OPTIMIZATION OF EQUIPMENT SELECTION AND ITS PLANNING FOR EARTHMOVING OPERATIONS

A DISSERTATION

submitted in partial fulfilment of
the requirements for the award of the degree
of
MASTER OF ENGINEERING
in
WATER RESOURCES DEVELOPMENT

By

TUSHAR DATT SHARMA

WATER RESOURCES DEVELOPMENT TRAINING CENTRE
UNIVERSITY OF ROORKEE
ROORKEE-247 667 (INDIA)

JANUARY, 2000

CANDIDATE'S DECLARATION

I hereby certify that the work which is presented in the dissertation entitled, **"OPTIMIZATION OF EQUIPMENT SELECTION AND ITS PLANNING FOR EARTHMOVING OPERATIONS"** being submitted by me in partial fulfillment of the requirement for the award of degree of **MASTER OF ENGINEERING in WATER RESOURCES DEVELOPMENT (MECHANICAL)** at the Water Resources Development Training Centre (WRDTC), University of Roorkee, Roorkee is an authentic record of my own work carried out during the period from July 16, 1999 to January 22, 2000 under the supervision of **Prof. Gopal Chauhan**, ***Professor*** WRDTC, University of Roorkee, Roorkee.

The matter embodied in this dissertation has not been submitted by me for the award of any other degree.

Place : Roorkee

Dated: January 22, 2000

Tushar Datt Sharma

(TUSHAR DATT SHARMA)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

(Prof. GOPAL CHAUHAN)
Professor,
WRDTC,
University of Roorkee,
Roorkee-247667 (U.P.)
INDIA

ACKNOWLEDGEMENT

I wish to express my deep sense of gratitude and sincere thanks to Prof. Gopal Chauhan, Professor, WRDTC, University of Roorkee, for his sincere advice, valuable guidance and constant encouragement for the preparation of this dissertation. He spared no pains to help me at every step of the preparation of this dissertation and gave me valuable suggestions.

I am grateful to Prof. Devadutta Das, Director for providing me an opportunity to study in this widely recognized Centre and accomplish this work. I would like to express my sincere thanks to all the faculty members and staff of WRDTC for their cooperation during my entire stay at Roorkee.

Also, I am very grateful to my parent department, Central Water Commission for deputing me to WRDTC to acquire higher knowledge in the field of Water Resources.

Finally, I wish to record special thanks to my parents and family members for their constant encouragement to finish the desired goal with due satisfaction.

Tushar Datt Sharma

(TUSHAR DATT SHARMA)

SYNOPSIS

Earthmoving operations are a major part of most of the Water Resources Development projects and have long been the subject of investigations into the proper selection and estimating their output prior to actually commencing site work. Proper earthmoving equipment selection, their optimum size and number is a major factor when trying to satisfy a projects' budget and schedule requirements. In all cases, the objective is to maximize the production, minimize the cost or to maximize the profits. This necessitates a broad knowledge of many subjects.

In general, excavating and hauling equipments viz. Dozers, Scrapers, Shovels, Loaders, Haulers are chosen for earthmoving operations and for each type of equipment selected, there are unique qualities that must be considered.

The selection of excavating and haulage equipments is a complex decision problem, which requires the identification of variables involved, their inter-relationship, and the method required to solve the problem. Literature review indicates that operational research optimization techniques currently in use display limitations. These techniques lack flexibility and often are invalidated by their inability to cope with a large number of variables, constraints and uncertainty, which are a natural part of the process of excavation and haulage. Simulation models can also be used as a non-deterministic method of calculating earthmoving productivity. But, such models may be difficult to use in practice because successful simulations depend on various factors such as accuracy of the input data, knowledge required to simulate, time requirements etc.

In view of the above, as an attempt, the present dissertation provides optimization models for motorised scrapers, towed scrapers, bulldozers and haulers and outlines analytical development of these models taking into account the major factors that influence the output from the earthmoving operations.

These models basically aim at determining the optimum size and number of these equipments for a given target of earthwork by taking into account the production of each type of equipment considered and the cost involved. Finally, all these models derive a Cost Index Number to select the proper equipment, its size and number/numbers. The models have been used to analyse the effect of various operating parameters on the performance of earthmoving operations particularly to the haulage of earth.

The selection of proper type of equipment for moving the earth against the different haul distance has been analyzed in terms of productivity.

The models have been developed through the programs written in Turbo C++ computer language. Various situations involving different operating parameters related to haulage and dozing can be analyzed by these models to optimize the eathmoving/hauling operations in real/site situations.

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS USED IN PROGRAMS

(Unless otherwise mentioned)

Symbols	Description	Units
HP	Horsepower	hp
e	Machine transmission efficiency	-
G	Gross vehicle weight of wheel mounted vehicle	Tonne
L	Payload	Tonne
E	Empty weight of vehicle	Tonne
Hcg	Heaped capacity of vehicle (given)	m^3
Hc	Adopted heaped capacity	m^3
Hcc	Calculated heaped capacity	m^3
Gc	Gross vehicle weight of crawler mounted vehicle	Tonne
Gpf	Load being pulled/towed by vehicle	Tonne
Tel	Tractive effort (while loaded)	Kg
Tee	Tractive effort (while empty)	Kg
Wdl	Weight on driving wheels (while loaded)	Kg
Wde	Weight on driving wheels (while empty)	Kg
fdl	Factor for weight on driving wheels (while loaded)	-
fde	Factor for weight on driving wheels (while empty)	-
ct	Coefficient of traction	-
k	Density of material being loaded into vehicle	Kg/m^3
v[0]	Minimum speed of vehicle	Km/hr
v[1]	Maximum speed of vehicle	Km/hr
vmax	Maximum speed specified for project area	Km/hr
RP	Rimpull	Kg
RPm	Maximum available rimpull	Kg
RR	Rolling resistance (onward journey, while loaded)	Kg
RRb	Rolling resistance (return journey, while empty)	Kg

GR	Grade resistance (onward journey, while loaded)	Kg
GRb	Grade resistance (return journey, while empty)	Kg
d_i	Length of i^{th} haul section	Km
v_i	Velocity of i^{th} haul section	Km/hr
eff.	Efficiency factor	-
Lf	Load factor	-
t_i	Travel time for i^{th} haul section	Minute
Lt	Loading time	Minute
Lt[i]	Loading time values for scraper (load growth curve)	Minute
LPT[i]	Load values for scraper (load growth curve)	m^3
sff	Speed factor (onward journey)	-
sfb	Speed factor (return journey)	-
D	Dump, delay and spotting time	Minute
CTC	Total cycle time	Minute
TPS	Trips per hour	-
P	Productivity	m^3/hr
DB	Drawbar pull available	Kg
NDB	Net drawbar pull available	Kg
Bc	Blade capacity of bulldozer	m^3
b	Bucket capacity of loader	m^3
tl	Time to load one bucket of loader	minute
CIN	Cost index number	Rs/m^3
TR	Total resistance	Kg
DF	Derating factor	-
h	Height of project from sea level	m
hrs	Number of scheduled production hours in a year	hrs
q_y	Peak production	m^3/hr
q_1	Required production	m^3/hr
Odc	Hourly owning & operating cost of dumper	Rs/hr
Ohl	Hourly owning & operating cost of loader	Rs/hr

M	Total quantity of earthwork required	m^3
IC	Overhead charges	Rs./hr
CTs	Sustained cycle time	Minute
Ps	Sustained productivity	m^3/hr
N(or Nu)	Number of haul units per loader	-
l	Arrival rate of haul units/hr	-
m	Service rate of loader/hr	-
LP	Loader productivity	m^3/hr
HP	Hauler productivity	m^3/hr
Ob	Hourly owning and operating cost of bulldozer	Rs/hr
Osr	Hourly owning and operating of scraper	Rs/hr
Opu	Hourly owning and operating of pusher	Rs/hr
Lo	Exact optimum load in scraper	m^3
to	Exact optimum loading time	Minute
Sr	Number of scrapers required	-
PCT	Pusher cycle time	Minute
PU	Number of pushers required	-
FCIN	Fleet cost index number	Rs/ m^3
Sr CIN	Cost index number of one scraper	Rs/ m^3

LIST OF APPENDICES

CONTENTS

CHAPTER – 1

INTRODUCTION

1.1 GENERAL

Earthmoving operations are a major part of most of the Water Resources Development projects and have long been the subject of investigations into estimating their output prior to actually commencing site work. Also the construction of many civil engineering projects, like highways, airports, buildings and industrial sites require the movement of earth and rock to alter the surface configuration or structural capacity of the ground. Different types of equipments have different productivity so, a construction operation can be carried out by a number of alternative equipments from different manufacturers. A problem which frequently confronts a construction manager as he plans to construct a project is the selection of the most suitable equipment. All types or sizes of equipment that might be used for the kind of work at construction site can not be afforded. A suitable combination of equipments for a given earthmoving operation is often needed in practice in order to maximize the production or to minimize the cost.

Determining the optimum size and number of equipments for an earthmoving operation is a task of critical importance because the variables affecting the equipment selection e.g. operating conditions, operation parameters, types of equipments available, required production rate, ownership costs, operating costs, etc. are numerous. Proper earthmoving equipment selection, their optimum size, and number is a major factor when trying to satisfy a project's, budget and scheduling requirements. In all cases, the objective is to maximize the production, minimize the cost or maximize the profits.

1.2 EARTHMOVING AND HEAVY EQUIPMENT

The construction of modern civil engineering and Water Resources Development projects is a complex affair, complicated in design. The construction planning and execution is a synthesis of speed, safety, efficiency and economy. Use of mechanical equipments offers some of these advantages. On major construction jobs, mechanization in indispensable while on many a small works use of mechanical equipment is really unnecessary. As with the increasing complexities and magnitude of work at most of the water resources projects, the deployment of mechanical equipment becomes worthwhile : for on increasing production and reducing costs; for replacing manual labour operations, particularly the heavy and hazardous tasks; for maintaining production where labour is scarce or too expensive to employ and for reducing material wastages. Whenever, the strategy of mechanization has been followed, in general there has been a reduction in the amount of labour employed without decline in the volume of production [Harris, 1994].

The two most important factors in deciding the extent of mechanization are : cost and time. The basic components of cost to be considered include labour costs and the costs of mechanical equipment.

In short, the decision about the extent of mechanization on a construction project should be taken after a consideration of available resources vis-à-vis the nature of construction and its magnitude. A combination of manual labour and machine work commensurate with speed and economy may be the answer in many cases.

The earthmoving and heavy equipment industry has always operated in a very competitive environment. For decades, cost effectiveness and productivity have been of paramount interest and great strides have been made in these areas. Since World War – II, the heavy equipment industry has experienced numerous cycles of good and bad times. The average period of these business cycles has been six years [Key, Jeffery, M., 1988].

Fortunately, a number of new and ongoing developments in the industry are providing equipment manufactures and contractors with improved means to deal with the intense competition and other challenges. These new developments include the availability of new alloys, high pressure hydraulic components, high strength light weight plastics, and laser assisted guidance systems for earthmoving equipment. The single biggest contribution towards improved productivity and efficiency may be the nearly universal availability of high powered micro computers and related equipments. The subject of improving cost effectiveness and productivity basically comprises of two areas : (i) improvements in equipment and methods and (ii) improvement in project and equipment management.

Without the proper working equipment, productivity decreases, delays increase, possible injuries occur and unnecessary costs are incurred. It is important for all parties involved in an earthmoving operation that the project begins with the most appropriate selection of equipment needed to perform the work. Proper selection of equipment contributes to project efficiency and to increased profits [Amirkhanian, S.N.; Baker Nancy J., 1992].

1.3 EARTHMOVING OPERATIONS AND EQUIPMENT SELECTION

Earthmoving operation include excavating, hauling, placing and compacting earth [Peurifoy, 1985]. In general, there are many types of equipments that can be choosen for earthmoving operation viz. dozers, scrapers, loaders, haulers (trucks) and compactors. For each type of equipment selected, there are unique qualities that must be considered e.g. horsepower size, productivity etc.

Equipment selection and planning includes besides it selection, the decision about working shifts, number and size of machines, matching of units working in a team, procurement schedule and the arrangement of necessary technical staff to operate, service and repair of the equipment. Planning of workshop facilities is also an important aspect of equipment planning. While the type of equipment selected usually depends upon soil and valley conditions and the characteristics of material to be handled, the number and size of machines selected depends upon the magnitude of work, working days available and number of shifts worked in a day. Though multi-shift operation is effective in increasing daily production, production in the evening shifts is usually less and these shifts add to difficulties of supervision and maintenance of equipment [Varma Mahesh, 1997]. All equipments working in a team must be properly matched in size. The plan of procurement should be coordinated with the production schedule of equipment.

Mathematical models or computerized analytical approach related to the operation of equipment can be used for planning and selection of construction equipments.

1.4 OPERATING PARAMETERS/VARIABLES & DEFINITIONS

1.4.1 Rolling Resistance

Rolling resistance is a resistance which is encountered by a vehicle in moving over a road or surface. This resistance varies considerably with the type and condition of the surface over which a vehicle moves, tire flexing and penetration of the tires into the travel surface. Soft earth offers a higher resistance as compared to hard surfaced roads. For vehicles which move on rubber tires the rolling resistance varies with the size, pressure and the tread design of the tires. For equipments which move on crawler tracks, the resistance varies primarily with the type and condition of road surface. A narrow tread high pressure tire gives lower rolling resistance then a broad-tread low pressure tire on a hard surfaced road. Conversely, for soft surface, broad tread low pressure tire will offer a lower rolling resistance than a narrow tread high pressure tire. For a stable, highly compacted and well maintained earth with optimum moisture content, the rolling resistance encountered will be very low.

Rolling resistance is expressed in kg (or pounds) of tractive pull required to move each gross tonne (ton) of vehicle load over a level surface of the specified type or condition as follows :

$$RR = \frac{P}{G}$$

Where,

RR = Rolling resistance, in kg/tonne (or lb/ton)

P = Pull required to move the vehicle along a level section of haul road at a uniform speed, in kg (or lb).

G = Gross vehicle weight, tonne (ton).

It has been found that the rolling rsistance of a rubber tired vehicle on a hard, smooth level surface amounts to approximately 40 lb/ton i.e. 20 kg/tonne of vehicle weight with bias or belted/bias tires and 30 lb/ton i.e. 15 kg/tonne with radial tires [Nunnally, 1977, Peurifoy, 1985]. For each inch of tire penetration into travel surface, rolling resistance increases approximately 30 lb/ton i.e. 15 kg/tonne. Hence, frequently, in the absence of empirical information, the following rule of thumb is used [Karafiath, L.L., 1988].

$$RR = 0.02\ G_W + C_r .\ G_w .\ S$$

Where,

RR = Rolling resistance in units of force,

G_w = Gross weight of vehicle or load on tire in the same units as RR,

S = Sinkage in inches or mm,

C_r = 0.015; when sinkage in inches or,

C_r = 6.0 when sinkage is in mm.

In case of free rolling tires, RR is the force needed to tow the tire at steady speed. Free rolling tires play a minor role (if any) in off-road hauling operations. A non driven front axle carries at most some 30% of the total loaded weight of the vehicle. Thus, the RR of driven tires predominates the RR of off-road hauling vehicles and, therefore, is of paramount importance for estimating the speed and productivity of off-road haul vehicles [Karafiath, L.L., 1988].

The RR of tires driven on level ground may be defined as follows :

$$RR = TRF - DRB$$

Where,

TRF = tractive force developed by the applied torque in the direction of drawbar pull.

DRB = net (usable) drawbar pull or rimpull .

In practice, the estimated RR is used in conjunction with rimpull speed curves supplied by the vehicle manufacturers. The rimpull is computed as

$$RP = \frac{T}{r}$$

Where,

RP = rimpull,

T = applied torque

r = rolling radius.

On road, or on hard soil surfaces, r is nearly constant for given inflation pressure and tire load and its values may be obtained from tire manufacturers tables. Off road, tire deformation and sinkage result in r varying along the contact area and its average value (weighted for the distribution of tractive stresses) is not readily obtainable. In off road situations, the average value of r is smaller than on road, resulting in a larger actual rimpull for a given applied torque then that determined for on-road applications. In estimating the attainable speed in off-road travel from rimpull-speed relations, the decrease of r and resulting increase of rimpull compensates somewhat for the under-estimation of the off-road RR by rule of thumb formules.

The off road RR of vehicles is the sum of the RR of its tires. However, it must be noted that the RR of the individual tries pertains to the tire load and torque that a particular axle transmits to the tire under the operating conditions of the vehicle. Axle loads for empty and fully loaded vehicles refer to vehicles traveling at a steady speed on level ground. Load transfer among the axles occurs when the vehicle travels on sloping ground and/or inertia forces act on the C.G. of vehicle due to acceleration or deceleration. Generally, for grades less than 12%, the effect of load transfer is minimal and may be disregarded.

In general, the RR of vehicles operating off-roads is significant factor in estimating the haul time and productivity of earthmoving vehicles. The determination of the RR of a vehicle is simple if the RR of individual tires is obtained by the analytical tire-soil interaction model. An examination of tire performance tests shows that RR of free rolling tires is frequently under-estimated and the RR of driven tires varies with the applied torque [Karafiath, L.L., 1988].

Track mounted crawler vehicles in effect travel over their own road (interior surface of their tracks), so they are often considered to have no rolling resistance. The rolling resistance of the surface on which the rated drawbar pull of crawler tractors is determined is actually 110 lb/ton i.e. 55 kg/tonne. However, when crawler tractors are towing wheeled equipment, the rolling resistance of the wheeled equipment must be considered when computing total resistance of the combinations.

1.4.2 Grade Resistance

Grade resistance is the component of vehicle weight that acts parallel to an inclined surface. When a vehicle moves up a sloping road, the total tractive effort required to keep the vehicle moving is increased in proportion to the slope of the road, conversely the tractive effort required to keep the vehicle moving is reduced in proportion to the slope of the road for a vehicle moving down a slopping road. That is, the grade resistance is a positive resistance for vehicles travelling uphill and a negative resistance (or propelling force) for vehicles travelling downhill. Grade is usually expressed in percent (1% grade has a rise or fall of 1 m in 100 m) which is the same as the tangent of the slope angle. For slopes less than 15°, i.e. for small angles the roadways making with the horizontal, sine of angle is approximately equal to the tangent of angle.

$$\text{Grade resistance} = W \times \sin\theta$$
$$\approx W \tan\theta$$

where,

W = vehicle weight

θ = angle of slope.

For most of the construction projects, the grade resistance is taken as 20 lb per gross ton of weight (i.e. 10 kg per gross tonne) for each 1% of grade [Peurifoy, 1985, Nunnally, 1977]. Hence grade resistance can be obtained from the following equation.

Grade resistance (lb/ton) = 20 x grade (%)

or Grade resistance (kg/tonne) = 10 x grade (%)

1.4.3 Effective Grade

The total resistance to the movement of a vehicle over a surface is the sum of its rolling resistance and the grade resistance.

Total resistance = Rolling resistance + Grade resistance.

The total resistance of vehicle may also be expressed as the percent grade that would have a grade resistance equal to the total resistance computed. This is referred to as effective grade, equivalent grade and is usually expressed as

$$\text{Effective grade (\%)} = \text{Grade (\%)} + \frac{\text{Rolling Resistance (kg/T)}}{10}$$

1.4.4 Coefficient of Traction & Tractive Effort

The maximum speed attainable by a moving machine naturally depends on the rolling resistance of the grade, however, the grip (traction) between the ground surface and the wheels or tracks must be sufficient to prevent slippage and therefore must exceed the supplied rimpull, if the vehicle is to move at all., i.e. if there is not sufficient traction, the full power of the engine cannot be used and the wheels or tracks will slip on the surface. Thus coefficient of traction between tires or crawler tracks and different haul surfaces is important for hauling units.

The coefficient of traction is defined as the factor by which the total load on the drive wheel or track should be multiplied before slipping occurs.

$$\text{Coefficient of traction, Ct} = \frac{\text{Maximum usuable pull}}{\text{Weight on drivers}}$$

Maximum useable pull is sometimes referred as tractive effort.

To determine the weight on drivers, total traction weight is used for crawler tractors and 4 wheel drive tractors. For other conditions, manufacturer's specification are usually referred.

1.4.5 Drawbar Pull & Rimpull

The power produced by a machine/tractor is expressed as drawbar pull for crawler tractors and rimpull for wheel mounted vehicles. Drawbar pull is the available pull which a crawler tractor can exert on a load that is being towed. It is expressed in kg or lb. In testing a tractor to determine its maximum drawbar pull at each of available speeds, the standard rolling resistance of 110 lb/ton ($\approx$55 kg/tonne) is specified by Nebraska tests. If a tractor is used on a haul road whose rolling resistance is different than 110 lb/ton (55 kg/tonne), effective draw bar is calculated as –

$$\text{Effective draw bar pull (kg)} = \text{Available DBP} - G\,(RR - 55)$$

Where,

G = Gross vehicle weight (tonne)

RR = Existing rolling resistance (kg/tonne).

The draw bar pull of a crawler tractor varies indirectly with the speed of each gear; it is highest in first gear and lowest in top gear i.e.

$$DBP \propto \frac{1}{speed}$$

Rimpull is the tractive force applied between the tires of the drive wheels and ground surface, measured in kg and is determined by the power of the engine and influenced by the efficiency of transmission, the gearing ratios and wheel arrangements. It is usually expressed as –

$$RP\ (kg) = \frac{270 \times HP \times Efficiency}{Speed\ (kmph)}$$

It is to be noted that, the usable drawbar pull and rimpull are limited by the traction which can be developed by the track or tires. And hence, tractive effort is usually referred to as maximum usable rimpull or DBP which clearly must be greater than the supplied rimpull or the wheels will spin.

Power available from a prime mover is affected by the altitude. For most practical purposes, it is assumed that for 4 cycle petrol engines and diesel engines, the losses in power due to altitude is approximately equal to 3% of sea level horsepower (or standard HP) for each 1000 ft of altitude above 3000 ft [Nunnally, 1977]. It is usually expressed by derating factor –

$$\text{Derating factor (\%)} = 3 \times \frac{(\text{Altitude in feet} - 3000)}{1000}$$

$$\approx 3\text{x} \frac{(\text{Altitude in metre} - 900)}{300}$$

Then the rated power available is

$$\text{Available power} = \frac{(100 - \text{derating factor})}{100} \times \text{standard power}$$

1.5 SCOPE OF STUDY

Out of the major earthmoving operations, excavation and haulage of earth have been primarily considered with more emphasis on haulage of earth. The optimization of selection of equipment for excavation and hauling is related to determining the optimum size and the number of equipment under given working condition required production, available time and resource for the project. The principal criteria for the selection should be maximizing the profit or to minimize the cost of production per unit of earth moved.

In the present study, comprehensive optimization models for selection of equipment have been developed as under :

(i) Optimization Model for Scraper (Motorised) production.

(ii) Optimization Model for Scraper (Towed) production.

(iii) Optimization Model for Hauler/ loader production (deterministic approach).

(iv) Optimization Model for Hauler/ loader production (probabilistic approach).

(v) Optimization Model for Bulldozer(Track mounted) production.

The models have been developed to determine the optimum size of the equipment and its number/numbers for the given conditions, with due consideration of the effect of various operating parameters on the performance of the equipment in terms of productivity.

CHAPTER – 2

LITERATURE REVIEW

Procedure of obtaining the size and number of haulers based on deterministic approach for a given target of production has been illustrated by Peurifoy (1979). The effect of size of truck on the cost of hauling the earth is discussed by assuming a fixed travel time instead of actual time. It is concluded that the use of larger trucks beyond its matching size with the excavator will not be justified unless the higher cost for the larger truck can be recovered by more economical performance during travel cycle. The effect of the size of the excavator on the cost of hauling earth for a given size of haul unit indicates that the cost of hauling earth in reduced as the size of shovel in increased.

On applying the theory of queues to determine the most economical number of haul units, Peurifoy presents a graphical solution for the optimization of excavator/ truck earthmoving system by considering it as a cyclic queuing system, and using Poisson distribution function tables.

Nunnally et al. (1977) have also discussed the deterministic and probabilistic approaches for haul unit selection. Cost analysis based on cost performance of haul only and load & haul is also described. Probabilistic approach is considered in detail. Simulation method to design the excavating & haul system is also discussed.

Regarding scraper production, Nunnally has presented a basic approach, although detailed description is not readily available. Same is the case for bulldozer production.

Harris et al. (1994) have illustrated with an example the selection of the truck loaded by given loader by utilizing the simulation technique. However, it is mentioned that models available from major equipment manufactures and operators are suited for particular situation.

Richard C. Ringwalt (1987) developed quickly applicable set of curves based on Bunching (Queuing) theory. When these curves are realistically applied, they assure the most economical match up of hauler fleet size per loader. The curves are plotted for K (i.e. hourly loaded cost/hourly hauler cost) versus percent of loader idle time for various N values ($N = 1 + \frac{\text{Travel time}}{\text{LoadingTime}}$). Number points along these curves indicate most economic 'n' value (i.e. number of haulers).

Abbas H. (1986) studied the scrapers using different Caterpillar models and presented equiproduction curves for some models and analyzed the effect of operating parameters on the productivity.

Brooks A.C. & Shaffer L.R. have also developed a set of curves based on queuing theory. The curves are plotted for quick estimation for production index.

Sabah Alkass, Frank Harris (1988) developed an expert system (ESEMPS) for earthmoving equipment selection in road construction. Knowledge for this particular domain was acquired from field practitioners such as planning engineers and equipment specialists. The results of its test case study were found encouraging, however, some refinement in output estimates and machine matching was recommended because it was felt that the data bank of adjustment controls did not fully reflect the working practice. The expert systems are only applicable to the specific domains.

Saeed Karshenes et al (1989) investigated the optimal truck capacities for various loader sizes based on specification and owning & operating costs of a number of front end loaders and trucks. The method of least squares was used to obtain the relationship between the cost and the capacity of the truck. Finally, a loader truck fleet optimization model was developed to minimize the cost subject to certain constraints.

A.K.W. Jayawardne, F.C. Harris et. al (1990), developed a linear programming model of road construction incorporating the project duration. The model attempts to optimize a comprehensive earthmoving system in road construction by comparing alternative fleets to provide an optimum material distribution and appropriate plant fleets to complete a project within the specified time.

Surji N. Amirkhanian and Nancy J. Baker (1992), developed an expert system for equipment selection for earthmoving operation. In this research project, a rule based expert system (i.e. VP-Expert) was used to develop a system for selecting the equipment. The system with 930 rules, interprets information concerning a particular project's soil conditions, operator's performance, and required earthmoving operations. The system was tested using a requirement for an actual project by three contractors. The contractors' selection compared favourably with the developed expert system with regard to the size and type of equipment needed to perform the various tasks.

Foad Farid and Thomas L. Koning (1994), developed a FLEET model which models multi-loader – truck systems using exponential distribution at steady state. FLEET is shown to be an accurate model for the steady state, multi-loader truck process if exponential distribution did accurately fit load and travel time distributions, but erlang or bea distribution more closely fits actual load and travel times in construction. Modelling load and travel times of loader-truck fleets with exponential, instead of beta distribution results in under estimation of production and over estimation of unit cost by 4 to 7%. Simulation results of a more accurate model that uses beta distribution and considers transient state, in addition to steady state, indicate that multi-loader – truck fleets may experience a 4-5% production increase and unit-cost decrease compared to FLEET results.

Hariss et. al (1994) presented a rough guide to select an appropriate system in terms of moving the excavated earth as shown in Fig. 2.1. It was based on field observations, but the proper analytical treatment has not been provided.

Douglas D. Gransberg (1996), presented an improved model for optimizing haul unit size and number based on a function of loading facility characteristics as modelled by a load growth curve. The model relies on the derivation of a cost index number to

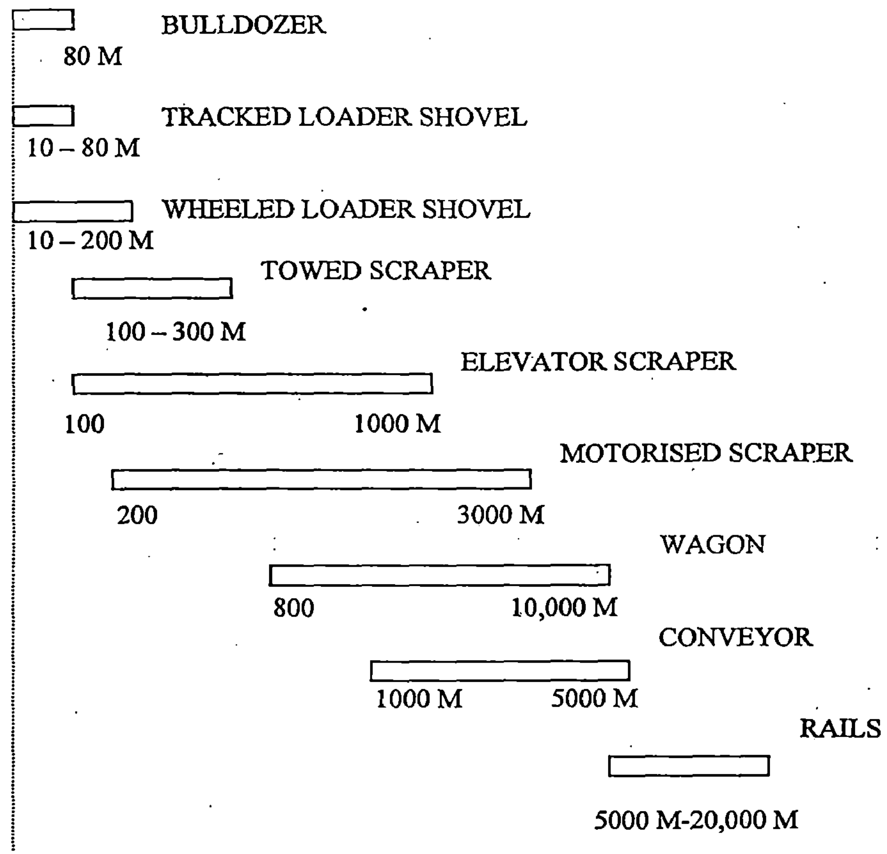


Figure 2.1 Systems for Earthmoving Transportation

determine the optimum size and number of haul units for the giving loading facilities. It provides a means to design the construction equipment fleet for a wide range of material moving projects.

Cliff Schexnayder and S.L. Weber et al. (1999) analysed the effect of truck payload weight on production. It was concluded that there was a diminished productivity increase when the load weight exceeded the truck's rated gravimetric capacity. Adding side boards actually caused the average monthly production of the fleet to decrease. The plot of load times versus load weight appears to indicate a human factors relationship between load time and providing the shovel operator load weight information with indicator lights mounted on the trucks.

A. Haidar, S. Naoum et. al (1999) investigated the feasibility of applying artificial intelligence methodologies to the optimization of excavating and haulage operations and the utilization of equipment in open-cast mining. The research was directed into the development of a decision support system X-pert Rule for the selection of open cast mine equipment (XSOME), which was designed using a hybrid knowledge-base system and genetic algorithms. The system was tested on four case studies to validate its accuracy. For each case study the equipment selected by XSOME was analyzed and compared with the actual equipment used by the contractor in the mine. A sensitivity analysis was performed on each case study to provide potential suggestions in areas where improvements could be made.

Simon D. Smith (1999) investigated the earthmoving productivity estimation using linear regression techniques. He suggested that the available deterministic models over-estimate productivity. Systems analysis methods can be used, such as queuing theory of simulation, but they are generally impractical to the average site planning engineer. To investigate the relationships between earthmoving operating conditions and both productivity and bunching, and to develop a deterministic model that will enable earthmoving operations to be planned for many different situations with relative ease, stepwise multiple regression techniques were used. The results indicated that there is a strong linear relationship between operating conditions and productivity.

CHAPTER – 3

PLANNING FOR EARTHMOVING OPERATIONS AND EQUIPMENTS

3.1 GENERAL

Earth moving operations include, besides excavation, all the allied operations such as dozing, hauling, and compacting etc. Excavation is the process through which earth is dug from one place and deposited at another. Dozing is the process of stripping top soil, clearing vegetation etc. by the equipment called Bulldozer. Hauling is the transportation of material by mobile units over highways or over project roads. In general, earthmoving equipments fall in two categories: Fixed-position and moving machines. The size of project, the topography, volume of earth of be removed and many other detailed factors influence the choice of type. The moving machine in general is used for ground levelling and bulk earthmoving whilst the static-type machine is usually operated on specific tasks. Fixed position machines include basically excavators like face shovels, backhoes, draglines and grabs. The excavator loosens the soil and loads without changing position, resulting in some loss of mobility when compared with scrapers and bucket loaders, but in compensation considerably more force may be applied at the excavation face. Moving machines include bulldozers, loaders, scapers, haulers, graders and trenching machines. The excavated materials is removed, transported and deposited in a cycle, which is a particularly useful feature when large volume of earth need to be moved over rough terrain.

3.2 EXCAVATING EQUIPMENTS

3.2.1 Bulldozer

A tractor equipped with a front-mounted blade is known as a dozer or bulldozer. Originally, the name applied to a particular dozer was derived from the type of blade used on the tractor. Thus, a tractor with a bull blade was called a bulldozer, a tractor with an angle blade was called an angle dozer, etc. However, all tractors with blades are commonly called either dozers or bulldozers. It is basically a moving excavation machine.

There are four basic blade types : straight, angle, universal, and cushion. The track-type tractor can have any of the four basic blades mounted on it, where as the wheel-type tractor is generally restricted to either the straight or cushioned blade or a lightweight U blade for the handling of light materials such as coal and wood chips.

All of these except the cushion blade may be tilted laterally to permit the concentration of tractor power on one end of the blade. Tilting is especially useful in cutting ditches and in breaking up soils that have a hard crust. Both the straight and universal blades allow the top of the blade to be moved forward or backward (pitched). Pitching the blade increases or decreases blade penetration by varying the angle of attack of the blade against the ground. Only the angle blade may be turned so that it is not perpendicular to the direction of travel of the tractor.

The straight blade is primarily a heavy-duty blade used in excavation and pioneering work. It is also suitable for drifting material and for trenching and backfilling. Since it can be tilted, it makes an excellent penetrating tool in hard excavation and can be successfully used to dislodge large boulders. By changing the pitch (tip), light materials can be drifted or hard soils may be dug, depending upon conditions.

Angle blade is generally wider and heavier than the straight blade for the same tractor, but it is lighter in cross section. As its name implies, the blade can be angled to the right or left of a normal S-blade position, up to a maximum angle of 25°. Angling blades can also be tilted in the same manner as the straight blade. however, due to the mounting on the C frame, the blade cannot be tipped. For comparable machines, the angling blade will be some 1 to 2 ft wider than the straight blade. Angle blades are most effectively employed in side casting, as in the construction of a sidehill road or in backfilling a trench.

The universal blade with its wings on each end of the blade can move large volume loads over long distances. Since it has low horsepower per foot of cutting edge and per loose cubic yard, it should not be used to obtain higher penetration or to move heavy materials.

The cushion blade is designed primarily for push-loading of scrapers. It is reinforced and cushioned to absorb the shock of contacting the push block of scrapers. it may also be used for general-purpose dozing and for cleanup work.

The bulldozer is a very versatile machine and is frequently used for :

(i) Land Clearing: Brush and small tress are removed by lowering the blade a few inches into the ground to strike and cut the roots. This is usually done in the lower speed ranges, and backing up may occasionally be necessary to clear the blade so that it can always out cleanly. Medium trees and brush 4 to 10in. in diameter usually require more than one pass. About 1 to 3 min is required for this operation, using tractors in excess of 100 drawbar horsepower. For large trees 12 to 30 inch in diameter, more care and more time are required.

(ii) Stripping : Stripping consists of the removal of topsoil that is not usable as fill material or as stable subgrade. Stripping operations should be planned so as to minimize haul distance. Stripping should be done so that the topsoil is dozed downgrade to scrapers working at the bottom of the grade. The scrapers then perform the necessary hauling. All stripping operations should be planned so that other earthmoving units can begin excavation as soon as possible.

(iii) Pioneering and Sidehill cuts: Sidehill cuts should always be started or pioneered from the top and then worked downward. Working downhill gives the advantage of gravity.

(iv) Backfilling : Dozers are the best equipment for backfilling because material may be pushed directly ahead of the machine over embankments in to ditches or directly against a structure. Angling dozers are excellent for backfilling ditches because they can drift material into the trench while maintaining forward motion.

(v) Spreading : When trucks, scrapers, or wagons are used for hauling, bulldozers are ideal spreading tools at the fill. The blade should be kept in the straight position so that the material is drifted directly under the cutting edge. Depth of spread is usually set in the job specifications to obtain the desired compaction. Tractor-dozers are also excellent tools for pulling compaction equipment on the fill.

(vi) Downhill and Slot Dozing : Whenever possible, dozing should be done downhill for greatest production. In downhill work, it is not necessary to travel down with each load. Several loads can be piled up at the brink of the hill and pushed to the bottom with one pass.

Two terms used to indicate the potential performance of a tractor and blade combination are horsepower per foot of cutting edge and horsepower per loose cubic yard (or HP per loose cubic meter). The horsepower per foot of cutting edge gives an indication of the blade's ability to penetrate a soil and obtain a full blade load of material. The horsepower per loose cubic yard is a measure of the blade's ability to move material once the blade is loaded.

3.2.2 Scrapers

The scraper is a self-loading, transporting and spreading machine predominantly used for general levelling. The scraper has been developed specifically to cater for the medium-distance haul say upto approximately 3 km. Essentially the earth is cut and loaded directly into the scraper box (or bowl), transported to the discharge area and finally spread in layers. The whole process take place in a continuous cycle. It is usual to cut on a downward gradient to take full advantage of gravity. For cutting, the bowl is lowered and the apron opened, forward movement of the machine directs the cutting edge into the soil causing it to boil upwards into the bowl. During the discharge stage ejection takes place whilst the unit is moving: again the height of the bowl is set to spread the material in a controlled layer and the soil is pushed out of the bowl with the aid of the ejector plate.

The scrapers used in modern earthmoving and heavy construction include towed wheel scrapers, wheel tractor-scrapers, elevating scrapers, and multiple scrapers. All scrapers are either powered or towed by a prime mover which can be (1) a track-type tractor (2) a two-axle wheel-type tractor, (3) a single-axle wheel-type tractor, or (4) a twin-engine unit.

3.2.2.1 Towed scraper

The towed scraper comprises a tractor, frequently this will be a bulldozer, and a towed bowl supported on two axles running on four wheels. Large volume tyres are needed to cope with the heavy loads to be transported over rough uneven surfaces. A powerful bulldozer, capable of providing 300 hp (223 kW) or more, fitted with specially strong and deep webbed tracks, is needed to provide the necessary traction during the loading operation. The loading cycle takes up to 2 min, with about 90% of the bowl filled in the first minute, but because of the slow travelling speed to the discharge area, output is considerably reduced. In practice, hauls greater than 300 m make the method uneconomic relative to other means of removing the material.

3.2.2.2 Motorised scraper

These scrapers have been developed to improve production on large earthmoving projects. Unlike the towed scraper, the engine is self-contained within the machine and power thus supplied directly to wheels. The whole unit is supported on very large volume tyres. The excavating and hauling action is carried out in a similar way to the towed scraper but frequently requires pushing assistance during the loading phase because of loss in traction when using wheels rather than tracks. Approximately 50-100m of travel is required to fill the bowl. Excavation is carried out in layers of from 150 to 300 mm in depth, the levelling action is thus achieved as a gradual process [Harris, 1994]. The haul speeds up to 60 km/h are possible with well-graded roads, yielding considerable improvements in output.

3.2.2.3 Towed scrapers versus wheel tractor-scrapers

The type of machine to be adopted depends upon the travelling distance; the tractor-pulled scraper is favoured for short hauls, while the motorised version is now almost universally preferred where the size of the project permits.

The advantageous economics of the crawler-drawn scraper (towed) as compared to the wheel tractor-scraper stem from its:

(a) Ability to minimize the adverse effects of high rolling resistance by laying down its own road.

(b) Superior tractive effort in mud ability to handle severe grades, ability to economically self-load some materials if the tractor is matched with a proper size of scraper typically lower undercarriage costs in severe rock work when compared to tire costs on a wheel tractor-scraper; and

(c) the limiting factor of the system that is speed-normally a maximum of 4 to 5 mph. Thus, this method must be classed as short-haul system-the maximum economical haul distance increases as total resistance increases.

3.2.2.4 Pusher & push-loading

Wheel tractor scrapers usually require the assistance of pusher tractors to obtain maximum production and lowest cost per cubic yard (or cubic meter). Three basic methods of operation for push tractors are brack-track loading, chain loading & shuttle loading. Back-track loading is the slowest of these loading methods because of the longer travel distance between loading positions. Chain and shuttle loading methods are almost equally efficient. The choice of loading methods will depend on the length of the cut area and whether or not scraper traffic can move through the cut area in both directions. In order to determine the most efficient loading time to obtain the maximum scraper production, optimum load time needs to be considered. The determination of optimum load time can be based on maximizing scraper production and assuming that an adequate number of pushers are available to insure that scrapers do not have to wait for pushers. When pusher resources are limited, an optimum load time based on optimizing pusher production rather than scraper production will be required.

well-maintained haul roads and adequate pusher power. These factors enable the wheel scraper to exploit its high-speed capability. Whenever high total resistance is encountered, tandem-engine scrapers will usually be economical.

Economic application zones as a function of one-way haul distance and total resistance are graphically illustrated in Figure 3.1. In general, single-engine, overhung scrapers operate best on relatively flat haul roads up to about 6,000 ft (2000 m) in length where maneuverability is important and adequate pusher power is available. Three-axle scrapers are best for long, relatively flat high-speed hauls. All-wheel drive, tandem-powered, and push-pull scrapers can usually load without pushers and can operate under conditions of high total resistance. Elevating scrapers can also self-load and can provide a finish grade, but they operate best on relatively short hauls.

3.3 HAULING EQUIPMENT

The transportation of excavated earth may be termed as hauling of earth. It is one of the biggest jobs done in earth moving. A wide range of equipment is available for transporting excavated material, including dump trucks, wagons, scrapers, conveyor belts, and trains. Although scrapers may be used solely as a transportation device, they are designed to both load and haul excavation.

The choice of system in transporting material from the loading point depends on many factors including :

(a) Site conditions ;

(b) Volume of material to be moved;

(c) Type of material;

(d) Time available;

In spite of the availability of the equipment mentioned above, the vast majority of excavation hauling is accomplished by trucks and wagons. The heavy-duty, rear-dump truck is the piece of equipment most used. The dump truck permits great flexibility in hauling operations, and the highway (or over-the-road) model can be moved rapidly

Pusher selection can be the most important single element of a properly applied scraper spread, since the entire spread's output is dependent on it. Field observations indicate that the ratio of scrapers to pusher may be taken as follows [after, Havers & Stubbs, 1971].

1. 3:1 ratio is the closest practical balance - it delivers lowest unit cost.
2. Either 2:1 or 4:1 ratios result in a significant cost penalty.
3. An improvement could be made in 4:1 ratio economy by lowering load time and carrying smaller load. This still results in some penalty.

Balance between loading and hauling equipment is essential for economic application for any interdependent load and haul system. Normal matching of push power to scraper size is as follows [Havers & Stubbs, 1971].

a. 300 hp scraper pushed by 270 hp pusher (usually -180 hp pusher in good *material* or 385 hp pusher in very hard, severe conditions).
b. 415hp scraper pushed by 385 hp pusher (usually pushed by -270 hp pusher in good material-tandem 2-385 hp pusher in very hard, severe conditions).
c. 550 hp scraper pushed by tandem 2-385 hp pushers- (usually pushed by -385 hp pusher in good material).
d. 550 hp scraper and up-tandem 2-385 hp. pushers.
e. Tandem 950 hp scraper (32/44 cu. yd struck/heaped capacity) pushed occasionally by 385 hp pusher, but usually tandem 2-385 hp. pushers.
f. Tandem scraper 950 hp and up (40/54 cu. yd struck/heaped capacity) and up- never less than tandem 2-385 hp. pushers, sometimes triple pushers.

3.2.2.5 Application zones of scrapers

Scrapers are primarily used on medium to long hauls. The operating area of medium hauls is sometimes referred to as the slow-speed zone. Long hauls correspond to the high-speed hauling zone. In this zone will normally be found good ground conditions,

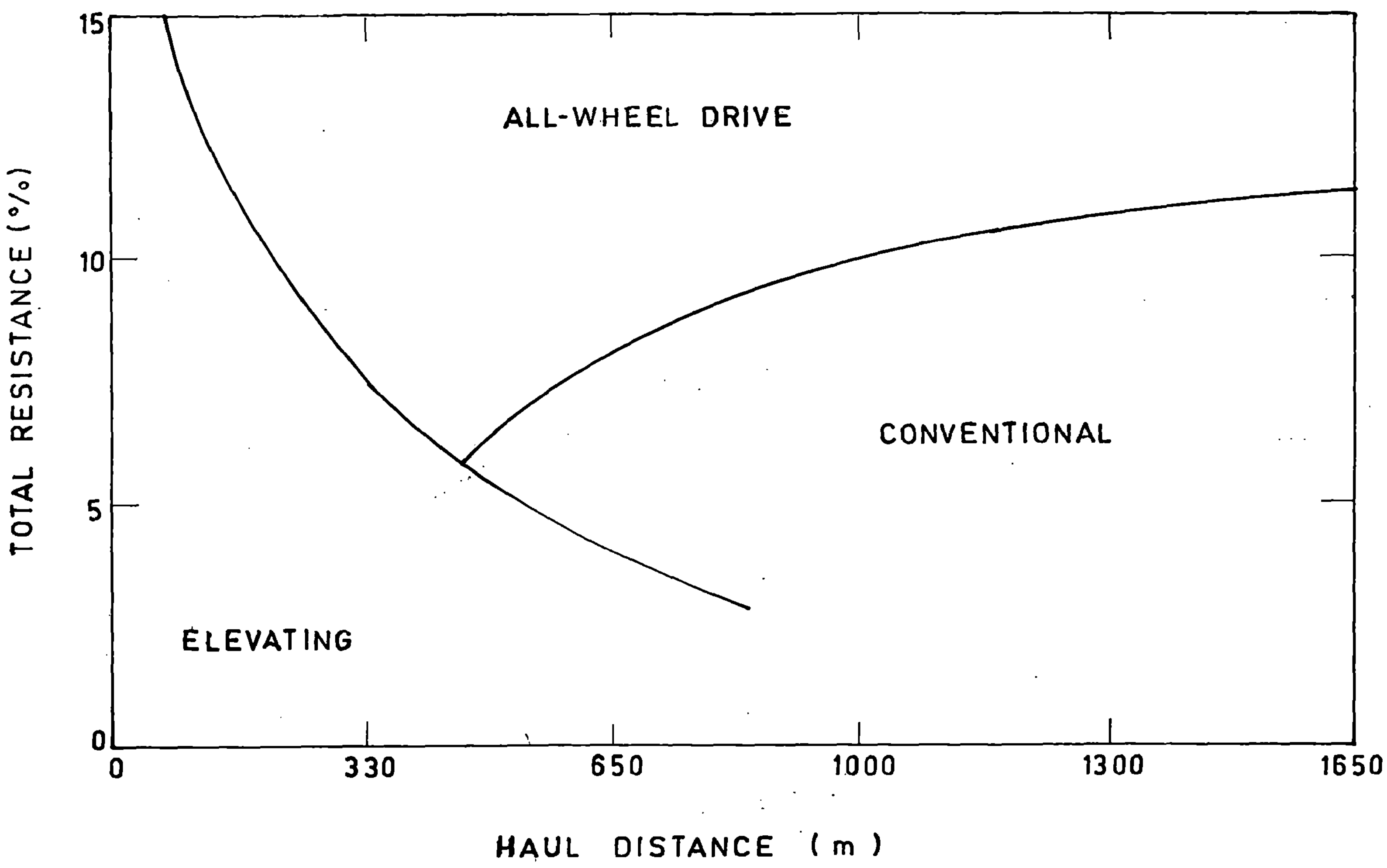


FIG. 3·1-ECONOMIC APPLICATION ZONES OF SCRPERS between job sites. Dump trucks are available in a wide variety of sizes and types. Earthmoving trucks are mainly used to transport material over distances between 1 & 10 km or even for longer journeys. For economic transporting haul roads should ideally be limited to a maximum gradient of 15% [Harris, 1994], preferably without bends and curves, have a hard surface and be continuously maintained in this condition by grading. The cost of a grader is easily recovered in the lower maintenance and repair costs of the trucks. When these savings are combined with the lower tyre wear and increased output resulting from the high speeds attainable, the grader proves a vital machine in the earthmoving operation.

3.3.1 On highway & Off-highway Hauling Units

The hauling units may be classified as "on-highway" and "off-highway". On-highway haulers are vehicles whose design and manufacture are intended to permit their use on the public road, and as such they are subject to motor vehicle registration. Off-highway haulers are vehicles whose design and manufacture are intended to preclude their use on public roads, thus they do not comply with on-highway limitations imposed by the various states, and are not subject to motor vehicle registration. Typical characteristics are widths in excess of 8 ft 6 in. and/or axle loads at rated payload in excess of 24,000 lb (or 12000 kg). Apart from the legal distinctions, there are specific design features which characterize the off vis-a-vis on high way hauling units [Havers & Stubbs, 1971]:

1. Body construction is specifically designed to absorb high-impact loading. The floor of the body may be constructed of high-tensile steel (100,000 psi) or aluminum and will generally be fitted with steel wear plates to reduce abrasion damage.
2. The power train will consist of a diesel engine-often turbo-charged coupled to a torque converter and power-shift or automatic transmission. The transmission will be of the planetary or countershaft design and will often be mounted remotely from the engine. A means is provided for the final gear reduction at the drive wheels. This is usually a planetary-type axle or an internal driving gear and ring gear in the wheel hub or final drive case.

3. The ratio of gross vehicle weight to horsepower (fly wheel or net) is generally lower than for an on-highway hauling unit. Top speed is slower than for a comparable on-highway hauling unit (about 60 to 70 kmph), but gradeability under adverse conditions of grade and rolling resistance is from 25 to 40 percent better.

4. In most cases, a rigid-type frame is used to hold all driving components in alignment. Box sections, I beams, or ship channel frame rails-generally separated by cylindrical torque tube-are used. On large hauling units, the frame may be built in two sections and joined by means of two or more vertical pins so as to articulate for better maneuverability.

5. Wheel and drive configuration may be 4 by 2, 4 by 4,6 by 2, or 4. Tires designed for earthmoving, mining, and logging service are standard equipment.

In general, the on-highway hauler is found on long, high-speed hauls over roads with low rolling resistance. In other conditions the off-highway hauler has to be used.

Major types of off-highway units include :

(i) The rear-dump truck carrying a body on the truck chassis.

(ii) Bottom-dump tractor-trailer unit.

(iii) A side-dump truck carrying the body on the truck chassis. This unit may be dumped either by its own integral hoist mechanism or by an independent hoist or "sky hook". Independent hoists are used in places where a fixed dump point is required, such as a hopper or crusher, and one dumping mechanism can thus handle many trucks.

(iv) Side-dump tractor-trailer unit.

(v) Rear-dump tractor-trailer unit.

3.3.2 Selection of a Hauling Unit

To select the right size and type of hauling unit, the estimator or equipment engineer must know thoroughly the nature of the job which has to be done.

In selecting a hauling unit, the general criteria used are :

1. The versatility of the hauling unit; can it be efficiently used in other applications?
2. The compatibility between the unit being considered and currently used hauling or loading equipment.
3. Possible restrictions on maneuvering space at the loading or dumping areas or on side and overhead clearances.
4. Possible vehicle weight or width restrictions on haul roads, including bridges.
5. The effect of extreme grades, either favourable or unfavourable, particularly on the loaded haul.

3.4 PLANNING FOR CONSTRUCTION EQUIPMENT

Varying with the degree of mechanization on a project, the cost of the equipment may range from 10 to 30% of the total cost of a construction project. [Varma, Mahesh 1997]. Besides being a sizeable portion of the capital investment, the proper selection and application of this equipment is essential to achieve the construction targets, and to keep construction costs low. Equipment planning includes besides the equipment selection, decision about working shifts, number and size of machines, the matching of units working in a team, procurement schedule and the arrangement of necessary technical staff to operate, service and repair of the equipment, workshop and stores facilities planning.

The Plant and Machinery Committee, 1974 has recommended that: two shift operation of equipment is economical and should be employed whenever possible. Three shift working results in frequent break-down and low availability, thus necessitating more standbys and increasing the cost. Single shift should be used where works are located in difficult terrain. Schedule working hours with 200 working days available in a year should be as follows under average working conditions:

Single shift operation = 1200

Two shift operation = 2000

Three shift operation = 2500

If more or less than 200 working days are available, working hours should be changed proportionately. For old machines, scheduled working hours should be taken as 80% of the above figures. No increase in scheduled working hours should be made on the basis of increased standby.

The norm for providing standby equipment should be as below:

Single shift operation = 10%

Two shift operation = 20%

Three shift operation = 30%.

Since the plant planning is done on peak requirements, it is worth-while that construction scheduling is done in such a manner that peak requirement is not substantially higher than that of average production. The new equipment acquired on a project would be utilized at least to the extent of 75% of its life.

All equipment must work in coordination, specially when a team work is involved in operation. The capacities of the equipments should be properly balanced so that the product of one machine is received by the other without loss of time. Suitable arrangement must be ensured for adequate servicing and maintenance of equipment. Arrangements for maintenance, servicing and repairs should include suitable repair shop or maintenance depots with necessary servicing equipment and well trained and conscientious workmen. Proper records of operation of equipment should be maintained for production, idle and breakdown hours, consumption of stores and spare parts, repairs etc.

CHAPTER – 4

OPTIMIZATION MODELS AND ALGORITHMS

4.1 GENERAL

Excavating and haulage equipment selection is a complex decision problem, which requires the identification of the variables involved, their inter-relationship, and the method required to solve the problem. Literature review indicates that operational research optimization techniques currently in use display limitations. These techniques lack flexibility and often are invalidated by their inability to cope with a large number of variables, constraints and uncertainty, which are natural part of the process of excavation and haulage [A. Haidar, S. Naoum, et.al. 1999].

Numerous techniques established to estimate the output of the earthmoving equipments provide the solution with varying degree of success. Since 1968 many researchers have provided simulation models that can be used as a non-deterministic method of calculating earthmoving productivity, thus allowing for truck-bunching , Queuing theory has also been used for similar purposes. Unfortunately, such models may be difficult to use in practice for the following possible reasons [Simon D. Smith, 1999] :

- The knowledge required to simulate with any degree of accuracy and confidence may not generally be available to the site manager who would have to rely on the exactness of the information supplied to him by the planning engineer.
- Successful simulations depend on the accuracy of the input data; for earthmoving operations, such data need to be obtained by very in-depth site and geotechnical investigations.
- Shi and AbouRizk (1997) summarized their view of the problem as "the major obstacles to (simulation) use by industry are the complexities involved in constructing a model and the resultant time requirements".

In view of the above, present study provides optimization models for scrapers, haulers and bulldozers and outlines analytical development of these models taking into account the major factors that influence the output from earthmoving operations apart from the necessary inputs of field observations wherever required. The model for motorized scrapers is developed for three-sections of haulage, for hauler, it is developed for five sections, while for the towed scrapers and bulldozers, it is developed for two-sections.

4.2 OPTIMIZATION MODEL FOR SCRAPER (Motorized Scraper)

4.2 1 Preliminary Remarks

The optimization model for scraper aims at determining the optimum size and number of scrapers for a given target of earthwork production by considering – the production of a scraper and cost involved. Scrapers powered by wheeled tractors usually require the assistance of pusher tractors to obtain maximum production and lowest cost per cubic meter of earth hauled, so the pushers become an integral part of the optimization model for scraper. The model developed relies on the derivation of a Cost Index Number (CIN) to determine the optimum size and number for the given task by adopting the scraper/scrapers giving the minimum CIN from among the various alternatives.

4.2.2 Model Formulation

The production of a scraper is given by

$$P = H_c \times TPS \times L_f \times C_f$$

Where,

H_c is the heaped capacity of the scraper in (Lm^3)

TPS is the number of trips performed per hour, and

L_f is the load factor

C_f is the correction factor.

i.e. the optimization is basically related to optimizing H_c and TPS. H_c in turn, depends on filling the bowl for a given optimum load time. TPS is related to the cycle time which basically comprises fixed cycle time and variable cycle time. The fixed cycle time is expressed as -

Fixed cycle time = loading time + dumping time + spreading time.

Thus, fixed cycle time can be optimized by finding the optimum load time. To optimize the loading time, the load growth curve of the given scraper is used. The optimum load time is the most efficient loading time for a particular scraper job at which scraper gives maximum production and beyond this time the scraper production per unit of time actually decreases. A typical load growth curve is shown in Fig. 4.1.

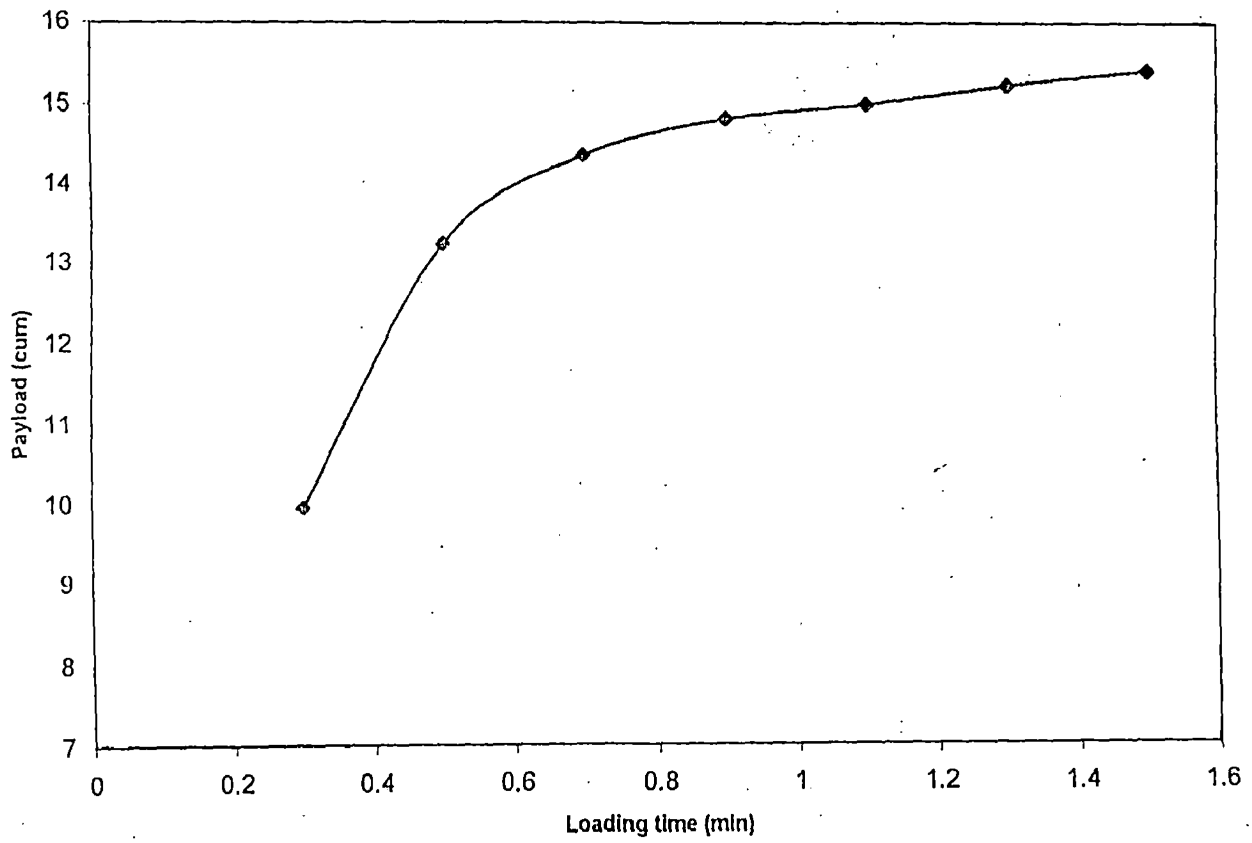


Fig.4.1 :Load growth curve of Scraper

The variable cycle time can be optimized by using maximum possible speeds for haul and return. Maximum speed attainable under the load and operating conditions is determined by the power available at the wheel, the limitations of traction and resistance encountered.

Tractive effort and total resistance are expressed as -

Tractive effort = coefficient of traction x weight on driving wheels,

Total resistance = [(rolling resistance, kg/tonne)+(grade resistance kg/tonne)] x G

where, G = gross vehicle weight in tonnes.

Representative rolling resistances for various types of wheels and surfaces are given in Table 4.1 and coefficient of traction for various road surfaces is given in Table 4.2 [Peurifoy, 1979].

Table 4.1 : Representative Rolling Resistances (in kilograms per metric ton of gross load)

Type of surface	Steel tires, Plain bearings	Crawler type track and wheel	Rubber tires, anti-friction bearings	
			High pressure	Low pressure
Smooth concrete	20	27	18	23
God asphalt	25-35	30-35	20-33	25-30
Earth, compacted and maintained	30-50	30-40	20-35	25-35
Earth, poorly maintained	50-75	42-55	52-70	35-50
Earth, rutted, muddy,	100-125	72-90	90-110	75-100
Loose, sand and gravel	140-160	80-100	130-145	110-130
Earth, very muddy rutted soft	175-200	100-120	150-200	140-170

Table 4.2 : Coefficient of Traction for Various Road Surfaces

Surface	Rubber tires	Crawler tracks
Dry, rough concrete	0.8 – 1.00	0.45
Dry, clay loam	0.50-0.70	0.90
Wet clay loam	0.40-0.50	0.70
Wet sand and gravel	0.30-0.40	0.35
Loose dry sand	0.20-0.30	0.30
Dry snow	0.20	0.15-0.35
Ice	0.10	0.10-0.25

Based on the total resistance encountered and the rimpull available maximum speed can be obtained for total resistance (positive and negative) by using rimpull versus speed table supplied by manufacturers.

The travel time obtained is further adjusted by applying a speed factor to the maximum vehicle speed. A table of the speed factors is given in Table 4.3.

Table 4.3 : Speed Factors for Motor Scraper (after Komatsu hand book)

Distance of haul load section m	Where a vehicle starts from stop	Where a vehicle enters a section in the course of travelling
0 – 150	0.30-0.45	0.55-0.60
150-300	0.45-0.60	0.60-0.70
300-500	0.50-0.65	0.65-0.75
500-700	0.60-0.70	0.75-0.85
700-1000	0.65-0.75	0.80-0.90
1000-	0.70-0.85	0.85-0.95

Optimum load time based on maximizing scraper production is determined by assuming that an adequate number of pushers are available to ensure that scraper do not wait for pushers. When pusher resources are limited, optimum load time based on optimizing pusher production is required.

The average loading time for self loading scrapers ranges from about 0.8 to 1.5 minute, whereas; for push loaded scrapers it ranges from about 0.6 to 1.0 minute (Nunnally, 1977). Table 4.4 gives typical loading times for various equipment and job conditions.

Table 4.4 : Typical Loading Times (minutes)

Job Conditions	Scraper Type						
	Single-engine, Overhung (push-loaded)	3-axle (push-loaded)	All-wheel Drive		Tandem		Elevating
			Push-loaded	Self-loaded	Push-loaded	Self-loaded	
Favourable	0.6	0.8	0.5	0.8	0.8	1.0	0.8
Average	0.8	1.0	.07	1.3	1.0	1.5	1.0
Unfavourale	1.2	1.5	1.1	1.8	1.3	2.0	1.5

Maneuver and dump time is the time required for the scraper to maneuver and dump its load. Average values have been found to range from 0.6 to 1.0 minute.

Table 4.5 gives the scraper maneuver and dump times (minutes).

Table 4.5 : Scraper maneuver and Dump Times (minutes)

	Scraper Type				
Job conditions	Single-engine, overhung	3-axle	All-wheel drive	Tandem	Elevating
Favourable	0.4	0.4	0.4	0.6	0.4
Average	0.6	0.7	0.6	1.0	0.5
Unfavourable	1.3	1.5	1.0	1.6	0.6

Spot and delay time represents the time required for positioning the scraper at the cut and waiting for pushers to begin loading if pusher is required. Table 4.6 gives scraper spot and delay times (minutes).

Table 4.6 : Scraper Spot and Delay Times (Minutes)

Job conditions	Scraper Type					
	Single-engine overhung (push)	3-axle (push)	All-wheel Drive		Tandem (push or self-load)	Elevating (self-load)
			Push-loaded	Self-load		
Favourable	0.4	0.4	0.1	0.1	Negligible	Negligible
Average	0.6	0.6	0.2	0.2	0.1	0.1
Unfavourable	0.8	0.9	0.5	0.3	0.2	0.2

4.2.3 Algorithm

(i) Find adopted heaped capacity Hc using pay load considerations and find corresponding loading time, Lt. as under

$$Hcc = \frac{L \times 1000}{k}$$

If,

$$Hcc \geq Hcg$$

Then take

$$Hc = Hcg$$

Otherwise take

$$Hc = Hcc, \text{ and}$$

$$Lt = \frac{Ltg \times Hc}{Hcg},$$

where,

L	=	Pay load on the machine in tonne
Hcg	=	Given heaped capacity of the machine (m^3)
k	=	Density of the soil in kg/m^3
Ltg	=	Loading time (given)
Hcc	=	Calculated heaped capacity (m^3)
Hc	=	Adopted heaped capacity (m^3)

(ii) Compute tractive effort while loaded

$$TEl = ct \times Wdl \times 1000 \quad ,kg$$

and $Wdl = fdl \times G$

where,

Wdl = load on the drive wheels;

fdl = factor for load on drive wheels

ct = coefficient of traction

G = gross vehicle weight (tonne).

(iii) Estimate the maximum rimpull available RPa in kg by equation

$$RPa = \frac{270xHPxe}{v[0]}$$

where,

HP = fly wheel horse power of the machine

e = efficiency of the machine

v[0] = minimum speed in km./hr

(iv) Adjust the maximum rimpull by derating factor DF,

$$\text{Maximum usable rimpull ,RPm} = \frac{RPa \times (100 - DF)}{100}$$

where,

$$DF = 3 \times \left[\frac{\text{Altitude in m} - 900}{300}\right]$$

(v) Check traction for loaded condition, if, TE1 > = RPm,

if yes, find ,

(v-a) Rimpull required, RP_i (in kg)

$$RP_i = (RR_i + GR_i) \times G$$

where,

RR_i = rolling resistance in kg/tonne for i^{th} section

GR_i = grade resistance in kg/tonne for i^{th} section

G = gross vehicle weight in tonnes

= E + L, in tonne

E is the empty weight of machine in tonne and

L is the pay load in tonnes.

(v-b) Check if $RPm >= RP_i$ and $RP_i > 0$

If, yes, then find speed v,

$$v = \frac{270 \times HP \times e}{RP_i}$$

(v-c) Find maximum speed attainable v_m by comparing v to v[0], v[1] supplied by the manufacturer, where, v[0] is the minimum speed and v[1] is the maximum speed of the vehicle.

(v-d) Compare v_m and v_{max}.

where, v_{max} is the maximum speed based on legal speed limit or other safety restrictions.

(v-e) Obtain average velocity for the given section by multiplying speed factor to the obtained speed v_m.

$$v_{average} = v_m \times \text{speed factor}.$$

It is the $v_{average}$ which is used in calculations.

(v-f) If, $RP_i <= 0$, then take

$v_m = v[1]$ and compare v_m with v_{max}.

Finally find

$$v_{average} = v_m \times \text{speed factor}.$$

(v-g) If, $RPm < RP_i$

Then write "scraper can not operate in this condition".

(vi) If, $TEl < RPm$,

i.e. rimpull is limited by tractive effort TEl.

Then take,

Maximum usuable rimpull , $RPm = TEl$;

Again repeat steps from (v-a to v-g).

Finally find $v_{average}$.

(vii) Time for the i^{th} section (except section 1) is given by

$$t_i = \frac{d_i}{v_{average}} \times 60 \quad \text{min.}$$

where,

d_i is the distance for the given section i.

and time for section – 1, t_1 is given by

$$t_1 = Lt + \left(\frac{d_1 - 0.1}{v_{average}}\right) x60 \qquad \text{(If load growth curve is not given)}$$

$$= Lt\,[n]\,(+\left(\frac{d_1 - 0.1}{v_{average}}\right) x60 \qquad \text{(If load growth curve is given)}$$

(viii) Compute total time of travel for hauling T_h.

$$T_h = \sum_{i=1}^{n} t_i$$

(ix) For return journey, the gross vehicle weight 'G' = E i.e. empty weight of machine. So that traction for empty condition TEE is

TEE = ct x Wde x 1000 , kg

and,

Wde = fde x G

where,

Wde = weight on driving wheels in tonne while empty

fde = factor for weight on driving wheels while empty.

and repeat steps from (iii) to (viii).

Finally, find time for return journey T_r -

$$T_r = \sum_{i=1}^{n} t_i \quad , \text{ where Lt or Lt[n]} = 0 \text{ for } t_1$$

(ix) Now, compute total time of journey T

$$T = T_h + T_r$$

(x) Adopt dump and delay times from the Table 4.5 and Table 4.6

(xi) Estimate cycle time TC less load time,

TC = T + D, where D is the dumping & delay times.

(xii) Optimize load time Lt by using load growth curve as under.

(xii-a) Compute trips per hr by using following equation, which uses 'j' as suffix for loading times adopted -

$$(TPS)_j = \frac{50}{(CTC)_j}$$

where,

$$(CTC)_j = (Lt)_j + (TC)_j$$

and $(TC)_j = (T)_j + (D)_j - (Lt)_j$

(xii-b) Calculate production P_j as

$(P_j) = (TPS)_j \times (LP)_j$

where,

$(LP)_j$ = load into the scraper for a given time span $\Delta(Lt)_j$

(xii-c) Compute production P_j for different load times, Lt_j's.

(xii-d) Obtain maximum production P_{max} out of all P_j's. Denote P_{max} as $(P_m)_j$

(xii-e) Corresponding to P_{max} find time $(Lt)_j$ and load $(LP)_j$. Denote $(Lt)_j$ as $(t_m)_j$ and $(LP)_j$ as $(LP_m)_j$.

(xii-f) Using numerical technique (forward difference method) obtain parabolic equation

$$P = 1 + p \times \Delta P_o + \frac{p(p-1)}{2!} \times {}^2P_o \qquad (i)$$

where,

$$p = \frac{[(tm)_j - (tm)_{j-1}]}{\Delta(Lt)_j},$$

$\Delta P_o = (Pm)_j - (Pm)_{j-1}$,

$$\Delta^2 P_o = (Pm)_{j-1} + (Pm)_{j+1} - 2 \times (Pm)_j$$

Differentiating equation (i) with respect to p, and equating it to zero, i.e.

$$\frac{dP}{dp} = 0$$

we get $t_o = (t_m)_{j-1} + \frac{1}{2}d - \frac{d_1 d}{d_2}$

and

$$LP_{max} = (LP_m)_{j-1} + p \times d_{11} + \frac{p(p-1)}{2} d_{22}$$

where,

$(t_m)_j$ = approximate optimum loading time,

t_o = exact optimum loading time

d = $\Delta(Lt)_i$

d_1 = ΔP_o

d_2 = $\Delta^2 P_o$

d_{11} = $(LP_m)_j - (LP_m)_{j-1}$

d_{22} = $(LP_m)_{j-1} + (LP_m)_{j+1} - 2 \times (LP_m)_j$

$(LP_m)_j$ = approximate optimum load corresponding to j^{th} value of loading time Lt

LP_{max} = exact optimum load

(xiii) Compute total cycle time

$CTC = TC + t_o$,

(xiv) Compute trips/hr

$$TPS = \frac{50}{CTC}$$

Using 50 minutes hour.

(xv) Compute productivity of one scraper

$(q) = (Hc) \times (TPS) \times Lf \times Cf$ (if load growth curve is not given)

$= LP_{max.} \times (TPS) \times Lf \times Cf$ (if load growth curve is given) where,

Lf is the load factor and

Cf is the correction factor.

(xvi) Estimate total number of scrapers required as

$$Sr = \frac{Q}{q}$$

where,

Q = total earthwork required in m^3/hr

(xvi) Based on type of loading (back track / chain loading), compute Pusher cycle time, PCT.

(xvii) Compute number of Pushers

$$Pu = \frac{CTC}{PCT}$$

(xviii) Compute cost index number of one scraper

$$SrCIN = \frac{Osr}{q}$$

where,

Osr = hourly owning and operating cost of the scraper.

(xix) Compute cost index number of the fleet

$$FCIN = \frac{(Pu \times Opu + Sr \times Osr)}{q \times Sr}$$

where,

Opu = hourly owning and operating cost of the pusher and

Sr = number of scrapers,

Pu = number of pushers.

(xx) Repeat steps from (i) to (xix) and find minimum cost index number FCIN amongst various alternatives

Select the model / size and number / numbers of the scraper giving minimum FCIN.

The necessary flow chart enclosed presents the steps of program of optimization model of motorised scraper.

The program (with load growth curve) has been given in Appendix – A, and the program (without load growth curve) has been given in Appendix - B.

4.3 OPTIMIZATION MODEL FOR TOWED SCRAPER

4.3.1 Preliminary Remarks

The optimization model for towed scraper basically stresses on fixing tractor capacity and load volume handled per unit of time to select the optimum size of the scraper and its number/numbers against a given target. Finally a cost index number may be derived to arrive at a convincing conclusion for selection of the machine.

4.3.2 Model Formulation

In this case also the production of the scraper is given by the equation similar to that for motorised scraper i.e.

$$\text{Production } (m^3/hr) = Hc \times TPS \times Lf \times Cf$$

where,

Hc = heaped capacity of the scraper

TPS = number of trips performed per hour and

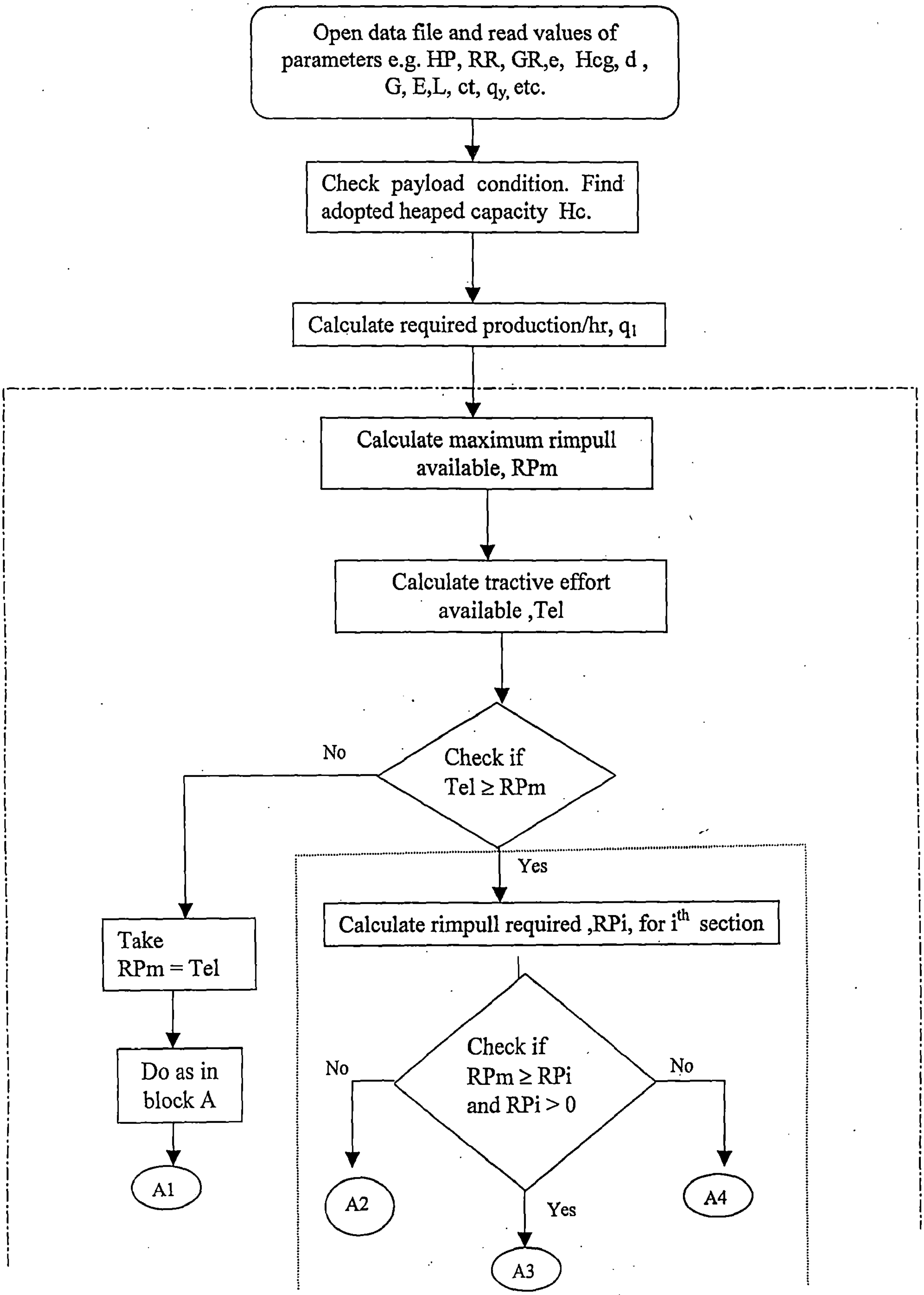
Lf = load factor

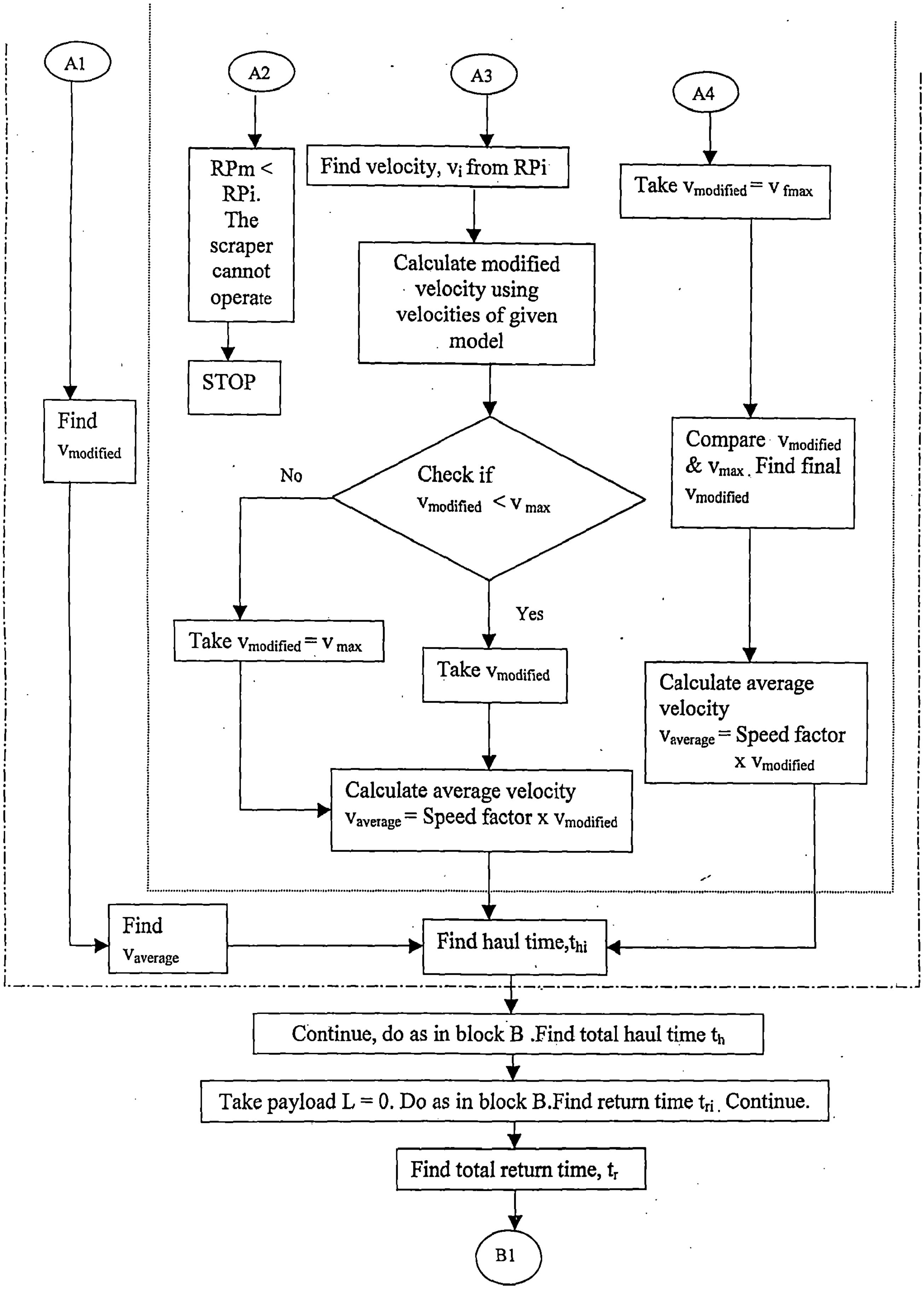
Cf = correction factor.

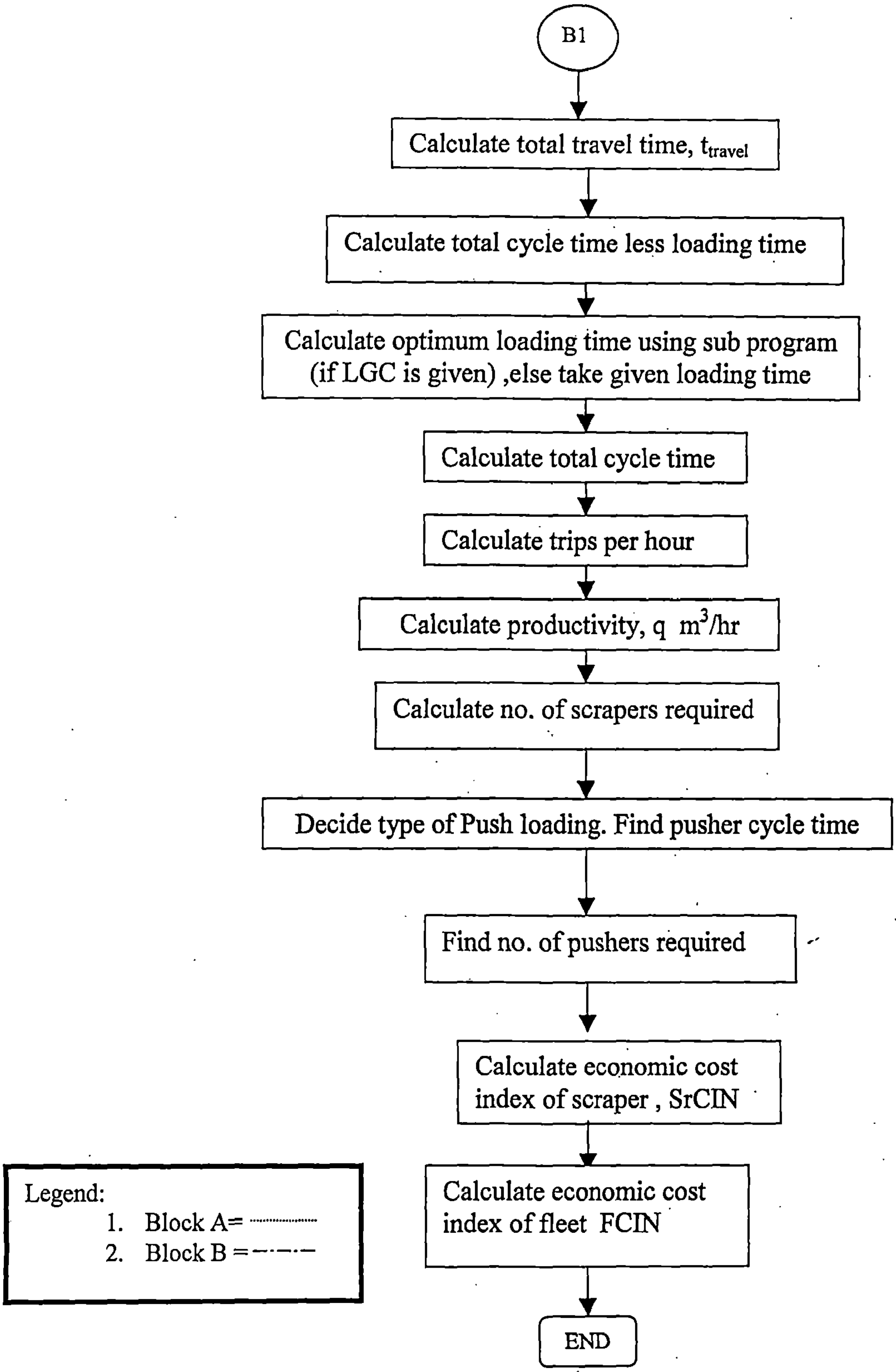
As in the case of motorized scraper, optimum production can be achieved by optimizing Hc and TPS.

Optimizing Hc and TPS basically remains same as in the motorized scraper, however, the method of optimizing the variable cycle time differs in the sense that for Crawler tractor pulled scrapers, the operating rolling resistance needs to be considered in a different manner than that in motorized one and the manufacturers usually provide draw bar pull vs. speed tables for crawler mounted tractors.

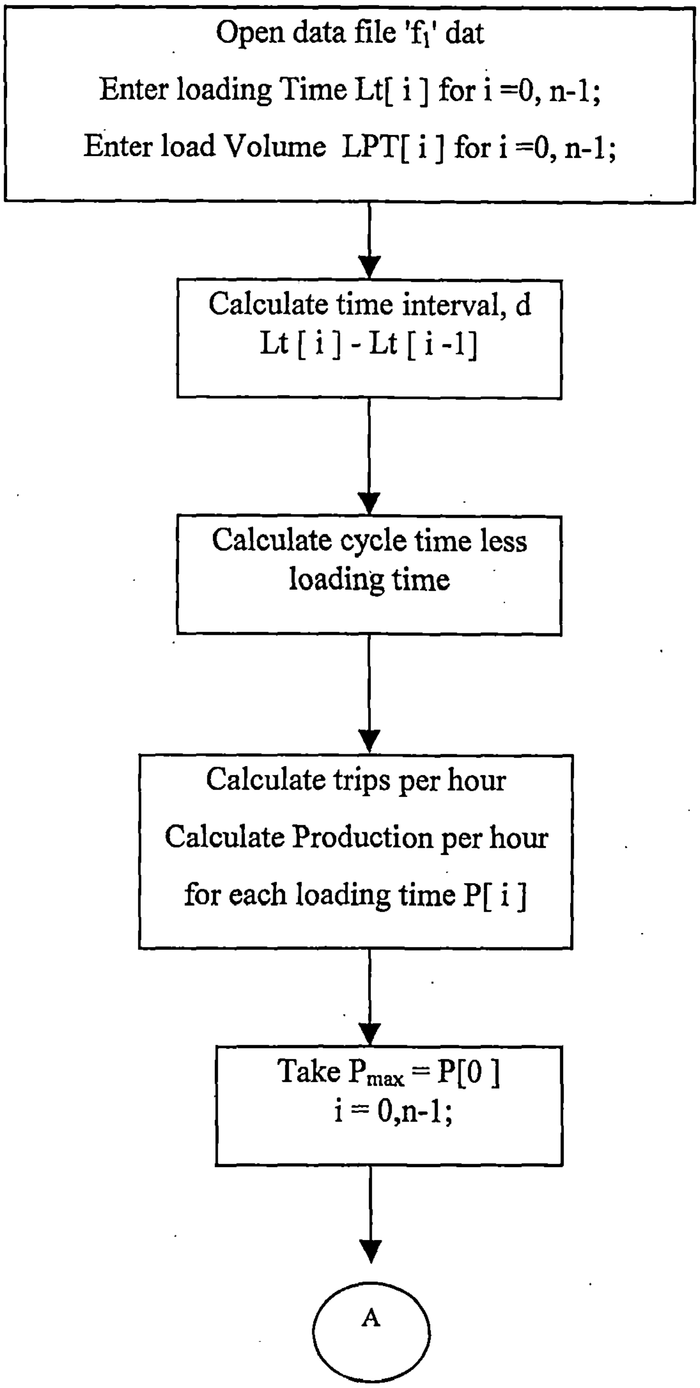
FUNCTIONAL FLOW CHART OF SCRAPER(MOTORISED) MODEL

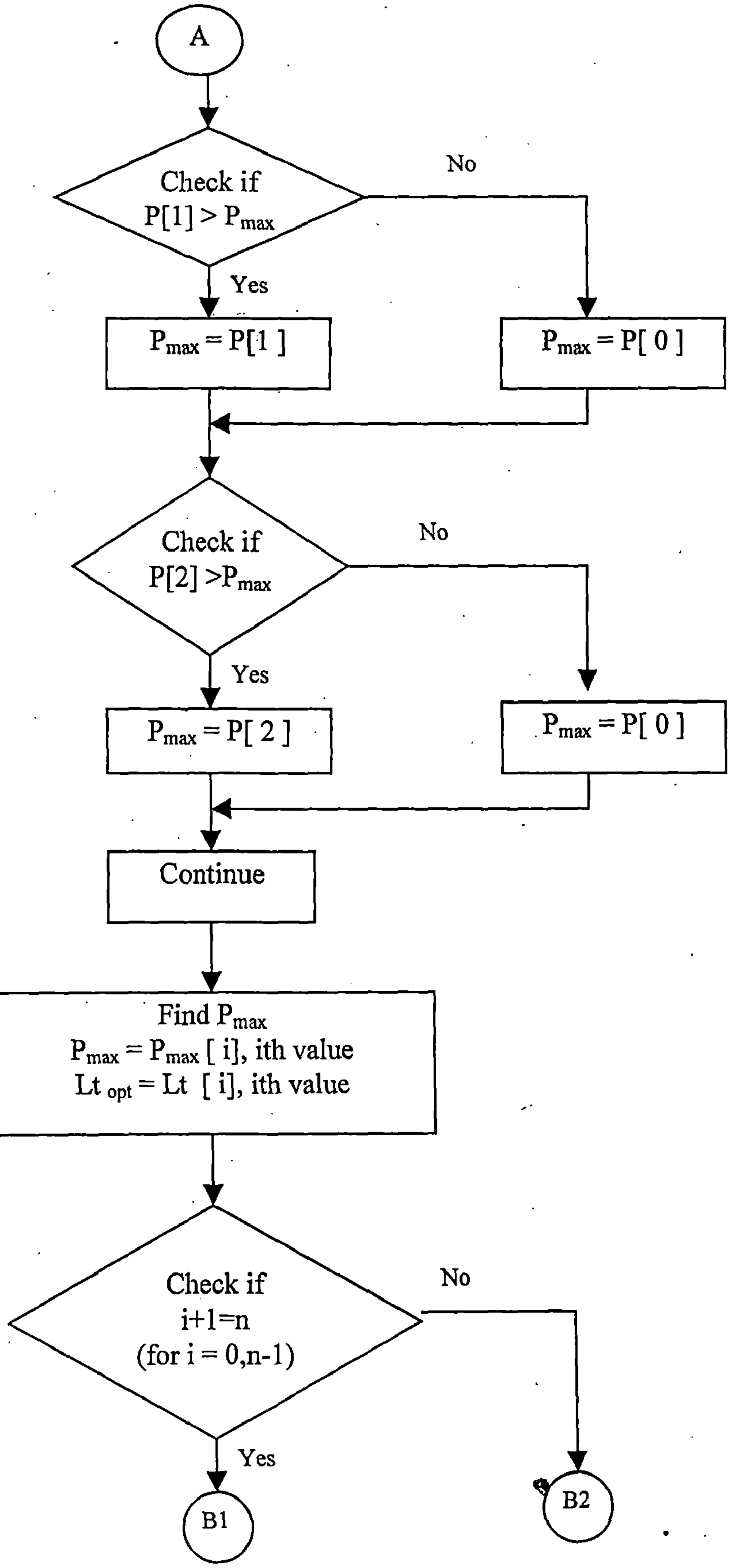


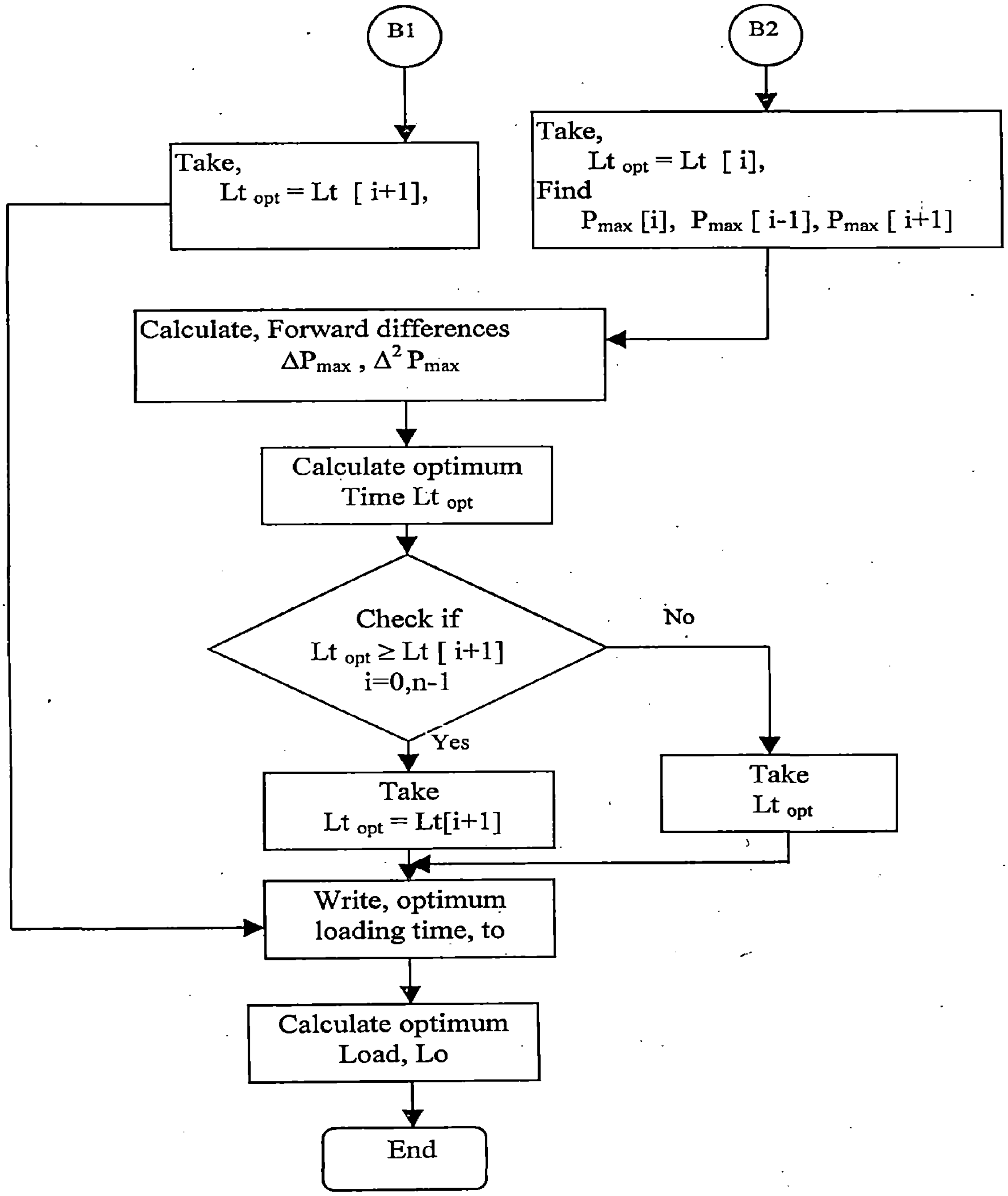

A1
A2
A3
A4
RPm < RPi. The scraper cannot operate
STOP
Find velocity, v_i from RPi
Calculate modified velocity using velocities of given model
Take $v_{modified} = v_{fmax}$
Find $v_{modified}$
Check if $v_{modified} < v_{max}$
No
Yes
Compare $v_{modified}$ & v_{max}. Find final $v_{modified}$
Take $v_{modified} = v_{max}$
Take $v_{modified}$
Calculate average velocity $v_{average}$ = Speed factor x $v_{modified}$
Calculate average velocity $v_{average}$ = Speed factor x $v_{modified}$
Find $v_{average}$
Find haul time, t_{hi}
Continue, do as in block B .Find total haul time t_h
Take payload L = 0. Do as in block B.Find return time t_{ri}. Continue.
Find total return time, t_r
B1


B1
Calculate total travel time, t_travel
Calculate total cycle time less loading time
Calculate optimum loading time using sub program (if LGC is given) ,else take given loading time
Calculate total cycle time
Calculate trips per hour
Calculate productivity, q m³/hr
Calculate no. of scrapers required
Decide type of Push loading. Find pusher cycle time
Find no. of pushers required
Calculate economic cost index of scraper , SrCIN
Calculate economic cost index of fleet FCIN
END
Legend:
1. Block A=
2. Block B =

FUNCTIONAL FLOW CHART OF SUB PROGRAM FOR OPTIMUM LOADING TIME




A
Check if
P[1] > Pmax
No
Yes
Pmax = P[1]
Pmax = P[0]
Check if
P[2] >Pmax
No
Yes
Pmax = P[2]
Pmax = P[0]
Continue
Find Pmax
Pmax = Pmax [i], ith value
Lt opt = Lt [i], ith value
Check if
i+1=n
(for i = 0,n-1)
No
Yes
B1
B2


B1
B2
Take,
Lt opt = Lt [i+1],
Take,
Lt opt = Lt [i],
Find
Pmax [i], Pmax [i-1], Pmax [i+1]
Calculate, Forward differences
ΔPmax , Δ2 Pmax
Calculate optimum
Time Lt opt
Check if
Lt opt ≥ Lt [i+1]
i=0,n-1
No
Yes
Take
Lt opt = Lt[i+1]
Take
Lt opt
Write, optimum
loading time, to
Calculate optimum
Load, Lo
End

G 10,118

In crawler tractor pulled scrapers, the drawbar pull is adjusted by a factor NRRC given by

NRRC = (Rc – 55) x Gc,

where,

Gc = gross weight of crawler tractor (tonne)

Rc = rolling resistance for tractor (kg/tonne)

Hence, the net draw bar pull available is calculated as

NDB = (DB – NRRC) x 0.85

where,

DB = draw bar pull available.

Factor 0.85 is used as a factor of safety [Peurifoy, 1979].

The tractive effort available, Tel is same for both loaded and unloaded condition. It is expressed as

Tractive effort, Tel = Ct x Gc x 1000

where,

Ct = coefficient of traction

If, PR be the total rolling resistance acting on the towed scraper and Gpf be the gross weight being pulled, then the total resistance encountered by the equipment can be expressed as

Total resistance = GR x (Gc + Gpf) + PR

while, PR and Gpf can be written as -

PR = Gpf x RR;

Gpf = L + Ge

Other symbols are defined as below -

L = pay load in the scraper bowl, (tonne)

Ge = empty weight of the scraper , (tonne)

GR = grade resistance, kg/tonne

Gpf = gross weight of load being pulled , (tonne)

RR = rolling resistance operating on load being pulled, kg/tonne

On the basis of a comparison of total resistance encountered and the net draw bar pull available in different gears, the maximum speed of travel for a particular tract of haul may be calculated. From the speed calculated (after adjusting it by the speed factor), time of movement for a given distance may be obtained. In this way variable cycle time may be estimated for the given condition.

The calculations for fixed cycle time etc. may be proceeded as in the motorized scraper.

4.3.3 Algorithm

(i) Find adopted heaped capacity Hc using pay load considerations and find corresponding loading time, Lt. as under

$$Hcc = \frac{L \times 1000}{k}$$

If,

$$Hcc > Hcg \quad ,$$

Then take

$$Hc = Hcg$$

Otherwise take

$$Hc = Hcc, \text{ and}$$

$$Lt = \frac{Ltg \times Hc}{Hcg},$$

where,

L = Pay load on the machine in tonne

Hcg = Given heaped capacity of the machine (m^3)

k = Density of the soil in kg/m^3

Ltg = Loading time given

Hcc = Calculated heaped capacity (m^3)

Hc = Adopted heaped capacity (m^3).

(ii) Estimates factor NRRC for crawler tractor,

$$NRRC = (Rc - 55) \times Gc$$

(iii) Estimate Net Draw bar pull available in each gear as

$$(NDB)_i = (DB_i - NRRC) \times 0.85$$

(iv) Find maximum net NDB available i.e. NDBm out of all NDBi's

Adjust the NDBm by derating factor.

(v) Find tractive effort available Tel,

$Tel = ct \times Gc \times 1000$, kg

where,

ct = coefficient of traction

Gc = gross weight of crawler tractor

(vi) Check if, $Tel \geq NDBm$,

If, yes then find

(vi-a) Total resistance, TR_i for the i[th] section,

$$TR_i = GR_i (Gc + Gpf) + PR$$

(vi-b) Check if, $NDBm \geq TR_i$ and $TR_i > 0$

If yes, then find

(vi-c) Maximum speed attainable by vehicle using performance table (DB vs velocity). On the basis of comparison of TR_i and NDB_i's of crawler tractor find operating speed v_m.

(vi-d) Compare v_m and v_{max}.

where,

v_{max} is the maximum speed based on legal speed limit or other safety restrictions.

Finally obtain speed $v_{average}$. It is the velocity to be used in calculating the time of haul.

(vi-e) If, $TR_i \leq 0$, then

Take $v_m = v[n]$ and compare v_m with v_{max}. Finally obtain speed $v_{average}$.

(vi-f) If, $NDBm < TR_i$,

Then, write "the scraper can not operate under the given conditions".

(vi-g) If Tel < NDBm,

i.e. net draw bar pull is limited by the tractive effort,

Then maximum usable drawbar pull is given

NDBm = TEl

Again, repeating steps from (v-a to v-g) calculate the operating speed for different sections.

(vi-h) For the return journey, take,

Tractive effort TEE = ct x Gc x 1000, and

Gpf = Ge.

Repeat steps from (v-a to v-g) and calculate the operating speed for different sections in return journey.

(vii) Compute time t_i for given i^{th} section as

$$t_i = \frac{d_i}{(v_{average})_i} \times 60$$ minutes, except for section 1

for section – 1,

$$t_1 = L_t + \left(\frac{d_1 - 0.1}{(v_{average})_i}\right) x60$$ (If load growth curve is not given)

$$= L_t[n] + \left(\frac{d_1 - 0.1}{(v_{averag})_1}\right) x60$$ (If load growth curve is given)

(viii) Compute total time of travel, T by adding haul time and return time

$$T = \sum_{i=1}^{n} t_i$$, for return journey, take Lt or Lt[n] = 0 for t_1

(ix) Choose dump, delay and spotting time D, from Table 4.5 and Table 4.6.

(x) Estimate cycle time less load time TC as

TC = T + D

(xi) Optimize load time Lt using load growth curve. Utilize the same scheme as in motorized scraper to obtain optimal load time, t_o.

(xii) Compute cycle time,

$$CTC = TC + t_o$$

(xiii) Compute trips per hours,

$$(TPS) = \frac{50}{CTC} \quad \text{(for 50 minute hour)}$$

(xiv) Compute productivity of one scraper q as -

$$q = LP_{max} \times (TPS) \times Lf \times Cf \quad \text{(if load growth curve is given)}$$

$$= Hc \times TPS \times Lf \times Cf \quad \text{(if load growth curve is not given)}$$

where,

LP_{max} is the optimum load obtained from load growth curve

Cf is the correction factor.

(xv) Estimate total number of scrapers required

$$Sr = \frac{Q}{q},$$

where Q = total earthwork required in m^3/hr

(xvi) Based on type of loading (Back track loading/chain loading), compute Pusher cycle time, PCT.

(xvii) Compute number of Pushers

$$Pu = \frac{CTC}{PCT}$$

(xviii) compute cost index number for one scraper

$$SrCIN = \frac{Osr}{q},$$

where,

Osr is the hourly owning & operating cost of one scraper.

(xix) compute cost index number CIN of fleet

$$FCIN = \frac{(Pu \times Opu + Sr \times Osr)}{(q \times Sr)}$$

Where,

Opu is hourly owning & operating cost of one Pusher

(xx) Repeat steps from (i) to (xix) and find minimum cost index number FCIN amongst various alternatives

Select the model / size and number / numbers of the scraper giving minimum FCIN.

The necessary flow chart enclosed presents the steps of program of optimization model of towed scraper.

The program (with load growth curve) has been given in Appendix – C and the program (without load growth curve) has been given in Appendix - D.

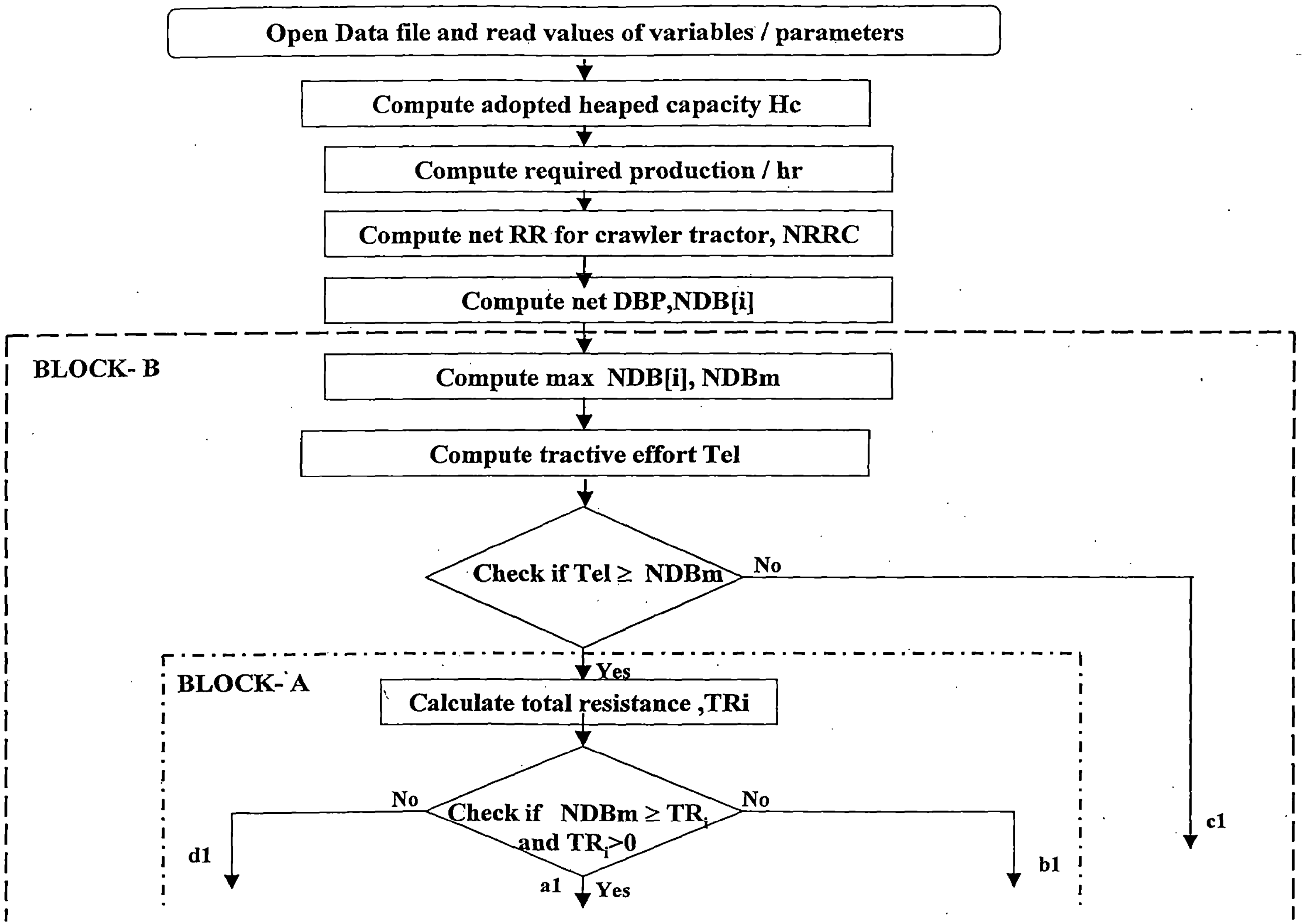
4.4 OPTIMIZATION MODEL FOR BULLDOZER

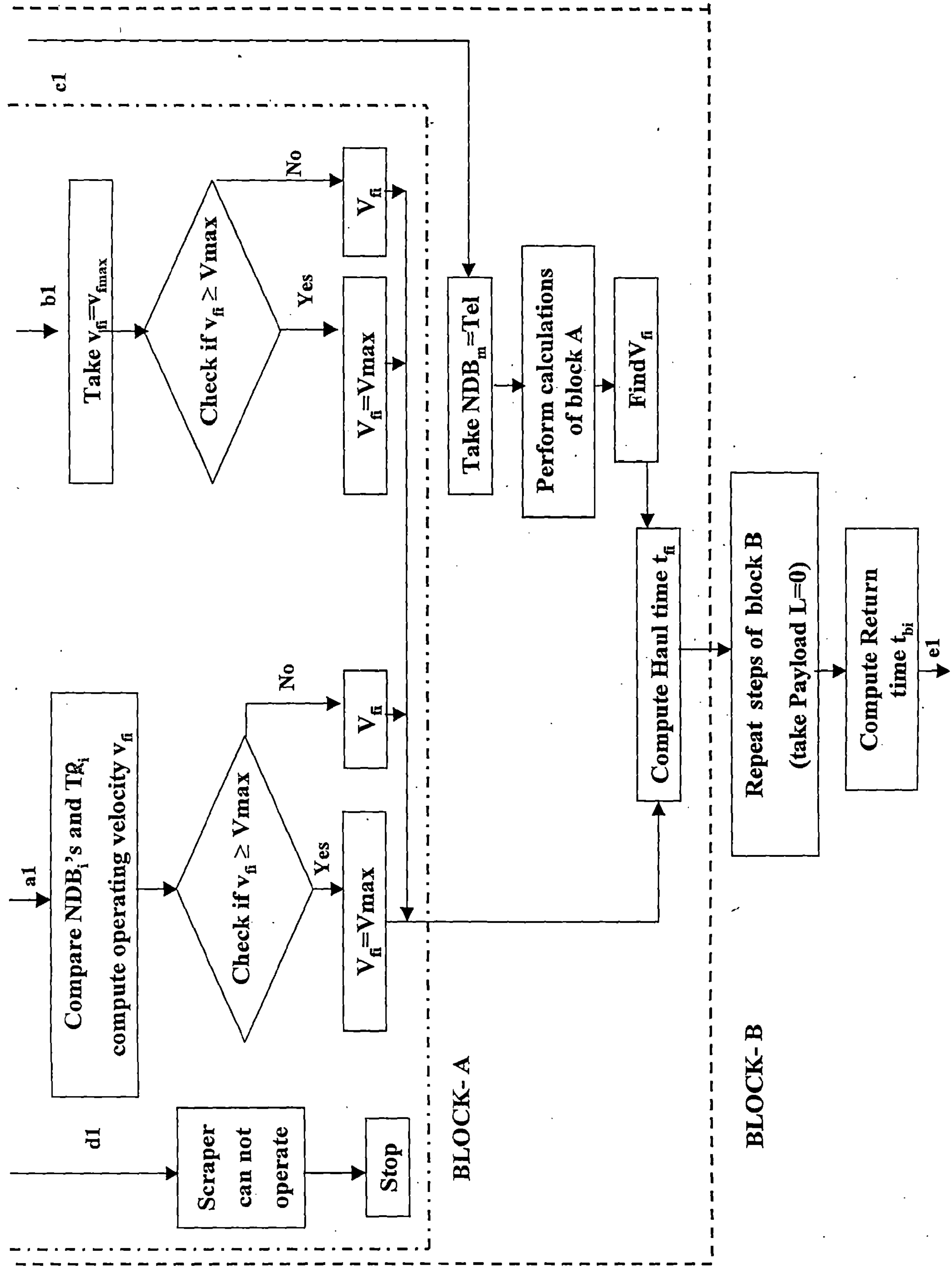
4.4.1 Preliminary Remarks

Tractor is a self powered earthmoving equipment. It may be crawler tractor or a wheel type tractor. A tractor equipped with a front mounted blade is known as a dozer or Bulldozer used primarily for dozing the earth for hand distances up to approximately 300 ft. [Peurifoy 1985]. Machine normally cuts for 50 ft (i.e. 15 m) then drifts blade load to dump over a high wall. [Caterpiller Performance Handbook, 2nd ed.]. The output of bulldozer varies with the conditions under which it operates. The crawler mounted machine has an advantage on short hauls with soft or muddy ground and a wheel mounted machines have an advantage on longer hauls & firm ground.

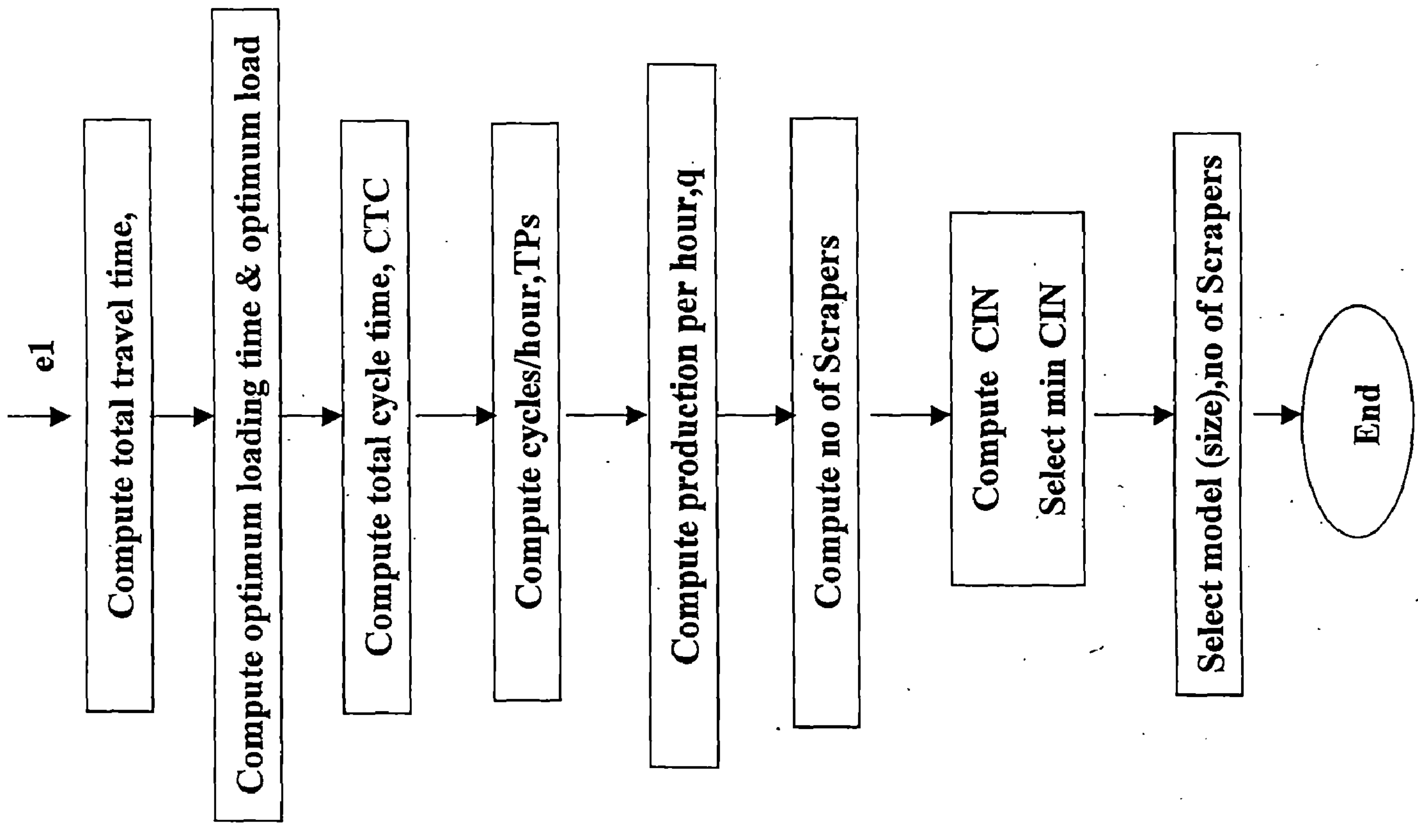
The optimization model of bulldozer carries out the production studies or outputs by considering the load handled by the dozer being equal to the approximate capacity of a bulldozer and based on the assumption that, it pushes the earth in a slot or trench with minimum end spillage. The approximate capacity of a bulldozer may be determined from the size of the load being pushed by the blade . Although the shape of the front slope of the earth will be irregular, it may be assumed as 2:1 slope. The output of a dozer is then related to required production (in m^3/hr), thereby giving the number of bulldozers for a chosen size. The size and number obtained is then, related to its cost, giving its cost performance in the form of CIN (Cost Index No). For the models available, CIN value may be computed and the model giving the minimum CIN value is selected for its size and numbers.

FUNCTIONAL FLOW CHART OF SCRAPER (TOWED) MODEL





a1
Compare NDB_i's and TR_i compute operating velocity v_fi
Check if v_fi ≥ Vmax
Yes
No
V_fi=Vmax
V_fi
b1
Take v_fi=v_fimax
Check if v_fi ≥ Vmax
Yes
No
V_fi=Vmax
V_fi
c1
Take NDB_m=Tel
Perform calculations of block A
Find V_fi
Compute Haul time t_fi
d1
Scraper can not operate
Stop
BLOCK- A
Repeat steps of block B (take Payload L=0)
Compute Return time t_bi
e1
BLOCK- B



e1
Compute total travel time,
Compute optimum loading time & optimum load
Compute total cycle time, CTC
Compute cycles/hour,TPs
Compute production per hour,q
Compute no of Scrapers
Compute CIN
Select min CIN
Select model (size),no of Scrapers
End

4.4.2 Model formulation

The production of bulldozer may be estimated by the following relation.

Production/hr = material moved per cycle x number of cycles performed/hr x efficiency factor x correction factor.

The production of dozer can be optimized by optimizing the number of cycles performed/hour and the material moved per cycle.

The material moved per cycle can be obtained by calculating the blade capacity as-

$$\text{Blade capacity} = \frac{1}{2} \times W \times H \times L \quad \text{(theoretically)}$$

where,

H = height of blade

L = length of blade

W = base of the triangle of earth formed

= 2H

Practically,

H ≈ 1.1H

W ≈ 2.2 H,

$$\text{So, the blade capacity, } Bc = \frac{1}{2} \times (1.1H) \times (2.2H) \times L$$

$$= \frac{2.42}{2} \times H^2 \times L = 1.21 \times H^2 \times L$$

However, based on observations, the International Harvester Company, U.S.A. has suggested [Varma Mahesh, 1997] that the blade capacity can be expressed as

$$\text{Blade capacity} \approx \frac{2.42}{3} \times H^2 \times L \approx 0.80 \times H^2 \times L$$

The number of cycles performed per hr. are based on the total cycle time as under

$$\text{No. of cycles/hr} = \frac{60}{\text{cycle Time}}$$

where,

Cycle time (minute) = Fixed cycle time + variable cycle time.

Fixed time includes time for loading, shifting gears and turning (if involved) Table 4.7 shows the values suggested for dozer fixed cycle time – [Nunnally, 1977]:-

Table 4.7 : Dozer Fixed Cycle Time

Operating conditions	Time (min)
Power shift transmission, average	0.05
Direct drive transmission, average	0.10
Hard digging	0.15

The variable time is the total travel time for dozing and returning. The dozer operating speeds are given in Table 4.8 [Nunnally 1977].

Table 4.8 : Dozer Operating Speeds

Operating conditions	Direct drive	Power shift
Dozzing : Hard materials, haul 30 m or less	First gear	2.5 km/hr
Hard materials, haul over 30 m	Second gear	3.2 km/hr
Loose materials, haul 30 m or less	Second gear	3.2 km/hr
Loose materials, haul over 30 m	Third gear	4.0 km/hr
Return : 30 m or less	Reverse speed in gear used for dozing	Maximum reverse speed in second range
Over 30 m	Highest reverse speed	Maximum reverse speed in third range.

The efficiency factor is obtained by considering 50 min. hr or 45 min hr. as under

Efficiency Factor, eff. = $\frac{50}{60}$ for 50 minutes an hour

= 45/60 for 45 minutes an hour.

From the given required target of production, the no. of dozers required can be obtained. By considering the hourly owning & operating cost of each dozer model, the cost performance can be evaluated by CIN as,

$$CIN = \frac{\text{O \& O cost/hour}}{\text{output/hour}}$$

Min. CIN gives the required optimized model of the dozer and its numbers for the given conditions.

4.4.3 Algorithm

(i) Compute the blade capacity, Bc using following relation

$$Bc = \frac{2.42}{3} \times H^2 \times L \qquad (m^3)$$

Where H = height/width of blade

L = length of blade

(ii) Calculate the total resistance, TR_i for the i^{th} section encountered by the machine

$$TR_i = GR_i \times Gc + E_p \qquad \text{for mild slopes upto } 12^o$$

Where,

TR_i = total resistance (kg)

Gc = Gross weight of machine in tonnes

GR_i = grade resistance, kg/tonne

Ep = force required to push the earth (kg).

The value of E_p may be approximated by the following relation,

$E_p = k \times Bc \times \mu$

where,

k = density of earth being pushed (kg/m^3)

Bc = Blade capacity (m^3)

μ = Coefficient of friction

The coefficient of friction in its limiting value may be taken as the coefficient of traction.

(iii) The calculation of net drawbar pull is same as in the Towed scraper model. By comparing the total resistance with the Net Drawbar Pull supplied by the machine, the haul speed can be obtained as in the Towed scraper model.

It has been suggested that dozing upto 30 m is usually done in lowest gear. Return speed can be obtained from reverse speeds supplied by the manufactures. Usually return is performed in maximum reverse speed as shown in Table 4.8

(iv) Dozing and return times are calculated by dividing the travel distance by dozing & return speeds respectively

Total travel time, T (min) = dozing time + Return time

$$= \left(\frac{\text{Dozing distance}}{\text{Dozing speed}} + \frac{\text{Return distnace}}{\text{Return speed}} \right) x60$$

(v) Total cycle time , CTC is obtained as

CTC = Travel time + fixed time + 0.03xT + 0.01 x T

(fixed time may be taken from Table 4.7).

(vi) Total cycles performed per hr are computed as

$$TPS_i = \frac{60}{\text{cycle time}} = \frac{60}{CTC}$$

(viii) Efficiency factor, $\text{eff.} = \frac{50}{60}$, for 50 min hr

$= \frac{45}{60}$, for 45 min hr

(ix) Production of one Bulldozer, q in m^3/hr is given by

$q = Bc \times TPS \times eff. \times Cf$ (Lm^3/hr)

or, $q = Bc \times TPS \times eff. \times Lf \times Cf$ (Bm^3/hr)

where,

Cf is the correction factor and

Lf is the load factor given by

$$\text{Load factor} = \frac{\text{Bank measure volume}\,(m^3)}{\text{Loose measure volume}\,(m^3)}$$

(x) The number of Bulldozers required can be obtained as ,

$$n = \frac{Q}{q}$$

where,

Q = Required production rate (m^3/hr)

(xi) For the hourly owing & operating cost of bulldozer being Ob. the cost Index number is given by

$$CIN = \frac{Ob}{q}$$

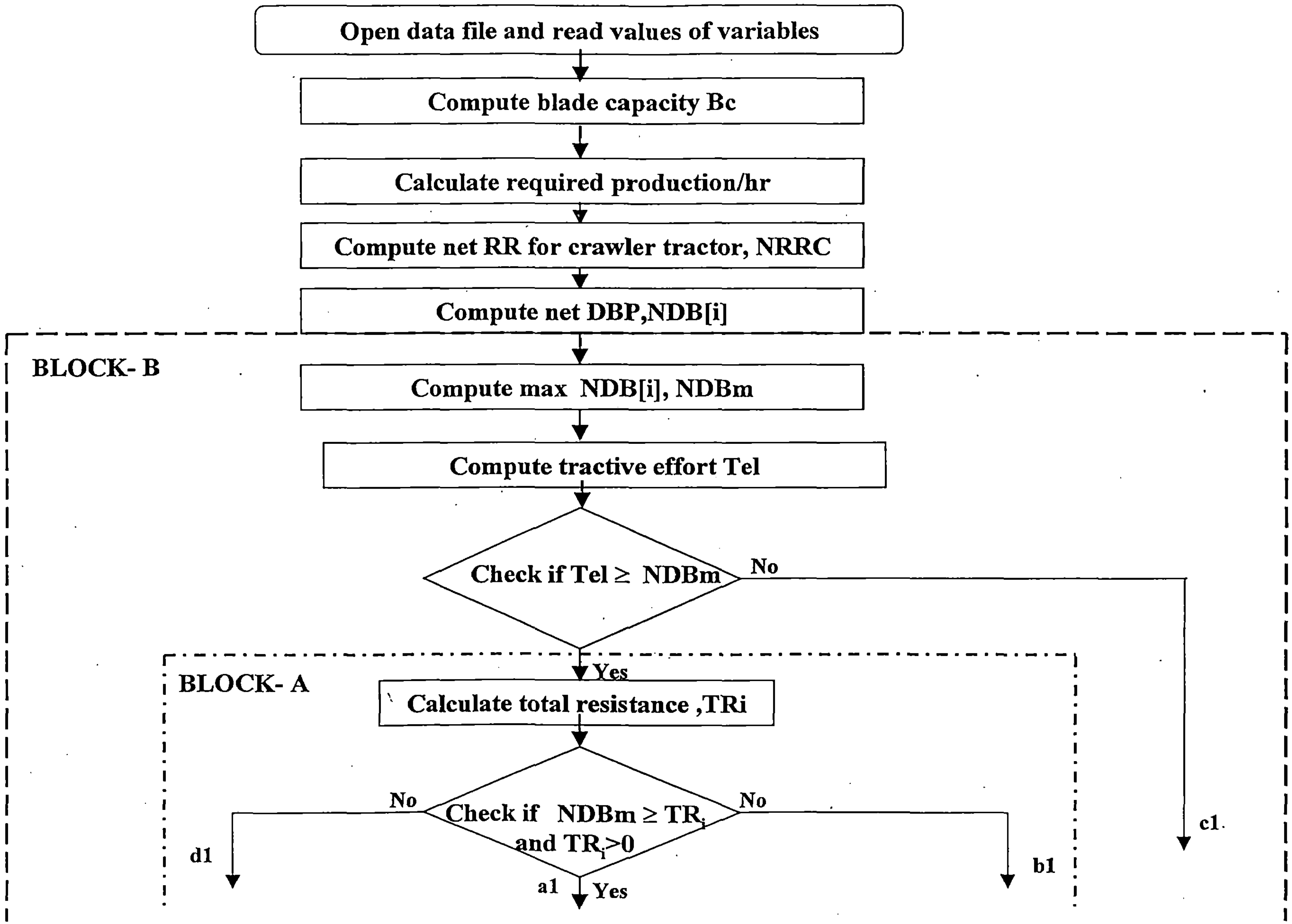
(xii) Select the model giving minimum value of CIN

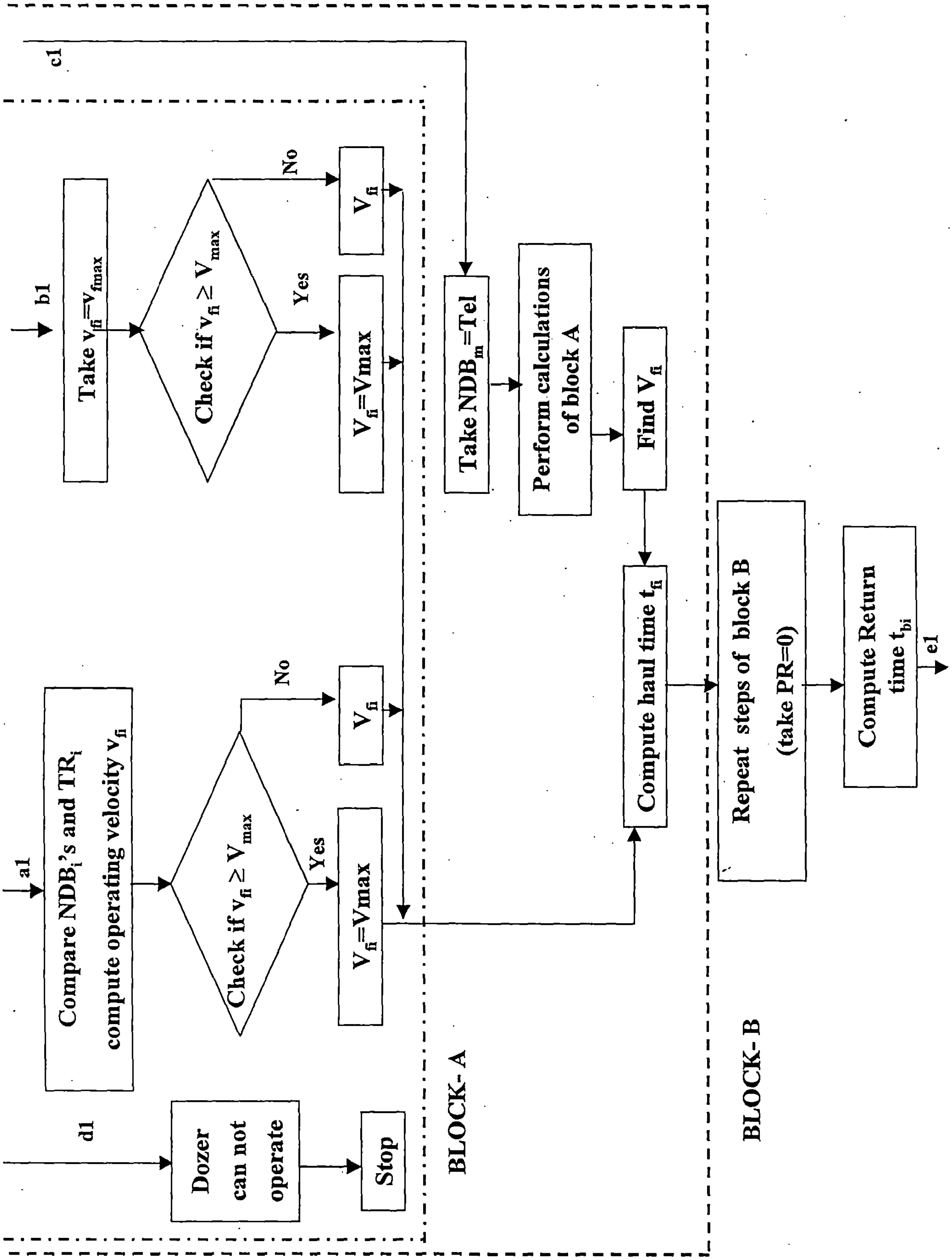
(xiii) Finally from the selected model of bulldozer, decide the size and its number/numbers.

The necessary flow chart -3 enclosed presents the steps of program of optimization model of bulldozer.

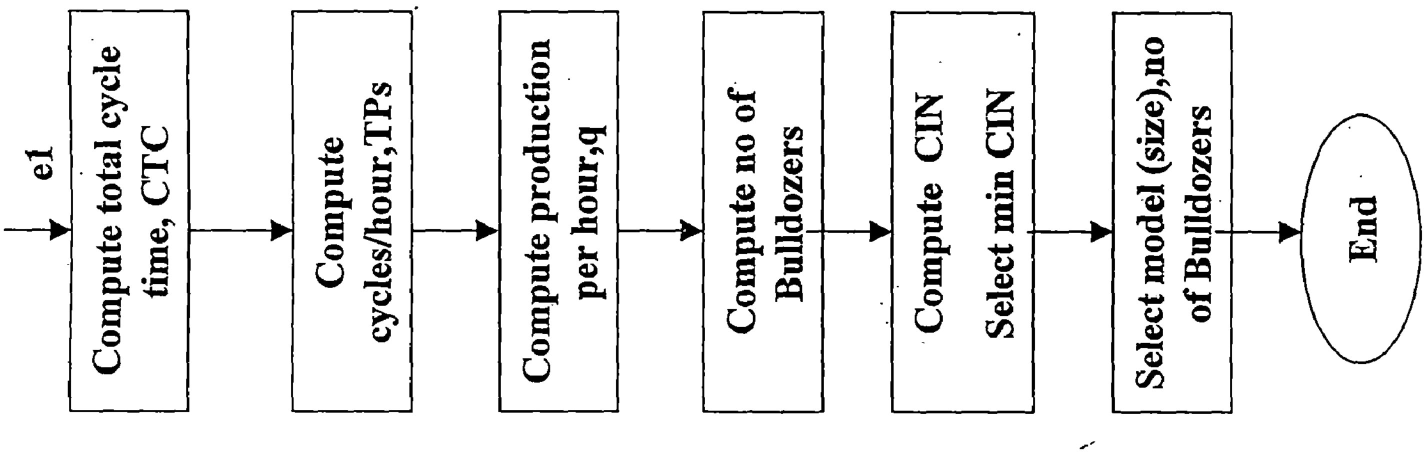
The program has been enclosed in Appendix – E.

FUNCTIONAL FLOW CHART OF BULLDOZER MODEL





a1
b1
c1
d1
Take $v_{fi}=v_{fmax}$
Check if $v_{fi} \geq V_{max}$
Yes
No
V_{fi}=Vmax
V_{fi}
Compare NDB_i's and TR_i compute operating velocity v_{fi}
Dozer can not operate
Stop
BLOCK- A
Take NDB_m=Tel
Perform calculations of block A
Find V_{fi}
Compute haul time t_{fi}
Repeat steps of block B (take PR=0)
Compute Return time t_{bi}
e1
BLOCK- B


e1
Compute total cycle time, CTC
Compute cycles/hour,TPs
Compute production per hour,q
Compute no of Bulldozers
Compute CIN
Select min CIN
Select model (size),no of Bulldozers
End

4.5 OPTIMIZATION MODEL FOR HAULERS

4.5.1 Preliminary Remarks

Dump trucks are used for hauling excavated material over distance between 1 and 10 km but longer journey are possible. Shovels/ excavators/ loaders are used to load the haulers depending on the condition. The system productivity of an earthmoving operation is limited by the production of the loading facility [Farid & Koning 1994]. Regardless of the size, number and speeds of the hauling units, the ability of the loading facility to load the haul units will determine the maximum productivity of the system [Dougles D. Gransberg 1997]. Therefore the characteristics of the loading facility must be considered in the planning of the hauling operations.

4.5.2 Deterministic Model

The Deterministic optimization model for haulers determines the optimum size and number of haul units for an earthmoving project. Loading facility characteristic are an integral part of the model. The model in based on Gransberg's improved optimization model. It estimates wasted time due to human error, mechanical failure etc and allocates a portion of it to each cycle to determine sustained cycle time and sustained productivity.

This differs from practice of using 45-50 min " productive hour". The model determines the number of haul units based on the production achieved and also on the basis of total cost involved for a given size of hauler. It is important to note that the ultimate goal of optimizing a hauling system is to maximize the productivity while minimizing total cost. In the model, for a chosen loading facility, a cost index number is derived to select optimum haul unit size and their number. Minimum CIN provides the optimum size and number of the haul units for the given project.

4.5.2.1 Model Formulation

Initially, on the basis of project production requirements i.e. amount of earthwork required M, and time span available, a loading facility (loader) is chosen and the number of loaders Sn is determined. Next, applying thumb rule i.e. the size of haulers being approximately 4-5 times the loading facility [Peurifoy 1985] is selected for initial trial.

From payload considerations, its adopted heaped capacity Hc is determined. The process then begins for selecting the optimum size and number of hauler for one loader as under –

Computation of total resistance involved gives the required rimpull. From the rimpull vs speed table, the maximum attainable speed can be found out for the existing condition (Total resistance, + ve or – ve). The maximum speed is then adjusted by using speed factors to give the operating speed. These factors are given in Table 4.9(a) & 4.9(b). In this way, the total travel time (i.e. variable cycle time) is obtained. Fixed cycle time includes dump time and delay time. It is taken from Tables 4.10 and Table 4.11.

Table 4.9(a) : Factors for Conversion of Maximum Speed to Average Speed – Unit Starting from Stop (after Havers & Stubbs, 1971)

Haul-road	Downgrade*			Level*			Upgrade*		
length in ft	1	2	3	1	2	3	1	2	3
0-250	0.45	0.40	0.35	0.38	0.34	0.30	0.30	0.25	0.20
251-500	0.45	0.48	0.42	0.44	0.40	0.36	0.36	0.31	0.26
500-750	0.63	0.46	0.50	0.50	0.46	0.42	0.42	0.37	0.32
751-1000	0.72	0.64	0.56	0.56	0.52	0.48	0.47	0.44	0.38
1001-1250	0.80	0.72	0.62	0.62	0.58	0.54	0.55	0.50	0.44
1251-1500	0.87	0.80	0.70	0.68	0.64	0.60	0.60	0.55	0.50
1501-1750	0.94	0.85	0.75	0.75	0.70	0.65	0.65	0.60	0.55
1751-2000	1.0	0.90	0.80	0.85	0.80	0.70	0.75	0.70	0.60

- Weight to power ratios : 1 = 136 kg / hp (300 lb / hp) and under : 2 = 137 – 182 kg / hp (301-400 lb / hp) : under : 3 = 183 kg / hr (401 lb) and over.

Table 4.9(b) : Factors for Conversion of Maximum Speed to Average Speed – Unit in Motion when Entering Road Section

Haul-road length in ft	Downgrade*			Level*			Upgrade*		
	1	2	3	1	2	3	1	2	3
0-250	0.70	0.62	0.55	0.65	0.58	0.50	0.60	0.54	0.46
251-500	0.40	0.70	0.60	0.75	0.66	0.56	0.68	0.62	0.52
500-750	0.90	0.78	0.66	0.85	0.74	0.62	0.75	0.70	0.58
751-1000	1.0	0.86	0.72	0.95	0.82	0.68	0.85	0.78	0.64
1001-1250	1.10	0.94	0.78	1.05	0.90	0.74	0.95	0.86	0.70
1251-1500	1.20	1.00	0.84	1.15	0.98	0.80	1.0	0.95	0.76
1501-1750	1.30	1.10	0.90	1.25	1.05	0.85	1.05	1.00	0.80
1751-2000	1.40	1.20	0.96	1.35	1.15	0.90	1.10	1.05	0.84

- Weight to power ratios : 1 = 136 kg / hp (300 lb / hp) and under : 2 = 137 – 182 kg / hp (301-400 lb / hp) : under : 3 = 183 kg / hp (401 lb/hp) and over.

Table 4.10 : Typical Dump Times (after Havers & Stubbs, 1971)

Operating condition	Rear dumper	Bottom dumper
Favourable	1.0	0.4
Average	1.3	0.7
Unfavourable	1.5 – 2.0	1.0 - 1.5

Table 4.11 : Typical Spotting Times (after Havers & Stubbs, 1971)

Operating condition	Rear dumper	Bottom dumper
Favourable	0.15	0.15
Average	0.30	0.50
Unfavourable	0.80	1.00

Total cycle time, CTC = loading time + travel time + dump & delay time .

Number of haulers required, $N = \dfrac{\text{Total cycle time}}{\text{Loading time}}$

Rounding off the optimum number of haul unit up, maximizes loading facility productivity LP ; while rounding down, maximizes the haul unit productivity HP. Therefore both are checked and higher of the two is selected.

Another way to determine the optimum number of haul unit involves analyzing both cases (i.e. rounding up to N1 or rounding down to N2) from cost viewpoint and selecting the number which gives the lesser total cost (TC) to complete the project.

To find optimum haul unit size, the wasted time w per cycle is calculated. Usually this figure should be obtained from field observations, but for simplicity, the model assumes a value of 20% of total cycle time CTC. It gives sustained cycle time Cs, for both cases (N1 or N2) and sustained productivity Ps

The total time TT to complete the haul of a given amount of material M is calculated from sustained cycle time Cs, the haul unit size Hc and the number of haul units N.

From the total time TT, number of haul units N, amount of haul material M, the hourly owning & operating cost of chosen haul unit, and overhead cost IC, The cost index number CIN can be calculated.

For the different models of haul units available, CIN in calculated and the haul unit giving minimum CIN for the given loading facility is selected for its size and number/numbers.

Again for the different loading facilities, the CIN values can be calculated and haul unit size providing minimum CIN should be adopted for its size and number/numbers.

4.5.2.2 Algorithm

(i) Let, the total quantity of earthwork = M (m^3)

The hourly required output = q (m^3/hr)

(based on the number of years for the project, the number of shifts and total number of working hours in a year)

(ii) Choose a loader of the following parameters –

Bucket capacity = b (m^3)

If the cycle time of loader (time to fill and load one bucket), is tl (minutes), then the standard loader productivity, LP is given by,

$$LP = \frac{b \times 60}{tl} \qquad (m^3/hr)$$

(iii) Compute number of loaders, Sn required for earthwork,

$$Sn = \frac{q}{LP}$$

(iv) Choose a hauler of size approximately 4 -5 times the size of loader bucket for the initial trial. The heaped capacity of the hauler is taken as, Hcg.

(v) $Hcg \approx 5 \times LP$ (m^3)

Calculate the adopted heaped capacity Hc of the hauler using payload considerations as in motorized scraper model.

(vi) Calculate tractive effort TEl.

Repeat steps from (i to ix) as in motorized scraper for calculating total travel time T

$$T = \sum_{i=1}^{n} t_i \quad \text{for n number of segments or sections of haul.}$$

(vii) Calculate the time required to load the hauler by the chosen loader i.e. the loading time Lt as -

$$Lt = \frac{Hc \times tl}{b}$$

(viii) Compute total cycle time, CTC for hauler

$$CTC = Lt + T + D,$$

where,

T = total travel time for haulage and return

D = dump and delay time.

(ix) The number of haulers per loader, N is obtained as -

$$N = \frac{CTC}{Lt}$$, round down N to N1, and round up N to N2.

(x) Rounding down N to N1, gives the hauler productivity, HP_1

$$HP_1 = \frac{Hc \times N1 \times 60}{CTC}$$

(xi) Check if, LP > HP1,

If yes, then, calculate A

$$A = N2 \times Lt - CTC$$

where,

A = additional time, each truck spends waiting to be loaded.

So, actual cycle time, ACT = CTC + A

Again calculate, the productivity of haul units by considering N2 units,

$$HP2 = \frac{Hc \times N2 \times 60}{CTC}$$

Select the number of haulers = N2 and

Hauler productivity = HP2

(xii) If $LP \leq HP1'$

Then, choose number of haulers = N1 and

Hauler productivity = HP1

(xiii) If X1 = hourly owning cost of hauler

Z = hourly operating cost of hauler

Hl = hourly owning & operating cost of loader

Then, calculate the total cost of earth work required.

$$TC1 = \frac{M \times CTC \times [(X1 + Z) \times N1 + Hl]}{N1 \times HC \times 60}$$

$$TC2 = \frac{M \times CTC \times [(X1 + Z) \times N2 + Hl]}{N2 \times HC \times 60}$$

If, $TC1 < TC2$, then select number of haulers = N1

Otherwise, select number of haulers = N2

(xiv) To select optimum haul unit size –

Calculate wasted time per cycle, $w \approx 0.20 \times CTC$,

Calculate sustained cycle time CTs = CTC + w (if number of haulers is N1.)

= CTC + w + A (if number of haulers is N2).

Calculate sustained productivity Ps of the haulers,

$$Ps = \frac{60 \times N \times Hc}{CTs},$$

where,

N is the number of haulers based on rounding off decision i.e., either N is equal to N1 or N2, depending upon the productivity out of HP1 and HP2, i.e. N is based on steps number xi and xii above.

(xv) Compute the total time, TT to haul the given quantity M,

$$TT = \frac{M \times CTs}{60 \times N \times Ps} \quad hrs$$

(xvi) Finally obtain cost index number

$$CIN = \frac{TT \times [N(X1 + Z) + IC]}{M}$$

Where,

IC = overhead costs

(xvii) Repeat steps from (v to xv) for other haul unit sizes.

(xviii) Select the haul unit giving minimum value of CIN for its size and number / numbers for a chosen loader.

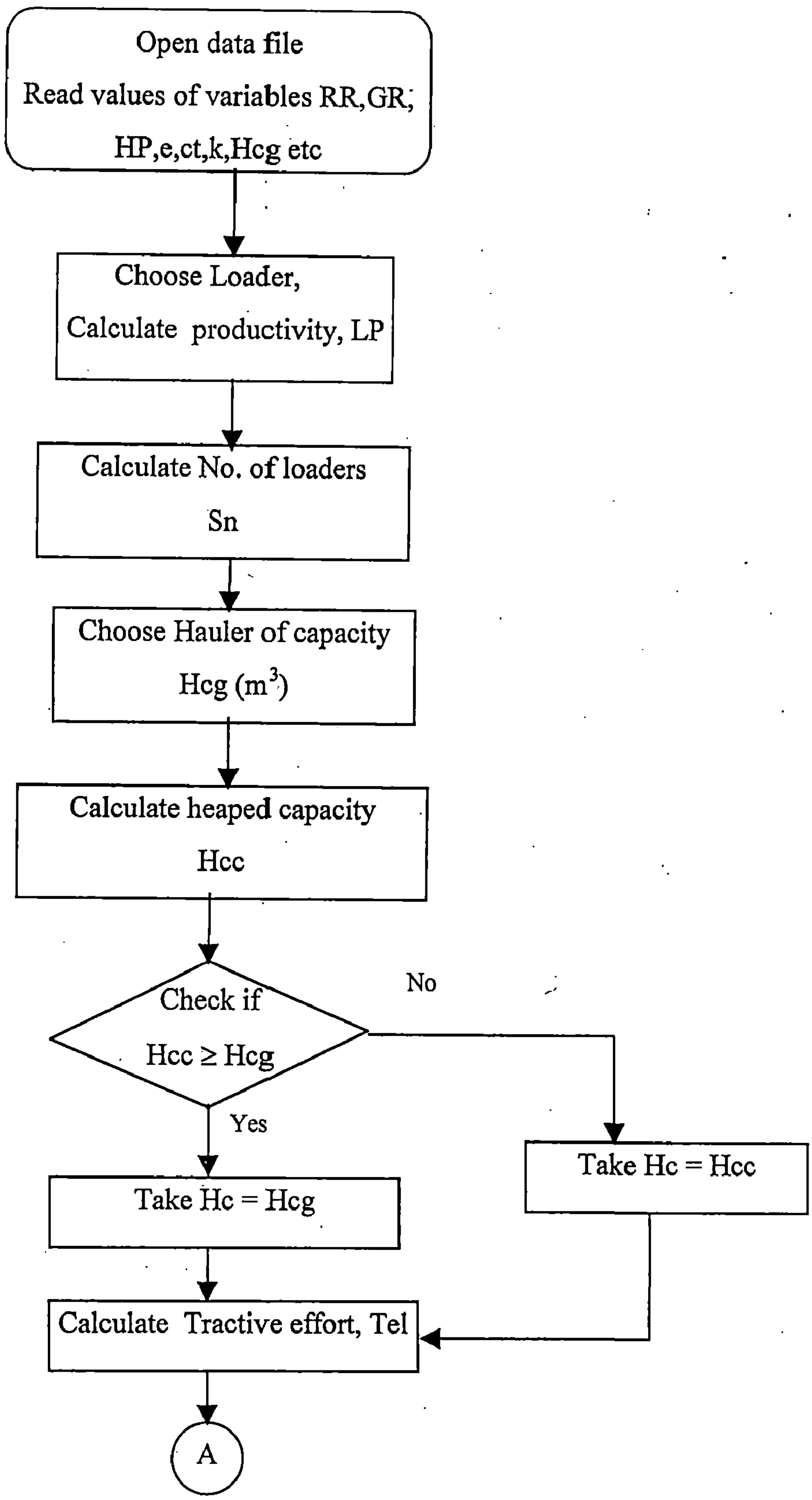
The necessary flow chart enclosed depicts the steps of the program. The program has been enclosed in Appendix – F.

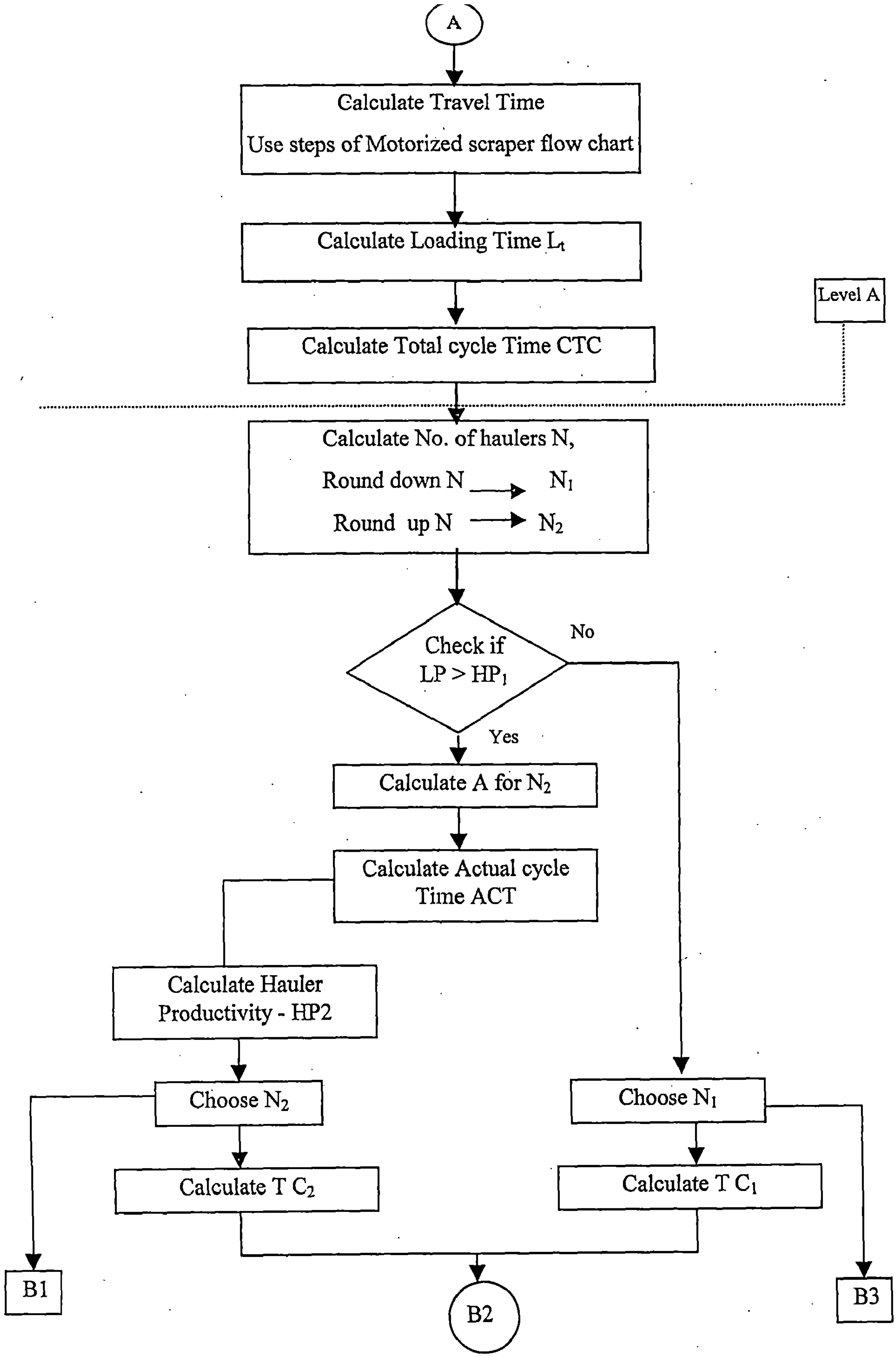
4.5.3 Probabilistic Model

4.5.3.1 Preliminary Remarks

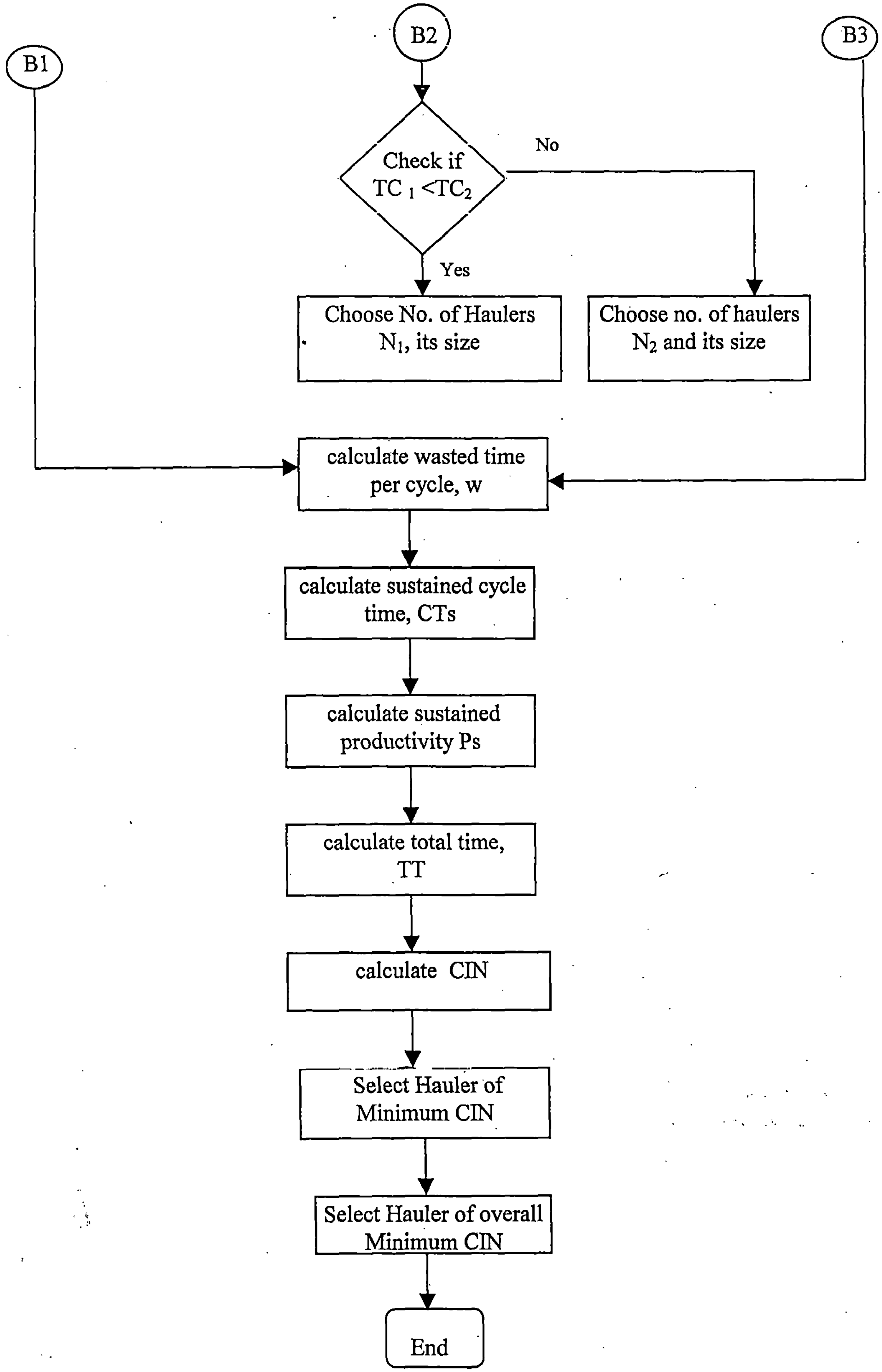
The theory of Queues can be applied to a situation involving an earth loader and hauling units to analyze statistically the cost of excavating / loading and hauling earth when using varying number of hauling units; from this the optimum number of units can be determined. Actual observations and cost determinations made on operating projects have verified the accuracy of this theory [O. Shea, J.B., G.N. Slutkin, L.R. Shaffer, 1964]. Queuing theory uses the exponential distribution to model activity durations because it fits many process operations well and it makes analytical formulation easier [Taha 1987]. When service time and transit times are random, the distributions of both times are negative exponential functions, and the system is cyclic queuing system analyzed by Griffiths [Peurifoy, 1979]. It may be mentioned that the exponential distribution is not appropriate for modelling activity durations of construction operations. The erlang or beta distribution more closely fits actual load and travel times in

FUNCTIONAL FLOW CHART FOR HAUL UNIT
(Deterministic - Optimization - Model)





A
Calculate Travel Time
Use steps of Motorized scraper flow chart
Calculate Loading Time Lt
Level A
Calculate Total cycle Time CTC
Calculate No. of haulers N,
Round down N → N1
Round up N → N2
Check if LP > HP1
No
Yes
Calculate A for N2
Calculate Actual cycle Time ACT
Calculate Hauler Productivity - HP2
Choose N2
Calculate T C2
Choose N1
Calculate T C1
B1
B2
B3


B1
B2
B3
Check if TC 1 <TC2
No
Yes
Choose No. of Haulers N1, its size
Choose no. of haulers N2 and its size
calculate wasted time per cycle, w
calculate sustained cycle time, CTs
calculate sustained productivity Ps
calculate total time, TT
calculate CIN
Select Hauler of Minimum CIN
Select Hauler of overall Minimum CIN
End

construction. However, the under-estimation of production by using exponential distribution may range from 4 – 7% [F. Farid, Thomas L. Koning, 1994].

4.5.3.2 Model formulation

An earthmoving system composed of one excavator and N trucks may be considered as a queuing system that is described as follows : a truck is loaded, travel to dump site, dumps, and returns to the back of the queue or, if there is no queue, begins loading immediately.

The parameters involved in the model are arrival rate of trucks and the servicing (loading) rate by the loader. An arrival is the occurrence marking entry of a truck in the queue and a service is said to be completed when the truck leaves the loader after being loaded.

Let,

n = number of haul units in the fleet,

λ = mean arrival rate of a particular haul unit (arrivals/hr).
= 1/travel time.

μ = mean loading rate of the loader (units/hr)
= 1/loading time

r = ratio of arrival rate to loading rate
= λ/μ
= loading time / travel time.

p_o = probability that no haul unit is available at the loader.

p_t = probability that one or more haul units are available at the loader.
The sum of p_t and p_o must be equal to one, i.e.
$p_t = 1 - p_o$,

Griffiths found that

$$p_o = \left[\sum_{i=0}^{n} \frac{n!}{(n-i)!}(r)^i\right]^{-1},$$

where,

n = number of haulers,

i = digit implying the number of hauler from 0 to n.

So the production rate of equipment is given by

$q = (1 - p_o)$ x normal production of production of loader.

$= (1-p_o) \times LP$ (m^3/hr)

Field observations indicate that, the expected production may be found out by using the equation [Nunnally, 1977],

Expected production = $1.03 \times LP \times p_t$ (m^3/hr)

In modelling the activities, travel time and loading time may be calculated as in the deterministic model of hauler/loader production, thereby giving the values of λ, μ & r. After having selected a haul unit, the cost performance can be evaluated. The size of haul unit alongwith its number giving the minimum cost index number is finally selected.

4.5.3.3 Algorithm

(i) Repeat steps from (i to viii) of Deterministic Optimization Model for hauler.

(ii) Compute mean arrival rate λ of the hauler and mean service rate μ of the loader as -

$$\lambda = \frac{1}{T},$$

$$\mu = \frac{1}{Lt}$$

where,

T = total travel time,

Lt = loading time.

and calculate the value of $r = \lambda/\mu$,

(iii) Calculate approximate optimum number of haulers n_l by using following relation

$$n_l = \frac{1}{r},$$ and roundup this figure.

(iv) Next consider the number of haulers from $n_l - m$ to $n_l + m$ for m = 2.

(v) Compute the probability p_t for each number of haulers,

$$p_t = 1 - p_o,$$

where,

$$p_o = \left[\sum_{i=0}^{n} \frac{n!}{(n-i)!} (r)^i \right]^{-1}$$

(vi) The expected production is given by

Expected production, $P_j = 1.03 \times LP \times p_t$ (m^3/hr)

It is to be noted that LP is taken at 100% loader efficiency because a loader typically operates at or near 100% efficiency when actually engaged in loading [Nunnally, 1977].

(vi) Taking the hourly owning and operating cost of the haul unit as Ohl and for loader as Ol, the fleet cost index number may be obtained by the following equation,

$$(CIN)_j = \frac{j \times Ohl + Ol}{Pj}$$

where,

j = number of haulers

(vii) The minimum value of CIN is selected and the corresponding value of j gives the optimum number of haul units for the chosen loader.

(viii) Repeat steps from (i to viii) for other haul unit sizes.

(ix) Select the haul unit, giving minimum value of CIN.

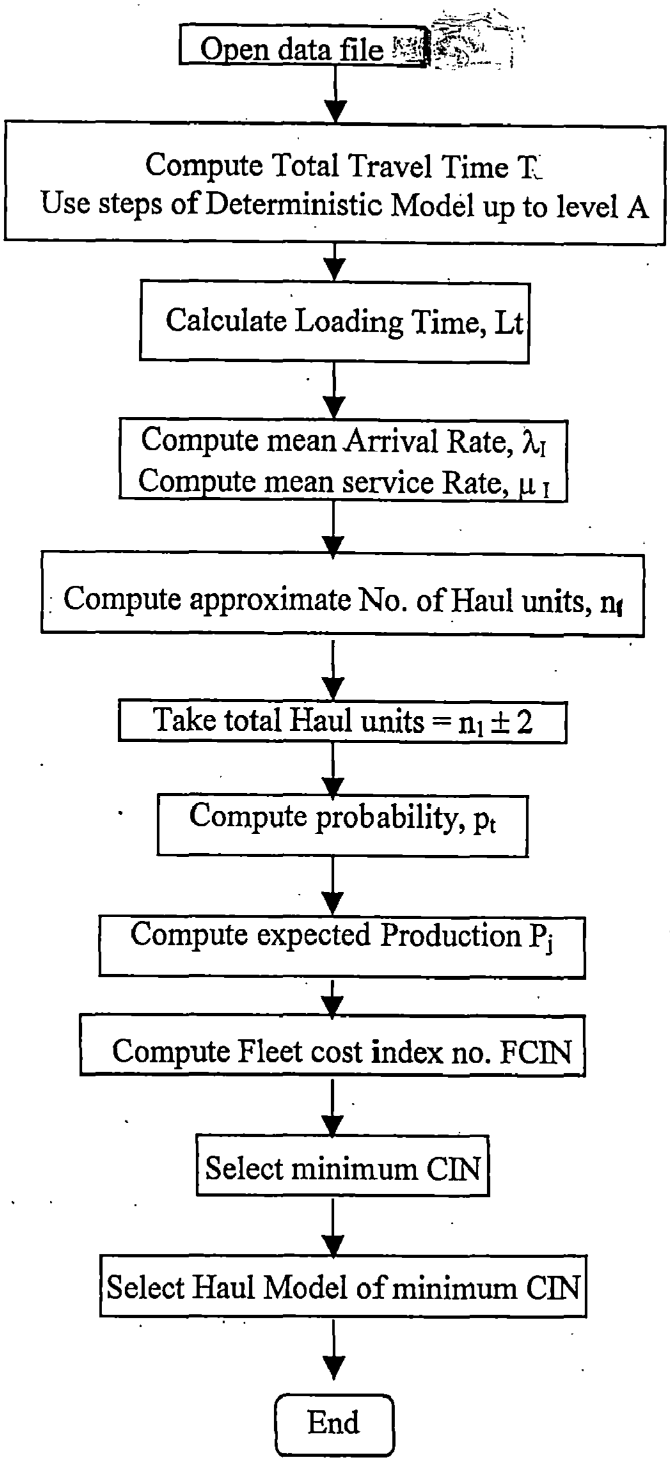
Finally, j = number of haul units

Hcg = size of haul unit (m^3).

The flow chart enclosed shows the steps of the program. The program has been given in Appendix – G.

FUNCTIONAL FLOW CHART FOR HAUL UNIT

(Probabilistic Model)



CHAPTER – 5

ANALYSIS

5.1 GENERAL

The present study comprises of the development of optimization models for (a) Scrapers (motorized) (b) Scrapers (towed) (c) haul units/loader combinations, and (d) Bulldozers(Track mounted) which are primarily concerned with the excavation and hauling of earth. The models finally translate the inputs into the productivity of equipment/equipments. In this chapter analysis is carried out to determine the productivity of different equipments for given parameter values. Before analyzing the productivity of the different equipments, it is of importance to discuss the term 'productivity', itself.

If a single type of machine or equipment is performing the job, then earthmoving productivity is defined as:

Earthmoving productivity = ideal production of the machine × correction factors. 5.1 (i)

For the scraper, the productivity is given by,

$P_{actual} = Hc \times TPs \times Cf.$

For Bulldozer, the productivity is given by

P_{actual} = Volume handled per cycle × cycle rate.

$= Bc \times TPs \times Cf.$

Where,

Bc = Blade capacity of the bulldozer

Hc = Heaped capacity of the scraper

TPs = Trips per hour

Cf = Correction factor.

If two types of machines work in combination, then the earthmoving productivity is dependent upon the production of server and client. Actual productivity for an earthmoving operation is the actual output that will be achieved:

P_{actual} =client cycle rate × volume handled by the client per cycle. 5.1 (ii)

Maximum prime output is the maximum output from the earthmoving system as determined by the output of the prime mover/server.

P_{max} = Server cycle rate × Load volume handled by the server per cycle. 5.1 (iii)

The scrapers excavate earth while in motion for about 50 – 100 m and, thereafter these machines behave as haul units. Effect of various operating parameters on productivity has been studied using optimization model of scrapers.

The bulldozers excavate and push the excavated earth. Effect of various operating parameters has been studied by using the optimization model of bulldozer.

For the loader/dumptruck system the loader is the prime mover and maximum prime output will be achieved if loader is working at maximum utilization and supplied with enough trucks.

Hence for loader /dumptruck system, the actual productivity is also given by equation for bulldozers and scrapers i.e.

P_{actual} = Hc × TPs × Cf.

where symbols are defined above.

The effect of various operating parameters i.e. rolling resistance, grade resistance and distance on the productivity of the haul unit is discussed through the optimization model of haul units by considering Haul pack RD-25 in detail.

5.2 ADOPTED PARAMETER VALUES & MODELS FOR ANALYSIS

Although for the purposes analysis, following values of the variables/parameters are taken in general, yet the values of these parameters can be changed by feeding appropriate data in the computer programs as per site conditions while other parameters are specified wherever required in the analysis.

(i) Efficiency of machine = 0.8

(ii)	Co-efficient of traction, ct	= 0.45	for wheeled equipments
		= 0.6	for track mounted equipments
(iii)	Rolling Resistance, RR	= 20 Kg/tonne	
(iv)	Grade Resistance, GR	= 0.0	
(v)	Efficiency factor	= 1 (i.e for ideal output)	
(vi)	Load Factor, Lf	= 1 (i.e. for ideal output)	
(vii)	Density of soil, k	= 1400 kg/m^3	

For scrapers, the following values of fixed cycle times are taken corresponding to favouable conditions in Tables 4.4, 4.5 & 4.6.

(a) For single engine overhung (push loaded)

Loading time	= 0.6 min
Dump time	= 0.4 min.
Spot & delay time	= 0.4 min.

(b) For tandem scrapers,

Loading time	= 0.8 min.
Dump time	= 0.6 min
Spot & delay time	= negligible

Heaped capacity has been adopted at 1:1 ratio (SAE rating) for scrapers, and 2:1 (SAE) for Haul units.

Scrapers considered include, scraper 229 H, Terex TS-14B (Manufactured in India). Towed scrapers include 463G push loaded by D8H (DD) and 435 G push loaded by D8H (DD).

Bulldozers taken for study include D31A, D50A, D80A, D155A (all manufactured in India). Values of fixed cycle times for bulldozer are taken from Table 4.7.

Haul units considered include Haulpack-Rear Dumper-25, RearDumper-35. To analyse the performance (productivity vs haul distance) of haul units, following fixed time values have been considered –

Dump & delay time = 1.0 min (favourable condition)

= 1.3 min (average condition)

Spot time = 0.15 min (favourable condition)

= 0.30 min (average condition)

5.3 SCRAPER

The trend of productivity is decreasing with increase in haul distance as shown in fig. 5.1 & 5.2. Scraper model 229H is single engine scraper of FHP 332 and heaped Capacity of 16.06 m^3, while Terex TS 14B is twin engine scraper of FHP 288 and heaped Capacity of 15.3 m^3. Productivity at each haul point distance is more for 229 H, but the gap of difference of the productivities becomes narrower with increasing total resistance, as shown in Fig. 5.1 and Fig. 5.2.

HP/gross weight ratio for 229 H and Terex 14B is 6.695 and 6.36 respectively, while HP/empty weight ratio for 229 H and Terex 14B is 12.488 and 12.02 respectively.

The nature of curves obtained can be explained as follows :

For given RR and GR, the rimpull RP required by scraper is given by

$$RP = (RR + GR) \times G \qquad 5.3\text{ (i)}$$

where,

G = Gross vehicle weight (while loaded),

= E + L,

E = Empty weight of vehicle

L = Pay load

RR = Rolling resistance,

GR = Grade resistance

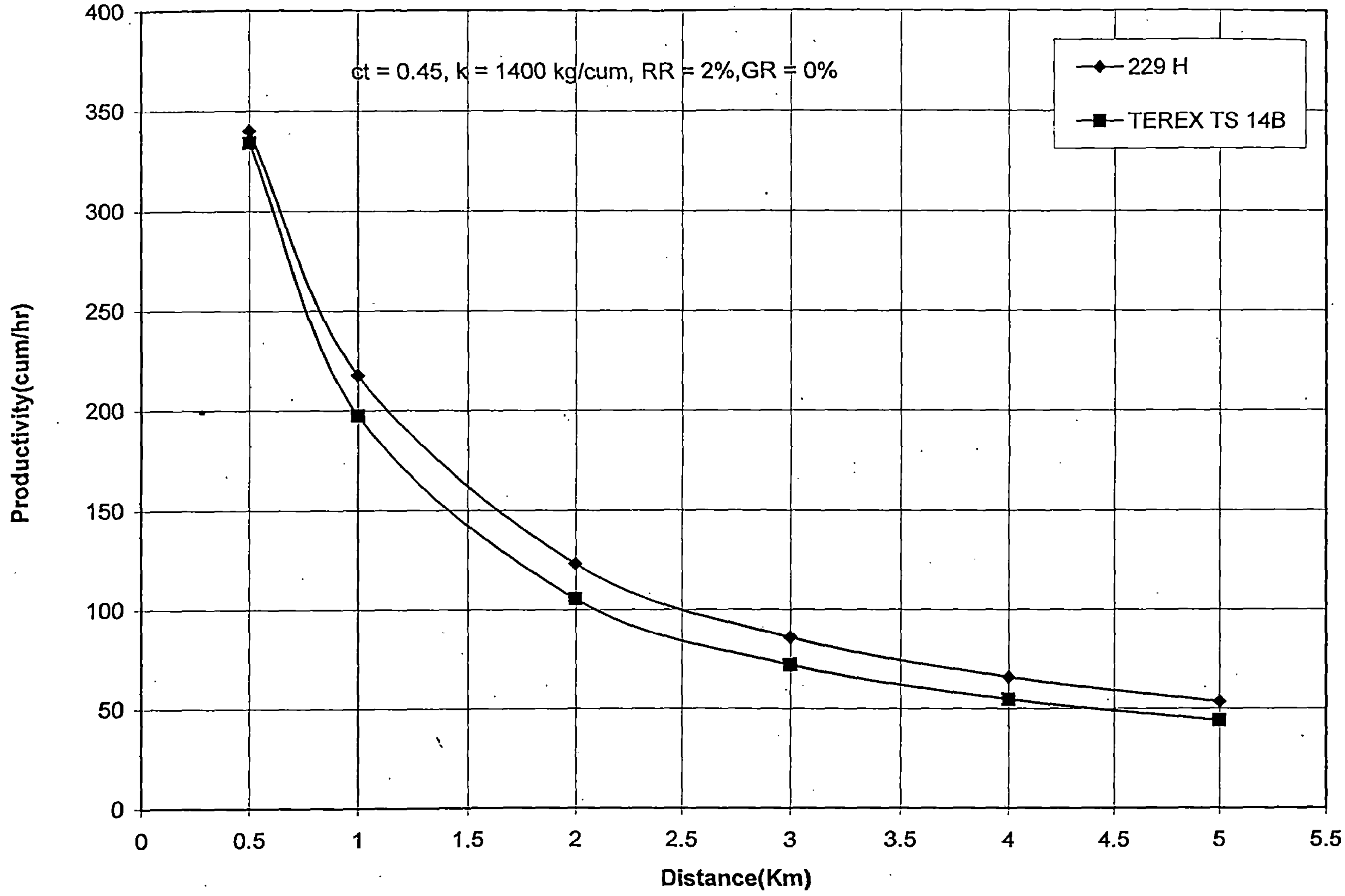


Fig 5.1 Comparison of productivities of scrapers for different haul distances

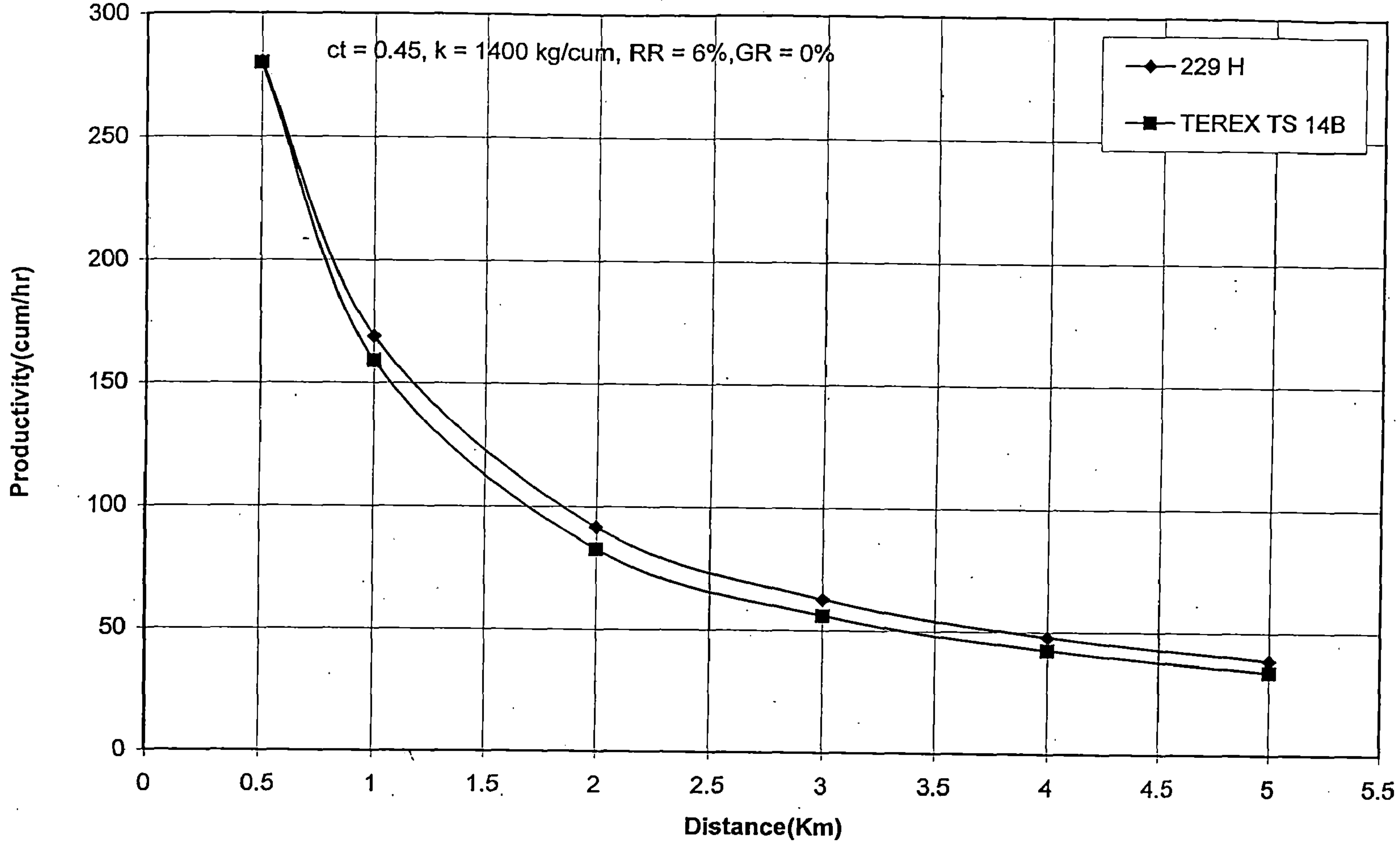


Fig 5.2 Comparison of productivities of scrapers for different haul distances

Also the rimpull is given by

$$RP = \frac{270xHPxe}{v}, \qquad 5.3\ (ii)$$

where,

e = efficiency of the machine

From eq. 5.3 (i) and 5.3 (ii)

$$v = \frac{270xHPxe}{(RR + GR)G} \qquad 5.3\ (iii)$$

Productivity of the scraper is given by

$$P = Hc \times TPS \times Lf$$

i.e. $$P \propto Hc \times TPS \qquad 5.3\ (iv)$$

where,

$$TPS\ \alpha\ \frac{1}{d\left[\frac{1}{v_1} + \frac{1}{v2}\right]} \text{, for given fixed cycle time} \qquad 5.3\ (v)$$

where,

v_1 and v_2 are the velocities of haul and return respectively.

Finally, from equations 5.3 (i), (ii), (iii), (iv) and (v), productivity can be expressed as

$$P\ \alpha \frac{Hc}{d} x \left[\frac{k_1.HP}{(RR_f + GR_f)G + (RR_b + GR_b)E}\right] \qquad 5.3\ (vi)$$

or,

$$P\ \alpha\ K_1/d, \qquad 5.3\ (vii)$$

where,

$$K_1 = \frac{Hc}{d} x \left[\frac{HP}{(RR_f + GR_f)G + (RR_b + GR_b)E}\right]$$

The symbols used in the above discussion are defined below,

k_1 = 270 x e,

RR_f = Rolling resistance for onward journey (while loaded)

GR_f = Grade resistance for onward journey (while loaded)

RR_b = Rolling resistance for return journey (while empty)

GR_b = Grade resistance for return journey (while empty)

Hence,

(i) For a given model Hc, HP, G and E being fixed, and given values of RR_f, GR_f, RR_b and GR_b,

$$\text{Productivity, } P \propto \frac{1}{d}$$

i.e. as haul distance increases, productivity decreases.

(ii) For different models, and given fixed distance d,

$P \alpha K_1$, from equation 5.3 (vii)

For 229 H model, the value of K_1 is 3.50 while for Terex is 3.18. So, the productivity at each given haul distance point is more for 229 H as compared to Terex TS 14B. The figures 5.1 and 5.2. show this trend of productivity.

Towed scrapers include 463G and 435G using tractor D8H (DD). The heaped capacity and the horse power of 463G-D8H combination are 20 m^3 and 270 horse power respectively. For 435G-D8H heaped capacity and horse power are 13 m^3 and 270 horse power respectively. The curves shown in Fig. 5.3 show the same trend of decreasing productivity with increasing haul distance and the productivity is higher for 463G – D8H at each haul point distance. The explanation for the same can be drawn from the analytical treatment given below.

The resistance encountered by the towed scraper is given by

$$TR = GR \times (Gc + Gpf) + Gpf \times RR \qquad 5.3(viii)$$

where,

Gpf = load being pulled by the tractor,

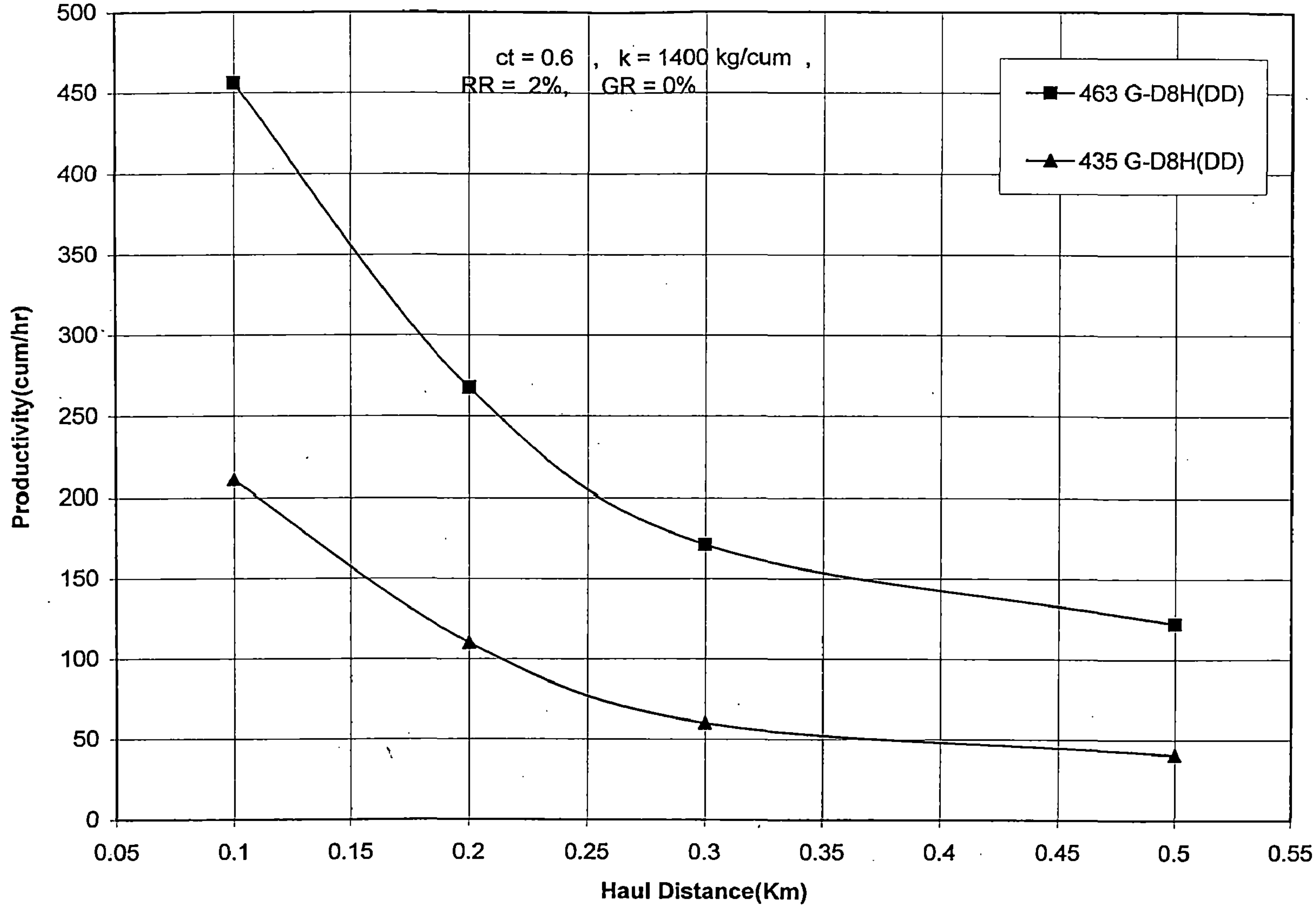


Fig 5.3 Variation of Productivity with haul distance for Towed Scrapers

= L + Ge (while loaded)

= Ge (while empty)

Gc = gross weight of the crawler mounted tractor,

RR = rolling resistance

GR = grade resistance

So,

TR = GR x Gc + Gpf x (GR + RR)

i.e. for onward journey (while loaded), the resistance encountered can be expressed as,

$$TR_f = GR_f \times Gc + (L + Ge) \times (GR_f + RR_f) \qquad 5.3(ix)$$

where, subscript 'f' refers to the forward (onward) journey

Let, the net draw bar pull required to overcome this resistance be NDB, then

TR = k_1 x NDB

where,

k_1 is the factor to scale down NDB depending upon the condition of tractor.

or, TR $\propto$ NDB,

since, NDB $\propto$ DB,

Therefore, TR $\propto$ DB 5.3(x)

Also, DB $\propto$ HP/v 5.3(xi)

Therefore from equation 5.3(ix), 5.3(x) & 5.3(xi).

$$v_f \propto \frac{HP}{GR_f x Gc + (L + Ge) x (GR_f + RR_f)} \qquad 5.3(xii)$$

The productivity of towed scraper can be expressed as,

$$P \propto Hc \times \frac{60}{Cycle\ time}$$

For given fixed time values, the productivity can be written as

$$P \propto \frac{Hc}{d\left[\frac{1}{v_f}+\frac{1}{v_b}\right]}, \qquad 5.3(xiii)$$

where,

v_f and v_b are the velocities of onward journey and return journey respectively.
Finally from equations 5.3(xii) and 5.3(xiii).

$$P \propto \frac{Hc}{d}\left[\frac{HP}{\{GR_f.Gc+(L+Ge)(GR_f+RR_f)\}+\{GR_b.Gc+(Ge)(GR_b+RR_b)\}}\right] \qquad 5.3(xiv)$$

where, subscripts 'f' and 'b' refer to onward journey (while loaded) and return journey respectively. The productivity can then be expressed in short as,

$$P \propto \frac{K_2}{d}$$

where,

$$K_2 = Hc\left[\frac{HP}{\{GR_f.Gc+(L+Ge)(GR_f+RR_f)\}+\{GR_b.Gc+(Ge)(GR_b+RR_b)\}}\right]$$

The values of factor K_2 for the given conditions for scrapers 463G-D8H and 435G-D8H are 5.304 and 4.72 respectively. So, the productivity at each haul point distance is more for 463G-D8H than 435G-D8H.

To study the effect of haul distance on the economic loading time (or optimum loading time) load growth curves for scrapers 229 H and Terex TS14B were assumed as shown in figures 5.4 & 5.6 and the aforesaid effect is presented in Fig. 5.5, & 5.7. The curves indicate that, the economic loading time increases with the increase in haul distance. It is expected also, since with the increase in haul distance, the total travel time increases and the tangent line to the load growth curve (used to obtain the optimum loading time) becomes flatter, (i.e. slope decreases), touches the load growth curve at

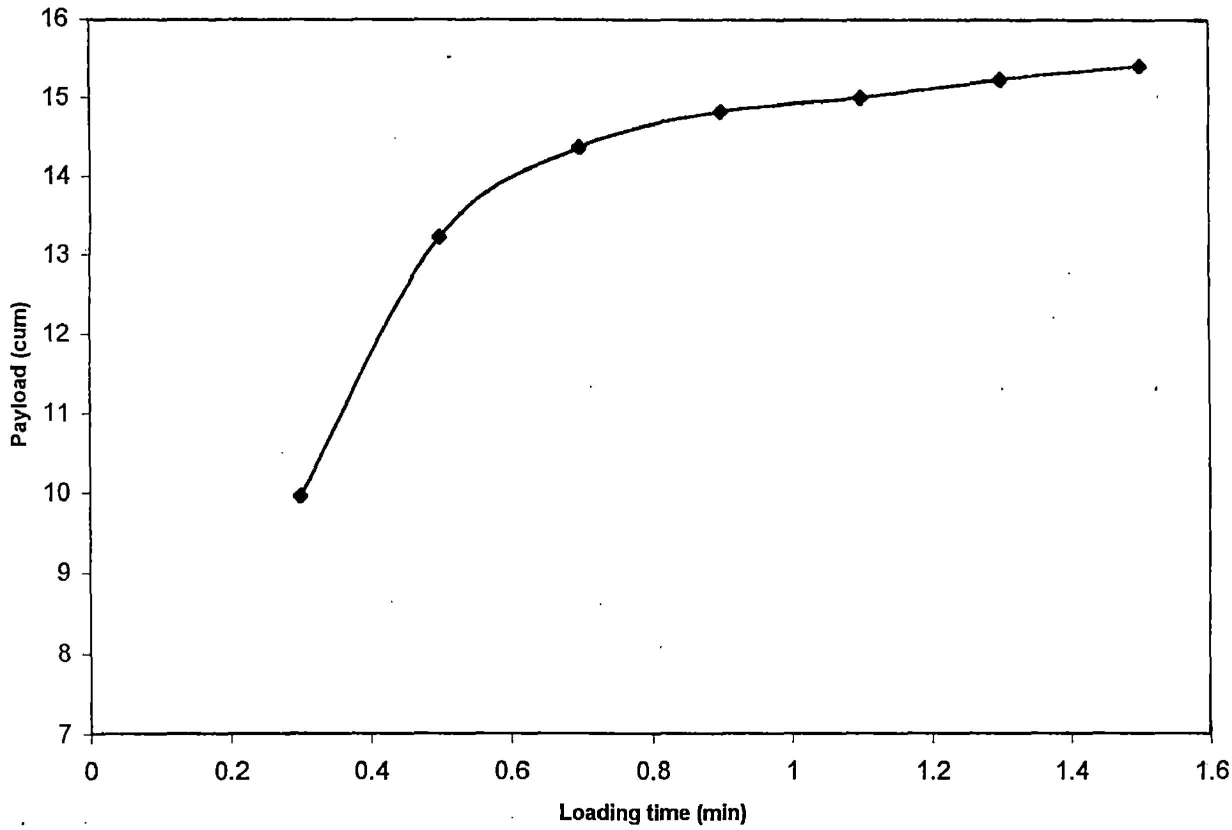


Fig. 5.4:Load growth curve of Scraper 229 H for Payload of 15.5 m3

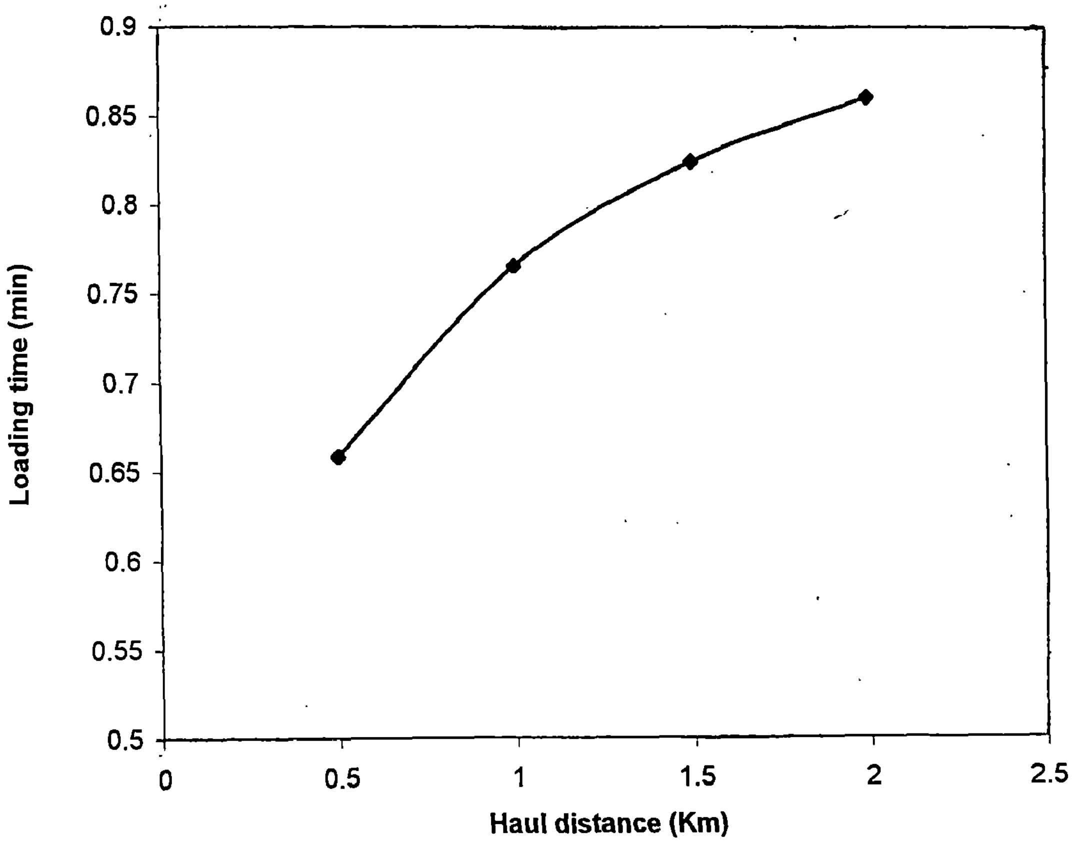


Fig. 5.5 Economic Loading Time V/s Haul distance for Scraper 229H

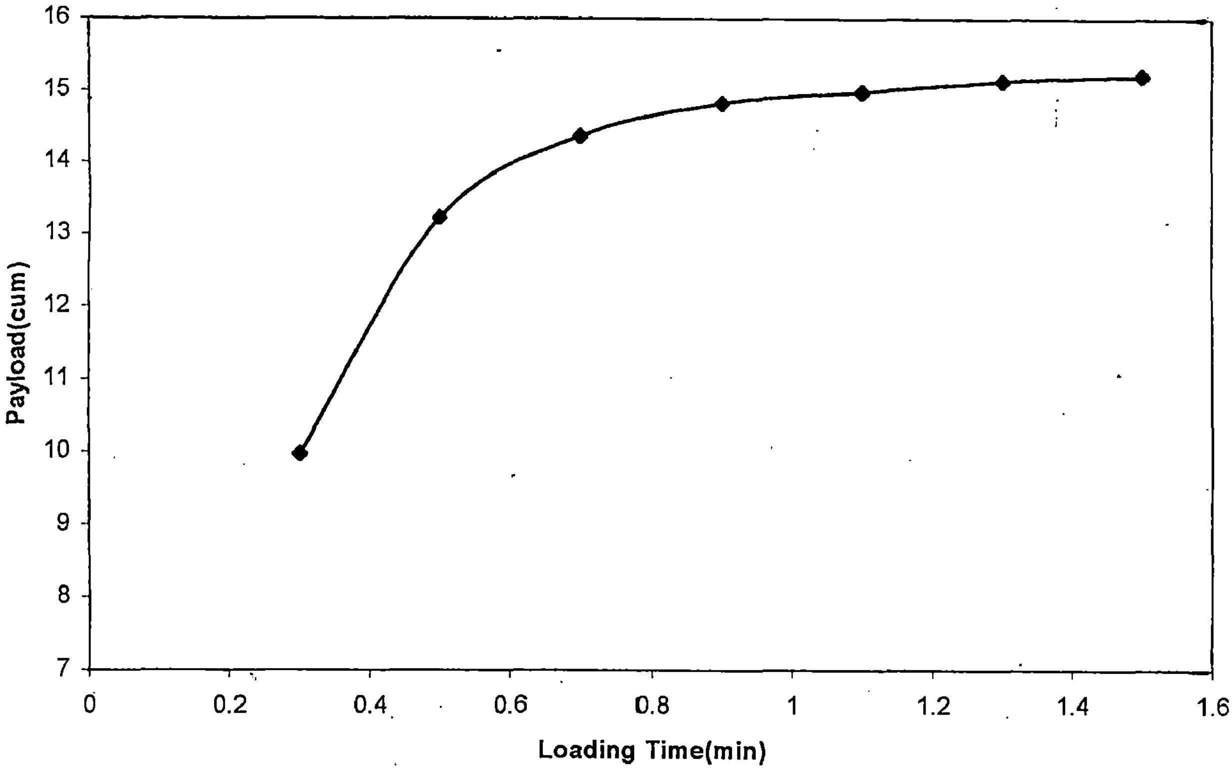


Fig.5.6 Load growth curve of Scraper TEREX TS 14B for payload of 15.3 m3

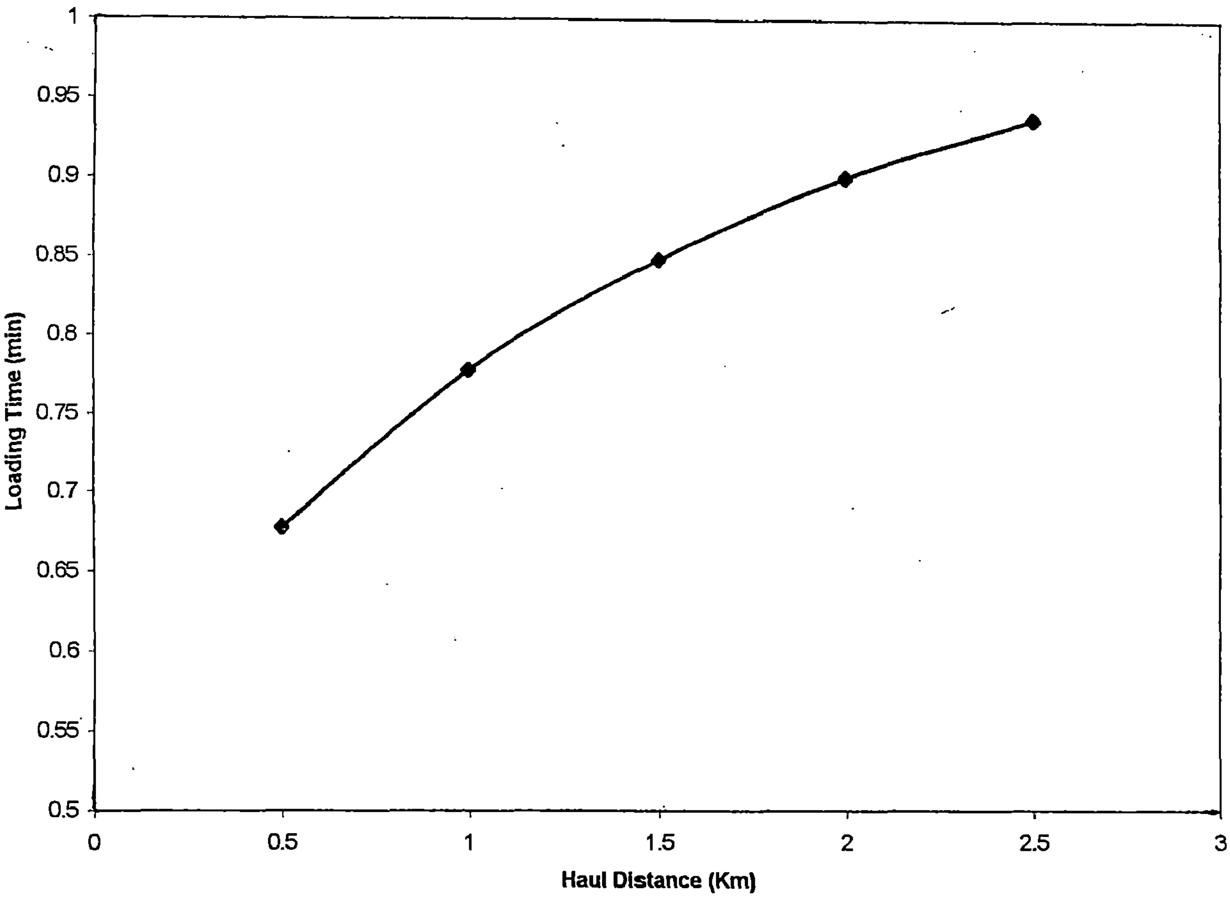


Fig. 5.7 :Economic Loading Time V/s Haul distance for Scraper TEREX TS 14B

some more distant point from the origin, thereby giving higher value of optimum loading time.

5.4 BULLDOZERS

Curves drawn for the bulldozers D31 A-17, D50A-15,D80A-12 and D155 A-1 in figure 5.8 indicate that the productivity (m^3/hr) decreases with increasing haul distance for all models. Also for any haul distance point, the productivity of the model of higher HP is more. It can be explained as follows:

When the track mounted bulldozer is moving the excavated earth , the total resistance, TR encountered by the bulldozer is

$$TR = Gc \times GR + k \times Bc \times f,$$

$$TR = k \times Bc \times f \qquad \text{for } GR = 0 \qquad 5.4\text{ (i)}$$

where,

Gc = Operating weight of bulldozer in tonnes,

GR = Grade resistance in kg,

k = Density of the earth in kg/m^3,

Bc = Blade capacity of the bulldozer,

f = Existing co-efficient of friction,

Let, Net draw bar pull required to overcome this total resistance be NDB,

i.e. $TR = k_1 \times NDB$

where k_1 is a factor to scale down NDB depending upon the condition of the bulldozer.

or, $TR \propto NDB$,

since, $NDB \propto DB$

or, $TR \propto DB$ 5.4 (ii)

from eq. 5.4 (i) and 5.4 (ii),

$$DB \propto k \times Bc \times f \qquad 5.4\text{ (iii)}$$

Also ,

$$DB \propto \frac{HP}{v} \quad \text{, for a track mounted equipment.} \qquad 5.4\text{ (iv)}$$

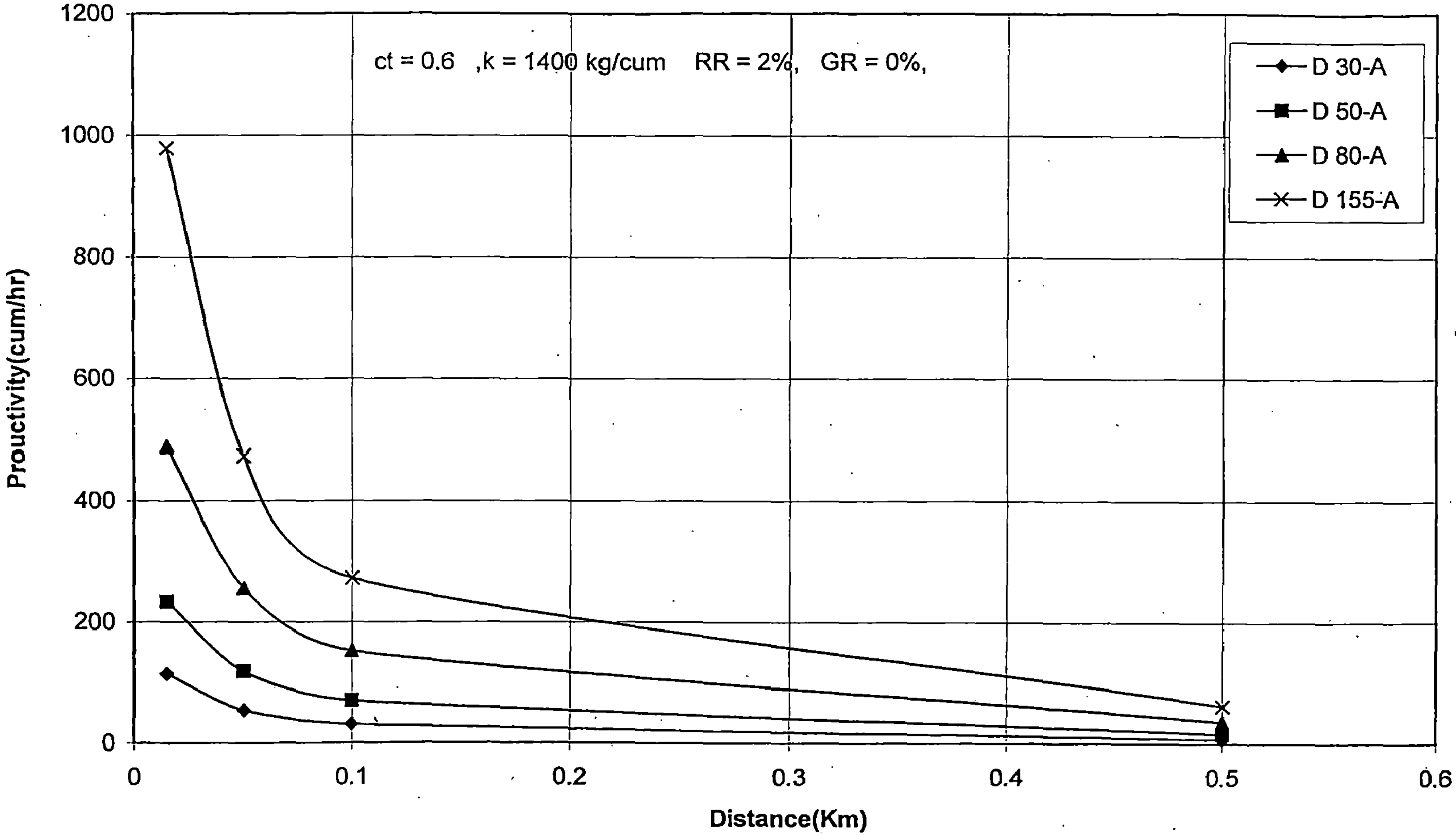


Fig 5.8 Variation of productivity with haul distance for track mounted bulldozres

So, from eq. 5.4 (iii) and 5.4 (iv),

$$k \times Bc \times f \propto \frac{HP}{v}$$

or, $$v \propto \frac{HP}{k \times Bc \times f} \qquad 5.4 \text{ (v)}$$

The productivity of bulldozer can be expressed as,

$$P \propto Bc \times TPs \qquad 5.4 \text{ (vi)}$$

where $TPs = \frac{60}{cycle\ time}$

For given fixed cycle times

$$TPs \propto \frac{v}{d}, \qquad 5.4 \text{ (vii)}$$

where d is the haul distance,

From equations 5.4 (vi) and 5.4 (vii),

$$P \propto \frac{Bc \times v}{d} \qquad 5.4 \text{ (viii)}$$

From eq. 5.4 (v) and 5.4 (viii)

$$P \propto \frac{Bc \times HP}{d \times k \times Bc \times f}.$$

or

$$P \propto \frac{HP}{d}$$ for given k and f

Hence,

(i) for a given model i.e. given HP

Productivity, $P \propto \frac{1}{d}$

i.e. as haul distance increases, productivity decreases.

(ii) For a given distance of haul,

Productivity, $P \propto HP$.

So, the higher horse power (HP) model have higher productivity at each haul distance point.

5.5 HAUL UNITS

The haul units are loaded by loaders/excavators by using thumb rule that the capacity of haul units is 4-5 times the capacity of loader. Productivity shows the same decreasing trend with increasing haul distance as in the case of bulldozers and scrapers. HP/G ratio of RD-25 (HP = 292) and RD-35 (HP = 355) is 6.39, 6.61, while HP/E ratio is 12.696, 16.21, and heaped capacities are 17.00 m^3, and 22.47 m^3 respectively. The productivity of loader is taken at 100% loader efficiency because a loader typically operates at or near 100% efficiency when actually engaged in loading [Nunnally 1977]. Figure 5.9 and 5.10 show the trend of productivity of haul units for different total resistances.

For RD-25, the value of K_1 of equation 5.3 (vii) is 3.6, while for RD-35, the value of K_1 is 5.25. Hence, utilizing the same analytical approach as discussed for scrapers, it can be concluded that :

(i) Productivity decreases with increasing haul distance.

(ii) Productivity is higher for higher HP model than the lower HP model for any given haul point distance.

5.6 SELECTION OF EQUIPMENTS WITH REFFERENCE TO HAUL DISTANCE

On the question of selecting the equipment for moving the earth for different haul distances,

(a) Figure 5.8 shows that,

Bulldozer of lower HP are useful for very short haul distance i.e upto 50 m only while bulldozers of higher HP may be selected for haul distances upto 100m.

(b) Figure 5.11 and 5.12 reveal that,

Motorized scrapers 229H & Terex TS 14B may be selected for the operating range from approximately 100 m to 1000 m, beyond which the haul units have an edge in terms of productivity. Although towed scrapers have lower productivity as compared to the motorised scrapers in this operating range, yet these may be used if need arises for the requirements of better traction.

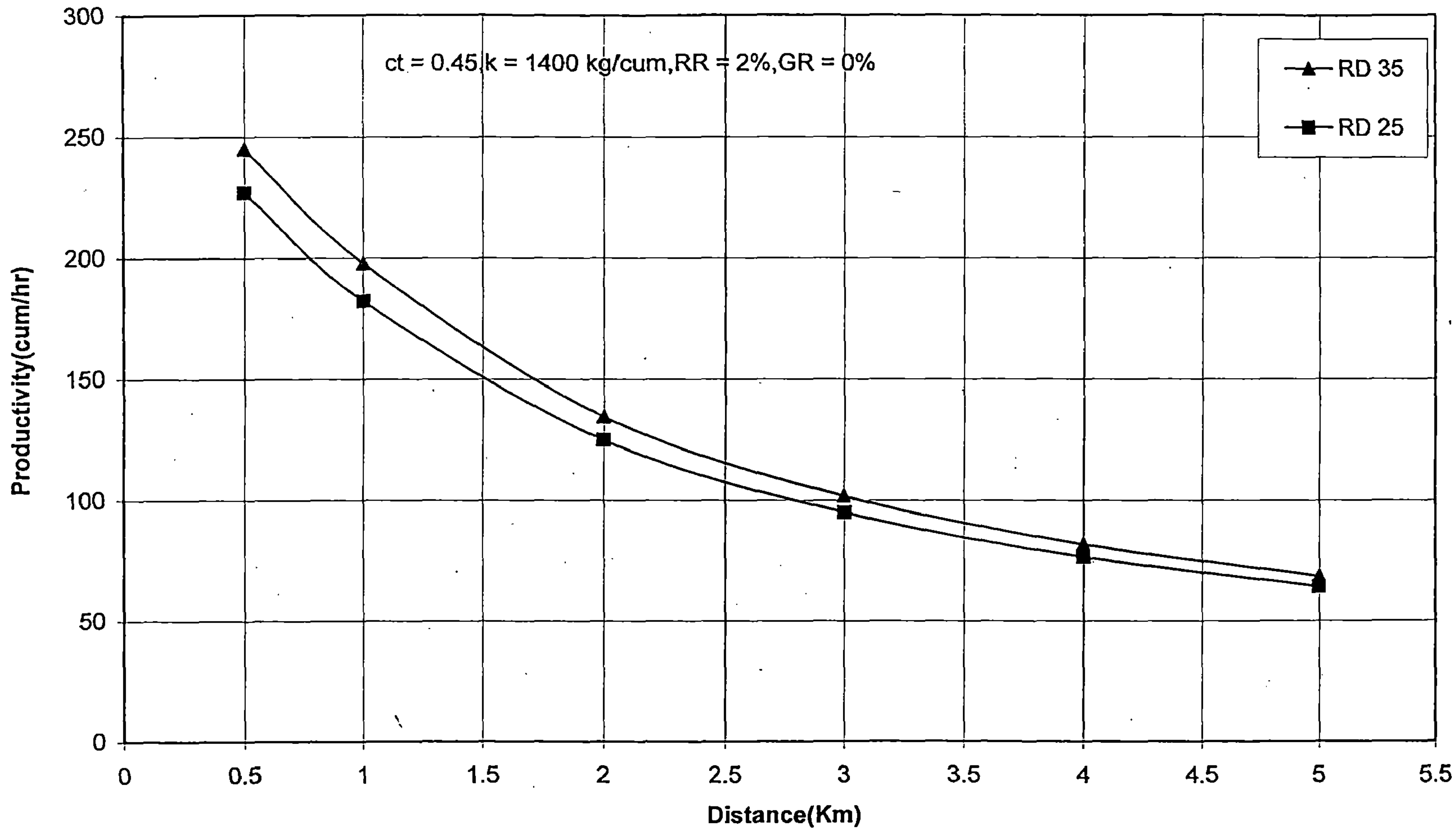


Fig 5.9 Comparison of productivities of dumpers for different haul distances

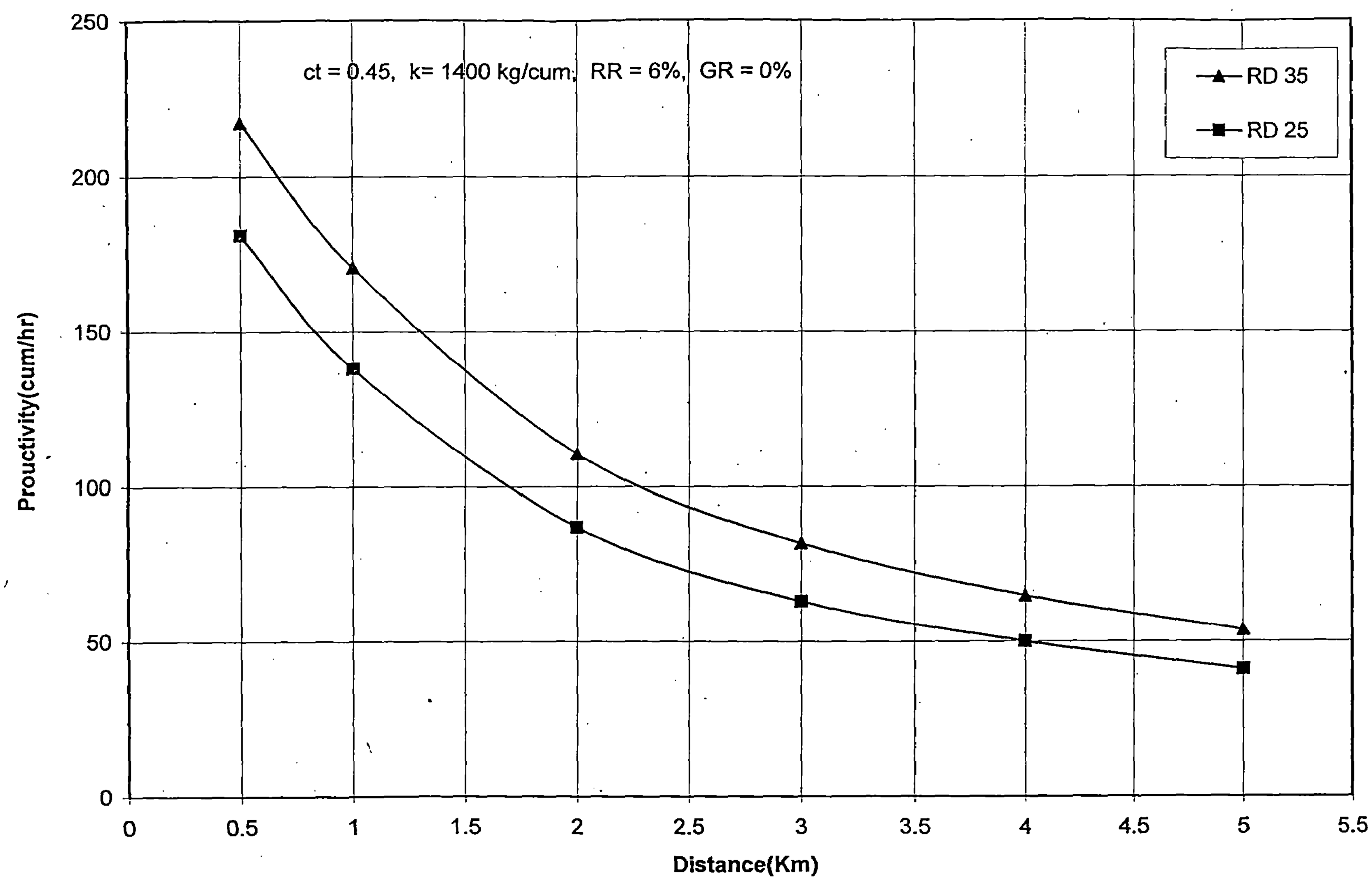


Fig 5.10 Comparison of productivities of dumpers for different haul distances

With increase in total resistance, this economic haul distance for motorised scrapers as compared to haul units decreases as shown in Fig. 5.11 and Fig. 5.12.

(a) Haul units have productivities higher than the scrapers beyond 1000 m and it makes them useful for the operating range from 1000 m onwards to higher values of haul distances.

5.7 PRODUCTIVITY AND GRADE

To find the effect of grade on the productivity the rolling resistance and haul length should be kept constant.

The effect of grade resistance on the productivity is depicted through curves drawn for RD-25 as shown in Fig. 5.13. The curves drawn indicate that the productivity decreases with increase in the grade resistance. The analytical explanation can be found from equation 5.3 (vi).

5.8 PRODUCTIVITY AND ROLLING RESISTANCE

To study the effect of RR on productivity, grade and haul distance should be kept constant. The curves shown in Fig. 5.14 indicate that for a particular haul distance and grade, the productivity decreases with increase in haul distance. Analytical explanation for this can also be found from equation 5.3 (vi).

5.9 PERFORMANCE OF MODELS

The optimization model for motorised scraper was used to obtain the values of productivities for different haul distances and total resistances by considering the Caterpillar 631-C model of scraper. The results obtained from the developed model were compared with the observed values of Cat 631-C model (after Caterpillar Performance Hand book, 2nd edition, 1971). These are depicted in Figure 5.15 and 5.16. It indicates that the productivities obtained from the present model are found to be in good agreement with the observed values for low resistances, while for higher resistances, the productivities obtained from the existing model are slightly lesser than the observed

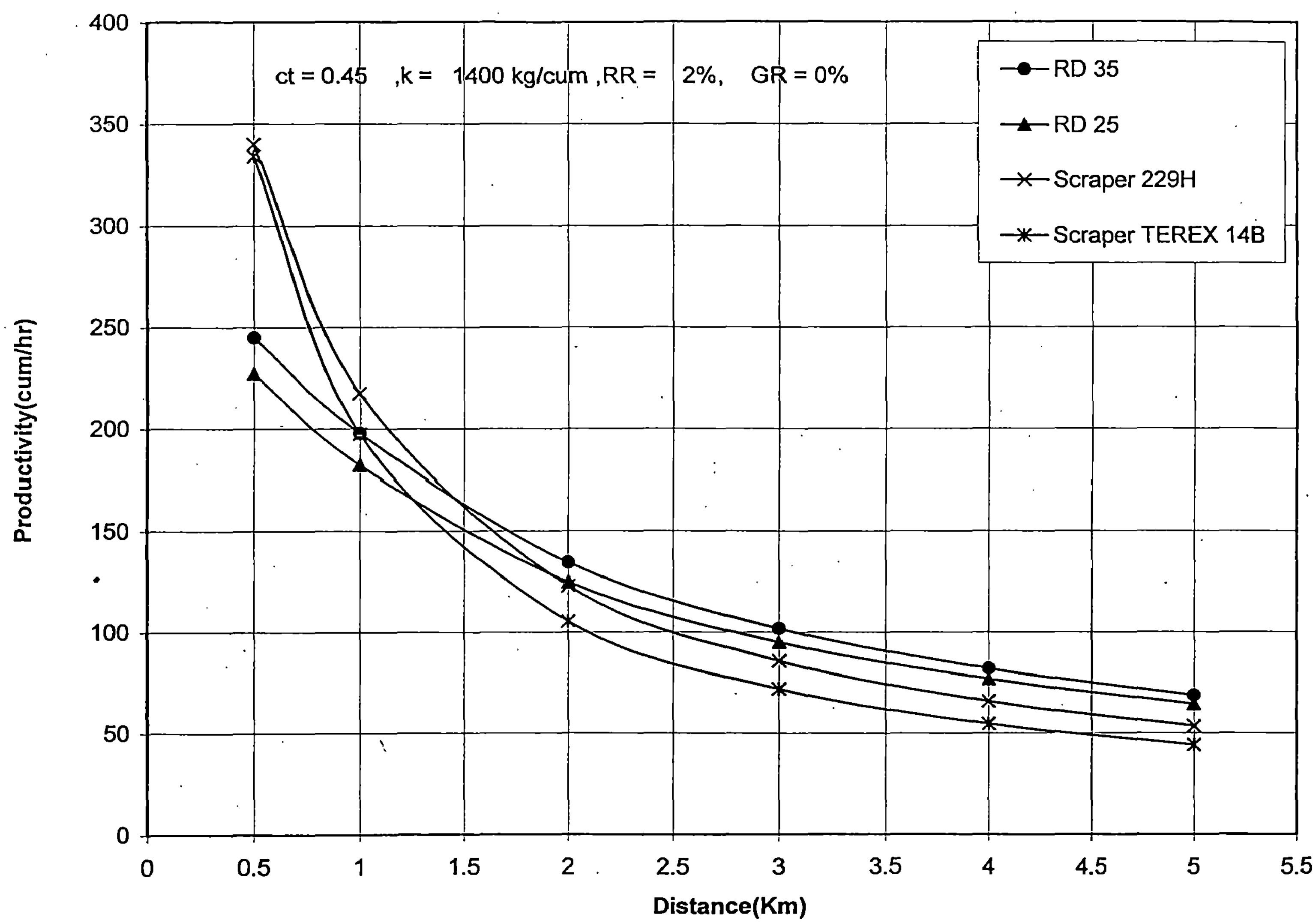


Fig 5.11 Productivity variation with haul distance for haulers and motorised scrapres

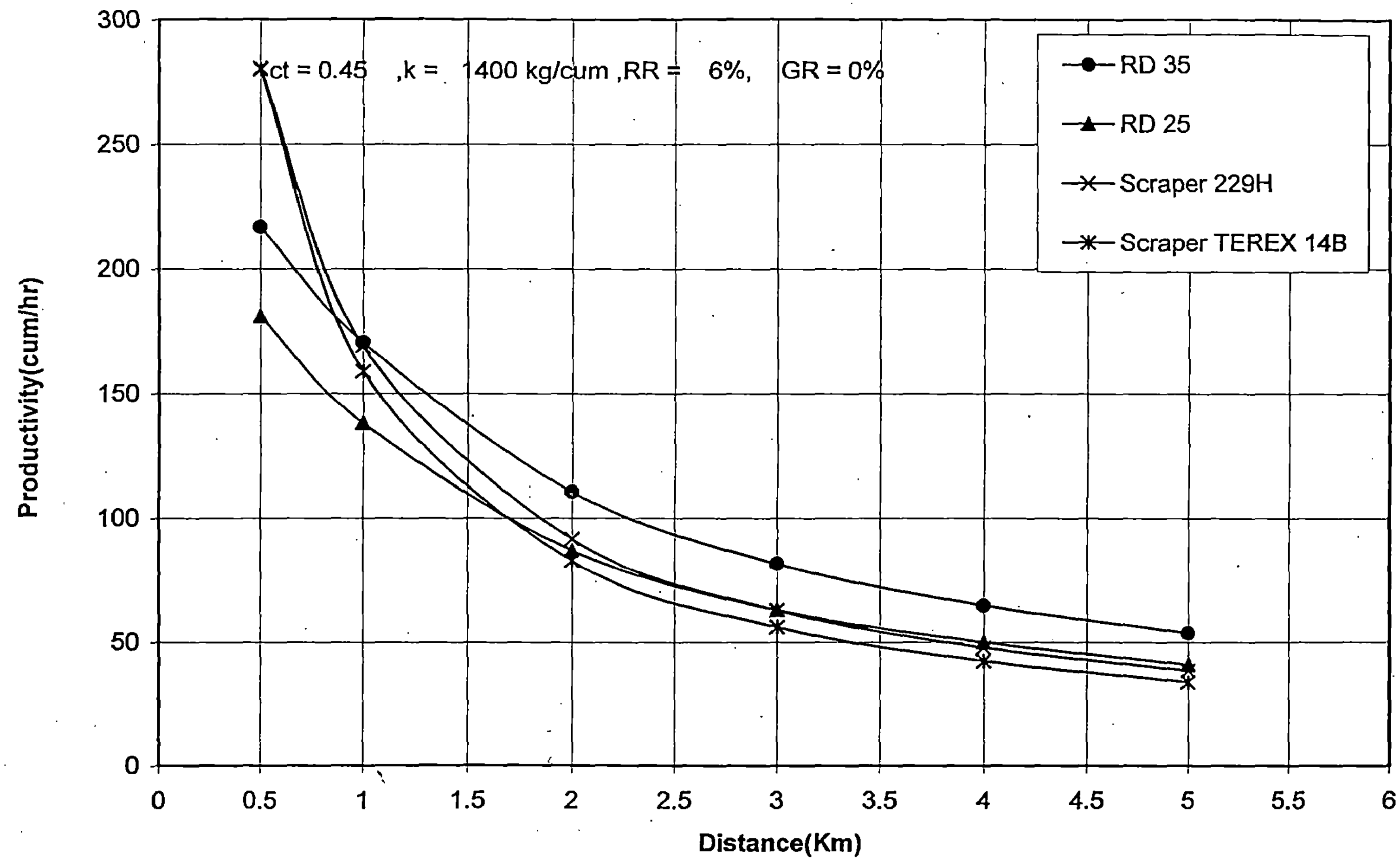


Fig 5.12 Productivity variation with haul distance for haulers and motorised scrapres

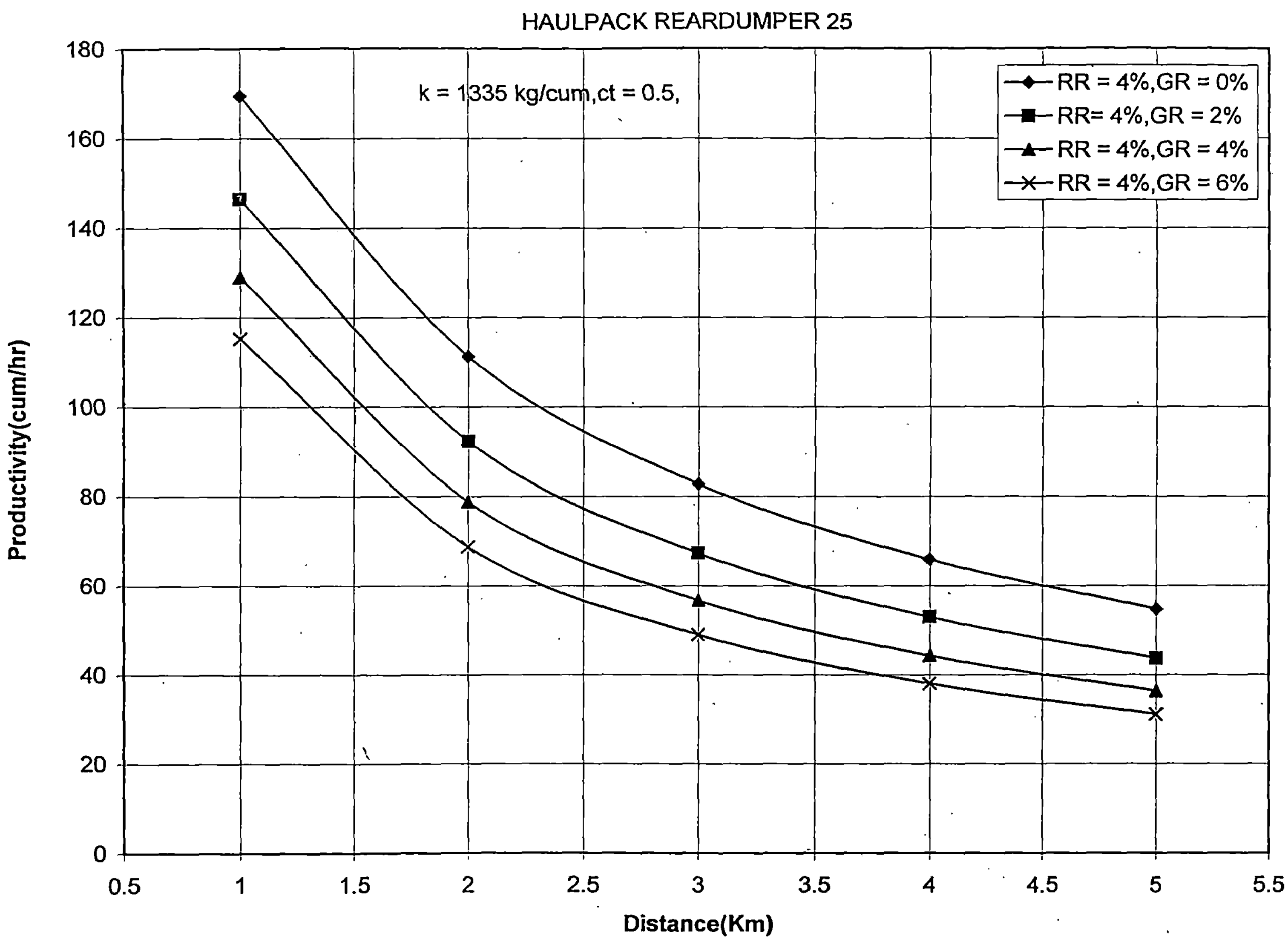


Fig 5.13 Effect of Grade Resistance on productivity for different haul distances

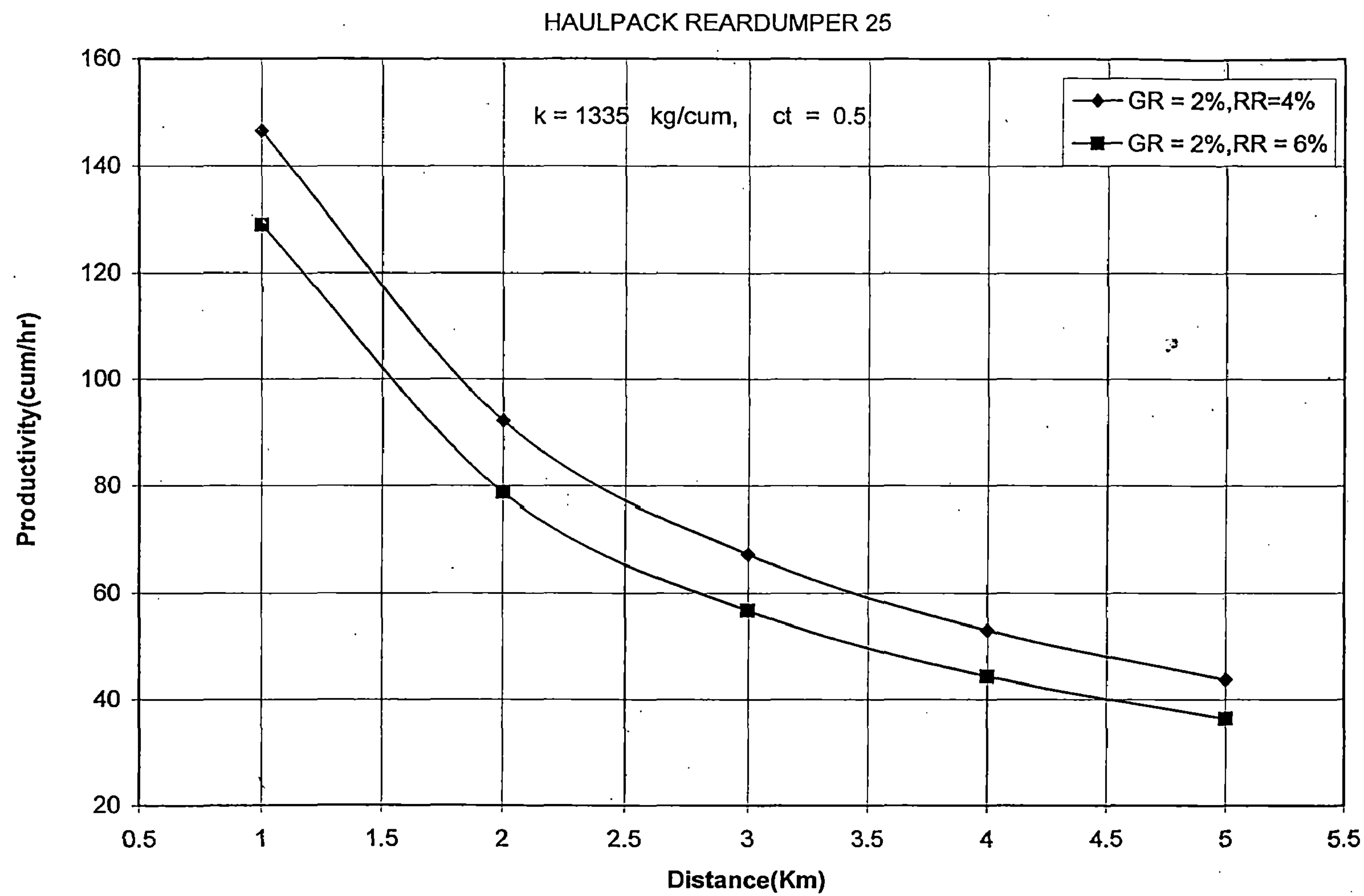


Fig 5.14 Effect of Rolling Resistance on productivity for different haul distances

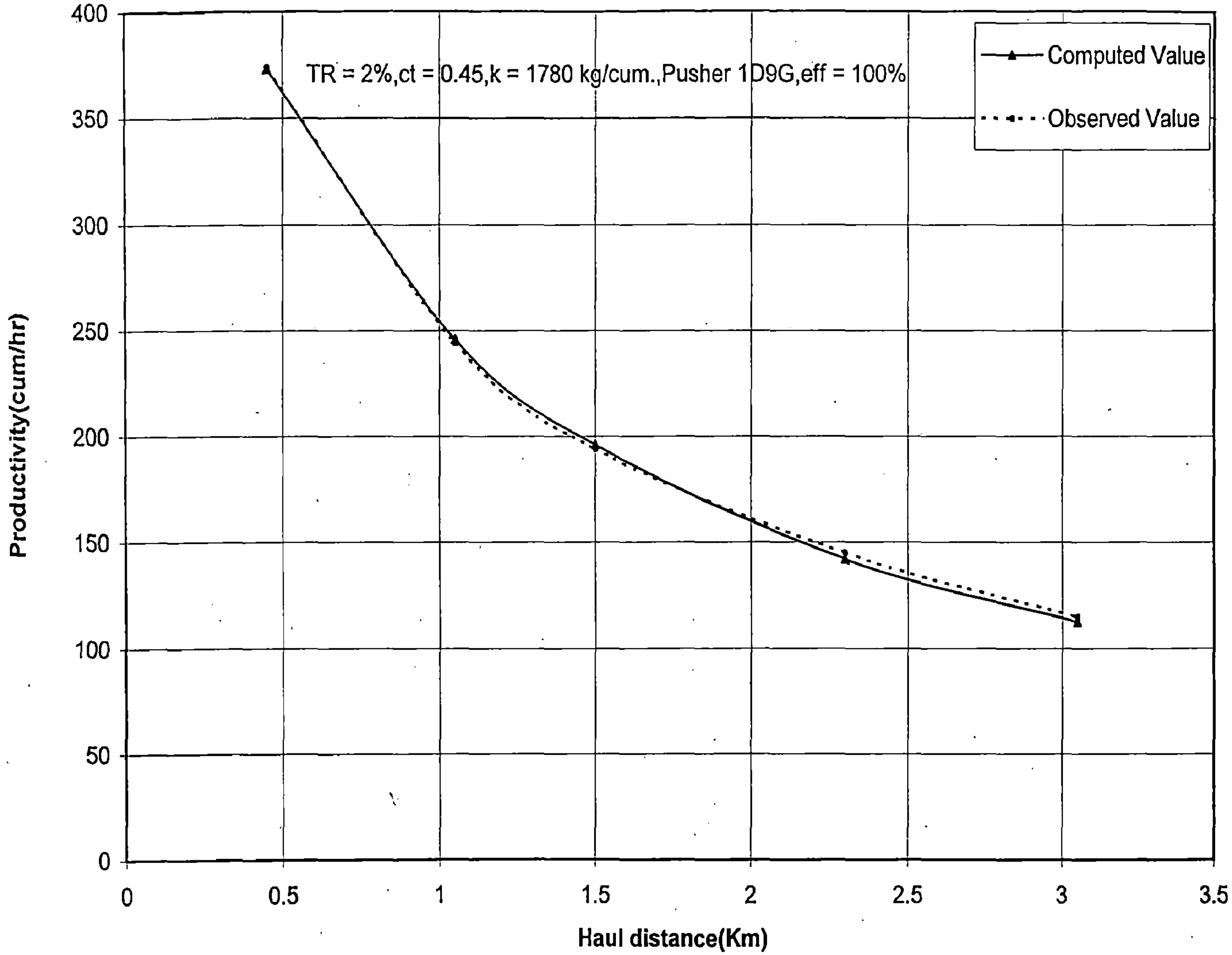


Fig.5.15 Comparison of productivities of observed values and computed values(using Model) for Scraper 631C

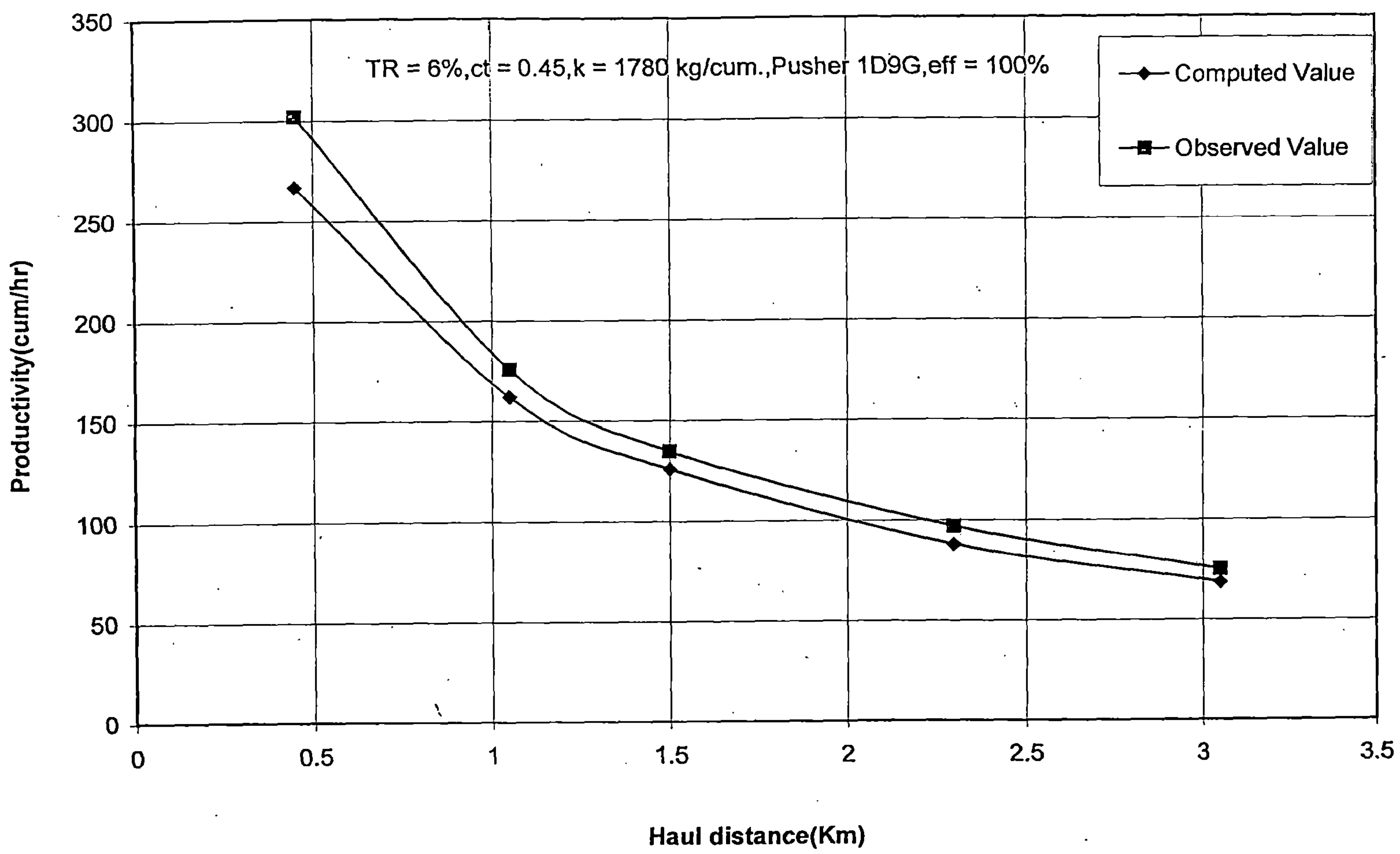


Fig.5.16 Comparison of productivities of observed values and computed values(using Model) for Scraper 631C

values. It may be attributed to the effect of the speed factor considered for each section of haul. Since in the analysis, an average value of speed factor was considered for converting the velocity to average velocity of scraper movement, it may be suggested that use of an appropriate value of speed factor preferably on the higher side of each speed factor range, may provide still better results.

Secondly, the model does not make use of retarder curves. It computes velocity for the vehicle descending a grade by the functional relationship of velocity with the HP, RR, GR, G and the efficiency i.e. equation 5.3 (vi). For total resistance being negative, the model always considers highest operating velocity for the vehicle. It over-estimates the velocity for total effective grade higher than 5% (while loaded) and for total effective grade higher than 10% (while empty), thereby finally giving lesser cycle time and consequently higher production per hour. It is an inherent limitation of the models for motorised scrapers and haul units.

In the absence of observed field values, the models of haul units and bulldozer could not be compared. However, the motorised scraper model and haul unit model resemble and it may be safely concluded that the haul unit model should also give better results as obtained by scraper model for Cat 631-C.

5.10 EQUIPRODUCTION CURVES

The equiproduction curves are plotted for haul pack RD 25 in figures 5.17, 5.18 & 5.19. An equiproduction curve is the curve joining the same values of productivity for given other parameters like RR, GR and haul distance. These curves are plotted – (i) by keeping RR constant and varying the values of GR and haul distance (ii) by keeping the distance constant and varying RR and GR.

The equiproduction curves show a peak production at a certain grade i.e optimum production is achieved at a particular grade for all road length at certain rolling resistance. This grade of peak production depends on rolling resistance (RR). For low RR, it is nearer to 0% i.e. level road and for higher values of RR, it shifts towards

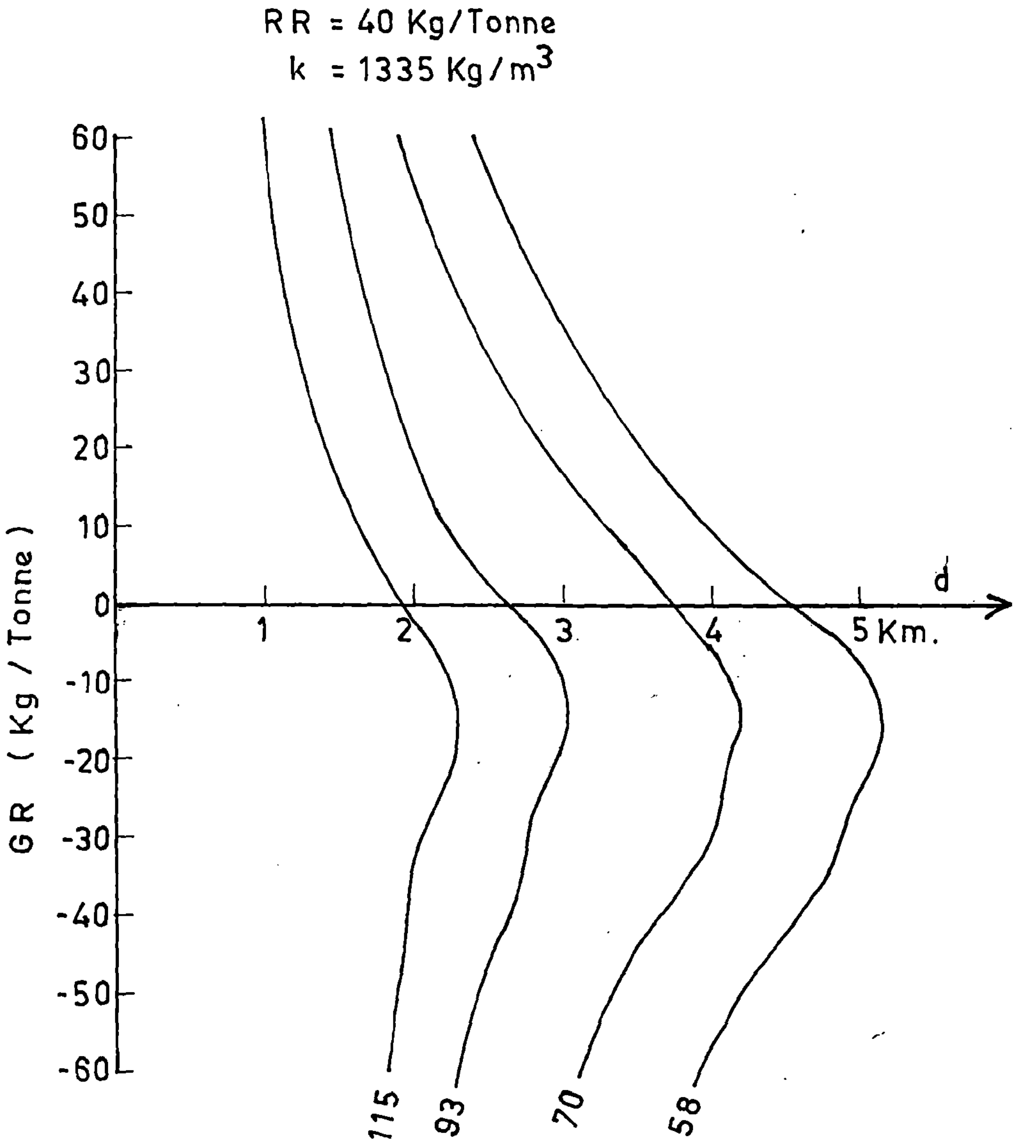


FIG.5·17- EQUIPRODUCTION CURVES FOR HAULPACK REAR DUMPER -25

RR = 60 Kg / Tonne
k = 1335 Kg / m^3

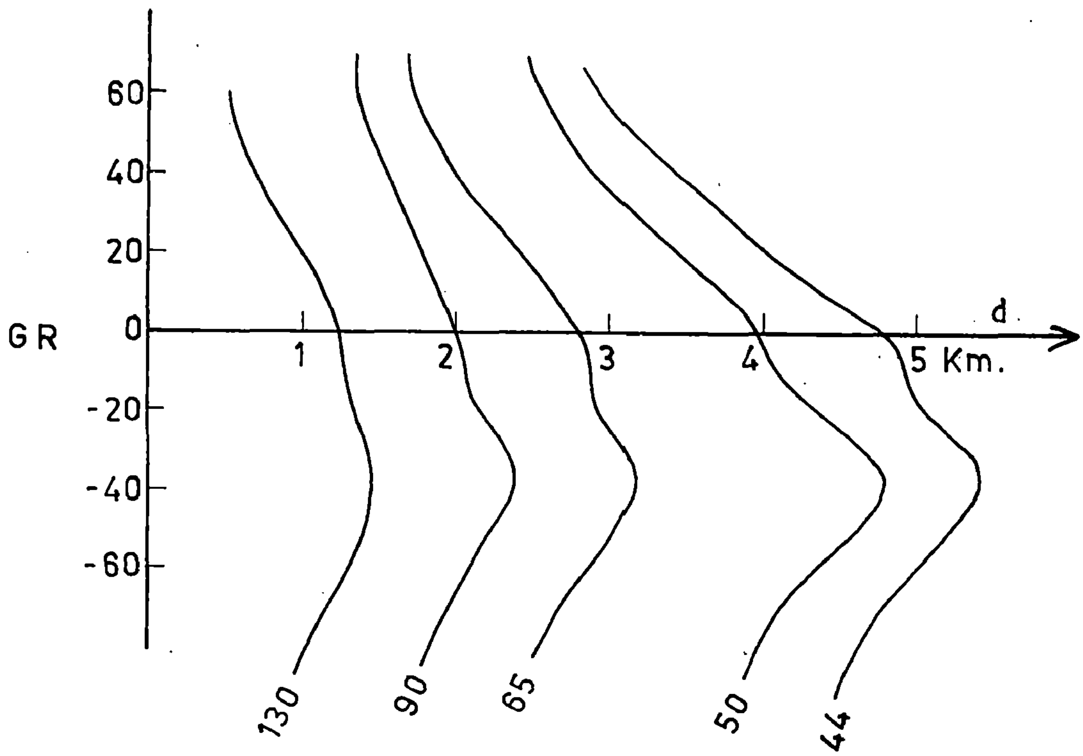


FIG.5·18-EQUIPRODUCTION CURVES FOR HAULPACK REAR DUMPER -25

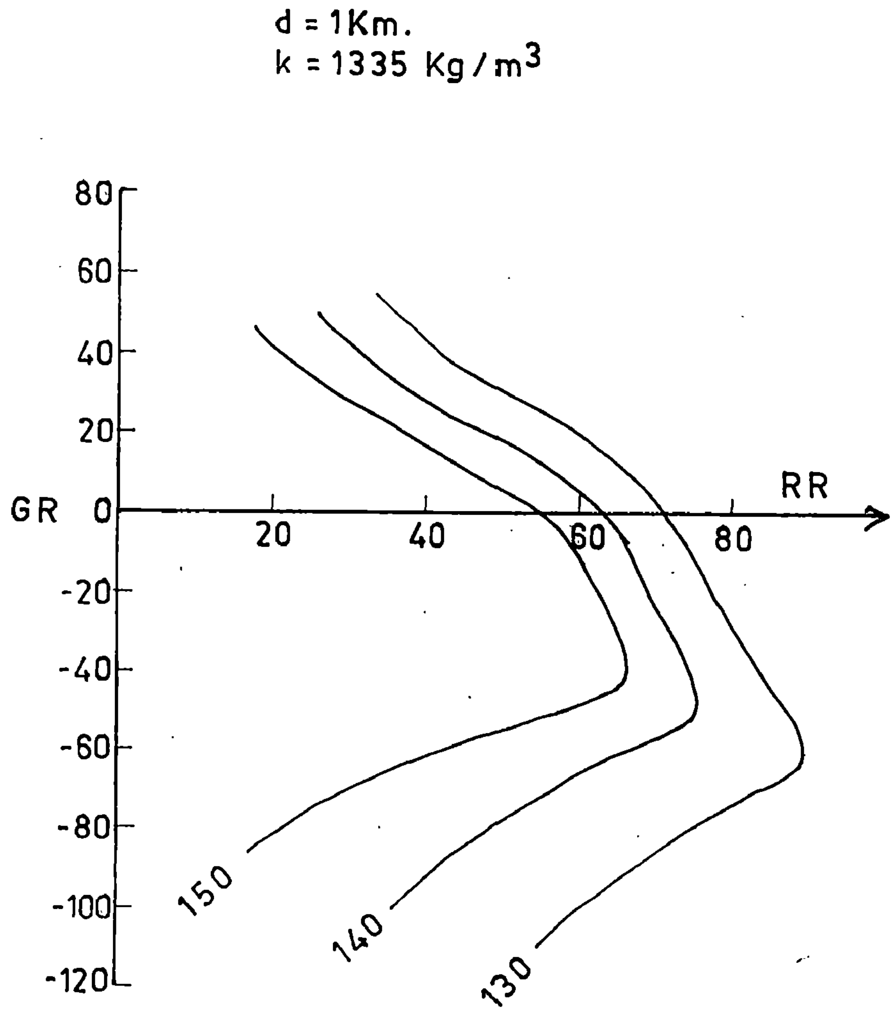


FIG. 5·19 - EQUIPRODUCTION CURVES FOR HAULPACK REAR DUMPER -25 negative grades. It is due the fact that while going down grade, grade resistance becomes grade assistance and the total resistance decreases. In some cases total resistance may become negative also, thereby needing the vehicles to operate through hydraulic retarder. The hydraulic retarder is used on the vehicle to absorb H.P. through the transmission cooling system.

In short, it can be observed from these figures that

(i) any change in the haul distance and grade results in decrease or increase in production.

(ii) With the increase in haul distance, the equiproduction lines get curved at certain grade. This grade, called `optimum grade' actually gives the optimum production of the haul unit for the given parameter values. This optimum grade is same for all haul distances at certain rolling resistance.

(iii) The optimum production of the haul unit for a given haul distance will be shifted towards higher negative grade each time the rolling resistance is increased.

(iv) The optimum grades are shifted towards higher negative grades with the increase in rolling resistance.

CHAPTER - 6
CONCLUSIONS

6.1 GENERAL

In the present study, optimization models have been developed for earthmoving operations involving motorized scrapers, towed scrapers, track mounted bulldozers and haulers. These models aim to determine the optimum size of a particular equipment and its number/numbers for completing a given target of earthwork. In order to analyze the above requirements, calculation of the productivity of the equipment/equipments under various situations involving different operating parameters becomes essential. And it is the productivity which forms the real basis for analyzing these models, as the derivation of cost index number is itself based on the productivity of the equipment/equipments alongwith their number and the hourly owning and operating costs.

The models developed for scrapers, bulldozer and haulers were used to obtain the productivity of – scraper 229H, Terex TS14B, scrapers 463G-D8H(DD), 435G-D8H(DD) bulldozers – D31A, D50A, D80A, D155A and haulers RD25, RD35 under different operating parameter values. From the analysis performed, the broad conclusions can be summarized as under :

6.2 SCRAPERS

(i) With the increase in haul distance, the productivity decreases for given values of total resistance (positive).

(ii) The higher horse power range model has higher productivity at each haul distance point for given values of total resistance in the case of motorised scrapers. While for towed scrapers, the higher heaped capacity model has higher productivity for same horse power.

(iii) Also, with the increase in total resistance (positive), the productivity decreases with increasing haul distance, but the gap of difference of productivities becomes narrower with haul distance becoming more. It has been analyzed for motorised scraper models.

(iv) With the increase in haul distance, the optimum or economic loading time increases.

6.3 BULLDOZERS

(i) With the decrease in dozing distance, the productivity increases for given values of total resistance.

(ii) Productivity of higher horse power range models is larger at each haul distance point for given value of total resistance.

6.4 HAULERS

(i) Productivity of haulers decreases with increase in hauling distance for given values of total resistance.

(ii) Lower horse power range models have lower productivity at each haul distance point for given value of total resistance.

(iii) The equiproduction curves constructed for RD-25 indicate that :

(a) with the increase in haul distance, the equiproduction lines get curved at certain grades, this grade is the 'optimum grade' which gives the optimum production of the haul unit for the given parameter values. This optimum grade is same for all haul distances at certain rolling resistance.

(b) the optimum production of the haul unit for a given haul distance will be shifted towards higher negative grade each time the rolling resistance is increased.

(c) the optimum grades are shifted towards higher negative grades with the increase in rolling resistance.

6.5 ECONOMIC HAUL DISTANCE

(i) Bulldozers (track mounted) of lower horse power range (~ 60 to 100 hp) may be used upto 50 m while models for higher HP range may be utilized upto nearly 100 m for low resistance values.

(ii) Figure 5.11 and 5.12 reveal that, motorized scrapers 229H & Terex TS 14B may be selected for the operating range from approximately 100 m to 1000 m, beyond which the haul units have an edge in terms of productivity. Although towed scrapers have lower productivity as compared to the motorised scrapers in this operating range, yet these may be used if need arises for the requirements of better traction.

(iii) With increase in total resistance, this economic haul distance for motorised scrapers as compared to haul units decreases as shown in Fig. 5.11 and Fig. 5.12.

(iv) Haul units have productivities higher than the scrapers beyond 1000 m and it makes them useful for the operating range from 1000 m onwards to higher values of haul distances.

6.6 SCOPE FOR FURTHER STUDY

(i) The optimization model for Motorized scraper has been developed for three sections of haul and three sections of return only. The optimum load time is based on maximizing scraper production and it is assumed that an adequate number of pushers are available to insure that scrapers do not have to wait for pushers. A more detailed study with more than 3 sections of haul road & considering the constraint of pusher could be done as an extension of the present work.

(ii) The optimization model for Towed scraper has been developed for two sections of haul and two sections of return only. This model could also be extended for more than 2 sections of haul road.

(iii) The optimization model for Bulldozer is utilisable for two sections of dozing and two sections of return. This model can also be extended for more than 2 sections.

(iv) The optimization model for Haul units (Deterministic model) can be used for five sections of haul and five sections of return. Besides increasing the number of haul road sections, detailed cycle time calculations of loader could be included in the model as against the time to load one bucket (i. e. cycle time of loaders) taken directly from tables supplied by the manufacturers corresponding to different loader sizes.

(v) The probabilistic model of haul units can be used for five sections of haul and five sections of return. Although the factorial function returns incorrect values for more than 7, yet the scope of the program has been extended upto factorial value of 11 and hence the factorial function in the program ceases to operate for more than 11 number of haul units per loader.

In general, it can be stated that although the above models are developed for the number of sections mentioned against each, yet these can be extended to `n' number of sections of haul and return without any difficulty. The basic modification needed is to add sections, modify the velocity for that section and use the haul & return distances of that particular section.

REFERENCES

1. Alkass, Sabah and Harris Frank (1988), "Expert system for earthmoving equipment selection in road construction", J. Constr. Engrg. and Mgmt., ASCE, 114(3), 426-439.
2. Amirkhanian, Serji N. and Baker, Nancy J (1992), "Expert system for equipment selection for earthmoving operations", J. Constr. Engrg. And Mgmt., ASCE, 118(2), 318-331.
3. Caterpillar Performance Handbook (1972), 2nd Ed.
4. Chauhan Gopal (1977), "Recent trends in design of construction equipment" M.E. Dissertation, WRDTC, University of Roorkee, Roorkee.
5. Farid, Foad and Koning, Thomas L. (1994), "Simulation varifies queuing program for selecting loader truck fleets", J. Constr. Engrg. and Mgmt., ASCE, 120(2), 386-404.
6. Gransberg Douglas, D. (1996), "Optimizing haul unit size and number based on loading facility characteristics", J. Constr. Engrg. and Mgmt., ASCE, 122(3), 248-253.
7. Hag, Abdalla Elamin Abbas (1986), "The effect of design parameters of earthmoving machinery in performance and production", M.E. Dissertation, WRDTC, University of Roorkee, Roorkee.
8. Haidar A., Naoum, S., Howes, R., and Tah J. (1999), "Genetic algorithms application and testing for equipment selection", J. Constr. Engrg. and Mgmt., ASCE, 125(1), 32-38.
9. Handbook of heavy construction (1971), edited by Havers & Stubbs, 2nd edition.
10. Harris, F. C. (1994), "Modern construction and ground engineering equipment and methods", 2nd, Edition, Longman Scientific & Technical, England.
11. Jayawardane A.K.W. & Harris F.C., (1990), "Further development of integer programming in earthwork optimization", J. Constr. Engrg. and Mgmt., ASCE, 116(1), 18-34.
12. Karafiath, Leslie L. (1988), "Rolling resistance of off-road vehicles", J. Constr. Engrg. and Mgmt., ASCE, 114(3), 458-471.

13. Karshenas Saeed (1989), "Truck capacity selection for earthmoving", J. Constr. Engrg. and Mgmt., ASCE, 115(2), 212-227.

14. Key, Jeffrey M., (1987), "Earthmoving and heavy equipment", J. Constr. Engrg. and Mgmt., ASCE, 113(4), 611-622.

15. Komatsu specifications and application handbook, Ist, Ed.

16. Nunnally, S.W. (1977), Managing construction equipment", Prentice Hall Inc. N.J.

17. O.Shea, J.B. Slutkin, G.N., and Shaffer, L.R., (1964), "An application of the theory of queues to the forecasting of shovel-truck fleet productions", Constr. Res. Series No.3, University of Illinois, Urbana, ILL.

18. Purifoy, R.L. (1979), "Construction planning equipment and methods", 3rd Ed., Mc-Graw Hills Inc., New York, N.Y.

19. Purifoy, R.L. (1985), "Construction planning equipment and methods", 4th Ed., Mc-Graw Hills Inc., New York, N.Y.

20. Ringwald Richard C., (1987), "Bunching theory applied to minimize cost", J. Constr. Engrg. and Mgmt., ASCE, 113(3), 321-326.

21. Schexnayder, Cliff, Weber Sandra L., & Brooks Brentwood T., (1999), "Effect of truck payload weight on production", J. Constr. Engrg. and Mgmt., ASCE, 125(1), 1-7.

22. Singh, Jagman (1980), "Art of earthmoving". 2nd Ed, Oxford & IBH Publication, New Delhi.

23. Smith Simon D., (1999), "Earthmoving productivity estimation using linear regression techniques", J. Constr. Engrg. and Mgmt., ASCE, 125(3), 133-141.

24. Taha, H.A. (1987), "Operations research 4th Ed., Mc-Millan Publishing Com., New York, N.Y.

25. Varma, Mahesh (1997), "Construction equipment and its planning and application"" 3rd Ed., Metropolitan Publishing Company, New Delhi.

APPENDIX - A

```
/*  ***********************************************************************

      PROGRAM OF  SCRAPER (MOTORISED)OPTIMISATION MODEL

                       (With  L.G.C)

  ***********************************************************************
*/

    #include<iostream.h>
    #include<conio.h>
    #include<math.h>
    #include<stdio.h>


    void main()

     {
     FILE*f1;
     FILE*f2;

 /*************************************/

        float HP,e,RR1,RR2,RR3,GR1,GR2,GR3;
        float RR1b,RR2b,RR3b,GR1b,GR2b,GR3b;
        float G,L,E,ct,d1,d2,d3;
        float d1b,d2b,d3b,vmax,fdl,fde;
        float k,Hc,Hcg,Hcc;
        float v[10];
        float sff1,sff2,sff3,sfb1,sfb2,sfb3;
        float hrs,q1,qy;
        float Osr,Opu;
        float RPm,D,eff,Lf;
        int i;
        float Lt[7],LPT[7];
        float DF,h;

 /*************************************/

clrscr();

f1=fopen("shma1.dat","r");
f2=fopen("shma1.res","w");
fscanf(f1,"%f%f%f%f%f%f%f%f",&HP,&e,&RR1,&RR2,&RR3,&GR1,&GR2,&GR3);
fscanf(f1,"%f%f%f%f%f%f",&RR1b,&RR2b,&RR3b,&GR1b,&GR2b,&GR3b);
fscanf(f1,"%f%f%f%f%f%f%f%f%f",&L,&E,&ct,&d1,&d2,&d3,&d1b,&d2b,&d3b);
fscanf(f1,"%f%f%f%f%f%f%f",&fdl,&fde,&k,&Hcg,&sff1,&sff2,&sff3);
fscanf(f1,"%f%f%f%f%f%f%f%f",&sfb1,&sfb2,&sfb3,&hrs,&qy,&Osr,&Opu,&vmax);
fscanf(f1,"%f%f%f%f",&D,&eff,&Lf,&h);

for(i=0;i<2;i++) fscanf(f1,"%f",&v[i]);

for(i=0;i<7;i++) fscanf(f1,"%f",&Lt[i]);

for(i=0;i<7;i++) fscanf(f1,"%f",&LPT[i]);
```

```
fclose(f1);
//fclose(f2);

        Hcc = (L*1000)/k ;

        if(Hcc>=Hcg)

        Hc = Hcg ;

        else

        Hc = Hcc ;

       cout<<"Adopted Heaped capacity of m/c is = "<<Hc<<endl;


       q1 = qy/hrs  ;

       if(h>900)

       DF = 3*((h-900)/300);

       else

       DF = 0;

       HP = HP*((100-DF)/100) ;

       cout<<"Available power is = "<<HP<<endl;

      fprintf(f2,"RESULTS OBTAINED FOR GIVEN CONDITION \n");
      fprintf(f2,"*************************************\n");

      fprintf(f2,"the value of required production, q1 is = ");
      fprintf(f2,"%0.3f\n",q1);

      RPm = (270*HP*e)/v[0];
      cout<<"RPm is ="<<RPm<<endl;


   //   ONWARD DIRECTION :

  //    SECTION A :

         float RP1,v1;
         float Tel;
         float Wdl;
         float t1,t11,t12;

        G = E + L ;

        Wdl = fdl*G ;

       Tel = ct*Wdl*1000 ;
       cout<<"Tel is ="<<Tel<<endl;
```

```
if(Tel>=RPm)

{

   RP1 = (RR1 + GR1)*G ;

   cout<<"RP1 is = "<<RP1<<endl;

   if((RPm>=RP1)&&(RP1>0))
     {
      v1 = (270*HP*e)/RP1;

      cout<<"vel v1 is ="<<v1<<endl;

     if(v1>=v[1])
     v1 = v[1];
     else v1;

      if(v1>=vmax)
     v1 = vmax;
     else
      v1;
     v1 = v1*sff1 ;

     cout<<"finally velocity v1 is = "<<v1<<endl;
    }
else if((RPm>=RP1)&&(RP1<=0))
    {
      v1 = v[1];
      if(v1>=vmax)
     v1 = vmax;
    else
      v1;
    v1 = v1*sff1 ;

     }
   else
  { cout<<"SCRAPER CAN'T OPERATE IN THIS CONDITION 1"<<endl; }

}

else if (Tel<RPm)
  {

   RPm = Tel;

   RP1 = (RR1 + GR1)*G ;

   if((RPm>=RP1)&&(RP1>0))
    {
      v1 = (270*HP*e)/RP1;

     if(v1>=v[1])
     v1 = v[1];
     else
     v1;
```

```
      if(v1>=vmax)
       v1 = vmax;
      else
        v1;
      v1 = v1*sff1 ;

      cout<<"finally velocity v1 is = "<<v1<<endl;
     }

    else if((RPm>=RP1)&&(RP1<=0))
      {
        v1 = v[1];
        if(v1>=vmax)
       v1 = vmax;
      else
        v1;
       v1 = v1*sff1 ;

        }
   else
    cout<<"SCRAPER CAN'T OPERATE IN THIS CONDITION 2"<<endl;

 }

       t11 = (Lt[6]*Hc)/Hcg ;
       cout<<"time for section t11 is =";
       cout<<t11<<endl;

       t12 = ((d1 - 0.1)/v1)*60 ;
       cout<<"time for section t12 is =";
       cout<<t12<<endl;

       t1 = t11 + t12;

       cout<<"time for section t1 is =";
       cout<<t1<<endl;

       getch();


//   ONWARD DIRECTION :

//    SECTION B :

      float RP2,v2;
      float t2;

     Tel = ct*Wdl*1000 ;

     RPm = (270*HP*e)/v[0];

   if(Tel>=RPm)

   {

      RP2 = (RR2 + GR2)*G ;
```

```
    if((RPm>=RP2)&&(RP2>0))

      {
       v2 = (270*HP*e)/RP2;

       if(v2>=v[1])
        v2 = v[1];
      else
        v2;

       if(v2>=vmax)
       v2 = vmax;
       else
       v2;
      v2 = v2*sff2 ;

     cout<<"finally velocity v2 is = "<<v2<<endl;
      }

 else if((RPm>=RP2)&&(RP2<=0))
     {
      v2 = v[1];
       if(v2>=vmax)
      v2 = vmax;
       else
       v2;
       v2 = v2*sff2 ;

      }
 else
   cout<<"SCRAPER CAN'T OPERATE IN THIS CONDITION 3"<<endl;

}
else if (Tel<RPm)
   {
    RPm = Tel;
    RP2 = (RR2 + GR2)*G ;
    if((RPm>=RP2)&&(RP2>0))

     {
       v2 = (270*HP*e)/RP2;

      if(v2>=v[1])
      v2 = v[1];
      else
      v2;

      if(v2>=vmax)
      v2 = vmax;
      else
       v2;
      v2 = v2*sff2 ;

      cout<<"finally velocity v2 is = "<<v2<<endl;
    }
```

```
 else if((RPm>=RP2)&&(RP2<=0))
     {
       v2 = v[1];
       if(v2>=vmax)
      v2 = vmax;
       else
       v2;
      v2 = v2*sff2 ;

     }
else
  cout<<"SCRAPER CAN'T OPERATE IN THIS CONDITION 4"<<endl;

 }

t2 =(d2/v2)*60;

cout<<"time for section t2 is =";
cout<<t2<<endl;
 getch();

//   ONWARD DIRECTION :

//    SECTION C :

 float RP3,v3,t3;

   Tel = ct*Wdl*1000 ;

   RPm = (270*HP*e)/v[0];

  if(Tel>=RPm)

  {

     RP3 = (RR3 + GR3)*G ;

     if((RPm>=RP3)&&(RP3>0))

       {

        v3 = (270*HP*e)/RP3;

      if(v3>=v[1])
       v3 = v[1];
      else
        v3;

      if(v3>=vmax)
       v3 = vmax;
      else
        v3;

       v3 = v3*sff3 ;

      cout<<"finally velocity v3 is = "<<v3<<endl;
     }
```

```
    else if((RPm>=RP3)&&(RP3<=0))
        {
          v3 = v[1];
          if(v3>=vmax)
         v3 = vmax;
          else
          v3;
      v3 = v3*sff3 ;

         }
  else
     cout<<"SCRAPER CAN'T OPERATE IN THIS CONDITION 5"<<endl;

  }
 else if (Tel<RPm)
     {
       RPm = Tel;
       RP3 = (RR3 + GR3)*G ;
       if((RPm>=RP3)&&(RP3>0))

        {

          v3 = (270*HP*e)/RP3;

          if(v3>=v[1])
         v3 = v[1];
          else
         v3;

          if(v3>=vmax)
         v3 = vmax;
          else
          v3;

          v3 = v3*sff3 ;

         cout<<"finally velocity v3 is = "<<v3<<endl;
       }

    else if((RPm>=RP3)&&(RP3<=0))
        {
          v3 = v[1];
          if(v3>=vmax)
         v3 = vmax;
          else
          v3;
          v3 = v3*sff3 ;

         }
    else
     cout<<"SCRAPER CAN'T OPERATE IN THIS CONDITION 6"<<endl;

    }

   t3 =(d3/v3)*60;
```

```
    cout<<"time for section t3 is =";
    cout<<t3<<endl;
     getch();

//    BACKWARD JOURNEY

//    SECTION C :

        float v3b;
       float Tee,RP3b;
       float Wde;
       float t3b,RPm1;

    G = E  ;

    Wde = fde*G ;

    Tee = ct*Wde*1000 ;

    RPm = (270*HP*e)/v[0];


    if(Tee>=RPm)

    {

       RP3b = (RR3b + GR3b)*G ;

       if((RPm>=RP3b)&&(RP3b>0))

         {
          v3b = (270*HP*e)/RP3b;

         if(v3b>=v[1])
          v3b = v[1];
         else
          v3b;

         if(v3b>=vmax)
          v3b = vmax;
         else
          v3b;
         v3b = v3b*sfb3 ;

        cout<<"finally velocity v3b1 is = "<<v3b<<endl;
       }

      else if((RPm>=RP3b)&&(RP3b<=0))
        {
          v3b = v[1];
          if(v3b>=vmax)
         v3b = vmax;
          else
          v3b;
          v3b = v3b*sfb3 ;

         }
```

```
      else
        cout<<"SCRAPER CAN'T OPERATE IN THIS CONDITION"<<endl;

      }
    else if (Tee<RPm)
      {
        RPm = Tee;

        RP3b = (RR3b + GR3b)*G ;

        if((RPm>=RP3b)&&(RP3b>0))

         {

           v3b = (270*HP*e)/RP3b;

           if(v3b>=v[1])
          v3b = v[1];
           else
          v3b;

          if(v3b>=vmax)
           v3b = vmax;
          else
           v3b;
           v3b = v3b*sfb3 ;

         cout<<"finally velocity v3b2 is = "<<v3b<<endl;
        }

       else if((RPm>=RP3b)&&(RP3b<=0))
         {
          v3b = v[1];
           if(v3b>=vmax)
          v3b = vmax;
           else
           v3b;
          v3b = v3b*sfb3 ;

          }
      else
       cout<<"SCRAPER CAN'T OPERATE IN THIS CONDITION"<<endl;

      }

      t3b =(d3b/v3b)*60;

      cout<<"time for section t3b is =";
      cout<<t3b<<endl;
       getch();

//      BACKWARD  JOURNEY :
//      SECTION B

        float v2b,RP2b;
        float t2b;
```

```
RPm = (270*HP*e)/v[0];

if(Tee>=RPm)

{
   RP2b = (RR2b + GR2b)*G ;

   if((RPm>=RP2b)&&(RP2b>0))
     {
      v2b = (270*HP*e)/RP2b;

     if(v2b>=v[1])
     v2b = v[1];
     else
     v2b;

    if(v2b>=vmax)
     v2b = vmax;
    else
      v2b;
    v2b = v2b*sfb2 ;
    cout<<"finally velocity v2b1 is = "<<v2b<<endl;
   }

  else if((RPm>=RP2b)&&(RP2b<=0))
    {
      v2b = v[1];
      if(v2b>=vmax)
     v2b = vmax;
    else
      v2b;
     v2b = v2b*sfb2 ;

     }
 else
  cout<<"SCRAPER CAN'T OPERATE IN THIS CONDITION"<<endl;

 }
else if (Tee<RPm)
  {
   RPm = Tee;

   RP2b = (RR2b + GR2b)*G ;

   if((RPm>=RP2b)&&(RP2b>0))

     {
      v2b = (270*HP*e)/RP2b;

    if(v2b>=v[1])
     v2b = v[1];
    else
     v2b;

    if(v2b>=vmax)
     v2b = vmax;
    else
```

```
          v2b;
          v2b = v2b*sfb2 ;

        cout<<"finally velocity v2b2 is = "<<v2b<<endl;
        }

      else if((RPm>=RP2b)&&(RP2b<=0))
        {
          v2b = v[1];
          if(v2b>=vmax)
         v2b = vmax;
        else
          v2b;
         v2b = v2b*sfb2 ;

         }
    else
     cout<<"SCRAPER CAN'T OPERATE IN THIS CONDITION"<<endl;

    }

   t2b =(d2b/v2b)*60;

   cout<<"time for section t2b is =";
   cout<<t2b<<endl;
    getch();


//      BACKWARD  JOURNEY :
//      SECTION A

        float v1b,RP1b;
       float t1b;

       RPm = (270*HP*e)/v[0];
   if(Tee>RPm)
   {
      RP1b = (RR1b + GR1b)*G ;

      if((RPm>=RP1b)&&(RP1b>0))

     {
        v1b = (270*HP*e)/RP1b;

       if(v1b>=v[1])
        v1b = v[1];
       else
        v1b;

        if(v1b>=vmax)
        v1b = vmax;
       else
         v1b;
       v1b = v1b*sfb1 ;

       cout<<"finally velocity v1b1 is = "<<v1b<<endl;
      }
```

```
    else if((RPm>=RP1b)&&(RP1b<=0))
      {
        v1b = v[1];
        if(v1b>=vmax)
       v1b = vmax;
      else
        v1b;
     v1b = v1b*sfb1 ;

       }
   else
    cout<<"SCRAPER CAN'T OPERATE IN THIS CONDITION"<<endl;

   }
 else if (Tee<RPm)
   {
     RPm = Tee;

    RP1b = (RR1b + GR1b)*G ;

     if((RPm>=RP1b)&&(RP1b>0))
      {
        v1b = (270*HP*e)/RP1b;

       if(v1b>=v[1])
       v1b = v[1];
       else
        v1b;

       if(v1b>=vmax)
       v1b = vmax;
       else
        v1b;
       v1b = v1b*sfb1 ;

      cout<<"finally velocity v1b2 is = "<<v1b<<endl;
     }

    else if((RPm>=RP1b)&&(RP1b<=0))
      {
        v1b = v[1];
        if(v1b>=vmax)
       v1b = vmax;
      else
        v1b;
      v1b = v1b*sfb1 ;

       }
   else
    cout<<"SCRAPER CAN'T OPERATE IN THIS CONDITION"<<endl;

   }

  t1b =(d1b/v1b)*60;

  cout<<"time for section t1b is =";
```

```
 cout<<t1b<<endl;
  getch();

//   TOTAL CYCLE TIME :

   float T,CT,CTc ;

   float Tps,q;

   T = t1+t2+t3+t1b+t2b+t3b ;

   fprintf(f2,"the value of travel time is T=");
   fprintf(f2,"%0.3f\n",T);

   CT = T + D - t11;

      float CTC[7],TPS[7],P[7];
      float d;
      int n=7;

       d = Lt[1]-Lt[0]   ;

        for(i=0;i<n;i++)
       {
       CTC[i] = Lt[i] + CT  ;
       TPS[i] = 50/CTC[i] ;

       P[i] = TPS[i] * LPT[i]  ;

       cout<<" HPR["<<i<<"] = "<<P[i]<<"    " ;
       cout<<" Lt["<<i<<"] = "<<Lt[i]<<"    ";
       cout<<" LPT["<<i<<"] = "<<LPT[i]<<endl;
       }
       getch();

     float Pm,t,tl,th,Lm;

     Pm = P[0] ;

     for(i=0;i<n;i++)
     {

      if(P[i]>Pm)
       Pm = P[i] ;
      if(Pm == P[i])
      {
       t=i;tl=i-1;
         if(i+1==n)
         th = i ;
         else
         { th = i+1 ; }
         Lm = LPT[i];
       }
     }
     cout<<"optimum time,t0 is"<<" "<<Lt[t]<<"  "
         <<" time1 is"<<" "<<Lt[tl]<<"  "
          <<"time2 is"<<Lt[th]<<endl;
```

```
    cout<<"P0 is = "<<Pm<<endl;
    cout<<"P1 is = "<<P[tl]<<endl;
    cout<<"P2 is = "<<P[th]<<endl;
    cout<<"optimum load ,LPT is = "<<Lm<<endl;
    cout<<"L1 is = "<<LPT[tl]<<endl;
    cout<<"L2 is = "<<LPT[th]<<endl;


   float  g1 = (Pm-P[tl]);
   float  d11 = (Lm - LPT[tl]);
   float  g2 = (P[tl] + P[th] -2* Pm);
   float  d22 = (LPT[tl] + LPT[th] -2*Lm);

   float to = Lt[tl] + 0.5*(d) - (g1*d)/g2 ;

   if(to>Lt[th])
   to = Lt[th];
   else
   to ;
   cout<<"Exact value of optimum loading time is = "<<to<<endl;
   getch();

    float p = (to- Lt[tl])/d ;

    float Lo = LPT[tl] + p*d11 + (p*(p-1)*d22)/2 ;

   cout<<"Exact value of optimum load is = "<<Lo<<endl;

   getch();

   CTc = to + CT;

  fprintf(f2,"the value of total cycle time CTc is =  ");
  fprintf(f2,"%0.3f\n",CTc);

 CT = CTc/eff ;
 Tps = 60/CT ;
 q =   Lo*Lf*Tps  ;

fprintf(f2,"the value of productivity q  is = ");
fprintf(f2,"%0.3f\n",q);

float S1,Sr;

S1 = q1/q ;

 int Srr = int(S1) ;

  Sr = Srr + 1 ;

  fprintf(f2,"No of scrapers required is Sr = ");
  fprintf(f2,"%0.3f\n",Sr);

  getch();

//   PUSH LOADING BY PUSHER
//   -----------------------
```

```
float PCT,Srp,Pu,PU,SrCIN,FCIN;
int bt;

cout<<"Type of loading "<<endl
    <<"Use 0(zero) for back track loading"<<endl
    <<"Use digit other than 0(zero) for chain loading"<<endl
    <<"give digit = ";
cin>>bt;

if( bt == 0)

  PCT = 1.5*to ;
else

  PCT = 1.3*to ;

cout<<"PCT is = "<<PCT<<endl;

Srp  = CTc/PCT ;

Pu = Sr/Srp ;

int pu = int(Pu) ;

PU = pu + 1 ;

fprintf(f2,"No of pushers required is PU = ");
fprintf(f2,"%0.3f\n",PU);

SrCIN =  Osr/q ;

fprintf(f2,"Economic cost index of Scraper is =");
fprintf(f2,"%0.3f\n",SrCIN);

FCIN = ((PU*Opu) + (Sr*Osr))/(q*Sr)  ;

fprintf(f2,"Economic cost index of fleet is =");
fprintf(f2,"%0.3f\n",FCIN);

getch();

}
```

APPENDIX - B

```
/*  *******************************************************************

      PROGRAM OF  SCRAPER (MOTORISED/WHEELED) OPTIMISATION MODEL

                        (Without  L.G.C)

   ********************************************************************
 */

     #include<iostream.h>
     #include<conio.h>
     #include<math.h>
     #include<stdio.h>


     void main()

      {
      FILE*f1;
      FILE*f2;

 /**************************************/

        float HP,e,RR1,RR2,RR3,GR1,GR2,GR3;
        float RR1b,RR2b,RR3b,GR1b,GR2b,GR3b;
        float G,L,E,ct,d1,d2,d3;
        float d1b,d2b,d3b,vmax,fdl,fde;
        float k,Hc,Hcg,Hcc;
        float v[10];
        float sff1,sff2,sff3,sfb1,sfb2,sfb3;
        float hrs,q1,qy;
        float Osr,Opu;
        float RPm,D,eff,Lf;
        int i;
        float Lt,DF,h;
 /**************************************/

clrscr();

f1=fopen("shmall.dat","r");
f2=fopen("shmall.res","w");
fscanf(f1,"%f%f%f%f%f%f%f%f",&HP,&e,&RR1,&RR2,&RR3,&GR1,&GR2,&GR3);
fscanf(f1,"%f%f%f%f%f%f",&RR1b,&RR2b,&RR3b,&GR1b,&GR2b,&GR3b);
fscanf(f1,"%f%f%f%f%f%f%f%f%f",&L,&E,&ct,&d1,&d2,&d3,&d1b,&d2b,&d3b);
fscanf(f1,"%f%f%f%f%f%f%f",&fdl,&fde,&k,&Hcg,&sff1,&sff2,&sff3);
fscanf(f1,"%f%f%f%f%f%f%f%f",&sfb1,&sfb2,&sfb3,&hrs,&qy,&Osr,&Opu,&vmax);
fscanf(f1,"%f%f%f%f%f",&D,&eff,&Lf,&Lt,&h);

for(i=0;i<2;i++) fscanf(f1,"%f",&v[i]);

fclose(f1);
//fclose(f2);

       Hcc = (L*1000)/k ;
```

```
    if(Hcc>=Hcg)

    Hc = Hcg ;

    else

    Hc = Hcc ;

   cout<<"Adopted Heaped capacity of m/c is = "<<Hc<<endl;

   Lt = (Lt*Hc)/Hcg ;
   cout<<"Adopted loading time of m/c is = "<<Lt<<endl;

  q1 = qy/hrs   ;

  if(h>900)

   DF = 3*((h-900)/300);
   else
   DF = 0;

   HP = HP*((100-DF)/100) ;

   cout<<"Available power is = "<<HP<<endl;

 fprintf(f2,"RESULTS OBTAINED FOR GIVEN CONDITION \n");
 fprintf(f2,"**********************************\n");

 fprintf(f2,"The value of required production, q1 is = ");
 fprintf(f2,"%0.3f\n",q1);

 RPm = (270*HP*e)/v[0];

//   ONWARD DIRECTION :

//   SECTION A :

     float RP1,v1;
     float Tel;
     float Wdl;
     float t1,t11,t12;

    G = E + L ;
    Wdl = fdl*G ;


  Tel = ct*Wdl*1000 ;


  if(Tel>=RPm)

  {

     RP1 = (RR1 + GR1)*G ;
     cout<<"RP1 is = "<<RP1<<endl;

     if((RPm>=RP1)&&(RP1>0))
```

```
      {
       v1 = (270*HP*e)/RP1;
       cout<<"vel v1 is ="<<v1<<endl;

      if(v1>=v[1])
      v1 = v[1];
      else v1;

     if(v1>=vmax)
      v1 = vmax;
     else
       v1;
      v1 = v1*sff1 ;

     cout<<"finally velocity v1 is = "<<v1<<endl;
    }
   else if((RPm>=RP1)&&(RP1<=0))
     {
       v1 = v[1];
       if(v1>=vmax)
      v1 = vmax;
     else
       v1;
     v1 = v1*sff1 ;

      }
    else
    cout<<"SCRAPER CAN'T OPERATE IN THIS CONDITION"<<endl;

 }

else if (Tel<RPm)
  {
    RPm = Tel;
    RP1 = (RR1 + GR1)*G ;

    if((RPm>=RP1)&&(RP1>0))

     {
       v1 = (270*HP*e)/RP1;

      if(v1>=v[1])
      v1 = v[1];
      else
      v1;

     if(v1>=vmax)
      v1 = vmax;
     else
       v1;
     v1 = v1*sff1 ;

     cout<<"finally velocity v1 is = "<<v1<<endl;
    }

   else if((RPm>=RP1)&&(RP1<=0))
```

```
        {
          v1 = v[1];
          if(v1>=vmax)
         v1 = vmax;
        else
          v1;
         v1 = v1*sff1 ;

        }
     else
      cout<<"SCRAPER CAN'T OPERATE IN THIS CONDITION"<<endl;

   }

         t11 = (Lt*Hc)/Hcg ;
         cout<<"time for section t11 is =";
         cout<<t11<<endl;

          t12 = ((d1 - 0.1)/v1)*60 ;
         cout<<"time for section t12 is =";
         cout<<t12<<endl;

         t1 = t11 + t12;

         cout<<"time for section t1 is =";
         cout<<t1<<endl;

     getch();

  //   ONWARD DIRECTION :

 //    SECTION B :

       float RP2,v2;
       float t2;

      Tel = ct*Wd1*1000 ;

      RPm = (270*HP*e)/v[0];

    if(Tel>=RPm)

    {

       RP2 = (RR2 + GR2)*G ;

       if((RPm>=RP2)&&(RP2>0))

         {
          v2 = (270*HP*e)/RP2;

         if(v2>=v[1])
         v2 = v[1];
         else
          v2;

        if(v2>=vmax)
```

```
      v2 = vmax;
     else
       v2;
      v2 = v2*sff2 ;

     cout<<"finally velocity v2 is = "<<v2<<endl;
    }

   else if((RPm>=RP2)&&(RP2<=0))
     {
       v2 = v[1];
       if(v2>=vmax)
      v2 = vmax;
     else
       v2;
      v2 = v2*sff2 ;

      }
  else
   cout<<"SCRAPER CAN'T OPERATE IN THIS CONDITION"<<endl;

  }
else if (Tel<RPm)
  {
    RPm = Tel;
    RP2 = (RR2 + GR2)*G ;

    if((RPm>=RP2)&&(RP2>0))

     {

       v2 = (270*HP*e)/RP2;

     if(v2>=v[1])
      v2 = v[1];
     else
      v2;

     if(v2>=vmax)
      v2 = vmax;
     else
       v2;
      v2 = v2*sff2 ;

     cout<<"finally velocity v2 is = "<<v2<<endl;
    }

   else if((RPm>=RP2)&&(RP2<=0))
     {
       v2 = v[1];
       if(v2>=vmax)
      v2 = vmax;
     else
       v2;
     v2 = v2*sff2 ;

      }
```

```
    else
     cout<<"SCRAPER CAN'T OPERATE IN THIS CONDITION"<<endl;

    }

   t2 =(d2/v2)*60;

   cout<<"time for section t2 is =";
   cout<<t2<<endl;
    getch();

 //   ONWARD DIRECTION :

//    SECTION C :

  float RP3,v3,t3;

    Tel = ct*Wdl*1000 ;

    RPm = (270*HP*e)/v[0];

   if(Tel>=RPm)

   {
      RP3 = (RR3 + GR3)*G ;

      if((RPm>=RP3)&&(RP3>0))

        {
         v3 = (270*HP*e)/RP3;

        if(v3>=v[1])
        v3 = v[1];
        else
        v3;

        if(v3>=vmax)
        v3 = vmax;
        else
         v3;

        v3 = v3*sff3 ;

       cout<<"finally velocity v3 is = "<<v3<<endl;
      }

     else if((RPm>=RP3)&&(RP3<=0))
       {
         v3 = v[1];
         if(v3>=vmax)
        v3 = vmax;
       else
         v3;
      v3 = v3*sff3 ;

        }
    else
```

```
    cout<<"SCRAPER CAN'T OPERATE IN THIS CONDITION"<<endl;

   }
 else if (Tel<RPm)
   {
     RPm = Tel;

     RP3 = (RR3 + GR3)*G ;

     if((RPm>=RP3)&&(RP3>0))

      {
        v3 = (270*HP*e)/RP3;

       if(v3>=v[1])
       v3 = v[1];
       else
       v3;

        if(v3>=vmax)
       v3 = vmax;
       else
        v3;

       v3 = v3*sff3 ;

      cout<<"finally velocity v3 is = "<<v3<<endl;
     }

    else if((RPm>=RP3)&&(RP3<=0))
      {
        v3 = v[1];
        if(v3>=vmax)
       v3 = vmax;
      else
        v3;
     v3 = v3*sff3 ;

      }
 else
    cout<<"SCRAPER CAN'T OPERATE IN THIS CONDITION"<<endl;
 }

  t3 =(d3/v3)*60;

  cout<<"time for section t3 is =";
  cout<<t3<<endl;
   getch();

//    BACKWARD JOURNEY

//    SECTION C :

     float v3b;
    float Tee,RP3b;
    float Wde;
    float t3b,RPm1;
```

```
G = E  ;

Wde = fde*G ;

Tee = ct*Wde*1000 ;

RPm = (270*HP*e)/v[0];


if(Tee>=RPm)

{

   RP3b = (RR3b + GR3b)*G ;

   if((RPm>=RP3b)&&(RP3b>0))

     {
      v3b = (270*HP*e)/RP3b;

     if(v3b>=v[1])
      v3b = v[1];
     else
      v3b;

      if(v3b>=vmax)
      v3b = vmax;
     else
       v3b;
     v3b = v3b*sfb3 ;

     cout<<"finally velocity v3b1 is = "<<v3b<<endl;
    }

   else if((RPm>=RP3b)&&(RP3b<=0))
     {
       v3b = v[1];
       if(v3b>=vmax)
      v3b = vmax;
     else
       v3b;
     v3b = v3b*sfb3 ;

      }
  else
   cout<<"SCRAPER CAN'T OPERATE IN THIS CONDITION"<<endl;

  }
else if (Tee<RPm)
  {
    RPm = Tee;

    RP3b = (RR3b + GR3b)*G ;

    if((RPm>=RP3b)&&(RP3b>0))
```

```
        {
          v3b = (270*HP*e)/RP3b;

         if(v3b>=v[1])
         v3b = v[1];
         else
         v3b;

        if(v3b>=vmax)
         v3b = vmax;
        else
          v3b;
         v3b = v3b*sfb3 ;

        cout<<"finally velocity v3b2 is = "<<v3b<<endl;
       }

      else if((RPm>=RP3b)&&(RP3b<=0))
        {
          v3b = v[1];
          if(v3b>=vmax)
         v3b = vmax;
        else
          v3b;
        v3b = v3b*sfb3 ;

         }
    else
      cout<<"SCRAPER CAN'T OPERATE IN THIS CONDITION"<<endl;

    }

   t3b =(d3b/v3b)*60;

   cout<<"time for section t3b is =";
   cout<<t3b<<endl;
    getch();

//     BACKWARD  JOURNEY :
//     SECTION B

       float v2b,RP2b;
        float t2b;

    RPm = (270*HP*e)/v[0];

   if(Tee>=RPm)

   {
      RP2b = (RR2b + GR2b)*G ;

      if((RPm>=RP2b)&&(RP2b>0))

        {
        v2b = (270*HP*e)/RP2b;

        if(v2b>=v[1])
```

```
      v2b = v[1];
      else
      v2b;

     if(v2b>=vmax)
      v2b = vmax;
     else
       v2b;
      v2b = v2b*sfb2 ;

     cout<<"finally velocity v2b1 is = "<<v2b<<endl;
    }

   else if((RPm>=RP2b)&&(RP2b<=0))
     {
       v2b = v[1];
       if(v2b>=vmax)
      v2b = vmax;
       else
       v2b;
      v2b = v2b*sfb2 ;

       }
  else
   cout<<"SCRAPER CAN'T OPERATE IN THIS CONDITION"<<endl;

  }
else if (Tee<RPm)
  {
    RPm = Tee;

    RP2b = (RR2b + GR2b)*G ;

    if((RPm>=RP2b)&&(RP2b>0))

     {
       v2b = (270*HP*e)/RP2b;

     if(v2b>=v[1])
      v2b = v[1];
     else
      v2b;

     if(v2b>=vmax)
      v2b = vmax;
     else
       v2b;
       v2b = v2b*sfb2 ;

     cout<<"finally velocity v2b2 is = "<<v2b<<endl;
    }

   else if((RPm>=RP2b)&&(RP2b<=0))
     {
       v2b = v[1];
       if(v2b>=vmax)
      v2b = vmax;
```

(iii) Calculate approximate optimum number of haulers n_1 by using following relation

$$n_1 = \frac{1}{r},$$ and roundup this figure.

(iv) Next consider the number of haulers from $n_1 - m$ to $n_1 + m$ for $m = 2$.

(v) Compute the probability p_t for each number of haulers,

$$p_t = 1 - p_o,$$

where,

$$p_o = \left[\sum_{i=0}^{n} \frac{n!}{(n-i)!} (r)^i \right]^{-1}$$

(vi) The expected production is given by

Expected production, $P_j = 1.03 \times LP \times p_t$ (m^3/hr)

It is to be noted that LP is taken at 100% loader efficiency because a loader typically operates at or near 100% efficiency when actually engaged in loading [Nunnally, 1977].

(vi) Taking the hourly owning and operating cost of the haul unit as Ohl and for loader as Ol, the fleet cost index number may be obtained by the following equation,

$$(CIN)_j = \frac{j \times Ohl + Ol}{Pj}$$

where,

j = number of haulers

(vii) The minimum value of CIN is selected and the corresponding value of j gives the optimum number of haul units for the chosen loader.

(viii) Repeat steps from (i to viii) for other haul unit sizes.

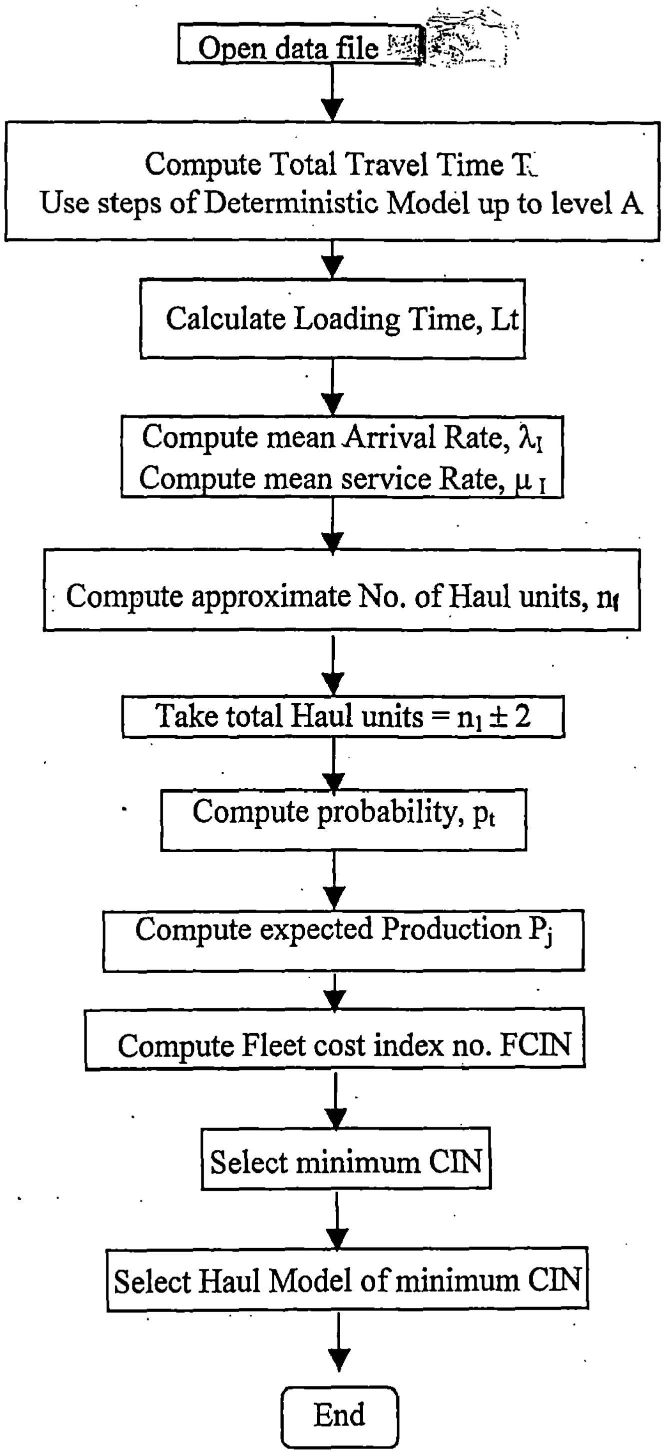
(ix) Select the haul unit, giving minimum value of CIN.

Finally, j = number of haul units

Hcg = size of haul unit (m^3).

The flow chart enclosed shows the steps of the program. The program has been given in Appendix – G.

FUNCTIONAL FLOW CHART FOR HAUL UNIT
(Probabilistic Model)



```
        else
          v2b;
         v2b = v2b*sfb2 ;

         }
     else
      cout<<"SCRAPER CAN'T OPERATE IN THIS CONDITION"<<endl;

     }

    t2b =(d2b/v2b)*60;

    cout<<"time for section t2b is =";
    cout<<t2b<<endl;
     getch();

//     BACKWARD  JOURNEY :
//     SECTION A

       float v1b,RP1b;
      float t1b;

      RPm =  (270*HP*e)/v[0];

    if((RPm>=RP1b)&&(RP1b>0))

    {
       RP1b =  (RR1b + GR1b)*G ;

       if(RPm>=RP1b)

         {
          v1b =  (270*HP*e)/RP1b;

         if(v1b>=v[1])
         v1b = v[1];
         else
         v1b;

         if(v1b>=vmax)
         v1b = vmax;
        else
          v1b;
          v1b = v1b*sfb1 ;

        cout<<"finally velocity v1b1 is = "<<v1b<<endl;
       }

      else if((RPm>=RP1b)&&(RP1b<=0))
        {
          v1b = v[1];
          if(v1b>=vmax)
         v1b = vmax;
        else
          v1b;
       v1b = v1b*sfb1 ;
```

```
        }
      else
       cout<<"SCRAPER CAN'T OPERATE IN THIS CONDITION"<<endl;

      }
    else if (Tee<RPm)
      {
        RPm = Tee;

       RP1b = (RR1b + GR1b)*G ;

        if((RPm>=RP1b)&&(RP1b>0))

         {
          v1b = (270*HP*e)/RP1b;

          if(v1b>=v[1])
          v1b = v[1];
          else
           v1b;

         if(v1b>=vmax)
          v1b = vmax;
         else
           v1b;
           v1b = v1b*sfb1 ;

         cout<<"finally velocity v1b2 is = "<<v1b<<endl;
        }

       else if((RPm>=RP1b)&&(RP1b<=0))
         {
           v1b = v[1];
           if(v1b>=vmax)
          v1b = vmax;
         else
           v1b;
         v1b = v1b*sfb1 ;

          }
      else
       cout<<"SCRAPER CAN'T OPERATE IN THIS CONDITION"<<endl;

      }

    t1b =(d1b/v1b)*60;

    cout<<"time for section t1b is =";
    cout<<t1b<<endl;
     getch();

//   TOTAL CYCLE TIME :

    float T,CT,CTc ;

    float Tps,q;
```

```
T = t1+t2+t3+t1b+t2b+t3b ;

fprintf(f2,"the value of travel time is T=");
fprintf(f2,"%0.3f\n",T);

CT = T + D;

CTc =  CT;

fprintf(f2,"the value of total cycle time CTc is =  ");
fprintf(f2,"%0.3f\n",CTc);

CT = CTc/eff ;
Tps = 60/CT ;
q =   Hc*Lf*Tps  ;

fprintf(f2,"the value of productivity q  is = ");
fprintf(f2,"%0.3f\n",q);

float S1,Sr;

S1 = q1/q ;

 int Srr = int(S1) ;

  Sr = Srr + 1 ;

fprintf(f2,"No of scrapers required is Sr = ");
fprintf(f2,"%0.3f\n",Sr);

  getch();

//   PUSH LOADING BY PUSHER
//   -----------------------

  float PCT,Srp,Pu,PU,SrCIN,FCIN;
  int bt;

  cout<<"Type of loading "<<endl
      <<"Use 0(zero) for back track loading"<<endl
      <<"Use digit other than 0(zero) for chain loading"<<endl
      <<"give digit = ";
  cin>>bt;

  if( bt == 0)

    PCT = 1.5*Lt ;
  else

    PCT = 1.3*Lt ;

 cout<<"PCT is = "<<PCT<<endl;

  Srp  = CTc/PCT ;

  Pu = Sr/Srp ;
```

```
  int pu = int(Pu) ;

  PU = pu + 1 ;

fprintf(f2,"No of pushers required is PU = ");
fprintf(f2,"%0.3f\n",PU);

  SrCIN =  Osr/q ;

fprintf(f2,"Economic cost index of Scraper is =");
fprintf(f2,"%0.3f\n",SrCIN);

  FCIN = ((PU*Opu) + (Sr*Osr))/(q*Sr)   ;

fprintf(f2,"Economic cost index of fleet is =");
fprintf(f2,"%0.3f\n",FCIN);

 getch();

}
```

APPENDIX - C

```
/* *********************************************************************

          PROGRAM OF  SCRAPER (TOWED) OPTIMISATION MODEL

                         (With  L.G.C)

   *********************************************************************
*/

    #include<iostream.h>
    #include<conio.h>
    #include<math.h>
    #include<stdio.h>


    void main()

     {
     FILE*f1;
     FILE*f2;

/***************************************/

       float HP,e,L,ct,RR1,RR2,GR1,GR2;
       float RR1b,RR2b,GR1b,GR2b;
       float Gpf,Gpb,Gc,Ge;
       float vmax,fdl,fde;
       float k,Hc,Hcg,Hcc;
       float v[10],vf[10],vb[10],DB[10],NDB[10],Lt[7],LPT[7];
       float sff1,sff2,sfb1,sfb2;
       float hrs,q1,qy;
       float Osr,Opu;
       float NDBm,Rc,NRRC,D,eff,Lf;
       float PR1,PR2,TR1,TR2,TR1b,TR2b,TEl,PR1b,PR2b;
       float vf1,vf2,vb1,vb2,tf1,tf2,tb1,tb2,df1,df2,db1,db2;
       int i;
       float DF,h;

/****************************************/

clrscr();

f1=fopen("src1.dat","r");
f2=fopen("src1.res","w");
fscanf(f1,"%f%f%f%f%f%f%f",&HP,&e,&RR1,&RR2,&Rc,&GR1,&GR2);
fscanf(f1,"%f%f%f%f",&RR1b,&RR2b,&GR1b,&GR2b);
fscanf(f1,"%f%f%f%f%f%f%f%f",&L,&Gc,&Ge,&ct,&df1,&df2,&db1,&db2);
fscanf(f1,"%f%f%f%f%f%f",&fdl,&fde,&k,&Hcg,&sff1,&sff2);
fscanf(f1,"%f%f%f%f%f%f%f",&sfb1,&sfb2,&hrs,&qy,&Osr,&Opu,&vmax);
fscanf(f1,"%f%f%f%f",&D,&eff,&Lf,&h);

for(i=0;i<5;i++) fscanf(f1,"%f",&vf[i]);
for(i=0;i<5;i++) fscanf(f1,"%f",&vb[i]);
for(i=0;i<5;i++) fscanf(f1,"%f",&DB[i]);
```

```
for(i=0;i<7;i++) fscanf(f1,"%f",&Lt[i]);
for(i=0;i<7;i++) fscanf(f1,"%f",&LPT[i]);


fclose(f1);
//fclose(f2);

        Hcc = (L*1000)/k ;

        if(Hcc>=Hcg)

        Hc = Hcg ;

        else

        Hc = Hcc ;

        cout<<"Adopted Heaped capacity of m/c is = "<<Hc<<endl;


       q1 = qy/hrs  ;

       if(h>900)

        DF = 3*((h-900)/300);
        else
        DF = 0;

        HP = HP*((100-DF)/100) ;

        cout<<"Available power is = "<<HP<<endl;


       fprintf(f2,"RESULTS OBTAINED FOR GIVEN CONDITION \n");
       fprintf(f2,"**************************************\n");

       fprintf(f2,"The value of required production, q1 is =");
       fprintf(f2,"%0.3f\n",q1);
//**********************************************************************

//**********************************************************************
        NRRC = (Rc - 55)*Gc ;

        cout<<"NRRC is = "<<NRRC<<endl;

        for(i=0;i<5;i++)
         {
          NDB[i] = ((DB[i] - NRRC)*.85) ;
         cout<<"NDB[i] is = "<<NDB[i];
         }
        NDBm = NDB[0]*((100-DF)/100) ;

        cout<<"NDBm is ="<<NDBm<<endl;
```

```
Gpf = L + Ge ;

PR1 = Gpf*RR1 ;
PR2 = Gpf*RR2 ;

cout<<"PR1 is ="<<PR1<<endl;

TEl = ct*Gc*1000 ;

cout<<"TEl is ="<<TEl<<endl;

if(TEl>=NDBm)
    {
     TR1 = GR1*(Gc + Gpf) + PR1 ;

     cout<<"TR1 is ="<<TR1<<endl;

     if((NDBm>=TR1)&&(TR1>0))
        {

          if(TR1==NDBm)

           vf1 = vf[0] ;

        else if((TR1>NDB[1])&&(TR1<=NDB[0]))
           vf1 = vf[0] ;

        else if((TR1>NDB[2])&&(TR1<=NDB[1]))
           vf1 = vf[1] ;

        else if((TR1>NDB[3])&&(TR1<=NDB[2]))
           vf1 = vf[2] ;

        else if((TR1>NDB[4])&&(TR1<=NDB[3]))
           vf1 = vf[3] ;

        else if(TR1<=NDB[4])
           vf1 = vf[4] ;

       cout<<"VEl vf1 is ="<<vf1<<endl;

         if(vf1>=vmax)
           vf1 = vmax ;
         else if(vf1<vmax)
           vf1;
           vf1 = vf1*sff1;
       cout<<"finally VEl vf1 is ="<<vf1<<endl;

     }

   else if((NDBm>=TR1)&&(TR1<=0))
     {
        vf1 = vf[4];
```

```
            if(vf1>=vmax)
                vf1 = vmax ;
            else if(vf1<vmax)
                vf1;vf1=vf1*sff1;

        }

    else
        {
          cout<<"Total Resistance is very high "<<endl
             <<"Tractor can't perform in this condition"<<endl;
        }

    }

else if(TEl<NDBm)
      {
        NDBm = TEl ;

        cout<<"NDBm for less TEl is ="<<NDBm<<endl;

        TR1 = GR1*(Gc + Gpf) + PR1 ;

        cout<<"TR1 for less TEl is ="<<TR1<<endl;

        if((NDBm>=TR1)&&(TR1>0))
           {
            if(TR1==NDBm)

               vf1 = vf[0] ;

            else if((TR1>NDB[1])&&(TR1<=NDB[0]))
               vf1 = vf[0] ;

            else if((TR1>NDB[2])&&(TR1<=NDB[1]))
               vf1 = vf[1] ;

            else if((TR1>NDB[3])&&(TR1<=NDB[2]))
               vf1 = vf[2] ;

            else if((TR1>NDB[4])&&(TR1<=NDB[3]))
               vf1 = vf[3] ;

            else if(TR1<=NDB[4])
               vf1 = vf[4] ;

           cout<<"VEl vf1 for less tel is ="<<vf1<<endl;

             if(vf1>=vmax)
               vf1 = vmax ;
             else if(vf1<vmax)
               vf1;
              vf1 = vf1*sff1;
          cout<<"finally VEl vf1 for less Tel is ="<<vf1<<endl;
```

```
            }

        else if((NDBm>=TR1)&&(TR1<=0))
            {
                vf1 = vf[4];

                if(vf1>=vmax)
                    vf1 = vmax ;
                else if(vf1<vmax)
                    vf1; vf1=vf1*sff1;
            }

        else
            {
                cout<<"Total Resistance is very high "<<endl
                    <<"Tractor can't perform in this condition"<<endl;
            }

    }


        float  t11 = (Lt[6]*Hc)/Hcg;

        cout<<"finally time for section t11 is ="<<t11<<endl;

        float  t12 = ((df1 - 0.1)/vf1)*60 ;
                cout<<"time for section t12 is =";
                cout<<t12<<endl;

                tf1 = t11 + t12;

                cout<<"time for section A is =";
                cout<<tf1<<endl;


// ONWARD MOVEMENT ;
// SECTION B :

            NDBm = NDB[0]*((100-DF)/100) ;
            cout<<"NDBm is ="<<NDBm<<endl;


    if(TEl>=NDBm)
        {
            TR2 = GR2*(Gc + Gpf) + PR2 ;

            cout<<"TR2 is ="<<TR2<<endl;

            if((NDBm>=TR2)&&(TR2>0))
                {

                    if(TR2==NDBm)

                    vf2 = vf[0] ;
```

```
            else if((TR2>NDB[1])&&(TR2<=NDB[0]))
               vf2 = vf[0] ;

            else if((TR2>NDB[2])&&(TR2<=NDB[1]))
               vf2 = vf[1] ;

            else if((TR2>NDB[3])&&(TR2<=NDB[2]))
               vf2 = vf[2] ;

            else if((TR2>NDB[4])&&(TR2<=NDB[3]))
               vf2 = vf[3] ;

            else if(TR2<=NDB[4])
               vf2 = vf[4] ;

          cout<<"VEl vf2 is ="<<vf2<<endl;

            if(vf2>=vmax)
               vf2 = vmax ;
            else if(vf2<vmax)
               vf2;
             vf2 = vf2*sff2;
          cout<<"finally VEl vf2 is ="<<vf2<<endl;

       }

     else if((NDBm>=TR2)&&(TR2<=0))
       {
           vf2 = vf[4];

          if(vf2>=vmax)
               vf2 = vmax ;
           else if(vf2<vmax)
               vf2; vf2 = vf2*sff2;

       }

     else
       {
          cout<<"Total Resistance is very high "<<endl
             <<"Tractor can't perform in this condition"<<endl;
       }

   }

else if(TEl<NDBm)
     {
       NDBm = TEl ;

      cout<<"NDBm for less TEl is ="<<NDBm<<endl;

       TR2 = GR2*(Gc + Gpf) + PR2 ;

      cout<<"TR1 for less TEl is ="<<TR2<<endl;
```

```
if((NDBm>=TR2)&&(TR2>0))
    {
     if(TR2==NDBm)

        vf2 = vf[0] ;

     else if((TR2>NDB[1])&&(TR2<=NDB[0]))
        vf2 = vf[0] ;

     else if((TR2>NDB[2])&&(TR2<=NDB[1]))
        vf2 = vf[1] ;

     else if((TR2>NDB[3])&&(TR2<=NDB[2]))
        vf2 = vf[2] ;

     else if((TR2>NDB[4])&&(TR2<=NDB[3]))
        vf2 = vf[3] ;

     else if(TR2<=NDB[4])
        vf2 = vf[4] ;

    cout<<"VEl vf1 for less tel is ="<<vf2<<endl;

     if(vf2>=vmax)
        vf2 = vmax ;
     else if(vf2<vmax)
        vf2;
     vf2 = vf2*sff2;
   cout<<"finally VEl vf1 for less Tel is ="<<vf2<<endl;

 }

else if((NDBm>=TR2)&&(TR2<=0))
 {
     vf2 = vf[4];

    if(vf2>=vmax)
        vf2 = vmax ;
     else if(vf2<vmax)
        vf2; vf2 = vf2*sff2;

 }

else
 {
   cout<<"Total Resistance is very high "<<endl
      <<"Tractor can't perform in this condition"<<endl;
 }

}

 tf2 = (df2/vf2)*60 ;

 cout<<"Time for section Bf in min. is = "<<tf2<<endl;
```

```
// BACKWARD JOURNEY
// SECTION B

        Gpb = Ge ;

        PR2b = Gpb*RR2b ;

          NDBm = NDB[0]*((100-DF)/100) ;

         cout<<"NDBm is ="<<NDBm<<endl;


  if(TEl>=NDBm)
        {
          TR2b = GR2b*(Gc + Gpb) + PR2b ;

          cout<<"TR2b is ="<<TR2b<<endl;

          if((NDBm>=TR2b)&&(TR2b>0))
              {

                 if(TR2b==NDBm)

                  vb2 = vb[0] ;

               else if((TR2b>NDB[1])&&(TR2b<=NDB[0]))
                  vb2 = vb[0] ;

               else if((TR2b>NDB[2])&&(TR2b<=NDB[1]))
                  vb2 = vb[1] ;

               else if((TR2b>NDB[3])&&(TR2b<=NDB[2]))
                  vb2 = vb[2] ;

               else if((TR2b>NDB[4])&&(TR2b<=NDB[3]))
                  vb2 = vb[3] ;

               else if(TR2b<=NDB[4])
                  vb2 = vb[4] ;

             cout<<"VEl vb2 is ="<<vb2<<endl;

               if(vb2>=vmax)
                  vb2 = vmax ;
               else if(vb2<vmax)
                  vb2;
                vb2 = vb2*sfb2;

            cout<<"finally VEl vb2 is ="<<vb2<<endl;

          }

        else if((NDBm>=TR2b)&&(TR2b<=0))
          {
              vb2 = vb[4];
```

```
        if(vb2>=vmax)
            vb2 = vmax ;
         else if(vb2<vmax)
            vb2; vb2 = vb2*sfb2;

    }

  else
    {
      cout<<"Total Resistance is very high "<<endl
         <<"Tractor can't perform in this condition"<<endl;
    }

  }

else if(TEl<NDBm)
    {
     NDBm = TEl ;

     cout<<"NDBm for less TEl is ="<<NDBm<<endl;

     TR2b = GR2b*(Gc + Gpb) + PR2b ;

     cout<<"TR2 for less TEl is ="<<TR2b<<endl;

     if((NDBm>=TR2b)&&(TR2b>0))
        {
         if(TR2b==NDBm)

            vb2 = vb[0] ;

         else if((TR2b>NDB[1])&&(TR2b<=NDB[0]))
            vb2 = vb[0] ;

         else if((TR2b>NDB[2])&&(TR2b<=NDB[1]))
            vb2 = vb[1] ;

         else if((TR2b>NDB[3])&&(TR2b<=NDB[2]))
            vb2 = vb[2] ;

         else if((TR2b>NDB[4])&&(TR2b<=NDB[3]))
            vb2 = vb[3] ;

         else if(TR2b<=NDB[4])
            vb2 = vb[4] ;

         cout<<"VEl vb2 for less tel is ="<<vb2<<endl;

          if(vb2>=vmax)
            vb2 = vmax ;
          else if(vb2<vmax)
            vb2;
           vb2 = vb2*sfb2;
```

```
            cout<<"finally VEl vb2 for less Tel is ="<<vb2<<endl;

         }

       else if((NDBm>=TR2b)&&(TR2b<=0))
         {
             vb2 = vb[4];

            if(vb2>=vmax)
              vb2 = vmax ;
             else if(vb2<vmax)
                vb2; vb2 = vb2*sfb2;

         }

       else
         {
           cout<<"Total Resistance is very high "<<endl
               <<"Tractor can't perform in this condition"<<endl;
         }

  }

        tb2 = (db2/vb2)*60 ;

        cout<<"Time for section Bb in min. is = "<<tb2<<endl;

//BACKWARD MOVEMENT
// SECTION A

       PR1b = Gpb*RR1b ;

         NDBm = NDB[0]*((100-DF)/100) ;

        cout<<"NDBm is ="<<NDBm<<endl;


  if(TEl>=NDBm)
       {
         TR1b = GR1b*(Gc + Gpb) + PR1b ;

         cout<<"TR1b is ="<<TR1b<<endl;

         if((NDBm>=TR1b)&&(TR1b>0))
           {

              if(TR1b==NDBm)

               vb1 = vb[0] ;

            else if((TR1b>NDB[1])&&(TR1b<=NDB[0]))
               vb1 = vb[0] ;

            else if((TR1b>NDB[2])&&(TR1b<=NDB[1]))
               vb1 = vb[1] ;
```

```
            else if((TR1b>NDB[3])&&(TR1b<=NDB[2]))
               vb1 = vb[2] ;

            else if((TR1b>NDB[4])&&(TR1b<=NDB[3]))
               vb1 = vb[3] ;

            else if(TR1b<=NDB[4])
               vb1 = vb[4] ;

          cout<<"VEl vb1 is ="<<vb1<<endl;

            if(vb1>=vmax)
               vb1 = vmax ;
            else if(vb1<vmax)
               vb1;
             vb1 = vb1*sfb1;

         cout<<"finally VEl vb1 is ="<<vb1<<endl;

      }

    else if((NDBm>=TR1b)&&(TR1b<=0))
      {
          vb1 = vb[4];

         if(vb1>=vmax)
             vb1 = vmax ;
          else if(vb1<vmax)
             vb1; vb1 = vb1*sfb1;

      }

    else
      {
        cout<<"Total Resistance is very high "<<endl
           <<"Tractor can't perform in this condition"<<endl;
      }

   }

 else if(TEl<NDBm)
      {
       NDBm = TEl ;

       cout<<"NDBm for less TEl is ="<<NDBm<<endl;

       TR1b = GR1b*(Gc + Gpb) + PR1b ;

       cout<<"TR1b for less TEl is ="<<TR1b<<endl;

       if((NDBm>=TR1b)&&(TR1b>0))
          {
           if(TR1b==NDBm)
```

```
vb1 = vb[0] ;

else if((TR1b>NDB[1])&&(TR1b<=NDB[0]))
   vb1 = vb[0] ;

else if((TR1b>NDB[2])&&(TR1b<=NDB[1]))
   vb1 = vb[1] ;

else if((TR2>NDB[3])&&(TR1b<=NDB[2]))
   vb1 = vb[2] ;

else if((TR2>NDB[4])&&(TR1b<=NDB[3]))
   vb1 = vb[3] ;

else if(TR1b<=NDB[4])
   vb1 = vb[4] ;

cout<<"VEl vb1 for less tel is ="<<vb1<<endl;

 if(vb1>=vmax)
   vb1 = vmax ;
 else if(vb1<vmax)
   vb1;
  vb1 = vb1*sfb1;

cout<<"finally VEl vb1 for less Tel is ="<<vb1<<endl;

}

else if((NDBm>=TR1b)&&(TR1b<=0))
 {
     vb1 = vb[4];

    if(vb1>=vmax)
        vb1 = vmax ;
     else if(vb1<vmax)
        vb1; vb1 = vb1*sfb1;

 }

else
 {
   cout<<"Total Resistance is very high "<<endl
      <<"Tractor can't perform in this condition"<<endl;
 }

}

 tb1 = (db1/vb1)*60 ;

 cout<<"Time for section Ab in min. is  = "<<tb1<<endl;

getch();

// TOTAL CYCLE TIME :
```

```
float T,CT,CTc ;

float Tps,q;

T = tf1+tf2+tb1+tb2 ;

cout<<"travel time is ="<<T<<endl;
fprintf(f2,"the value of travel time is T=");
fprintf(f2,"%0.3f\n",T);

CT = T + D - t11;

  float CTC[7],TPS[7],P[7];
  float d;
  int n=7;

   d = Lt[1]-Lt[0]  ;

    for(i=0;i<n;i++)
   {
   CTC[i] = Lt[i] + CT  ;
   TPS[i] = 50/CTC[i] ;

   P[i] = TPS[i] * LPT[i]  ;

   cout<<" HPR["<<i<<"] = "<<P[i]<<"   " ;
   cout<<" Lt["<<i<<"] = "<<Lt[i]<<"   ";
   cout<<" LPT["<<i<<"] = "<<LPT[i]<<endl;
   }
   getch();

  float Pm,t,tl,th,Lm;

  Pm = P[0] ;

  for(i=0;i<n;i++)
  {

   if(P[i]>Pm)
    Pm = P[i] ;
   if(Pm == P[i])
   {
    t=i;tl=i-1;
     if(i+1==n)
     th = i ;
     else
     { th = i+1 ; }
     Lm = LPT[i];
    }
  }
  cout<<"optimum time,t0 is"<<" "<<Lt[t]<<"  "
     <<" time1 is"<<" "<<Lt[tl]<<"  "
      <<"time2 is"<<Lt[th]<<endl;
   cout<<"P0 is = "<<Pm<<endl;
```

```
   cout<<"P1 is = "<<P[tl]<<endl;
   cout<<"P2 is = "<<P[th]<<endl;
   cout<<"optimum load ,LPT is = "<<Lm<<endl;
   cout<<"L1 is = "<<LPT[tl]<<endl;
   cout<<"L2 is = "<<LPT[th]<<endl;


  float  g1 = (Pm-P[tl]);
  float  d11 = (Lm - LPT[tl]);
  float  g2 = (P[tl] + P[th] -2* Pm);
  float  d22 = (LPT[tl] + LPT[th] -2*Lm);

  float to = Lt[tl] + 0.5*(d) - (g1*d)/g2 ;


  if(to>Lt[th])
  to = Lt[th];
  else
  to ;
  cout<<"Exact value of optimum loading time is = "<<to<<endl;
  getch();

   float p = (to- Lt[tl])/d ;

   float Lo = LPT[tl] + p*d11 + (p*(p-1)*d22)/2 ;

  cout<<"Exact value of optimum load is = "<<Lo<<endl;

  getch();

 T =  t12 + tf2 + tb1 + tb2  ;

 CTc = to + T + D;

fprintf(f2,"the value of total cycle time CTc is =  ");
fprintf(f2,"%0.3f\n",CTc);

CT = CTc/eff ;
Tps = 60/CT ;
q =   Lo*Lf*Tps  ;

fprintf(f2,"the value of productivity q  is = ");
fprintf(f2,"%0.3f\n",q);

float S1,Sr;

S1 = q1/q ;

 int Srr = int(S1) ;

  Sr = Srr + 1 ;

fprintf(f2,"No of scrapers required is Sr = ");
fprintf(f2,"%0.3f\n",Sr);
```

```
      getch();

//  PUSH LOADING BY PUSHER
//  -----------------------

      float PCT,Srp,Pu,PU,SrCIN,FCIN;
      int bt;

      cout<<"Type of loading "<<endl
         <<"Use 0(zero) for back track loading"<<endl
         <<"Use digit other than 0(zero) for chain loading"<<endl
         <<"give digit = ";
      cin>>bt;

      if( bt == 0)

        PCT = 1.5*to ;
      else

        PCT = 1.3*to ;

     cout<<"PCT is = "<<PCT<<endl;

      Srp  = CTc/PCT ;

      Pu = Sr/Srp ;

      int pu = int(Pu) ;

      PU = pu + 1 ;


    fprintf(f2,"No of pushers required is PU = ");
    fprintf(f2,"%0.3f\n",PU);

      SrCIN =  Osr/q ;

    fprintf(f2,"Economic cost index of Scraper is =");
    fprintf(f2,"%0.3f\n",SrCIN);

      FCIN = ((PU*Opu) + (Sr*Osr))/(q*Sr)  ;

    fprintf(f2,"Economic cost index of fleet is =");
    fprintf(f2,"%0.3f\n",FCIN);

  getch();

}
```

APPENDIX - D

```
/*  *****************************************************************

        PROGRAM OF  SCRAPER (TOWED) OPTIMISATION MODEL

                    (Without  L.G.C)

  *****************************************************************
 */

    #include<iostream.h>
    #include<conio.h>
    #include<math.h>
    #include<stdio.h>


    void main()

     {
     FILE*f1;
     FILE*f2;

/*************************************/

        float HP,e,L,ct,RR1,RR2,GR1,GR2;
        float RR1b,RR2b,GR1b,GR2b;
        float Gpf,Gpb,Gc,Ge;
        float vmax,fdl,fde;
        float k,Hc,Hcg,Hcc;
        float v[10],vf[10],vb[10],DB[10],NDB[10],Ltg;
        float sff1,sff2,sfb1,sfb2;
        float hrs,q1,qy;
        float Osr,Opu;
        float NDBm,Rc,NRRC,D,eff,Lf;
        float PR1,PR2,TR1,TR2,TR1b,TR2b,TE1,PR1b,PR2b;
        float vf1,vf2,vb1,vb2,tf1,tf2,tb1,tb2,df1,df2,db1,db2;
        int i;
        float DF,h;

/*************************************/

clrscr();

f1=fopen("src2.dat","r");
f2=fopen("src2.res","w");
fscanf(f1,"%f%f%f%f%f%f%f",&HP,&e,&RR1,&RR2,&Rc,&GR1,&GR2);
fscanf(f1,"%f%f%f%f",&RR1b,&RR2b,&GR1b,&GR2b);
fscanf(f1,"%f%f%f%f%f%f%f%f",&L,&Gc,&Ge,&ct,&df1,&df2,&db1,&db2);
fscanf(f1,"%f%f%f%f%f%f",&fdl,&fde,&k,&Hcg,&sff1,&sff2);
fscanf(f1,"%f%f%f%f%f%f%f",&sfb1,&sfb2,&hrs,&qy,&Osr,&Opu,&vmax);
fscanf(f1,"%f%f%f%f%f",&D,&eff,&Lf,&Ltg,&h);

for(i=0;i<5;i++) fscanf(f1,"%f",&vf[i]);
for(i=0;i<5;i++) fscanf(f1,"%f",&vb[i]);
for(i=0;i<5;i++) fscanf(f1,"%f",&DB[i]);
```

```
fclose(f1);
//fclose(f2);

        Hcc = (L*1000)/k ;

        if(Hcc>=Hcg)

        Hc = Hcg ;

        else

        Hc = Hcc ;

        cout<<"Adopted Heaped capacity of m/c is = "<<Hc<<endl;


        q1 = qy/hrs  ;

        if(h>900)

        DF = 3*((h-900)/300);
        else
        DF = 0;

        HP = HP*((100-DF)/100) ;

        cout<<"Available power is = "<<HP<<endl;


       fprintf(f2,"RESULTS OBTAINED FOR GIVEN CONDITION \n");
       fprintf(f2,"**************************************\n");

       fprintf(f2,"The value of required production, q1 is = ");
       fprintf(f2,"%0.3f\n",q1);
//*********************************************************************

//*********************************************************************
        NRRC = (Rc - 55)*Gc ;

        cout<<"NRRC is = "<<NRRC<<endl;

        for(i=0;i<5;i++)
         {
          NDB[i] = ((DB[i] - NRRC)*.85) ;
         cout<<"NDB[i] is = "<<NDB[i];
         }
        NDBm = NDB[0]*((100-DF)/100) ;

        cout<<"NDBm is ="<<NDBm<<endl;


        Gpf = L + Ge ;
```

```
PR1 = Gpf*RR1 ;

cout<<"PR1 is ="<<PR1<<endl;

TEl = ct*Gc*1000 ;

cout<<"TEl is ="<<TEl<<endl;

if(TEl>=NDBm)
    {
     TR1 = GR1*(Gc + Gpf) + PR1 ;

     cout<<"TR1 is ="<<TR1<<endl;

     if((NDBm>=TR1)&&(TR1>0))
         {

            if(TR1==NDBm)

             vf1 = vf[0] ;

         else if((TR1>NDB[1])&&(TR1<=NDB[0]))
             vf1 = vf[0] ;

         else if((TR1>NDB[2])&&(TR1<=NDB[1]))
             vf1 = vf[1] ;

         else if((TR1>NDB[3])&&(TR1<=NDB[2]))
             vf1 = vf[2] ;

         else if((TR1>NDB[4])&&(TR1<=NDB[3]))
             vf1 = vf[3] ;

         else if(TR1<=NDB[4])
             vf1 = vf[4] ;

        cout<<"VEl vf1 is ="<<vf1<<endl;

           if(vf1>=vmax)
             vf1 = vmax ;
           else if(vf1<vmax)
             vf1;
             vf1 = vf1*sff1;
        cout<<"finally VEl vf1 is ="<<vf1<<endl;

      }

   else if((NDBm>=TR1)&&(TR1<=0))
      {
         vf1 = vf[4];

        if(vf1>=vmax)
             vf1 = vmax ;
         else if(vf1<vmax)
```

```
            vf1; vf1 = vf1*sff

        }

    else
        {
          cout<<"Total Resistance is very high "<<endl
             <<"Tractor can't perform in this condition"<<endl;
        }

    }

else if(TEl<NDBm)
     {
       NDBm = TEl ;

      cout<<"NDBm for less TEl is ="<<NDBm<<endl;

       TR1 = GR1*(Gc + Gpf) + PR1 ;

      cout<<"TR1 for less TEl is ="<<TR1<<endl;

       if((NDBm>=TR1)&&(TR1>0))
          {
           if(TR1==NDBm)

              vf1 = vf[0] ;

           else if((TR1>NDB[1])&&(TR1<=NDB[0]))
              vf1 = vf[0] ;

           else if((TR1>NDB[2])&&(TR1<=NDB[1]))
              vf1 = vf[1] ;

           else if((TR1>NDB[3])&&(TR1<=NDB[2]))
              vf1 = vf[2] ;

           else if((TR1>NDB[4])&&(TR1<=NDB[3]))
              vf1 = vf[3] ;

           else if(TR1<=NDB[4])
              vf1 = vf[4] ;

          cout<<"VEl vf1 for less tel is ="<<vf1<<endl;

           if(vf1>=vmax)
              vf1 = vmax ;
           else if(vf1<vmax)
              vf1;
            vf1 = vf1*sff1;
        cout<<"finally VEl vf1 for less Tel is ="<<vf1<<endl;

       }

     else if((NDBm>=TR1)&&(TR1<=0))
```

```
        {
            vf1 = vf[4];

            if(vf1>=vmax)
                vf1 = vmax ;
            else if(vf1<vmax)
                vf1; vf1 = vf1*sff
        }

    else
        {
          cout<<"Total Resistance is very high "<<endl
              <<"Tractor can't perform in this condition"<<endl;
        }

  }

     float Lt = (Ltg*Hc)/Hcg  ;

     float  t11 = Lt;

      cout<<"finally time for section t11 is ="<<t11<<endl;

    float  t12 = ((df1 - 0.1)/vf1)*60 ;
          cout<<"time for section t12 is =";
          cout<<t12<<endl;

          tf1 = t11 + t12;

          cout<<"time for section A is =";
          cout<<tf1<<endl;


// ONWARD MOVEMENT ;
// SECTION B :

       NDBm = NDB[0]*((100-DF)/100) ;
       cout<<"NDBm is ="<<NDBm<<endl;
       PR2 = Gpf*RR2 ;


  if(TEl>=NDBm)
       {
        TR2 = GR2*(Gc + Gpf) + PR2 ;

        cout<<"TR2 is ="<<TR2<<endl;

        if((NDBm>=TR2)&&(TR2>0))
           {

              if(TR2==NDBm)

               vf2 = vf[0] ;

            else if((TR2>NDB[1])&&(TR2<=NDB[0]))
```

```
            vf2 = vf[0] ;

          else if((TR2>NDB[2])&&(TR2<=NDB[1]))
            vf2 = vf[1] ;

          else if((TR2>NDB[3])&&(TR2<=NDB[2]))
            vf2 = vf[2] ;

          else if((TR2>NDB[4])&&(TR2<=NDB[3]))
            vf2 = vf[3] ;

          else if(TR2<=NDB[4])
            vf2 = vf[4] ;

        cout<<"VEl vf2 is ="<<vf2<<endl;

          if(vf2>=vmax)
            vf2 = vmax ;
          else if(vf2<vmax)
            vf2;
          vf2 = vf2*sff2;
        cout<<"finally VEl vf2 is ="<<vf2<<endl;

      }

    else if((NDBm>=TR2)&&(TR2<=0))
      {
        vf2 = vf[4];

        if(vf2>=vmax)
            vf2 = vmax ;
         else if(vf2<vmax)
            vf2; vf2 = vf2*sff2;

      }

    else
      {
        cout<<"Total Resistance is very high "<<endl
          <<"Tractor can't perform in this condition"<<endl;
      }

  }

else if(TEl<NDBm)
    {
     NDBm = TEl ;

     cout<<"NDBm for less TEl is ="<<NDBm<<endl;

     TR2 = GR2*(Gc + Gpf) + PR2 ;

     cout<<"TR1 for less TEl is ="<<TR2<<endl;

     if((NDBm>=TR2)&&(TR2>0))
```

```
{
 if(TR2==NDBm)

    vf2 = vf[0] ;

 else if((TR2>NDB[1])&&(TR2<=NDB[0]))
    vf2 = vf[0] ;

 else if((TR2>NDB[2])&&(TR2<=NDB[1]))
    vf2 = vf[1] ;

 else if((TR2>NDB[3])&&(TR2<=NDB[2]))
    vf2 = vf[2] ;

 else if((TR2>NDB[4])&&(TR2<=NDB[3]))
    vf2 = vf[3] ;

 else if(TR2<=NDB[4])
    vf2 = vf[4] ;

 cout<<"VEl vf1 for less tel is ="<<vf2<<endl;

  if(vf2>=vmax)
    vf2 = vmax ;
  else if(vf2<vmax)
    vf2;
  vf2 = vf2*sff2;
cout<<"finally VEl vf1 for less Tel is ="<<vf2<<endl;

}

else if((NDBm>=TR2)&&(TR2<=0))
 {
    vf2 = vf[4];

   if(vf2>=vmax)
       vf2 = vmax ;
    else if(vf2<vmax)
       vf2; vf2 = vf2*sff2;

 }

else
 {
   cout<<"Total Resistance is very high "<<endl
      <<"Tractor can't perform in this condition"<<endl;
 }

}

    tf2 = (df2/vf2)*60 ;

    cout<<"Time for section 2 in min. is tf2 = "<<tf2<<endl;

// BACKWARD JOURNEY
```

// SECTION B

```
Gpb = Ge ;

PR2b = Gpb*RR2b ;

NDBm = NDB[0]*((100-DF)/100) ;

cout<<"NDBm is ="<<NDBm<<endl;

if(TEl>=NDBm)
    {
      TR2b = GR2b*(Gc + Gpb) + PR2b ;

      cout<<"TR2b is ="<<TR2b<<endl;

      if((NDBm>=TR2b)&&(TR2b>0))
        {

          if(TR2b==NDBm)

            vb2 = vb[0] ;

          else if((TR2b>NDB[1])&&(TR2b<=NDB[0]))
            vb2 = vb[0] ;

          else if((TR2b>NDB[2])&&(TR2b<=NDB[1]))
            vb2 = vb[1] ;

          else if((TR2b>NDB[3])&&(TR2b<=NDB[2]))
            vb2 = vb[2] ;

          else if((TR2b>NDB[4])&&(TR2b<=NDB[3]))
            vb2 = vb[3] ;

          else if(TR2b<=NDB[4])
            vb2 = vb[4] ;

        cout<<"VEl vb2 is ="<<vb2<<endl;

          if(vb2>=vmax)
            vb2 = vmax ;
          else if(vb2<vmax)
            vb2;
           vb2 = vb2*sfb2;

        cout<<"finally VEl vb2 is ="<<vb2<<endl;

      }

    else if((NDBm>=TR2b)&&(TR2b<=0))
      {
```

```
            vb2 = vb[4];

           if(vb2>=vmax)
               vb2 = vmax ;
            else if(vb2<vmax)
               vb2; vb2 = vb2*sfb2;

        }

      else
        {
          cout<<"Total Resistance is very high "<<endl
             <<"Tractor can't perform in this condition"<<endl;
        }

    }

else if(TEl<NDBm)
     {
       NDBm = TEl ;

      cout<<"NDBm for less TEl is ="<<NDBm<<endl;

       TR2b = GR2b*(Gc + Gpb) + PR2b ;

      cout<<"TR2 for less TEl is ="<<TR2b<<endl;

       if((NDBm>=TR2b)&&(TR2b>0))
           {
            if(TR2b==NDBm)

               vb2 = vb[0] ;

            else if((TR2b>NDB[1])&&(TR2b<=NDB[0]))
               vb2 = vb[0] ;

            else if((TR2b>NDB[2])&&(TR2b<=NDB[1]))
               vb2 = vb[1] ;

            else if((TR2b>NDB[3])&&(TR2b<=NDB[2]))
               vb2 = vb[2] ;

            else if((TR2b>NDB[4])&&(TR2b<=NDB[3]))
               vb2 = vb[3] ;

            else if(TR2b<=NDB[4])
               vb2 = vb[4] ;

           cout<<"VEl vb2 for less tel is ="<<vb2<<endl;

             if(vb2>=vmax)
               vb2 = vmax ;
             else if(vb2<vmax)
               vb2;
              vb2 = vb2*sfb2;
```

```
        cout<<"finally VEl vb2 for less Tel is ="<<vb2<<endl;

      }

    else if((NDBm>=TR2b)&&(TR2b<=0))
      {
         vb2 = vb[4];

        if(vb2>=vmax)
           vb2 = vmax ;
         else if(vb2<vmax)
            vb2; vb2 = vb2*sfb2;

      }

    else
      {
        cout<<"Total Resistance is very high "<<endl
           <<"Tractor can't perform in this condition"<<endl;
      }

 }

     tb2 = (db2/vb2)*60 ;

     cout<<"Time for section 2 in min. is tb2 = "<<tb2<<endl;

//BACKWARD MOVEMENT
// SECTION A

    PR1b = Gpb*RR1b ;

      NDBm = NDB[0]*((100-DF)/100) ;

     cout<<"NDBm is ="<<NDBm<<endl;

  if(TEl>=NDBm)
     {
       TR1b = GR1b*(Gc + Gpb) + PR1b ;

       cout<<"TR1b is ="<<TR1b<<endl;

       if((NDBm>=TR1b)&&(TR1b>0))
          {

            if(TR1b==NDBm)

             vb1 = vb[0] ;

          else if((TR1b>NDB[1])&&(TR1b<=NDB[0]))
             vb1 = vb[0] ;
```

```
            else if((TR1b>NDB[2])&&(TR1b<=NDB[1]))
               vb1 = vb[1] ;

            else if((TR1b>NDB[3])&&(TR1b<=NDB[2]))
               vb1 = vb[2] ;

            else if((TR1b>NDB[4])&&(TR1b<=NDB[3]))
               vb1 = vb[3] ;

            else if(TR1b<=NDB[4])
               vb1 = vb[4] ;

          cout<<"VEl vb1 is ="<<vb1<<endl;

            if(vb1>=vmax)
               vb1 = vmax ;
            else if(vb1<vmax)
               vb1;
             vb1 = vb1*sfb1;

          cout<<"finally VEl vb1 is ="<<vb1<<endl;

        }

      else if((NDBm>=TR1b)&&(TR1b<=0))
        {
           vb1 = vb[4];

          if(vb1>=vmax)
               vb1 = vmax ;
           else if(vb1<vmax)
               vb1; vb1 = vb1*sfb1;

        }

      else
        {
          cout<<"Total Resistance is very high "<<endl
             <<"Tractor can't perform in this condition"<<endl;
        }

    }

else if(TEl<NDBm)
     {
       NDBm = TEl ;

      cout<<"NDBm for less TEl is ="<<NDBm<<endl;

       TR1b = GR1b*(Gc + Gpb) + PR1b ;

      cout<<"TR1b for less TEl is ="<<TR1b<<endl;

       if((NDBm>=TR1b)&&(TR1b>0))
          {
```

```
if(TR1b==NDBm)

    vb1 = vb[0] ;

else if((TR1b>NDB[1])&&(TR1b<=NDB[0]))
    vb1 = vb[0] ;

else if((TR1b>NDB[2])&&(TR1b<=NDB[1]))
    vb1 = vb[1] ;

else if((TR2>NDB[3])&&(TR1b<=NDB[2]))
    vb1 = vb[2] ;

else if((TR2>NDB[4])&&(TR1b<=NDB[3]))
    vb1 = vb[3] ;

else if(TR1b<=NDB[4])
    vb1 = vb[4] ;

cout<<"VEl vb1 for less tel is ="<<vb1<<endl;

if(vb1>=vmax)
    vb1 = vmax ;
  else if(vb1<vmax)
    vb1;
  vb1 = vb1*sfb1;

cout<<"finally VEl vb1 for less Tel is ="<<vb1<<endl;

}

else if((NDBm>=TR1b)&&(TR1b<=0))
  {
     vb1 = vb[4];

     if(vb1>=vmax)
         vb1 = vmax ;
      else if(vb1<vmax)
         vb1; vb1 = vb1*sfb1;

  }

else
  {
   cout<<"Total Resistance is very high "<<endl
      <<"Tractor can't perform in this condition"<<endl;
  }

}

  tb1 = (db1/vb1)*60 ;

  cout<<"Time for section 2 in min. is tb1 = "<<tb1<<endl;
```

```
getch();


// TOTAL CYCLE TIME :

    float T,CT,CTc ;


    float Tps,q;

    T = tf1+tf2+tb1+tb2 ;

    cout<<"travel time is ="<<T<<endl;
    fprintf(f2,"the value of travel time is T=");
    fprintf(f2,"%0.3f\n",T);


    CT = T + D;



    CTc = CT;

    fprintf(f2,"the value of total cycle time CTc is =  ");
    fprintf(f2,"%0.3f\n",CTc);


    CT = CTc/eff ;
    Tps = 60/CT ;
    q =  Hc*Lf*Tps  ;


    fprintf(f2,"the value of productivity q  is = ");
    fprintf(f2,"%0.3f\n",q);


    float S1,Sr;


    S1 = q1/q ;

     int Srr = int(S1) ;

      Sr = Srr + 1 ;


    fprintf(f2,"No of scrapers required is Sr = ");
    fprintf(f2,"%0.3f\n",Sr);


      getch();

 //  PUSH LOADING BY PUSHER
 //  -----------------------
```

```
float PCT,Srp,Pu,PU,SrCIN,FCIN;
int bt;

cout<<"Type of loading "<<endl
   <<"Use 0(zero) for back track loading"<<endl
   <<"Use digit other than 0(zero) for chain loading"<<endl
   <<"give digit = ";
cin>>bt;

if( bt == 0)

  PCT = 1.5*Lt ;
else

  PCT = 1.3*Lt ;

cout<<"PCT is = "<<PCT<<endl;

 Srp  = CTc/PCT ;

 Pu = Sr/Srp ;

 int pu = int(Pu) ;

 PU = pu + 1 ;


fprintf(f2,"No of pushers required is PU = ");
fprintf(f2,"%0.3f\n",PU);

 SrCIN =  Osr/q ;

fprintf(f2,"Economic cost index of Scraper is =");
fprintf(f2,"%0.3f\n",SrCIN);


 FCIN = ((PU*Opu) + (Sr*Osr))/(q*Sr)  ;


fprintf(f2,"Economic cost index of fleet is =");
fprintf(f2,"%0.3f\n",FCIN);


getch();

}
```

APPENDIX - E

```
/*  *******************************************************************

         PROGRAM  OF  BULLDOZER  OPTIMISATION  MODEL


   *******************************************************************
*/

    #include<iostream.h>
    #include<conio.h>
    #include<math.h>
    #include<stdio.h>


    void main()

     {
     FILE*f1;
     FILE*f2;

/**************************************/

        float HP,e,L,ct,RR1,RR2,GR1,GR2;
        float RR1b,RR2b,GR1b,GR2b;
        float Gpf,Gpb,Gc;
        float vmax;
        float k,Bc;
        float v[10],vf[10],vb[10],DB[10],NDB[10];

        float hrs,q1,qy;
        float Odz,f;
        float NDBm,Rc,NRRC,D,eff,Lf;
        float PR1,PR2,TR1,TR2,TR1b,TR2b,TE1,PR1b,PR2b;
        float vf1,vf2,vb1,vb2,tf1,tf2,tb1,tb2,df1,df2,db1,db2;
        int i;
        float DF,h;
/**************************************/

clrscr();

f1=fopen("bdoz1.dat","r");
f2=fopen("bdoz1.res","w");
fscanf(f1,"%f%f%f%f%f%f%f",&HP,&e,&RR1,&RR2,&Rc,&GR1,&GR2);
fscanf(f1,"%f%f%f%f",&RR1b,&RR2b,&GR1b,&GR2b);
fscanf(f1,"%f%f%f%f%f%f",&Gc,&ct,&df1,&df2,&db1,&db2);
fscanf(f1,"%f%f",&k,&Bc);
fscanf(f1,"%f%f%f%f",&hrs,&qy,&Odz,&vmax);
fscanf(f1,"%f%f%f%f%f",&D,&eff,&Lf,&f,&h);

for(i=0;i<5;i++) fscanf(f1,"%f",&vf[i]);
for(i=0;i<5;i++) fscanf(f1,"%f",&vb[i]);
for(i=0;i<5;i++) fscanf(f1,"%f",&DB[i]);


fclose(f1);
```

```
//fclose(f2);

      q1 = qy/hrs  ;

      if(h>900)

       DF = 3*((h-900)/300);
       else
       DF = 0;

       HP = HP*((100-DF)/100) ;

       cout<<"Available power is = "<<HP<<endl;


     fprintf(f2,"RESULTS OBTAINED FOR GIVEN CONDITION \n");
     fprintf(f2,"************************************\n");

     fprintf(f2,"The value of required production, q1 is = ");
     fprintf(f2,"%0.3f\n",q1);
//*********************************************************************

//*********************************************************************
        NRRC = (Rc - 55)*Gc ;

        cout<<"NRRC is = "<<NRRC<<endl;

        for(i=0;i<5;i++)
         {
           NDB[i] = ((DB[i] - NRRC)*.85) ;
         cout<<"NDB[i] is = "<<NDB[i];
          }
        NDBm = NDB[0]*((100-DF)/100) ;

        cout<<"NDBm is ="<<NDBm<<endl;

        Gpf = 0.0 ;

        PR1 = k*Bc*f ;


       cout<<"PR1 is ="<<PR1<<endl;

        TE1 = ct*Gc*1000 ;

        cout<<"TE1 is ="<<TE1<<endl;

     if(TE1>=NDBm)
         {
            TR1 = GR1*(Gc + Gpf) + PR1 ;

            cout<<"TR1 is ="<<TR1<<endl;

            if((NDBm>=TR1)&&(TR1>0))
            {
```

```
            if(TR1==NDBm)

             vf1 = vf[0]  ;

          else if((TR1>NDB[1])&&(TR1<=NDB[0]))
              vf1 = vf[0]  ;

          else if((TR1>NDB[2])&&(TR1<=NDB[1]))
              vf1 = vf[1]  ;

          else if((TR1>NDB[3])&&(TR1<=NDB[2]))
              vf1 = vf[2]  ;

          else if((TR1>NDB[4])&&(TR1<=NDB[3]))
              vf1 = vf[3]  ;

          else if(TR1<=NDB[4])
              vf1 = vf[4]  ;

        cout<<"VE1 vf1 is ="<<vf1<<endl;

           if(vf1>=vmax)
              vf1 = vmax ;
           else
              vf1;
        cout<<"finally VE1 vf1 is ="<<vf1<<endl;

        }

     else if((NDBm>=TR1)&&(TR1<=0))
        {
         vf1 = vf[4];

        if(vf1>=vmax)
              vf1 = vmax ;
         else
              vf1;

        }

    else
        {
       cout<<"Total Resistance is very high "<<endl
           <<"Tractor can't perform in this condition"<<endl;
       }

 }

 else if(TE1<NDBm)
    {
       NDBm = TE1 ;

      cout<<"NDBm for less TE1 is ="<<NDBm<<endl;

       TR1 = GR1*(Gc + Gpf) + PR1 ;

      cout<<"TR1 for less TE1 is ="<<TR1<<endl;
```

```
        if((NDBm>=TR1)&&(TR1>0))
        {
          if(TR1==NDBm)

              vf1 = vf[0] ;

          else if((TR1>NDB[1])&&(TR1<=NDB[0]))
              vf1 = vf[0] ;

          else if((TR1>NDB[2])&&(TR1<=NDB[1]))
              vf1 = vf[1] ;

          else if((TR1>NDB[3])&&(TR1<=NDB[2]))
              vf1 = vf[2] ;

          else if((TR1>NDB[4])&&(TR1<=NDB[3]))
              vf1 = vf[3] ;

          else if(TR1<=NDB[4])
              vf1 = vf[4] ;

         cout<<"VEl vf1 for less tel is ="<<vf1<<endl;

           if(vf1>=vmax)
              vf1 = vmax ;
           else
              vf1;

       cout<<"finally VEl vf1 for less Tel is ="<<vf1<<endl;

        }

     else if((NDBm>=TR1)&&(TR1<=0))
        {
         vf1 = vf[4];

        if(vf1>=vmax)
              vf1 = vmax ;
         else
              vf1;
        }

   else
     {
     cout<<"Total Resistance is very high "<<endl
        <<"Tractor can't perform in this condition"<<endl;
     }

   }

   tf1 = (df1/vf1)*60 ;

   cout<<"Time for section Af in min. is  = "<<tf1<<endl;

// ONWARD MOVEMENT ;
// SECTION B :
```

```
        NDBm = NDB[0]*((100-DF)/100) ;
       cout<<"NDBm is ="<<NDBm<<endl;
        PR2 = k*Bc*f ;

if(TE1>=NDBm)
    {
        TR2 = GR2*(Gc + Gpf) + PR2 ;

        cout<<"TR2 is ="<<TR2<<endl;

        if((NDBm>=TR2)&&(TR2>0))
        {

             if(TR2==NDBm)

              vf2 = vf[0] ;

          else if((TR2>NDB[1])&&(TR2<=NDB[0]))
              vf2 = vf[0] ;

          else if((TR2>NDB[2])&&(TR2<=NDB[1]))
              vf2 = vf[1] ;

          else if((TR2>NDB[3])&&(TR2<=NDB[2]))
              vf2 = vf[2] ;

          else if((TR2>NDB[4])&&(TR2<=NDB[3]))
              vf2 = vf[3] ;

          else if(TR2<=NDB[4])
              vf2 = vf[4] ;

        cout<<"VE1 vf2 is ="<<vf2<<endl;

           if(vf2>=vmax)
              vf2 = vmax ;
           else
              vf2;

      cout<<"finally VE1 vf2 is ="<<vf2<<endl;

        }

    else if((NDBm>=TR2)&&(TR2<=0))
        {
         vf2 = vf[4];

        if(vf2>=vmax)
              vf2 = vmax ;
         else
              vf2;
        }

else
 {
  cout<<"Total Resistance is very high "<<endl
```

```
        <<"Tractor can't perform in this condition"<<endl;
  }

}

else if(TEl<NDBm)
   {
      NDBm = TEl ;

     cout<<"NDBm for less TEl is ="<<NDBm<<endl;

      TR2 = GR2*(Gc + Gpf) + PR2 ;

     cout<<"TR1 for less TEl is ="<<TR2<<endl;

      if((NDBm>=TR2)&&(TR2>0))
      {
        if(TR2==NDBm)

             vf2 = vf[0] ;

        else if((TR2>NDB[1])&&(TR2<=NDB[0]))
             vf2 = vf[0] ;

        else if((TR2>NDB[2])&&(TR2<=NDB[1]))
             vf2 = vf[1] ;

        else if((TR2>NDB[3])&&(TR2<=NDB[2]))
             vf2 = vf[2] ;

        else if((TR2>NDB[4])&&(TR2<=NDB[3]))
             vf2 = vf[3] ;

        else if(TR2<=NDB[4])
             vf2 = vf[4] ;

  cout<<"VEl vf1 for less tel is ="<<vf2<<endl;

         if(vf2>=vmax)
             vf2 = vmax ;
         else
             vf2;

cout<<"finally VEl vf1 for less Tel is ="<<vf2<<endl;

      }

    else if((NDBm>=TR2)&&(TR2<=0))
      {
       vf2 = vf[4];

      if(vf2>=vmax)
            vf2 = vmax ;
       else
            vf2;
      }
```

```
     else
       {
        cout<<"Total Resistance is very high "<<endl
          <<"Tractor can't perform in this condition"<<endl;
       }

     }

       tf2 = (df2/vf2)*60 ;

      cout<<"Time for section Bf in min. is = "<<tf2<<endl;

// BACKWARD JOURNEY
// SECTION B

       Gpb = 0.0 ;

       PR2b = 0.0 ;

          NDBm = NDB[0]*((100-DF)/100) ;

         cout<<"NDBm is ="<<NDBm<<endl;

    if(TE1>=NDBm)
        {
           TR2b = GR2b*(Gc + Gpb) + PR2b ;

           cout<<"TR2b is ="<<TR2b<<endl;

           if((NDBm>=TR2b)&&(TR2b>0))
           {

                if(TR2b==NDBm)

                 vb2 = vb[0] ;

             else if((TR2b>NDB[1])&&(TR2b<=NDB[0]))
                 vb2 = vb[0] ;

             else if((TR2b>NDB[2])&&(TR2b<=NDB[1]))
                 vb2 = vb[1] ;

             else if((TR2b>NDB[3])&&(TR2b<=NDB[2]))
                 vb2 = vb[2] ;

             else if((TR2b>NDB[4])&&(TR2b<=NDB[3]))
                 vb2 = vb[3] ;

             else if(TR2b<=NDB[4])
                 vb2 = vb[4] ;

           cout<<"VEl vb2 is ="<<vb2<<endl;

              if(vb2>=vmax)
                 vb2 = vmax ;
              else
                 vb2;
```

```
            vb2 = vmax ;
          else
            vb2;

       cout<<"finally VEl vb2 for less Tel is ="<<vb2<<endl;

        }

      else if((NDBm>=TR2b)&&(TR2b<=0))
        {
         vb2 = vb[4];

        if(vb2>=vmax)
            vb2 = vmax ;
         else
             vb2;
        }

    else
    {
     cout<<"Total Resistance is very high "<<endl
      <<"Tractor can't perform in this condition"<<endl;
    }
  }

     tb2 = (db2/vb2)*60 ;

     cout<<"Time for section Bb in min. is = "<<tb2<<endl;

//BACKWARD MOVEMENT
// SECTION A

       PR1b = 0.0 ;

          NDBm = NDB[0]*((100-DF)/100) ;

         cout<<"NDBm is ="<<NDBm<<endl;


    if(TE1>=NDBm)
        {
          TR1b = GR1b*(Gc + Gpb) + PR1b ;

          cout<<"TR1b is ="<<TR1b<<endl;

          if((NDBm>=TR1b)&&(TR1b>0))
          {

               if(TR1b==NDBm)

                vb1 = vb[0] ;

            else if((TR1b>NDB[1])&&(TR1b<=NDB[0]))
                vb1 = vb[0] ;

            else if((TR1b>NDB[2])&&(TR1b<=NDB[1]))
                vb1 = vb[1] ;
```

```
     cout<<"finally VE1 vb2 is ="<<vb2<<endl;

      }

   else if((NDBm>=TR2b)&&(TR2b<=0))
      {
       vb2 = vb[4];

      if(vb2>=vmax)
            vb2 = vmax ;
       else
            vb2;
      }

else
{
 cout<<"Total Resistance is very high "<<endl
    <<"Tractor can't perform in this condition"<<endl;
}

}

else if(TE1<NDBm)
   {
     NDBm = TE1 ;

     cout<<"NDBm for less TE1 is ="<<NDBm<<endl;

      TR2b = GR2b*(Gc + Gpb) + PR2b ;

     cout<<"TR2 for less TE1 is ="<<TR2b<<endl;

      if((NDBm>=TR2b)&&(TR2b>0))
      {
        if(TR2b==NDBm)

            vb2 = vb[0] ;

        else if((TR2b>NDB[1])&&(TR2b<=NDB[0]))
            vb2 = vb[0] ;

        else if((TR2b>NDB[2])&&(TR2b<=NDB[1]))
            vb2 = vb[1] ;

        else if((TR2b>NDB[3])&&(TR2b<=NDB[2]))
            vb2 = vb[2] ;

        else if((TR2b>NDB[4])&&(TR2b<=NDB[3]))
            vb2 = vb[3] ;

        else if(TR2b<=NDB[4])
            vb2 = vb[4] ;

       cout<<"VE1 vb2 for less tel is ="<<vb2<<endl;

         if(vb2>=vmax)
```

```
        else if((TR1b>NDB[3])&&(TR1b<=NDB[2]))
            vb1 = vb[2] ;

        else if((TR1b>NDB[4])&&(TR1b<=NDB[3]))
            vb1 = vb[3] ;

        else if(TR1b<=NDB[4])
            vb1 = vb[4] ;

      cout<<"VE1 vb1 is ="<<vb1<<endl;

         if(vb1>=vmax)
            vb1 = vmax ;
         else
            vb1;

     cout<<"finally VE1 vb1 is ="<<vb1<<endl;

      }

   else if((NDBm>=TR1b)&&(TR1b<=0))
      {
       vb1 = vb[4];

      if(vb1>=vmax)
            vb1 = vmax ;
       else
        vb1;
      }

else
{
cout<<"Total Resistance is very high "<<endl
 <<"Tractor can't perform in this condition"<<endl;
}

}

 else if(TE1<NDBm)
    {
      NDBm = TE1 ;

     cout<<"NDBm for less TE1 is ="<<NDBm<<endl;

      TR1b = GR1b*(Gc + Gpb) + PR1b ;

     cout<<"TR1b for less TE1 is ="<<TR1b<<endl;

      if((NDBm>=TR1b)&&(TR1b>0))
      {
        if(TR1b==NDBm)

            vb1 = vb[0] ;

        else if((TR1b>NDB[1])&&(TR1b<=NDB[0]))
            vb1 = vb[0] ;
```

```
          else if((TR1b>NDB[2])&&(TR1b<=NDB[1]))
              vb1 = vb[1] ;

          else if((TR2>NDB[3])&&(TR1b<=NDB[2]))
              vb1 = vb[2] ;

          else if((TR2>NDB[4])&&(TR1b<=NDB[3]))
              vb1 = vb[3] ;

          else if(TR1b<=NDB[4])
              vb1 = vb[4] ;

         cout<<"VEl vb1 for less tel is ="<<vb1<<endl;

           if(vb1>=vmax)
              vb1 = vmax ;
           else
              vb1;

   cout<<"finally VEl vb1 for less Tel is ="<<vb1<<endl;

        }

     else if((NDBm>=TR1b)&&(TR1b<=0))
        {
         vb1 = vb[4];

        if(vb1>=vmax)
              vb1 = vmax ;
         else
              vb1;
        }

 else
  {
 cout<<"Total Resistance is very high "<<endl
 <<"Tractor can't perform in this condition"<<endl;
  }

}

       tb1 = (db1/vb1)*60 ;

 cout<<"Time for section Ab in min. is tb1 = "<<tb1<<endl;

  getch();

// TOTAL CYCLE TIME :

 float T,CT,CTc ;

 float Tps,q;

 T = tf1+tf2+tb1+tb2 ;

 cout<<"travel time is ="<<T<<endl;
```

```
fprintf(f2,"the value of travel time is T=");
fprintf(f2,"%0.3f\n",T);

float at = .03*T;
float dt = .01*T;

CT = T + D;

CTc = at + dt + CT;

fprintf(f2,"the value of total cycle time CTc is =  ");
fprintf(f2,"%0.3f\n",CTc);

CT = CTc/eff ;
Tps = 60/CT ;
q =  Bc*Lf*Tps   ;

fprintf(f2,"the value of productivity q  is = ");
fprintf(f2,"%0.3f\n",q);

float S1,Sr;

S1 = q1/q ;

 int Srr = int(S1)  ;

  Sr = Srr + 1 ;

fprintf(f2,"No of bulldozers required is Sr = ");
fprintf(f2,"%0.3f\n",Sr);

  getch();

float  SrCIN =  Odz/q ;

fprintf(f2,"Economic cost index of bulldozer is =");
fprintf(f2,"%0.3f\n",SrCIN);

float  FCIN =  (Sr*Odz)/(q*Sr)   ;

fprintf(f2,"Economic cost index of fleet is =");
fprintf(f2,"%0.3f\n",FCIN);

 getch();

}
```

APPENDIX - F

```
/*      ***************************************************************
         PROGRAM  OF  OPTIMISATION  MODEL  OF  HAUL  UNITS
                  (Deterministic Model)

     ***************************************************************
*/

      #include<iostream.h>
      #include<conio.h>
      #include<math.h>
      #include<stdio.h>

      void main()

       {
       clrscr();
       FILE*f1;
       FILE*f2;
//*****************************************************************


         float HP,e,RR1,RR2,RR3,RR4,RR5,GR1,GR2,GR3,GR4,GR5;
         float RR1b,RR2b,RR3b,RR4b,RR5b,GR1b,GR2b,GR3b,GR4b,GR5b;
         float G,L,E,ct,d1,d2,d3,d4,d5;
         float d1b,d2b,d3b,d4b,d5b,vmax,fdl,fde;
         float ks,Hc,Hcg,Hcc;
         float v[5];
         float sff1,sff2,sff3,sff4,sff5,sfb1,sfb2,sfb3,sfb4,sfb5;
         float hrs,q1,qy;
         float Odc,Ohl;
         float RPm,Lt;
         int i;
         float dc,tl,b,D,M,IC;
         float DF,h,LP;

f1=fopen("shma2.dat","r");
fscanf(f1,"%f%f%f%f%f%f%f",&HP,&e,&RR1,&RR2,&RR3,&RR4,&RR5);
fscanf(f1,"%f%f%f%f%f",&GR1,&GR2,&GR3,&GR4,&GR5);
fscanf(f1,"%f%f%f%f%f",&RR1b,&RR2b,&RR3b,&RR4b,&RR5b);
fscanf(f1,"%f%f%f%f%f",&GR1b,&GR2b,&GR3b,&GR4b,&GR5b);
fscanf(f1,"%f%f%f%f%f%f%f%f",&L,&E,&ct,&d1,&d2,&d3,&d4,&d5);
fscanf(f1,"%f%f%f%f%f",&d1b,&d2b,&d3b,&d4b,&d5b);
fscanf(f1,"%f%f%f%f",&fdl,&fde,&ks,&Hcg);
fscanf(f1,"%f%f%f%f%f",&sff1,&sff2,&sff3,&sff4,&sff5);
fscanf(f1,"%f%f%f%f%f",&sfb1,&sfb2,&sfb3,&sfb4,&sfb5);
fscanf(f1,"%f%f%f%f%f",&hrs,&qy,&Odc,&Ohl,&vmax);
fscanf(f1,"%f%f%f%f%f%f",&tl,&b,&D,&M,&IC,&h);
for(i=0;i<2;i++) fscanf(f1,"%f",&v[i]);
f2=fopen("shma2.res","w");
fclose(f1);
//fclose(f2);

        Hcc = (L*1000)/ks ;

        if(Hcc>=Hcg)
```

```
        Hc = Hcg ;

        else

        Hc = Hcc ;

       cout<<"Adopted Heaped capacity of m/c is = "<<Hc<<endl;


       q1 = qy/hrs  ;

       LP = (b*60)/tl ;

       float Snn = q1/LP ;

       int S1 = int(Snn) ;

       int Sn = S1 + 1 ;

       cout<<"Loaders required = "<<Sn<<endl;


       if(h>900)

       DF = 3*((h-900)/300);
       else
       DF = 0;

       HP = HP*((100-DF)/100) ;

       cout<<"Available power is = "<<HP<<endl;

     fprintf(f2,"*************************************************** \n");
     fprintf(f2,"RESULTS OF MODEL(Deterministic Model)  \n");
     fprintf(f2,"*************************************************** \n");
     fprintf(f2,"Required Hourly Production(based on Peak Production qy)
q1(cum/hr) is\n");
     fprintf(f2,"%0.3f\n",q1);


     RPm = (270*HP*e)/v[0];

   //   ONWARD DIRECTION :

  //    SECTION A :

         float RP1,v1;
         float Tel;
         float Wdl;
         float t1;
        G = E + L ;
        Wdl = fdl*G ;

       Tel = ct*Wdl*1000 ;

       if(Tel>=RPm)
```

```
{
    RP1 = (RR1 + GR1)*G ;

   if((RPm>=RP1)&&(RP1>0))

     {
      v1 = (270*HP*e)/RP1;

     if(v1>=v[1])
     v1 = v[1];
      else
     v1;

      if(v1>=vmax)
     v1 = vmax;
     else
      v1;
      v1 = v1*sff1 ;
     cout<<"finally velocity v1 is = "<<v1<<endl;

    }
  else if((RPm>=RP1)&&(RP1<=0))
    {
      v1 = v[1];
      if(v1>=vmax)
     v1 = vmax;
    else
      v1;
      v1 = v1*sff1 ;

     }
   else
     {
      cout<<"Resistance is high"<<endl
          <<"Take other Model "<<endl;
      }

}

else if (Tel<RPm)
  {
   RPm = Tel;

   RP1 = (RR1 + GR1)*G ;

   if((RPm>=RP1)&&(RP1>0))

    {
      v1 = (270*HP*e)/RP1;

    if(v1>=v[1])
     v1 = v[1];
    else
       v1;

    if(v1>=vmax)
     v1 = vmax;
```

```
     else
       v1;
      v1 = v1*sff1 ;

     cout<<"finally velocity v1 is = "<<v1<<endl;
    }

   else if((RPm>=RP1)&&(RP1<=0))
     {
       v1 = v[1];
       if(v1>=vmax)
      v1 = vmax;
      else
       v1;  v1 = v1*sff1;
      }
  else
    {
       cout<<"Resistance is high"<<endl
           <<"Take other Model "<<endl;
     }

}

 t1 =(d1/v1)*60;

 cout<<"time for section Af is =";
 cout<<t1<<endl;
  getch();


//   ONWARD DIRECTION :

//    SECTION B :

      float RP2,v2;
      float t2;

     Tel = ct*Wd1*1000 ;

     RPm = (270*HP*e)/v[0];

   if(Tel>=RPm)

   {

      RP2 = (RR2 + GR2)*G ;

      if((RPm>=RP2)&&(RP2>0))

        {
         v2 = (270*HP*e)/RP2;

        if(v2>=v[1])
        v2 = v[1];
       else
        v2;
```

```
     if(v2>=vmax)
      v2 = vmax;
     else
       v2;
     v2 = v2*sff2 ;

     cout<<"finally velocity v2 is = "<<v2<<endl;
    }

  else if((RPm>=RP2)&&(RP2<=0))
     {
       v2 = v[1];
       if(v2>=vmax)
      v2 = vmax;
     else
       v2;
      v2 = v2*sff2 ;
       }
 else
     {

     cout<<"Resistance is high"<<endl
       <<"Take other Model"<<endl;

      }


 }
else if (Tel<RPm)
  {
    RPm = Tel;

    RP2 = (RR2 + GR2)*G ;

    if((RPm>=RP2)&&(RP2>0))

     {
      v2 = (270*HP*e)/RP2;

     if(v2>=v[1])
      v2 = v[1];
     else
      v2;

     if(v2>=vmax)
      v2 = vmax;
     else
       v2;
     v2 = v2*sff2 ;
     cout<<"finally velocity v2 is = "<<v2<<endl;
    }

  else if((RPm>=RP2)&&(RP2<=0))
     {
       v2 = v[1];
       if(v2>=vmax)
      v2 = vmax;
```

```
        else
        v2;
        v2 = v2*sff2 ;
       }
   else
       { cout<<"Resistance is high"<<endl
            <<"Take other Model"<<endl;
       }

   }

  t2 =(d2/v2)*60;

  cout<<"time for section Bf is =";
  cout<<t2<<endl;
   getch();

//   ONWARD DIRECTION :

//    SECTION C :
   float RP3,t3,v3;

   Tel = ct*Wdl*1000 ;

   RPm = (270*HP*e)/v[0];

  if(Tel>=RPm)

  {

     RP3 = (RR3 + GR3)*G ;

     if((RPm>=RP3)&&(RP3>0))

       {
        v3 = (270*HP*e)/RP3;

       if(v3>=v[1])
       v3 = v[1];
        else
       v3;

      if(v3>=vmax)
       v3 = vmax;
      else
        v3;
      v3 = v3*sff3 ;
      cout<<"finally velocity v3 is = "<<v3<<endl;
     }

  else if((RPm>=RP3)&&(RP3<=0))
      {
        v3 = v[1];
        if(v3>=vmax)
       v3 = vmax;
        else
        v3;
```

```
        v3 = v3*sff3 ;
        }
    else
       {  cout<<"Resistance is high"<<endl
              <<"Take other Model"<<endl;
       }

    }
  else if (Tel<RPm)
    {
      RPm = Tel;

      RP3 = (RR3 + GR3)*G ;

      if((RPm>=RP3)&&(RP3>0))

       {
         v3 = (270*HP*e)/RP3;

        if(v3>=v[1])
        v3 = v[1];
        else
         v3;

       if(v3>=vmax)
        v3 = vmax;
       else
         v3;
        v3 = v3*sff3 ;
       cout<<"finally velocity v3 is = "<<v3<<endl;
      }

     else if((RPm>=RP3)&&(RP3<=0))
       {
         v3 = v[1];
         if(v3>=vmax)
        v3 = vmax;
       else
         v3; v3 = v3*sff3;
        }
    else
      {
         cout<<"Resistance is high"<<endl
             <<"Take other Model"<<endl;
       }

   }

   t3 =(d3/v3)*60;

   cout<<"time for section Cf is =";
   cout<<t3<<endl;
    getch();

//   SECTION D :

      float v4,RP4;
```

```
    float t4;

   G = E + L ;
   Wdl = fdl*G ;

  Tel = ct*Wdl*1000 ;

  RPm = (270*HP*e)/v[0];

 if(Tel>=RPm)

 {
     RP4 = (RR4 + GR4)*G ;

    if((RPm>=RP4)&&(RP4>0))

      {
       v4 = (270*HP*e)/RP4;
       cout<<"vel v4 is ="<<v4<<endl;

      if(v4>=v[1])
      v4 = v[1];
      else
      v4;

       if(v4>=vmax)
      v4 = vmax;
      else
       v4;
       v4 = v4*sff4 ;

     cout<<"finally velocity v4 is = "<<v4<<endl;
    }
   else if((RPm>=RP4)&&(RP4<=0))
     {
       v4 = v[1];
       if(v4>=vmax)
      v4 = vmax;
      else
       v4;
       v4 = v4*sff4 ;

      }
    else
      {
       cout<<"Resistance is high"<<endl
           <<"Take other Model "<<endl;
       }

 }

else if (Tel<RPm)
  {
    RPm = Tel;

    RP4 = (RR4 + GR4)*G ;
```

```
      if((RPm>=RP4)&&(RP4>0))

       {
         v4 = (270*HP*e)/RP4;

        if(v4>=v[1])
        v4 = v[1];
        else
        v4;

       if(v4>=vmax)
        v4 = vmax;
       else
         v4;
        v4 = v4*sff4 ;

       cout<<"finally velocity v4 is = "<<v4<<endl;
      }

     else if((RPm>=RP4)&&(RP4<=0))
      {
         v4 = v[1];
         if(v4>=vmax)
        v4 = vmax;
       else
         v4; v4 = v4*sff4;
        }
    else
      {
         cout<<"Resistance is high"<<endl
             <<"Take other Model"<<endl;
      }

  }

   t4 =(d4/v4)*60;

   cout<<"time for section Df is =";
   cout<<t4<<endl;
    getch();

//    SECTION E :

      float v5,RP5;
      float t5;

     G = E + L ;

     Wdl = fdl*G ;

    Tel = ct*Wdl*1000 ;

    RPm = (270*HP*e)/v[0];

    if(Tel>=RPm)

   {
```

```
RP5 = (RR5 + GR5)*G ;

if((RPm>=RP5)&&(RP5>0))

  {

   v5 = (270*HP*e)/RP5;
   cout<<"vel v5 is ="<<v5<<endl;

   if(v5>=v[1])
  v5 = v[1];
   else
  v5;

 if(v5>=vmax)
  v5 = vmax;
 else
   v5;
   v5 = v5*sff5 ;

 cout<<"finally velocity v5 is = "<<v5<<endl;
}
else if((RPm>=RP5)&&(RP5<=0))
{
   v5 = v[1];
   if(v5>=vmax)
  v5 = vmax;
  else
   v5;
   v5 = v5*sff5 ;

  }
else
  {
   cout<<"Resistance is high"<<endl
       <<"Take other Model"<<endl;
   }

}

else if (Tel<RPm)
  {
    RPm = Tel;

    RP5 = (RR5 + GR5)*G ;

    if((RPm>=RP5)&&(RP5>0))

     {
      v5 = (270*HP*e)/RP5;

     if(v5>=v[1])
     v5 = v[1];
     else
      v5;
```

```
        if(v5>=vmax)
         v5 = vmax;
        else
          v5;
         v5 = v5*sff5 ;

        cout<<"finally velocity v5 is = "<<v5<<endl;
       }

     else if((RPm>=RP5)&&(RP5<=0))
        {
          v5 = v[1];
          if(v5>=vmax)
         v5 = vmax;
        else
          v5; v5 = v5*sff5 ;
         }
    else
       {
          cout<<"Resistance is high"<<endl
              <<"Take other Model"<<endl;
        }

 }

  t5 =(d5/v5)*60;

  cout<<"time for section Ef is =";
  cout<<t5<<endl;
   getch();

//    BACKWARD JOURNEY

//    SECTION E

    float v5b;
    float Tee,RP5b;
    float Wde;
    float t5b;

  G = E  ;
  Wde = fde*G ;
  Tee = ct*Wde*1000 ;

  RPm = (270*HP*e)/v[0];

  if(Tee>=RPm)

  {
      RP5b = (RR5b + GR5b)*G ;

     if((RPm>=RP5b)&&(RP5b>0))

       {
        v5b = (270*HP*e)/RP5b;

      if(v5b>=v[1])
```

```
      v5b = v[1];
     else
      v5b;

     if(v5b>=vmax)
      v5b = vmax;
     else
       v5b;
       v5b = v5b*sfb5 ;
     cout<<"finally velocity v5b is = "<<v5b<<endl;
    }

   else if((RPm>=RP5b)&&(RP5b<=0))
     {
       v5b = v[1];
       if(v5b>=vmax)
      v5b = vmax;
     else
       v5b;
       v5b = v5b*sfb5 ;
      }
  else
    {
     cout<<"Resistance is high"<<endl
          <<"Take other Model"<<endl;
    }

  }
else if (Tee<RPm)
  {
    RPm = Tee;

    RP5b = (RR5b + GR5b)*G ;

    if((RPm>=RP5b)&&(RP5b>0))

     {
       v5b = (270*HP*e)/RP5b;

      if(v5b>=v[1])
      v5b = v[1];
      else
      v5b;

     if(v5b>=vmax)
      v5b = vmax;
     else
       v5b;
      v5b = v5b*sfb5 ;
     cout<<"finally velocity v5b2 is = "<<v5b<<endl;
    }

   else if((RPm>=RP5b)&&(RP5b<=0))
     {
       v5b = v[1];
       if(v5b>=vmax)
      v5b = vmax;
```

```
        else
          v5b;
         v5b = v5b*sfb5 ;
         }
     else
       {  cout<<"Resistance is high"<<endl
              <<"Take other Model of higher HP"<<endl;

       }

     }

   t5b =(d5b/v5b)*60;

   cout<<"time for section Eb is =";
   cout<<t5b<<endl;
    getch();

//  SECTION D :

     float v4b;
     float RP4b;
     float t4b;

   G = E  ;
   Wde = fde*G ;
   Tee = ct*Wde*1000 ;

   RPm = (270*HP*e)/v[0];

   if(Tee>=RPm)

   {
      RP4b = (RR4b + GR4b)*G ;

      if((RPm>=RP4b)&&(RP4b>0))

        {
         v4b = (270*HP*e)/RP4b;

        if(v4b>=v[1])
        v4b = v[1];
        else
        v4b;

       if(v4b>=vmax)
        v4b = vmax;
       else
         v4b;
         v4b = v4b*sfb4 ;
       cout<<"finally velocity v4b1 is = "<<v4b<<endl;
       }

     else if((RPm>=RP4b)&&(RP4b<=0))
       {
         v4b = v[1];
         if(v4b>=vmax)
```

```
      v4b = vmax;
     else
       v4b;
       v4b = v4b*sfb4 ;
      }
  else
    {
     cout<<"Resistance is high"<<endl
          <<"Take other Model "<<endl;
     }

 }
else if (Tee<RPm)
  {
    RPm = Tee;

    RP4b = (RR4b + GR4b)*G ;

    if((RPm>=RP4b)&&(RP4b>0))

     {
       v4b = (270*HP*e)/RP4b;

      if(v4b>=v[1])
      v4b = v[1];
      else
      v4b;

      if(v4b>=vmax)
      v4b = vmax;
     else
       v4b;
      v4b = v4b*sfb4 ;
     cout<<"finally velocity v4b2 is = "<<v4b<<endl;
    }

   else if((RPm>=RP4b)&&(RP4b<=0))
     {
       v4b = v[1];
       if(v4b>=vmax)
      v4b = vmax;
     else
       v4b;
      v4b = v4b*sfb4 ;
      }
  else
    {  cout<<"Resistance is high"<<endl
          <<"Take other Model "<<endl;
    }

 }

 t4b =(d4b/v4b)*60;

 cout<<"time for section Db is =";
 cout<<t4b<<endl;
 getch();
```

```
//    SECTION C :

        float v3b;
        float RP3b;
        float t3b;

      G = E  ;
      Wde = fde*G ;
      Tee = ct*Wde*1000 ;

      RPm = (270*HP*e)/v[0];

      if(Tee>=RPm)

      {
         RP3b = (RR3b + GR3b)*G ;

         if((RPm>=RP3b)&&(RP3b>0))

           {
            v3b = (270*HP*e)/RP3b;

            if(v3b>=v[1])
           v3b = v[1];
           else
           v3b;

          if(v3b>=vmax)
           v3b = vmax;
          else
            v3b;
            v3b = v3b*sfb3 ;
          cout<<"finally velocity v3b1 is = "<<v3b<<endl;
         }

        else if((RPm>=RP3b)&&(RP3b<=0))
          {
            v3b = v[1];
            if(v3b>=vmax)
           v3b = vmax;
          else
            v3b;
            v3b = v3b*sfb3 ;
           }
       else
         {
           cout<<"Resistance is high"<<endl
                <<"Take other Model"<<endl;
            }


       }
     else if (Tee<RPm)
       {
         RPm = Tee;
```

```
      RP3b = (RR3b + GR3b)*G ;

      if((RPm>=RP3b)&&(RP3b>0))

       {
         v3b = (270*HP*e)/RP3b;

        if(v3b>=v[1])
        v3b = v[1];
        else
        v3b;

       if(v3b>=vmax)
        v3b = vmax;
       else
         v3b;
        v3b = v3b*sfb3 ;
       cout<<"finally velocity v3b2 is = "<<v3b<<endl;
      }

     else if((RPm>=RP3b)&&(RP3b<=0))
       {
         v3b = v[1];
         if(v3b>=vmax)
        v3b = vmax;
       else
         v3b;
        v3b = v3b*sfb3 ;
        }
   else
     {  cout<<"Resistance is high"<<endl
            <<"Take other Model"<<endl;

       }

   }

  t3b =(d3b/v3b)*60;

  cout<<"time for section Cb is =";
  cout<<t3b<<endl;
   getch();

//     BACKWARD  JOURNEY :
//     SECTION B

      float v2b,RP2b,t2b;

  RPm = (270*HP*e)/v[0];

  if(Tee>=RPm)

  {
     RP2b = (RR2b + GR2b)*G ;

     if((RPm>=RP2b)&&(RP2b>0))
```

```
       {
        v2b = (270*HP*e)/RP2b;

       if(v2b>=v[1])
       v2b = v[1];
       else
        v2b;

      if(v2b>=vmax)
       v2b = vmax;
      else
        v2b;
        v2b = v2b*sfb2 ;
      cout<<"finally velocity v2b is = "<<v2b<<endl;
     }

    else if((RPm>=RP2b)&&(RP2b<=0))
      {
        v2b = v[1];
        if(v2b>=vmax)
       v2b = vmax;
      else
        v2b;
       v2b = v2b*sfb2 ;
       }
   else
     {
       cout<<"Resistance is high"<<endl
            <<"Take other Model"<<endl;
     }

   }
 else if (Tee<RPm)
   {
     RPm = Tee;

     RP2b = (RR2b + GR2b)*G ;

     if((RPm>=RP2b)&&(RP2b>0))

      {
        v2b = (270*HP*e)/RP2b;

       if(v2b>=v[1])
       v2b = v[1];
       else
       v2b;

      if(v2b>=vmax)
       v2b = vmax;
      else
        v2b;
       v2b = v2b*sfb2 ;
      cout<<"finally velocity v2b is = "<<v2b<<endl;
     }

    else if((RPm>=RP2b)&&(RP2b<=0))
```

```
          {
            v2b = v[1];
            if(v2b>=vmax)
           v2b = vmax;
          else
            v2b;
           v2b = v2b*sfb2 ;
            }
       else
         {
           cout<<"Resistance is high"<<endl
                <<"Take other Model"<<endl;
          }

       }

      t2b =(d2b/v2b)*60;

      cout<<"time for section Bb is =";
      cout<<t2b<<endl;
       getch();

//      BACKWARD  JOURNEY :
//      SECTION A

        float v1b,RP1b;
       float t1b;

       RPm = (270*HP*e)/v[0];

      if((RPm>=RP1b)&&(RP1b>0))

      {
         RP1b = (RR1b + GR1b)*G ;

         if(RPm>=RP1b)

           {
            v1b = (270*HP*e)/RP1b;

            if(v1b>=v[1])
           v1b = v[1];
            else
           v1b;

          if(v1b>=vmax)
           v1b = vmax;
          else
            v1b;
           v1b = v1b*sfb1 ;
          cout<<"finally velocity v1b is = "<<v1b<<endl;
          }

       else if((RPm>=RP1b)&&(RP1b<=0))
         {
           v1b = v[1];
           if(v1b>=vmax)
```

```
      v1b = vmax;
     else
       v1b;
      v1b = v1b*sfb1 ;
      }
  else
    {
      cout<<"Resistance is high"<<endl
          <<"Take other Model "<<endl;
    }

  }
else if (Tee<RPm)
  {
    RPm = Tee;

   RP1b = (RR1b + GR1b)*G ;

    if((RPm>=RP1b)&&(RP1b>0))

     {
      v1b = (270*HP*e)/RP1b;

      if(v1b>=v[1])
      v1b = v[1];
      else
      v1b;

     if(v1b>=vmax)
      v1b = vmax;
     else
       v1b;
      v1b = v1b*sfb1 ;
     cout<<"finally velocity v1b is = "<<v1b<<endl;
    }

   else if((RPm>=RP1b)&&(RP1b<=0))
     {
       v1b = v[1];
       if(v1b>=vmax)
      v1b = vmax;
      else
       v1b;
    v1b = v1b*sfb1 ;
      }
  else
    {
      cout<<"Resistance is high"<<endl
          <<"Take other Model "<<endl;
    }

  }

 t1b =(d1b/v1b)*60;

 cout<<"time for section Ab is =";
 cout<<t1b<<endl;
```

```
getch();

//   TOTAL CYCLE TIME :

float T,TC,CTC ;

Lt = (Hc*tl)/b ;

T = t1+t2+t3+t4+t5+t1b+t2b+t3b+t4b+t5b ;
cout<<" Total travel time is = "<<T<<endl;
fprintf(f2,"Total travel time  T(min) is\n");
fprintf(f2,"%0.3f\n",T);

TC = T  ;

CTC = Lt + TC + D;
cout<<" Total cycle time is = "<<CTC<<endl;
fprintf(f2,"Total cycle time  CTC(min) is\n");
fprintf(f2,"%0.3f\n",CTC);

float pp = Hc*(60/CTC) ;
cout<<" productivity of single hauler is = "<<pp<<endl;

//ROUNDING BASED ON PRODUCTIVITY :

 float N;

N = CTC/Lt;

cout<<"N is = "<<N<<endl;

 int N2;

 int N1 = int(N) ;
N2 = N1 + 1 ;

float HP1,HP2,A,ACT,Nu;

HP1 = (Hc*N1*60)/CTC ;

if(LP>HP1)
{
A = (N2*Lt) - CTC ;
ACT = CTC + A ;

HP2 = (Hc*N2*60)/ACT ;

cout<<"Hauler productivity is = "<<HP2<<endl;
fprintf(f2,"Hauler productivity is HP(cum/hr) is\n");
fprintf(f2,"%0.3f\n",HP2);

cout<<" No. of haulers is = "<<N2<<endl;
fprintf(f2,"No. of haulers based on productivity is\n");
```

```
   fprintf(f2,"%d \n",N2);

   Nu = N2;
   cout<<"Nu is ="<<Nu<<endl;
    }

   else
   {
    cout<<"Hauler productivity is = "<<HP1<<endl;
   fprintf(f2,"Hauler productivity HP(cum/hr) is\n");
   fprintf(f2,"%0.3f\n",HP1);

    cout<<"No. of haulers is = "<<N1<<endl;
   fprintf(f2,"No. of haulers based on productivity is\n");
   fprintf(f2,"%d \n",N1);

   Nu =  N1;
   cout<<"Nu is ="<<Nu<<endl;
    }

// ROUNDING BASED ON TOTAL COSTS

 float Tc1,Tc2;

 Tc1 = ((M)*(CTC)*((Odc*N1) + Oh1))/(N1*Hc*60) ;
 Tc2 = ((M)*(CTC)*((Odc*N2) + Oh1))/(N2*Hc*60) ;

 cout<<"TC1 is ="<<Tc1<<endl;
   fprintf(f2,"Total cost Tc1(Rs) is\n");
   fprintf(f2,"%0.3f\n",Tc1);

 cout<<"TC2 is ="<<Tc2<<endl;
   fprintf(f2,"Total cost Tc2(Rs) is\n");
   fprintf(f2,"%0.3f\n",Tc2);


 if(Tc1<Tc2)
 {
  cout<<"No. of Haulers is = "<<N1<<endl;
   fprintf(f2,"No. of haulers based on total cost is\n");
   fprintf(f2,"%d \n",N1);

 }
 else

 cout<<"No. of Haulers is = "<<N2<<endl;
   fprintf(f2,"No. of haulers based on total cost is\n");
   fprintf(f2,"%d \n",N2);


//  OPTIMUM HAUL UNIT SIZE

    float CTs,w,Ps,TT,CIN;

   w = .20*CTC ;

    if(Nu==N1)
```

```
    CTs = CTC + w ;

    else

   CTs = CTC + w + A ;
   cout<<"Sustained cycle time is ="<<CTs<<endl;
   fprintf(f2,"Sustained cycle time CTs(min) is\n");
   fprintf(f2,"%0.3f\n",CTs);

    getch();

    Ps = (60*Nu*Hc)/CTs ;

    TT = (M*CTs)/(60*Nu*Hc)   ;

    cout<<"Sustained productivity is = "<<Ps<<endl;
    fprintf(f2,"Sustained productivity  Ps(cum/hr) is\n");
   fprintf(f2,"%0.3f\n",Ps);

//      cout<<"Total time to complete the haul is ="<<TT<<endl;
//   fprintf(f2,"Total time to complete the haul TT(hours) is\n");
//   fprintf(f2,"%0.3f\n",TT);


    CIN = (TT*(Nu*(Odc) + IC))/M ;

   cout<<"Cost Index number is = "<<CIN<<endl;
   fprintf(f2,"Cost Index number CIN(Rs/cum) is\n");
   fprintf(f2,"%0.3f\n",CIN);

   getch();

 }
```

APPENDIX - G

```
/*      *******************************************************************
          PROGRAM  OF  OPTIMISATION  MODEL  OF  HAUL  UNITS
                 (Probabilistic Model)

     *******************************************************************
*/

     #include<iostream.h>
     #include<conio.h>
     #include<math.h>
     #include<stdio.h>
     #include<iomanip.h>

       int factorial(int n)
       {
         if(n<0) return 0;
         int f = 1;
         while(n>1)
         f *= n-- ;
         return f;
        }

     void main()

      {
      clrscr();
      FILE*f1;
      FILE*f2;
//**********************************************************************

        float HP,e,RR1,RR2,RR3,RR4,RR5,GR1,GR2,GR3,GR4,GR5;
        float RR1b,RR2b,RR3b,RR4b,RR5b,GR1b,GR2b,GR3b,GR4b,GR5b;
        float G,L,E,ct,d1,d2,d3,d4,d5;
        float d1b,d2b,d3b,d4b,d5b,vmax,fdl,fde;
        float ks,Hc,Hcg,Hcc;
        float v[5];
        float sff1,sff2,sff3,sff4,sff5,sfb1,sfb2,sfb3,sfb4,sfb5;
        float hrs,q1,qy;
        float Odc,Ohl;
        float RPm,Lt;
        int i;
        float dc,tl,b,D;
        float DF,h;
f1=fopen("shma3.dat","r");
fscanf(f1,"%f%f%f%f%f%f%f",&HP,&e,&RR1,&RR2,&RR3,&RR4,&RR5);
fscanf(f1,"%f%f%f%f%f",&GR1,&GR2,&GR3,&GR4,&GR5);
fscanf(f1,"%f%f%f%f%f",&RR1b,&RR2b,&RR3b,&RR4b,&RR5b);
fscanf(f1,"%f%f%f%f%f",&GR1b,&GR2b,&GR3b,&GR4b,&GR5b);
fscanf(f1,"%f%f%f%f%f%f%f%f",&L,&E,&ct,&d1,&d2,&d3,&d4,&d5);
fscanf(f1,"%f%f%f%f%f",&d1b,&d2b,&d3b,&d4b,&d5b);
fscanf(f1,"%f%f%f%f",&fdl,&fde,&ks,&Hcg);
fscanf(f1,"%f%f%f%f%f",&sff1,&sff2,&sff3,&sff4,&sff5);
fscanf(f1,"%f%f%f%f%f",&sfb1,&sfb2,&sfb3,&sfb4,&sfb5);
fscanf(f1,"%f%f%f%f%f",&hrs,&qy,&Odc,&Ohl,&vmax);
fscanf(f1,"%f%f%f%f",&tl,&b,&D,&h);
```

```
for(i=0;i<2;i++) fscanf(f1,"%f",&v[i]);
f2=fopen("shma3.res","w");
fclose(f1);
//fclose(f2);

        Hcc = (L*1000)/ks ;

        if(Hcc>=Hcg)

        Hc = Hcg ;

        else

        Hc = Hcc ;

       cout<<"Adopted Heaped capacity of m/c is = "<<Hc<<endl;

       q1 = qy/hrs   ;

       float LP = (b*60)/t1 ;

       float Snn = q1/LP ;

       int S1 = int(Snn) ;

       int Sn = S1 + 1 ;

       cout<<"Loaders required = "<<Sn<<endl;

       if(h>900)

       DF = 3*((h-900)/300);
       else
       DF = 0;

       HP = HP*((100-DF)/100) ;

       cout<<"Available power is = "<<HP<<endl;

    fprintf(f2,"************************************************** \n");
    fprintf(f2,"RESULTS OF MODEL(Deterministic Model)  \n");
    fprintf(f2,"************************************************** \n");
    fprintf(f2,"Required Hourly Production(based on Peak Production qy)
q1(cum/hr) is\n");
    fprintf(f2,"%0.3f\n",q1);

    RPm = (270*HP*e)/v[0];

  //   ONWARD DIRECTION :

 //    SECTION A :

        float RP1,v1;
        float Te1;
        float Wd1;
        float t1;
       G = E + L ;
```

```
  Wdl = fdl*G ;

 Tel = ct*Wdl*1000 ;

 if(Tel>=RPm)

 {
    RP1 = (RR1 + GR1)*G ;

    if((RPm>=RP1)&&(RP1>0))

      {
       v1 = (270*HP*e)/RP1;
       cout<<"vel v1 is ="<<v1<<endl;

     if(v1>=v[1])
      v1 = v[1];
      else
      v1;

     if(v1>=vmax)
      v1 = vmax;
     else
       v1;
       v1 = v1*sff1 ;

     cout<<"finally velocity v1 is = "<<v1<<endl;
    }
   else if((RPm>=RP1)&&(RP1<=0))
     {
       v1 = v[1];
       if(v1>=vmax)
      v1 = vmax;
     else
       v1;
       v1 = v1*sff1 ;

      }
    else
      {
       cout<<"Resistance is high"<<endl
           <<"Take other Model "<<endl;
     }
 }

else if (Tel<RPm)
  {
    RPm = Tel;

    RP1 = (RR1 + GR1)*G ;

    if((RPm>=RP1)&&(RP1>0))

     {
       v1 = (270*HP*e)/RP1;

     if(v1>=v[1])
```

```
        v1 = v[1];
       else
          v1;

       if(v1>=vmax)
        v1 = vmax;
       else
         v1;
        v1 = v1*sff1 ;

       cout<<"finally velocity v1 is = "<<v1<<endl;
      }

     else if((RPm>=RP1)&&(RP1<=0))
       {
         v1 = v[1];
         if(v1>=vmax)
        v1 = vmax;
       else
         v1;  v1 = v1*sff1;
        }
    else
      {
         cout<<"Resistance is high"<<endl
             <<"Take other Model "<<endl;
       }

  }

   t1 =(d1/v1)*60;

   cout<<"time for section Af is =";
   cout<<t1<<endl;
    getch();

 //   ONWARD DIRECTION :

//    SECTION B :

      float RP2,v2;
      float t2;

     Tel = ct*Wd1*1000 ;

     RPm = (270*HP*e)/v[0];

   if(Tel>=RPm)

   {
      RP2 = (RR2 + GR2)*G ;

      if((RPm>=RP2)&&(RP2>0))

        {
         v2 = (270*HP*e)/RP2;

       if(v2>=v[1])
```

```
      v2 = v[1];
     else
      v2;

     if(v2>=vmax)
      v2 = vmax;
     else
       v2;
     v2 = v2*sff2 ;

     cout<<"finally velocity v2 is = "<<v2<<endl;
    }

   else if((RPm>=RP2)&&(RP2<=0))
     {
       v2 = v[1];
       if(v2>=vmax)
      v2 = vmax;
     else
       v2;
      v2 = v2*sff2 ;
      }
  else
        {

       cout<<"Resistance is high"<<endl
           <<"Take other Model "<<endl;
       }

  }
else if (Tel<RPm)
  {
    RPm = Tel;

    RP2 = (RR2 + GR2)*G ;

    if((RPm>=RP2)&&(RP2>0))

     {
      v2 = (270*HP*e)/RP2;

     if(v2>=v[1])
      v2 = v[1];
     else
      v2;

     if(v2>=vmax)
      v2 = vmax;
     else
       v2;
     v2 = v2*sff2 ;
     cout<<"finally velocity v2 is = "<<v2<<endl;
    }

   else if((RPm>=RP2)&&(RP2<=0))
     {
       v2 = v[1];
```

```
          if(v2>=vmax)
         v2 = vmax;
        else
          v2;
          v2 = v2*sff2 ;
         }
     else
         { cout<<"Resistance is high"<<endl
               <<"Take other Model"<<endl;

          }

     }

    t2 =(d2/v2)*60;

    cout<<"time for section Bf is =";
    cout<<t2<<endl;
     getch();

  //   ONWARD DIRECTION :

//    SECTION C :

     float v3,t3,RP3;

     Tel = ct*Wdl*1000 ;

     RPm = (270*HP*e)/v[0];

    if(Tel>=RPm)

    {
       RP3 = (RR3 + GR3)*G ;

       if((RPm>=RP3)&&(RP3>0))

         {
          v3 = (270*HP*e)/RP3;

        if(v3>=v[1])
         v3 = v[1];
        else
         v3;

        if(v3>=vmax)
         v3 = vmax;
        else
          v3;
        v3 = v3*sff3 ;
        cout<<"finally velocity v3 is = "<<v3<<endl;
       }

      else if((RPm>=RP3)&&(RP3<=0))
       {
          v3 = v[1];
          if(v3>=vmax)
```

```
        v3 = vmax;
       else
         v3;
        v3 = v3*sff3 ;
        }
    else
       {  cout<<"Resistance is high"<<endl
              <<"Take other Model "<<endl;

         }

    }
  else if (Tel<RPm)
    {
      RPm = Tel;

      RP3 = (RR3 + GR3)*G ;

      if((RPm>=RP3)&&(RP3>0))

       {
         v3 = (270*HP*e)/RP3;

       if(v3>=v[1])
        v3 = v[1];
       else
         v3;

       if(v3>=vmax)
        v3 = vmax;
       else
         v3;
        v3 = v3*sff3 ;
       cout<<"finally velocity v3 is = "<<v3<<endl;
      }

     else if((RPm>=RP3)&&(RP3<=0))
       {
         v3 = v[1];
         if(v3>=vmax)
        v3 = vmax;
       else
         v3; v3 = v3*sff3;
        }
    else
      {
         cout<<"Resistance is high"<<endl
              <<"Take other Model"<<endl;
      }

  }

  t3 =(d3/v3)*60;

  cout<<"time for section Cf is =";
  cout<<t3<<endl;
   getch();
```

```
//    SECTION D :

      float v4,RP4;
      float t4;
      G = E + L ;
      Wdl = fdl*G ;

    Tel = ct*Wdl*1000 ;

    RPm = (270*HP*e)/v[0];

    if(Tel>=RPm)

    {
       RP4 = (RR4 + GR4)*G ;

       if((RPm>=RP4)&&(RP4>0))

         {

          v4 = (270*HP*e)/RP4;
          cout<<"vel v4 is ="<<v4<<endl;

        if(v4>=v[1])
         v4 = v[1];
         else
         v4;

        if(v4>=vmax)
         v4 = vmax;
        else
          v4;
          v4 = v4*sff4 ;

        cout<<"finally velocity v4 is = "<<v4<<endl;
       }
      else if((RPm>=RP4)&&(RP4<=0))
        {
          v4 = v[1];
          if(v4>=vmax)
         v4 = vmax;
        else
          v4;
          v4 = v4*sff4 ;

         }
       else
         {
          cout<<"Resistance is high"<<endl
              <<"Take other Model "<<endl;
          }

    }

   else if (Tel<RPm)
     {
```

```
RPm = Tel;

RP4 = (RR4 + GR4)*G ;

if((RPm>=RP4)&&(RP4>0))

 {
  v4 = (270*HP*e)/RP4;

 if(v4>=v[1])
  v4 = v[1];
 else
    v4;

 if(v4>=vmax)
  v4 = vmax;
 else
   v4;
  v4 = v4*sff4 ;

 cout<<"finally velocity v4 is = "<<v4<<endl;
}

else if((RPm>=RP4)&&(RP4<=0))
  {
   v4 = v[1];
   if(v4>=vmax)
  v4 = vmax;
 else
   v4; v4 = v4*sff4;
  }
else
  {
    cout<<"Resistance is high"<<endl
        <<"Take other Model "<<endl;
  }

}

t4 =(d4/v4)*60;

cout<<"time for section Df is =";
cout<<t4<<endl;
 getch();

//    SECTION E :

  float v5,RP5;
 float t5;

 G = E + L ;
 Wdl = fdl*G ;

Tel = ct*Wdl*1000 ;

RPm = (270*HP*e)/v[0];
```

```
if(Tel>=RPm)
{
   RP5 = (RR5 + GR5)*G ;

   if((RPm>=RP5)&&(RP5>0))

     {
      v5 = (270*HP*e)/RP5;
      cout<<"vel v5 is ="<<v5<<endl;

    if(v5>=v[1])
     v5 = v[1];
     else
     v5;

    if(v5>=vmax)
     v5 = vmax;
    else
      v5;
      v5 = v5*sff5 ;

    cout<<"finally velocity v5 is = "<<v5<<endl;
   }
  else if((RPm>=RP5)&&(RP5<=0))
    {
      v5 = v[1];
      if(v5>=vmax)
     v5 = vmax;
    else
      v5;
      v5 = v5*sff5 ;

     }
   else
     {
      cout<<"Resistance is high"<<endl
          <<"Take other Model "<<endl;
     }
}

else if (Tel<RPm)
  {
    RPm = Tel;

    RP5 = (RR5 + GR5)*G ;

    if((RPm>=RP5)&&(RP5>0))

     {
      v5 = (270*HP*e)/RP5;

    if(v5>=v[1])
     v5 = v[1];
    else
       v5;
```

```
        if(v5>=vmax)
         v5 = vmax;
        else
          v5;
         v5 = v5*sff5 ;

        cout<<"finally velocity v5 is = "<<v5<<endl;
       }

      else if((RPm>=RP5)&&(RP5<=0))
        {
          v5 = v[1];
          if(v5>=vmax)
         v5 = vmax;
        else
          v5;  v5 = v5*sff5 ;
          }
     else
       {
          cout<<"Resistance is high"<<endl
              <<"Take other Model "<<endl;
       }
   }

    t5 =(d5/v5)*60;

    cout<<"time for section Ef is =";
    cout<<t5<<endl;
     getch();

 //    BACKWARD JOURNEY

//     SECTION E

       float v5b;
      float Tee,RP5b;
      float Wde;
      float t5b;

    G = E  ;
    Wde = fde*G ;
    Tee = ct*Wde*1000 ;

    RPm = (270*HP*e)/v[0];

    if(Tee>=RPm)

    {
       RP5b = (RR5b + GR5b)*G ;

       if((RPm>=RP5b)&&(RP5b>0))

         {
          v5b = (270*HP*e)/RP5b;

        if(v5b>=v[1])
         v5b = v[1];
```

```
      else
       v5b;

      if(v5b>=vmax)
       v5b = vmax;
      else
        v5b;
        v5b = v5b*sfb5 ;
      cout<<"finally velocity v5b1 is = "<<v5b<<endl;
     }

    else if((RPm>=RP5b)&&(RP5b<=0))
      {
        v5b = v[1];
        if(v5b>=vmax)
       v5b = vmax;
      else
        v5b;
        v5b = v5b*sfb5 ;
       }
   else
     {
      cout<<"Resistance is high"<<endl
           <<"Take other Model "<<endl;
      }

   }
 else if (Tee<RPm)
   {
     RPm = Tee;

     RP5b = (RR5b + GR5b)*G ;

     if((RPm>=RP5b)&&(RP5b>0))

      {
        v5b = (270*HP*e)/RP5b;

      if(v5b>=v[1])
       v5b = v[1];
      else
       v5b;

      if(v5b>=vmax)
       v5b = vmax;
      else
        v5b;
       v5b = v5b*sfb5 ;
      cout<<"finally velocity v5b2 is = "<<v5b<<endl;
     }

    else if((RPm>=RP5b)&&(RP5b<=0))
      {
        v5b = v[1];
        if(v5b>=vmax)
       v5b = vmax;
      else
```

```
       v5b;
      v5b = v5b*sfb5 ;
      }
 else
    {  cout<<"Resistance is high"<<endl
           <<"Take other Model "<<endl;
    }

}

t5b =(d5b/v5b)*60;

cout<<"time for section Eb is =";
cout<<t5b<<endl;
 getch();

   float v4b;
  float RP4b;
  float t4b;

G = E  ;
Wde = fde*G ;
Tee = ct*Wde*1000 ;

RPm = (270*HP*e)/v[0];

if(Tee>=RPm)

{

   RP4b = (RR4b + GR4b)*G ;

   if((RPm>=RP4b)&&(RP4b>0))

     {
      v4b = (270*HP*e)/RP4b;

    if(v4b>=v[1])
     v4b = v[1];
    else
     v4b;

    if(v4b>=vmax)
     v4b = vmax;
    else
      v4b;
      v4b = v4b*sfb4 ;
    cout<<"finally velocity v4b1 is = "<<v4b<<endl;
   }

  else if((RPm>=RP4b)&&(RP4b<=0))
    {
      v4b = v[1];
      if(v4b>=vmax)
     v4b = vmax;
    else
      v4b;
```

```
        v4b = v4b*sfb4 ;
        }
   else
     {
       cout<<"Resistance is high"<<endl
            <<"Take other Model. "<<endl;
     }

 }
 else if (Tee<RPm)
   {
     RPm = Tee;

     RP4b = (RR4b + GR4b)*G ;

     if((RPm>=RP4b)&&(RP4b>0))

      {
        v4b = (270*HP*e)/RP4b;

      if(v4b>=v[1])
       v4b = v[1];
      else
       v4b;

      if(v4b>=vmax)
       v4b = vmax;
      else
        v4b;
       v4b = v4b*sfb4 ;
      cout<<"finally velocity v4b is = "<<v4b<<endl;
     }

    else if((RPm>=RP4b)&&(RP4b<=0))
      {
        v4b = v[1];
        if(v4b>=vmax)
       v4b = vmax;
      else
        v4b;
       v4b = v4b*sfb4 ;
        }
   else
     {  cout<<"Resistance is high"<<endl
             <<"Take other Model "<<endl;
     }
 }

  t4b =(d4b/v4b)*60;

  cout<<"time for section Db is =";
  cout<<t4b<<endl;
   getch();

//    SECTION C :

      float v3b;
```

```
    float RP3b;
    float t3b;

  G = E  ;
  Wde = fde*G ;
  Tee = ct*Wde*1000 ;

  RPm = (270*HP*e)/v[0];

  if(Tee>=RPm)

  {
     RP3b = (RR3b + GR3b)*G ;

     if((RPm>=RP3b)&&(RP3b>0))

       {
        v3b = (270*HP*e)/RP3b;

      if(v3b>=v[1])
       v3b = v[1];
      else
       v3b;

      if(v3b>=vmax)
       v3b = vmax;
      else
        v3b;
        v3b = v3b*sfb3 ;
      cout<<"finally velocity v3b is = "<<v3b<<endl;
     }

    else if((RPm>=RP3b)&&(RP3b<=0))
      {
        v3b = v[1];
        if(v3b>=vmax)
       v3b = vmax;
      else
        v3b;
        v3b = v3b*sfb3 ;
       }
   else
     {
       cout<<"Resistance is high"<<endl
            <<"Take other Model "<<endl;
     }

   }
 else if (Tee<RPm)
   {
     RPm = Tee;

     RP3b = (RR3b + GR3b)*G ;

     if((RPm>=RP3b)&&(RP3b>0))

       {
```

```
        v3b = (270*HP*e)/RP3b;

      if(v3b>=v[1])
       v3b = v[1];
      else
       v3b;

      if(v3b>=vmax)
       v3b = vmax;
      else
        v3b;
       v3b = v3b*sfb3 ;
      cout<<"finally velocity v3b is = "<<v3b<<endl;
     }

   else if((RPm>=RP3b)&&(RP3b<=0))
     {
       v3b = v[1];
       if(v3b>=vmax)
      v3b = vmax;
     else
       v3b;
      v3b = v3b*sfb3 ;
       }
  else
    {  cout<<"Resistance is high"<<endl
            <<"Take other Model "<<endl;
    }
 }

 t3b =(d3b/v3b)*60;

 cout<<"time for section Cb is =";
 cout<<t3b<<endl;
  getch();

//    BACKWARD  JOURNEY :
//    SECTION B

      float v2b,RP2b,t2b;

 RPm = (270*HP*e)/v[0];

 if(Tee>=RPm)

 {
    RP2b = (RR2b + GR2b)*G ;

    if((RPm>=RP2b)&&(RP2b>0))

      {
       v2b = (270*HP*e)/RP2b;

     if(v2b>=v[1])
      v2b = v[1];
     else
       v2b;
```

```
      if(v2b>=vmax)
       v2b = vmax;
      else
        v2b;
        v2b = v2b*sfb2 ;
      cout<<"finally velocity v2b is = "<<v2b<<endl;
     }

    else if((RPm>=RP2b)&&(RP2b<=0))
      {
        v2b = v[1];
        if(v2b>=vmax)
       v2b = vmax;
      else
        v2b;
       v2b = v2b*sfb2 ;
       }
   else
     {
      cout<<"Resistance is high"<<endl
           <<"Take other Model"<<endl;
     }

   }
 else if (Tee<RPm)
   {
     RPm = Tee;

     RP2b = (RR2b + GR2b)*G ;

     if((RPm>=RP2b)&&(RP2b>0))

      {
        v2b = (270*HP*e)/RP2b;

      if(v2b>=v[1])
       v2b = v[1];
      else
       v2b;

      if(v2b>=vmax)
       v2b = vmax;
      else
        v2b;
       v2b = v2b*sfb2 ;
      cout<<"finally velocity v2b is = "<<v2b<<endl;
     }

    else if((RPm>=RP2b)&&(RP2b<=0))
      {
        v2b = v[1];
        if(v2b>=vmax)
       v2b = vmax;
      else
        v2b;
       v2b = v2b*sfb2 ;
```

```
            }
       else
         {
           cout<<"Resistance is high"<<endl
                <<"Take other Model "<<endl;
          }

       }

     t2b =(d2b/v2b)*60;

     cout<<"time for section Bb is =";
     cout<<t2b<<endl;
      getch();

//      BACKWARD  JOURNEY :
//      SECTION A

        float v1b,RP1b;
       float t1b;

       RPm =  (270*HP*e)/v[0];

     if((RPm>=RP1b)&&(RP1b>0))

     {
         RP1b =  (RR1b + GR1b)*G ;

        if(RPm>=RP1b)

          {
           v1b =  (270*HP*e)/RP1b;

         if(v1b>=v[1])
          v1b =  v[1];
         else
          v1b;

         if(v1b>=vmax)
          v1b = vmax;
         else
            v1b;
          v1b =  v1b*sfb1  ;
         cout<<"finally velocity v1b is = "<<v1b<<endl;
        }

       else if((RPm>=RP1b)&&(RP1b<=0))
         {
           v1b =  v[1];
           if(v1b>=vmax)
          v1b =  vmax;
         else
           v1b;
          v1b =  v1b*sfb1  ;
          }
      else
        {
```

```
        cout<<"Resistance is high"<<endl
            <<"Take other Model"<<endl;
      }

    }
    else if (Tee<RPm)
      {
        RPm = Tee;

       RP1b = (RR1b + GR1b)*G ;

        if((RPm>=RP1b)&&(RP1b>0))

         {
           v1b = (270*HP*e)/RP1b;

         if(v1b>=v[1])
          v1b = v[1];
         else
          v1b;

         if(v1b>=vmax)
          v1b = vmax;
         else
           v1b;
          v1b = v1b*sfb1 ;
         cout<<"finally velocity v1b is = "<<v1b<<endl;
        }

       else if((RPm>=RP1b)&&(RP1b<=0))
         {
           v1b = v[1];
           if(v1b>=vmax)
          v1b = vmax;
         else
           v1b;
        v1b = v1b*sfb1 ;
          }
     else
       {
           cout<<"Resistance is high"<<endl
               <<"Take other Model"<<endl;
        }
     }

     t1b =(d1b/v1b)*60;

     cout<<"time for section Ab is =";
     cout<<t1b<<endl;
      getch();

//   TOTAL CYCLE TIME :

    float T,TC,CTC ;

    Lt = (Hc*tl)/b ;
```

```
T = t1+t2+t3+t4+t5+t1b+t2b+t3b+t4b+t5b ;
cout<<" Total travel time is = "<<T<<endl;
fprintf(f2,"Total travel time  T(min) is\n");
fprintf(f2,"%0.3f\n",T);

TC = T  ;

CTC = Lt + TC + D;
cout<<" Total cycle time is = "<<CTC<<endl;
fprintf(f2,"Total cycle time  CTC(min) is\n");
fprintf(f2,"%0.3f\n",CTC);

   float l=60.0/T,m=60.0/Lt;
   float n11 = CTC/Lt;
   int n1 = int(n11);

   int j,n=5;
   float x,y,z;
   float P;

    int N[5] = {(n1-2),(n1-1),(n1),(n1+1),(n1+2)};

for(i=0;i<n;i++)
{
    double  p;double sum = 0 ; float f11,f12,f21,f22,f111,f211;

      for(j=0;j<=N[i];j++)
       {
        y = (l/m) ;
       x = pow(y,j);

       if(N[i]<0||j<0||j>N[i])
        p = 0 ;

        else if(N[i]<=7)
        {
          f11 = N[i];
          f12 = factorial(N[i]);
          f22 = factorial(N[i]-j) ;
           p = (f12/f22)*x ;
         }

         else if((N[i]>6)&&((N[i]-6)==2))
         {
            f11 = (N[i])*(N[i]-1);
            f12 = f11*factorial(N[i]-2) ;
            f21 = (N[i]-j)*(N[i]-j-1) ;
            f22 = f21*factorial(N[i]-j-2) ;

            p = (f12/f22)*x ;
         }

       else if((N[i]>6)&&((N[i]-6)==3))
         {

            f11 = (N[i])*(N[i]-1)*(N[i]-2) ;
            f12 = f11*factorial(N[i]-3) ;
```

```
        f21 = (N[i]-j)*(N[i]-j-1)*(N[i]-j-2);
        f22 = f21*factorial(N[i]-j-3) ;

        p = (f12/f22)*x ;
    }

else if((N[i]>6)&&((N[i]-6)==4))
    {

        f11 = (N[i])*(N[i]-1)*(N[i]-2)*(N[i]-3) ;
        f12 = f11*factorial(N[i]-4) ;
        f21 = (N[i]-j)*(N[i]-j-1)*(N[i]-j-2)*(N[i]-j-3);
        f22 = f21*factorial(N[i]-j-4) ;

        p = (f12/f22)*x ;
    }

 else if((N[i]>7)&&((N[i]-7)==4))
    {

        f11 = (N[i])*(N[i]-1)*(N[i]-2)*(N[i]-3) ;
        f12 = f11*factorial(N[i]-4) ;
        f21 = (N[i]-j)*(N[i]-j-1)*(N[i]-j-2)*(N[i]-j-3);
        f22 = f21*factorial(N[i]-j-4) ;

        p = (f12/f22)*x ;
    }

 else if((N[i]>7)&&((N[i]-7)==5))
    {

        f11 = (N[i])*(N[i]-1)*(N[i]-2)*(N[i]-3);
        f111 =f11*(N[i]-4) ;
        f12 = f111*factorial(N[i]-5) ;
        f21 = (N[i]-j)*(N[i]-j-1)*(N[i]-j-2)*(N[i]-j-3);
        f211 =f21*(N[i]-j-4);
        f22 = f211*factorial(N[i]-j-5) ;

        p = (f12/f22)*x ;
    }

  else

   cout<<"Program can't work"<<endl;

  sum = sum + p;

  }

  z = 1/sum  ;

  P = LP*(1-z);

 float C[4];
 float CIN[4];
 float CINm; int Nh ;
```

```
C[i] = (Odc*N[i] + Ohl);
CIN[i] =( C[i]/P );

cout<<"  \n " <<endl;

cout<<setw(8)<<"Productivity"

  <<setw(12)<<" haulers"

  <<setw(12)<<"CIN"<<endl;

cout<<setw(8)<<P

  <<setw(12)<<N[i]

  <<setw(15)<<CIN[i];

 getch();

   CINm = CIN[0];

      if(CIN[i]<=CINm)
      CINm = CIN[i];
      if(CINm == CIN[i])
      Nh = N[i];

     cout<<setw(20)<<"Optimum haulers"
       <<setw(8)<<Nh <<endl;
     getch();
```