

1-1-1978

The application of Q-GERT to materials handling problems in open pit mining.

Carlos A. Valenzuela

Follow this and additional works at: <http://preserve.lehigh.edu/etd>



Part of the [Industrial Engineering Commons](#)

Recommended Citation

Valenzuela, Carlos A., "The application of Q-GERT to materials handling problems in open pit mining." (1978). *Theses and Dissertations*. Paper 2132.

This Thesis is brought to you for free and open access by Lehigh Preserve. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of Lehigh Preserve. For more information, please contact preserve@lehigh.edu.

THE APPLICATION OF Q-GERT TO MATERIALS
HANDLING PROBLEMS IN OPEN PIT MINING

By
Carlos A. Valenzuela

A Thesis
Presented to the Graduate Committee
of Lehigh University
in Candidacy for the Degree of
Master of Science
in
Industrial Engineering

Lehigh University

1978

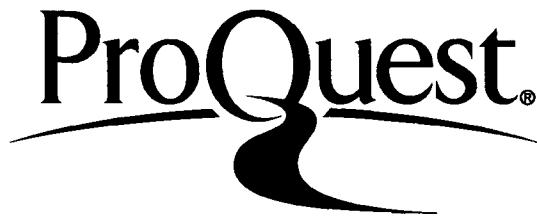
ProQuest Number: EP76405

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



ProQuest EP76405

Published by ProQuest LLC (2015). Copyright of the Dissertation is held by the Author.

All rights reserved.

This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

This thesis is accepted and approved in partial fulfillment of the requirements for the degree of Master of Science.

May 5, 1978
(date)

Dr. ~~Gay~~ E. Whitehouse,
Professor in Charge

Professor George E. Kane
Chairman of the Industrial
Engineering Department

ACKNOWLEDGEMENTS

The author would like to express his sincere appreciation to Dr. Gary E. Whitehouse for his interest, valuable suggestions and guidance during the preparation of this thesis. Special thanks are extended to Mr. Joseph R. Latoff, of the Direct Labor Department of Mack Trucks, Inc., for providing technical data on Off-Highway trucks. Mrs. Susan I. Adleman also deserves special thanks for her valuable assistance in typing part of the manuscript.

The author is indebted to Dr. Ben L. Wechsler for his help and encouragement during the process of learning a new language.

Most of all, the author wishes to thank her wife, Luz, for her help in preparing the diagrams and her continuing support. And finally, the author thanks his parents, Mr. and Mrs. Carlos Valenzuela, for making all of this possible.

TABLE OF CONTENTS

	<u>Page</u>
Certificate of Approval.....	ii
Acknowledgements.....	iii
List of Tables.....	vi
List of Figures.....	vii
Abstract.....	1
1. Introduction	
General Background.....	2
Statement of the Problem.....	11
Problem Approach.....	18
Bibliography.....	20
2. Description of Q-GERT.	
Introduction.....	21
The Q-GERT Analysis Program.....	24
Data Card Types.....	30
Program Inserts Using FORTRAN.....	31
Statistical Collection and Output.....	32
Data Collected on Q-nodes.....	34
Data Collected on Servers.....	34
Special Features.....	34
Q-GERT Symbols.....	37
Bibliography.....	40

3. Building Basic Models with Q-GERT.	
Introduction.....	41
A Basic First Model.....	41
Description of Q-GERT Output Reports.....	46
Embellishing the Basic Model.....	56
Bibliography.....	64
4. A General Model.	
Introduction.....	65
Statement of the Problem.....	65
The Model.....	69
Q-GERT Program Features.....	74
Discussion of Results.....	78
Potential Experiments with the Model.....	87
Summary.....	88
Bibliography.....	92
5. Enhancements to the General Model.	
Introduction.....	93
Haulage Times.....	93
Intersection Analysis.....	98
Truck Failure and Preventive Maintenance.....	99
Shovel Breakdowns and Maintenance.....	101
Conclusions.....	103
Bibliography.....	106
Vita.....	107

LIST OF TABLES

<u>Table</u>		<u>Page</u>
1	Estimated World and United States Production of Crude Metallic and Nonmetallic Ores and Coal by Sur- face Mining.....	3
2	Performance of System with Dis- patcher.....	83
3	Comparison of Shovel Performance for Two Values of Crossing Times of Intersection M and Fleet Size of 26 Trucks.....	85

LIST OF FIGURES

Figure		Page
1.1	A Typical Off-Highway Truck.....	8
1.2	A Schematic of Truck and Shovel System...	16
2.1	Q-GERT Development Cycle.....	21
2.2	Basic Nodes.....	23
2.3	The Q-GERT Cycle.....	25
3.1	Results for Basic Model.....	61
4.1	Schematic of Mine.....	66
4.2	Q-GERT General Model.....	71
4.3	Q-GERT Program Listing.....	77
4.4	Output Sample.....	79
4.5	Cost Curve.....	83
4.6	Prototype of Mine layout by O'Neil.....	89
4.7	Reduction of O'Neil Diagram.....	90
5.1	Typical Performance Curve.....	94
5.2	Haultime Calculation in UF.....	96

ABSTRACT

Surface is being extensively used as the main production system to mine a variety of essential raw materials in the world. This heavy utilization requires that managers be able to optimize the processing of enormous quantities of material. Many complex problems arise when it comes to finding the optimal procedure to allocate trucks, shovels, tractor-scrapers and other resources.

In this work the potential applicability of Q-GERT, a network modeling technique and a simulation language, to model truck-haulage problems is investigated.

A typical open pit mine that uses large equipment with a medium haul cycle was selected for analysis using Q-GERT, and results of several experiments are discussed. Emphasis is placed in demonstrating the powerful capabilities of the language to help determine those critical factors that have the greatest influence on costs, such as fleet composition, assignment of equipment, etc.

Chapter One

INTRODUCTION

1.1 General Background·

Surface mining is applied essentially to mine metallic ores like copper, iron and aluminum, and to nonmetallic ores like clays, gypsum, phosphate rock, sand, gravel and stone. Coal also is frequently obtained by surface methods.

In the United States, production by surface methods parallels that of the world (12) with one exception: about 90% of the copper and iron ore is mined from the surface, whereas elsewhere only half the copper ore is surface mined. Table 1 is a summary of the estimated world and United States production of crude metallic and nonmetallic ores and coal in 1969. It can be seen that two-thirds of the 9162 million tons are mined from the surface, ranging from a third of the coal, 57% of the metallic ores, and nearly all clays, stone, sand and gravel.

Table 1.
 Estimated World and United States Production of Crude Metallic and
 Nonmetallic Ores and Coal by Surface Mining, 1969*
 (Million Tons)

	World			United States			Total U. S. % of World
	Total	Surface	%	Total	Surface	%	
Metallic ores.. 1905		1088	57	490	413	84	26
Nonmetallic ores..... 1360		1088	80	200	164	82	15
Clay, stone, sand & gravel 2994		2994	100	1688	1651	98	56
Coal..... 2603		<u>998</u>	<u>34</u>	<u>528</u>	<u>183</u>	<u>35</u>	<u>18</u>
Totals and percents	9162	6168	67	2906	2411	83	32

*Data from U.S. Bureau of Mines sources. Some allowance has been made in world production for basic data on metallic, nonmetallic ores and coal which are unavailable from some Communist countries, and the estimates are considered to be reasonable. Some statistics from clay, stone, sand and gravel are not available from many countries, as well as those of the Communist Bloc. However, the table in its entirety does convey the order of importance of surface mining in the world.

†Except clay, stone, sand and gravel.

It generally is considered that surface mining is more advantageous than underground mining in recovery, grade control, economy, flexibility of operation, safety and working environment. There are, however, many deposits that are too small, irregular and deeply buried to be extracted economically by surface methods. Furthermore, even where mineralization extends to a greater depth in open pits, the rapidly increasing amount of overburden to be handled imposes economic limits beyond which mining either must be abandoned or converted from open-pit to underground.

There are various authorities who feel that future conditions will force a reversal of the trend from underground to surface mining--with a gradual shift back to underground. Changing public attitudes on environmental control will, to some extent, tend to increase the cost of surface operations. Laws might prohibit surface mining around urban areas. However, full ore recovery (for conservation of mineral wealth), coupled with caving control, pose numerous technical and cost problems that discourage consideration of underground operations. Perhaps the greatest force toward underground mining results from improved geological and geophysical techniques that will uncover mineral deposits beyond the economic depths of surface stripping.

But such shifts will be slow in coming unless pro-

ductivities in underground excavation methods escalate considerably more rapidly than in the past. Surface mining today has productivity rates of 100 tons per man-shift for small ferrous and nonferrous mines and up to 500 for large coal and industrial mineral operations, including all waster and ore handled. In comparison, underground mining achieves but 10 to 60 tons per man-shift, or about one-tenth that for surface methods.

The demand for mineral commodities in the future will intensify greatly not only in the USA but in the entire world as well, and consequently the production of these products must increase many times over. This pressing need for mineral raw materials will pose not a small technological challenge to satisfy it. Given the advantages of surface mining, it is expected that it will be extensively utilized as the main production system.

This heavy utilization, on the other hand, requires that managers be able to optimize the processing of enormous quantities of material. As O'Neil points out (2), "to mining companies, this means a larger enterprise with regard to plant and equipment, the reduction of high labor costs through medianization and automation, and finally the development of scientific methods to minimize management's risk in making decisions."

Materials handling is considered one of the most important problems with which the management must deal,

and a variety of solutions using a broad spectrum of techniques have been offered. Operation research techniques, ranging from linear programming to simulation, have been extensively tested and applied. This, together with an increasing usage of computers in mining operations, have provided management with valuable tools for the study and design of complex systems.

Although a great variety of equipment can be utilized to handle materials, the most widely used combination is truck and shovel.

Historically, open pit mines had employed rail haulage almost exclusively, it being the only reliable high-volume method available early in the 20th century. Inflexibility, however, hindered rail haulage from the beginning, as track relining at both the shovel and the waste dumps proved very costly and time-consuming (8).

Railroad haulage can be employed in surface mining for moving ore and/or waste out of the pit to a crushing plant or dump. For handling rought rock, the shovel-train system is excellent. The ideal application is in a large, but not excessively deep, open pit mine from which the coarsely blasted ore and waste must be transported over 4.8 Km (3 mi.). If, in addition, a high production rate is to be maintained, railroad haulage probably will afford a lower cost per ton transported.

However, as the pits deepen, the locomotive's maxi-

mum gradeability of about 4% becomes critical. The greater the differences in elevation involved, the greater are the track grade space requirements; for example, a 3% ramp out of a pit only 90 m is almost 3.3 Km long, thus making the cycle time uneconomically long.

Haulage trucks designed expressly for mine service were introduced about 38 years ago. The first models to arrive on the scene hauled about 15 tons and easily outperformed the modified highway trucks that had been used for mine haulage up to that time. The specialized trucks obviously had a greater potential for mining operations, and once this fact was recognized, competitive pressures in both the mining and truck-manufacturing industries drove the new vehicles toward ever more efficient performance.

Trucks that are built to operate in mines or pits, or in other types of excavation in which the use of public roads is not required, are not subject to any legal restrictions in size or weight.

Off-highway rear dump trucks may be 2.7 to 8 meters wide, 7.12 to 15 meters long, 3.4 to 5.5 meters high, with loading height (body sides) between 2.4 and 5 meters. Capacity range is from 12 to 190 tons, with body ratings up to 103 meters struck. Larger models are being tested. Empty weight (chassis and body) may equal the payload capacity, or be as low as 4/5 of it.

Construction is heavier than in highway trucks, in order to stand up under conditions of rough footing, heavy loads, and short hauls. Substantial amounts of high-strength steel may be used.

Top speed is usually 40 to 55 Km/hr, with some models going over 65. Road condition and tire wear limit practical speed.

Figure 1.1 shows a very big truck.

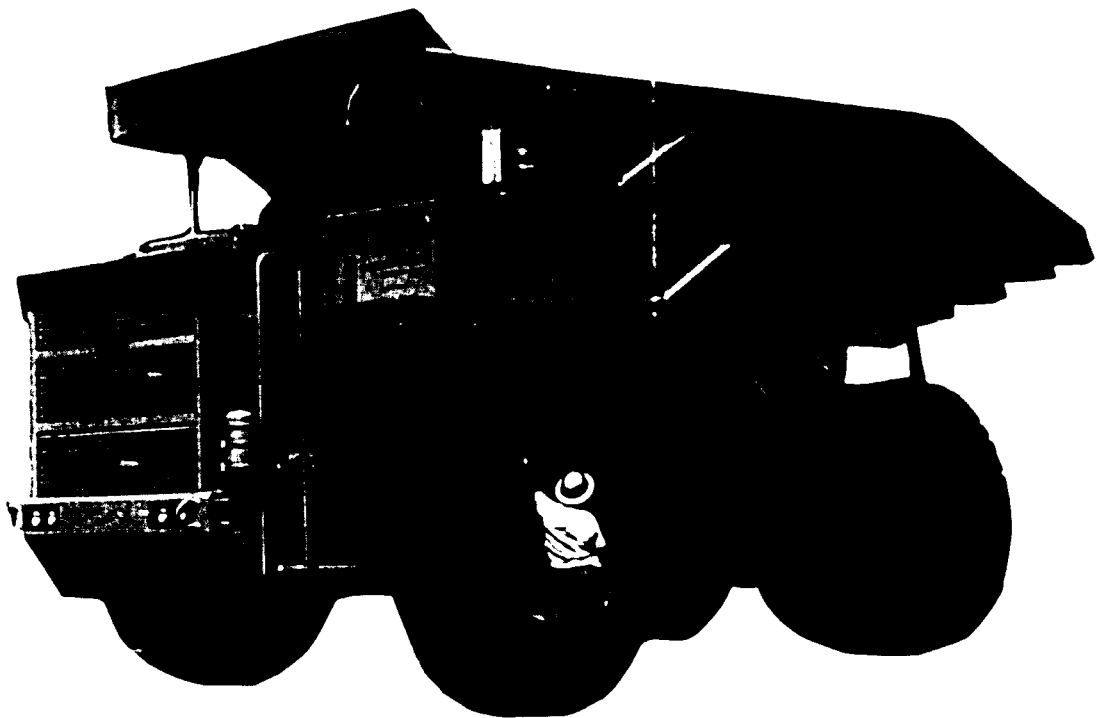


Fig. 1.1 A Typical Off-Highway Truck



Fig 1.1 Cont.

A large percentage of the off-highway trucks are diesel-electrics, and diesel automotive engines can frequently be of 1000 HP (746 KW).

A desired hauling capacity can be obtained by a few large trucks, or a larger number of small ones. Truck fleets may contain only one size unit or a variety.

A big truck should move dirt more cheaply than a small one of the same speed, particularly on long hauls. Its purchase and maintenance costs are usually lower on a per meter basis, and there is a definite saving in driver's wages.

On the other hand, a large truck requires substantially more room to maneuver and if it does not have it, may waste so much time getting placed to load or dump that its production will be smaller and more costly than that of a small truck.

Revolving shovels were the first important power excavators. Part-swing, steam-powered dipper models mounted on railroad cars or barges were in use over 130 years ago.

Other models--clamshell, dragline, and ice, were gradually developed. In the first 20 years of this century, steam power was largely replaced by internal combustion engines and electric motors.

Also, full-swing replaced part-swing, and crawler self-propelled mountings became standard. More recently,

rubber-tired carriers have become common.

Until around 1950 practically all these machines manipulated their buckets by means of cables (wire ropes), winding onto mechanically or electrically driven drums. Swing and travel depended on gear arrangements. Controls were mechanical, air, hand-or-foot-pressure hydraulic, or combinations of these systems.

Now, in 1978, production of cable and mechanical drives in backhoes has been largely replaced by hydraulics. Hydraulic-operated cranes are standard among smaller machines, and the hydraulic clamshell has gained acceptance. Also, the front end tractor loader (hydraulic) is competing successfully with the dipper shovel in many kinds of work. As a result, the importance of the cable-operated excavators has declined greatly, particularly among small machines. However, they are dominant as draglines, clamshells, and big cranes and dipper models, and are still very widely owned and used in all sizes and types. Bucket capacities of 107 m³ are not unusual.

1.2 Statement of the Problem.

In pursuing the reduction of materials handling costs in open-pit mining, management is often confronted with several problems when it comes to finding the optimal procedure to allocate trucks, shovels, tractor-scrappers and other resources. Usually production and system con-

figuration relationships are unknown in large operations and developing a mathematical model may become almost impossible; therefore, the vast majority of solutions attempted involve the use of computer simulation techniques.

It must be kept in mind, however, that the materials handling system is really a subsystem of a bigger one that embraces the overall mine operations. If the objective of a mining company is to produce a certain quality ore at minimum cost, the optimum operation is not achieved by minimizing all costs, since all phases of mining are interrelated. Production of blasting costs might well increase loading, hauling, and crushing costs out of proportion to the resultant savings in explosives. Hence, trade-offs are necessary. It is evident then that an optimum system is achieved through a blending of suboptimal components which, taken together, yield the desired result.

The main purpose of a haul-cycle study is to determine the size of the shovel and number of carriers which will minimize the cost of moving the specified amount of material from a given source to a given destination. In most mines, the distance between these two points varies as mining progresses, changing equipment requirements as a consequence.

Due to extreme complexity of large materials-handling systems, productivity effects seldom can be

predicted when operating procedures are changed. The haphazard growth of open-pit operations, and of equipment selection for materials handling, create new problems.

Solutions essentially are answers to the following questions:

- What size truck is best to use with a given shovel?
- What is the best assignment of trucks to shovels in the system?
- What is the influence of physical changes in the haulage layout on the system?
- What will be the total production and unit operating costs for a given fleet of trucks and shovels?
- How will the truck-dispatching system affect production and unit operating costs?

Various methods have been used in the past to analyze pit loading and truck haulage equipment systems. In April 1968, Dr. Lucien Duckstein (2), Professor of Systems Engineering at the University of Arizona and a consultant in operations research to Computer Research, Ltd., presented "Optimization of Open Pit Traffic Patterns" at the Seventh Annual Symposium on Operations Research and Computer Applications in the Mineral Industries, Colorado School of Mines, Golden, Colorado. This paper was a theoretical dissertation, purely mathematical in nature,

showing the mathematical framework that could be used in the analysis of traffic flow in open pit mines using various shovel-truck combinations. The particular question investigated was the potential value of using a truck-dispatching system as opposed to the classical method of assigning specific trucks to specific shovels. While solutions were possible only under considerably simplified assumptions (no failure of equipment, very simple distributions of haul times, etc.), the results indicated that a considerable savings could be realized by dispatching empty trucks to the next available shovel rather than to a preassigned one. (This policy is usually called continuous dispatching.)

However, the proper analysis of more complex situations, as Dr. Duckstein recognized, requires the using of different tools, like simulation. Through simulation, various operating policies and predictions may be tested before they are implemented in reality.

In 1964, an article by Madge (5) described the computer simulation of truck movement in an open pit mine operation consisting of two pits symmetrically located with respect to the concentration site. The purpose of the simulation was to help determine fleet requirements and to examine the effects of alternate ore removal procedures at the pits.

In 1966, O'Neil (8) presented a hybrid probabilistic-deterministic approach to the open pit material handling

problem. This model simulated more complex systems than previous models by considering transportation of material from multiple mine faces to multiple mine destinations and by being adaptable to a wide variety of mine designs. The model cycles trucks between their assigned loading and discharge points over a measured haulage route and up to 5 shovels and as many as 10 trucks per shovel can be operated in it. A special feature is the standard simulation of truck movements which enables each vehicle to perform according to its mechanical capabilities and the physical profile of the haul load. The computer program was written in DAFT, a dialect of basic IBM-FORTRAN developed by the computer center at Pennsylvania State University for use with its IBM 7074 computer system.

Figure 1.2 presents the basic aspects of the system. A single stage materials-handling scheme is employed for both ore and waste. Loaded trucks are directed to a crusher or stockpile if ore is being mined, while overburden is conveyed to a waste dump. Each shovel is assigned a given number of trucks with two associated haulage routes for waste or ore, respectively. Records are kept of all waiting times at loading and dumping points, and of any interference with truck travel on the haul roads (i.e., any moving queues), and a current journal of ore and waste production is maintained.

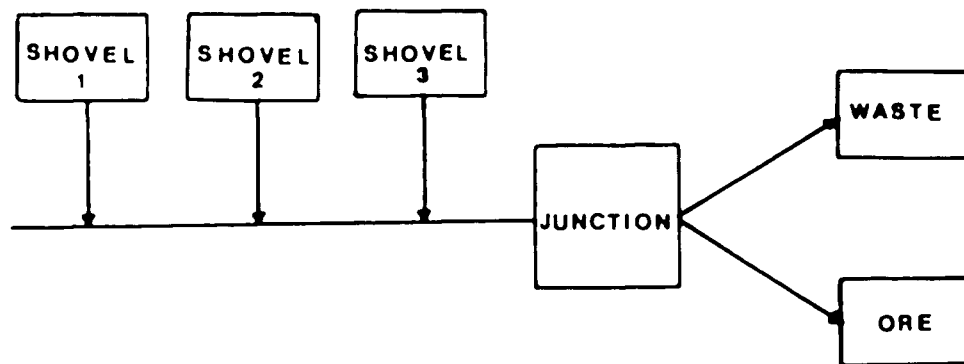


Fig. 1.2 A Schematic of Truck and Shovel System.

While O'Neil's model provided an excellent approach to a solution to this problem, several restrictions reduce the applicability of the simulator:

- The program execution time is too long, indicating that the program should be examined to minimize computational inefficiencies, as O'Neil himself points out.
- There is a fixed assignment of trucks to shovels, without permitting the user to study the effect of continuous dispatching.
- Equipment breakdown-time is not taken into account.
- Statistical capabilities are reduced.
- There is a lack of flexibility, in general. Even minor changes in the model to include non-standard features, mean cumbersome modifications to the program. Of course, this is true for any model implemented directly in a general-purpose computer language.

In 1967, Manula and Venkataramani (6) improved O'Neil's simulator while keeping its basic structure. The execution time was reduced, but nevertheless the other restrictions still apply.

Other solutions, although reduced in scope due to the complexity of the problem, have been suggested by several authors who have used dynamic programming, queueing theory, linear programming (including transportation), etc.

Various truck manufacturers have developed complex computer program for truck performance evaluation which are tailor-made for their specific haulage units and some of those programs have been reported in technical papers. These programs employ detailed input variables such as torque converter rpms in converter and lock-up gear condition for each shaft in the transmission, stall torques for each shaft, and transfer case, differential, and planetary gear ratios. However, nowhere in the literature are these computer programs completely revealed and they require details not readily available from all manufacturers.

Nevertheless, some material can be used as reference to compare results of a given model. For instance, the Caterpillar Tractor Company developed a program specifically designed for their machines entitled "Travel Time and Earth-moving Production Program." This program gives the time required for scrapers and bulldozers to travel each segment of the haul road with maximum and minimum load,

total time for the haul road and similar calculations for travel empty on the return load. These times represent maximum vehicle potential and ideal operator performance.

1.3 Problem Approach.

The main objective of this thesis is to show how Q-GERT networks can be applied with advantage to analyze truck-haulage problems.

As Pritsker (4) points out:

Q-GERT is a network modeling vehicle and a computer analysis tool. GERT is an acronym for Graphical Evaluation and Review Technique. The Q is appended to indicate that queueing systems can be modeled in graphic form. The similarity of names GERT and PERT is used to indicate that Q-GERT can be used to model projects consisting of sets of activities. In fact, GERT generalizes PERT concepts and Q-GERT augments GERT with the addition of queueing and decision capabilities. A fundamental contribution of Q-GERT is its method for graphically modeling systems in a manner that permits direct computer analysis. The Q-GERT Analysis Program has been developed to provide this computer analysis. .

Q-GERT has been designed, developed and used for studying the procedural aspects of manufacturing, defense and service systems. It satisfies the need for a network approach to the modeling of systems that involve procedural, risk and random elements. Q-GERT networks are models of systems that consist of activities, servers and queues. Q-GERT employs simulation procedures to analyze a Q-GERT network, and the Q-GERT Analysis Program can be viewed as a simulation language, much like GPSS.

The Q-GERT Analysis Program was written in ANSI standard FORTRAN IV and can be run on

any computer that has a FORTRAN IV compiler. Versions of the program to accommodate 100, 200 or 1000 word networks have been developed.

In this work we investigate the potential applicability of Q-GERT to model truck-haulage problems. Of course, a truck-and-shovel model can be easily adjusted to evaluate scrapers and pushers, trucks and conveyors, etc.

BIBLIOGRAPHY

1. Atkinson, T., Selection of Open-Pit Excavating and Loading Equipment, Institution of Mining and Metallurgy, May 1971.
2. Duckstein, L., Optimization of Open Pit Traffic Patterns, 7th Annual Symposium on Operations Research and Computer Applications in the Mineral Industries, Colorado School of Mines, Golden, Colorado, 1968.
3. Gibbs, L. W., et al., System Analysis for Truck and Shovel Selection, SME Transactions, December 1977.
4. Kress, Ralph H., What Big Trucks Need to Grow On, Society of Mining Engineers, May 1971.
5. Madge, D. N., Simulation of Truck Movements in an Open Pit Mining Operation, Canadian Operations Research Society Journal, vol. 2, No. 1, 1964.
6. Manula, C. B., and Venkataramani, R., Open-Pit Truck Haulage Simulator (SIMULATOR II), College of Earth & Mineral Sciences, Pennsylvania State University, 1967.
7. Nichols, H. L., Moving the Earth, North Castle Books, 1976.
8. O'Neil, T. J., Computer Simulation of Materials Handling in Open Pit Mining, M. S. Thesis, Pennsylvania State University, 1966.
9. Pritsker, A. A. B., Modeling and Analysis Using Q-GERT Networks, John Wiley & Sons, 1977.
10. Shannon, R. R., Systems Simulation, the Art and Science, Prentice-Hall, 1975.
11. Sisselman, R., Copper: Iran's Latest Desert Bonanza, E/MJ, February 1978.
12. SME Mining Engineering Handbook, Society of Mining Engineers of AIME, 1973.

Much like GPSS, Q-GERT (2) Analysis Program is a transaction-oriented simulation language where the structure of the system being simulated is described in the form of a network diagram with a fixed set of predefined node types. Each node represents a specific action that is characteristic of some basic operation that can occur in the system. Connections between the nodes of the diagram indicate the sequence of actions that occur in the system.

Flowing through the network are items referred to as transactions. Transactions are directed through the network according to the branching characteristics of the nodes. Transactions can represent physical objects, information, or a combination of the two. Different types of nodes are included in Q-GERT to allow for the modeling of complex queueing situations and project management systems. Activities can be used to represent servers of a queueing system and Q-GERT networks can be developed to model sequential and parallel service systems. The nodes and branches of a Q-GERT model describe the structural aspects of the system. A process approach is taken in which the flow of a transaction is modeled.

Transactions originate at source nodes and travel along the branches of the network. Each branch has a start node and an end node as shown in the next figure 2.2.

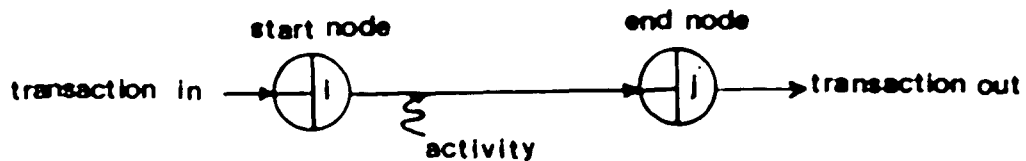


Fig. 2.2 Basic Nodes

Transactions moving across a branch are delayed in reaching the end node associated with the branch by the time to perform the activity that the branch represents. When reaching the end node, the disposition of the transaction is determined by the node type, the status of the system, and the attributes associated with the transaction. The transaction continues through the network until no further routing can be performed. Typically, this occurs at sink nodes of the network but may occur at other nodes to allow for the destruction of information flow.

Transactions have attribute values that allow different types of objects (or the same type of object with different attribute values) to flow through the network. Procedures are available to assign and change attribute values of transactions at the various nodes of the network.

As transactions flow through the network model, statistics are collected on travel times, the status of servers and queues, and the times at which nodes are released. Thus, a statistical data collection scheme is embedded directly in a Q-GERT network model. The Q-GERT Analysis Program employs a simulation procedure to analyze

the network. The simulation procedure involves the generation of transactions, the processing of the transactions through the network, and the collection of statistics required to prepare automatically a summary report as dictated by the Q-GERT network model.

Pritsker (2) presents an overview of the components involved in a Q-GERT systems analysis. The Q-GERT symbol set is provided by the designer. The analyst brings to the systems analysis a knowledge of the system and the scenarios to be evaluated. Based on the scenarios and the purpose for model building, the Q-GERT symbols are combined into a network form. At this point, discussions can be held and the network revised and embellished.

The next step is to describe the Q-GERT network in a computer readable form. The Q-GERT Analysis Program specifies the procedure to enter the network in the computer.

The Q-GERT Analysis Program provides summary reports of the information requested by the analyst.

2.2 The Q-GERT Analysis Program

The Q-GERT Analysis Program employs discrete event

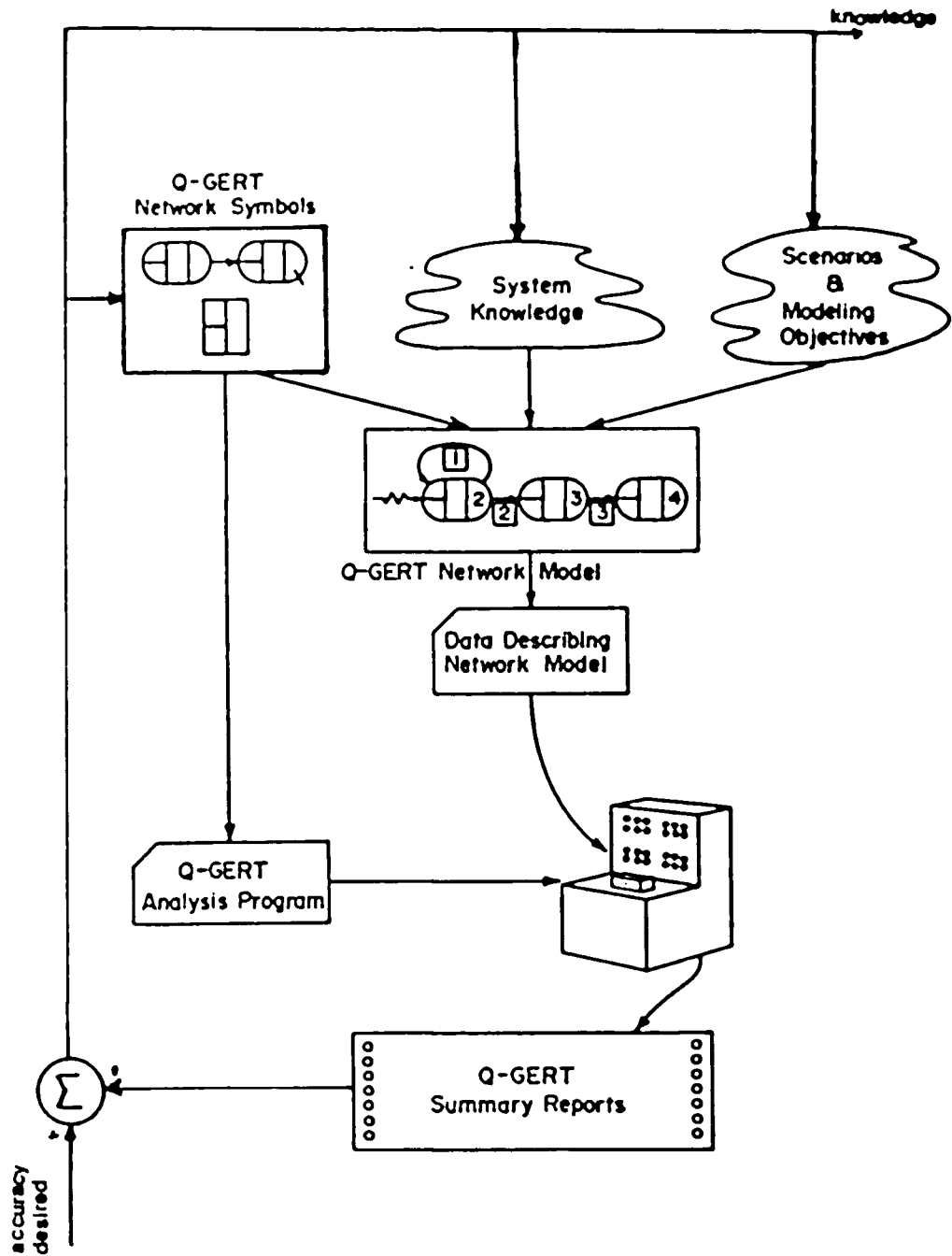


FIG. 2.3 The Q-GERT Cycle

procedures to simulate the flow of transactions through a network. Basically, only one event type is included in the program: the arrival of a transaction at a node. All the decision logic that can occur when a transaction arrives at a node are included in the program and the appropriate actions are taken based on the network model provided by the analyst. This includes the collection of statistical quantities.

The analysis of a Q-GERT network is performed on a digital computer by the Q-GERT Analysis Program. The Q-GERT Analysis Program is written in ANSI FORTRAN IV (2) and has been run on a wide class of computers.

The Q-GERT Analysis Program employs simulation techniques to analyze the flow of transactions through the network in order to obtain statistical estimates of the quantities prescribed on the Q-GERT network. The estimates can be obtained from a single simulation run or over a specified number of runs. The completion of one simulation run of the network can be defined by specifying any of the following quantities: 1) the number of transactions reaching sink nodes; 2) the number of sink nodes to be released; and 3) a specified time period. These values and other information concerning the analysis procedure are specified on one data input card to the Q-GERT Analysis Program. One data card is used to indicate the end of data input. All other data cards for the Q-GERT Analysis

Program are used to describe the Q-GERT network elements.

To start the analysis program after the network has been described (inputted), each source node for the network is evaluated. This could be considered as a special start of simulation event. However, we consider the start of simulation to be equivalent to an arrival of a transaction at the source node which causes the source node to be released. At each source node, transactions are generated and marked and then are routed according to the branching characteristics prescribed for the source node. The performance of activities associated with each branch selected from the source nodes are simulated by selecting a time for the activity in accordance with the distribution type and parameter values prescribed for the activity. An event corresponding to the arrival of the transaction at the end node of the activity is scheduled and placed on an event calendar.

When all source nodes have been considered in this fashion, time is advanced to the time of the next (first) event which is removed from the event calendar. The type of node to which the transaction is arriving is first examined. If it is not a Q-node, the number of incoming transactions to release the node is decreased by 1. If the node is not released, that is, it requires more incoming transactions, no further action is taken and time can be advanced to the next event time. If the node is

released, statistics are collected if necessary, marking is performed if necessary, and the transaction is routed along the branches emanating from the node just released. If the node has deterministic branching, identical transactions are routed along each branch emanating from the node. If the node has probabilistic branching, a selection of one branch is made using a pseudo-random number generator. For each branch selected, an activity time is obtained and the transaction is scheduled to arrive at the end node at the current time plus the activity time. This arrival-of-transaction event is placed on the event calendar. After all branches have been selected and their associated events scheduled, the next event is removed from the event calendar and the above process is repeated.

When a transaction arrives at a sink node, a check is made to see if the simulation run is completed. If not, the process continues. If the run is completed, summary statistics for one run of the simulation are stored. In addition, each time an event is taken from the event calendar, the time of the event is compared to a total time allocated for the simulation. If all time has expired, then the simulation run of the network is also considered to be completed, and statistics on the run are stored.

When a transaction arrives at Q-node, a slightly more complex decision process is involved. First, a check is made to see if the queue is full. If it is, the transaction

either balks from the Q-node or blocks its current service activity. If it balks and there is no balking node prescribed, the transaction is deleted from the system. If a balking node was prescribed, the transaction is routed directly to the balking node. If blocking occurs, the service activity which just completed processing the transaction is not made available for processing another transaction.

If the queue is not full but the servers following the queue are all busy, the transaction is placed in the queue in its proper position. If a server is available, the transaction is scheduled to arrive at the node following the service activity at the current time plus the service time. Statistics are maintained on the number of transactions in the queue and the busy time for servers.

When a transaction completes a service activity, additional processing must be performed. Not only must the transaction be routed to the end node of the service activity, but the disposition of the service activity must be considered. The logic involved in determining the disposition of the service activity involves examining the Q-node associated with the service activity. If no transactions are in the Q-node, the service activity is made idle. If a transaction is waiting at the Q-node, it is removed from the Q-node and the transaction is routed along the service activity, that is, the transaction is scheduled

to arrive at the end node of the service activity. If, before removing the transaction from the Q-node, the Q-node was at its maximum capacity, a check is also made to unblock any service activities preceding the Q-node.

When two events have the same time of occurrence, a complex tie-breaking procedure is used. For an event that is to occur immediately, that, the activity associated with the event involves no delay time, a last-in, first out (LIFO) rule is employed. For events that are scheduled to occur at a future time, the tie-breaking rule is first-in, first-out (FIFO).

2.3 Data Card Types.

The data cards required by the Q-GERT Analysis Program are uniquely identified with a 3 character alphanumeric identification (ID) specified in the first field of the card. More characters can be used as the card identifier, but only the first three characters are significant.

An alphabetical listing of the data card types is presented below.

<u>Card Code Name</u>	<u>Brief Description</u>
ACT	<u>Activity</u> description
BEG	<u>Begin</u> network analysis
COL	<u>Collect</u> user statistics labels
DEF	<u>Definition</u> of subnetwork
DEL	<u>Delete</u> card from subnetwork
DUP	<u>Duplicate</u> last subnetwork defined
ESN	<u>End</u> subnetwork definition
FIN	<u>Finish</u> of all Q-GERT input
GEN	<u>General</u> project and network data

<u>Card Code Name</u>	<u>Brief Description</u>
HIS	<u>H</u> istogram data for user statistics
INS	<u>I</u> nsert a card in a subnetwork
LIN	<u>L</u> ink a subnetwork node to network
MAT	<u>M</u> atch node description
MOD	<u>M</u> odification of network data
PAR	<u>P</u> arameter data values
QUE	<u>Q</u> ueue node description
REG	<u>R</u> egular node description
REP	<u>R</u> eplace a card in a subnetwork
SEE	<u>S</u> eeds for random number streams
SEL	<u>S</u> elector node description
SIN	<u>S</u> ink node description
SOU	<u>S</u> ource node description
STA	<u>S</u> tatistics node description
TIM	<u>T</u> ime persistent variable data
TRA	<u>T</u> race of specific nodes
VAS	<u>V</u> alue <u>a</u> ssignment to attributes

The order in which data cards are submitted for processing is very flexible. The first card of a deck describing a network must be a BEG card or a FIN card. BEG cards indicate the end of one network description and the beginning of the cards that describe a new network. A FIN card indicates the end of the last network description. The only other requirement regarding card order is that the start node referenced on an ACT (activity) card must be defined prior to the ACT card. Node definitions are given by REG, SOU, SIN, STA and QUE cards.

2.4 Program Inserts using FORTRAN.

The capability of inserting user's program written in FORTRAN is provided in Q-GERT. Three subprograms are included in Q-GERT to assist the user in the interface with Q-GERT. These are:

- SUBROUTINE UI, To initialize user defined variables and to create special initial conditions.
 - SUBROUTINE UO, to output user specified information and to perform any end-of-run computations.
 - FUNCTION UF, to contain user written procedures.
- Access to a variety of Q-GERT subroutines and variables is possible, thus making the language extremely flexible. A complete listing of all subprograms that can be accessed by the user is given in (2).

2.5 Statistical Collection and Output.

The Q-GERT Analysis Program maintains information on statistics nodes (STA), sink nodes (SIN), Q-nodes (QUE), and servers. All the data collected are presented in the Q-GERT summary report. In addition, the capability of inserting programs written in FORTRAN is provided, so as to allow the user to obtain statistics not included in Q-GERT's standard output. Access to several subroutines is possible. Five types of variables can be associated with statistics and sink nodes. Each of the variables refers to a time that a specified node is released. The five types of variables are:

1. Time of first release (F).
2. Time of all releases (A).
3. Time between releases (B). The time of first

release is used only as a reference point for the first value of the time between releases.

4. Interval statistics (I). This statistic relates to the transaction that releases the node. It records the interval of time from the marking of the transaction to the release time of the statistics node. Recall that all source nodes are mark nodes. If a transaction passes through two or more mark nodes before reaching a node at which interval statistics are collected, it is the time of last marking that is referenced when the time interval is computed.
5. Delay statistics (D). This statistic relates to nodes at which transaction requirement is greater than one. The delay time is the time interval from the first arrival of a transaction at the node until the node is released. A delay time is computed each time the node is released. Thus, if a node requires three incoming transactions for each release, a D specification for statistics would compute the time from the first transaction arrival until the arrival time of the third transaction. If the node was released twice, then the second delay time would be the time between the arrival of the fourth transaction and the time of arrival of the sixth transaction.

For each of the above five types of variables, the Q-GERT Analysis Program obtains estimates for the mean and standard deviation of the variables.

2.6 Data Collected on Q-nodes.

The variables of interest associated with a Q-node are:

- The average number in the Q-node.
- The average number of transactions balking from the Q-node.
- Minimum and maximum number of transactions ever in the Q-node.

2.7 Data Collected on Servers.

The Q-GERT summary report yields the following statistics associated with servers:

- Average server utilization.
- Minimum and maximum values of average server utilization.
- Longest consecutive period of time that the server is busy or idle.
- Fraction of time the server is blocked.

2.8 Special Features.

Q-GERT also provides the user with some special features to facilitate the modeling and programming work.

Some of them are:

- Associating attributes with transactions.
 - a) Assigning attributes to transactions.
 - b) Ranking transactions in Q-nodes based on attribute values.
 - c) Routing transactions based on attribute values.
- S-nodes: Nodes to select from among available servers and/or queues.
 - a) Routing transactions to parallel Q-nodes.
 - b) Removing transactions from parallel Q-nodes.
 - c) Selecting from among parallel nonidentical servers.
 - d) Assembling different transactions prior to service.
- Match node: A node that halts the flow of transactions until a set of transactions with a common attribute value has arrived to it.
- Nodal modification: The conditional replacement of a node by another node.

The number of attributes associated with each transaction in the network is defined by the modeler through data input. In addition to the user defined attribute values, a transaction's mark time is automatically considered as one of its attribute values. Just as with the mark time, the assignment of attribute values to a transaction is made at a node. Any node in a Q-GERT network can be used to perform

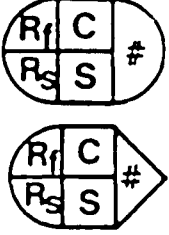
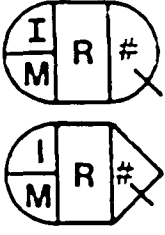

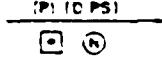
this assignment function. Only when a transaction passes through the node will the values be assigned.

When specifying an assignment of an attribute value to a transaction, the attribute number and the computational procedure for obtaining the value for the attribute are prescribed. The attribute number to which a value is to be assigned is set by the modeler based on the situation that he is modeling. The modeler prescribes the value to be assigned in the same manner as he prescribes the time to perform an activity. That is, he specifies a function type and a parameter identifier. Thus, attribute values can be obtained from any of the following functions:

- BE, beta distribution;
- BP, beta distribution fitted to three parameters;
- CO, constant;
- ER, Erlang distribution;
- EX, exponential distribution;
- GA, gamma distribution;
- IN, incremental function;
- LO, lognormal distribution;
- NO, normal distribution;
- PO, Poisson distribution;
- TR, triangular distribution;
- UF, user function; and
- UN, uniform distribution.

The symbolism associated with value assignment (VAS) is placed in the central portion of the node just prior to the node number. For each assignment to be made, three items of information are required: the attribute number; the label for the function type; and the parameter identifier.

2.9 Q-GERT Symbols.

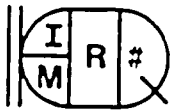
Symbol	Concept	Definition
	Regular Node	<p>R_f is the number of incoming transactions required to release the node for the first time.</p> <p>R_s is the number of incoming transactions required to release the node for all subsequent times.</p> <p>C is the criterion for holding the attribute set at a node.</p> <p>S is the statistics collection type of marking. $\#$ is the node number.</p>
	Queue Node	<p>I is in the initial number of transactions at the Q-node.</p> <p>M is the maximum number of transactions permitted at the Q-node.</p> <p>R is the ranking procedure for ordering transactions at the Q-node.</p> <p>$\#$ is the Q-node number.</p>
		Pointer to a source node or from a sink node.
		<p>P is the probability of taking the activity (only used if probabilistic branching from the start node of the activity specified).</p> <p>D is the distribution or function type from which the activity time is to be determined.</p> <p>PS is the parameter set number (or constant value) where the parameters for the activity time are specified.</p>

Symbol	Concept	Definition
--------	---------	------------

\square is the activity number.
 \textcircled{N} is the number of parallel servers associated with the activity (only used if the start node of the activity is a Q-node).



Routing of a transaction that balks from a Q-node. This symbol can not emanate from a regular node.



Blocking indicator (only used with Q-nodes that can force preceding service activities to hold transactions because the Q-node is at its maximum capacity).



indicates deterministic branching from the node.



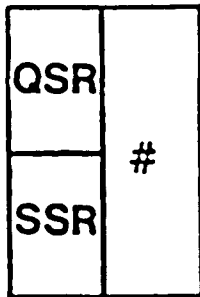
indicates probabilistic branching from the node.



Nodal Modification



is the activity number causing nodal modification.
 N_1 is the node number to be replaced when activity # is completed.
 N_2 is node number to be inserted when activity # is completed.



Selector node or S-node

QSR is the queue selection rule for routing transactions to or from Q-nodes (see Table 5-2).
 SSR is the server selection rule for deciding which server to make busy if a choice exists (see Table 5-3).
 # is the S-node number.

Symbol	Concept	Definition
	Balking	Blocking from an S-node.
	Match Node	# is match node number. Transactions are routed from N_1 to N_3 and N_2 to N_4 when a match occurs.
	Routing Indicator	Routing indicator for transaction flow to or from Q-nodes to S-nodes or Match nodes.
	Assembly by S-nodes	ASM is the queue selection rule that requires transactions to be assembled from two or more queues.
	Blocking	Blocking at an S-node.

(#) Acknowledgement : This chapter was mainly composed of excerpts from (2), and was included to provide an insight into the language and modeling technique.

BIBLIOGRAPHY

1. Pritsker, A. A. B., GERT: Graphical Evaluation and Review Technique, The Rand Corporation, RM-4973-NASA, Santa Monica, California, April 1966.
2. Pritsker, A. A. B., Modeling and Analysis Using Q-GERT Networks, John Wiley & Sons, 1977.
3. Pritsker, A. A. B., The GASP-IV Simulation Language, John Wiley & Sons, 1974.
4. Shannon, Robert E., Systems Simulation, the Art and Science, Prentice-Hall, 1975.
5. Whitehouse, Gary E., Systems Analysis and Design Using Network Techniques, Prentice-Hall, 1973.
6. Whitehouse, G. E., Wechsler, B. L., Applied Operations Research: A Survey, John Wiley & Sons, 1976.

Chapter Three

BUILDING BASIC MODELS WITH Q-GERT

3.1 Introduction.

The information provided in Chapter 2, although not sufficient to master the concepts underlying Q-GERT, can be used to understand some basic models of mining operations.

3.2 A Basic First Model.

A first basic model designed with the idea of showing how Q-GERT handles a given system will be presented. Let us assume that in a truck-haulage system representing a given open-pit mining operation, a subsystem is identified, consisting of a single shovel and 4 trucks assigned to it. The shovel is currently mining ore which must be conveyed by the trucks to a crusher or destination point. Trucks must travel a certain distance; the road is in good condition and ample enough to have two lanes, but not ample enough to allow passing of one truck to another slower one. Two queues can form, one at the shovel waiting for service and the other at the crusher, waiting for dump. Only one truck at a time can be dumping at the crusher. According to time studies conducted, the following data describes the operations in the system:

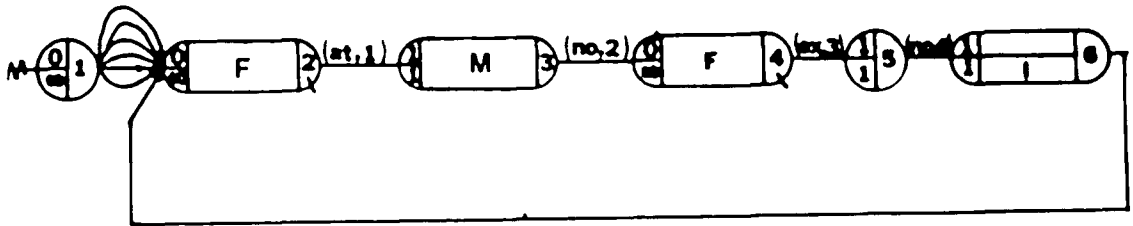
Loading Time: lognormal, with mean 2.0 min. and standard deviation 0.5 min.

Dumping Time: Exponential, with mean 0.70 min.

Travel Times: $\mu = 4$ min., $\sigma = 0.9$; from shovel to crusher
 $\mu = 1.5$ min., $\sigma = 0.2$ min.; from crusher to shovel.

The number of trucks in the system is 6, all of them waiting for loading service at the beginning of the simulation. A shift of 480 minutes of uninterrupted service will be simulated. Truck and shovel availability is 100%.

The Q-GERT network representing this situation is shown below:

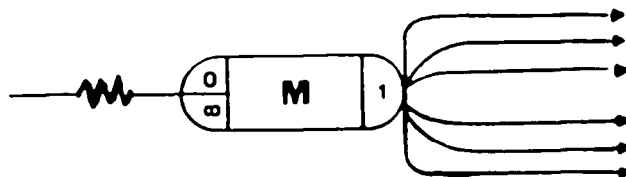


Let us explain this basic model. The passage of time is represented in a Q-GERT network by a branch. Branches are the graphical representation of activities. That is to say, operations like loading, traveling and dumping are activities. An activity may last zero time which is the case from activity going from node 6 to node 2.

The arrival of trucks to the system can be modeled

by 6 branches leaving node 1. Node 1 represents, then, a source node and it will be used only once in this model, just to generate the 6 transactions needed for the simulation.

There are several important features that must be noted about the graphical representation. For instance, if we reproduce the source node below, we can see that label 1 has been assigned to it. The six branches leaving node 1 represent trucks and since they are not emanating



from a Q-node, they cannot be taken as service activities. The letter M represents a mark. Each transaction is marked upon its arrival at node 1, and this time will be used later in the model to calculate the haulage cycle time.

The number 0 in the left upper corner of the node is the initial number of transactions required to release node 1. The "number" ∞ represents the subsequent number of transactions required to release node 1.

Nodes similar to 1 are nodes 3, 5, and 6. Node 6 is a statistical node. Since we wish to record the total travel time (that is, shovel-crusher-shovel), we request the calculation of an interval statistic. An interval statistic is specified at a node by placing an "I" in

the lower center portion of the node. This specification causes Q-GERT to collect statistics on the interval of time between the marking ("M") of a transaction and its arrival at the node where the statistical calculation is requested.

Nodes 2 and 4 are Q-nodes. On the left hand side of each node there is space for two pieces of information specified as 0 and infinite respectively.

The zero represents the initial number of transactions in the queue and infinity represents the maximum number of transactions allowed in the queue. At the center of each Q-node, there is a symbol ("F"=First-in, first-out) which is indicating the procedure for ranking transactions in the queue. In other words, the information contained in nodes 2 and 4 are indicating that there are no transactions at the Q-nodes initially, that an infinite queue is allowed to form before the server and that transactions will reside in the queue on a first-in, first-out basis. It must be noted that Q-nodes have different information specified on their input side than non Q-nodes, and that Q-nodes only provide storage space and do not cause explicit time delays for transactions. The hash mark in the lower righthand corner identifies a Q-node.

Let us now consider the different activities in the model. Each activity in a Q-GERT network is assigned

a time description. The time is specified by a function type and a parameter identifier. For function types that relate to random variables, the identifier is a parameter set that points to where the values of the parameters for the function are maintained (specifically a row in an array). For example, the activity going from node 2 to 3 will represent a loading operation and is characterized by the LOGnormal distribution whose parameters are stored in parameter set 1. This specifies that the activity time is a sample from a lognormal distribution. If instead of LOGnormal distribution, the loading time was a constant, then the parameter identifier would be the value of the constant.

When an activity has no function on its arrow, it means that the time needed to perform such activity is constant with value zero.

The numbers in the square boxes and circles deserve an explanation. The number in the square box below the arrow represents the activity number provided by the modeler to identify uniquely that activity. If absent, Q-GERT assigns its own activity number. The number in the circle applies only to service activities and is the number of parallel servers on that branch. The default value is always one.

Now we are in position to explain the model itself. Node 1, the source node, generates 6 trucks that join on

a FIFO basis the queue in front of the shovel represented by activity 1. That queue identified by node number 2 has an infinite capacity and an initial number of zero.

Loading time is performed on activity 1 with 1 server in a time sampled from a lognormal distribution.

Node 3 is simply a mark-node which will be used to calculate the haulage cycle.

After a travel time given by a normal distribution whose parameters are stored in parameter set 2, trucks arrive at the crusher where they must form a queue if it is already occupied. That queue is represented by node 4 and is ranked FIFO.

Dumping time is an exponential distributed variable. Node 5 is there only to differentiate dumping and return trip activities. Once a truck has already completed a round trip and is back to the shovel, node 6 is in charge of collecting statistics on travel time.

3.3 Description of Q-GERT Output Reports. (*)

Q-GERT maintains automatically statistics on sink nodes, Q-nodes, servers and statistics nodes. All the data collected are presented in the Q-GERT summary report.

A summary of output definitions will be given below.

****FINAL RESULTS FOR FIRST SIMULATION****

TOTAL ELAPSED TIME--total time required for the first

* Taken from (2), pages 81-87.

simulation run

****NODE STATISTICS****--headings for node statistics

NODE--sink or statistics node number

LABEL--eight character name associated with node

AVE.--the estimate of the mean time. The time could be a release time, a delay time or an interval of time (see STAT TYPE below).

STD.DEV.--the standard deviation of the time

NO OF OBS.--number of observations obtained during the first simulation run for the statistical variable of interest

STAT TYPE--statistics type specified for the node by the user, that is: F,A,B,I or D.

****NUMBER IN Q-NODE****--section of the report which provides information on Q-node statistics

NODE--the Q-node number for which statistics are to be printed

LABEL--a user supplied name associated with the Q-node

AVE.--the time weighted average number of transactions in the Q-node for the first simulation run

MIN.--the minimum number of transactions in the Q-node for the first simulation run

MAX.--the maximum number of transactions in the Q-node for the first simulation run

****SERVER UTILIZATION****--heading to indicate server utilization section of the report

SERVER--the activity number associated with the branch
representing the service activity

LABEL--a user supplied name associated with the service
activity

NO.PARALLEL SERVERS--the number of parallel servers
represented by the activity

AVE.--For single servers, that is, the number of parallel
servers is 1, this quantity is the fraction of
time the server is busy. For multiple servers,
the average number of busy servers during the
first simulation run.

MAX.IDLE(TIME OR SERVERS)--for a single server activity,
the largest time period for
which the single server was
idle. For a multiple ser-
ver activity, the largest
number of servers who are
idle simultaneously.

MAX.BUSY(TIME OR SERVERS)--for a single server activi-
ty, the largest time period
that the single server was
busy. For a multiple ser-
ver activity, the largest
number of servers that were
simultaneously busy.

****BLOCKED TIME PER UNIT TIME****--section of report where
statistics on the time a
server is blocked per
unit time is reported

SERVER--the server number for which blocking statistics
are being reported

LABEL--a user supplied name associated with the service
activity

AVE.--the total time the server was blocked divided by
the total elapsed time. This represents the frac-
tion of time the server was blocked.

LONGEST PERIOD BLOCKED--the longest consecutive time
that the server was in a
blocked status

****NUMBER BALKING PER UNIT TIME****--a heading to indicate
the balking section of
this report

NODE--the number of the node from which balking is oc-
curring

LABEL--the user supplied name that describes the Q-node
from which balking is occurring

AVE.--the total number of transactions balking from the
Q-node divided by the total elapsed time for this
run. This represents the balking rate.

RESULTS FOR RUN i--this report provides a summary of statistics obtained on run i. The values printed on this report have the same definitions as those given for the report on "FINAL RESULTS FOR FIRST SIMULATION."

FINAL RESULTS FOR n SIMULATIONS--heading for defining n, the number of runs for which this report is a summary

NODE STATISTICS--heading for node statistics

NODE--sink or statistics node number

LABEL--the user supplied name that describes the node

PROBABILITY--an estimate of the probability that the node is released during a run and is computed as the total number of runs on which this node was released at least once divided by the total number of runs.

AVERAGE--an estimate of the mean time requested for the node. This is an average of all values collected over all runs.

STD.DEV.--the standard deviation of the time requested for the node

SD OF AVE--the standard deviation divided by the square
root of the number of observations

MIN.--the minimum value observed for the node over all
runs

MAX.--the maximum value observed for the node over all
runs

STAT TYPE--statistics type specified for the node user

****AVERAGE NUMBER IN Q-NODE****--heading to indicate the
section of report dealing
with statistics on the
average number in Q-nodes.
The variable of interest in
this section of the report
is the average number in a
Q-node as recorded during
one complete run.

NODE--the Q-node number for which statistics are to be
printed

LABEL--a user supplied name associated with the Q-node

AVE.--the average over multiple runs of the one run
time weighted average number of transactions in a
Q-node

STD.DEV.--the standard deviation of the average number
of transactions in a Q-node

SD OF AVE--the standard deviation divided by the square
root of the number of observations

NO. OF OBS.--number of runs

MIN.--the smallest average number of transactions in a
Q-node obtained from one run

MAX.--the largest average number of transactions in a
Q-node obtained from one run

****NUMBER IN Q-NODE****--heading to indicate values are
obtained regarding the number of
transactions in a Q-node (as op-
posed to the average number of
transactions in a Q-node)

MIN.--Smallest number of transactions that were in the
Q-node in any simulation run

MAX.--the largest number of transactions in the Q-node
in any simulation run

****AVERAGE SERVER UTILIZATION****--heading to indicate the
section of the report
dealing with statistics
on the average server
utilization

SERVER--the activity number associated with the branch
representing the service activity

LABEL--a user supplied name associated with the service
activity

NO. PARALLEL SERVERS--the number of parallel servers
represented by the activity

AVERAGE--the average over multiple runs of the average server utilization obtained on a single run

STD.DEV.--the standard deviation of the average server utilization obtained on a single run

SD OF AVE--the standard deviation divided by the square root of the number of observations

NO. OF OBS.--number of runs

MIN.--the smallest average server utilization obtained from one run

MAX.--the largest average server utilization obtained from one run

****EXTREME VALUES****--section of the report which deals with extreme values for server utilization which may have occurred on any run

MAX.IDLE (TIME OR SERVERS)--if the number of parallel servers is 1, this value is the longest consecutive period of time that the server was idle on any of the runs. If the number of parallel servers is greater than 1, this value is the maximum number of servers that were simultaneously idle on any one of the runs.

MAX.BUSY(TIME OR SERVERS)--if the number of parallel servers is 1, this value is the longest consecutive period of time that the server was busy on any of the runs. If the number of parallel servers is greater than 1, this value is the largest number of simultaneously busy servers on any run.

****AVERAGE BLOCKED TIME PER UNTI TIME****--heading to indicate the section of the report in which average blocking statistics are reported

SERVER--the server number for which blocking statistics are being reported

LABEL--a user supplied name associated with the service activity

AVE.--the average of the fraction of time the server was blocked on a single run

STD.DEV.--the standard deviation of the fraction of time the server was blocked on a single run

SD OF AVE--the standard deviation divided by the square root of the number of observations

NO.OF OBS.--number of runs

MIN.--the smallest fraction of time the server was
blocked on any of the runs

MAX.--the largest fraction of time the server was
blocked on any of the runs

****BLOCKED TIME OF SERVER****--heading to indicate the sec-
tion of the report dealing
with data on blocked time of
servers

LONGEST PERIOD BLOCKED--longest consecutive period of
time that the server was blocked
on any of the runs.

****AVERAGE NUMBER BALKING PER UNIT TIME****--heading
to indicate the section of the report in which
average balking statistics are reported

NODE--the node number for which balking statistics are
being reported

LABEL--a user supplied name associated with the node
from which balking occurs

AVE.--the average of the balking rates obtained from
each run

STD.DEV.--the standard deviation of the balking rates
obtained from each run

SD OF AVE--the standard deviation divided by the square
root of the number of observations

NO. OF OBS.--number of runs.

3.4 Embellishing the Basic Model.

As described in the previous section, Q-GERT can give a great amount of information about the system's behavior, but in some practical cases that must be complemented by non-standard statistics. For instance, we would like to obtain information on at least three critical factors:

- Total production in the system
- Total costs
- Cost per ton.

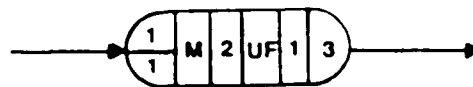
These data are not possible to obtain directly from Q-GERT's standard output, but Q-GERT does have a solution for that: the user written function, UF. As was told in Chapter 2, the user written function allows the user to insert their own code using ANSI FORTRAN. Q-GERT specifies where programming inserts may be used. Two possible locations are provided for making programming inserts. These are:

- At the start time of an activity; and
- At a node release time.

In the former case, the user function replaces the function type specification that normally is associated with the time to traverse the activity. In the latter case, the user function specification replaces the function type specification that normally specifies the

mechanism by which a value is assigned to a specific attribute of a transaction passing through the node. Hence, in the first case, the user function is called from a branch and, in the second case, it is called from a node. The parameter identifier in these situations is a user function number. The code letters UF are used to indicate that a user function is to be employed at a branch or node. Thus, the notation (UF,1) indicates that user function number 1 is to be employed.

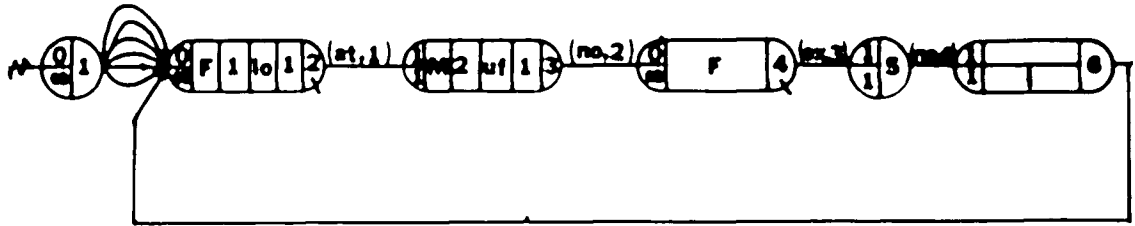
To obtain the desired statistics, however, we will use the second form of user function. The symbolism for this second type is shown below, where a user function was added to node 3:



At node 3, user function 1 is used to compute a value that will be assigned to attribute 2 of the transaction flowing through node 3. User function 1 will be a programming insert that will enable us to calculate the desired statistics.

In addition to UF, we will need two more subroutines specifically to handle I/O operations and initialization procedures. Those, provided by Q-GERT also (and maintained as dummy subroutines, were they not needed), are subroutine UI to initialize user's collected statistics and

variables, and subroutine UO to output those values calculated in UF. So, the new model becomes:



A few major changes distinguish this model from the previous one. At node 2, assignment of values to attributes of the transaction currently occupying the node is made at the portion appended at the bottom of node 2. Every transaction passing through that node will contain in attribute 1, a sample drawn from a lognormal distribution, the amount of material to load. This is used on activity 1, where the amount to be loaded on the truck being serviced is given by the value of attribute 1, or (AT,1).

On node 3 the major modification is the use of a user function which will accumulate the amount loaded. Later, subroutine UO will print out those values plus the cost calculations.

In order to accomplish these calculations, let us use one more datum, i.e. the capacity of a truck in the system. We were fortunate enough to have had access to data from off-highway trucks manufactured by Mack

Trucks (1). A typical 65-ton (59 metric tons) truck can be loaded with up to 75 tons (68 metric tons), and in practice this is customarily done, although it is not a good procedure.

So a truck will be loaded to its near maximum capacity, say 73 tons (66 metric tons).

The three subroutines incorporated in the Q-GERT program are shown below:

a) Subroutine UI, to initialize total amount loaded.

```
          SUBROUTINE UI
          COMMON/VCOM1/TOTLDS
C         INITIALIZE TOTAL AMOUNT LOADED
C
          TOTLDS=0.
          RETURN
          END
```

b) Function UF, to accumulate amount loaded. As it was described in chapter 2, Q-GERT provides the user with control capability of several internal Q-GERT functions and variables. In this particular case a call to subroutine GETAT is made to obtain the values of attributes 1 and 2 of the current transaction.

```
          FUNCTION UF(IFN)
          COMMON/VCOM1/TOTLDS
          DIMENSION ATT(2)
          UF=0.0
          CALL GETAT(ATT)
          TOTLDS=TOTLDS + ATT(1)
          RETURN
          END
```

c) Subroutine UO, to print out user's statistics.

```

SUBROUTINE UO
COMMON/TCO/1/TOTLOS
WRITE(6,100)
100 FORMAT (//1X,*SIMULATION RESULTS*,/1X,18(*-*))
TOTLOS=(TOTLOS*73.)/2.
TOTC=592. + 240.*6.
COSTON=TOTC/TOTLOS
COSTON=COSTON*100.
WRITE(6,200) TOTLOS,COSTON,TOTC
200 FORMAT (//1X,*TOTAL TONS LOADED/SHIFT=*,F10.1,3X,
1*COST/TON(CENTS)=*,F10.2,3X,*TOTAL COST(DOLLARS)=
*,F10.1)
RETURN
END

```

The Q-GERT program is shown below. A detailed description of each card and its parameters can be found in (2).

```

*** INPUT CARDS ***
GEN,VALENZUELA,FIRST,3,2),1978,1,(10)480.,3,E,(14)2*
SOU,1,C*
QUE,2/SHOVEL*
VAS,2,1,LD,1*
ACT,1,2*
ACT,1,2*
ACT,1,2*
ACT,1,2*
ACT,1,2*
ACT,1,2*
ACT,1,2*
REG,3,1,1,(5)4*
VAS,3,2,J=,1*
ACT,2,3,AT,1,1/SHOVEL*
QUE,4/CRUSHER*
ACT,3,4,ND,2,2/OTRIP*
REG,5,1,1*
ACT,4,5,EX,3,3/DUMP*
STA,5,1,1,(5)2*
ACT,5,6,ND,4,+ /DOWN*
ACT,6,2*
PAR,1,2,,0,5,,.5*
PAR,2,4,,0,10,,.9*
PAR,3,0.7,0,1.5*
PAR,4,1.5,0,4,0.2*
FIN*

```

Some results are quite lengthy; only a sample will be shown in next page.

GENET SIMULATION PROJECT FIRSTIM
DATE 3/28/1978

BY VALENZUELA

FINAL RESULTS FOR 3 SIMULATIONS

AVERAGE MODE STATISTICS

MODE	LABEL	PROBABILITY	AVE.	STD.DEV.	SD OF AVE	NO OF OBS.	MIN.	MAX.	STAT TYPE
6		1.0000	6.1933	.1255	.0725	3.	6.0900	6.3330	1

AVERAGE NUMBER IN Q-MODE

MODE	LABEL	AVE.	STD.DEV.	SD OF AVE	MIN.	MAX.	AVERAGE WAITING TIME	SD OF AVE	MAX.	NUMBER IN Q-MODE
2	CRUSHER	1.7534	:0773	1.0690	2.0559	3.9291	:2339	:1333	3.0000	3.0000

AVERAGE SERVER UTILIZATION

SERVER	LABEL	NO. PARALLEL SERVERS	AVE.	STD.DEV.	SD OF AVE	NO. OF OBS.	MIN.	MAX.	MAX. TIME ON SERVERS	MAX. BUST
1	BOYEL DUMP	1	:7029	:0401	:0001	3.	:9990	1.0000	7.0000	602.0000

Fig. 3.1 Results for Basic Model

Valuable information can be drawn from the computer output. As we can see, the average time for a round trip was 6.19 min. No truck spent more than 6.333 min., i.e. was the maximum time; and no less than 6.09 min., the minimum time.

Although not shown in the sample, 235 trucks on the average completed the haul cycle, carrying a total of 16959 tons (15417 metric tons) at a cost/ton of 11.98 cents and a total cost of 2032 dollars, assuming costs of \$0.5/truck/min and \$592/shovel/shift.

Let us now examine the waiting times in the queues. Two trucks were waiting at any time at the shovel on the average, showing that perhaps the system would do equally well with less trucks. Maximum number in the queue was 5. To the designer this could mean that either we are using too many trucks in the system or the shovel is too small. In the end, the designer will have to make a trade-off between production goals and costs (operation costs and investment in new equipment).

The average waiting time for the shovel was 3.94 min., which can be deemed not excessive depending on production requirements.

The crusher shows a very low utilization, pin-pointing again that the troubles are with the shovels (or too many trucks).

And again this is demonstrated by the average server utilization statistics which show the shovel being heavily utilized (almost 100% of the time), a policy that will lead to catastrophic results in real life.

If we run this model with 5, 4, and 3 trucks the following table is obtained:

<u>Trucks</u>	<u>Haul Time (min)</u>	<u>Tons</u>	<u>Cost (dollars)</u>	<u>Cost/ton (cents)</u>
6	6.33	17349	2032.	11.71
5	6.28	16979	1781.	10.55
4	6.18	15655	1552.	9.91 (*)
3	6.24	12280	1312.	10.68

This table shows that the cost/ton function has a convex drape, and a minimum is attained using 4 trucks in the system. The total amount produced is reduced by 1694 tons (1540 metric tons) when compared to the previous case when 6 trucks were being used. But, the average utilization of the shovel is also reduced to a more reasonable value, 89.7%.

BIBLIOGRAPHY

1. Mack Trucks, Inc., Your Mack Off-Highway Line, and other catalogs, 1977.
2. Pritsker, A.A.B., Modeling and Analysis Using Q-GERT Networks, John Wiley & Sons, 1977.

Chapter Four

A GENERAL MODEL

4.1 Introduction.

In chapter three a basic model of a single shovel system was discussed. For that simple case, the Q-GERT network was simple and straightforward, although yielding very useful information. This chapter will be dedicated to show how Q-GERT can be used to model a much more complex situation.

The type of mine layout we selected for our Q-GERT model represents a typical open pit mine that uses large equipment with a medium haul cycle. In the same fashion a set of data was assumed, based upon the works done by several other authors. The most valuable source of information was, however, a master's thesis by O'Neil (4), and a simulator designed by Venkataramani and Manula at the Pennsylvania State University in 1967 (3).

4.2 Statement of the Problem.

A schematic of the mine that was modeled is shown in figure 4.1. Four benches are currently being mined by a set of homogeneous shovels. One type of trucks is assumed.

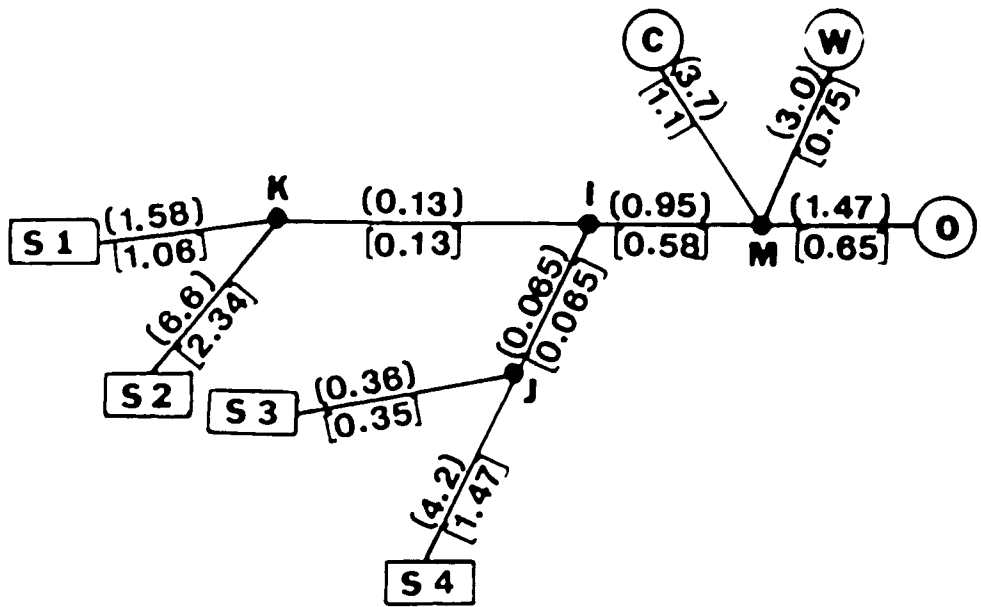


Fig. 4.1 Schematic of Mine.

Shovels 1 and 4 are mining ore to be conveyed to a crusher (C) and stockpile (O) respectively, and shovels 2 and 3 are mining overburden to be dumped on a waste dump (W).

Digging conditions are considered good. Maximum grades encountered are 12% and rolling resistance is constant at 2%. Rolling resistance includes friction in wheel bearings, tire flexing, and tire penetration into the haul road surface. The value is expressed as a percentage of the gross vehicle weight. Grade resistance is a gravitational force.

Data on trucks were furnished by Mack Trucks, Inc. From the extense variety of trucks available, we decided to use data pertaining to the 65-ton trucks (M-65BX, 59 metric tons). The fleet size is a variable whose best value we will attempt to find.

In the model, 100% of equipment availability is assumed. The length of the shift (time unit of the simulation), is 8 hours of uninterrupted operations.

There are four road intersections in the system, I, J, K, M; M being the most critical one since vehicles from up to four lanes may be competing to cross it at a given time.

A dispatcher is located at junction I. His main function is to allocate empty trucks to shovels. Among the several criteria that could have been selected, the allocation in this case will be based on shortest average queue size in front of the shovels.

The remaining data was estimated as follows:

a) Loading times.- A shovel loads a truck in a time normally distributed with mean 2.11 minutes and standard deviation of 0.66 minutes.

b) Dumping times.- Dump time distribution is exponential

with mean 0.74 min.

c) Haul and Return.- The time to complete a haul cycle will actually depend upon haulage profile, machine performance, and travel constraints. The time to travel a given distance is actually stochastic, although in this model constant values are assumed. In a real life situation this assumption can be easily modified. Q-GERT provides access to 11 commonly used probability distribution functions. In addition, through utilization of user function UF, it is possible to code any other p.d.f, and also read an empirical distribution in table form. Being a FORTRAN function which is responsibility of the user, UF can be used to attempt a combined simulation approach, i.e., to incorporate truck performance curves, theoretical estimation of the haulage profile, etc.

In figure 4.1, travel time values are indicated on each branch. The value above the branch is the travel time from the shovel to dumping destination, the returning time is given by the values below each branch.

Cost data were taken from (6), and for this case these are:

Trucks : \$240/shift/unit
Shovels : \$592/shift/unit

4.3 The Model.

Figure 4.2 shows the Q-GERT network for the situation described above. Each transaction in the model will represent a truck having the following attributes:

<u>ATTRIBUTE</u>	<u>DESCRIPTION</u>
1	<ul style="list-style-type: none">- At the beginning of the simulation it will contain a counter of the number of trucks generated for the system.- At any other time, it will contain the shovel number to which the truck is assigned by dispatcher.
2	<ul style="list-style-type: none">- Destination of transaction;<ul style="list-style-type: none">0 : returning truck1 : crusher2 : waste3 : ore stockpile
3	<ul style="list-style-type: none">- Loading time.

In addition, the following user written function codes will be used:

<u>CODE</u>	<u>DESCRIPTION</u>
1,2,3,4	- To accumulate total amount loaded by shovels 1,2,3,4 respectively.
5	- To simulate dispatcher's functions. The shovel number to which a given truck is assigned is stored in attribute 1.

The transactions flowing through the network are trucks which are generated on node 50, a source node. Node 50 is a "Take-All" type node and is used to generate a finite number of trucks for the model. In figure 4.2, 16 trucks will be generated.

Node 51 represents the initial assignment of trucks to shovels, this assignment made in user function with code 5. At the beginning of the simulation, trucks will start attempting to cross intersection I.

Shovels are represented by activities 1 to 4, and the nodes preceding them are the queues that can be formed in front of each shovel.

Once a truck has been loaded in a time stored in attribute 3, it initiates the journey to its assigned

destination (attribute 2). To illustrate the logic of the model we will trace the movements of one truck in particular. For instance, a truck that has just finished the loading operation at shovel 1 (represented by activity 1), must then travel to reach junction K in a time, assumed constant, of 1.58 minutes.

At junction K this truck must compete with others to occupy the intersection. The selection rule specified for selector node 14 is Preferred Order (POR); this means that the node will choose a transaction to be served by the junction according to a priority assigned by the user. In this case, preference is given to loaded trucks, and among them, to those conveying ore. So, the selection order at node 14 is 9,10,36.

The truck will then cross junction K in a time drawn from an exponential distribution with parameter 0.07 min., and continues toward the next intersection, I. At this intersection, it must compete with those trucks coming from shovels 3 and 4.

Node 21 is a "Take-First" node that will send the transaction to the next intersection M if attribute 2 is greater than zero (a loaded truck) or to intersections K or J if the truck is returning empty.

Node 24 will finally direct the truck to its assigned dumping destination where it joins the respective queue (if any) and perform the dump operation.

After finishing the dump operation, the transaction must return to a shovel to be loaded. To distinguish those returning trucks, at nodes 28,29, and 30, attribute 2 is set equal to zero. When a truck has completed the dumping operation, it is sent back to compete again for intersection M. This is the most critical intersection, since queues 32,33,22,31 are bidding for its service.

Once the truck has crossed intersection M on his way back, it is sent to node 34, which represents the dispatcher. On node 34 a user function with code 5 will assign the shovel to which the truck will be directed and store this value in attribute 1.

A truck assigned to shovel 1, for instance, will travel without further interruptions to its shovel. But a truck assigned to shovel 4 will probably have to stop at intersections I and J, since an empty truck has low priority to cross any intersections in the model.

Nodes 39,40,41, and 42 are statistical nodes to collect time between departures of transactions from these nodes

A shift of 480 minutes was simulated, and 6 runs were made for each possible fleet size.

4.4 Q-GERT Program Features.

In this section a brief discussion of the program is offered, emphasizing the description of the user function.

As it was mentioned earlier in this chapter, the user function (UF) is used to accomplish two tasks:

- Collect statistics on the total amount loaded,
- and
- Simulate the dispatcher's behavior.

At time zero, the initial allocation of trucks to shovels is made on node 51. This is accomplished by the following section of the program:

```

C*** ALLOCATION AT TIME TNOW=0.
C
5  IF(TNOW.GT.0.0) GO TO 10
   IX=ATT(1)
   JX=(IX-1)/4
   UF=FLOAT(IX-4*JX)
   GO TO 20
```

First a test is made at instruction 5 to see if TNOW is equal to zero. If TNOW is greater than zero, a branch is made to instruction 10. At the beginning of the simulation, however, TNOW is zero and will remain in that

state until node 50 had finished generating the number of trucks needed for the system. Hence for each transaction passing through node 51, a call is made to UF which will assign the shovel number to attribute 1. At first, however, attribute 1 will contain the generation number assigned in node 50 (through the Incremental function IN). This value is transformed to an integer IX, and then at statement 5+3 an assignment module 4 is made, so as to allocate the trucks as evenly as possible.

When TNOW is greater than zero, the allocation must be performed in a different way, this time by assigning trucks to the shovel which has had the smallest average queue size thus far. The section of the program that performs this allocation is:

```

10 TIMEI=TNOW-TLAST
   DO 11 I=1,4
   QNOW(I)=TINIO(I)
   AQSIZE(I)=(QNCW(I)-QLAST(I))/TIMEI
11  QLAST(I)=QNOW(I)
   TLAST=TNOW
C
C   FIND THE MINIMUM OF AQSIZE AND STORE IT IN TEMP
C
   TEMP=AQSIZE(1)
   K=1
   DO 15 I=2,4
   IF(TEMP.LE.AQSIZE(I)) GO TO 15
   TEMP=AQSIZE(I)
   K=I
15  CONTINUE
   UF=K
20  RETURN
   END

```

TINIQ(I), I=1,2,3,4, is a subprogram available for use in UF that computes the time integrated number in Q-node I from time TLAST to time TNOW. TLAST is the time of the last call to UF with code 5, that is, the time of the last data collection.

At statement 10, the time interval for data collection is calculated and stored in TIMEI. The DO loop is responsible for calculating the average queue size so far for each shovel. This is AOSIZE(I).

At statement 11, QLAST(I), which is the time-integrated number in the i^{th} queue up until the last data collection, is then updated to QNOW(I). At the next statement, TLAST is updated to TNOW.

A well known algorithm to find the location of the minimum value in a vector is then used with vector AOSIZE(I) and the resulting location I, i.e., the i^{th} shovel, is then returned as the value of the user function UF.

As usual, subroutines UI and UO are used to initialize all user's variables, and to print out collected statistics not included in the standard Q-GERT output. These are simple subroutines and will not be discussed.

*** INPUT CARDS ***

```
GEN, JALENZJEL1, OPEN-PIT, 4, 10, 1979, 4, C, 99999, 680, J, F, J, 4°
SOU, 50, 0, 1, 1, 1°
VAS, 50, 1, 1, 1, 1°
REG, 51, 1, 1, 1, 1°
VAS, 51, 1, 1, 1, 1°
ACT, 50, 50, 191A1, 1, LE, 25°
ACT, 50, 51, 191A53, R°
QUE, 1/05MOV1, 1013, 1, 1°
VAS, 1, 2, CO, 1, 1, 1, 1, 1°
QUE, 2/05MOV2, 1013, 1, 1°
VAS, 2, 2, CO, 2, 1, 1, 1, 1°
QUE, 3/05MOV3, 1013, 1, 1°
VAS, 3, 2, CO, 3, 1, 1, 1, 1°
QUE, 4/05MOV4, 1013, 1, 1°
VAS, 4, 2, CO, 2, 1, 1, 1, 1°
ACT, 51, 1, CO, 1, 14, 191A1, EQ, 1°
REG, 5, 1, 1, 1, 1°
VAS, 5, 4, UF, 1, 1°
REG, 6, 1, 1, 1, 1°
VAS, 6, 4, UF, 2, 1°
REG, 7, 1, 1, 1, 1°
VAS, 7, 4, UF, 3, 1°
REG, 8, 1, 1, 1, 1°
VAS, 8, 4, UF, 4, 1°
ACT, 1, 1, AT, 1, 1, /LOAD1, 1, 1°
ACT, 2, 6, AT, 1, 2, /LOAD2, 1, 1°
ACT, 3, 7, AT, 1, 3, /LOAD3, 1, 1°
ACT, 4, 8, AT, 1, 4, /LOAD4, 1, 1°
QUE, 9/JCTK, 1013, 1, 1, 1, 1°
QUE, 10/JCTK, 1013, 1, 1, 1, 1°
QUE, 16/JCTK, 1013, 1, 1, 1, 1°
QUE, 11/JCTJ, 1013, 1, 1, 1, 1°
QUE, 12/JCTJ, 1013, 1, 1, 1, 1°
QUE, 17/JCTJ, 1013, 1, 1, 1, 1°
ACT, 5, 9, CO, 1, 5, 1, 1°
ACT, 6, 10, CO, 6, 6, 1, 1°
ACT, 7, 11, CO, 7, 7, 1, 1°
ACT, 8, 12, CO, 8, 8, 1, 1°
SEL, 14/JUNCTK, 1713, 10, 36°
SEL, 15/JUNCTJ, 1711, 12, 37°
REG, 16, 1, 1, 1, 1°
ACT, 14, 16, EX, 6, 5/JUNCTK, 1, 1°
REG, 17, 1, 1, 1, 1°
ACT, 15, 17, EX, 6, 5/JUNCTJ, 1, 1°
QUE, 18/JCTK, 1013, 1, 1, 20°
STA, 4C/SMOV2, 1, 1, 1, 1, 1°
ACT, 16, 14, CO, 6, 13, 191A2, GT, 1°
ACT, 18, 47, CO, 7, 14, 191A2, EQ, 1°
ACT, 40, 2, 1°
QUE, 19/JCT1, 1013, 1, 1, 23°
QUE, 35/JCT1, 1013, 1, 1, 23°
ACT, 17, 19, CO, 7, 165, 191A2, GT, 0°
STA, 42/SMOV4, 1, 1, 1, 1, 1°
ACT, 17, 42, CO, 1, 47, 191A2, EQ, 3°
ACT, 42, 4, 1°
SEL, 20/JUNCTI, 17114, 14, 35°
ACT, 51, 35, 191A1, GT, 1°
REG, 21, 1, 1, 1, 1°
ACT, 20, 21, EX, 5, 7/JUNCTI, 1, 1°
QUE, 22/JCT1, 1013, 1, 1, 23°
QUE, 32/JCT1, 1013, 1, 1, 23°
QUE, 33/JCT1, 1013, 1, 1, 23°
QUE, 31/JCT1, 1013, 1, 1, 23°
REG, 3A, 1, 1, 1, 1°
ACT, 21, 34, 191A2, EQ, 0°
ACT, 21, 27, CO, 3, 35, 191A2, GT, 0°
ACT, 3A, 37, CO, 3, 165, 191A1, EQ, 4°
ACT, 38, 35, CO, 7, 13, 191A1, EQ, 2°
STA, 41/SMOV1, 1, 1, 1, 1, 1°
ACT, 38, 41, CO, 3, 13, 191A1, EQ, 3°
ACT, 41, 1, 1°
```

CONT...

```
SFL, 23/JUNCTM, 17122, 31, 32, 33°
REG, 24, 1, 1, 1, 1°
ACT, 23, 24, EX, 7, 8/JUNCTM, 1, 1°
QUE, 29/DOCM°
QUE, 29/DOCM°
QUE, 27/DOCM°
REG, 34, 1, 1, 1, 1°
VAS, 34, 1, 1, 1, 1°
ACT, 24, 34, CO, 3, 37, 191A2, EQ, 1°
ACT, 24, 34, CO, 3, 191A7, EC, 1°
ACT, 24, 34, CO, 8, 58, 191A2, EQ, 0°
ACT, 24, 27, CO, 1, 47, 191A2, EQ, 2°
REG, 28, 1, 1, 1, 1°
REG, 29, 1, 1, 1, 1°
REG, 30, 1, 1, 1, 1°
VAS, 28, 2, CO, 0, 0°
VAS, 29, 2, CO, 0, 0°
VAS, 30, 2, CO, 0, 0°
ACT, 25, 28, EX, 1, 9/DOCM°
ACT, 26, 29, EX, 1, 10/DOCM°
ACT, 27, 30, EX, 3, 11/DOCM°
ACT, 29, 12, CO, 1, 1°
ACT, 29, 33, CO, 0, 75°
ACT, 30, 31, CO, 0, 65°
STA, 39/SMOV1, 1, 1, 1, 1, 1°
ACT, 34, 39, 191A1, GT, 1°
ACT, 34, 39, CO, 1, 19, 191A1, EQ, 1°
ACT, 34, 36, CO, 0, 17, 191A1, EQ, 2°
ACT, 39, 1, 1°
PAR, 2, 2, 1, 0, 18, 0, 66°
PAR, 3, 0, 7, 4, 0, 18°
PAR, 4, 0, 19, 0, 0, 3, 6°
PAR, 5, 0, 0, 0, 0, 0, 7°
PAR, 6, 0, 0, 0, 0, 0, 7, 3°
PAR, 7, 0, 36, 0, 0, 0, 15°
FIN°
```

Fig. 4.3 Q-GERT Program Listing

Once the Q-GERT network has been drawn, and all logic errors been debugged, it is straightforward to set up the Q-GERT program itself.

4.5 Discussion of Results.

Detailed statistical information can be obtained from the standard Q-GERT output. Figure 4.4 shows a sample of the output for the model, which presents the solution deemed the best (fleet size, 26 trucks) for a time to cross intersection M of 3.6 seconds.

It can be seen that with 26 trucks in the system, shovels are been heavily used:

Shovel 1: 95.48%

Shovel 2: 87.67%

Shovel 3: 84.41%

Shovel 4: 75.20%

Some reasons account for the fact of the uneven utilization of shovels. One reason that heavily influenced the uneven workload allocation is the position of each shovel in the mine layout. Shovel 4 is more difficult to access; the trucks assigned to it must cross junctions M, I, and J, and probably stop at each of them. On the other hand, a truck returning to shovel 1 must only stop

```

SIMULATION RESULTS
-----
NUMBER OF TRUCKS IN THE POOL = 74
SMOKE, NO. TOTAL LOADS
-----
1 13807.8
2 16687.7
3 16676.1
4 17663.5

*****0145 5931: 99.918 12914851
*****061786: 99.918 12914851

GENE SIMULATION PROJECT OF EMPLOY BY VALLENMUELLER
DATE 07/18/1978

ORIGINAL RESULTS FOR 3 SIMULATIONS**

**AVERAGE POOL STATISTICS**
NODE LABEL PROBABILITY AVE. STD. DEV. SD OF AVE NO. OF MIN. MAX. STAT
**AVERAGE NUMBER IN Q-MODE**
**AVERAGE WAITING TIME**
**AVERAGE NUMBER IN Q-MODE**
**AVERAGE SERVICE UTILIZATION**

```

(expected time to cross "M" = 3.6 secs.)

Fig. 4.4 Output Sample.

at intersection M (lanes are supposed ample enough to allow limited passing maneuvers through the right).

These results obviously reflect the kind of priorities we set up in the model. In a real life situation this allocation should be inconvenient, and a tradeoff solution must be attempted.

Using the algorithm described before for trucks allocation, the dispatcher assigned trucks unevenly, as the following figures show it:

<u>To Shovel</u>	<u>Trucks Allocated During Simulation</u>
1	215
2	201
3	196
4	162

This indicates that the algorithm should be changed to reflect other type of priorities if necessary.

The average number in the queues, as well as the average waiting times are:

<u>Shovel</u>	<u>Ave. Number in Queue</u>	<u>Ave. Waiting Time (min)</u>
1	1.32	2.93
2	1.55	3.63
3	0.85	2.11
4	1.03	2.91

Statistics nodes 39 to 42 are giving the mean time between arrivals to each shovel, and these results also show that the dispatching policy is not the most appropriate for the model:

<u>Shovel</u>	<u>Interarrival time (min) to Shovel</u>
1	2.21
2	2.33
3	2.48
4	2.79

Of course, since the allocation was uneven, the average amount loaded by each shovel was also different.

It may be also interesting to see some statistics on the occupancy of intersections. For instance, when the time needed to cross "M" is .36 min. on the average, the results are:

<u>Intersection</u>	<u>Ave. Utilization</u>	<u>Ave. Waiting Time (min)</u>
K	8.73%	0.01493
J	19.21%	0.0532
I	45.13%	0.2130
M	44.76%	0.0549

Also,

<u>Intersection</u>	<u>Ave. Number in Queue</u>	<u>Max. Number Queue</u>
K	0.0018	5
J	0.0187	7
I	0.1644	22
M	0.0372	10

Contrary to our initial guess, it is junction I the most critical one. If a new design of the mine layout is to be attempted, intersection I should be carefully considered.

After running the program with different fleet sizes, we obtained the figures summarized in table 4.1, from which the cost curve of figure 4.5 is drawn. The cost curve shows a convex shape, with a minimum attained at 15.036 cents and a fleet size of 26 trucks.

Table 4.1

Performance of System with Dispatcher

(Expected time to cross M : 3.6 secs)

<u>No. of Trucks</u>	<u>Tons Hauled</u>	<u>Cost/Ton (Cents)</u>
16	38303	16.20
23	51631	15.74
24	53728	15.32
25	55626	15.04
26	57249	15.036 ←-- Opt.
27	58441	15.14
28	59739	15.21
29	61678	15.32
30	62079	15.41

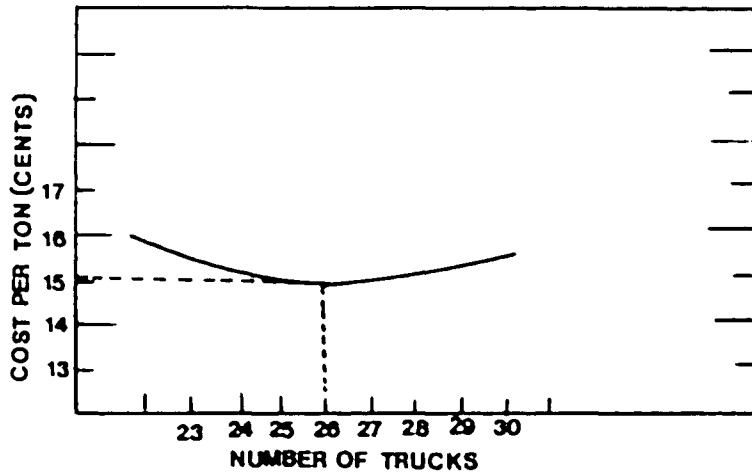


Fig. 4.5 Cost Curve.

Results indicate that the best size of the fleet in the model is 26 trucks. However, we must emphasize that these results are by no means the optimum solutions for the system. Problems resulting from a rather poor allocation lead us to believe that there must be a better solution, closer to the real optimum.

On the other hand, and contrary to our initial guess, the time to cross intersection M is not as critical as the time to cross intersection I. For instance, if we increase the time needed to cross M from 3.6 secs. to 21.6 secs. the average utilization of the junction augments from 17.7% to 44.8%, indicating a nonlinearity in the relationship between time to cross and utilization (the time was increased 6 times, resulting only in an increase of 2.5 times the utilization).

Table 4.2 shows the influence of these two extremes of the time to cross M in cost and production figures.

Table 4.2

Comparison of Shovel Performance for Two Values
of Crossing Times of Intersection M and Fleet Size
of 26 Trucks.

SHOVEL 1	Lambda=0.06 min.	Lambda=0.36 min.
Ave. No. in Queue	1.32	1.27
Ave. Waiting Time	2.93	2.81
Max.No. in Queue	8	6
Utilization	95.48%	94.74%
Production (Tons)	15862.	15623.
<hr/>		
<u>SHOVEL 2</u>		
Ave. No. in Queue	1.54	1.53
Ave. Waiting Time	3.63	3.65
Max No. in Queue	7	8
Utilization	87.67%	87.70%
Production	14469.	14531.
<hr/>		

Table 4.2 (Cont.)

SHOVEL 3	$\text{Lambda}=0.06 \text{ min.}$	$\text{Lambda}=0.036 \text{ min.}$
Ave. No. in Queue	0.85	0.85
Ave. Waiting Time	2.10	2.14
Max. No. in Queue	5	6
Utilization	84.41%	83.86%
Production (Tons)	14474.	14014.
<hr/>		
<u>SHOVEL 4</u>		
Ave. No. in Queue	1.03	0.88
Ave. Waiting Time	2.90	2.57
Max. No. in Queue	7	7
Utilization	75.20%	73.20%
Production (Tons)	12443.	12047.
<hr/>		

Examining the results of table 4.2 seem to indicate that there is no much difference in the values obtained for both crossing times. However, if we look at the problem from the production standpoint, we will see that when the time is bigger, production is reduced by 1034 tons/day or, more impressively, by 248160 tons/year. This diminished production may more than justify a detailed

analysis of the intersection.

On the other hand, if we reduce the time to cross intersection I to 0.10, that is half of what it was, production increases by 1036 tons/day or 248640 tons/year. Obviously this is a sizable increment which could presumably pay for the effort of an in depth study of "I". Such an study would involve reconsidering the kind of priorities we set up. An in-depth study of intersections, however, is out of the scope of this work.

4.6 Potential Experiments with the Model.

One of the most notorious advantages of Q-GERT is that it allows the user to conduct experiments with a given model without changing the entire logic of the system.

An analyst interested in evaluating alternatives to the current layout may perform the following experiments without changing the network itself.

- Test different values of times needed to cross a given intersection, and evaluate the influence of these changes in production and costs.
- Change loading times and distributions. Test influence of various truck and shovel sizes.

- Consider different allocation priorities.
- Test different priorities in occupying an intersection.
- Change distribution functions describing times to travel along the network.
- etc.

Other experiments requiring major and minor changes in the network are really unlimited. However, it may be worthy to mention a few:

- Test the model with fixed assignment of trucks to shovels. For this purposes, UF may not be required, and node 34 would become a simple "Take-First" node.
- Study the influence of different intersection patterns.
- Study the effects of changing certain paths in the network.

4.7 Summary.

We have seen how Q-GERT was applied to a typical mining problem. Also, the immense potential in terms of analysis capabilities offered to the user was envisioned. The mine chosen, although reduced in size, represents the typical complexity of a real one.

Other approaches have been attempted in the past to analyze similar situations with various degrees of success. A well known work by O'Neil (improved later by Manula and

Venkataramani), which we described previously, presented a much more simple mine layout to demonstrate the usefulness of his simulation program.

It is not difficult to show that O'Neil's layout can be easily analyzed using Q-GERT. The general prototype is presented in figure 4.6.

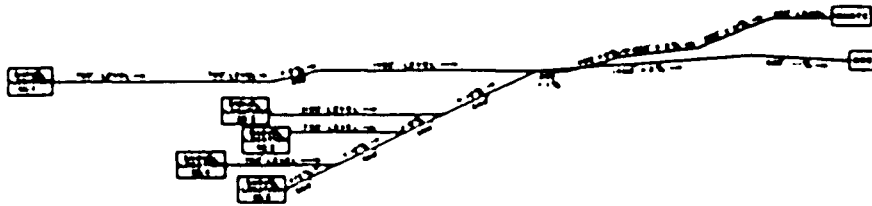


Fig. 4.6 Prototype of Mine Layout presented by O'Neil.

Only overburden is mined at shovel location 1, thus establishing a separate and independent subsystem within the main mining plan. There are no shovels at locations 3 and 4. Shovel at location 2 will mine only overburden, and shovel at location 5 will have a variable ore/waste ratio.

Thus, the problem can be reduced to the following networks in figure 4.7:

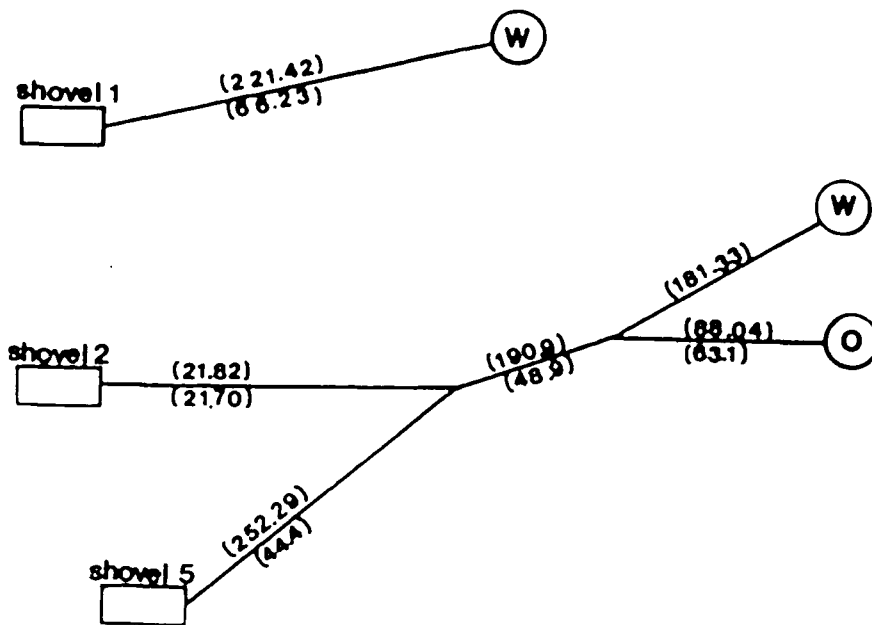


Fig. 4.7. Reduction of O'Neil's Diagram to a Network.

It is easy to see that this system is really a subset of the general model we have discussed throughout the chapter.

Many other models encountered in the literature are also amenable of reduction to the general model presented here. Some particular problems and enhancements will be discussed in next chapter.

The time spent in organizing, programming, debugging, compiling, and executing the Q-GERT network was reasonably short. Q-GERT is especially well adapted to get useful results quickly. As always happens, the most cumbersome part was the problem definition and data collection. Once this problem was overcome, organizing and programming the first version of the model took 4 to 5 hours. Compilation and execution of the final version(s) took 38 seconds on Lehigh's CDC 6400, requiring 107200 words (60 bit-words). The execution cost (not including I/O) was \$4.58 in non-prime time according to Lehigh University standards.

BIBLIOGRAPHY

1. Atkinson, T., Selection of Open Pit Excavating and Loading Equipment, Institution of Mining and Metallurgy, May 1971.
2. Gibbs, L.W., et al., System Analysis for Truck and Shovel Selection, SME Transactions, Dec. 1977, pages 354-359.
3. Manula, C.B., and Venkataramani, R., Open Pit Truck Haulage Simulator (SIMULATOR II), College of Earth & Mineral Sciences, The Pennsylvania State University, 1967.
4. O'Neil, T.J., Computer Simulation of Materials Handling in Open Pit Mining, M.S. thesis, The Pennsylvania State University, 1966.
5. Pritsker, A.A.B., Modeling and Analysis Using Q-GERT Networks, John Wiley & Sons, 1977.
6. SME Mining Engineering Handbook, Society of Mining engineers of AIME, 1973.

Chapter Five

ENHANCEMENTS TO THE GENERAL MODEL

5.1 Introduction.

In Chapter 4 we described what it meant to be a general model. However, several features can be added to that model so as to augment the user's perspectives insofar as Q-GERT applications are concerned.

Several potential enhancements will be discussed as well as the way to implement them. Specifically, we have found that the following may be points of special interest:

- Haulage times
- Intersection analysis
- Truck failures and preventive maintenance
- Shovel breakdowns and/or preventive maintenance.

The actual details of each analysis are not included, because a particular implementation depends on the experience and preferences of the modeler. As yet model construction is an art rather than a science.

5.2 Haulage Times.

Different approaches to calculate the travel times between any two points in the network can be attempted. If time studies are available and there is a need to

evaluate existing conditions, a probability distribution function will likely describe travel times. Up to 11 built-in probability distribution functions can be used in Q-GERT, and through the using of UF, the limits are really open-ended.

An alternative to the above, is to make use of truck performance data provided by equipment manufacturers. The easiest way to incorporate that data into the model is to fit the performance curves to a standard equation and insert this equation into UF. Most of the times the least squares method gives a good approximation, since most truck performance curves suggest an exponential curve. Figure 5.1 shows a typical performance curve.

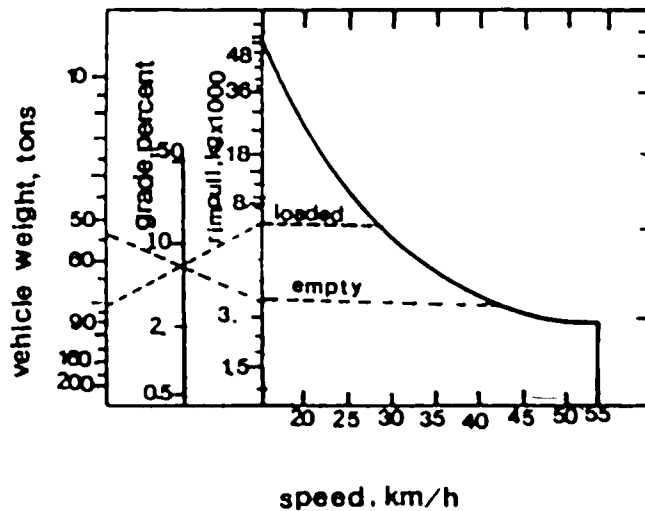


Figure 5.1 Typical Performance Curve for a Loaded Truck

This form of performance curve is highly recommended since it allows for evaluation of the hauling unit under varying load conditions. It shows the relationship between vehicle speed and tractive effort or rimpull capacity.

If the rimpull of a truck is in excess of the requirements necessary to overcome the combined rolling resistance and grade resistance, then the truck will accelerate. This acceleration can be calculated theoretically by the equation of Newton's Second Law of Motion, $F=M \cdot A$.

However, it is evident that constant acceleration equation cannot be applied since rimpull varies greatly with vehicle speed.

To overcome this difficulty, a number of piecewise approximations using small incremental changes in velocity throughout a small increment in time can be used. This suggests an iterative procedure which can be used to obtain useful information.

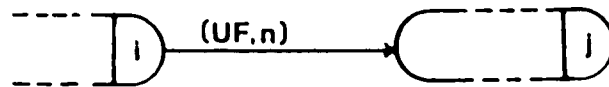
An equation relation vehicle speed in km/h and total grade can be approximated by using the least squares method. Any standard statistical package should provide such equation. In the same way, the time to traverse a distance between two points in the mine layout could be obtained, by making some simplifying assumptions such as constant average speed on a road segment, constant rolling

resistance, etc.

Let us assume that we have been able to fit an equation to relate time with distance (i,j) , where i,j are any two road segments in the mine layout. That equation will be represented by the function $f(t)=G(D(i,j),dig)$; dig being parameters like rimpull, speed, rolling resistances, grades, etc.

Such equation can be readily incorporated in Q-GERT through UF, the user written function.

Thus the activity "traveling from i to j ," is represented by:



where n represents the identifier of user function number u . The coding of UF would look like:

```
FUNCTION UF(IFN)
COMMON /UCOM/ user's defined variables
  :
  UF=equation to calculate time
  :
RETURN
END
```

Figure 5.2 Haultime calculation in UF.

This method, however, can hardly provide much accuracy since it probably will not take into account certain factors like changes in rolling resistance and/or grade resistance, handling corners, etc.

A better alternative to the above, is to apply an iterative procedure assuming a constant acceleration, during a short period of time, which in turn determines vehicular rimpull from truck performance curves. This rimpull fixes a new acceleration rate for the next incremental time period and the procedure is repeated.

The vehicle would accelerate at a rapid rate from standstill. The acceleration rate would be reduced as velocity increases until the tractive effort equalled the combined rolling and grade resistance, at which time a steady state velocity would be achieved. This steady state velocity would be maintained until it was necessary to brake the truck to a stop.

The potential speed a truck can attain is to be calculated from performance curves or from the equations representing these curves. Distance traveled should be continuously recorded.

O'Neil (2), describes a simulation model that uses a hybrid probabilistic-deterministic approach. Three sets of data are required to operate his model, i.e., mechanical capabilities, haulage profile and travel constraints.

An interesting feature of this model is the subroutine HAULTIME, set up to calculate the time to travel a given number of sections, using the iterative procedure already described. The same approach has been attempted

by a number of other authors

A similar routine can be implemented for use in the Q-GERT model to calculate the haul time between any two points. However, this approach may have a few drawbacks:

- It may slow down the computer execution time to a great extent, thus offsetting the possible advantages of the method.
- It requires a sizable amount of data, sometimes not readily available.
- The degree of accuracy achieved cannot be much higher (if it is higher at all), than the accuracy obtained by conducting time studies and fitting that data to a probability distribution function.

This method could be especially applicable when no data from actual operations can be obtained, i.e. when designing a totally new mine layout.

5.3 Intersection Analysis.

Congestion at intersection zones may become a highly critical cost factor, especially in mining operations using large fleets. A detailed analysis of intersections is then justified. Q-GERT, like GPSS, is a simulation language essentially designed to deal advantageously with queueing situations. As it was mentioned in Chapter 4, each intersection in the general model was divided into a number of cells determined by pairs of intersecting lanes in the

junction. A cell is the minimum area that may be occupied by competing trucks when attempting to cross the junction.

If the congestion at the intersections is deemed to be highly critical, a separate study may be worthwhile. In the past, GPSS has been extensively used by traffic engineers to conduct detailed analyses of traffic control situations (1). Although some of the ideas may not be directly applicable in open pit mining problems, some can be adapted.

However, incorporating detailed subnetworks to deal with intersection problems in a network model representing big mining operations may not be advisable, since it would introduce unnecessary complications, and the model may grow to unmanageable sizes. The solution is, then, to conduct intersection analyses separately and incorporate the results in the mining operations model.

5.4 Truck Failure and Preventive Maintenance.

In actual operations, a 100% of truck availability is almost impossible to attain. There will always be failures occurring randomly to equipment that reduce the availability figures significantly. Preventive maintenance is another factor influencing availability.

The first problem, i.e. random failures, is not easy to accommodate in the Q-GERT network. Several problems must be considered first to provide a workable framework.

The model designer should have information about the truck failure pattern, obtainable through statistical data on mean line between failures. Assuming this data fit a known probability distribution function (p.d.f.) or can be inputted as a table to the model, then the modeler will face the problem of how to handle this failure factor in the model. A failure can be considered an exogenous event that will occur at any place in a random fashion. The problem, in Q-GERT, is how to deal with the problem of truck breakdowns that can happen at any place in the network. It is perfectly feasible for a truck to fail in its way up to dump ore, for example, or when it's being loaded, or when attempting to cross an intersection, etc. In most cases it is not simply a matter of "killing" a transaction from the system, but also a problem of disturbing the behavior of the remaining components. If a loaded truck fails, for instance, just when attempting to cross an intersection, that will lead to causing other problems also:

- Some of the lanes will get temporarily blocked, as well as the intersection capacity be reduced and sometimes get blocked also.
- The load carried by the broken truck must be transferred to another truck, which means that an empty truck is assigned to the task by the dispatcher.

5.5 Shovel Breakdowns and Maintenance.

In section 5.4 we explored some potential enhancements where mainly transactions were involved, i.e. items that flow through a network. The situation involving shovels is different, at least from the Q-GERT modeling point of view. This time certain facilities may not be available to process transactions requesting service.

As in the case of trucks, the time between successive failures of a shovel is generally a random variable. Let us consider some of the problems that should be analyzed to obtain a general model describing shovel breakdowns (maintenance can be considered as a particular case).

Several actions must be taken when a shovel suffers a breakdown:

- If a truck is being loaded, interrupt its service. Send the truck semi-loaded to its destination or some place else (perhaps to another shovel).
- Instructs the dispatcher to stop allocating trucks to the broken shovel.
- Any trucks waiting for this shovel must be reassigned. In the network these trucks should not pass through the server "shovel", unless this was done with service time equal to zero. In that case, statistics on server utilization should not take this zero services in considera-

tion.

- Schedule an event "end-of-repair" for this shovel.

This simply means that at the end of activity repair, the shovel should be set available again.

In addition statistics must be collected on all the variables of interest.

Fortunately, Q-GERT does provide some useful sub-routines which may help in handling these situations. The three of them that are the most likely to be used in UF are:

- Subroutine STAGO(NSERV,NODE,TIME,ICATT,ATT), which causes server NSERV to stop processing a transaction and sends transaction stopped to node NODE with a time delay of TIME. If ICATT is set to 1, the attributes of the transaction are changed to the vector ATT. The status of server NSERV is then updated according to the condition of the Q-nodes preceding it. If NODE=0, the transaction is lost to the system. If server NSERV is idle, no action is taken.
- Function REMST(NSERV), which returns REMST as the service time remaining on service activity NSERV. If NSERV is idle, a value of zero is returned. If there are parallel service activities, REMST is set to the smallest

remaining service time.

- Function RCAPQ(NODE), which returns RCAPQ as the remaining capacity of Q-node NODE.

A decision must be made about generation of "repair-transactions". It seems much easier to set up a separate generator for each shovel, thus avoiding the problems of considering joint probability distribution functions in the case of several shovels of different types.

5.6 Conclusions.

Q-GERT, in its present version released in 1977, is a highly structured simulation language and network technique. The simulation language provides a quick, low-cost means to gain insight into a general haulage system which has been the object of a number of works in the past. Several simulation languages have been used, ranging from standard FORTRAN IV ANSI to GPSS V, with various degrees of success to increase the knowledge about such systems and those factors bearing the greatest influence on costs.

We have seen how easily a general haulage model was implemented in Q-GERT, and how useful was the information we obtained from it.

A fundamental contribution of Q-GERT to the world of simulation languages, is its method for graphically modeling systems which permits better insight into complex queueing situations (Q-GERT could be also used to model some non-queueing situations). Conversion of a Q-GERT network to a Q-GERT program is straightforward.

The great variety of features available in Q-GERT, some of which we have used and discussed in this work, make the language a very attractive choice in the analysis of complex haulage systems. Reinforcing this attractiveness are some other factors like:

- A manual, which can also be used as a textbook, provides excellent documentation for the use of this language.
- The language is highly transportable, since it is written in standard FORTRAN IV ANSI.
- For the same reason, the user has potentially at least, unlimited access to all internal functions of the language.

For the sake of fairness, we must mention some problems encountered when using Q-GERT. For instance, it was not possible to obtain histograms requested at Q-nodes. We were told, however, that this problem is being fixed.

The other disadvantage is related to some features we used when modeling with GPSS V. We missed, for example, capabilities to implement directly some blocks like GATE, PREEMPT, as well as MSAVEVALUE and certain boolean logic features. Of course, this functions can be simulated through the user written function and Selector node.

BIBLIOGRAPHY

1. Blum, A.M., A Vehicle Traffic Simulator, IBM Systems Journal, Vol. 3, pages 41-50, 1964.
2. O'Neil, T.J., Computer Simulation of Materials Handling in Open Pit Mining, M.S. thesis, the Pennsylvania State University, 1966.
3. Pritsker, A.A.B., Modeling and Analysis Using Q-GERT Networks, John Wiley & Sons, Inc, 1977.

VITA

Carlos Alberto Valenzuela, son of Carlos and Graciela Valenzuela, was born in Valdivia, Chile, on May 6, 1947.

He is married to the former Luz Elena Silva of Santa Cruz, Chile.

Mr. Valenzuela was graduated from Internado Barros Arana and received the degree of Electrical Engineer from the Catholic University, Santiago, Chile, in 1972. He was employed for two years with The Chuquimata Copper Company in Chile, then went to Venezuela where he worked for two years with the Public Works Ministry as a Systems Analyst, and taught courses in statistics at the Catholic University Andres Bello, Caracas. He is currently a graduate research assistant with the Center for Information Science of Lehigh University.