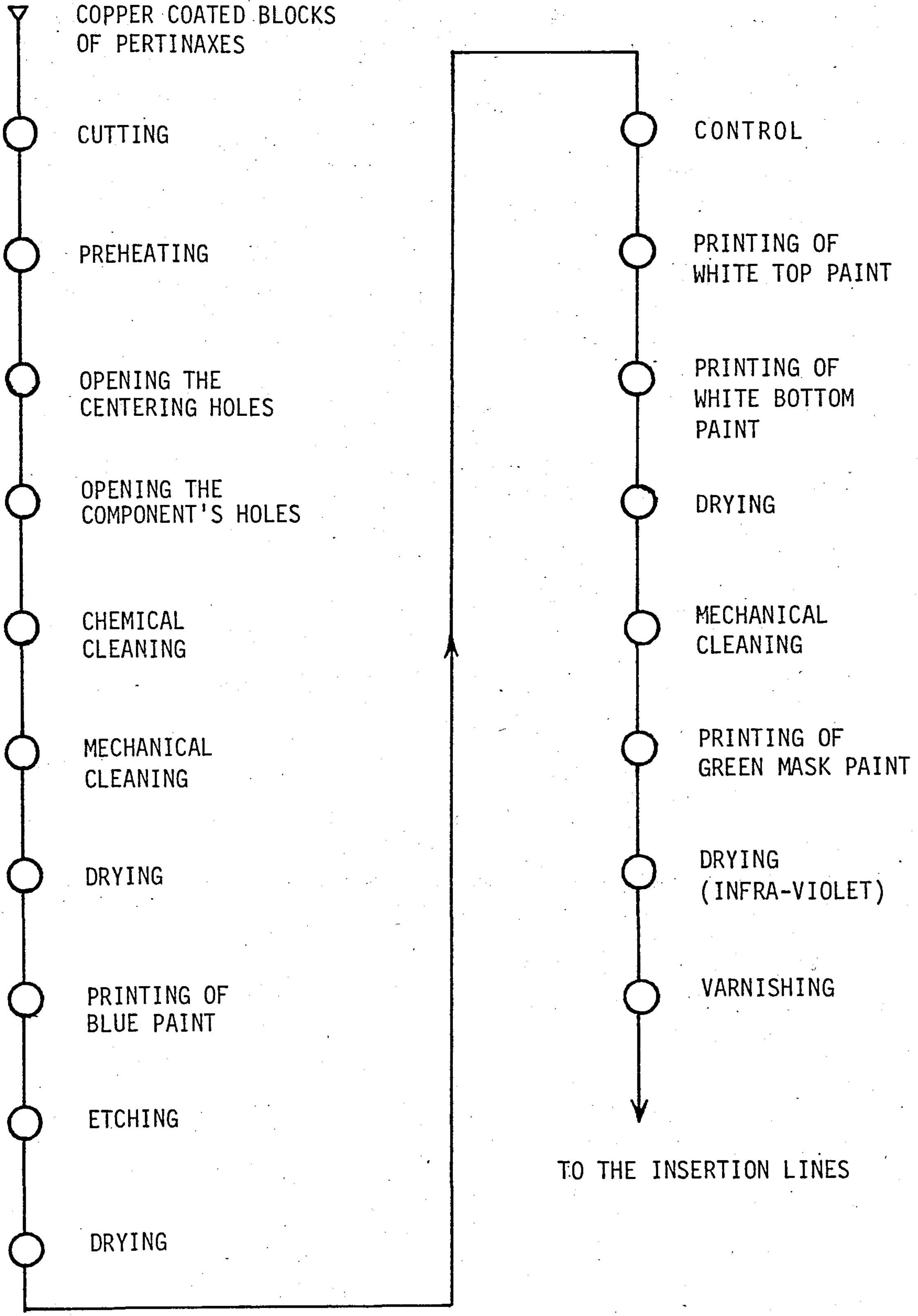


Figure 1.9 PCB Preparation Flowchart

on failure policy is applied to the machine. If periodic replacement is chosen, then a suitable periodic replacement time is selected and applied.

Every month, priority for maintenance is given to machinery with respect to the cost due to unit production time loss. Overdue schedules from the previous months are also taken into consideration in assigning priorities.

The duties of maintenance department in Bekoteknik are not restricted with unexpected failures or planned (preventive) replacement activities. The additional job requests originating from the production department such as constructing small conveyors, kinds of jigs, apparatus, tables, racks etc. are tried to be met by the maintanence department. Due to this current extra workload, once the maintenance engineer, who is responsible for preparing the maintenance schedules, can not assign jobs satisfactorily, undesired delays may cause significant production time losses.

II. MODELS USED IN THE PROGRAM

2.1 Machine Scheduling

For any week, the machines whose periodic replacement activities are to be done are selected according to the unit cost incurred upon failure of those machines. When this function of the program is to be used, the program asks the user the wee k for which scheduling is wanted to be performed, i.e. current week's number (the weeks of the year is supposed to be numbered from 1 -first week of January- to 52 -last week of December).

The scheduling within the program is performed at the subprogram "procedure macsec'. In this procedure, two sequential operations of selection and sorting are performed, and the results of these two operations are displayed on the same screen which is vertically seperated in two.

First, all machines present in the files are checked. If a machine's last date of maintenance plus its periodic replacement time equals or exceeds the current date, that machine is selected as a candidate. Then, the selected machines are considered and sorted in descending order with respect to their unit time costs. The term 'unit time cost' is used to represent the cost which will be incurred if a machine is not operational for one minute.

When the sorting is finished, the machines are taken one by one and if the maintenance activity of that machine can be performed with the current available labor hours, that machine is assigned and required work-hour is subtracted from the current available man-hour. At the beginning, available manhours is determined from the number of workers and qualified personnel, and as the machines are assigned, it continuously decreases. Machine scheduling activity is illustrated in Figure 2.1.

For each week, maintenance delay would contribute an increase in the associated production loss cost of that machine by dividing it with a factor, $(\text{availability})^n$, where n denotes number of weeks maintenance is postponed. This would increase its priority for maintenance in the following weeks. Thus, the machine scheduling is performed according to the index $(\text{production loss cost})/(\text{availability})^n$.

If a machine requires more manpower than the current available amount, then it is skipped and next one is assigned. This procedure continues until no more machines can be assigned.

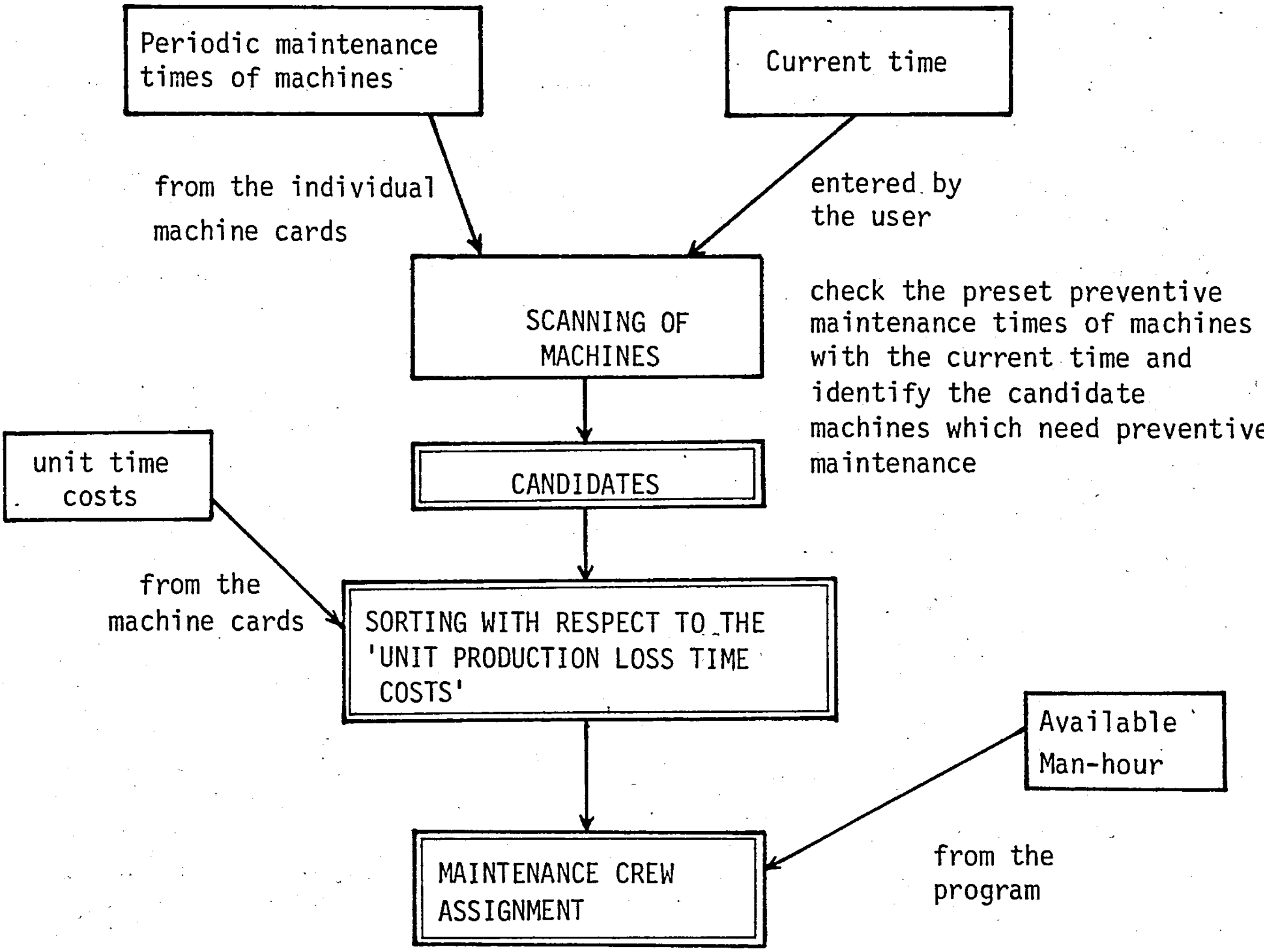

Periodic maintenance times of machines
Current time
from the individual machine cards
entered by the user
SCANNING OF MACHINES
check the preset preventive maintenance times of machines with the current time and identify the candidate machines which need preventive maintenance
CANDIDATES
unit time costs
from the machine cards
SORTING WITH RESPECT TO THE 'UNIT PRODUCTION LOSS TIME COSTS'
Available Man-hour
from the program
MAINTENANCE CREW ASSIGNMENT

Figure 2.1 Maintenance scheduling flowchart performed by the program

At the end, in the display two tables are seen. At left, the machines whose periodic replacement activities are to be performed; at right, those whose maintenance can be realized with the current available labor.

The maintenance activities for the scheduled machines are not assumed to be automatically carried out. This is due to the fact that planned and real life situations may differ, and for some reason the maintenance of some machine could not be fullfilled. For example, a machine with a considerable unit time cost might have failed and, of course, the repair of that machine will be more crucial. In fact, it is always more reasonable to assign failed machines before the scheduled ones in practice. Therefore, when the maintenance of a machine is done, it should seperately be entered into the program.

2.2 Determination of Optimal Replacement Times of Spare Parts

The maintenance activities performed are entered only to the machine files and necessary information about the spare parts are automatically transferred to the spare part files. Among these information, the replacement dates are analysed. Those replacements which occured as a result of a failure are considered and the times elapsed until the failures are taken as data inputs of the analyses.

The Weibull distribution is one of the most widely used lifetime distribution. Therefore this one is selected to be used in this study for analysing past failure data and finding optimal replacement times of parts.

An illustration of how Weibull distribution arises in practice is presented in Kapur (11). The model is related to the machines subject to voltage fluctuations. The shocks received may eventually lead into machine failure.

Following assumptions are made:

1. The failures occur completely at random and independently. That is, the occurence of a failure does not provide any information as to when the next failure will occur.

2. The probability of an occurence during any interval of time Δt is proportional to the length of the interval, the constant of proportionality being λ. That is P[one failure during Δt] = $\lambda \Delta t$.

3. A function p(t) denotes the probability that the system will fail at time t.

If R(t) is defined to be the probability that the system is surviving at the time t,then for a small interval of time Δt, we have

$$R(t\pm\Delta t) = P \begin{array}{l} \text{(system surviving at time t} \cap \text{no failures occur} \\ \text{during } \Delta t) \cup \text{(system surviving at time t} \cap \text{one} \\ \text{failure occurs)} \cup \ldots . \end{array}$$

$$= R(t)(1-\lambda\Delta t) + R(t)\lambda\Delta t[1-p(t)] + R(t)(\lambda\Delta t)^2[1-p(t)]^2 + \ldots$$

rearranging and dividing by Δt yields

$$\frac{R(t+\Delta t) - R(t)}{\Delta t} = -\lambda R(t)p(t) + (\text{terms of the order } (\Delta t)^n, n \geq 1)$$

taking the limit as $\Delta t \rightarrow 0$, we get

$$\frac{dR(t)}{dt} = - \lambda R(t)\ p(t)$$

Rearranging and solving for R(t) gives

$$\int \frac{dR(t)}{R(t)} = - \int \lambda p(t)\ dt$$

$$\ln R(t) = - \int \lambda p(t)\ dt$$

$$R(t) = \exp \{- \int \lambda p(t)\ dt\}$$

define $$P(t) = \int_0^t p(\tau)\ d\tau$$

then $$R(t) = \exp\{ -\lambda P(t)\}$$

and $$f(t) = \lambda p(t) \exp\{ -\lambda P(t)\} , \quad t \geq 0$$

We can now observe how a distribution is selected.
If p(t) is a constant, f(t) would be exponential.
If $\lambda p(t)$ is a power function of t given by $(\beta/\theta)(t/\theta)^{\beta-1}$, then f(t) is Weibull. In this case, as time passes, the system will deteriorate if $\beta > 1$; that is, a failure will have a greater chance of occuring with increasing age.

A linear function with $\beta = 2$ would give rise to the Rayleigh distribution, which is a special case of Weibull distribution.

The weibull failure density function is

$$f(t) = \frac{\beta}{\theta^{\beta}} t^{\beta-1} \exp\{-(t/\theta)^{\beta}\} \quad t \geq 0$$

where β known as the shape parameter, and θ known as the scale parameter, both always positive.

From the above, cumulative density function is found to be

$$F(t) = 1 - \exp\{-(t/\theta)^{\beta}\}$$

and the reliability function

$$R(t) = 1 - F(t) = \exp\{-(t/\theta)^{\beta}\}$$

and hence, the hazard function (or instantaneous failure rate)

$$h(t) = \frac{f(t)}{R(t)} = \beta \frac{t^{\beta-1}}{\theta^{\beta}}$$

Following figures illustrate various forms of failure density function f(t), reliability function R(t), and hazard function h(t) respectively with changing β.

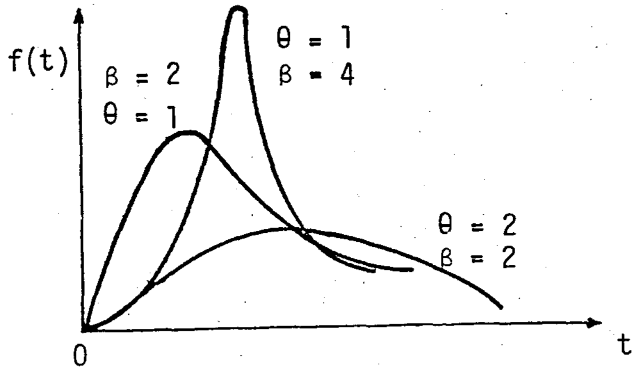


Figure 2.2 The failure density function for the Weibull distribution

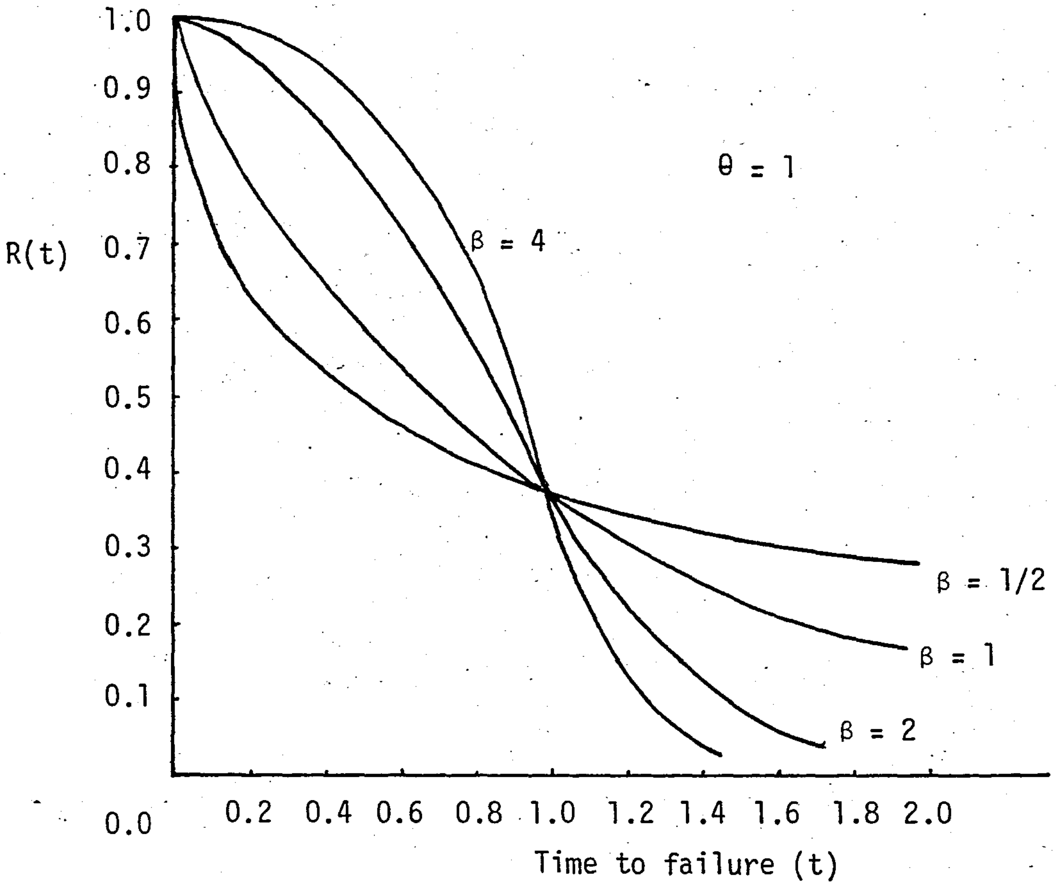


Figure 2.3 The weibull reliability function

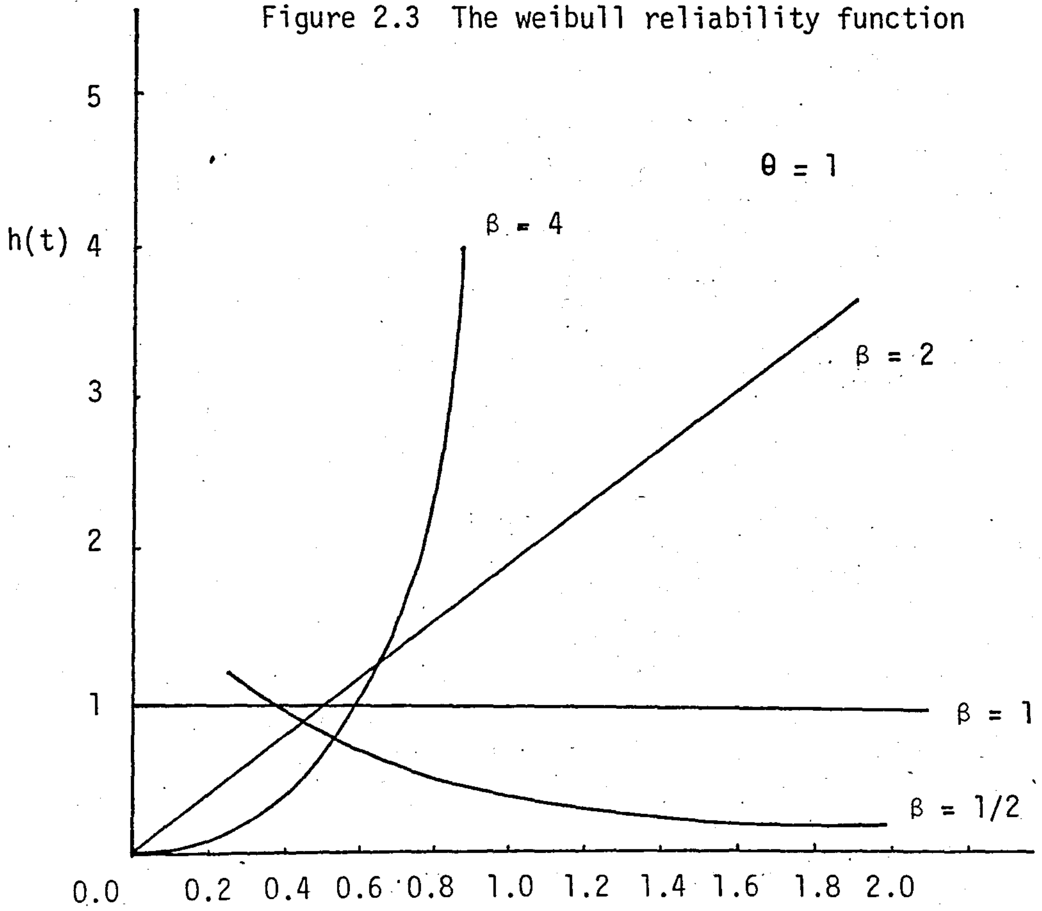


Figure 2.4 The weibull hazard function

It is worth to note that the hazard function is decreasing for $\beta < 1$, increasing for $\beta > 1$, and constant when β is exactly one.

In particular, when we consider the weibull distribution with $\beta = 1$, the expression for the reliability function reduces to

$$R(t) = \exp \{-t/\theta\}$$

and the hazard function reduces to

$$h(t) = 1/\theta \quad , \text{ a constant.}$$

When $\beta \leq 1$, this means that the probability that the part fails does not increase with time. So whenever β is found to be less than or eqaul to one, the program warns the user that it is not meaningful to apply periodic replacement to that part.

In order to determine the shape and scale parameters of the Weibull lifetime model, Mennon estimators are used (Mennon, 16). The shape parameter β, is estimated as:

$$\beta = 1/\{ \frac{6}{\pi^2} \left[\sum_{i=1}^{n} (\ln t_i)^2 - (\sum_{i=1}^{n} \ln t_i)^2/n \right] /(n-1)\}^{1/2}$$

where t_i's are the failure times of the component analysed and n is the total number of failures observed.

The scale parameter, θ is estimated by employing:

$$\theta = \exp \{ \frac{1}{n} \sum_{i=1}^{n} \ln t_i - 0.577/\beta\}$$

Once the paratemetrs are found, the optimal replacement time t_p is found by minimizing the expected total cost which is:

$$E[\text{Total Cost}] = \frac{C_f\ F(t_p) + C_p\ R(t_p)}{\int_0^{t_p} \tau\ f(\tau)d\tau + t_p\ R(t_p)}$$

where : C_p = planned replacement cost (material used ± labor)

C_f = failure replacement cost (C_p + cost incurred due to production time losses)

The integral in the above equation is approximated by a summation in the calculations, as follows:

$$\int_0^{t_p} \tau f(\tau)\ d\tau = \Sigma\ t_i\ f(t_i)\ \Delta t$$

where

$$\Delta t = \frac{t_p}{n} \quad , n = \text{number of time intervals,}$$

$$t_i = t_{i-1} + \Delta t \quad , \quad t_1 = \Delta t$$

In the expected total cost formulation:

$$F(t) = 1 - \exp\ \{-(t/\theta)^{\beta}\}$$

$$R(t) = \exp\ \{-(t/\theta)^{\beta}\}$$

$$f(t) = \frac{\beta t^{\beta-1}}{\theta^{\beta}}\ \exp\{-(t/\theta)^{\beta}\}$$

and the formulation becomes:

$$E[\text{Cost}] = \frac{C_f(1 - \exp(-(t/\theta)^{\beta})) + C_p \exp(-(t/\theta)^{\beta})}{\frac{\beta}{\theta^{\beta}} \sum_{i=1}^{n} [t_i^{\beta} \exp(-(t_i/\theta)^{\beta}) \Delta t] + t_p \exp(-(t_p/\theta)^{\beta})}$$

to find t_p we set

$$\frac{d}{dt}(E[\text{Cost}]) = 0$$

$$\left[\frac{C_f \beta}{\theta^{\beta}} t_p^{\beta-1} e^{-(t_p/\theta)^{\beta}} - \frac{C_p \beta}{\theta^{\beta}} t_p^{\beta-1} e^{-(t_p/\theta)^{\beta}} \right] \cdot$$

$$\left[\frac{\beta}{\theta^{\beta}} \sum_{i=1}^{n} t_i^{\beta} e^{-(t_i/\theta)^{\beta}} \Delta t + t_p e^{-(t_p/\theta)^{\beta}} \right] -$$

$$\left[c_f(1 - e^{-(t_p/\theta)^{\beta}}) + c_p e^{-(t_p/\theta)^{\beta}} \right] \cdot$$

$$\left[\frac{\beta}{\theta^{\beta}} t_p^{\beta} e^{-(t_p/\theta)^{\beta}} + e^{-(t_p/\theta)^{\beta}} - \frac{\beta t_p^{\beta}}{\theta^{\beta}} e^{-(t_p/\theta)^{\beta}} \right] = 0$$

simplifying the above, we get

$$e^{-(t_p/\theta)^{\beta}} \frac{\beta t_p^{\beta-1}}{\theta^{\beta}} (C_f - C_p) \left[\frac{\beta}{\theta^{\beta}} \sum_{i=1}^{n} t_i^{\beta} e^{-(t_i/\theta)^{\beta}} \Delta t \right.$$

$$\left. + t_p e^{-(t_p/\theta)^{\beta}} \right] = (e^{-(t_p/\theta)^{\beta}} (C_p - C_f) + C_f) e^{-(t_p/\theta)^{\beta}}$$

rearranging terms we get

$$e^{-(t/\theta)^{\beta}} = \frac{\frac{\beta t_p^{\beta-1}}{\theta^{\beta}} (C_f - C_p) \frac{\beta}{\theta^{\beta}} \sum_{i=1}^{n} t_i^{\beta} e^{-(t_i/\theta)^{\beta}} - C_f}{(1 + \frac{\beta t_p^{\beta}}{\theta^{\beta}}) (C_p - C_f)}$$

For any given t_p the right hand side of the above equation is found to be A , let's say.

Then

$$\exp(-(t/\theta)^{\beta}) = A$$

$$(t/\theta)^{\beta} = -\ln A \quad \rightarrow \quad t^{\beta} = -\theta^{\beta} \ln A$$

$$\ln t = \ln(-\theta^{\beta} \ln A) / \beta$$

$$t = \exp(\ln(-\theta^{\beta} \ln A)/\beta)$$

this is a recursive formula as $t = g(t)$. To find the optimal t_p we first assign an initial value for t and letting $t = g(t)$ we find another t. This iteration continues until two successive t's are found to be close enough with a small tolerance ε (this method of solving $t = g(t)$ is called the successive approximations method).

2.3 Finding the Optimal Order Quantity and Reorder Point of Spare Parts

In periodic replacement maintenance systems, it is extremely important to keep the necessary amount of spares at hand, ready to use. The inventory model used in this study is a deterministic single item model with static demand. Although in real life situations, the demands of spare parts are stochastic, in near ideal cases where unexpected failures are very rare, it is meaningful to assume a deterministic demanc which occur only due to planned replacements.

The general behavior of the deterministic, economic lot size models is shown in Figure 2.5 where

b = backlog level

T_p = Production period

I_{max} = Maximum inventory level

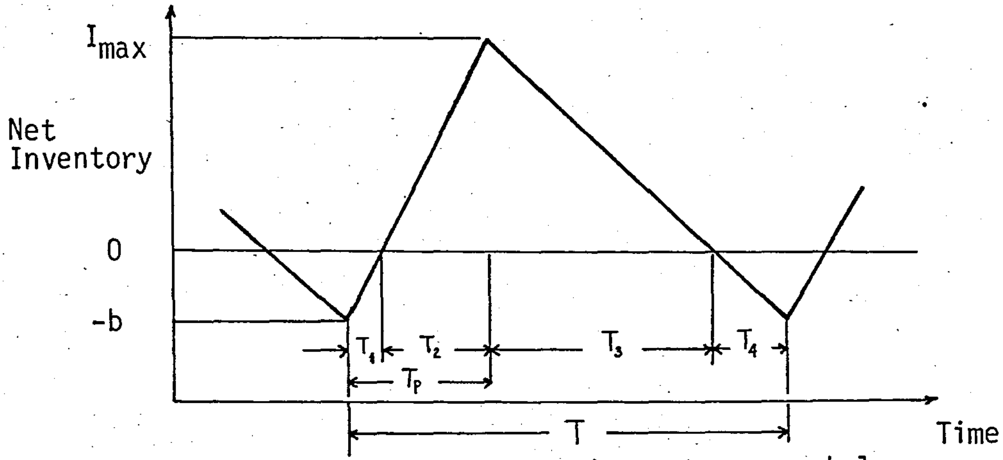


Figure 2.5 A cycle for inventory model

Because the order size and the demand rate are constant, orders are received at equally spaced intervals of length $T = Q/D$.

$$I_{max} = Q(1 - \frac{D}{p}) - b$$

In our case, the production rate is infinite and backorders are not allowed ($P \rightarrow \infty$ and $b = 0$), hence $I_{max} = Q$ and $T = T_3$ because $T_1 = T_2 = T_4 = 0$. Then the representation of the inventory state can be traced as in Figure 2.6.

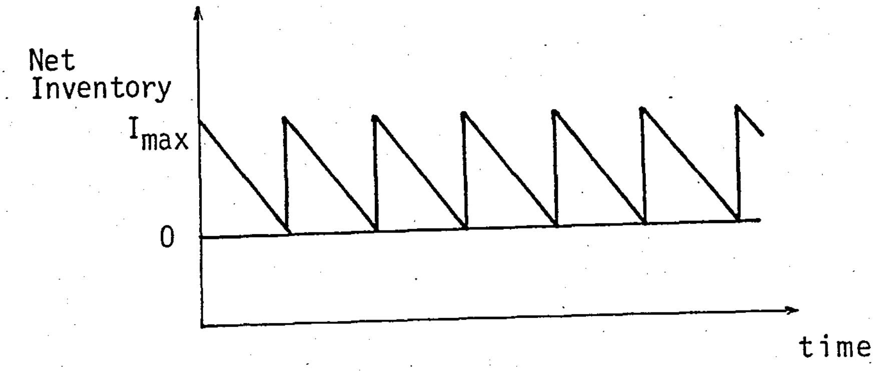


Figure 2.6 Cycle for inventory model with infinite input rate, no backlogging

The average inventory over a cycle becomes $\bar{I} = \frac{Q}{2}$. The average cost per cycle is the sum of procurement and inventory costs during one cycle.

Defining

A : fixed cost of an order

D : yearly demand

C : unit cost

i : annual inventory carrying cost rate

Q : order quantity

the average cost per cycle is

$$A + CQ + hT\bar{I}$$

By substituting the values, and multiplying by the number of cycles per year (D/Q) we obtain:

$$\text{Cost/Year} = \frac{AD}{Q} + CD + iC\ Q/2$$

which is minimized by setting the derivative with respect to Q to zero,

$$\frac{-AD}{Q^2} + \frac{iC}{2} = 0$$

and we get $$Q = \sqrt{\frac{2AD}{iC}}$$

On the other hand, reorder point in terms of net inventory is given by $r = \tau D - mQ$ where τ is the lead time between an order placement and order receival. m is the greatest integer less than or equal to τ/T and mQ stands for the on-order inventory immediately prior to ordering.

In this study, the yearly demand is found by dividing 52 (number of weeks a year) to the optimal replacement time of the component (in weeks) found as the result of the previous model..

Annual inventory carrying cost rate, i, is taken to be 50 percent. Before the model runs for a spare part, it asks the user to enter the fixed cost of placing and order, and if any change in unit cost and lead time occured, they can also be changed. While recording the performed maintenances into the program, if any spare part drops to the reorder point, the user is immediately warned on the screen.

III. DESCRIPTION OF THE PROGRAM

The program developed in this study can help the maintenance planner in many respects.

3.1 Information Files

First, the program acts as a data input, data output media, information storing file. In the machine file, 100 machines and 500 spare parts can be catalogued, and all necessary information about these machines and components can be written.

The following information about the machines are available:

- Name of the machine
- Code no
- Periodic Replacement Time (in weeks)
- Revision Time
- Cost per minute Lost (i.e. cost which will be incurred if that machine fails one minute)
- Availability of the machine up to last maintenance date
- Maintenance date
- Type of maintenance (a: for failure; p: for periodic)

- Labor used at the maintenance (number of electrical or mechanical technicians, and number of qualified personnel)
- Duration of failure and maintenance in minutes
- Spare parts used (for each machine, up to five spare parts can be reserved, and number of spare replaced is directly written to the corresponding column)
- Cost incurred for each specific maintenance activity.

The first five information are written on the machine record once, and do not change any more unless the machine is disposed. The other items are entered on the file whenever a maintenance is performed. One maintenance will be written on each line, and at the end of each line the cost associated with that maintenance is calculated automatically. The maintenance cost is composed of labor cost, cost of components replaced and cost of production loss due to maintenance. If the maintenance is performed on overtime no production loss cost is incurred, but labor cost changes according to overtime labor cost.

Although the cost associated with each maintenance activity is calculated automatically, the user has the option of changing it. The reason of allowing this option is that only labor, spare and production time loss costs are included in each cost figure. However some other costs might arise during a maintenance activity and it can be manually loaded to the program.

On each machine record, the figure at the top middle of the record represents the availability of the machine up to the last maintenance date.

At the bottom of the record, the mean values of labors, durations and costs are given. While scheduling the machines, these numbers are taken as labor requirement and expected duration of the machine.

Total time a machine interrupted from production is another important characteristic of the machine, and is represented at the end of failure duration column. A typical machine record is exemplified in Figure 3.1.

The spare part records, illustrated in Figure 3.2 contains the following information:

- Name
- Code No
- Minimum stock (reorder point)
- Unit (pieces, kilogram, meter, etc...)
- Unit price
- Order quantity
- Replanishment time (lead time between placing and receiving of an order)
- Number of machine to which that spare part belongs
- Current amount available
- Date of receiving an order
- Amount received
- Name of supplier
- Date of use of the component
- Type of usage (why the part is replaced, a : for failure; p: for periodic replacement)
- Suggested periodic replacement time of the component

* * MACHINE RECORD * *

Availability : 0.9981

NAME : 80 ton hidpres
CODE NO : 1827
FILE NO : 58
COST CENTER : 1

Periodic Replacement Time : 8
Revision Time : 52
Cost per Minute Lost (x100 TL) : 90

MAINTENANCE :

NO	DATE	TYPE	Electric Technician	Mechanic Technician	Senior Personnel	Stopping Duration	Maintenance Duration		Spares Used	Associated Cost	
							Normal time	Over time		Failure	Periodic
1	4	a	0	3	2	25	15	0	21 22	2660	0
2	12	a	1	3	1	30	30	0	21 24 25 28	2560	0
3	15	a	1	3	1	20	12	0	21	1923	0
4											
5											
6											
7											
8											
9											
10											
Mean Values :			1	3	1	25	19	0		2381	0
TOTAL DOWNTIME :			75 minutes								

Figure 3.1 A typical machine record

× × SPARE PART RECORD × ×

NAME : PIM Min. Stock Level: 4 Order Quantity: 9
CODE NO : 6827 Unit: Ad Lead time: 6
FILE NO : 21 Unit Price (100 TL): 120 Machine No: 58
Current Inventory: 11

NO	Arrival Date	Lot Size	Supplied From	Date Used	Type of Usage
1	0	5	Stock	4	a
2	10	9	Supplier 21	12	a
3				15	a
4					
5					
6					
7					
8					
9					
10					

Recommended Replacement Time : 17

Figure 3.2 A typical spare part record

The first eight items are written on the spare part record once, at the beginning. But during the program, minimum stock point and order quantity can be changed when the spare inventory model is used. Also unit price and lead time may be updated.

The current amount available at the inventory always updates itself whenever an order is received or the spare is used.

Date of use and type of usage are not always written on the spare record. Whenever that spare is shown to be replaced at its corresponding machine, these two columns are automatically updated on the spare record. However, when needed, they may also be written seperately on the record.

The first three columns on the record are written when an order is received.

Finally, suggested periodic replacement time of the component appears if the corresponding model is run for this spare part at least once.

3.2 Menu Presentation

The main menu of the program is portrayed as

```
S > To change system parameters
0 > Initialization
1 > Scheduling of Machinery
2 > To find file numbers of machines
3 > To find mean values of maintenance duration, labor
    requirement and cost
```

4 > To inspect spare part records

5 > To see list of existing machines

6 > To enter information on machine records

7 > To find periodic maintenance time of machines

8 > To get report about maintenance activities

9 > To find labor required for planned periodic maintenance activities

Q > To Quit program

3.2.1 Option S

Some system parameters may be changed by making use of this option. The following constants may be changed.

- Cost per minute of an electrical or mechanical technician
- Cost per minute of a qualified personnel
- Number of electric technicians
- Number of mechanical technicians
- Number of qualified personnels
- Workhours per week

3.2.2 Option O

Erases all information in the machine file and creates a new file ready to use. To protect the information a code is to be entered to make the option active. If a wrong code is entered the program returns to the main menu. The code of this program is 'Z'.

When the correct code is entered, the program warns the user that it is going to delete all information about the machines, and waits for 'E' to confirm or 'H' to return to the main menu.

3.2.3 Option 1

When this option is selected, the user is asked to enter the number of week for which he desires to schedule the maintenance activities. A number between 1 and 52 is entered, and as output two lists are observed. At the left part of the screen, machines whose periodic maintenance are to be done that week are listed. At the right part of the screen, the machines whose periodic replacement that can be done with the current labor available are listed in the order of decreasing unit cost. The detailed explanation of this option is given in section 2.1.

3.2.4 Option 2

The machines catalogued on the file can only be referred with their file numbers. However the user may not recognize the file number of a machine, but may remember the name of the machine he wants to see.

When this option is selected the user is asked to enter the name of that machine.

The output is either a machine's file number or a notice as 'No such machine exists in the file'. And the user is asked whether he or she wants to find file number of another machine. If 'e', 'E' or 'x' is pressed this option (2) is reselected automatically. Any other key returns to the main menu.

3.2.5 Option 3

On the machine records, at the bottom, a line shows the average number of electricians, mechanics, senior maintenance personnel and the average duration of failure and maintenance as well as average cost. Whenever option 3, in the main menu, is selected, this line is updated.

Only last ten maintenances are kept in the machine records, but the avergae number is average of the whole past data of the machine.

Whenever the program 'Bakım' is runned, this option is automatically performed before the program begins. But, it is also put as an option, into the main menu, to enable the user to get the averages when he enters new data.

3.2.6 Option 4

Every activity concerning the spare parts are activated by thes option. When it is selected a submenu for operations about the spare parts is observed. The spares submenu is portrayed as:

1 > Initialization of spare parts records
2 > To examine parts and to enter data on their records
3 > To find optimal replacement times of spare parts
4 > To find order quantity and reorder point of spare parts
5 > To see list of existing spare parts
6 > To initially open the spare part records
7 > To find monetary value of all spares currently present at the inventory
8 > To return to the main menu

3.2.6.1 Spares Option 1

Erases all information in the spare part file and creates a new file.

Same protection precautions are taken as option 0 of the main menu, and the code is again 'Z'.

After entering the code, an 'H' returns the user to the spares submenu and 'E' confirms to delete all past data.

3.2.6.2 Spares Option 2

Is used to see the spare parts records. When activated, asks the user which spare he or she wants to examine. A file number between 1 and 500 is entered and that spare's record appears on the screen.

For every spare, last 30 data are kept in its record, but they are displayed 10 by 10 on the screen. Only first 10 data appears first, pressing any key displays the next 10 data, and pressing any key once more brings the last 10 data about the spare

Now the user has three choices:

i) 1 is entered in order to write the receiving times, and quantities of spares on their records.
ii) 2 is entered in order to write the unexpected usages of spare parts. For example a spare may be broken when trying to install it in the machine, and a second one is installed. The one which is broken may be entered in this option.
iii) Any other key is entered to see another spare part record or return to the spares submenu.

3.2.6.3 Spares Option 3

This option calculates the optimal replacement times of spare parts, assuming that the failures come from a weibull distribution. The data input of the model are the time intervals up to a failure. If no failure or only one failure of a spare is observed, model stops giving a message as 'There is not sufficient information'. If at least 2, or more failures of the spare were observed, then the model finds shape and scale parameters of the weibull distribution associated with that spare.

If the shape parameter is less than or equal to 1, a message as 'It is not rational to apply periodic replacement for this spare part' appears and again the model stops.

However, if the shape parameter is greater than 1, a periodic replacement time, t_p, is found as optimal as described in section 2.2.

3.2.6.4 Spares Option 4

This option calculates the optimal order quantity and reorder point (minimum stock level) of spare parts.

First, it asks for the file number of the spare for which the user wants to determine order quantity. If previously a replacement time for that spare was calculated, that is used to find yearly demand. Otherwise, the program asks to enter the replacement time of that spare first, then continues.

It asks the cost of ordering, and also checks the unit price and replenishment time. If they are changed, it accepts the new values as the actual ones. With the above inputs, it uses the model described in section 2.3, and finds the order quantity and minimum level of inventory for the spare part. The user has the choice of accepting these two values as true for the future or rejecting it.

3.2.6.5 Spares Option 5

This option displays a list of all 500 spare parts' file numbers with corresponding code numbers.

The spare parts whose code numbers are zero in this list are to be understood as empty ones, i.e. no such spare part exists.

3.2.6.6 Spares Option 6

This option is used to create a record on the spare file for a new spare part. For each newly created spare part, the following information entered into the record.

Name
Code No
Minimum Stock
Unit
Unit Price (x100 TL)
Order quantity
Replenishment Time
Machine No
Current Inventory Level

Machine No is the file number of the machine in which that spare part is used.

3.2.6.7 Spares Option 7

In a periodic replacement maintenance policy, a number of spare parts have to be kept at the inventory. One may be interested in determining the amount invested in spare parts, any time. When this option is selected, the monetary value of all spares currently present at the inventory are calculated and displayed.

3.2.7 Option 5

This option displays a list of all 100 machines on the file with their code numbers versus file numbers. Zero code number means that the machine record is yet empty i.e. no machine is written on the record corresponding to that file number.

3.2.8 Option 6

This option is used to create a new record on the machine file for a new machine. For each newly created machine, following information entered into the record.

Name
Code No
Periodic Replacement Time
Revision Time
Cost per Minute Lost

Cost per minute lost is the cost which will be incurred to the company, if that machine fails one single minute. In determining this number, the user should be very careful because in factories where the output rate is high, the maintenance costs associated with the production losses are very high when compared with other costs. In this program, the production loss costs are completely determined from this value.

3.2.9 Option 7

This option is used to find the periodic maintenance times of the machines in the machine file. It is assumed that the spares of one machine are replaced at the same time, i.e. at the periodic maintenance time of the machine. This time is determined from the replacement times of the spares of the machine.

3.2.10 Option 8

This option is used to get reports for specific periods of time, about the activities of the maintenance department. First, the interval for which report is required is entered into the program. For that period, the following information is obtained:

Total work-hours available in that period
Time spent on maintenances on regular time
Time spent on maintenances on overtime
Spare time spent or time spent on other activities
Total production loss in hours
Production losses due to periodic maintenances (hours)
Production losses due to failure maintenances (hours)

* * * * * * * *

Cost of labor for normal work-hours
Cost of labor for overtime
Cost of total labor
Cost of spare parts replaced
Total direct costs
Cost of production losses due to periodic maintenances
Cost of production losses due to failure maintenances

TOTAL COST ....

Table 3.1 Outlook of a Report of Maintenance Department

In Table 3.1, the information above the stars are related with times spent on specific maintenance activities in hours. Below the stars, the figures represent the costs associated with the specific times.

3.2.11 Option 9

This option is used to find the required number of electricians, mechanics and senior maintenance personnel for the maintenance department assuming only periodic maintenances occur.

3.3 Application

The model developed is implemented as a pilot study in a production department of a factory. For the sake of illustration only a small section is considered. 25 machines and 70 spare parts are entered into the program. Each maintenance activity is recorded into the program for 26 weeks and monthly reports prepared for the first 6 months are presented in Appendix D.

In this application, cost centers in the machine records are numbered as follows:

1. PCB manufacturing shop
2. Machine shop
3. Painting shop
4. Manual TV PCB insertion line
5. TV Main Chasis adjustment line
6. TV final assembly line
7. Cable preparation shop
8. General machines

In these reports, current available labor and working conditions of the maintenance department are considered. That is eight electricians, seven mechanics and three senior personnel are entered into the program; weekly working hours are considered to be forty five hours. The effective time is taken as 90% of this amount after deducting the workers' personal time. Unit overtime labor costs are twice the normal labor costs. The production loss costs are calculated by summing the products of individual failure duration of machines with the unit production loss costs.

It is observed that the man-hours utilized for the maintenance activities is only a small portion of the available labor time. This is mostly due to the fact that there are plenty of other machines that the maintenance personnel give service to.The corresponding maintenance time for those machines are seen in the item stated as idle time and other activities.

Using the results of the prepared reports, some conclusions about the performance of maintenance department can be drawn. These reports can be supplemented with graphics. For this purpose, DrGraph working under CP/M 80 operating system is utilized to transform the tabulated cost figures into graphical form. The data stored in specified files can be easily retrieved as the data set for the figures. These figures are illustrated in Appendix E for the factory under study. It is observed that various graphical options are available for the user.

IV. CONCLUSIONS AND SUGGESTIONS

4.1 Conclusions

Maintenance planning model developed in this study is evaluated on the basis of capabilities expected from decision support systems (see section 1.2).

1. The problems concerning the maintenance planning in medium and large scale plants are often unstructured. Total production losses are hardly quantifiable due to complex interrelationships among the machines. Depending on the status of buffer stocks, a machine failure may cause production losses on other related machines. Certain predecessor machines may be blocked due to the buffer limitations. Also machines placed after the failed one may be starved due to unavailable material flow. The possibility of conflicting managerial objectives such as not allowing any overtime, keeping spare parts stock at a low level or to avoid unexpected failures as much as possible, makes the decision making process more difficult.

2. Both managerial and operational control level decisions are concerned in planning maintenance activities. While decisions concerning setting policies and choosing objectives are taken at upper levels of management; assuring effectiveness, efficient usage of resources and high availability of machinery in performing maintenance activities are left to the responsibility of the operational management.

3. While using the developed program and taking decisions, upper level decision-maker will use the results obtained by the maintenance manager. Hence the model will act as a communication media between them which allows the information flow.

4. Predetermined steps of decisions can directly be given to the model, and it can be used from that time on, at any level of decision-making process. It allows the decision-maker apply his own techniques and judgemental criterias.

5. The model is completely user oriented which makes it flexible and process-independent. Special attention is paid in designing the program in order to make it easy to use. The model, does not require the user to have any practice on computers.

As defined earlier, a decision support system is any system that makes some contribution to decision-making, and considering the above aspects, the model developed in this study can be classified as a decision support system. The model, even satisfies the conditions required by the restrictive definition of decision support systems, which is stated as interactive computer based systems that help decision-makers utilize data and models to solve unstructured problems.

But it would be more realistic to view this program, at somewhere between MIS and DSS, because only one single model is developed and provided for each purpose, and the validity of the models for the situations concerned are not checked.

By making use of the models developed in this study, a decision-maker can easily plan and apply every avtivity concerning maintenance department. In planned, periodic replacement maintenance policies, most important characteristics such as replacement times of components, periodic maintenance times of machines, scheduling machines for maintenance with the current labor available, order levels and order quantities of components can be determined with the model.

Reports prepared by the program for specific periods of time, may help to visualize the performance of the maintenance department in time. Some information which may be required for efficient managing such as the percentage of the overtimes, the ratio of cost due to failures to costs incurred by periodic replacements, invested amount on spares inventory, total production time loss costs are provided by the program.

4.2 Suggestions

For making efficient use of the proposed 'Maintenance Planning' program, first of all, a reliable past data about the machinery should be available. Otherwise, one should have to wait until enough data accumulates within the program.

When a spare part drops to the order level, it is good practice to check the optimal replacement time of the spare part first. This information can be used to project the yearly demand of the component in the inventory model, and more reliable results can be obtained.

A decision support system is said to be never complete. Many possible extensions to the program developed in this study are possible. A few of them will be mentioned here.

The program may be interfaced with a simulation model to create scenarios through which "What if" type of analysis can be carried out. The results of different scenarios may be realized by generating machine failure times. Thus, the advantages of preventive maintenance policies can be observed more effectively.

In an extended study, another module can be designed to specify the state of the job shop to be studied. Unit costs of production losses can be computed with the aid of this module, rather than entering those figures manually. This would also give an idea about the criticality of each machine.

An estimate for the expected number of failure replacements of machinery can also be added to the program. This would aid in maintenance personnel planning.

In scheduling the machines, ordering spares and determining optimal replacement times, other models can also be added to the program and results of different models can be compared in order to choose and investigate the most appropriate one.

APPENDIX A

LIMITATIONS OF THE PROGRAM

- Up to 100 machines and 500 spare parts can be stored in the program, and whenever a machine or component record is to be referred, one should enter the corresponding number.
- Maintenance durations and downtime durations are in minutes.
- Preventive maintenance period, revision period and suggested replacement time of spare parts are in weeks.
  Whenever needed, the dates also are entered in weeks. For example first week of February is entered as 5, second week of March is entered as 10, etc.
- Maintenance type is either 'a', if it is done due to an unexpected failure, or 'p', if it is done due to a periodic replacement.
- A maximum number of 5 spare parts can be entered for a maintenance activity.

APPENDIX B

NOTES ON PROGRAMMING ASPECTS

Turbo Pascal is used as the programming language. This "Maintenance Planning" model occupies a 39 Kbits of memory. It is developed using the CP/M 80 operating system. It is also modified and currently available under the MS-DOS operating system.

The program is run for the sample factory data on the NCR Decision Mate micro computer.

APPENDIX C - THE PROGRAM LIST

```
PROGRAM MAKINA (INPUT,OUTPUT);
 CONST
  MAXNOOFMACHINES=100; XA=10; ydkxa=30;
  MAXNOOFYDK=500; hi=0.5;
 TYPE
  PARCAISMI=STRING[15];
  MACHINENAME=STRING[15];
  SS=STRING[10]; CS=STRING[1];
  RG2=1..5;
  RANGE=1..10;  YDKRANGE=1..30;
  RG=1..MAXNOOFMACHINES;
  ARG=ARRAY[RG] OF INTEGER;
  AR=ARRAY[RANGE] OF INTEGER;
  MACHINE=RECORD
           NAME:MACHINENAME;  AVAILABILITY,TOPIMAK:REAL;
           MAINPERIOD,REVPERIOD,ORTEL,ORTMEK,ORTUS,MZA:BYTE;
           NUMBER,CODENO,COSTPMINLOST,ORTBAKSUR,ORTDURSUR,PORTCOST,FORTCOST,FAZLA:INTEGE
           MAINDATE,ELEKTRIKCI,MEKANIKCI,USTA,EMNIYET:ARRAY[RANGE] OF INTEGER;
           FAILUREDUR,MAINTDUR,FMMAINDUR,PARCACOST,FCOST,PCOST:AR;
           MAINTYPE:ARRAY[RANGE] OF CS; SPARESUSED:ARRAY[RANGE,RG2] OF INTEGER;
          END;
   PARCA=RECORD
          isim:parcaismi;
          ynumber,ycod,ucost,reptime,makno,ydkfazla:integer;
          rorderq,leadt,mnlvl,xza:byte;
          birim:string[5];
          tip:array[ydkrange] of cs;
          kimden:array[range] of ss;
          kullandate:array[ydkrange] of integer;
          geldate,gelmik:array[range] of integer;
         END;
   SABIT=RECORD  sabitler:AR;  END;
   MACHF=FILE OF MACHINE;
   ydkf=file of parca;
 VAR
   EB,CW:INTEGER; sabitfile:file of sabit;sabitrec:sabit;
   MACHINEFILE:MACHF; YDKFILE:YDKF; YDKREC:PARCA; MACHINEREC:MACHINE;
   I,VA,WORKWEEK,weekhour,UN,EN,MN,DIC,DUC:INTEGER; G,CH:CHAR;

procedure intoku(x,y,u:byte; var sayi:integer);
   var c,l,i:byte; result:integer; st:string[4];
   begin lowvideo; gotoxy(x,y);
         for l:=1 to u do write(' '); highvideo;
        gotoxy(x,y); c:=0;st:=''; ch:=' ';
        read(kbd,ch);
        while (ch=' ') and (c<u) do
            begin write(' '); c:=c+1; read(kbd,ch) end;
         while (c<=u) and (ord(ch)<>13) do
            begin
               case ord(ch) of
                  8  : if st<>'' then
                          begin write(ch); lowvideo; write(' ');
                                highvideo; write(ch);
                                delete(st,c,1); c:=c-1
                          end;
                  32 : if st='' then begin c:=c+1; write(ch)
                                    end;
                  48..57 : begin c:=c+1;
                                 if c<=u then begin write(ch);
```

```
                                                   st:=concat(st,ch)
                                             end
                           end
               end;
               if c<=u then read(kbd,ch)
            end;
         if st<>'' then val(st,sayi,result);
         gotoxy(x,y); if sayi<>0 then write(sayi:u)
      end;
rocedure realoku(x,y,u:byte; var sayi:real);
  var c,l,i:byte; result:integer; st:string[10];
  begin lowvideo; gotoxy(x,y);
         for l:=1 to u do write(' '); highvideo;
        gotoxy(x,y); c:=0;st:=''; ch:=' ';
        read(kbd,ch);
        while (ch=' ') and (c<u) do
            begin write(' '); c:=c+1; read(kbd,ch) end;
         while (c<=u) and (ord(ch)<>13) do
            begin
               case ord(ch) of
                  8  : if st<>'' then
                          begin write(ch); lowvideo; write(' ');
                                highvideo; write(ch);
                                delete(st,c,1); c:=c-1
                          end;
                  32 : if st='' then begin c:=c+1; write(ch)
                                     end;
                  48..57 : begin c:=c+1;
                                 if c<=u then begin write(ch);
                                                   st:=concat(st,ch)
                                             end
                           end
               end;
               if c<=u then read(kbd,ch)
            end;
         if st<>'' then val(st,sayi,result);
         gotoxy(x,y); if sayi<>0 then write(sayi:u:0)
      end;
  procedure blank(x,y,u:byte);
      var j: byte;
      begin lowvideo;
                gotoxy(x,y);
                     for j:=1 to u do write(' ');
            highvideo
      end;
ROCEDURE FORM1;  VAR I:INTEGER;
EGIN CLRSCR;GOTOXY(30,2);LOWVIDEO; WRITELN('**  MAKINANIN  **');HIGHVIDEO;
      WRITELN('       ADI:');
      WRITELN('       KODU:             BAKIM PERIYODU:');
      WRITELN(' DOSYA NOSU:          REVIZYON PERIYODU:');
      WRITELN('MALIYET MER:',' ':15,'Dakikalik Durma Maliyeti (x100TL):');
      LOWVIDEO;WRITELN('     YAPILAN  BAKIMLAR:   ');HIGHVIDEO;
      WRITELN('             Elek Meka  Usta  Durma  Bakim    Kullanilan');
      WRITELN('No Tarih Tip trik  nik  gucy suresi suresi   Yedek Parca    Maliyet(x100TL
      WRITELN('':35,'Nor FMesai',' ':15,'Periodik  Arizi');
      FOR
```

```
       I:=1 TO XA DO  WRITELN(I:2);
    WRITELN('    TOPLAM DURMA SURESI:     dakika');WRITE('ORTALAMALAR:');
END;

PROCEDURE MACSEC; {Bakim Zamani Gelen Makinalari Secer.}
 VAR TEMP,II,AB,J,ji,TOPELSUR,NBM,NBH,TOPMEKSUR,TOPUSSUR:INTEGER;
   COSTVEC:ARRAY[RG] OF REAL; rtemp:real;
   ALLDONE,ND,RD,TH:BOOLEAN; JUMP,M,N:RG;
   BM,MEIN,MMIN,MUN:ARRAY[RG] OF INTEGER;
{1} BEGIN  CLRSCR;GOTOXY(2,15); cw:=1;
       WRITELN('BAKIM PLANLAMASI YAPMAK ISTEDIGINIZ');
       WRITE(' HAFTA NUMARASINI GIRINIZ (1..52)  > ');
       intoku(38,16,2,cw);
         while NOT (CW IN [1..52]) DO  begin   gotoxy(20,20);
           write('Lutfen tekrar girin [1..52] > ');
           gotoxy(38,16);clreol; intoku(38,16,2,cw);end;
       CLRSCR; RESET(MACHINEFILE); GOTOXY(28,1); LOWVIDEO;
       WRITELN(' Bu Hafta (',cw:2,'.hafta) '); HIGHVIDEO;
       WRITELN('Bakilmasi Gereken Makina/Techizat..');  NBM:=0;ab:=0;
       FOR I:=1 TO 5 DO
{2}    BEGIN FOR II:=1 TO 20 DO
{3}        BEGIN READ(MACHINEFILE,MACHINEREC);
               WITH MACHINEREC DO
{4}            BEGIN  if codeno<>0 then
                begin EB:=0;
                   FOR J:=1 TO XA DO
{5-5}              BEGIN IF EB<MAINDATE[J] THEN EB:=MAINDATE[J]; {5} END;
                   IF (EB+MAINPERIOD)<=CW THEN
{6}                BEGIN NBM:=NBM+1;BM[NBM]:=NUMBER;
                       COSTVEC[NBM]:=COSTPMINLOST*ORTDURSUR;
                       MEIN[NBM]:=ORTEL*ORTBAKSUR;
                       MMIN[NBM]:=ORTMEK*ORTBAKSUR; MUN[NBM]:=ORTUS*ORTBAKSUR
                       GOTOXY(1+AB,3+II);
                       WRITE(NBM:2,':',BM[NBM]:2,' ');
                       i:=eb+mainperiod-cw;
                       if availability > 0 then
                       costvec[nbm]:=costvec[nbm]/exp(i*ln(availability));
                   END;
                end;
               END;
           END;  AB:=AB+7;
{2}    END;   LOWVIDEO; GOTOXY(35,2); WRITE(NBM);
       FOR I:=2 TO 23 DO
       BEGIN GOTOXY(37,I);WRITE(' '); END; HIGHVIDEO;
       GOTOXY(38,2);
       WRITE('Bakilabilecek Makina / Techizat. ');
       {sorting CostVec}
       jump:=nbm;
       WHILE jump>1 do
       Begin jump:=jump div 2;
           REPEAT    alldone:=true;
               for m:=1 to nbm-jump do
               Begin n:=m+jump;
                   if costvec[m]<costvec[n] then
                   begin rtemp:=costvec[m]; costvec[m]:=costvec[n];
                     costvec[n]:=rtemp; temp:=bm[m];bm[m]:=bm[n];
                     bm[n]:=temp; temp:=mein[m]; mein[m]:=mein[n];
                     mein[n]:=temp; temp:=mmin[m]; mmin[m]:=mmin[n];
                     mmin[n]:=temp; temp:=mun[m]; mun[m]:=mun[n];
                     mun[n]:=temp;     alldone:=false
```

```
                             end
                       end
                 UNTIL alldone
           end; NBH:=0;TOPELSUR:=0;
           TOPMEKSUR:=0;TOPUSSUR:=0;AB:=0;II:=0;
           FOR I:=1 TO NBM DO
{9}        BEGIN II:=II+1;IF II>20 THEN  BEGIN II:=0;AB:=AB+7;END;
                 ND:=(TOPELSUR+MEIN[I])<=(EN*WORKWEEK);
                 RD:=(TOPMEKSUR+MMIN[I])<=(MN*WORKWEEK);
                 TH:=(TOPUSSUR+MUN[I])<=(UN*WORKWEEK);
                 IF ND AND RD AND TH THEN
{10}             BEGIN NBH:=NBH+1;GOTOXY(38+AB,3+II);LOWVIDEO;
                       WRITE(NBH:2,':',BM[I]:3); HIGHVIDEO;
                       TOPELSUR:=TOPELSUR+MEIN[I];
                       TOPMEKSUR:=TOPMEKSUR+MMIN[I];
                       TOPUSSUR:=TOPUSSUR+MUN[I];
{10}             END;
              END; gotoxy(71,2); lowvideo; write(nbh); highvideo;
         GOTOXY(3,24);WRITE('Devam etmek icin bir tusa basin.');
            while not(keypressed) do delay(100);
  END;

procedure makinagor;         {MAKINALARI GORMEK / BAKIM ISLEMEK}
 var sa,iscilik,percost,j,b,tl,mm:integer;s:string[15];yno,ykul:array[1..5] of integer;
    scs,ytip:string[1];n,giren,cikan:byte;
 dogruparca,ydkvar,yvarmi,bakim:boolean;
   begin clrscr;gotoxy(3,23);
         write('    KAC NUMARALI MAKINAYI GORMEK ISTIYORSUNUZ ? > ');
         intoku(53,23,3,sa);
         WHILE sa in [1..maxnoofmachines] do
         begin clrscr;form1;
               yvarmi:=false;  bakim:=false;gotoxy(1,1);
               seek(machinefile,sa-1);
               read(machinefile,machinerec);
               WITH machinerec do
               begin
              {3}    gotoxy(14,3);write(name); gotoxy(14,4);write(codeno);
                     gotoxy(14,5);write(number); gotoxy(14,6);write(fazla);
                     gotoxy(42,4);write(mainperiod);
                     gotoxy(42,5);write(revperiod);
                     gotoxy(63,6);write(costpminlost);
                     FOR j:=1 to xa do
                     begin
                    {4}    gotoxy(6,10+j);write(maindate[j]);
                           gotoxy(11,10+j);write(maintype[j]:1);
                           gotoxy(15,10+j);write(elektrikci[j]);
                           gotoxy(20,10+j);write(mekanikci[j]);
                           gotoxy(26,10+j);write(usta[j]);
                           gotoxy(31,10+j);write(failuredur[j]);
                           gotoxy(36,10+j);write(maintdur[j]);
                           gotoxy(40,10+j);write(fmmaindur[j]);
                           for i:=1 to 5 do
                           begin if sparesused[j,i]<>0 then
                                 begin gotoxy(42+3*i,10+j);
                                       write(sparesused[j,i]:3);
                                 end;
                           end; gotoxy(63,10+j);write(pcost[j]);
                           gotoxy(71,10+j);write(fcost[j]);
                 {4} end; gotoxy(15,22); write(ortel);
                     gotoxy(20,22); write(ortmek);
                    gotoxy(26,22); write(ortus);
                    gotoxy(31,22);write(ortdursur);
                    gotoxy(27,21);write(topimak:6:0);
                    gotoxy(38,22); write(ortbaksur);
                    gotoxy(63,22); write(portcost);
                    gotoxy(71,22); write(fortcost);
                    gotoxy(45,3);write(availability:6:4);
                    for i:=1 to 5 do  begin ykul[i]:=0;yno[i]:=0; end;
                    gotoxy(20,23); write(' BAKIM ISLEMEK ISTER MISINIZ ? (E,*)   > ');
                    gotoxy(60,23); clreol; read(kbd,g); write(g);
                    IF g in ['E','e','*'] THEN
                    begin
                 {5}      mza:=mza+1;
                          if mza>10 then mza:=1;
                          bakim:=true; b:=0;
                          intoku(5,10+mza,2,b);
                          if b<>0 then maindate[mza]:=b; b:=0; scs:='';
                          gotoxy(11,10+mza); read(scs);
                          if scs<>'' then maintype[mza]:=scs; scs:='';
                          intoku(14,10+mza,2,b);
                          if b<>0 then elektrikci[mza]:=b; b:=0; intoku(19,10+mza,2,b);
                          if b<>0 then mekanikci[mza]:=b;b:=0;intoku(25,10+mza,2,b);
                          if b<>0 then usta[mza]:=b; b:=0;intoku(30,10+mza,4,b);
                          if b<>0 then failuredur[mza]:=b; b:=0;
                          topimak:=topimak+failuredur[mza];
                          gotoxy(29,21);write(topimak:6:0);
availability:=(60.0*maindate[mza]*weekhour-topimak)/(60.0*maindate[mza]*weekhour);
                          gotoxy(45,3);write(availability:6:4);
                          intoku(36,10+mza,4,b);if b<>0 then maintdur[mza]:=b; b:=0;
                          intoku(40,10+mza,4,b);if b<>0 then fmmaindur[mza]:=b;b:=0;
                          i:=0; b:=1;
                          repeat  ydkvar:=false; tl:=0; intoku(44+3*i,10+mza,3,tl);
                                if tl<>0 then
                                begin  i:=i+1; ykul[i]:=maindate[mza];
                                     yno[i]:=tl; sparesused[mza,i]:=tl;
                                     yvarmi:=true; ytip:=maintype[mza];
                                end;
                                if (tl=0) or (i=5) then ydkvar:=true;
                          until ydkvar;  b:=sa;
                 {5}end;
          {3} end;
              seek(machinefile,sa-1);  write(machinefile,machinerec); tl:=0;
              if yvarmi then
              begin for j:=1 to 5 do
                 begin  if yno[j]<>0 then
                 begin  mm:=yno[j]-1;seek(ydkfile,mm); read(ydkfile,ydkrec);
                    with ydkrec do
                    begin  dogruparca:=false;
                  {B}       if makno=b then dogruparca:=true;
                            if dogruparca then
                       begin
                          xza:=xza+1;
                          if xza>30 then xza:=1;
                          tl:=tl+ucost;
                          kullandate[xza]:=ykul[j];tip[xza]:=ytip;
                          ydkfazla:=ydkfazla-1;
                          if ydkfazla<=mnlvl THEN
                          begin  gotoxy(54,1);lowvideo;
```

```
                                  write('DIKKAT:');
                                  highvideo; write('siparis seviyesine');
                                  gotoxy(59,2); write('dusen Yedek Parcalar');
                                  gotoxy(73,2+j); lowvideo;
                                  write(ynumber:3); highvideo;
                           end
                        end
                        else begin gotoxy(1,24);
                                write('Bu makinaya ait olmayan yedek(ler): ');
                                gotoxy(37+3*(j-1),24);write(yno[j]:3);yno[j]:=0;
                             end;
                     end;  seek(ydkfile,mm);  write(ydkfile,ydkrec);
                  end;
                  end;
               end; b:=0;
               if bakim then
            begin    seek(machinefile,sa-1); read(machinefile,machinerec);
               with machinerec do
               begin  for j:=1 to 5 do
                  begin sparesused[mza,j]:=yno[j]; gotoxy(42+3*j,10+mza);
                        if sparesused[mza,j]<>0 then write(sparesused[mza,j]:3);
                  end;
         iscilik:=((elektrikci[mza]+mekanikci[mza])*dic+usta[mza]*duc) div 10;
                     percost:=iscilik*(maintdur[mza]+2*fmmaindur[mza]) div 10;
                     parcacost[mza]:=tl;
                     emniyet[mza]:=costpminlost*failuredur[mza];
                     if (maintype[mza]='a') OR (maintype[mza]='A') then
                        fcost[mza]:=tl+percost+emniyet[mza];
                     if (maintype[mza]='p') or (maintype[mza]='P') then
                        pcost[mza]:=tl+percost+emniyet[mza];
                     gotoxy(63,10+mza);if pcost[mza]<>0 then write(pcost[mza]);
                     gotoxy(71,10+mza);if fcost[mza]<>0 then write(fcost[mza]); b:=0;
                     intoku(76,10+mza,4,b);  if b<>0 then
                     begin if pcost[mza]<>0 then pcost[mza]:=b;
                           if fcost[mza]<>0 then fcost[mza]:=b;
                     end; b:=0;
                     gotoxy(3,23); clreol;
                     write('Bu satirda degisiklik yapacak misiniz? (E,*)  > ');
                     read(kbd,g); write(g);
                     if g in ['e','E','*'] then mza:=mza-1;
          {9}  end;  seek(machinefile,sa-1);   write(machinefile,machinerec);
            end;  sa:=sa+1;  gotoxy(1,23); clreol;
               writeln('  KAC NUMARALI MAKINAYI GORMEK ISTIYORSUNUZ? > ');
               write('    (Bir sonraki icin <RETURN> ; 0=DUR  )');
               intoku(48,23,3,sa);
     {2} end;
{1}end;
procedure makinayaz;
   var sa,b:integer; s:string[10]; n:byte;
   begin clrscr; gotoxy(1,17);
         writeln('        MAKINALARI GORMEK / BAKIM ISLEMEK  ICIN  1');
         write('MAKINA KARTLARINI ACMAK / DEGISTIRMEK    ICIN  2     girin  > ');
         read(kbd,g); write(g);
         CASE g of
            '1':makinagor;
            '2':begin clrscr; gotoxy(1,23);
                write('  YAZMAK ISTEDIGINIZ MAKINA NUMARASINI GIRINIZ. > ');
         intoku(51,23,3,sa);
         WHILE sa in [1..maxnoofmachines] do
         begin clrscr; form1; seek(machinefile,sa-1); read(machinefile,machinerec);
               WITH machinerec do
               begin gotoxy(14,3);write(name); gotoxy(14,4);write(codeno);
                     gotoxy(14,5);write(number); gotoxy(14,6);write(fazla);
                     gotoxy(42,4);write(mainperiod);  gotoxy(42,5);write(revperiod);
                     gotoxy(63,6);write(costpminlost);  s:='';
                     gotoxy(14,3); read(s);
                     if s <>'' then name:=s; s:='';  b:=0;intoku(14,4,4,b);
                     if b<>0 then codeno:=b; b:=0; n:=0;intoku(14,6,4,b);
                     if b<>0 then fazla:=b; b:=0;
                     blank(42,4,2);gotoxy(42,4);read(n);
                     if n<>0 then mainperiod:=n; n:=0;
                     blank(42,5,2);gotoxy(42,5); read(n);
                     if n<>0 then revperiod:=n; n:=0;intoku(62,6,4,b);
                     if b<>0 then costpminlost:=b; b:=0;
               end; seek(machinefile,sa-1); write(machinefile,machinerec);
               sa:=sa+1; gotoxy(1,23);
               writeln('  YAZMAK ISTEDIGINIZ MAKINA NUMARASINI GIRIN. > ');
               write('    (Bir sonraki icin <RETURN> ;  0=DUR  )');
               intoku(49,23,3,sa);
         end
         end
   end;
   end;

{$I RT15.PAS}
{$I RT2.PAS}
{$I RT3.PAS}
```

```
rocedure agebal;
  const e=0.1E-1;
  var n:byte; ar,tfark,b,sr,i:integer; tamam:boolean;
     parmal,Cp,Cf,Ct,landa,dhead,beta,lnsum,lnsqrsum,t,altsum,
     a,k,tp,dt:real;
  begin  g:='e';
    while g in ['e','E','*'] do
    begin clrscr;gotoxy(3,10);
      writeln('Bu secenek, herhangi bir yedek parcanin ne kadar');
      writeln('  zamanda bir degistirilmesi gerektigini onerir..');
      writeln;write('  Yedek Parcanin dosya numarasini girin. > ');
      b:=1; intoku(46,13,3,b);
      seek(ydkfile,b-1); read(yokfile,ydkrec);
      with ydkrec do
      begin
        n:=0; lnsum:=0; lnsqrsum:=0;
        for i:=2 to ydkxa do
        begin
          if (tip[i]='a') or (tip[i]='A') or (tip[i]='p')
            or (tip[i]='P') then
          begin n:=n+1; tfark:=kullandate[i]-kullandate[i-1];
            if tfark=0 then tfark:=1;
            lnsum:=lnsum+ln(abs(tfark));
            lnsqrsum:=lnsqrsum+sqr(ln(abs(tfark)));
          end;
        end; ar:=makno; parmal:=ucost*100;
      end;
      if n>=2 then
      begin
        Ct:=6/(sqr(3.14)*(n-1));
        dhead:=sqrt(Ct*(lnsqrsum-sqr(lnsum)/n));
        beta:=1/dhead;
        landa:=exp(lnsum/n-0.577*dhead);
        if beta>1 then
        begin    gotoxy(5,14);
          writeln('" Shape Parameter "  BETA = ',beta:7:3);
          writeln(' ':4,'" Scale Parameter "  TETA = ',landa:7:3);
          clreol;  k:=beta/exp(beta*ln(landa));
          seek(machinefile,ar-1); read(machinefile,machinerec);
          with machinerec do
          begin
            Cp:=parmal+(dic*(ortel+ortmek)+duc*ortus)*ortbaksur;
            Cf:=Cp+costpminlost*ortdursur*100;
          end;  if (Cp=Cf) then Cf:=Cp+0.0001;
          t:=5; va:=0; tamam:=false;
          REPEAT
            dt:=t/5; altsum:=0; va:=va+1; lnsum:=dt;
            for i:=1 to 5 do
            begin
              altsum:=altsum+exp(beta*ln(lnsum))*dt*
               exp(-exp(beta*ln(lnsum/landa)));
              lnsum:=lnsum+dt;
            end;
            a:=(exp(ln(t)*(beta-1))*k*k*(Cf-Cp)*altsum-Cf)/
             ((1+k*exp(beta*ln(t)))*(Cf-Cp));
            tp:=-ln(abs(a))*exp(beta*ln(landa));
            tp:=exp(ln(abs(tp))/beta);
            if (abs(tp-t)<e) or (va=25) then tamam:=true;
            t:=abs(tp); gotoxy(70,23);write(t:8:2);
          UNTIL tamam; i:=round(t); gotoxy(69,23);
```

```
               write(' ':10);  gotoxy(5,18); lowvideo;
               writeln('"',b:3,'" Nolu Yedek Parcanin',' ':23);
               gotoxy(5,19);
               write('ONERILEN DEGISIM SURESI ',i:2,' ( ',t:7:3);
               writeln(' ) HAFTADIR.'); highvideo;
               write(' ':4,'Bunu Kabul Etmek Ister misiniz? (E,*) > ');
               read(kbd,ch); write(ch);
               if ch in ['E','e','*'] then
               begin
                  seek(ydkfile,b-1); read(ydkfile,ydkrec);
                  ydkrec.reptime:=i;
                  seek(ydkfile,b-1); write(ydkfile,ydkrec);
               end;
            end;
            if beta=1 then
            begin   gotoxy(5,14);
               writeln('DAGILIM ZAMANA GORE SABIT GOZUKUYOR');
               write(' ':4,'PLANLI BAKIM UYGULAMAK VERIMLI DEGIL');
               clreol;
            end;
            if beta<1 then
            begin  gotoxy(5,14);
               writeln('DAGILIM ZAMANA GORE AZALAN BOZULMA IHTIMALI GOSTERIYOR');
               write(' ':4,'PLANLI BAKIM YAPMAK VERIMLI DEGIL !');clreol;
            end;
         end;
         if n<2 then
         begin  gotoxy(15,19); lowvideo;
            write('YETERLI BILGI YOK'); highvideo;
         end;  gotoxy(2,22); writeln('Baska bir yedek parcanin');
         write(' onerilen degisim suresini bulmak ister misiniz? ');
          write('(E,*) > ');  read(kbd,g); write(g);
        end;
   end;

PROCEDURE ORTBUL;
  VAR J,IEB,EB:BYTE;TOPEL,TOPMEK,TOPUS,TOPBAK,toplamsure,
      topelsur,topmeksur,topussur,fn,pn:INTEGER;
      PTOPMAL,FTOPMAL,TOPDUR,mtek,etek,utek,durreal:REAL;
      costvec,bm,sure,mein,mmin,mun:array[RG] of integer;
   BEGIN
      RESET(MACHINEFILE);
      FOR I:=1 TO MAXNOOFMACHINES DO
      BEGIN SEEK(MACHINEFILE,I-1); READ(MACHINEFILE,MACHINEREC);
         IF MACHINEREC.MAINDATE[1]<>0 THEN
         BEGIN
            WITH MACHINEREC DO
            BEGIN
               IEB:=0;  pn:=0; fn:=0;
               FOR J:=1 TO XA DO
               BEGIN  IF  MAINDATE[J]<>0 THEN IEB:=J;
                  IF (MAINTYPE[J]='A') or (MAINTYPE[J]='a')
                   THEN FN:=FN+1;
                  IF (MAINTYPE[J]='P') or (MAINTYPE[J]='p')
                   THEN PN:=PN+1;
               END;   if ieb<>0 then
               begin
                  TOPEL:=0;TOPMEK:=0;TOPUS:=0;
                  TOPDUR:=0;TOPBAK:=0;PTOPMAL:=0;FTOPMAL:=0;
                  FOR J:=1 TO IEB DO
                  BEGIN  TOPEL:=TOPEL+ELEKTRIKCI[J];
                     TOPMEK:=TOPMEK+MEKANIKCI[J];
                     TOPUS:=TOPUS+USTA[J];
                     TOPBAK:=TOPBAK+MAINTDUR[J]+FMMAINDUR[J];
                     PTOPMAL:=PTOPMAL+PCOST[J];
                     FTOPMAL:=FTOPMAL+FCOST[J];
                  END;  topdur:=topimak;
                  if maindate[10]<>0 then
                  begin   topel:=topel+ortel;
                     topmek:=topmek+ortmek; topus:=topus+ortus;
                     topdur:=topimak+ortdursur;
                     ptopmal:=ptopmal+portcost; pn:=pn+1;
                     ftopmal:=ftopmal+fortcost; fn:=fn+1;
                     topbak:=topbak+ortbaksur; ieb:=ieb+1;
                  end; etek:=topel;mtek:=topmek;
                  utek:=topus;durreal:=topbak;
                  ORTEL:=round(etek/IEB); ORTMEK:=ROUND(mtek/IEB);
                  ORTUS:=ROUND(utek/IEB);
                  ORTDURSUR:=ROUND(TOPDUR/IEB);
                  ORTBAKSUR:=ROUND(durreal/IEB);
                  if pn<>0 then PORTCOST:=ROUND(PTOPMAL/PN);
                  if fn<>0 then FORTCOST:=ROUND(FTOPMAL/FN);
                  SEEK(MACHINEFILE,I-1);WRITE(MACHINEFILE,MACHINEREC);
               end;
            END;
         END;
      END;
   END;

procedure kodara;
   var b,d,n:integer; ok,kontrol:boolean;
   begin  clrscr; gotoxy(3,16);
          writeln('DOSYA NUMARASINI OGRENMEK ISTEDIGINIZ ');
          write('  MAKINANIN KOD NUMARASINI GIRINIZ   > ');
          intoku(40,17,4,b); reset(machinefile);
          n:=0; ok:=false; kontrol:=false;
          REPEAT  read(machinefile,machinerec);
          if machinerec.codeno=b then
          begin
                ok:=true;  d:=machinerec.number;
          end;  n:=n+1;
          if n>=maxnoofmachines then
          begin
               ok:=true; kontrol:=true;
          end;
          UNTIL ok; clrscr;
          if not kontrol then
          begin  gotoxy(3,16);
             writeln('KOD NUMARASI "',b,'"  OLAN MAKINANIN');
             write('  DOSYA NUMARASI ',d,'  dir.');
          end;
          if kontrol then
          begin     gotoxy(3,16);
             write('MEVCUT MAKINALAR ARASINDA "',b);
              writeln('"  KOD NUMARALI');
             write('  BIR MAKINA BULUNAMADI..');
          end;
   end;

procedure adara;
   var s:string[15]; d,n:integer; ok,kontrol:boolean;
   begin  clrscr; gotoxy(3,16);
          writeln('DOSYA NUMARASINI OGRENMEK ISTEDIGINIZ ');
          write('  MAKINANIN ADINI GIRINIZ    > '); blank(31,17,15);
          gotoxy(31,17); read(s); reset(machinefile);n:=0;
          ok:=false; kontrol:=false;
          REPEAT    read(machinefile,machinerec);
          if machinerec.name=s then
          begin
               ok:=true; d:=machinerec.number;
          end; n:=n+1;
          if n>=maxnoofmachines then
          begin  ok:=true; kontrol:=true;
          end;
          UNTIL ok; clrscr;
          if not kontrol then
          begin gotoxy(3,16);
                writeln('ADI  "',s,'"   OLAN MAKINANIN');
                write('  DOSYA NUMARASI ',d,'  dir.');
          end; if kontrol then
          begin  gotoxy(3,16);
             writeln('MEVCUT MAKINALAR ARASINDA  "',s,'"   ADINDA');
             write('  BIR MAKINA BULUNAMADI..');
          end;
    end;

procedure ara;
   begin  g:='e';
        WHILE g in ['e','E','*'] do
        begin  clrscr;gotoxy(3,18);
              writeln('Dosya numarasini aradiginiz makinanin');
              writeln(' ':12,'KOD NUMARASINI verecekseniz 1');
              write(' ':21,'ADINI verecekseniz 2      girin > ');
              read(kbd,ch); write(ch);
              case ch of
              '1':kodara;
              '2':adara   end;  gotoxy(3,22);
              writeln('Baska bir makinanin dosya numarasina ');
              write('  bakmak ister misiniz ? (E,*) > ');
              read(kbd,g);write(g);
        end;
   end;

procedure ydkkodara;
   var b,d,n:integer; ok,kontrol:boolean;
   begin  clrscr; gotoxy(3,16);
         writeln('DOSYA NUMARASINI OGRENMEK ISTEDIGINIZ');
         write('  YEDEK PARCANIN KOD NUMARASINI GIRINIZ   > ');
         intoku(45,17,4,b); reset(ydkfile);  n:=0;
         ok:=false; kontrol:=false;
         REPEAT  read(ydkfile,ydkrec);
                 if ydkrec.ycod=b then
                 begin ok:=true; d:=ydkrec.ynumber;
                 end; n:=n+1;
                 if n>=maxnoofydk then
                 begin ok:=true; kontrol:=true;
                 end;
         UNTIL ok;
         clrscr;  if not kontrol then
         begin  gotoxy(3,16);
              writeln('KOD NUMARASI "',b,'"  OLAN YEDEK PARCANIN');
              write('  DOSYA NUMARASI ',d,'  dir.');
         end;  if kontrol then
         begin  gotoxy(3,16);
                write('MEVCUT YEDEK PARCALAR ARASINDA "',b);
                 writeln('"  KOD NUMARALI');
                write('  BIR YEDEK PARCA BULUNAMADI..');
         end;
   end;

procedure ydkadara;
   var s:string[15]; d,n:integer; ok,kontrol:boolean;
   begin  clrscr; gotoxy(3,16);
         writeln('DOSYA NUMARASINI OGRENMEK ISTEDIGINIZ');
         write('  YEDEK PARCANIN ADINI GIRINIZ   > ');
         blank(36,17,15);gotoxy(36,17); read(s); reset(ydkfile);
         n:=0; ok:=false; kontrol:=false;
         REPEAT  read(ydkfile,ydkrec);
               if ydkrec.isim=s then
               begin ok:=true; d:=ydkrec.ynumber;
               end;n:=n+1;
               if n>=maxnoofydk then
               begin ok:=true; kontrol:=true;
               end;
         UNTIL ok;clrscr;
         if not kontrol then
         begin gotoxy(3,16);
               writeln('ADI  "',s,'"   OLAN YEDEK PARCANIN');
               write('  DOSYA NUMARASI ',d,'  dir.');
         end; if kontrol then
         begin gotoxy(3,16);
            write('MEVCUT YEDEK PARCALAR ARASINDA  "',s);
             writeln('"   ADINDA');
            write('  BIR YEDEK PARCA BULUNAMADI..');
         end;
   end;

procedure ydkara;
begin    g:='e';
      WHILE g in ['e','E','*'] do
      begin clrscr;gotoxy(3,18);
            writeln('Dosya numarasini aradiginiz YEDEK PARCANIN');
            writeln(' ':12,'KOD NUMARASINI vereceksemiz 1');
            write(' ':21,'ADINI vereceksemiz 2     girin > ');
            read(kbd,ch); write(ch);
            case ch of
            '1':ydkkodara;
            '2':ydkadara  end; gotoxy(3,22);
            writeln('Baska bir yedek parcanin dosya numarasina ');
            write('  bakmak ister misiniz ? (E,*) > ');
            read(kbd,g);write(g);
      end;
end;

procedure masraf;
   var ii:integer;  stokpara:real;
   begin           clrscr;
         reset(ydkfile); stokpara:=0;
         for ii:=1 to maxnoofydk do
         begin
               read(ydkfile,ydkrec);
               with ydkrec do   stokpara:=stokpara+ydkfazla*ucost;
         end; gotoxy(1,16);
         writeln('HALEN STOKTA MEVCUT');
         write('YEDEK PARCALARIN DEGERI  ',stokpara/10:8:2);
          write('  binTL  dir.');
         gotoxy(3;23);write('Devam etmek icin bir tusa basiniz.');
         while not(keypressed) do delay(100);
   end;

procedure kackisilazim;
   const yilcalsur=10392;
   var tsumel,tsummek,tsumus,yildursur:real;
       i,elger,mekger,usger:integer;
   begin clrscr;  reset(machinefile);
         tsumel:=0; tsummek:=0; tsumus:=0;
         for i:=1 to maxnoofmachines-1 do
         begin read(machinefile,machinerec);
               if machinerec.codeno<>0 then
               begin with machinerec do
                     begin yildursur:=ortdursur*52/mainperiod;
                           tsumel:=tsumel+yildursur*ortel;
                           tsummek:=tsummek+yildursur*ortmek;
                           tsumus:=tsumus+yildursur*ortus;
                     end;
               end;
         end; elger:=round(0.11*tsumel/yilcalsur);
         mekger:=round(0.11*tsummek/yilcalsur);
         usger:=round(0.11*tsumus/yilcalsur);
         clrscr;gotoxy(5,7);  lowvideo;
         write('MEVCUT MAKINALARIN PERIODIK BAKIMI ICIN ');
          write('GEREKLI PERSONEL :'); highvideo; gotoxy(25,11);
         write('     ELEKTRIKCI : ',elger);highvideo;gotoxy(25,13);
         write('      MEKANIKCI : ',mekger); gotoxy(25,15);
         write('KALIFIYE ELEMAN : ',usger); gotoxy(3,23);
         write('Devam etmek icin bir tusa basiniz.');
         while not(keypressed) do delay(100);
   end;

PROCEDURE LISTE(VAR XXFILE:MACHF;XXREC:MACHINE); VAR  K,J:INTEGER;
   BEGIN CLRSCR;CW:=0;
         WRITE('MEVCUT MAKINALARIN KOD VE DOSYA NUMARALARI:');
         RESET(XXFILE); FOR I:=1 TO 5 DO
         BEGIN  FOR J:=1 TO 20 DO BEGIN READ(XXFILE,XXREC);
                WITH XXREC DO  BEGIN   GOTOXY(1+CW,1+J);
                WRITE(CODENO:8,' : ',NUMBER:3); END;END;
                CW:=CW+15; END;
         GOTOXY(14,23);
         WRITE('DEVAM ETMEK ICIN HERHANGI BIR TUSA BASINIZ.');
         READ(KBD,CH); if ch='#' then cw:=0;
   END;

procedure ydkform;    var i:integer;
   begin clrscr;gotoxy(27,1);lowvideo;
         writeln('** YEDEK  PARCANIN  **');
         highvideo;  writeln;
         write('       ADI:                 MIN.STOK:        ');
          writeln('SIPARIS MIKTARI:');
         write('     KODU:                    BIRIM:  ');
          writeln('          GELIS SURESI:');
         write('DOSYA NOSU:       BIRIM FIAT(x100TL):       ');
          writeln('    MAKINA NOSU:');
         writeln(' ':48,'MEVCUT STOK:');
         write('      gelis      gelis                   ');
          writeln('        kullanim    kullanim      ');
         write('No   tarihi    miktari     kimden-geldigi   ');
          writeln('   tarihi       sebebi      ');
         writeln; for i:=1 to xa do writeln(i:2);
         gotoxy(1,20); write('Onerilen degisim suresi :');
   end;

procedure ydklist(var xxfile:ydkf;xxrec:parca);   var k,j:integer;
    begin CLRSCR;CW:=0;
       WRITE('MEVCUT YEDEK PARCALARIN KOD VE DOSYA NUMARALARI:');
       RESET(XXFILE);
       FOR K:=1 TO 5 DO
       BEGIN  FOR I:=1 TO 5 DO
          BEGIN  FOR J:=1 TO 20 DO
             BEGIN  READ(XXFILE,XXREC);
                WITH XXREC DO    BEGIN   GOTOXY(1+CW,1+J);
                WRITE(YCOD:8,' : ',YNUMBER:3);  END;
             END;CW:=CW+15;
          END;GOTOXY(14,23);
          WRITE('Devam etmek icin bir tusa basin. > ');
          while not(keypressed) do delay(100);    CW:=0;
       END;
   end;
```

```
procedure ydkbasla;   var i,ii:integer; g:char;
   begin  clrscr;lowvideo;gotoxy(5,4);write('  D I K K A T  ');highvideo;gotoxy(5,7);
            write('PROGRAMIN HAFIZASINDAKI BUTUN YEDEK PARCA');gotoxy(5,9);
            write('BILGILERINI SILMEK UZERESINIZ...');
            REPEAT gotoxy(5,11); clreol;
                   write('ISTEDIGINIZ GERCEKTEN BU MU ? (E / H) > ');
                   read(kbd,g);write(g);
            UNTIL g in ['e','E','h','H'];
            IF g in ['e','E'] THEN
            begin gotoxy(50,19);write('Lutfen Bekleyiniz.'); rewrite(ydkfile); i:=1;
                  WHILE i<=maxnoofydk do
                  begin seek(ydkfile,i-1);
                        WITH ydkrec do
                        begin isim:=''; birim:=''; ynumber:=i; ycod:=0;xza:=0;
                              ucost:=0; rorderq:=0; leadt:=0;ydkfazla:=0;
                              mnlvl:=0; reptime:=0; makno:=0;
                              for ii:=1 to ydkxa do
                              begin kullandate[ii]:=0; tip[ii]:=' ';
                              end;
                              FOR ii:=1 to xa do
                              begin kimden[ii]:='';geldate[ii]:=0;
                                    gelmik[ii]:=0;
                              end; write(ydkfile,ydkrec); end; i:=i+1;end; end;
   end;

procedure ydkyaz;  var sa,j,b:integer; s:string[10];n:byte; makvar:boolean;
   begin clrscr;gotoxy(3,23);
         write('KAC NUMARALI YEDEK PARCAYI YAZMAK ISTIYORSUNUZ? > ');
         intoku(53,23,3,sa);
         WHILE sa in [1..maxnoofydk] do
           begin clrscr;ydkform; seek(ydkfile,sa-1);read(ydkfile,ydkrec);
                WITH ydkrec do
                begin  gotoxy(12,3);write(isim);gotoxy(12,4);write(ycod);
                     gotoxy(12,5);write(ynumber);gotoxy(38,3);write(mnlvl);
                     gotoxy(38,4);write(birim);gotoxy(38,5);write(ucost);
                     gotoxy(61,3);write(rorderq);gotoxy(61,4);write(leadt);
                     gotoxy(61,5); write(makno);
                     gotoxy(61,6);write(ydkfazla);
                     gotoxy(27,20); if reptime<>0 then write(reptime); s:=''; b:=0; n
                     blank(12,3,10);gotoxy(12,3); read(s); if s<>'' then isim:=s;s:='
                     intoku(12,4,4,b); if b<>0 then ycod:=b;b:=0;
                     blank(38,3,2);gotoxy(38,3); read(n); if n<>0 then mnlvl:=n;n:=0;
                     blank(38,4,10);gotoxy(38,4); read(s); if s<>'' then birim:=s;s:=
                     intoku(38,5,4,b); if b<>0 then ucost:=b;b:=0;
                     blank(61,3,2);gotoxy(61,3); read(n); if n<>0 then rorderq:=n;n:=
                     blank(61,4,2);gotoxy(61,4); read(n); if n<>0 then leadt:=n;n:=0;
                     intoku(61,5,3,b); if b<>0 then makno:=b; b:=0;
                     intoku(61,6,3,b); if b<>0 then ydkfazla:=b; b:=0;
                end;   {with ydkrec}
                seek(ydkfile,sa-1); write(ydkfile,ydkrec); sa:=sa+1;
                gotoxy(3,23);
                writeln('KAC NUMARALI YEDEK PARCAYI YAZMAK ISTIYORSUNUZ? > ');
                write('    (Bir sonraki icin <RETURN> ;  0=dur  )');
                blank(54,23,3);gotoxy(54,23);read(sa); end;
   end;

procedure ydkgor;
      VAR sa,j,giren,cikan,b:integer; s:string[10];scs:string[1]; n:byte;
      begin clrscr;gotoxy(3,23);
           write('KAC NUMARALI YEDEK PARCAYI GORMEK ISTIYORSUNUZ? > ');
```

```
intoku(53,23,3,sa);
WHILE sa in [1..maxnoofydk] do
begin clrscr;ydkform; seek(ydkfile,sa-1);read(ydkfile,ydkrec);
   WITH ydkrec do
      begin  if (geldate[4]=0) and (ydkfazla=0) then
         begin   giren:=0; cikan:=0;
            for j:=1 to ydkxa do begin if kullandate[j]<>0 then
            cikan:=cikan+1; end;
            for j:=1 to 3 do giren:=giren+gelmik[j];
            ydkfazla:=giren-cikan;
         end;
         gotoxy(12,3);write(isim);gotoxy(12,4);write(ycod);
         gotoxy(12,5);write(ynumber);gotoxy(38,3);write(mnlvl);
         gotoxy(38,4);write(birim);gotoxy(38,5);write(ucost);
         gotoxy(61,3);write(rorderq);gotoxy(61,4);write(leadt);
         gotoxy(61,5);write(makno);gotoxy(61,6);write(ydkfazla);
         gotoxy(27,20); if reptime<>0 then write(reptime);
         FOR j:=1 to 10 do
         begin gotoxy(8,9+j);write(geldate[j]:2);
            gotoxy(18,9+j);write(gelmik[j]:3);
            gotoxy(28,9+j);write(kimden[j]:10);
            gotoxy(49,9+j);write(kullandate[j]:2);
            gotoxy(61,9+j);write(tip[j]:1);
         end;
         if xza>10 then
         begin gotoxy(5,23);
            write('Devam etmek icin bir tusa basin. >');
            while not(keypressed) do delay(100);gotoxy(1,10);
            for j:=11 to 20 do writeln(j:2,' ':76);
            for j:=11 to 20 do
            begin
               gotoxy(49,j-1);write(kullandate[j]:2);
               gotoxy(61,j-1);write(tip[j]:1);
            end;
         end;
         if xza>20 then
         begin gotoxy(5,23);
            write('Devam etmek icin bir tusa basin. >');
            while not(keypressed) do delay(100);gotoxy(1,10);
            for j:=21 to 30 do writeln(j:2,' ':76);
            for j:=21 to 30 do
            begin
               gotoxy(49,j-11);write(kullandate[j]:2);
               gotoxy(61,j-11);write(tip[j]:1);
            end;
         end;    gotoxy(1,22);
         write(' BU YEDEK PARCAYA STOK GIRISI ');
          writeln('YAPMAK ICIN  1 ');
         write('PROGRAMSIZ MALZEME KULLANISI ISLEMEK ICIN');
          write('  2   girin > '); clreol;
         read(kbd,g); write(g);   b:=0; s:=''; scs:='';
         CASE g of
            '1':begin gotoxy(1,10);
            FOR j:=1 to 10 do
            begin
               write(j:2,' ':5,geldate[j]:2,' ':8,gelmik[j]:3);
                writeln(' ':7,kimden[j]:10,' ':41);b:=0;
            end;
            for j:=1 to 10 do
            begin  intoku(9,9+j,2,b);
               if b<>0 then geldate[j]:=b;b:=0;
```

```
                     intoku(20,9+j,3,b);
                     if b<>0 then
                     begin  gelmik[j]:=b;
                        ydkfazla:=ydkfazla+b;
                     end;        b:=0; s:='';
                     gotoxy(18,9+j);write(gelmik[j]:3,'  ');
                     blank(28,9+j,10);gotoxy(28,9+j);read(s);
                     if s<>'' then kimden[j]:=s; s:='';
                     gotoxy(28,9+j);write(kimden[j]:10,'  ');
                  end; end;
                  '2':begin  gotoxy(1,10); xza:=xza+1;
                  if xza>30 then xza:=1;
                  if xza<=10 then
                  begin   for j:=1 to 10 do
                     begin  write(j:2,' ':5,geldate[j]:2,' ':8);
                         write(gelmik[j]:3,' ':7,kimden[j]:10);
                        write(' ':11,kullandate[j]:2,' ':10);
                         writeln(tip[j]:1,' ':11);
                     end;  b:=0;
                     intoku(50,9+xza,2,b);
                     if b<>0 then
                      kullandate[xza]:=b; b:=0;
                     blank(61,9+xza,1);gotoxy(61,9+xza); read(scs);
                     if scs<>'' then tip[xza]:=scs; scs:='';
                     ydkfazla:=ydkfazla-1;
                  end;
                  if (xza>10) and (xza<=20) then
                  begin     for j:=11 to 20 do
                     begin write(j:2,' ':46,kullandate[j]:2,' ':10);
                         writeln(tip[j]:1,' ':11);b:=0;
                     end; intoku(50,xza-1,2,b);
                  if b<>0 then  kullandate[xza]:=b;
                  b:=0;  blank(61,xza-1,1); gotoxy(61,xza-1);
                  read(scs);if scs<>'' then tip[xza]:=scs; scs:='';
                  ydkfazla:=ydkfazla-1;
               end;
               if (xza>20) and (xza<=30) then
               begin  for j:=21 to 30 do
                  begin  write(j:2,' ':46,kullandate[j]:2,' ':10);
                     writeln(tip[j]:1,' ':11);b:=0;
                  end;  intoku(50,xza-11,2,b);
                  if b<>0 then   kullandate[xza]:=b; b:=0;
                  blank(61,xza-11,1);gotoxy(61,xza-11); read(scs);
                  if scs<>'' then tip[xza]:=scs; scs:='';
                  ydkfazla:=ydkfazla-1;
               end;
               end   {case: 2}
            end  {case}
         end;   {with ydkrec}
         seek(ydkfile,sa-1); write(ydkfile,ydkrec); gotoxy(1,22);
         clreol; gotoxy(1,23);clreol;gotoxy(3,23); sa:=sa+1;
         writeln('GORMEK ISTEDIGINIZ YEDEK PARCA Nosunu GIRIN. > ');
         write('    (Bir sonraki icin <RETURN> ;  0=DUR  )');
         intoku(50,23,3,sa);
      end;
   end;

procedure rapor;
   var mes,normesai,fzmesai,fimak,imak,bakismal,yparmal,fmismal,
      imakmal,fimakmal:real;     j:integer;
```

```
procedure hardcopy;
   begin writeln(lst,' ':29,' BAKIM  MUHENDISLIGI'); j:=0;
     WRITE(LST,' ':24,CW-VA+1:2,'..',CW:2,'. HAFTALAR DURUM ');
      WRITELN(LST,'RAPORU');WRITELN(lst);
     WRITE(LST,'     Normal Mesai',' ':39);
      WRITELN(LST,va*weekhour*(en+mn+un):5,'     saat');
     write(lst,'     Makina Bakimlarina Harcanan Normal ');
      writeln(lst,'Mesai',' ':11,normesai/60:8:2,'  saat');
     WRITE(LST,'     Makina Bakimlarina Harcanan Fazla  ');
      WRITELN(LST,'Mesai',' ':11,FZMESAI/60:8:2,'  saat');
     write(lst,'     Aylak Sure (Diger islere harcanan ');
      write(lst,'mesai)',' ':11);
     if (va*weekhour*(en+mn+un)-normesai/60)>0 then
     writeln(lst,va*weekhour*(en+mn+un)-normesai/60:8:2,'  saat')
      else  writeln(lst,j:8,'  saat');
     WRITE(LST,'     Toplam Imalat Aksamasi',' ':29);
      WRITELN(LST,(IMAK+FIMAK)/60:8:2,'  saat');
     write(lst,'     Periodik Bakimlardan Dolayi Imalat ');
      writeln(lst,'Aksamasi',' ':8,imak/60:8:2,'  saat');
     WRITE(LST,'     Arizi Bakimlardan Dolayi Imalat ');
      WRITELN(LST,'Aksamasi',' ':11,FIMAK/60:8:2,'  saat');
     WRITELN(lst);
     write(lst,'       **     **     **     **     **  ');
      writeln(lst,'   **     **     **');writeln(lst);
     write(lst,'     Makina Bakimlarina Harcanan Normal ');
      writeln(lst,'Mesai Maliyeti  ',bakismal/10:8:2,'  binTL');
     WRITE(LST,'     Makina Bakimlarina Harcanan Fazla Mesai');
      WRITELN(LST,' Maliyeti   ',fmismal/10:8:2,'  binTL');
     write(lst,'     Bakima Harcanan Toplam Iscilik Maliyeti');
      writeln(lst,' ':12,(bakismal+fmismal)/10:8:2,'  binTL');
     WRITE(LST,'     Degisen Yedek Parcalarin Maliyeti');
      WRITELN(LST,' ':18,YPARMAL/10:8:2,'  binTL');
     write(lst,'     TOPLAM DIREKT GIDERLER',' ':29);
      writeln(lst,(yparmal+bakismal+fmismal)/10:8:2,'  binTL');
     WRITE(LST,'     Periodik Bakimlardan Aksayan Imalatin ');
      WRITELN(LST,'Maliyeti',' ':5,IMAKMAL/10:8:2,'  binTL');
     write(lst,'     Arizi Bakimlardan Aksayan Imalatin ');
      writeln(lst,'Maliyeti',' ':8,fimakmal/10:8:2,'  binTL');
     WRITELN(lst);WRITELN(lst);
     WRITE(LST,'     TOPLAM  MALIYET .............');
      WRITE(LST,'................      ');
   write(lst,(yparmal+bakismal+fmismal+imakmal+fimakmal)/10:8:2);
      writeln(lst,'  binTL');
     writeln(lst,' ':55,'=================');
  end;

begin clrscr; va:=1; reset(machinefile); gotoxy(20,3);
      write(' KAC HAFTALIK RAPOR ISTIYORSUNUZ ? > ');
      intoku(58,3,2,va);
      while not (va in [1..52]) do
      begin gotoxy(30,20); write('Lutfen tekrar girin [1.52] ');
            intoku(58,3,2,va);
      end;
      gotoxy(20,5);
      write(' KACINCI HAFTAYA KADAR RAPOR ISTIYORSUNUZ ? > ');
      intoku(66,5,2,cw);
      while not (cw in [1..52]) do
      begin gotoxy(30,20); write('Lutfen tekrar girin [1.52] ');
            intoku(66,5,2,cw);
      end;
```

```
clrscr; gotoxy(30,1); lowvideo;
write(' BAKIM MUHENDISLIGI '); gotoxy(25,2);
write('"',cw-va+1:2,'..',cw:2,'". HAFTALAR DURUM RAPORU');
highvideo; gotoxy(1,4);
write('Normal Mesai',' ':47,va*weekhour*(en+mn+un):5);
 writeln('    saat');
  normesai:=0; fzmesai:=0; imak:=0; fimak:=0; bakismal:=0;
  yparmal:=0; fmismal:=0; imakmal:=0; fimakmal:=0;
for i:=1 to maxnoofmachines do
begin read(machinefile,machinerec);
  if machinerec.maindate[1]>0 then
  begin with machinerec do
     begin for j:=1 to xa do
        begin
           if (maindate[j]>cw-va) and (maindate[j]<=cw) then
           begin
              normesai:=normesai+maintdur[j]*(elektrikci[j]+
               mekanikci[j]+usta[j]);
              fzmesai:=fzmesai+fmmaindur[j]*(elektrikci[j]+
               mekanikci[j]+usta[j]);
              if (maintype[j]='p') or (maintype[j]='P') then
              begin imak:=imak+failuredur[j];
              imakmal:=imakmal+failuredur[j]*costpminlost;
              end; if (maintype[j]='a') or
               (maintype[j]='A') then
              begin fimak:=fimak+failuredur[j];
              fimakmal:=fimakmal+failuredur[j]*costpminlost;
              end; yparmal:=yparmal+parcacost[j];
          bakismal:=bakismal+maintdur[j]*
           ((elektrikci[j]+mekanikci[j])*dic+usta[j]*duc);
          fmismal:=fmismal+fmmaindur[j]*
           ((elektrikci[j]+mekanikci[j])*dic+usta[j]*duc);
               end;   end;   end;   end;
end; gotoxy(1,5);  j:=0;
write('Makina Bakimlarina Harcanan Normal Mesai');
 writeln(' ':19,normesai/60:8:2,'  saat');
write('Makina Bakimlarina Harcanan Fazla  Mesai');
 writeln(' ':19,fzmesai/60:8:2,'  saat');
write('Aylak Sure (Diger islere harcanan mesai)',' ':19);
if (va*weekhour*(en+mn+un)-normesai/60)>0 then
writeln(va*weekhour*(en+mn+un)-normesai/60:8:2,'  saat')
 else     writeln(j:8,'  saat');
write('Toplam Imalat Aksamasi',' ':37);
 writeln((imak+fimak)/60:8:2,'  saat');
write('Periodik Bakimlardan Dolayi Imalat Aksamasi');
 writeln(' ':16,imak/60:8:2,'  saat');
write('Arizi Bakimlardan Dolayi Imalat Aksamasi');
 writeln(' ':19,fimak/60:8:2,'  saat');writeln;
write('    **    **     **    **     **     **   ');
 writeln('  **     **'); writeln;
write('Makina Bakimlarina Harcanan Normal Mesai ');
 writeln('Maliyeti',' ':10,bakismal/10:8:2,'  binTL');
write('Makina Bakimlarina Harcanan Fazla Mesai ');
 writeln('Maliyeti',' ':11,fmismal/10:8:2,'  binTL');
write('Bakima Harcanan Toplam Iscilik Maliyeti');
 writeln(' ':20,(bakismal+fmismal)/10:8:2,'  binTL');
write('Degisen Yedek Parcalarin Maliyeti');
 writeln(' ':26,yparmal/10:8:2,'  binTL');
write('TOPLAM DIREKT GIDERLER',' ':37);
 writeln((yparmal+bakismal+fmismal)/10:8:2,'  binTL');
```

```
      write('Periodik Bakimlardan Aksayan Imalatin ');
       writeln('Maliyeti',' ':13,imakmal/10:8:2,'  binTL');
      write('Arizi Bakimlardan Aksayan Imalatin Maliyeti');
       writeln(' ':16,fimakmal/10:8:2,'  binTL');
      writeln;writeln;
      write('TOPLAM  MALIYET........................',' ':20);
      write((yparmal+bakismal+fmismal+imakmal+fimakmal)/10:8:2);
       writeln('  binTL
      write(' ':58,'=================');  gotoxy(3,24);
      write('Dokum Almak Ister misiniz ? (E) > ');read(kbd,g);
      write(g); if g in ['e','E'] then hardcopy;
end;
```

```
procedure ydksiparis;
   var sa,b:integer; ydkA,ydkC,ydkt,ydkQ,ydkD,ydkr:real;
   begin ch:='E';
      WHILE ch in ['E','e','*'] do
      begin clrscr;
         write('Bu secenek bir Yedek Parcanin optimal siparis ');
          wrieln('miktari ve optimal minimum');
         write('stok miktarini onerir. Yedek Parca Nosunu girin. > ');
         intoku(52,2,3,sa); ydkC:=0; ydkt:=0;
         if sa in [1..maxnoofydk] then
         begin
            seek(ydkfile,sa-1);read(ydkfile,ydkrec);
            WITH ydkrec do
            begin while reptime=0 do
               begin gotoxy(1,5);  b:=0;
                  write('Bu Yedek Parca Icin Onerilen Degisim ');
                   write('Zamanini Girin. > ');
                  intoku(56,5,2,b); reptime:=b;
               end;
               gotoxy(1,7);  ydkA:=0;  ydkC:=ucost*100;
               write('Bir Siparis Vermenin Sabit ');
                write('Maliyetini Giriniz. > ');
               realoku(50,7,8,ydkA); gotoxy(1,8);
               write('ASAGIDAKILER GECERLI ISE <RETURN> ,DEGIL');
                writeln(' ISE DOGRU DEGERLERI GIRINIZ.');
               writeln('           Birim Maliyet (TL) : ',ydkC:7:0);
               ydkC:=0;
               writeln('Siparis - Gelis suresi (hafta) : ',leadt);
               realoku(41,9,8,ydkC);
               if ydkC<>0 then ucost:=round(ydkC/100);
               realoku(41,10,8,ydkt);
               if ydkt<>0 then leadt:=round(ydkt);
               ydkD:=52/reptime;
               ydkQ:=sqrt(2*ydkA*ydkD/(hi*ucost*100));
               if ydkQ=0 then ydkQ:=1;
               ydkr:=leadt*ydkD-ydkQ*trunc(leadt*ydkD/ydkQ);
               gotoxy(1,13);
               writeln('" ',sa,' " Nolu Yedek Parcanin :');
               write('Optimal Siparis Miktari  ');
                writeln(round(ydkQ),'  ',birim);
               write('  Minimum Stok Miktari  ',round(ydkr));
                writeln('  ',birim,'   olarak bulundu.');
               write('Bu Degerleri Kabul Etmek Ister misiniz.');
                write('(E,*)  > ');
               read(kbd,g); write(g); if g in ['E','e','*'] then
               begin
                    rorderq:=round(ydkQ); mnlvl:=round(ydkr);
               end;
               end; seek(ydkfile,sa-1); write(ydkfile,ydkrec);
             end;   gotoxy(3,20);
          write('BASKA BIR YEDEK PARCA ICIN CALISACAK MISINIZ?');
           write(' (E,*) > ');  read(kbd,ch); write(ch);
       end;
   end;
procedure periodbul;
   var sa,enkucuk:integer;
   begin ch:='e';
      WHILE ch in ['e','E','*'] do
      begin clrscr; gotoxy(1,3);
         writeln('Bu secenek bir makinanin optimal bakim periodunun');
         writeln('kac hafta olmasi gerektigini onerir.');writeln;
         writeln; write(' BAKIM PERIODUNU BULMAK ISTEDIGINIZ ');
          write('MAKINA Nosunu GIRIN. > ');
         intoku(60,7,3,sa);
         if sa in [1..maxnoofmachines] then
         begin
            reset(machinefile); reset(ydkfile); enkucuk:=100;
            FOR i:=1 to maxnoofydk do
            begin seek(ydkfile,i-1); read(ydkfile,ydkrec);
               if sa=ydkrec.makno then
               begin if ydkrec.reptime=0 then agebul;
                  seek(ydkfile,i-1); read(ydkfile,ydkrec);
                  if ydkrec.reptime<enkucuk then
                   enkucuk:=ydkrec.reptime;
               end;
            end;   gotoxy(1,10);
            write(sa,' Nolu makinanin onerilen bakim periodu ');
             writeln(enkucuk,' haftadir.');
            write('Bunu bakim periodu olarak kabul etmek ister ');
             write('misiniz? (E,*) > ');
            read(kbd,g); write(g); if g in ['E','e','*'] then
            begin seek(machinefile,sa-1);
               read(machinefile,machinerec);
               machinerec.mainperiod:=enkucuk;
               seek(machinefile,sa-1);
               write(machinefile,machinerec);
            end;
         end; gotoxy(1,20); write('Baska bir makinanin bakim ');
          write('periodunu bulmak ister misiniz? (E,*) > ');
         read(kbd,ch); write(ch);
      end;
   end;

procedure yedekgor; var sa,j,b:integer; ss,s:string[15];
 begin
repeat  clrscr;  gotoxy(1,13); writeln;
writeln('0> DOSYADAKI YEDEK PARCA BILGILERINI SILMEK');
writeln('1> YEDEK PARCALARI    GORMEK/ISLEMEK');
writeln('2> ADI/KODU BILINEN YEDEK PARCANIN DOSYA NUMARASINI BULMAK');
writeln('3> ONERILEN DEGISIM SURESINI BULMAK ICIN');
writeln('4> SIPARIS MIKTARI ve MINIMUM STOK SEVIYESINI BULMAK');
writeln('5> DOSYA/KOD NoLARINI GORMEK');
writeln('6> YEDEK PARCA KARTLARINI YAZMAK');
writeln('7> STOKTA MEVCUT YEDEK PARCALARIN PARASAL DEGERINI BULMAK');
writeln('8> ANA MENUYE DONMEK'); writeln(' > ');
        read(kbd,ch); write(ch);
        while not (ch in ['0'..'8']) do
        begin gotoxy(4,23);clreol;
              read(kbd,ch); write(ch);
        end;
        CASE ch of
        '0':BEGIN  CLRSCR;GOTOXY(3,22);
               WRITE('BU SECENEGI KULLANABILMEK ICIN SIFREYI ');
                WRITE('GIRMELISINIZ ! '); READ(KBD,CH); write(CH);
               IF CH IN ['Z','z'] THEN YDKBASLA;
            END;
        '1':ydkgor;
        '2':ydkara;
        '3':agebul;
        '4':ydksiparis;
        '5':ydklist(ydkfile,ydkrec);
        '6':ydkyaz;
        '7':masraf
        end
until ch ='8';
end;
procedure basla;
   var j,i,ii:integer;g:char;
   BEGIN  CLRSCR;LOWVIDEO;
      GOTOXY(5,4);WRITE('  D I K K A T  ');
      HIGHVIDEO;GOTOXY(5,7);
      WRITE('PROGRAMIN HAFIZASINDAKI BUTUN MAKINA BILGILERINI ');
      GOTOXY(5,9);WRITE('SILMEK UZERESINIZ...');
      REPEAT GOTOXY(5,11);CLREOL;
         WRITE('ISTEDIGINIZ GERCEKTEN BU MU ? (E / H)  > ');
         READ(KBD,G); WRITE(G);
      UNTIL G IN ['E','e','H','h'];
      IF G IN ['e','E'] THEN
   begin  gotoxy(50,20); write('Lutfen Bekleyiniz ');
      rewrite(machinefile); i:=1;
      while i<=maxnoofmachines do
      begin seek(machinefile,i-1);
      with machinerec do
      begin name:=''; codeno:=0;number:=i;costpminlost:=0;
         mainperiod:=0;revperiod:=0;   mza:=0;  fazla:=0;
         for ii:=1 to xa do
         begin maindate[ii]:=0;maintype[ii]:=' ';emniyet[ii]:=0;
            elektrikci[ii]:=0;mekanikci[ii]:=0;
            parcacost[ii]:=0;
            usta[ii]:=0;failuredur[ii]:=0; maintdur[ii]:=0;
            fmmaindur[ii]:=0;fcost[ii]:=0;pcost[ii]:=0;
            for j:=1 to 5 do sparesused[ii,j]:=0;
         end; fortcost:=0; availability:=0;topimak:=0;
         ortus:=0;ortdursur:=0;ortel:=0;ortmek:=0;
         ortbaksur:=0;   write(machinefile,machinerec);
      end;  i:=i+1;
   end;
end;
END;

procedure sabitdegistir;
   begin  clrscr;gotoxy(3,3);
      writeln('BU SECENEK PROGRAM ICINDEKI SABIT SAYILARIN');
      write('  DEGERLERINI  DEGISTIRMEK  ICIN  KULLANILIR. ');
      gotoxy(37,8);write('eski   yeni');  gotoxy(3,19);
      writeln('AYNI KALMASINI ISTEDIGINIZ DEGERLER ICIN <RETURN>');
      write('  DEGISTIRMEK ISTEDIKLERINIZ ICIN DOGRU DEGERLERI GIRIN');
      seek(sabitfile,0); read(sabitfile,sabitrec);with sabitrec do
      begin       gotoxy(3,9);
         writeln('HAFTALIK CALISMA SURESI (dakika):',workweek:5);
         writeln('  TOPLAM USTA ELEMAN SAYISI        :',un:5);
         writeln('  TOPLAM ELEKTRIKCI SAYISI         :',en:5);
         writeln('  TOPLAM MEKANIKCI SAYISI          :',mn:5);
         writeln('  DAKIKALIK ISCI UCRETI (net TL)   :',dic:5);
         writeln('  DAKIKALIK USTA UCRETI (net TL)   :',duc:5);
         writeln('  HAFTALIK CALISMA SAATI           :',weekhour:5);
         i:=0; intoku(43,9,4,i); if i<>0 then sabitler[1]:=i; i:=0;
         workweek:=sabitler[1]; gotoxy(43,9); write(workweek:5);
         intoku(43,10,4,i); if i<>0 then sabitler[2]:=i; i:=0;
         un:=sabitler[2];   gotoxy(43,10);write(un:5);
         intoku(43,11,4,i); if i<>0 then sabitler[3]:=i; i:=0;
         en:=sabitler[3];  gotoxy(43,11);write(en:5);
         intoku(43,12,4,i); if i<>0 then sabitler[4]:=i; i:=0;
         mn:=sabitler[4];  gotoxy(43,12);write(mn:5);
         intoku(43,13,4,i); if i<>0 then sabitler[5]:=i; i:=0;
         dic:=sabitler[5];   gotoxy(43,13);write(dic:5);
         intoku(43,14,4,i); if i<>0 then sabitler[6]:=i; i:=0;
         duc:=sabitler[6];  gotoxy(43,14);write(duc:5);
         intoku(43,15,4,i); if i<>0 then sabitler[7]:=i; i:=0;
         weekhour:=sabitler[7];  gotoxy(43,15);write(weekhour:5);
      end;seek(sabitfile,0); write(sabitfile,sabitrec);
         gotoxy(1,19);writeln;writeln;
         writeln('  Devam etmek icin bir tusa basin > ');
         while not(keypressed) do delay(100);
   end;

PROCEDURE ANAMENU; VAR G:INTEGER;
    BEGIN CLRSCR;LOWVIDEO;GOTOXY(16,2);
          WRITE(' ETKILESIMLI BAKIM PLANLAMASI PROGRAMI ');
          GOTOXY(28,5);WRITE(' SECENEKLER ');HIGHVIDEO;
          GOTOXY(20,7);WRITE('S> PROGRAM SABITLERINI DEGISTIRME');
          GOTOXY(20,8);WRITE('0> YENI BASTAN BASLAMA');
          GOTOXY(20,9);WRITE('1> BAKILACAK MAKINALARI SAPTAMA');
          GOTOXY(20,10);WRITE('2> ADI/KODU BILINEN MAKINALARIN');
          GOTOXY(20,11);WRITE('   DOSYA NUMARASINI BULMAK');
          GOTOXY(20,12);
          WRITE('3> ORTALAMA BAKIM SURESI,PERSONEL IHTIYACI VE');
          GOTOXY(20,13);WRITE('   MALIYET HESABI YENILEME');
          GOTOXY(20,14);WRITE('4> YEDEK PARCA STOKLARINI GORME');
          GOTOXY(20,15);WRITE('5> MAKINA LISTESINI GORME');
          GOTOXY(20,16);WRITE('6> MAKINALARI / BAKIMLARI ISLEME');
          GOTOXY(20,17);WRITE('7> BAKIM PERIODU SAPTAMA');
          GOTOXY(20,18);WRITE('8> BAKIM FAALIYETLERI RAPORU ALMA');
          GOTOXY(20,19);
          WRITE('9> PERIODIK BAKIM FALIYETI PERSONEL HESABI');
          GOTOXY(20,20);WRITE('Q> PROGRAMDAN CIKIS');
          REPEAT GOTOXY(19,22);CLREOL;LOWVIDEO;
                 WRITE('> ');HIGHVIDEO;
                 GOTOXY(22,22);
                 READ(KBD,G); WRITE(G);
          UNTIL  G IN ['0'..'9','Q','q','S','s'];
          CASE G OF
              '0': BEGIN GOTOXY(3,22);
       WRITE('BU SECENEGI KULLANABILMEK ICIN SIFREYI GIRMELISINIZ. ');
                        READ(KBD,CH); write(CH);
                        IF CH IN ['Z','z'] THEN BASLA;
                   END;
              '1':MACSEC;
              'S':SABITDEGISTIR;
              's':SABITDEGISTIR;
              '7':PERIODBUL;
              '3':ORTBUL;
              '4':YEDEKGOR;
              '8':RAPOR;
              '6':MAKINAYAZ;
              '2':ARA;
              '9':KACKISILAZIM;
              '5':LISTE(MACHINEFILE,MACHINEREC)
       END
  END;

BEGIN    {Program Makina}
    clrscr;gotoxy(25,8);
    writeln('----------------------------');
    writeln(' ':24,'BAKIM  PLANLAMASI  PROGRAMI');
    writeln(' ':24,'----------------------------');
    gotoxy(28,13);writeln('BOGAZICI  UNIVERSITESI');
    writeln(' ':27,'ENDUSTRI  MUHENDISLIGI');
    writeln(' ':34,'Bolumu');
    writeln;write(' ':29,'VER 1.0  MAYIS 1985');
      ASSIGN(MACHINEFILE,'B:NAKINA.REK');
    {$I-} RESET(MACHINEFILE) {$I+}; ASSIGN(YDKFILE,'B:YEDEK.PAR');
    {$I-} RESET(YDKFILE) {$I+};  ASSIGN(sabitfile,'B:prog.sab');
    {$I-} reset(sabitfile) {$I+};    ortbul;
    seek(sabitfile,0); read(sabitfile,sabitrec);with sabitrec do
    begin
     workweek:=sabitler[1];un:=sabitler[2];
     en:=sabitler[3];mn:=sabitler[4];
     dic:=sabitler[5]; duc:=sabitler[6];weekhour:=sabitler[7];
    end;
    REPEAT
         ANAMENU
    UNTIL  G in ['Q','q'];
    CLOSE(MACHINEFILE);   CLOSE(YDKFILE);  close(sabitfile);
END.
```

APPENDIX D

COMPUTER OUTPUTS FOR THE SELECTED FACTORY
REFLECTING MONTHLY MAINTENANCE REPORTS
DURING A SIX MONTH STUDY

```
                          BAKIM  MUHENDISLIGI
                      1.. 4. HAFTALAR DURUM RAPORU

Normal Mesai                                         3240       saat
Makina Bakimlarina Harcanan Normal Mesai               20.92    saat
Makina Bakimlarina Harcanan Fazla  Mesai                8.00    saat
Aylak Sure (Diger islere harcanan mesai)             3219.08    saat
Toplam Imalat Aksamasi                                 11.47    saat
Periodik Bakimlardan Dolayi Imalat Aksamasi             4.00    saat
Arizi Bakimlardan Dolayi Imalat Aksamasi                7.47    saat

   **     **     **     **     **     **     **     **

Makina Bakimlarina Harcanan Normal Mesai Maliyeti    1363.00   binTL
Makina Bakimlarina Harcanan Fazla Mesai Maliyeti      576.00   binTL
Bakima Harcanan Toplam Iscilik Maliyeti              1939.00   binTL
Degisen Yedek Parcalarin Maliyeti                     181.60   binTL
TOPLAM DIREKT GIDERLER                               2120.60   binTL
Periodik Bakimlardan Aksayan Imalatin Maliyeti        660.00   binTL
Arizi Bakimlardan Aksayan Imalatin Maliyeti          2534.00   binTL


TOPLAM  MALIYET .............................        5314.60   binTL
                                                  ================


                          BAKIM  MUHENDISLIGI
                      5.. 8. HAFTALAR DURUM RAPORU

Normal Mesai                                         3240       saat
Makina Bakimlarina Harcanan Normal Mesai               30.73    saat
Makina Bakimlarina Harcanan Fazla  Mesai                0.00    saat
Aylak Sure (Diger islere harcanan mesai)             3209.27    saat
Toplam Imalat Aksamasi                                 12.93    saat
Periodik Bakimlardan Dolayi Imalat Aksamasi             6.17    saat
Arizi Bakimlardan Dolayi Imalat Aksamasi                6.77    saat

   **     **     **     **     **     **     **     **

Makina Bakimlarina Harcanan Normal Mesai Maliyeti    1976.80   binTL
Makina Bakimlarina Harcanan Fazla Mesai Maliyeti        0.00   binTL
Bakima Harcanan Toplam Iscilik Maliyeti              1976.80   binTL
Degisen Yedek Parcalarin Maliyeti                      51.80   binTL
TOPLAM DIREKT GIDERLER                               2028.60   binTL
Periodik Bakimlardan Aksayan Imalatin Maliyeti        880.00   binTL
Arizi Bakimlardan Aksayan Imalatin Maliyeti          1637.00   binTL


TOPLAM  MALIYET .............................        4545.60   binTL
                                                  ================
                         BAKIM  MUHENDISLIGI
                     9..12. HAFTALAR DURUM RAPORU

Normal Mesai                                             3240       saat
Makina Bakimlarina Harcanan Normal Mesai                   34.43    saat
Makina Bakimlarina Harcanan Fazla  Mesai                   13.67    saat
Aylak Sure (Diger islere harcanan mesai)                 3205.57    saat
Toplam Imalat Aksamasi                                     14.05    saat
Periodik Bakimlardan Dolayi Imalat Aksamasi                 2.37    saat
Arizi Bakimlardan Dolayi Imalat Aksamasi                   11.68    saat

   **     **     **     **     **     **     **     **

Makina Bakimlarina Harcanan Normal Mesai Maliyeti   2325.20   binTL
Makina Bakimlarina Harcanan Fazla Mesai Maliyeti     984.00   binTL
Bakima Harcanan Toplam Iscilik Maliyeti             3309.20   binTL
Degisen Yedek Parcalarin Maliyeti                    279.30   binTL
TOPLAM DIREKT GIDERLER                              3588.50   binTL
Periodik Bakimlardan Aksayan Imalatin Maliyeti      1444.00   binTL
Arizi Bakimlardan Aksayan Imalatin Maliyeti         2267.50   binTL


TOPLAM  MALIYET ..............................      7300.00   binTL
                                                  ================
```

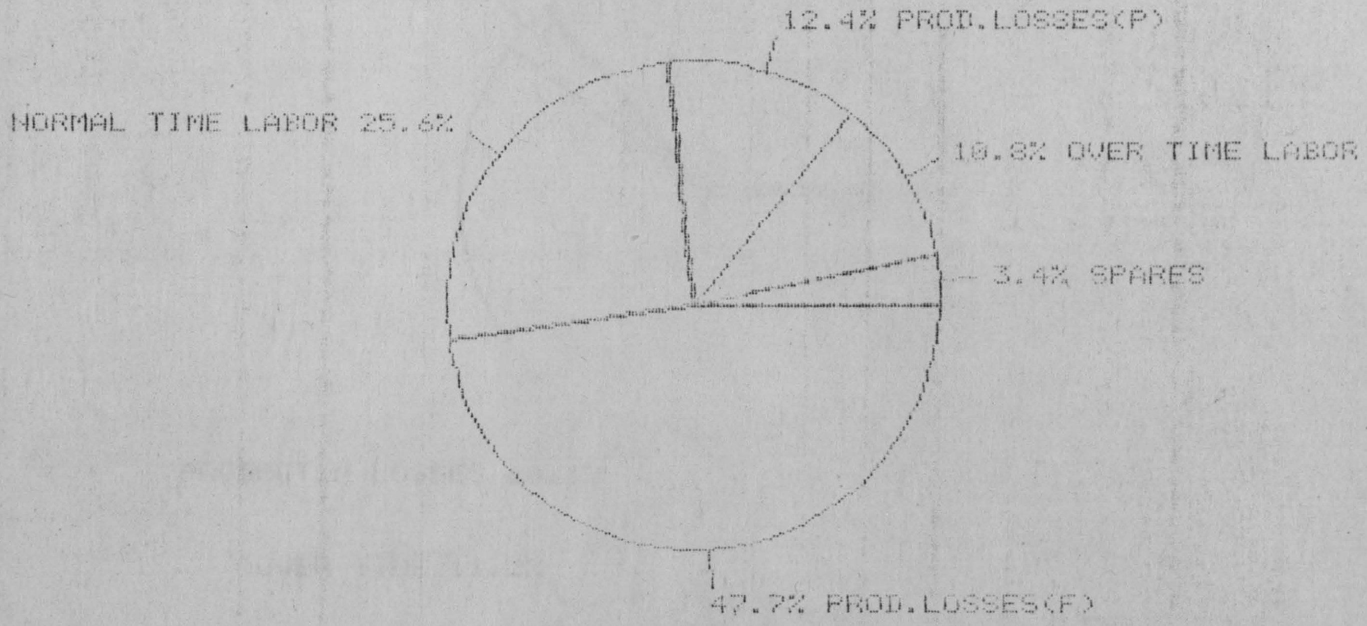
```
                         BAKIM  MUHENDISLIGI
                    13..16. HAFTALAR DURUM RAPORU

Normal Mesai                                             3240       saat
Makina Bakimlarina Harcanan Normal Mesai                   15.92    saat
Makina Bakimlarina Harcanan Fazla  Mesai                    0.83    saat
Aylak Sure (Diger islere harcanan mesai)                 3224.08    saat
Toplam Imalat Aksamasi                                      7.62    saat
Periodik Bakimlardan Dolayi Imalat Aksamasi                 5.10    saat
Arizi Bakimlardan Dolayi Imalat Aksamasi                    2.52    saat

   **     **     **     **     **     **     **     **

Makina Bakimlarina Harcanan Normal Mesai Maliyeti   1046.20   binTL
Makina Bakimlarina Harcanan Fazla Mesai Maliyeti      50.00   binTL
Bakima Harcanan Toplam Iscilik Maliyeti             1096.20   binTL
Degisen Yedek Parcalarin Maliyeti                     53.50   binTL
TOPLAM DIREKT GIDERLER                              1149.70   binTL
Periodik Bakimlardan Aksayan Imalatin Maliyeti       902.00   binTL
Arizi Bakimlardan Aksayan Imalatin Maliyeti          453.00   binTL


TOPLAM  MALIYET ..............................      2504.70   binTL
                                                  ================
                         BAKIM  MUHENDISLIGI
                    17..20. HAFTALAR DURUM RAPORU

Normal Mesai                                              3240      saat
Makina Bakimlarina Harcanan Normal Mesai                    35.98   saat
Makina Bakimlarina Harcanan Fazla  Mesai                    15.33   saat
Aylak Sure (Diger islere harcanan mesai)                  3204.02   saat
Toplam Imalat Aksamasi                                      13.65   saat
Periodik Bakimlardan Dolayi Imalat Aksamasi                  6.65   saat
Arizi Bakimlardan Dolayi Imalat Aksamasi                     7.00   saat

    **        **        **        **        **        **        **        **

Makina Bakimlarina Harcanan Normal Mesai Maliyeti         2572.60   binTL
Makina Bakimlarina Harcanan Fazla Mesai Maliyeti          1128.00   binTL
Bakima Harcanan Toplam Iscilik Maliyeti                   3700.60   binTL
Degisen Yedek Parcalarin Maliyeti                          185.10   binTL
TOPLAM DIREKT GIDERLER                                    3885.70   binTL
Periodik Bakimlardan Aksayan Imalatin Maliyeti            2546.00   binTL
Arizi Bakimlardan Aksayan Imalatin Maliyeti               2100.00   binTL


TOPLAM  MALIYET ..............................            8531.70   binTL
                                                      ================
```

```
                         BAKIM  MUHENDISLIGI
                    21..24. HAFTALAR DURUM RAPORU

Normal Mesai                                              3240      saat
Makina Bakimlarina Harcanan Normal Mesai                    27.28   saat
Makina Bakimlarina Harcanan Fazla  Mesai                     6.27   saat
Aylak Sure (Diger islere harcanan mesai)                  3212.72   saat
Toplam Imalat Aksamasi                                      13.63   saat
Periodik Bakimlardan Dolayi Imalat Aksamasi                  1.72   saat
Arizi Bakimlardan Dolayi Imalat Aksamasi                    11.92   saat

    **        **        **        **        **        **        **        **

Makina Bakimlarina Harcanan Normal Mesai Maliyeti         1949.80   binTL
Makina Bakimlarina Harcanan Fazla Mesai Maliyeti           526.40   binTL
Bakima Harcanan Toplam Iscilik Maliyeti                   2476.20   binTL
Degisen Yedek Parcalarin Maliyeti                           98.60   binTL
TOPLAM DIREKT GIDERLER                                    2574.80   binTL
Periodik Bakimlardan Aksayan Imalatin Maliyeti             309.00   binTL
Arizi Bakimlardan Aksayan Imalatin Maliyeti               3080.00   binTL


TOPLAM  MALIYET ..............................            5963.80   binTL
                                                      ================
```

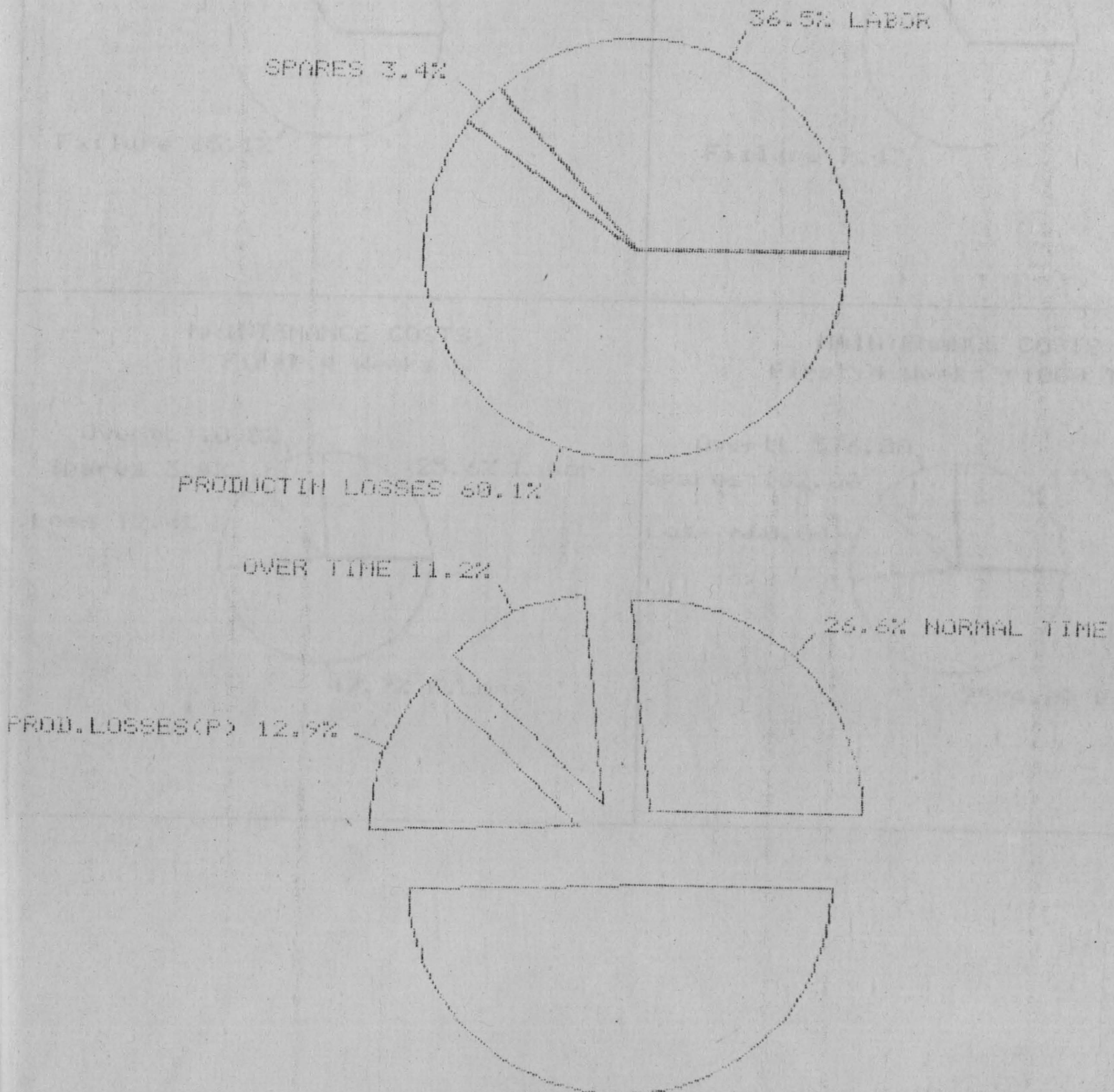
APPENDIX E

GRAPHICAL REPRESENTATIONS OF MAINTENANCE COSTS

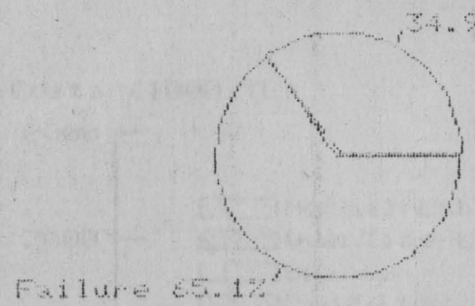
MAINTENANCE COSTS
First 4 Weeks



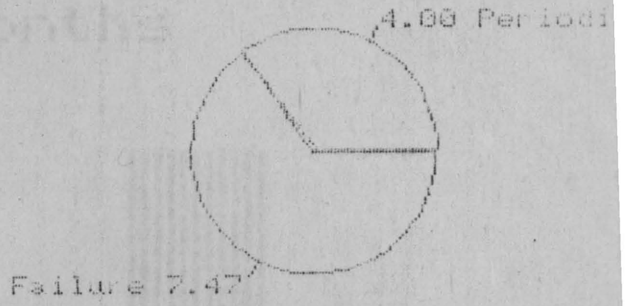
MAINTENANCE COSTS
FIRST 4 WEEKS



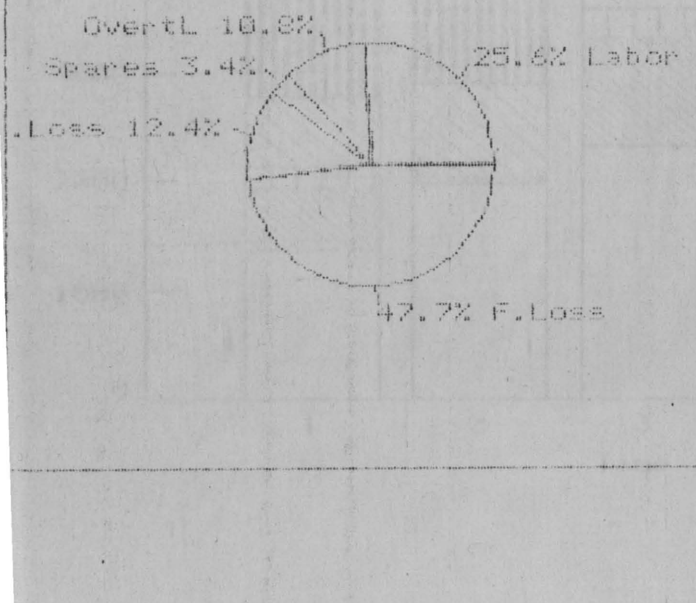
PRODUCTION TIME LOSSES
First 4 Weeks



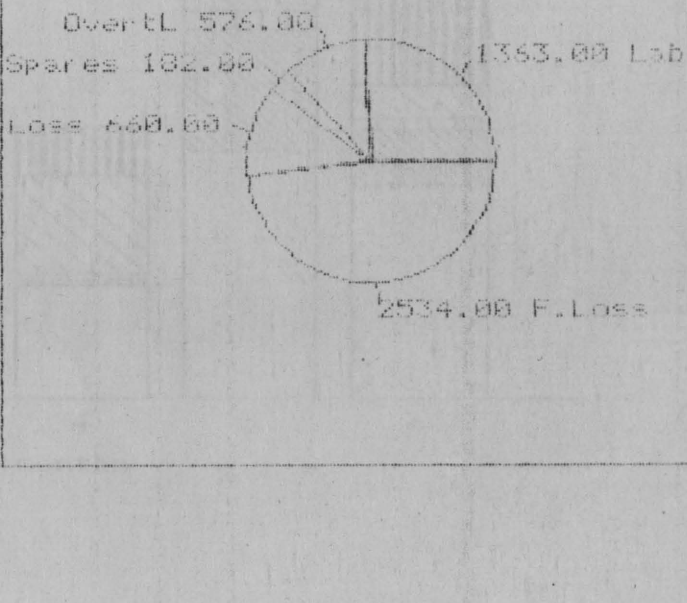
PRODUCTION TIME LOSSES
First 4 Weeks (hours)



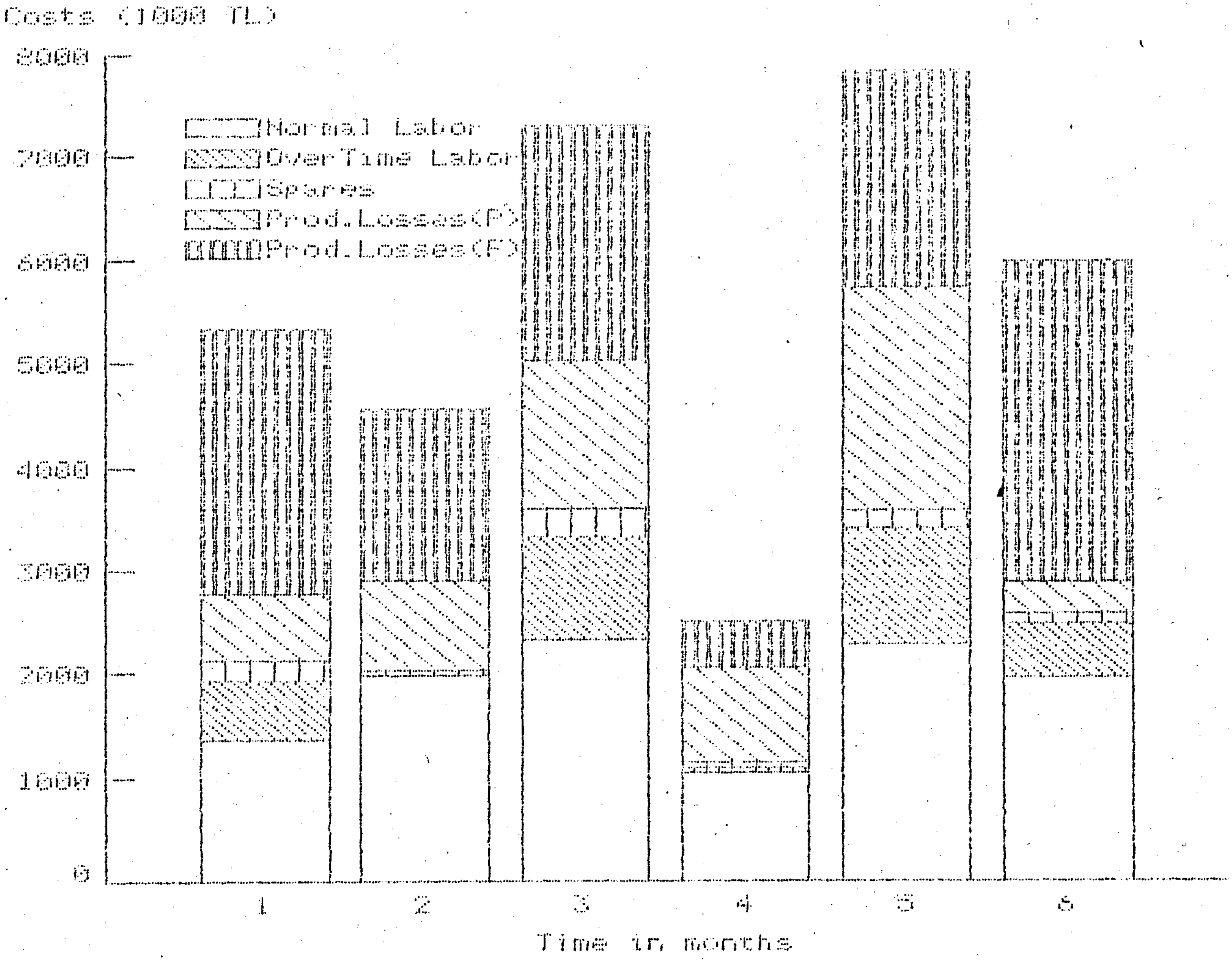
MAINTENANCE COSTS
First 4 Weeks

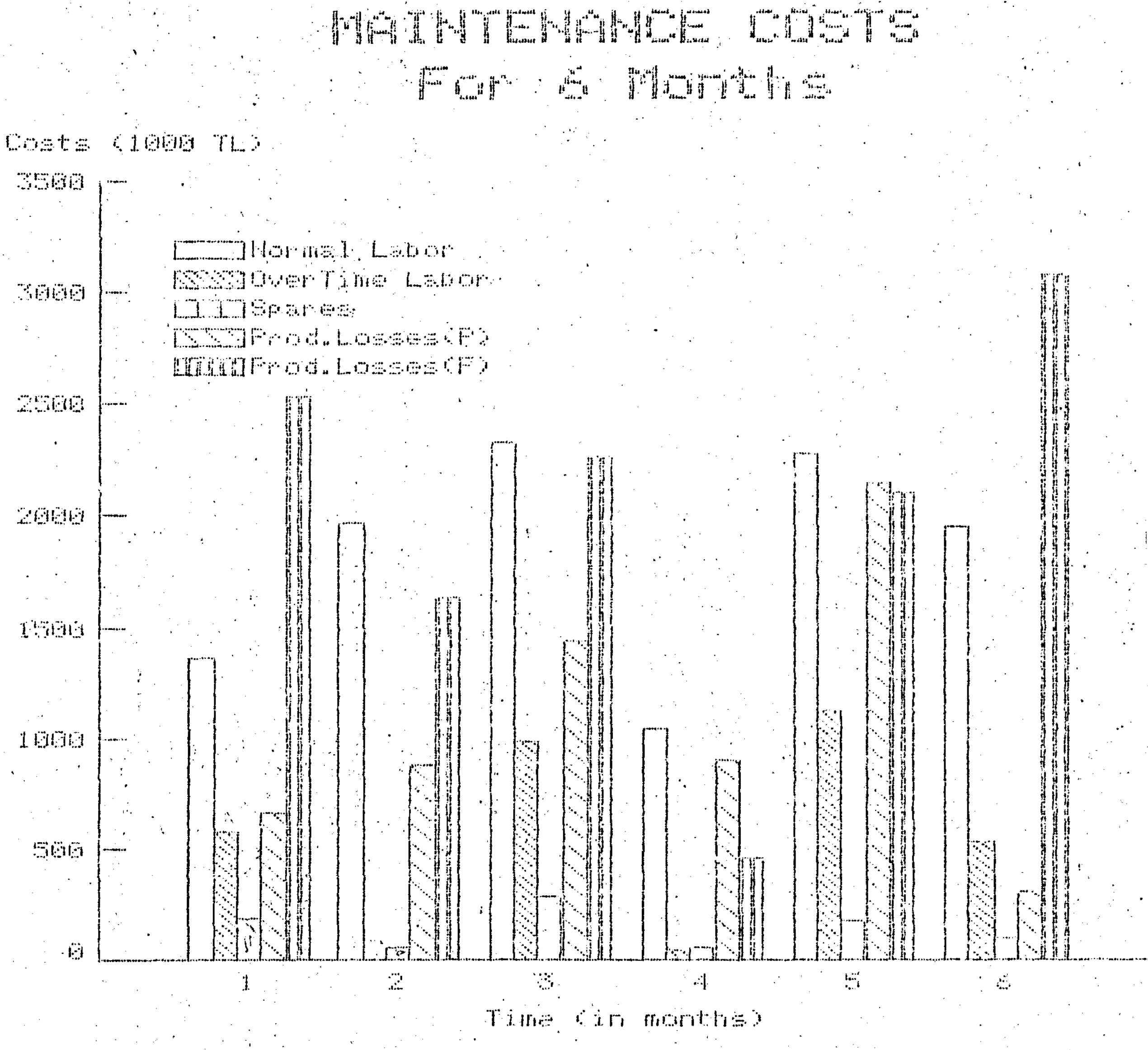


MAINTENANCE COSTS
First 4 Weeks (1000 TL)



MAINTENANCE COSTS
For 6 Months




MAINTENANCE COSTS
For 6 Months
Costs (1000 TL)
3500
3000
2500
2000
1500
1000
500
0
Normal Labor
OverTime Labor
Spares
Prod.Losses(P)
Prod.Losses(F)
1
2
3
4
5
6
Time (in months)

REFERENCES

1. Aktan, Ü., A decision support system for quality Control, Thesis, Department of Industrial Engineering, Boğaziçi University, Istanbul (1984).

2. Anthony, R.N.,Planning and Control Systems: A Framework for Analysis, Harward University Graduate School of Business Administration, Studies in Management Control, Cambridge, Mass. (1965).

3. Bakım, Seminer notları, Bekoteknik.

4. Barlow, R.E. and Porschen, Mathematical Theory of Reliability, John Wiley and Sons, Inc. N.Y. (1965).

5. Douglas, C. Montgomery, Lynwood A. Johnson, Operations Research in Production Planning, Scheduling and Inventory Control, John Wiley and Sons, Inc. N.Y. (1972).

6. Gorry, G.A. and M.S. Scott Morton,"A framework for Management Information Systems", *Sloan Management Review,* Vol. 13, Fall 1971, pp. 55-70.

7. Grogono, Peter, Programming in Pascal, Addison-Wesley Publishing (1980).

8. Hackarthorh, R. and P. Keen,"Organizational Strategies for Personal Computing in Decision Support Systems", *MIS Quarterly,* Vol. 15, No. 3, September 1981.

9. Jardine, A.K.S., "Component and System Replacement Decisions, Electronics systems effectiveness and life cycle costing", *NATO ASI Series* 1983.

10. Jardine, A.K.S., Maintenance, Replacement and Reliability, John Wiley and Sons, Inc. N.Y. (1973).

11. K.C. Kapur, L.R. Lamberson, Reliability in Engineering Design, John Wiley and Sons, Inc. N.Y. (1977).

12. Karpak, B., Makina Bakımda Benzetim, *Yöneylem Araştırması Bildiriler,* 1976, p.179.

13. Kaylan, A.R., Tüpraş Seminer Notları, Istanbul (1985).

14. Kudret Tan, Ö., Application of planned replacement policy in a copper-wire drawing company, Thesis, Department of Industrial Engineering, Boğaziçi University, Istanbul (1982).

15. Maintenance Management for efficient operational reliability, Paper, Idhammar konsult AR.

16. Mennon, M.V., Estimation of Shape and Scalle Parameters of the Weibull Distribution, *Technometrics,* Vol. 5, N. 2, May 1963.

17. Özkaya, G.,"Elemanları rassal olarak bozulan sistemlerin yenilenmesi ve dinamik bir sistem yenileme modeli", *Yöneylem Araştırması Bildiriler,* 1979, p. 324.

18. Simon, H.A., The New Science of Management Decisions, Harper and Row Publishers, New York (1960).

19. Sprague, Jr. R.H., Carlson, E.D., Building Effective Decision Support Systems, Prentice-Hall (1982).

20. Steffans, J., "The survey report for problem oriented training programme on maintenance", Otosan (1984).

21. Steffens, J., Planlı Bakım ve Onarım Seminer Notları, Kısım I, II, Segem (1984).

22. Şatır, A.,"Preventive Replacement of equipment subject to weibull distributed failures", A Case Study, Department of Industrial Engineering, Middle East Technical University, Ankara (1983).