OPTIMUM MULTI PERIOD
OPEN PIT MINE PRODUCTION
SCHEDULING

by

Kadri Dagdelen

A thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Doctor of Philosophy (Mining Engineering).

Golden, Colorado
Date August 31, 1985

Signed: Kadri Dagdelen
Kadri Dagdelen

Approved:
Thys B. Johnson
Thesis Advisor

Golden, Colorado
Date September 2, 1985

Matthew J. Hrebar
Matthew J. Hrebar
Acting Department Head
Mining Engineering

ABSTRACT

The scheduling of production from open pit mines such that returns from the mining operation are maximized has been an unsolved problem. The current production scheduling methods are geared toward obtaining a feasible solution which has at best the largest net present values of 8 to 10 different scenarios considered. With these "optimum" solutions there is no guaranty that the mine plan achieved is the true optimum. The various attempts to apply mathematical optimization to the production scheduling problem have proved unproductive because of the large computer memory requirements and solution time for a problem with 300,000 variables and about 900,000 constraints.

The open pit scheduling problem considered in this dissertation is formulated as a large scale linear program and solved by decomposing the problem using lagrangian relaxation. Subgradient methods are used in modifying the lagrange multipliers corresponding to the side constraints consisting of blending and capacity requirements of the mining system. The lagrangian subproblem which is the multi time period sequencing problem is solved by another algorithm developed in this dissertation. This algorithm which is based on the network structure of the sequencing constraints decomposes the multi time period sequencing problem into series of efficiently solvable single time period problems.

The relationship between the lagrange multipliers and the traditional cut off grade concepts is discussed. A new interpretation of cut off grades is given.

The proposed method's applicability to real problems is demonstrated by developing necessary computer programs in FORTRAN on the VAX 11-750 computer. The Lerchs and Grossmann's 3-Dimensional Graph theoritic three algorithm is programmed to solve the subproblems of the multi time period sequencing problem.

The proposed method will overcome the limitation of existing techniques by improving the discounted cash flows and by decreasing the engineering time spent on scheduling open pit mines.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

There are a few special people who contributed directly and indirectly to the successful completion of this research. I am thankful and deeply indebted to all of them.

In particular I am very grateful to Dr. Thys B. Johnson, my thesis advisor, for his efforts in providing financial, moral and intellectual support at critically needed times. Without his support and encouragement this work would not have become a reality.

I am very thankful to Randal Barnes for the times we have spent in the I-Club discussing production scheduling problem. Although we have spent more time discussing things other than production scheduling, I must add, they were probably the essential support to the whole adventure.

I am very thankful to Dr. William Hustrulid for making sure that a Ph.D. degree is not for everyone, but for those few who really deserve it.

I owe thanks to Prof. Matthew Hrebar, Dr. Charles Berry of the Mineral Economics Department, and Dr. Harold Taylor of the Mathematics Department for serving on my Ph.D. committee and for reading the manuscript.

I am thankful to Dr. Dominique Francois-Bongarcon for teaching me everything I know about "French Pit Parametrization".

I am also very thankful to my fellow graduate students of the past five years, especially Mike Nilsen, Francisco Sameniego, Jeff Myers, Leo Van Sambeek, Roger Dewey and Felipe Calizaya. I will treasure the good times we had together forever.

Lastly, I am thankful to Sharon Dunning for her patience and marvelous work in typing the manuscript.

I. INTRODUCTION

In recent years, the quantity of metals produced from open pit mining operations increased at the expense of underground mining. Today, open pit mining provides 60% of the metal needs of the modern world's consumption. The shift to greater production from open pit mining came mainly as a result of productivity improvements through economics of scale. In general, large scale, low grade operations characterize today's open pit mines. These large scale open pit mines are very capital intensive, with initial capital requirements in the range of five hundred million to a billion dollars. Mining the low grade ores does not leave much room for inefficiencies. There is a greater need for these operations to be planned and operated very efficiently to achieve the best returns on investments. The most efficient mining plan is defined as the mine plan which under existing economic conditions and technological capabilities would provide maximum profits for the mining company. The methodology for defining the optimum mine plan for an open pit mining system is the topic of this dissertation.

1.1 NEED FOR OPTIMUM OPEN PIT MINE DESIGN

It is generally accepted that the objective of a modern firm should be to maximize the wealth of the stockholder, the legal owners of the firm . In order to reach this objective, every project should be designed and operated so as to maximize the firm's investment decision criteria. There are a number of investment decision criteria used in evaluating the profitability of a given deposit. The most widely used technique of project evaluation is Net Present Value (NPV) analysis. The NPV of a project is generally maximized when optimizing a mine design.

The decision variables that influence the NPV of an open pit mining operation can be given as (Blackwell 1971):

1. physical capacities of various stages in the production process,
2. cutoff grades,
3. sequence of extraction.

The influence of the first variable, the physical capacities of various stages in the production process on NPV, is twofold. First, the capacities of various stages determine the magnitude of capital investments and costs. Second, these capacities determine the rate at which valuable minerals will be recovered. As a result this first variable influences the NPV by affecting both the cash inflows and the cash outflows.

The mining industry is one of the few industries where capacities of various stages in the production process do not directly determine the rate of final output alone. The rate of recovery of metals from a deposit is influenced by cutoff grades, and the sequence of extraction as well as limitations imposed by different capacities. Therefore, it is through a combination of factors that these variables affect both the positive and negative cashflows and as a result the NPV of a mining operation.

As stated by Johnson (1968), given the capacities of various stages of the production process, determining cutoff grades and the sequence of extraction such that the maximum possible NPV is realized, has been an unsolved problem. Yet it is stated by Blackwell (1971), Lilico (1973), and Wells (1978) that policy decisions made with respect to each one of these variables are so important that an arbitrary choice may have substantial negative effects on the NPV of a deposit and can cause a mine to operate with lower profits than the maximum attainable, or in some cases a wrong choice can cause a deposit to be uneconomic and submarginal even though the deposit could be mined at a profit.

As the mining industry mines lower and lower grade ores, planning of a mine with an optimum combination of these decisions parameters is becoming more important. Realizing this point, Johnson (1968) said:

> The mining of a mineral deposit in such a manner that at depletion the maximum possible profit is realized, has been an unsolved problem since man's discovery of usable elements buried beneath the earth's surface. In the days when high grade reserves were adequate to supply our needs, the attention given to this problem was negligible. The philosophy at the time was to extract the material in an orderly fashion, keeping in the high grade until depletion. Right or wrong, profits were high, so no question of optimum profitability confronted the operators. Since World War II and the depletion of the most accessible of the world's high grade reserves, the mining industry has been forced into working with lower grade material. The sequence of extraction has now become more important; and in many cases, has become a problem whose solution is vital to the existence of a profitable operation.

Although the need to obtain a mine plan with optimum combinations of values for these parameters has been obvious, the methods to obtain it has not been (Johnson 1968; Blackwell 1971; Gangwar 1974). Since the relationship between these variables and NPV is complex and each one of these variables are interdependent, the mine planning methods used in the mining industry today are geared toward a feasible solution and at best usually results in the largest NPV of the 8-10 different scenarios considered.

With these "optimum" solutions there is no guaranty that the mine plan achieved is optimum and there is not even a clue of how far one is from the true optimum.

1.2 Scope of Work

The overall objective in mine planning is to come up with a mine plan which results in a maximum NPV of depletion of the deposit. Although this objective is the ultimate goal in mine planning, its achievement requires analysis of a broad spectrum of interrelated parameters: optimization with respect to physical capacities; optimization with respect to capital and operational expenditures; optimization with respect to extraction schedule of the deposit.

Although optimization of a mine plan in terms of these parameters are interrelated, the scope of this research primarily encompasses optimization with respect to extraction scheduling of a deposit. This means that the critical parameters with respect to physical system capacities, their capital and operational expenditures are assumed to be known and the optimum production schedule which will give maximum discounted cash flows at depletion of the deposit is searched for.

Hence, the objective of this thesis is twofold. The first one is to formulate a mathematical model which will take into account complex interactions of interdependent planning variables in order to come up with an optimum production schedule which will give maximum discounted cashflows at the depletion of the reserve. The second objective is to develop a solution algorithm that will solve this model in a reasonable amount of time.

In Chapter 2, traditional mine planning methods will be reviewed and the structure of the complex interactions between design parameters and NPV will be explored. Also a critical review of heuristic optimization techniques will be given along with a discussion of previous attempts for solving production scheduling problems by mathematical optimization techniques. In Chapter 3 a mathematical model which describes the interaction of design elements such that their combined effect is considered in coming up with an optimum mine plan is formulated. The structure of the problem formulated in chapter 3 plays a very important role in the development of a solution algorithm. Hence, in Chapter 4 the structure of the production scheduling model formulated in Chapter 3 will be described. The solution algorithm developed in this dissertation is based on the concept of lagrange multipliers. Chapter 5 discusses the concept of lagrange multipliers as they relate to mining. In Chapter 6 a solution algorithm to the mine production scheduling problems in terms of subgradient optimization will be given. In Chapter 7, the algorithm for multi-period sequencing problems is presented. Chapter 8 will discuss the development of a computer program to implement the solution algorithm. This chapter also discusses the possible modification necessary for implementation of the model to a full size mine. Chapter 9 gives conclusions of the research study and recommendations for future research.

CHAPTER 2

CRITICAL REVIEW OF MINE PLANNING METHODS

2.1 MINE PLANNING

The planning of an extraction schedule over a particular time horizon, typically the life of a deposit, is referred to as mine planning (Johnson, 1968). Figure 1 shows the generally accepted circular analysis one follows during mine planning. With present practice, mine planning usually starts by assuming capacities for different units in the production process (i.e. mine, mill, refinery) and continue with attempts to determine the size of the pit. What follows after this initial step are the trials made by the design engineers to develop a plan of extraction through time until the depletion of the deposit. This plan must be within the limitations imposed by economic, physical, legal and geological constraints such that maximum possible NPV will be realized at the depletion of reserves.

2.2. TRADITIONAL APPROACH TO MINE PLANNING

Traditionally, the individual steps of mine planning are categorized as long range, medium range and operational (or short range) planning (Johnson, 1968; Pana, 1965; Pana and Carlson, 1966).

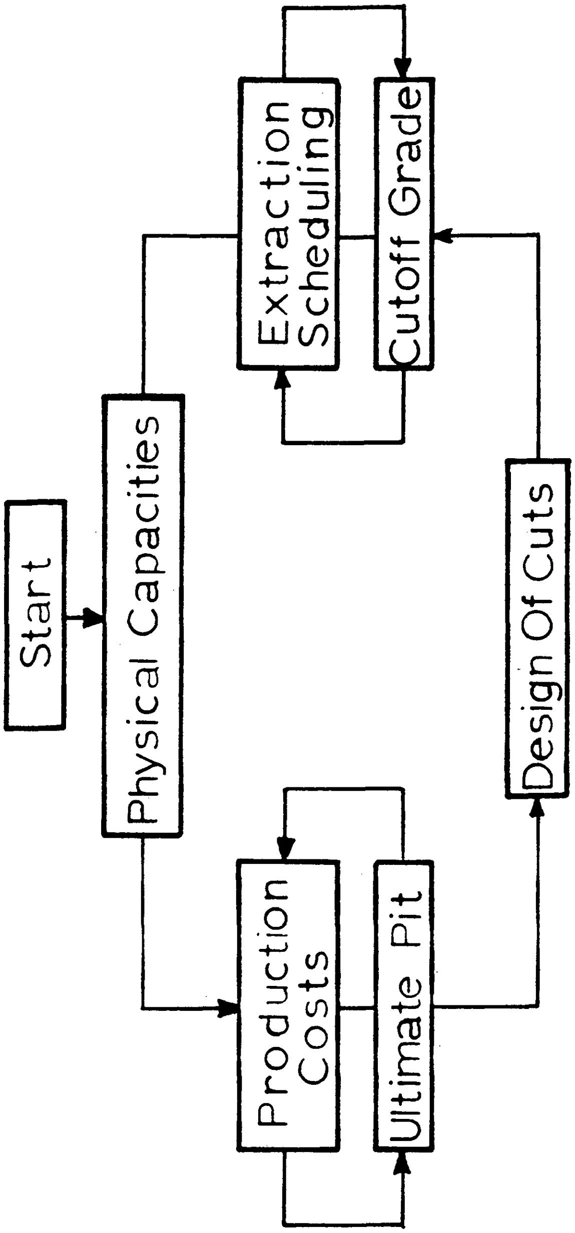


Figure 1. Steps of Traditional Planning by Circular Analysis

The traditional approach to mine planning, until the advent of the computer and its general acceptance was the trial and error hand calculated, cross sectional method (Koskiniemi, 1979; Soderburg, 1968). Using this method, the deposit is divided into cross sections, and the pit limits are drawn on the cross-sections by visually observing the ore grades and considering pit slope angles, geology and the economic objectives.

In an effort to locate the pit limits on a cross-section, a number of trial and error approximations are made; the trial pits are expanded or contracted to obtain a pit increment that satisfies the minimum required profitability. This profitability is directly related to the ratio of tons of waste to the tons (yards) of ore. This ratio is commonly known as the break-even stripping ratio (Soderburg, 1968) and defined as:

$$\text{B.E.S.R.} = \frac{\text{(Recoverable value)/(ton ore) - Production Costs/ton ore}}{\text{Stripping Cost/ton waste}}$$

(In addition to the above formula, the definition of break-even stripping ratio sometimes include a minimum acceptable profit as a cost.)

The classification of ore and waste within the increments are made by calculating a fixed break-even cutoff grade. Based on this classification, tons of material within the various zones are estimated by planimetering the areas on the sections (Pana 1965, Johnson 1968). The sections are then put together such that they make a workable pit by smoothing between sections. As discussed by Lane (1964), Johnson (1968) and Mason (1984), there are many limitations in mine planning with the traditional methods, especially with respect to the basic foundations that these traditional methods rely on. For one, the basis for the final pit limits is an arbitrary trial and error process which cannot assure optimality. Second, the concepts of constant cutoff grades and break-even stripping ratios have serious short-comings because these do not maximize discounted cashflows.

2.3. COMPUTERIZED APPROACH TO MINE PLANNING

In recent years, however, most of the mine planning techniques traditionally done by hand have been replaced by computer techniques. The computerized approach to planning mines eliminated some of the limitations imposed by traditional techniques. The most important changes are discussed in the following sections.

2.3.1. Block Model Development

In recent years, most mine plans for open pit mining are based on geologic and economic block models. The block models are developed by partitioning the deposit into rectangular blocks (Barnes 1982; Carlson, et al., 1966; Chen 1976; Johnson 1968; Pana 1965; Crawford 1979) (Figure 2). The various grades and material characteristics are assigned to the blocks by interpolating the drill hole information. In recent years often inverse square distance and the kriging are used as the interpolation technique for geologic block model development (David, 1975, Journel 1978).

The information assigned to a given block in the geologic block inventory are the grades of different minerals existing in the orebody and their respective recoveries when processed (sometimes volumes). Crawford [12] provides a list of items which are normally included in a typical geologic block model for a porphyry type copper deposit (Table 1). For a more detailed discussion of block models one can refer to Stanley (1979).

The information required to convert the geologic block model into an economic block model is given in Table 2. The typical information contained in Table 2 are the various

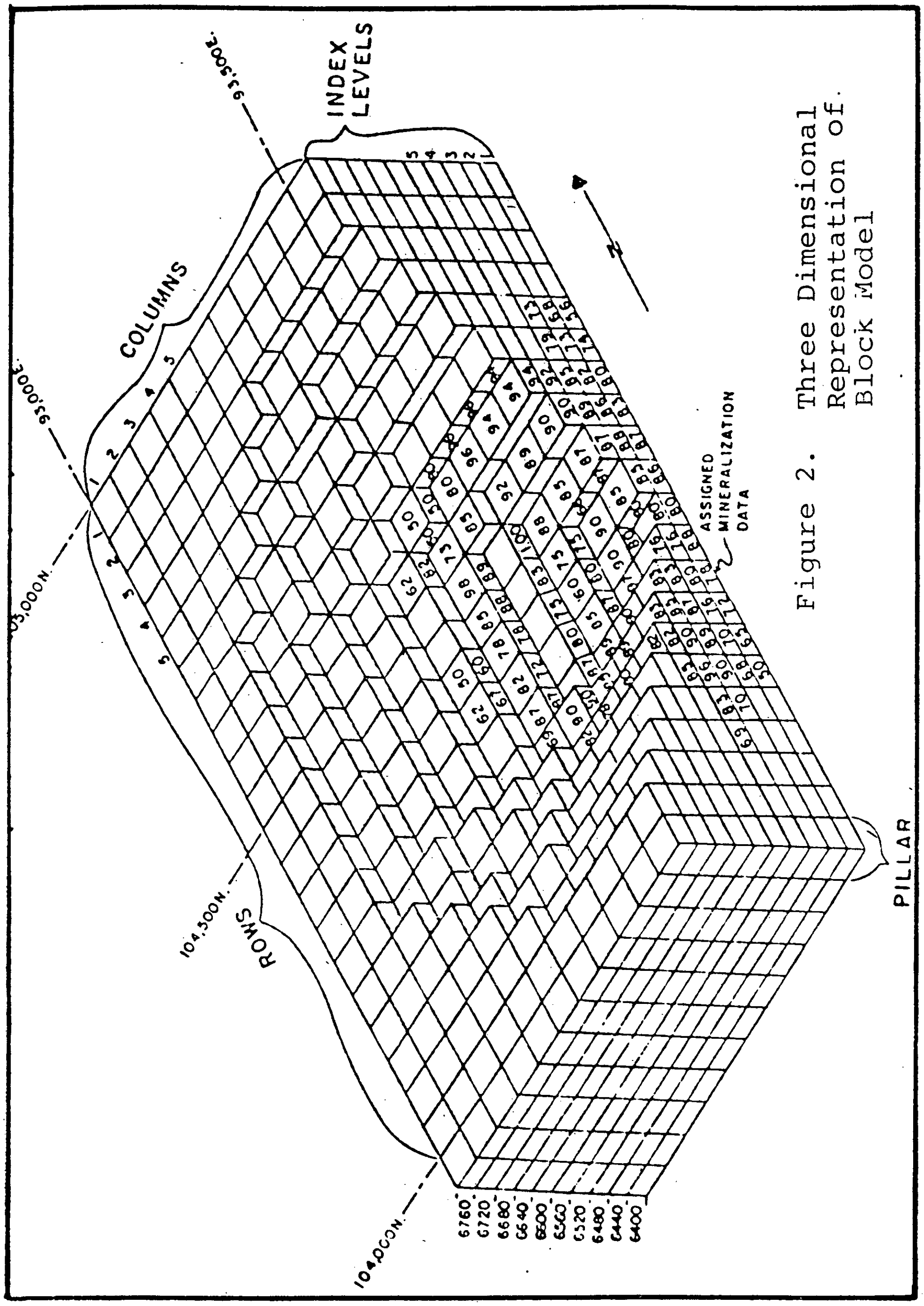


Figure 2. Three Dimensional Representation of Block Model

Table 1. Information Contained in a Block of Typical Copper Porphyry deposit. (After Crawford, and Davey 1979)

Items	Units
Copper (Cu)	%
Molybdenite (MoS_2)	%
Gold (Au)	oz/ton
Silver (Ag)	oz/ton
Copper Concentrate Recovery	%
Copper Concentrate Grade	%
Copper Smelting Recovery	%
Copper in Blister	%
Copper Refining Recovery	%
MoS_2 Concentrate Recovery	%
MoS_2 Conversion Recovery	%
Gold Concentrate Recovery	%
Gold Refining Recovery	%
Silver Concentrate Recovery	%
Silver Refining Recovery	%

Table 2. List of Unit Cost Items for Block Evaluation. (After Crawford and Dawey, 1979)

Drilling	$/ton ore and waste
Blasting	$/ton ore and waste
Loading	$/ton ore and waste
Hauling	$/truck hour
Haul roads	$/truck hour
Waste dumps	$/ton waste
Pit pumping	$/ton ore
Mine general	$/ton ore
	$/ton waste
Ore reloading	$/ton ore
Ore haulage	$/ton of ore
Concentrating	$/ton or ore
Concentrate delivery	$/ton concentrate
Smelting	$/ton concentrate
General plant	$/ton ore
Blister casting loading and freight	$/ton blister
Refining	$/ton blister
Selling and Delivery	$/lb refined copper

Metal Prices

Copper	$/lb	Gold	$/oz
MoS_2	$/lb	Silver	$/oz

cost items and prices of the metals contained in the deposit.

Based on the economic information given in Table 2, and grade, recovery and volume information given in Table 1, the economic value of each block is determined. The mining costs are further modified according to some linear function of elevation of the block being evaluated to take cost variations with respect to mining depths into account.

It is very important to note here that all the variables are in current year dollars and therefore the value of each block is the value one would get if the block is mined today. Also the information given in tables 1 and 2 is only provided as an illustration and will of course vary for different operations.

2.3.2. Ultimate Pit Limit Analysis

Ultimate pit limit analysis is one area of mine planning which has received a lot of attention in terms of computer use in recent years. There are many efficient algorithms and computer programs developed to determine the ultimate pit limits.

The computerized ultimate pit limit algorithms are classified by Kim (1979) as either heuristic or true optimizers. Kim (1979) defined heuristics as methods which work in most of the cases but lack rigorous mathematical proof. On the other hand the true optimizing algorithms are mathematically proven techniques that are guaranteed to find the economically optimum pit.

The list of ultimate pit limit algorithms can be given as follows.

a.) Moving Cone Heuristics Algorithms

1. Moving Cone of Kennecott (Pana, 1965; Pana and Carlson, 1966)
2. Heuristic Moving Cone by Lemieux (1979).
3. Heuristic Moving Cone by Mario and Slama (1973).
4. Heuristic Algorithm by Phillips (1973).
5. Heuristic Algorithm by Korobov (1974).

b.) Dynamic Programming Heuristics

1. Lerch and Grossmann 2-D Dynamic Programming (Lerchs and Grossmann, 1965)
2. Johnson's 3-D Dynamic Programming Heuristic (Johnson, 1970; Johnson and Sharp, 1971)
3. Barnes' Best-Valued Dynamic Programming (Barnes, 1980 and 1982)

4. Braticevic 3-D Dynamic Programming (Braticevic, 1984)

c.) True Optimizing Algorithms

1. Lerchs and Grossmann's 3-D Graph Algorithm (Lerchs and Grossmann's; 1965, Gilbert, 1966; Lipkewich and Borgman, 1969; Chen, 1976)
2. Johnson's Network Flow Algorithm (Johnson, 1968; Davis and Williams, 1973)

The moving cone methods are the most commonly used heuristics in the mining industry because they are simple to understand and simple to implement on the computer.

The moving cone algorithm is a simple trial and error process of analyzing many pits by moving the vertex of the inverted cone from one positive block to another. At each positive valued block in the economic inventory, a set of blocks are identified by a cone with its sides designed to comply with maximum slope angles of an actual pit and with its vertex being on the positive block. If the sum of the value of all the blocks enclosed within the cone is positive then these blocks are made part of the overall optimum pit. The algorithm stops after all the cones centered on positive blocks are evaluated. Despite its popularity, one of the problems with the moving cone method is that since the vertex of the cone is positioned on a single positive block, the evaluation of a particular cone cannot take into consideration the possible contribution of other cones. Barnes (1982) and Gauthier and Gray (1971) gave examples of cases where the moving cone algorithms, because of what is referred to as "the effect of overlap", will not identify the optimum pit.

The second type of heuristic algorithms are based on the dynamic programming concept. The Learch-Grossman 2-D algorithm is a method to determine the final pit limits on a 2-dimensional cross-section. The other dynamic programming heuristics are based on the original 2-D Lerchs-Grossman algorithm and they try to find the 3-dimensional pit limits by repeated applications of the 2-D dynamic programming algorithm. Barnes (1982) showed that these dynamic programming algorithms, while not optimizing routines, will provide good outer bounds on the ultimate pit limit.

As to true optimizing algorithms there are two. The first one was again developed by Lerchs and Grossman (1965) and is based on the concepts of graph theory. The application of this algorithm is reported by Lipkewich and Borgman (1969) and by Chen (1976). The second one is the network flow algorithm which was developed by Johnson (1968) based on the theory of network flows. Its application is reported by Davis and Williams, (1975) and Williams (1974).

Whether it is moving cone or true optimizing ultimate pit limit routines all these algorithmically different methods have a common objective. The objective of all the algorithms is to find a pit such that the sum of the undiscounted block values contained in the pit is maximized; i.e., the optimal pit.

2.3.3. Design of Incremental Cuts

Having defined the ultimate pit limits by computerized techniques, the next step in mine planning is to design a extraction sequence for the deposit. The sequence of cuts are like road maps to be followed in mining a deposit (Figure 3). They are needed for orderly extraction and development of the mine through the years. They also lay the necessary foundation for determining the cashflow distribution from the project.

There are two reported approaches to design incremental cuts by computers. One is by using a interactive computer program referred to as "cone miner", and the other is by using one of the ultimate pit limit algorithms repeatedly, on a modified block model.

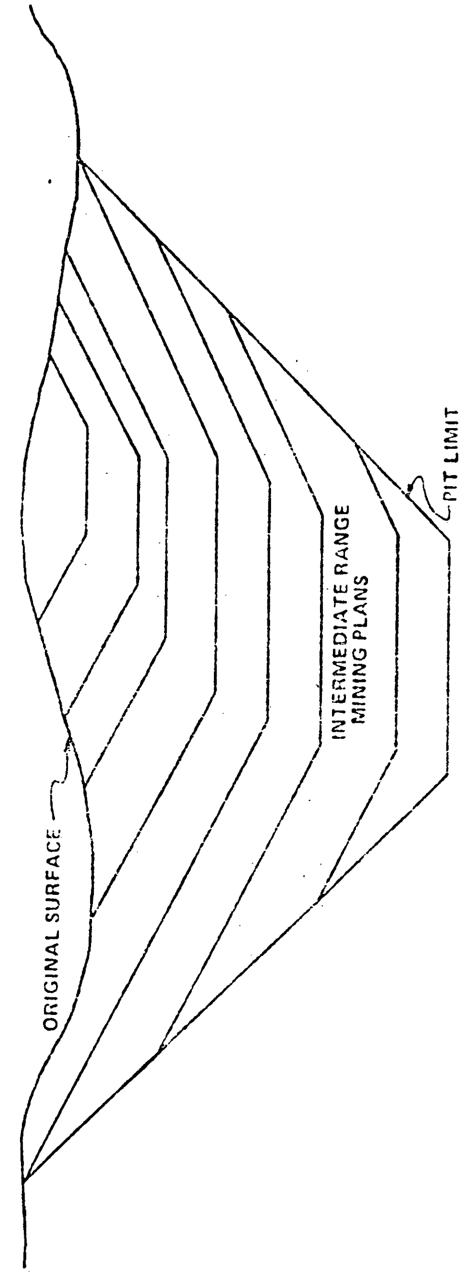


Figure 3. Intermediate Range Mining Plans.

The first use of the computerized "cone miner" approach is reported by Kennecott engineers (Pana, 1965; Pana and et al., 1966). Later Fairfield and Leigh (1969) described its use in RTZ Consultants. The most recent description of the use of the "cone miner" in sequencing of cuts is given by Mathieson (1982).

The "cone miner" is an interactive computer program used in order to generate incremental pits starting from an initial pit position and leading toward the ultimate pit. The incremental pits are generated by the "cone miner" for a given location of a polygon base and pit slopes. The total values of all the blocks falling within the cone, the total tons of material and its average grade are calculated by the program and outputed for an engineer's evaluation.

Analyzing through different incremental pits by way of fast computer calculations, and mining the "next best" pit increment in the deposit, yields a sequence of phases or cuts leading towards the ultimate pit limit.

Another method which is employed for medium range planning is the repeated use of existing ultimate pit limit algorithm on a modified block model to obtain nested pits. As reported by Crawford (1976), and Mathieson (1982), this approach is based on the fact that increasing or decreasing either cutoff grades or the price of a metal, the size of the pit can be expanded or contracted. By successively changing cutoff grades, a number of nested incremental pits can be generated which can in turn be used as the sequences to be followed in development of the pit.

2.3.4. Cutoff Grades

The cutoff grade is defined as the criterion used to distinguish between ore and waste in a mineral deposit (Lane 1964, Johnson 1968).

In many cases, grade tonnage curves are determined for each cut increment during planning of cuts (Lilico 1973). Having obtained these, the next step in planning is to schedule various ore and waste extractions such that unit operations of the mining system are compatible with one another in providing the maximum NPV (i.e., all constraints are satisfied and the objective of maximizing NPV is realized).

The classification of ore and waste in each increment requires the decision variable, cutoff grade, to be defined.

The cutoff grade used to classify ore and waste for production scheduling purposes is generally taken as a fixed break-even cutoff grade (Milner, 1977; Mathieson, 1982; Mason, 1984). The break-even cutoff grade is defined as the minimum grade of ore such that the revenues generated just cover all the costs of mining and processing.

2.3.5. Production Scheduling

Having determined various cuts to be developed and the cutoff grade to distinguish between ore and waste, the next step in open pit mine planning is to develop a schedule of extraction during the life of the mine.

To accomplish this, a deposit is analyzed bench by bench and the quantities of ore and waste material in each bench are tabulated by phases or cuts as seen in Table 3. Table 3 is a hypothetical tonnage inventory of cut sequences as illustrated in Figure 4.

The timing of increments and the outline of yearly mining schedules are determined by considering various constraints such as ore production requirements, waste removal capacities and the average mill feed grade requirements. This is normally done by utilizing a hand held calculator or a computer by a trial and error approach. Such hypothetical annual production schedules are illustrated in Figure 5.

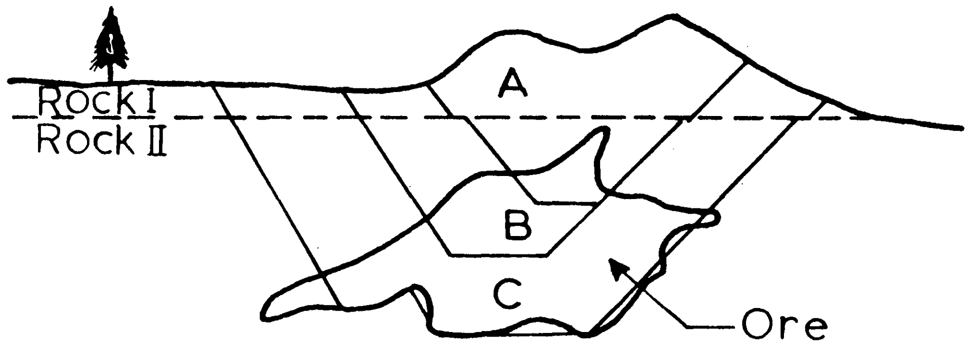


Figure 4. A Hypothetical Pit Development Shown in Cross-Section

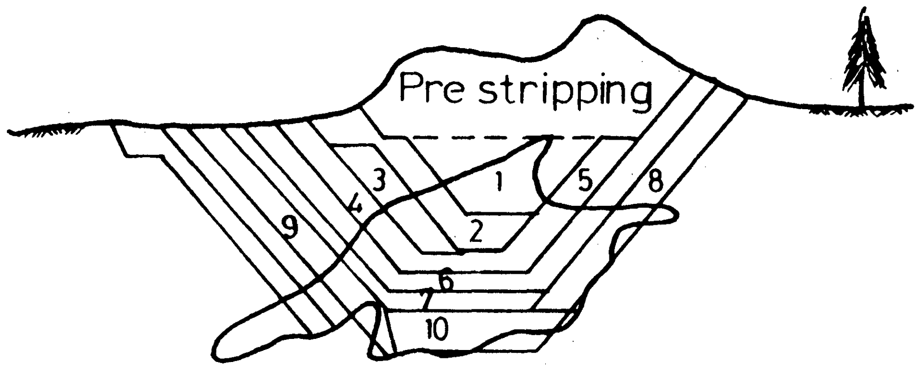


Figure 5. Annual Production Schedules of Cuts Shown in Cross-Section

Table 3. Hypothetical Tonnage Inventory of Cut Sequences Given in Figure 4

	THOUSANDS OF TONS								
	Phase "A"			Phase "B"			Phase "C"		
Bench	Cu	Ore	Waste	Cu	Ore	Waste	Cu	Ore	Waste
3100	--	--	20,000						
3050			32,000						
3000			45,000						
2950			40,000			3,000			4,000
2900			20,000			15,000			16,000
2850	9	2,000	4,000			20,000			18,000
2800	15	3,000	5,000			16,000			21,000
2750	25	5,000	2,000	34	8,000	4,000			17,000
2700				35	8,000	4,000	37	10,000	3,000
2650				31	7,000	3,000	34	10,000	4,000
2600				29	6,000	2,000	68	18,000	6,000
2550							96	24,000	3,000
2500							31	7,000	1,000

2.3.6 Summary

The changes that the application of computers brought in mine planning methods can be summarized as:

1. The 2-dimensional cross-sections were replaced by 3-dimensional geologic and economic block models (Pana 1965; Pana and Carlson 1966; Johnson 1968; Gauthier and Gray 1971; Chen 1976; Crawford 1979, Stanley 1979; Barnes).

2. The trial and error break-even stripping ratio analysis of ultimate pit limits are being phased out and 3-dimensional ultimate pit limit algorithms are taking their place (Pana 1965; Lerchs and Grossmann 1965; Pana and Carson 1966; Gilbert 1966; Johnson 1968; Libkewich and Borgman 1969; Johnson and Sharp 1971; Marino and Slama 1973; Phillips 1973; Korobov 1974; Chen 1976; Lemieux 1979; Barnes 1982; Braticevic 1984).

3. Medium range incremental pit development sequences are carried out by interactive computer programs such as a "cone miners" (Robinson and Prenn 1973; Milner 1977; Mathieson 1982; Decker, Garg and Steel

1984) or by the repeated use of ultimate pit limit computer programs on the modified block models (Crawford 1979; Barnes 1980, 1982; Mathieson 1982).

2.4 SHORTCOMINGS OF CURRENT MINE PLANNING PRACTICES

Although the application of computer methods overcame some of the limitations imposed by traditional techniques in obtaining an optimum mine plan most of the improvements came solely in the way of speed of data manipulations and calculations. An examination of computerized techniques reported in the literature reveals that even though the techniques of mine planning have appeared to change, the fundamental concepts have not. The building blocks of mine planning are still the break-even analysis and a trial and error approach.

2.4.1 Shortcomings of Break-even Pit Limit Analysis

The ultimate pit limit analysis performed by any one of the open pit algorithms mentioned in section 2.2.2 give the break-even pit, i.e., the pit where overall revenues from the pit are maximized by mining every increment of material for which revenues are at least equal to or greater than the costs. The inherent assumption made in determining these revenues and costs is that the deposit can be mined and processed instantaneously with unlimited capacities.

There is not a single mine where the capacities of various subsystems, grades and other requirements do not constrain the mining process. Hence, the size of the pit determined by assuming an unconstrained mining system may not represent the actual mineable reserves, when the constraints are imposed and the objective is to obtain a return on investment at least equal to the discount rate. The pit limits obtained by the ultimate pit limit algorithms may over estimate the size of the mineable reserves by including the increment of ore whose contribution is less than the rate required on a given investment.

The true optimum pit limit is a function of the production schedule for the mine and depends on the limitations imposed by the various subsystems; its true location can only be determined by taking into account complete interactions between the subsystems and the grade variability of the deposit. Without these considerations, the ultimate pit limit obtained by sophisticated computer programs may be an inferior NPV pit which maximizes the undiscounted profits rather than the one that maximizes the NPV of the operation.

There is nothing wrong however with this type of analysis if the ultimate pit limit determined this way is viewed as the probable reserve picture of the operation. There are serious limitations when this is not the case and this pit outline is viewed as the true optimum pit which will provide maximum discounted returns for the operation.

2.4.2 Shortcomings of Fixed Cutoff Grades

Cutoff grade is defined as the criteria used to distinguish between ore and waste in a mineral deposit. The grade at which marginal revenues equal marginal costs (break-even grade) is considered to be the lowest grade of ore one should mine and process (Vicker, 1961). Hence this grade is traditionally taken as the cutoff grade (see figure 6).

Inherent assumptions in using the break-even cutoff grade is again the assumption of instantaneous mining and milling with unlimited capacities. The decisions made based on the break-even cutoff grade make no attempt to consider the limitations imposed on the mining system by various mining, milling and smelting capacities and interactions they make the deposit geology.

When the economic objective is the maximization of discounted profits (NPV's) the determination of what is an

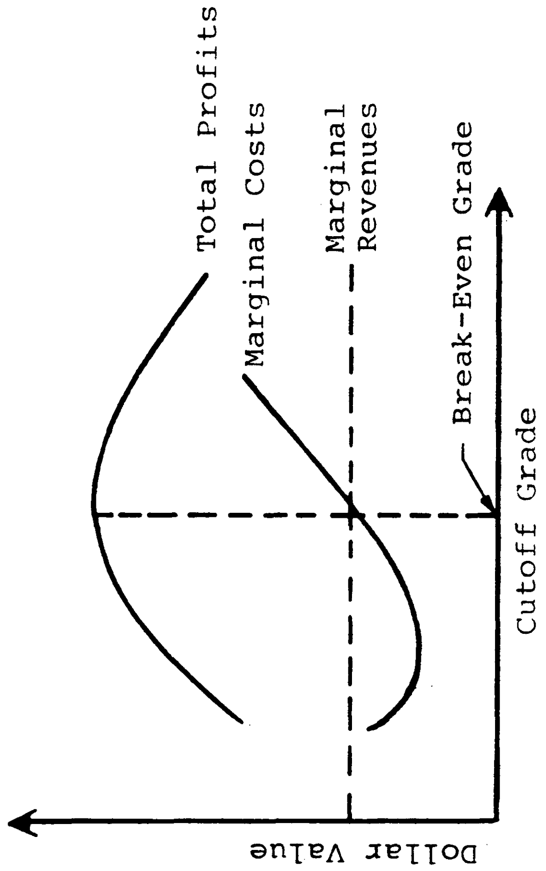


Figure 6. The Cut of Grade and Break-Even Point

ore block depends upon when that block of material will be mined and the value of the blocks which follow it. Since the timing of when a block of material is to be mined depends on the limitations imposed by system capacities, and their interactions with the grade distribution of the deposit, the cutoff grade decision must be taken as a function of the production schedule if the objective is to maximize the discounted cash flows.

Hence, applying a fixed break-even cutoff grade to distinguish between ore blocks and waste blocks does not always lead to the best mining schedules because maximization of undiscounted profits does not necessarily give the best economic results when the objective is to maximize NPV of the project and the system is constrained.

It is not only the above assumption which imposes limitations on the breakeven cutoff grade. Another underlying assumption in using the fixed break-even cutoff grade has been the fixed cost assumption. The costs are assumed to stay constant not only through time but also through the deposit. Doing an analysis by assuming costs to be the same throughout the deposit is not a realistic assumption.

The following examples will illustrate the issues raised in the preceding discussion.

2.4.3. Cutoff Grade Example

Mason (1984) gave an excellent example of a case showing why the analysis of cutoff grades cannot simply be based on a simple break-even analysis when the objective is to maximize discounted profits. His example will be included here.

Assume

1. Mining system is concentrator limited,
2. Mining capacity is unlimited,
3. Refinery capacity is unlimited,
4. The selling price and cost of mining and processing are constant.

Assumption 1, 2 and 3 state that the production rate from this operation will be determined according to what the concentrater can process.

Assume that an orebody as shown in Figure 7 is to be mined in three push-backs going from the top cut 1 to cut 3. The ore and waste are intermixed in the most part and the objective is to determine the cutoff grade in order to separate ore from waste in each cut such that maximum economic outcome is realized.

Given the above assumptions, the cutoff grade chosen for each cut will determine what proportion of the cut is ore and what proportion is waste. As a result the time

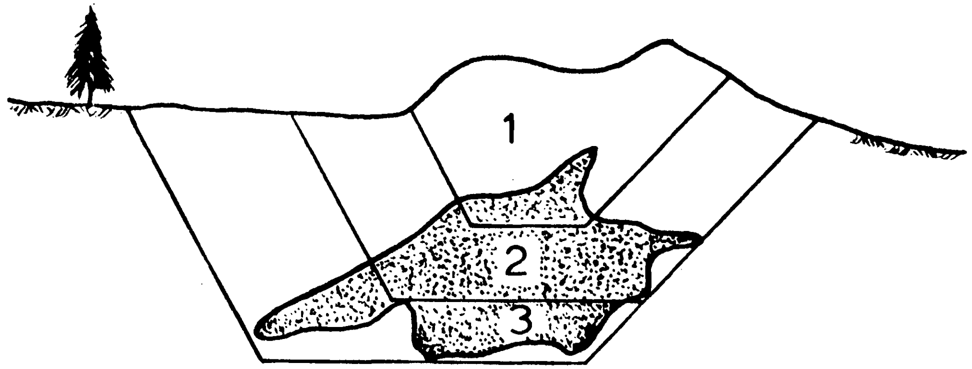


Figure 7. Sequential Pit Development and the Location of the Ore

required to mine the individual cuts will depend on how much ore is in the cut (therefore it depends on the cutoff grade) and how fast it can be concentrated.

The analysis of the cutoff grade in cut 3 is the simplest one to analyze because decisions with respect to this cut are not influenced by any future cuts. If the cutoff grade is set to zero in this cut, all the material in the cut would be treated as ore and the value of the cut will at best be some small amount because much of the material concentrated would cost more to process then the revenues they would generate. If the cutoff grade is increased slightly, some of the non-profitable material will be sent to the waste dump instead of the concentrater and as a result the value of the cut will increase.

The value of the cut will continue to increase with higher cutoff grades until the "break-even" cutoff grade is reached.

At the break-even grade, the material mined will generate just enough revenue to exactly cover its own variable costs for processing and marketing. If the cutoff grade is raised above this point, some material which could have made a contribution to the overall value of the cut would be sent to the waste dump and therefore the value of the cut would be reduced. The typical value of cut 3 vs cutoff grade is shown in Figure 8.

As indicated by Mason (1984), for the analysis of cut 2, however there are additional complications. Analyzing the cutoff grade of cut 2 in the context of NPV indicates that the cutoff grade determination in cut 2 must consider its effect on the value of cut 3. The mining of cut 2 with a cutoff grade somewhat higher than the break-even grade will require some of the ore which is making only a slight contribution to the value of the cut to be thrown away. As a result, the cut would have less ore to be processed and the cut would be mined sooner, therefore the cash flows from cut 3 would be available earlier. For a small increase in the cutoff grade the value of bringing forward later cash flows will outweigh the value lost by throwing away material which is slightly above the break-even grade. As one increases the cutoff grade further, more valuable material will be thrown away and eventually the value lost will exactly balance the gains from bringing forward cut 3. At this cutoff grade the combined values of the two cuts is a maximum. If the cutoff grade is raised further, the value lost by throwing away material from cut 2 will outweigh the gains from bringing forward the cut 3. This analysis of cut

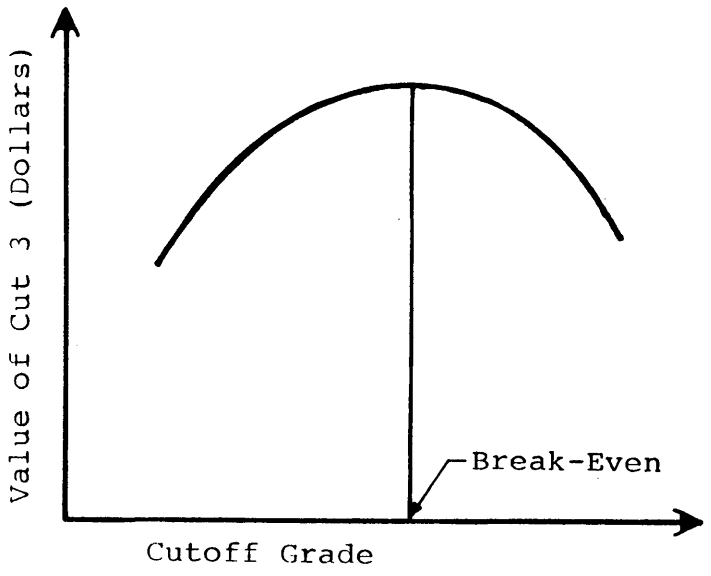


Figure 8. The Cutoff Grade vs. Value of Cut 3

2 with respect to varying cutoff grades is illustrated in Figure 9.

The mining of cut 2 with a cutoff grade below the break-even point will cause two things to happen; (1) The value of cut 2 will be reduced because non-contributing material would be mined and processed, and (2) The value of cut 3 will be reduced by delaying its cash flow. Therefore, mining cut 2 with a cutoff grade below the break-even would result in a serious reduction of the total values of cut 2 and cut 3.

The major conclusions to emerge from this cutoff example is that the cutoff grade in a cut is a function of future values of cuts, yet future values of cuts are also a function of the very cutoff grade one is trying to determine. Therefore, the cutoff grades and production scheduling are two interdependent variables that must be determined with respect to one another. The cutoff grade determined by considering the value of future cuts will generally be higher than the simple break even-cutoff grade.

2.4.4. Cutoff Grades As a Function of Total Mining System

The cutoff grade analysis in the previous example was based on the assumption that the mine was only constrained by the concentrator capacity and therefore the production

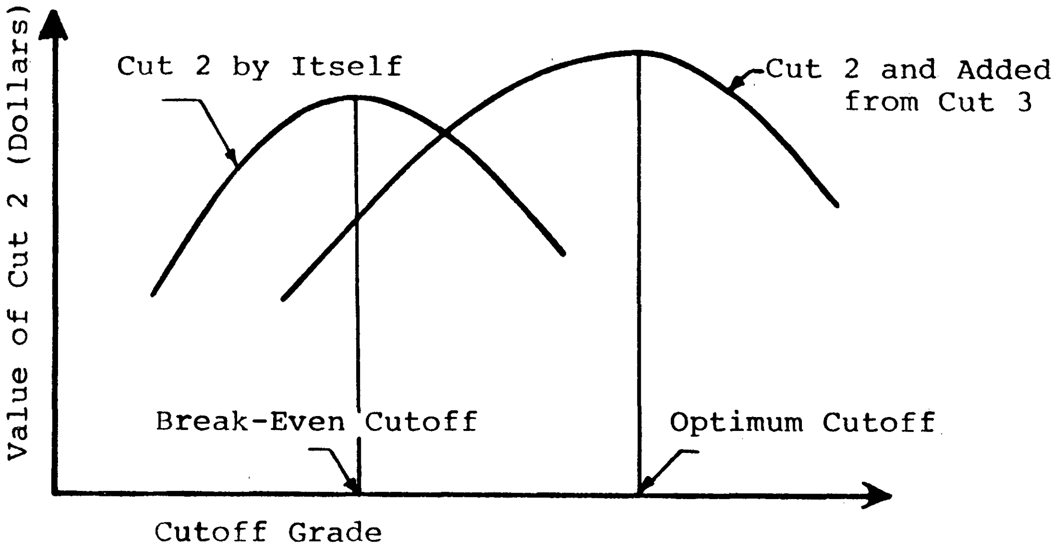


Figure 9. Cutoff Grade vs. Value of Cut 2

rate was controlled by the concentrator (remember, the mining and refinery capacities were unlimited). These assumptions in essence are very simplistic and were made purposely to make a point that there is an important difference between the break-even cutoff grade which maximizes undiscounted profits and the optimum cutoff grade which maximizes the total discounted profits when the system is constrainted.

In a real mining situation, however, the assumptions of the previous example may not be true, the mining system will not only be constrained by a single concentrator capacity but also by the capacities of other subsystems of the production process. Depending upon the grade distribution of material being mined, the production rate, i.e., the timing of the revenues in future cuts, will be controlled not only by the concentrator capacity but also by the limitations imposed by other sub-units such as mining capacity of ore and waste and smelter capacity.

Lane (1964) was one of the pioneers who realized this and showed that the cutoff calculated by way of the cutoff grade example of the earlier section was actually one of six cutoff grades one would need to analyze in choosing the optimum. To see this let us look at the analysis of what he defined as balancing cutoff grades.

Lane's theory of cutoff grade is based on a mining model which consisted of three stages: mining, concentrating and refining. He viewed the mine as the area where ore is extracted from the ground, the concentrator as a facility where the crude ore is upgraded by various processes to a concentrate and a refinery as a plant where the concentrate goes through the finishing process to produce a product which is ready for manufacturing. In this model, each stage is assumed to have its own associated cost and limiting capacity.

Before going into this full model consisting of three stages, consider a new model where the mining system consists of two stages, mine and mill. In this context, the analysis of cutoff grades in cut 2 of Figure 9 will be carried out again, this time assuming the hypothetical deposit to be mined under the limitations of both the mining and the concentrator capacities.

Since the mine and mill both might have restricting capacity, depending on the grade distribution in the cut, the time it will take to mine cut 2 will be a function of the cutoff grade chosen. For instance, if the cutoff grade chosen defines most of the cut 2 material as ore, and concentrator capacity is greater than mining capacity the ore would be processed as fast as it is mined. Therefore the system will be limited by the mining capacity. On the other hand if the concentrator capacity is smaller than the mining capacity and the cutoff grade chosen again makes the major portion of cut 2 ore, then the system would be mined and processed as fast as the concentrator can process the ore, therefore the system would be limited by the concentrator. Now let us consider this last case again. This time assume that the cutoff grade chosen classifies most of the material in the cut as waste and very little as ore. Having the concentrator capacity less than mining capacity may not make this system concentrator limited because most of the mining capacity will still be engaged in removing waste. Therefore the time it takes to mine the increment may still be determined by the mining capacity.

Therefore, depending upon what cutoff grade is chosen and the grade distribution existing in the increment, one of the units in the mining system will determine when cut 3 can be mined. Therefore the limitations imposed by the capacity of various units in the system have a direct influence on the combined value of cut 2 and cut 3 and, as a result, influence the choice of cutoff grade in cut 2. Since the value of the cutoff grade in cut 2 is not known, its value needs to be chosen such that value of cut 2 will be maximum and at the same time its discounting effect on the value of cut 3 kept at a minimum.

Figure 10 shows two curves each representing the NPV of cut 2 and cut 3 at different cutoff grades. The curve labeled as Vm is the case when cut 2 is mined at the rate determined by the mining capacity and the curve labeled as Vc is the concentrator limiting case and represents the situation when the cut 2 is mined at the rate determined by the concentrator capacity.

There are three cutoff grades which are of interest on these curves. They are indicated as g_m, g_{cm} and g_c. The cut off grade g_m results in the maximum combined discounted value when cut 2 is mined at maximum mining capacity. The grade g_c is the cutoff grade which gives maximum discounted value when cut 2 is mined at the maximum concentrator capacity, the g_{cm} is the cutoff grade at which there is enough ore and waste such that both the mine and mill can operate at full capacity in mining of cut 2.

If one chooses g_m as the cutoff grade in mining cut 2, the mining system as seen in Figure 10 will be concentrator limited i.e., cutoff grade g_m will proportion ore and waste in cut 2 in favor of ore such that milling of this ore cannot be accomplished as fast as it is mined and therefore it will take longer to process the cut than to mine it. And

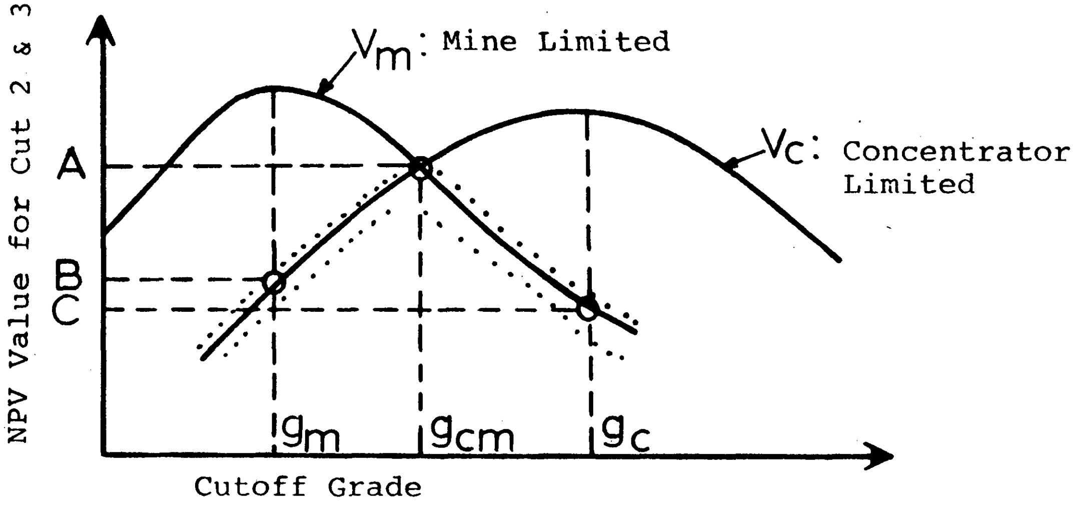


Figure 10. Cutoff Grade vs. Value of Increment 2 and 3

the best the operation will be able to do in terms of combined NPV value for cut 2 and cut 3 will be at the point B on the NPV-axis. If the cutoff grade g_c is chosen as the cutoff grade to be used in cut 2, most of the material in cut 2 would be considered as waste. Therefore most of the mining capacity will be engaged in removing waste material and therefore ore removal will not be able to keep up with the concentrator's daily tonnage requirements. At this cutoff grade the system is said to be mine limited and the combined value of cut 2 and cut 3 will at best be at point C on the NPV-axis.

At the cutoff grade g_{cm}, however, both NPV curves intersect with each other. This indicates that both the mine and mill will operate at their full capacity in removing and processing the cut 2 material. This cutoff grade, g_{cm}, will proportion the material in cut 2 such that the mining capacity can provide enough ore to the concentrator as well as removing all the waste necessary to uncover this ore. This cutoff grade is referred to as the balancing cutoff grade by Lane (39).

In this hypothetical example, it will be most advantageous to mine cut 2 with the balancing cutoff grade of g_{cm}, because this is the grade where combined values of cut 2 and cut 3 will be maximized. This does not mean however that the balancing cutoff grades will be the optimum ones for all cases. As shown by Lane (1964), depending on the grade distribution of the cut being analyzed and the assumed capacities, the optimum cutoff grade can be any one of these three cutoff grades. Figure 11 shows a hypothetical case where g_m would be the optimum cutoff grade instead of g_{mc}.

If the model in the proceeding discussion is upgraded to include all three stages of the mining system instead of just two, the actual cutoff grade would have to be determined by choosing a cutoff grade from among 6 cutoff grades as shown in Figure 12. The grade values g_m, g_c, g_r are the limiting cutoff grades for the mine, mill and refinery and G_{mc}, G_{mr}, G_{cr} are the balancing cutoff grades between mine and mill, mine and refinery, and mill and refinery respectively.

As the preceding discussion has clearly shown there is more to the analysis of cutoff grades than simply calculating the break-even cutoff grade when the objective is to maximize the net present value of a deposit rather than maximizing total profits when there are capacity limitations involved. The decision as to what to consider as ore and waste in a mining increment of the deposit is a function of

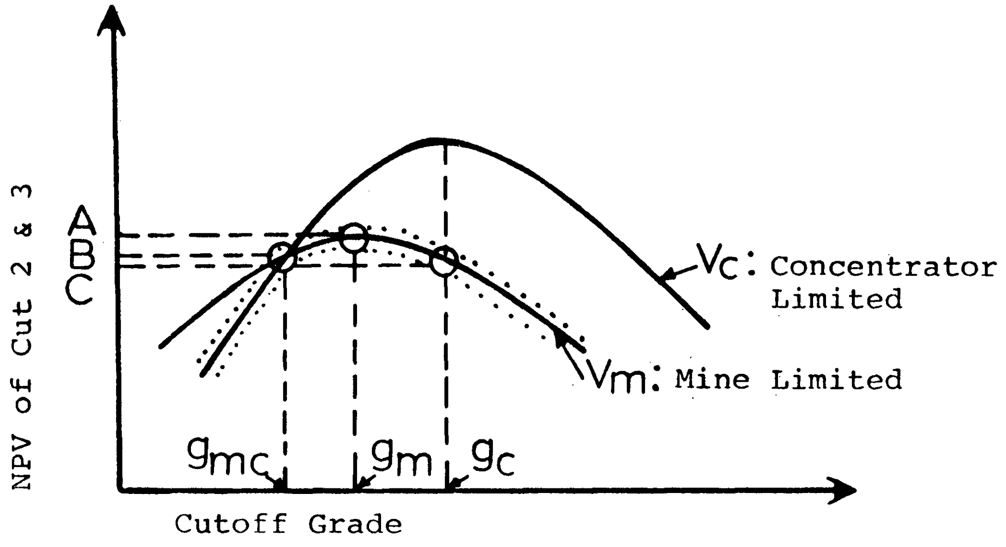


Figure 11. Cutoff Grade vs. Value of Increment 2 and 3 for Two Constraint Case

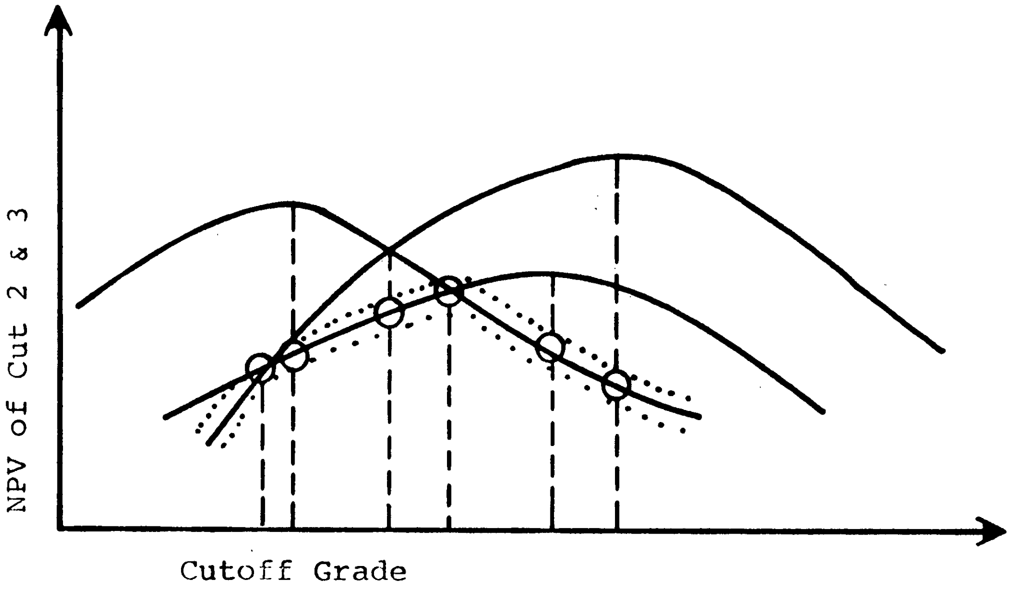


Figure 12. Cutoff Grade vs. Value of the Increment 2 and 3 for Three Constraint Case

the interactions of the mining units with the ore grade distribution.

But all too often, the decision as to whether a block is ore or waste is taken in the context of the break even cutoff grade without considering the influence of the entire system and other blocks in the system. It is obvious that a prior decision like this as to what is ore and what is waste will not always lead to the best mining schedules. The model which will be proposed in this work will eliminate this a priori decision process and will provide the context where this decision is made as a function of the total mining, milling, refining system.

2.3.5 Summary

The economic break even analysis has significant merit when taken in the context of maximizing total profits from a deposit when the total mining system is assumed to be unconstrained. However, it has serious drawbacks when taken in the context of NPV analysis and when the problem is constrained such as in actual practice.

As pointed out by Carlisle (1954) the pit which is optimized on the basis of break even analysis will result in a lower NPV than the maximum attainable NPV. The actual maximum NPV pit will always be smaller than the break even pit obtained for the unconstrained case.

The break even cutoff grade assumes operations are not constrained by the capacity limitations of various subunits of the system and therefore does not consider future effects of this cutoff grade decision. As a result, the break even cutoff grade when used in production scheduling will at best maximize the undiscounted profits. Therefore development of any mine plan based on maximization of undiscounted profits when considered within the time value of money framework is an inferior production schedule (i.e., lower NPV for the operation) than the one attainable.

As a result, it can be concluded that even with the application of the computers, mining engineers are far from getting a plan that is best as long as:

1. One does not realize the ultimate pit limits obtained by the existing algorithms are the outline of the probable reserves and mistakenly treat this pit limit as the optimum contour where one will maximize the NPV value of the operation.

2. The cut generation and sequencing of cuts are designed on the basis of a trial and error approach

using an arbitrary profitability criteria (stripping ratios).

3. Production schedules are carried out by a trial and error approach based on fixed break-even cutoff grades.

2.5 HEURISTIC OPTIMIZATION TECHNIQUES

If a deposit is analyzed, with the traditional concepts or with computerized techniques by following the sequence of logic given in section 2.2, the deposit may be rejected because it would not meet a company's required minimum rate of return. Furthermore, designing a mine based on a single ultimate pit limit and a single fixed cutoff grade and a single sequence of cuts will usually result in lower returns than the mine is capable of generating.

As a result, planning a mine with the approaches described has many faults. Realizing these limitations, many attempts have been made to address and alleviate some of the problems through various heuristic, empirical techniques.

2.5.1. Concept of Profcost

It was pointed out as early as 1968 that the optimum ultimate pit limit which maximizes the NPV of the deposit may not be the same as one found by break even analysis (Ericson 1968, Halls, et al., 1969, Wells 1978). This realization might have resulted from observations that one can always find another pit, smaller in size, than the break even pit which gave higher discounted returns. Hence, in order to ensure an ultimate pit limit design which maximized discounted cash flows the concept of imposing some minimum profit from every increment of ore was suggested by Ericson (1968). (Incidently, the profit suggested by Ericson behaves like any of the cost items in the calculations and therefore is referred to as "profcost".)

Although it was not clearly understood at the time of its invention, the idea of a profit requirement serves as a parameter to reduce the size of the ultimate pit limit so that the pit found by ultimate pit limit algorithms is smaller than the break-even pit.

Even though the idea of profcost is effective in reducing the size of the pit, there is no guarantee that the truly optimum pit can be found by this method. The assignment of a wrong value to profcost can have a more serious effect than having the break-even pit at hand, by causing the size of the pit found to be smaller than the optimum one; hence, swinging things in the other extreme.

As a result of this difficulty of assigning values to profcost, various other methods were developed and suggested to find the NPV optimum pit.

2.4.2. Concept of Parametrization

Since the optimum size of the ultimate pit which yields the maximum NPV in a constrained system may be smaller than the break-even pit and its exact location cannot be known without knowing the optimum production schedule, current attempts to find the optimum schedule are based on a trial and error approach. Instead of basing the final overall mine plan on a single pit, a number of different size trial pits are generated. Full mine planning as pertaining to sequence design and production scheduling are then carried out considering all the pits generated, and choosing the one that gives maximum NPV (Halls, Bellum, and Lewis 1969, Wells 1978, Suriel 1984).

There are various approaches to finding these different size pits. One way uses successively increasing profcosts as mentioned earlier. As the profcosts are increased progressively the generated pits become smaller and smaller.

Another approach is to vary the cutoff grade for the deposit. It is common practice in mine planning to impose a planning cutoff grade on the geologic block model so as to differentiate between ore and waste blocks before the final mill or waste decision is made. By changing the value of a fixed cutoff grade and then preprocessing the blocks, one obtains a number of successfully different pits to analyze.

The generation of multiple pits by this method was first suggested by the Kennecott Engineers (Halls, Bellum, and Lewis 1969). Its use is reported by Blackwell (1971) in planning of the Bougainville deposit. Lilico (1973) also used this approach to demonstrate the concept of "optimization" in mine planning.

The most recent technique of generating multiple pits were suggested by Francois-Bongarcon (1978). This method, which is referred to as parametrization of reserves, works with original block grades and determines a series of different size pits, each containing the maximum quantity of metal for its size. The details of the algorithm are described in a number of publications (Francois-Bongarcon and Marachal, 1976; Francois-Bongarcon, 1978; Dagdelen and Francois-Bongarcon, 1982).

The advantages of the parametrization algorithm is that the generation of all the different pits is accomplished in one single run. This is in contrast to the previous procedures where the different size pits are obtained as a result of changing the same variable and applying the ultimate pit limit algorithm repeatedly. The disadvantage is that one has no idea where exactly the largest, upper bound, break even pit is unless the full economic analysis is carried out on all the pits found by the algorithm.

2.4.3. Alternate Sequences of Extraction

The design of extraction sequences starting from a known location in a deposit and leading towards the ultimate pit limit was discussed previously in section 2.23. There can be many different feasible extraction schedules in a reasonable size deposit leading towards the ultimate pit each having its own cash flow schedule depending upon the engineer's judgement and common sense.

These extraction schedules when coupled with the geology and grade distribution of the deposit will result in different ore and waste schedules. As a result each extraction schedule will produce a different realization of NPV for the deposit. Therefore, during mine planning with existing tools as described in 2.2.3, more than one plan of extraction needs to be analyzed. Referring to this aspect of mine planning for the Baugainville deposit Blackwell (1971) makes the following statement:

> The final target was to establish the sequence in which the orebody should be mined. As can be imagined, an orebody of this size gives fairly considerable scope for different permutations. In simple terms, however, there were two obvious extreme possibilities, which consisted of commencing operations at either end of the pit; the one possibility having a high grade and stripping ratio, the other extreme having these parameters with lower values. The choice of strategy was made by simulating these two extremes together with intermediate combinations.
>
> . . .
>
> In all, four mining sequences were considered.

Hence, it is common practice in heuristic optimization to do economic analysis on a number of different sequences of extraction for each pit generated by the parametrization algorithms.

Although looking at more than one sequence of extraction is a good idea, there is still no assurance that the best sequence is included among the ones generated. Furthermore, nobody can prove that mining "next best" ore increments in the development of the deposit will result in an extraction schedule which will maximize the NPV.

2.6 TRUE OPTIMIZING TECHNIQUES

As seen in previous discussions, there are many problems with the heuristic optimization techniques. Realizing the problems with existing heuristic optimization techniques, various attempts have also been made to formulate and solve the production scheduling problem as a mathematical program where the solution obtained is guaranteed to be the optimum.

These attempts can be listed as;

1) Johnson's Linear Programming Approach (Johnson, 1968).
2) Integer Programming Approach (Gangwar, 1973)
3) Asarco's Lagrange Multiplier Approach (Williams and Davis, 1973)

Although the production scheduling problem can be successfully formulated as a linear or integer programming problem, there have been problems in coming up with an efficient solution algorithm. When formulated as a large scale mathematical optimization problem (See Chapter 3) the number of variables and constraints involved with the production scheduling model of a typical mine is in the hundreds of thousands and it is not possible to efficiently solve large scale linear programs simply by applying commonly available solution algorithms. The existing algorithms work efficiently on small problems yet become ineffective when applied to large ones.

It has been discussed in the literature that sometimes the only way to solve large problems is to recognize their special structure (Lasdon, 1970). For some problems which do not possess a special structure optimum solutions cannot be obtained due to the large memory and computational time requirements. The classical problems such as airline scheduling problem and the traveling salesman problem in operations research are of this type (Lasdon, 1970; Murty, 1979).

Therefore in order to solve the production scheduling problem, past attempts have either tried to define the structure of the problem and made use of this structure in the solution algorithm (Johnson 1968, Davis and Williams 1973) or tried to fit this large scale problem into a smaller model (Gangwar 1974).

2.6.1 Dantzig-Wolfe Decomposition Algorithm

Johnson (1968) applied the Dantzig-Wolfe decomposition principle for solving large scale linear programming problems to the production scheduling problem. Johnson's method was to decompose the problem into two problems by including the capacity blending constraints in the master problem and the sequencing constraints in the subproblem. As a result, solving the subproblem became a matter of solving ultimate pit limit problems and solving the total problem thus became a matter of solving ultimate pit limit problems a number of times for successive iterations until conditions for the termination are satisfied.

In solving subproblems, Johnson proved that the structure of the single time period subproblem is a network flow problem and therefore suggested using the maxflow algorithm for solving ultimate pit limit problem. Since his work this algorithm is considered one of the true optimum ultimate pit limit algorithms available to the mining industry.

2.6.2 Pure Integer Programming Algorithms

Gangwar (1973) attempted to formulate and solve the production scheduling problem as a zero-one integer linear program. In order to take into account the uncertainties associated with grades assigned to blocks and for the demand of the final product, Gangwar (1973) introduced an extra chance constraint in the formulation of the problem. In this formulation, depending upon the choice of approximations made to convert a stochastic constraint into a deterministic one, the binary problem either becomes a non-linear binary integer model or binary linear model. Hence the solution methodology suggested by Gangwar (1973) included either the existing 0-1 integer programming algorithms applicable to non-linear programs or to linear integer programs. As such, as a possible solution algorithm, Gangwar suggested use of Lawler and Bell's (1966) binary programming algorithms for the nonlinear case, and Geofrian's (1966, 1967) implicit enumeration methods, Lemke's and Spielberg's (1967) direct search algorithms or Gomory's (1963) integer programming algorithm for the linear case.

It is important to realize here that all these algorithms are capable of generating solutions in a reasonable amount of time for only a small number of variables and constraints. As a result, realizing the limitations of the solution algorithms proposed to solve the production scheduling problem Gangwar optimistically said:

> All the available integer programming algorithms require the total number of binary variables to be kept reasonably small, in the neighborhood of 300-400 variables. For the open pit design problem this poses a serious limitation on the problem size. To circumvent this limitation for large pit designs we need to aggregate and disaggregate the blocks and the time span of the scheduling periods in a multistage programming approach; so that in

> the initial stage of planning (for the determination of the long range mining plans) the unit scheduling period can be 2 to 5 years and the unit block size will be kept as large as possible such that it can be mined, milled and processed in one time period.

The maximum number of blocks that can be considered with the direct application of an existing integer programming algorithm such as the one suggested by Gangwar (1973) is in the neighborhood of 30 to 40 blocks assuming a 5 year production scheduling span. This number of blocks, in this author's opinion, cannot realistically represent the problem being solved.

2.6.3 The Lagrange Multiplier Approach

In 1973, another attempt was made to solve the production scheduling problem by taking the special structure of the problem into account. This attempt came from Asarco's needs for a production schedule for their open pit mines: the problem was contracted out to Systems Control Incorporated in California (Williams, 1973).

In their attempt to come up with a procedure to determine which blocks of ore and waste should be removed in each mining period so as to maximize the discounted cash flow for the mine subject to the mining and milling operating constraints, S.C.I. used the idea of Langrange multipliers in coming up with a solution alogorithm to the problem. Although their application of Lagrange multipliers to the problem is very crude, at least they have expanded the idea of the penalty parameter concept introduced by Lerchs and Grossmann in 1965 and identified as the "assignment of Lagrange multipliers to the constraint set" by Johnson (1968).

Realizing the size of the full scale multi-time period production scheduling problem was extremely large, S.C.I. concluded it was not practical to optimize with respect to all mining periods and all types of constraints simultaneously. Instead, an approximate sequential decomposition and synthesis procedure was employed which treats the scheduling problem as a sequence of tractible problems. Thus the scheduling problem in the S.C.I. approach was solved one period at a time. By doing this, they lost the ability to take into account the effect of the time periods on one another.

2.6.4 Summary

The decomposition of the production scheduling problem by way of the Dantzig & Wolfe method may lead to optimum solutions which are in terms of the fraction of blocks being mined. This, in turn, can result in a production schedule which is not feasible for a given mining condition.

The integer programming approach (Gangwar, 1973) eliminates the shortcomings of fractional mining of the blocks yet poses more serious problems in terms of realistic modeling of the problem and as a result obtaining an optimum production schedule.

Asarco's lagrange multiplier approach alleviates the problems of the previous two methods but still solves the problems incrementally on the basis of single time periods. By solving the problem as they have done, the production schedule which will result in maximum NPV can be missed.

To further overcome the difficulties of the existing mathematical optimization approaches, the production scheduling problem will be formulated as a large scale linear programming problem in Chapter 3. Based on the structures of the problem, a solution algorithm based on the generalized lagrange multipliers concepts of mathematics will be proposed in Chapter 6 and 7. The proposed method is the extension of the concept which are already familiar to the mining engineers. Before going into the development of the proposed method the relationship between lagrange multipliers and cutoff grades will be introduced in chapter 5.

CHAPTER 3

MATHEMATICAL FORMULATION OF THE PRODUCTION SCHEDULING PROBLEM

3.1 INTRODUCTION

The set of values assigned to the critical mine planning variables such as capacities of different units in the production process, cut off grades and sequence of extraction define an economic outcome. They also define three distinct sets of actions required during the production stage. These are:

1. To mine or not mine a block of material;
2. When to mine a block if it is to be mined;
3. Once it is mined what to do with it (i.e. whether or not to process it).

The decision of when to mine a block of material and once it is mined what kind of action to take is commonly referred to as production scheduling and this is the basic purpose of mine planning as discussed in Chapter 2. The mathematical model which will provide mining engineers with the necessary guidance for the actions identified above will be referred to as the production scheduling model and the definition and formulation of this model will be the purpose of this chapter.

In coming up with a production scheduling model, such that analyzing the model will result in the best mining schedule, two factors must be carefully considered. (First, the model should be a representation of reality. Secondly, with the production scheduling problem as discussed in (Chapter 2, no a priori decision should be made as to whether a block should be mined or not, nor as to whether the block is ore or waste.) These decisions should be made by considering the influence of the entire system in the model.

Therefore, the mathematical model of the production scheduling problem being formulated must consider the state of the whole mining system by including all the pertinent geological, physical and economical constraints and factors.

With every mathematical model certain assumptions are made to represent the realities of the problem being solved. Under the guidance of the preceding discussion, the following assumptions are made for the production scheduling model formulated in this dissertation.

3.2 ASSUMPTIONS OF THE MODEL

Assumption 1.

It is assumed that the mineralized orebody can be divided into a finite numbers of rectangular blocks of material, and the expected value of average grade for each block can realistically be estimated.

Assumption 2.

The usual open pit mining process proceeds in a manner such that the practice of benching gives the sides of the pit a step like structure. Thus a block of any mineable shape having vertical sides is acceptable as long as its vertical height does not exceed the height of the bench.

Assumption 3.

The safe pit slope angle is the angle between the sides of a mining cut and a horizontal plane at which the material can stand without support for orderly mining of the deposit. It is assumed that these slope angles are known before hand throughout the deposit for all rock types. This slope may vary in the vertical direction as well as horizontally.

Assumption 4.

Let N represent the total number of blocks to be considered for the T time period production scheduling problem. In order to remove a block n which is directly overlain (restricted) by another block in its cone, the overlaying blocks must be removed first in such a manner that the wall slopes of the resulting pit do not exceed the safe pit slopes in any direction. Therefore, if Γ_x represents the set of overlaying restrictive blocks which must be completely mined in order to expose block x, then it is assumed that the elements of Γ_x are completely known for each block n where n = 1, 2, ..., N.

Assumption 5.

All the pertinent economic information such as prices of commodities being mined, costs of mining, milling, smelting and marketing and discount rates are available or can be estimated not only for the existing year but also for the total duration of the production scheduling period. It is important to emphasize here that great care should be given to the collection and estimation of this basic data since the solution of the production scheduling problem is only as good as the data on which it is based.

Assumption 6.

It is assumed that the most critical constraints in addition to the geometric constraints for the long and medium range production schedules may be classified into the following categories;

1. Mining capacity (including ore and waste)
2. Concentrator capacity
3. Refinery capacity
4. Mill feed grade
5. Production required.

Although these categories represent a somewhat simplistic classification, they represent the categories of most concern in present practice (Mathieson, 1982).

It is further assumed that the upper and lower bounds for these constraints are known and can be expressed as linear relationships of the material being mined.

Assumption 7.

Conditions other than the geometric factors may influence the orderly mining of a pit. These conditions may arise due to operating procedures such as the location of haul roads. It is assumed that these influences play a rather important role in short term scheduling and not in long term scheduling. Therefore they are considered to be of only minor importance in determining long and medium range production scheduling and will not be considered in the model developed in this study.

Assumption 8.

It is assumed that the economic effects of factors such as depreciation and taxes on production scheduling requires a separate research study and therefore are not taken into account in this model.

3.3 DEFINITION OF VARIABLES AND TERMS OF THE MODEL

BLOCK

$$\boxed{n}$$

X_n^{mt} Variable

let X_n^{mt} indicates fraction of block n to be used as material type m in time period t.

$$m = \begin{cases} 1 \text{ for ore} \\ 2 \text{ for waste} \end{cases}$$

$X_n^{mt} = 1$ if block n is mined as material type m in time period t.

$X_n^{mt} = 0$ otherwise.

3.3.1 Objective Function

Since the objective of the model will be to maximize the NPV of revenues, decisions will have to consider the value of the block for each time period of the scheduling duration with respect to its material type.

let C_n^{mt} be the coefficient in the objective function for variables representing block n for material type m in time period t.

C_n^{1t} then is the dollar value per unit volume of the block n when mined and processed in time period t as ore.

C_n^{2t} is the stripping cost per unit volume of the block n when it is mined as waste in time period t.

$C_n^{1(t+1)} = C_n^{1t}/(1+d)$ the coefficient for successive time periods is discounted by the interest rate (d) acceptable to the firm.

$C_n^{2(t+1)} = C_n^{2t}/(1+d)$ The cost of stripping will also be discounted by the interest rate d from one time period to another

With these definitions the terms of the objective function will be:

$$\text{Maximize } Z = \sum_t \sum_m \sum_n C_n^{mt} X_n^{mt}$$

where: $n = 1, \ldots N$, N is total number of blocks in the deposit

$m = 1, \ldots M$, M is 2 since the material type is either ore or waste.

$t = 1, 2 \ldots T$ where T is total number of periods.

3.4.2. Constraints of the Model

Mining Capacity:

let $\underline{W}^t$ and $\overline{W}^t$ be lower and upper bounds for stripping capacity. Also let a_n be the volume of the material in block n. The stripping capacity constraints for time period t will then be as follows:

The lower bound

$$\sum_n a_n X_n^{2t} > \underline{W}^t$$

and the upper bound

$$\sum_n a_n X_n^{2t} < \overline{W}^t$$

Concentrator capacity:

let $\underline{0}^t$, $\overline{0}^t$ be the lower and upper bounds of the concentrator capacity requirements respectively. Also let a_n be the volume of material in block n.

The lower bound constraint for concentrate capacity in time period t is:

$$\sum_n a_n X_n^{1t} > \underline{0}^t$$

The upper bound constraint for concentrate capacity in time period t is:

$$\sum_n a_n X_n^{1t} < \overline{0}^t$$

For the feed grade requirements, let g_n be the grade assigned to block n. Also let $\underline{G}_n$, and $\overline{G}_n$ be the lower and upper values for the concentrator mill feed grade requirements repectively, then the lower limit constraint for the mill feed grade in time period t will be

$$\sum_n g_n X_n^{1t} > \underline{G}^t \cdot \sum_n X_n^{1t}$$

or,

$$\sum_n (g_n - \underline{G}^t)\ X_n^{1t} > 0$$

and the upper bound will be

$$\sum_n g_n\ X_n^{1t} < \overline{G}^t \cdot \sum_n X_n^{1t}$$

or

$$\sum_n (g_n - \overline{G}^t)\ X_n^{1t} < 0$$

3.4.3 Available Block Volume Constraints

Since a given block can only be mined once, either as ore or waste, the following constraint must be included to enforce this.

$$\sum_{m=1}^{2} \sum_{t=1}^{T} X_n^{mt} < 1 \qquad \forall\ n$$

or

$$X_n^{11} + X_n^{12} + \ldots + X_n^{1T} + X_n^{21} + X_n^{22} + \ldots + X_n^{2T} < 1$$

3.3.5. Sequencing Constraints

In order to mine a block n, in a given time period t, all the restricting blocks must be mined in or before time period t. The set of constraints to ensure these geometric limitations take the form shown on the following page for the hypothetical four block case given in Figure 13.

These constraints indicate that for a uniform 45 degree slope there might be as many as nine constraints per block per time period. If one is looking at a ten time period problem there would be 90 constraints per block. Hence, for a reasonable size deposit (100,000 block deposit) there will be about 900,000 constraints just for sequencing.

The feasible region formed by the sequencing constraints for a given block n will be represented by a set Γ_n, and from now on, the sequencing constraints of block n will be referred to by $X_k^{mt} \varepsilon \Gamma_n$. Therefore, this abstraction $X_k \varepsilon \Gamma_n$ will mean the following: If the block n is mined, all the blocks in its cone must also be mined. This representation will save time and space in describing the tedious sequencing constraints.

3.4 FORMULATION OF TWO TIME PERIOD PROBLEM

To illustrate the formulation of the production scheduling problem, assume that an open pit deposit is made

$$
\begin{array}{cccccccccccccc}
-X_1^{m1} & & & & +X_4^{m1} & & & & & & & & & \leqslant 0 \\
 & -X_2^{m1} & & +X_4^{m1} & & & & & & & & & & \leqslant 0 \\
 & & -X_3^{m1} & X_4^{m1} & & & & & & & & & & \leqslant 0 \\
-X_1^{m1} & & & +X_4^{m1} & -X_1^{m2} & & & +X_4^{m2} & & & & & & \leqslant 0 \\
 & -X_2^{m1} & & +X_4^{m1} & & -X_2^{m2} & & +X_4^{m2} & & & & & & \leqslant 0 \\
 & & -X_3^{m1} & +X_4^{m1} & & & -X_3^{m2} & +X_4^{m2} & & & & & & \leqslant 0 \\
 & & & & & & & & & & & & & \\
-X_1^{m1} & & & +X_4^{m4} & -X_1^{m2} & & & +X_4^{m2} & +\ldots. & -X_1^{mn} & & & +X_4^{mn} & \leqslant 0 \\
 & -X_2^{m1} & & +X_4^{m1} & & -X_2^{m2} & & +X_4^{m2} & +\ldots. & & -X_2^{mn} & & +X_4^{mn} & \leqslant 0 \\
 & & -X_3^{m1} & +X_4^{m1} & & & -X_3^{m2} & X_4^{m2} & +\ldots. & & & -X_3^{mn} & +X_4^{mn} & \leqslant 0
\end{array}
$$

$$X_n^{mt} \geqslant 0 \quad \forall n,\ m,\ t$$

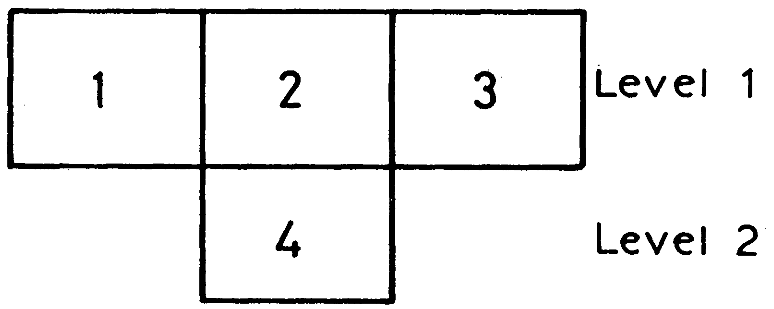


Figure 13. Cross-Sectional Representation of the Four Block, Two Layer Case

up of four blocks and scheduling is to be carried out for two time periods and only upper bounds exist as various constraints. (See Figure 13).

With these assumptions the problem would be formulated as follows;

$$\text{Maximize} \sum_{n=1}^{4} \sum_{m=1}^{2} C_n^{m1} X_n^{m1} + \sum_{n=1}^{4} \sum_{m=1}^{2} C_n^{m2} X_n^{m2}$$

subject to

$$\sum_{n=1}^{4} a_n X_n^{11} < \overline{0}^{\,1}$$

$$\sum_{n=1}^{4} a_n X_n^{21} < \overline{W}^{\,1}$$

$$\sum_{n=1}^{4} (g_n - \overline{G}^1)\ aX_n^{11} < 0$$

$$\sum_{n=1}^{4} a_n X_n^{12} < \overline{0}^2$$

$$\sum_{n=1}^{d} a_n X_n^{22} < \overline{W}^2$$

$$\sum_{n=1}^{4} (g_n - \overline{G}^2)\ X_n^{11} < 0$$

$$x_1^{11} + x_1^{21} + x_1^{12} + x_1^{22} < 1$$

$$x_2^{11} + x_2^{21} + x_2^{12} + x_2^{22} < 1$$

$$x_3^{11} + x_3^{21} + x_3^{12} + x_3^{22} < 1$$

$$x_4^{11} + x_4^{21} + x_4^{12} + x_4^{22} < 1$$

CHAPTER 4

STRUCTURE OF THE PRODUCTION SCHEDULING MODEL

4.1 INTRODUCTION

In order to solve large scale mathematical programs, similar to the one formulated in Chapter 3, one should make efficient use of the special structure of the problem. Therefore it is appropriate here to discuss the general structure of the production scheduling problem before discussing the solution algorithm given in the following chapters.

The compact representation of the problem formulated in the previous chapter can be given in terms of matrix notation. This representation is as follows:

$$\text{Maximize } C^1X^1 + C^2X^2+\ldots C^nX^n$$

Subject to

$$\begin{array}{lcccc} & AX^1 & & & < b^1 \\ \text{Capacity-Blending} & & AX^2 & & < b^2 \\ \text{Constraints} & & & & \\ & & & AX^t & < b^t \\ & & & & \\ & EX^1 & & & < 0 \end{array}$$

Sequencing Constraints

$$EX^1 + EX^2 < 0$$

$$\vdots$$

$$EX^1 + EX^2 + \quad + EX^t < 0$$

Reserve Constraint

$$IX_n^1 + IX_n^2 + \quad + IX_n^t < 1 \quad \forall \, n$$

$$X_n^{mt} > 0 \quad \forall n, m, t$$

where:

C^t = 1 x (n.m) vector of objective function cost coefficients

X^t = (n.m) x 1 vector of variables

${X_n}^{mt}$ = (m.t) x 1 vector of variables

A^t = (K x (m.n) matrix of capacity and blending constraint coefficents for time period t.

b^t = (K x 1) vector of right hand side coefficients in time period t.

E = w x (n.m) matrix of (0, 1 or -1) coeffients for the sequencing constraints

n = block number

m = ore or waste indicator superscript

K = number of capacity and blending constraints for a given time period

w = number of overlaying restrictive blocks in any Γ_n.

This formulation of the production scheduling problem contains an important mathematical structure. As indicated in the formulation, there are two distinct sets of constraints. The first set deals with the capacity restrictions and grade limitations in each time period. This set will be referred to as capacity blending constraints. The second set of constraints are for making sure that the geometric and sequence limitations imposed by the pit slopes are not violated. These will be referred to as sequencing constraints.

The characteristics of the sequencing constraints are that all the coefficients in the coefficient matrix E are either 0, 1, or -1. A matrix A is called unimodular if the determinant of every square submatrix of A equals 0, +1 or -1. It has been proven that if a problem with a coefficient matrix which is unimodular is solved, the solution set will be necessarily 0 or 1 if the right hand sides are also 0 or 1 (Johnson 1968). This feature of sequencing constraints gives the sequencing problem a network structure and this plays an important role in obtaining integer solutions in the solution algorithm developed in Chapter 6.

4.2 STRUCTURE OF SINGLE TIME PERIOD PROBLEM

In order to explore and to further develop an understanding of the production scheduling problem as formulated in previous sections, let us look at the single time period production scheduling model.

Assuming that the objective is to maximize profits in mining the deposit in a single time period this model can be formulated as follows:

$$\text{Maximize } CX$$

Subject to:

$$\begin{array}{lll}
AX & < & b \\
EX & < & 0 \\
\sum_{m} X_n^m & < & 1 \quad \forall\ n \\
X_n^m & > & 0 \quad \forall\ m \text{ and } n
\end{array}$$

If the capacity blending constraints of the single time period problem in the above formulation are removed one obtains the following problem:

Maximize CX

$$\begin{array}{llccl} \text{Subject to:} & EX & \leq & 0 & \\ \text{Sequencing Constraints} & \sum_{m} X_n^m & \leq & 1 & \forall\ n \\ & X_n^m & \geq & 0 & \forall\ n,\ m \end{array}$$

This reduced problem may be stated so as to maximize the revenues (based on block current time period block values) subject to slope constraints.

Hence this problem is the formulation of the ultimate pit limit problem on which a number of papers have been published. As a result, the ultimate pit limit problem is a special case of the general single time period production scheduling problem.

If the feasible set of solutions for the sequencing constraints is represented as Γ the single time period problems can be written as

$$\begin{aligned} \text{Max} \quad & CX \\ & AX \leq b \\ & X \in \Gamma \end{aligned}$$

Where Γ indicates the set of blocks which are feasible for the slope constraints and includes the requirements of $\sum_{m} X_n^m \leq 1$ and $X_n^m \geq 0$.

In this form of formulation, constraints of type $AX \leq b$ will be referred to as side constraints. Constraints of $X \in \Gamma$ will be refered to as the constraint set with special structure. In coming up with a solution algorithm to the production scheduling problem, the side constraints will be relaxed by multiplying them by lagrange multipliers and putting them into the objective function. This method, the lagrangian relaxation method (Murty, 1979), will be the basis of the solution algorithm developed in this thesis and will be introduced in the next chapter.

CHAPTER 5

BASIC CONCEPTS OF LAGRANGE MULTIPLIERS

5.1 INTRODUCTION

The concept of lagrange multipliers as presented in the mathematical optimization theory will be a very important part of the solution technique developed in this dissertation. In the area of mathematics, the theoritical developments and subsequent use of the lagrange multiplier approach for solving optimization problems goes back to the early 1960's (Everett, 1963). Since then, a class of integer problems with a special structure have been successfully solved by using this concept (Geofrian, 1974).

In the area of mining, use of the lagrange multiplier concept also goes back to late 1960. Although it was not realized as such, the concept of cutoff grades, profcosts and the concept of parameterization are all simplistic applications of the lagrange multiplier theory. In this chapter, the concept of lagrange multipliers will be introduced. Specifically, in order to make the reader more comfortable with the concept of Lagrange multipliers, the concept of classical constraint optimization will be reviewed and the similarities between cutoff grades and Lagrange multipliers will be developed.

5.2 SINGLE CONSTRAINT PROBLEM AND ITS LAGRANGIAN

Assume a single layer deposit of six blocks with an assigned grade g_i for each block i (i=1, to 6) (Figure 14). If the problem is to find n blocks such that n is greater than or equal to $\underline{0}$ and less than or equal to $\overline{0}$ among the existing six blocks which yields the maximum quantity of metal, then the formulation of this problem in terms of the terminology discussed in Chapter 3 is as follows.

$$\text{Max } Q = g_1\ X_1{}^1 + g_2\ X_2{}^1 + g_3\ X_3{}^1 + g_4\ X_4{}^1 + g_5\ X_5{}^1 + g_6\ X_6{}^1$$

Subject to:

$$\underline{0} < X_1{}^1 + X_2{}^1 + X_3{}^1 + X_4{}^1 + X_5{}^1 + X_6{}^1 < \overline{0} \qquad \text{(a)}$$

$$X_n^1 = 1 \text{ if the block is mined} \qquad \text{(b) (5.1)}$$

$$X_n^1 = 0 \text{ otherwise} \qquad \text{(c)}$$

If the multiplier u_i is assigned to the constraint (5.1) (a). The lagrangian of this problem can be written (Murty 1979) as:

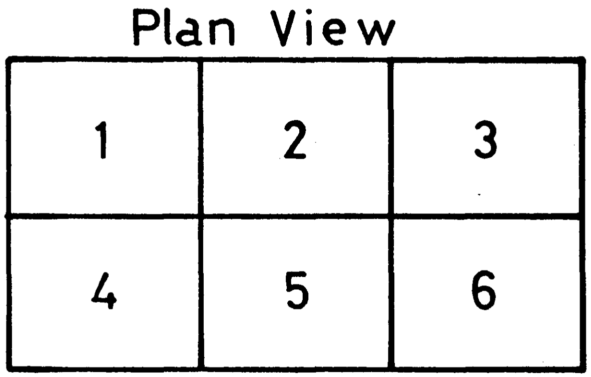


Figure 14. Plan View Representation of 6 Block Single Layer Mine

$$\text{Max } L(u,x) = (g_1 - u_1)\, X_1{}^1 + (g_2 - u_1)\, X_2{}^1 + (g_5 - u_1)\, X_3{}^1 + (g_4 - u_1)\, X_4{}^1 + (g_5 - u_1)\, X_5{}^1 + (g_6 - u_1)\, X_6{}^1$$

Subject to:

$$X_n{}^1 \text{ is } 0 \text{ or } 1 \qquad (5.2)$$

$$u_i > 0$$

In the classical optimization theory, in order to solve a constrained problem the general procedure is to transform the problem into a lagrangian form and then take the derivative of the lagrangian with respect to the variables. Here we will deviate from this classical approach and look at the problem from another angle. This approach will be as follows.

Set $u_1 = 0$ and solve the langragian in problem 2. Setting $u_i = 0$ gives lagrangian to be

$$\text{Max } L\,(0,x) = g_1\, X_1{}^1 + g_2\, X_2{}^1 + g_3\, X_3{}^1 + g_4\, X_4{}^1 + g_5\, X_5{}^1 + g_6\, X_6{}^1$$

$$X_n{}^1 \text{ is } 0 \text{ or } 1 \qquad (5.3)$$

Clearly the function in problem (5.3) is maximized by setting $X_n{}^1$ equal to 1 if $g_i > 0$ and setting $X_n{}^1 = 0$ otherwise.

Notice that setting the langrangian $u_1 = 0$ has the same effect as relaxing constraint (a) of problem (5.1).

Now that it has been determined which variables are going to be $X_n^1 = 1$ or $X_n^1 = 0$ one needs to check if the solution obtained satisfies constraint (a) of problem (5.1).

If the number of blocks mined $(\sum_n X_n^1)$ is within the specified bounds, $(\underline{0} < \sum_n X_n^1 < \overline{0})$, the solution is feasible and the solution is optimal for a problem with an upper bound constraint equal to $\sum X_n^1$ (Everett, 1963). If the number of blocks mined is larger than the allowable number $(\sum_n X_n^1 > \overline{0})$ then: set $u_1 = c_1 > 0$ and maximize the lagrangian in problem (5.2), which is;

$$\text{Max } L(u,x) = (g_1 - c_1)X^1{}_1 + (g_2 - c_1)X_2{}^1 + (g_3 - c_1)X_3{}^1$$
$$+ (g_4 - c_1)X_4{}^1 + (g_5 - c_1)X_5{}^1 + (g_6 - c_1)X_6{}^1$$

Subtracting some value c_1 greater than zero from the block grades, g_i, will reduce the grade coefficients in the objective function and depending on the value of block grades and the value of the multiplier c_1, the coefficient $(g_i - c_1)$ will become negative or zero for some of the blocks when $g_i < c_i$.

Clearly the lagrangian in problem (5.2) is maximized by setting $X_n^1 = 1$, if the objective function coefficient for the lagrangian is $(g_i - c_1) > 0$ and $X_n{}^1 = 0$ for $(g_i - c_1) < 0$.

After this step, the constraint must be checked again. Since some of the blocks previously mined may not be mined in this iteration, the number of block mined $(\sum_n X_n^1)$ might be within the limitation imposed by the total concentrator tonnage requirements; if it is we stop. Otherwise one increases the value of the lagrange multiplier and the process is repeated. In some cases the amount of material mined will be less than the lower bound $(\sum_n X_n < \underline{0})$. When this happens, one needs to lower the value of the multiplier u_1 and iterate again. By doing this some of the blocks which are penalized more than they should have been and not mined in the previous iteration get a chance to be mined again in this iteration. This procedure continues until the total number of blocks mined are within the capacity limitation and the optimality conditions given in theorem 6.2 are satisfied. (It should be realized that the inherent assumption in the above discussion is of course that all blocks contain ore material).

It has been known in the mining industry for a long time that in order to come up with nested incremental pits, one can increase or decrease the value of the cutoff grade (Halls-Bellum 1969). As was demonstrated in the preceding paragraphs, a single static cutoff grade behaves like the lagrange multiplier of the one constraint problem in mathematical terms.

5.3 LAGRANGIAN FOR TWO CONSTRAINT PROBLEM

In the previous analysis, only a single constraint was considered. This was the constraint on the total ore tons to be processed by the concentrator. This single constraint case is very simplistic and does not represent any actual mining situation. In a realistic mining environment there are many other constraints which limit the system as well. For example, there are constraints on the total tons to be mined in terms of ore and waste. Furthermore, mining of a given block is also constrained by the mining of overlaying blocks, etc. Therefore, to make the mining model a little more realistic, a second constraint limiting the total tons moved from the mine will be introduced. (Assume also that only upper bounds exist on these constraints). At the same time the block configuration will be changed such that the six blocks are located on a cross section as shown in Figure 15. Moving from the single level model of figure 14 to the two level problem of Figure 15 necessitates the inclusion of

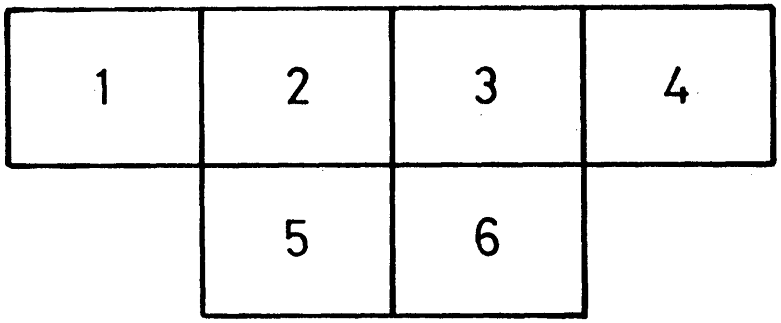


Figure 15. Cross-Sectional Representation of Six Block, Two Layer Case

sequencing constraints in the model, hence the formulation of the problem becomes:

$$\text{Max } Q = g_1 X_1^1 + g_2 X_2^1 + g_3 X_3^1 + g_4^1 + g_5 X_5^1 + g_6 X_6^1$$

$$\text{s.t.: } X_1^1 + X_2^1 + X_3^1 + X_4^1 + X_5^1 + X_6^1 < \bar{O} \qquad (5.4)$$

$$X_1^1 + X_2^1 + X_3^1 + X_4^1 + X_5^1 + X_6^1 + X_1^2 + X_2^2 + X_3^2 + X_4^2 + X_5^2 + X_6^2 < \bar{T}$$

Sequencing constraints:

$$(X_5^1+X_5^2) - (X_1^1+X_1^2) < 0 \qquad (X_6^1+X_6^2) - (X_2^1+X_2^2) < 0$$

$$(X_5^1+X_5^2) - (X_2^1+X_2^2) < 0 \qquad (X_6^1+X_6^2) - (X_3^1+X_3^2) < 0$$

$$(X_5^1+X_5^2) - (X_3^1+X_3^2) < 0 \qquad (X_6^1+X_6^2) - (X_4^1+X_4^2) < 0$$

$X_n^1 = 1$ if the block n is mined as ore

$X_n^1 = 0$ otherwise

$X_n^2 = 1$ if the block n is mined as waste

$X_n^2 = 0$ otherwise

If the set of feasible pits satisfying the sequencing constraints are represented by Γ_n the formulation of the two constraint problem becomes

$$\text{Max } Q = g_1\ X_1{}^1 + g_2\ X_2{}^1 + g_3\ X_2{}^1 + g_4\ X_4{}^1 + g_5\ X_5{}^1 + g_6\ X_6{}^1$$

Subject to:

$$X_1{}^1 + X_2{}^1 + X_3{}^1 + X_4{}^1 + X_5{}^1 + X_6{}^1 \qquad < \quad \overline{0}$$

$$\begin{aligned}&X_1{}^1 + X_2{}^1 + X_3{}^1 + X_4{}^1 + X_5{}^1 + X_6{}^1 + X_1{}^2\\ &X_2{}^2 + X_3{}^2 + X_4{}^2 + X_5{}^2 + X_6{}^2 \qquad < \overline{T}\end{aligned}$$

$$X_n{}^1,\ X_n{}^2\ \varepsilon\ \Gamma_n \text{ set of feasible pits} \qquad (5.5)$$

$$X_n{}^1,\ X_n{}^2 \text{ are } 0 \text{ or } 1 \text{ and } X_n{}^1 + X_n{}^2 < 1$$

When multiplier $u_1 > 0$, and $u_2 > 0$ are assigned to the concentrator capacity and to the total mining capacity respectively, the lagrangian of problem (5.5) becomes:

$$\begin{aligned}\text{Max } L(u,x) = {}& (g_1 - u_1 - u_2)X_1{}^1 + (g_2 - u_1 - u_2)X_2{}^1 - (g_2 - u_1 - u_2)X_3{}^1\\ & (g_4 - u_1 - u_2)\ X_4{}^1 + \ (g_5 - u_1 - u_2)\ X_5{}^1 + (g_6 - u_1 - u_2)\ X_6{}^1\\ & - u_2\ X_1{}^2 - u_2\ X_2{}^2 - u_2\ X_3{}^2 - u_2\ X_4{}^2 - u_2\ X_5{}^2 - u_2\ X_6{}^2\end{aligned}$$

Subject to:

$X_n^1, X_n^2 \in \Gamma_n$ set of feasible pits (5.6)

X_n^1, X_n^2 are 0 or 1 and $X_n^1 + X_n^2 \leq 1. \quad \forall n.$

5.3.1 Solution To The Two Constraint Problem

As can be seen, the lagrangian of two constraint problems has a modified objective function subject to the sequencing constraints. This modification is the result of penalties assigned by the multipliers. This problem is nothing more than an ultimate pit limit problem with a modified objective function. Therefore, in solving a six block, two layer, two constraint production scheduling problem, one can start again by assigning 0 values to the ($u_1 = 0$, $u_2 = 0$) multipliers. Now problem (5.5) becomes

$$\text{Max } Q = g_1 X_1^1 + g_2 X_2^1 + g_3 X_3^1 + g_4 X_4^1 + g_5 X_5^1 + g_6 X_6^1$$

Subject to:

$X_n^1, X_n^2 \in \Gamma$ set of feasible pits (5.7)

X_n^1, X_n^2 are 0 or 1 $X_n^1 + X_n^2 \leq 1.$

To solve problem (5.5) by the lagrange multiplier approach problem (5.7) is solved first to determine which blocks to mine ($X_n^1 = 1$). A given block is mined if $g_i > 0$ (the grade assigned to a block) is positive. Of course mining a block with a positive $g_i > 0$ will require mining some waste blocks that fall into the cone of a positive block. Once it is determined what blocks should be mined, a check must be made to see if the mining capacity constraints and the concentrator capacity limitations are violated. If the number of ore blocks mined($\sum_n X_n^1$) exceeds the capacity ($\sum_n X_n^1 > \overline{O}$), and/or the number of total blocks mined exceeds the total mining capacity limitations ($\sum_n X_n^1 + \sum_n X_n^2 > \overline{T}$) then one or both of the lagrange multipliers will be increased by a certain amount, ($u_1 = c_1$, $u_2 = c_2$) and problem 5.7 solved again. By repeated modification of the multipliers the number of blocks to be mined can be brought into the feasible region. This feasible solution is optimal for a problem where the capacity limitations are equal to the number of blocks mined (Everett, 1963).

There may be cases however that no matter how the multipliers are modified, one will not be able to obtain a feasible solution or optimal to the problem. This condition is referred to as the "condition of gaps" and will be discussed in more detail in Chapter 7.

The solution algorithm developed in Chapter 6 in terms of lagrangian optimization theory will simply follow a similar method of solution in solving production scheduling problems. The side constraints corresponding to each time period of the multi time period production scheduling problem (i.e., blending and capacity constraints) will be relaxed by multiplying each of them by a lagrange multiplier and including them in the objective function. The lagrangian problem will then be solved by the algorithm developed for the multi time period sequencing problem in Chapter 7. Each time, the results will be compared with constraint requirements. Before going into the theoretical development of the solution algorithm, let us look at the relationship between the lagrange multipliers and the traditional mining concepts such as cutoff grades.

5.4. CUTOFF GRADES AND THEIR ASSOCIATION WITH LAGRANGE MULTIPLIERS

In mining, generally one speaks of two types of cutoff grades. One cutoff grade is to decide if a given block should be mined or not and treated as ore; the other is to decide what to do with the block if it is mined and originally not designated as ore by the first cut off grade, i.e., for ore, and waste separation after mining. These two types of cutoff grades are known as "in situ" and "in pit" or "milling" cutoff grades and they correspond to the lagrange multipliers of the two constraint problem discussed in the previous section. This similarity will be formalized here.

In solving problem 5.5 one not only needs to decide what blocks to mine but also what blocks will be mined as ore and as waste. The decision whether to mine a block as ore or waste in general is taken as a result of the following comparison:

Case 1:

If the objective function coefficient of X_n^1 in problem (5.5) is

$$(g_n - u_1 - u_2) > 0 \qquad (5.8)$$

then one should set $X_n^1 = 1$ (i.e., the block n should be mined as ore) to maximize the objective function in problem (5.5).

Case 2:

If the objective function coefficient of X_n^1 in problem (5.5) is

$$(g_n - u_1 - u_2) < 0 \qquad (5.9)$$

and the block n must be mined because of the slope constraints there are two alternatives one has to consider with respect to this block. If

$$(g_n - u_1 - u_2) > - u_2 \Rightarrow g_n - u_1 > 0 \qquad (5.10)$$

then setting $X_n^1 = 1$ (i.e., block n should be treated as ore) will maximize the objective function in problem (5.5). However, if

$$(g_n - u_1 - u_2) < -u_2 \Rightarrow g_n - u_1 < 0$$

then one should set $X^1 = 0$ and $X_n^2 = 1$ (i.e., block n should be mined and taken to the waste dump. Making decisions as indicated above will determine the right choice in maximizing the objective function of problem (5.6).

It should be apparent that the lagrange multipliers (u_1 and u_2) assigned to the constraint set of problem (5.6) affects the decision making process and defines a choice with respect to alternative actions concerning what to do about a given block. In essence the combined value for (u_1 + u_2) is the in situ cutoff grade and u_1 by itself is the "after mining" cutoff grade. To further demonstrate that these lagrange multipliers are the two cutoff grades used in the traditional mine planning let us look at these cutoff grades from the mining point of view. The value function for a given block is;

$$V_n = (s - r) \cdot y \cdot g_n - c - m$$

where:

V_n = value of block n (\$/ton)

g_n = grade of block n (lbs/ton)

s = selling price (\$/lb of metal)

r = refining cost (\$/lb)

c = concentrator cost (\$/ton)

m = mining cost (\$/ton)

y = concentrator and smelter recovery.

Normally if a given block value V_n

$$(s - r).y.g_n - c - m > 0 \tag{5.11}$$

then this block is mined as ore. However, if the block value V_n is negative ($V_n < 0$) and the block n must be mined because of the slope constraints then the block is normally considered as ore if

$$(s - r).y. g_n - c - m > -m \tag{5.12}$$

and the block is considered as waste if

$$(s - r).y. g_n - c - m < -m. \tag{5.13}$$

If both sides of inequalities (5.11), (5.12), and (5.13) are divided by [(s - r).y] and c/[(s - r).y], and m/[(s - r).y] are substituted for u_1 and u_2 respectively then inequality (5.11) becomes;

$$g_n - u_1 - u_2 > 0$$

and inequality (5.12) becomes

$$g_n - u_1 - u_2 > - u_2$$

and inequality (5.13) becomes

$$g_n - u_1 - u_2 < - u_2$$

which are the same set of conditions one checks in deciding mine, not mine, process, or not to process a block when lagrange multipliers u_1 and u_2 are assigned to the mining and milling constraints of problem 5.6.

5.5 FRENCH PIT PARAMETRIZATION AND LAGRANGE MULTIPLIERS

Francois-Bongarcon (1978) developed an algorithm to determine the number of nested pits for the ultimate pit limit analysis. This algorithm is known as the French pit parameterization method.

Although there are a number of papers published on the subject, it is still unclear to many mining engineers what the French parametrization concept really is.

The concept of French parametrization can be viewed as a single time period production scheduling problem using a two constraint model as discussed in section (5.3). In solving the two constraint problem every time the multipliers (u_1 and u_2) are adjusted, a different pit is obtained which includes a fixed ore tonnage and a fixed total (ore and waste) tonnage. This is exactly what the French pit parametrization algorithm does and the parameters λ and θ they use (Marechal and Francois-Bongarcon 1976) are identical to the Lagrange multipliers u_1 and u_2. Although Matheron (1976) and Francois-Bongarcon (1978) never realized this was the case in parametrization of the reserves, looking at their algorithm from this perspective makes their algorithm more powerful than it was originally envisioned. Taking the original French approach a few steps further and considering the parametrization concepts within the framework of the lagrangian theory not only provides an efficient model to explain the concept of cutoff grades but also gives the necessary intuition in understanding the production scheduling algorithm developed in this thesis.

5.6 THE ASSOCIATION OF PROFCOSTS TO LAGRANGE MULTIPLIERS

In sections 5.2 and 5.3, the objective of the model for a single time period and single and two constraint cases was the maximization of the quantity of the metal recovered subject to a tonnage constraint. In general this objective funtion is not valid. No realistic analysis can be carried out without considering various economic factors.

The commonly accepted objective function which incorporates various economic factors such as revenues and costs is maximization of profits or cash flows.

If the objective is to maximize profits from the pit subject to some concentrator and stripping capacity restrictions and if C_n^1 is the dollar value of the block as ore, and C_n^2 is the stripping cost per block, the formulation of the model for the six block, single layer case as shown on Figure 14, is as follows:

$$\text{Max } Z = C_1^1 X_1^1 + C_2^1 X_2^1 + C_3^1 X_3^1 + C_4^1 X_4^1 + C_5^1 X_5^1 + C_6^1 X_6^1 + C_1^2 X_1^2 + C_2^2 X_2^2 + C_3^2 X_3^2 C_4^2 X_4^2 + C_5^2 X_5^2 + C_6^2 X_6^2 \tag{5.13}$$

Subject to

$$X_1^1 + X_2^1 + X_3^1 + X_4^1 + X_5^1 + X_6^1 \quad \leq \overline{O}$$

$$X_1^1 + X_2^1 + X_3^1 + X_4^1 + X_5^1 + X_6^1 + X_1^2 + X_2^2 + X_3^2 + X_4^2 + X_5^2 + X_6^2 \quad \leq \overline{T}$$

X_n^1 X_n^2 ε Γ_n where Γ_n is the set feasible pits and X_n^2 X_n^1 are 0 or 1.

if one assigns muliplier u_1 to concentrator capacity constraint and multiplier u^2 to the stripping capacity constraint then the lagrangian of this problem 1 is:

$$\begin{aligned}\text{Max } L(u,x) = {} & (C_1{}^1 - u_1)X_1{}^1 + (C_2{}^1 - u_1)X_2{}^1 \ (C_3{}^1 - u_1)X_3{}^1 + \\ & (C_5{}^1 - u_1)X_5{}^1 + (C_6{}^1 - u_1)X_6{}^1 + (c_1{}^2 - u_2)X_1{}^2 + \\ & (C_3{}^2 - u_2)\ X_2{}^2 + (C_3{}^2 - u_2)\ X_3{}^3 + \\ & (C_4{}^2 - u_2)\ X_4{}^2 + (C_5{}^2 - u_2)\ X_5{}^2 + \\ & (C_6{}^2 - u_2)\ X_6{}^2\end{aligned} \tag{5.14}$$

Subject to

X_n^1, X_n^2 ε Γ_n where Γ_n is the set of feasible pits

and X_n^1 , and X_n^2 are 0 or 1.

The solution to problem (5.13) can be obtained again by solving the lagrangian problem (problem 5.14) by modifying the multiplier u_1 and u_2 until the ore tonnage constraint is satisfied. It is clearly seen that trying to solve the single constraint profit maximization problem by modifying the lagrangian multiplier is identical to coming up with successive nested pits by either adjusting the mining costs or the price as discussed in section 2.4 which discusses heuristic optimization techniques.

5.7 ASSOCIATION OF THE LAGRANGE MULTIPLIERS TO CUTOFF GRADES UNDER AN ECONOMIC OBJECTIVE

When the objective function being maximized represent the quantity of metal as in section 5.2 and 5.3, the lagrange multipliers associated to capacity constraints can be defined as the cutoff grades. This relationship was shown in section 5.6. When the objective is maximization of net present value as in section 5.6, the multipliers associated to ore tonnage and total tonnage constraints no longer represent cutoff grades. According to linear programming theory, (see Dantzig, 1960), they are defined as the marginal values of the constraints. For example, u_1 is defined as the marginal value of the concentration capacity and u_2 as the marginal value of the total mining capacity constraint.

Although defined as the marginal values of the constraints, under an economic objective function, the lagrange multipliers are still related to the original definition of cutoff grades of section 5.4. This relationship will be shown next.

When the objective function is maximization of revenues, a block of material should be mined as ore if

$$C_n^1 - u_1 - u_2 > 0$$

Since $C_n^1 = (P - r).y.g_1 - c - m$

$$[(P - r).y.g_1 - c - m] - u_1 - u_2 > 0$$

and this defines an "in situ" cut off grade:

$$g_1 = \frac{(c + m) + u_1 + u_2}{(P - r).y} \tag{5.14}$$

Furthermore, if a block of marginal material is mined in order to uncover a block of ore, this block should also be considered as ore if:

$$C_n^1 - u_1 - u_2 > C_n^2 - u_2$$

where $C_n^1 = (P - r).y.g - c - m$, and C_n2 is the stripping cost ($C_n^2 = m$). Hence

$$[(P - r).y.g - c - m] - u_1 - u_2 > m - u_2$$

and this defines an "in pit" mill cutoff grade as:

$$g_2 = \frac{C + u_1}{(P - r).y} \tag{5.15}$$

5.7.1 The Fundamental Difference

Traditionally, when the mining system is considered to be unconstrainted, the "in situ" cutoff grade is defined as

$$g_1{}^* = \frac{c + m}{(P - r).y} \tag{5.16}$$

and "in pit" mill cutoff grade is defined as

$$g_2{}^* = \frac{c}{(P - r).y} \tag{5.17}$$

It can be seen that the fundamental difference between the cutoff grades of the constrained case and the cutoff grades of the unconstrained case is the terms for the multipliers u_1 and u_2 in the nominator of equation (5.14) and equation (5.15).

It is a known fact in linear programming that if the system capacities have slacks, that is when there is more capacity than needed, the multipliers have to be equal to zero in the optimal solution in order to satisfy the optimality conditions. Thus when there is more capacity then needed, the system becomes as if it is unconstrained and multipliers u_1 and u_2 goes to zero and equation (5.14), and (5.16) become identical.

The same discussion can be extended to "in situ" milling cutoff grades. The only difference between the two cutoff grade equations, namely (5.15) and (5.17), is the term u_1. This term, which is equal to the marginal value of concentrator capacity, again has to equal zero if the concentrator capacity is greater than what is needed. Therefore, when the problem is assumed to have unlimited capacities, u_1 again goes to zero ($u_1 \rightarrow 0$) and the "inpit" cutoff grade defined for the constrained case becomes identical to the "inpit" cutoff grade of the unconstrained case.

It should be pointed out here that the three cutoff grades Lane (Lane 1964) referred to in his paper for the two constraint case correspond to the cutoff grades of the constrained mining system we have been discussing in this section. For example, Lane's concentrator limited grade (g_c) can be defined as:

$$g_c = \frac{(c) + u_1}{(P - r).y}$$

mine limited cutoff grade (g_m) can be defined as:

$$g_m = \frac{(C + m) + u_2}{(P - r).y}$$

balancing cutoff grade (g_{mc}) can be defined as:

$$g_{mc} = \frac{(C + m) + u_1 + u_2}{(P - r).y}$$

and these cutoff grades can directly be determined when the multipliers are known.

The cutoff grade discussion up to now did not include the multipliers associated with the sequencing constraints. When the mining system is constrained by the sequencing constraints as well as mining and concentrator capacities, then the in situ cutoff grades for a block of material will be:

$$g_1 = \frac{(C + m) + u_1 + u_2 + [u_3 + \dots u_n]}{(P - r).y}$$

The "in pit" mill cutoff grade will be

$$g_2 = \frac{(C) + u_1 + [u_3 + \dots + u_n]}{(P - r).y}$$

where multipliers u_3 ... u_n will correspond to individual sequencing constraints 3 to n.

In summary, this relationship between the Lagrange multipliers and cutoff grades puts the concept of cutoff grades into the right perspective when the total mining system is considered to be a constrained process.

5.8 EXTENSION OF A LAGRANGE MULTIPLIER APPROACH TO PRODUCTION SCHEDULING PROBLEM

As shown in earlier discussions of this chapter, the concepts of lagrange multipliers have been used in the mining industry under different names in solving the single time period production scheduling problem. Since it has been argued that the ultimate objective is not solving the single time period problem, but solving the multi time period production scheduling problem then it seems natural to extend the solution concepts of lagrange multipliers to the multi time period problem. This will be the basic approach to be used in developing a solution algorithm for the multi time period production scheduling problem. In solving the production scheduling problem multipliers will be assigned to each of the capacity-blending constraints for a given time period. And then, a systematic way to modify the multipliers will be presented. This approach is known as the lagrangian relaxation approach in operation research literature and will be the topic of the next chapter.

CHAPTER 6

SOLUTION ALGORITHM OF MULTI TIME PERIOD SCHEDULING

6.1 THE PRODUCTION SCHEDULING PROBLEM AND ITS LAGRANGIAN

In this chapter, the solution algorithm to the multi time period scheduling problem will be given. The abstract form of the production scheduling problem as formulated in chapter 3 can be written as:

$$\begin{array}{lllllll}
\text{Max} & C^1 X^1 & + C^2 X^2 & & + C^T X^T & & \\
 & & & & & & \\
 & AX^1 & & & & < b^1 & \\
 & & + AX^2 & & & < b^2 & \\
 & & & & + AX^T & < b^T & \\
 & & & & & & \\
 & EX^1 & & & & < 0 & \\
 & EX^1 & + EX^2 & & & < 0 & (6.1) \\
 & EX^1 & + EX^2 & & + EX^T & < 0 & \\
 & & & \sum_{m=1}^{2} \sum_{t=1}^{T} X_n & & < 1 & \forall_n \\
 & & & X_n^{mt} & & > 0 & \forall\ n, m, t
\end{array}$$

Problem (6.1) is difficult to solve because of a few constraints of the type $AX^t < b^t$ which are the capacity and blending constraints for different time periods. Otherwise, without these constraints problem (6.1) has a "nice" network structure.

As mentioned earlier in Chapter 4, unless this special structure is used it is quite difficult, if not impossible to solve a production scheduling problem.

In order to take advantage of the network structure of the sequencing constraints, it appears appropriate to use lagrangian techniques (Lasdon, 1970, Geofrian 1974, Shapiro 1979 and Fisher 1981) in the solution procedure for the production scheduling problem.

The Lagrangian technique provides an efficient approach to reduce the original problem from its present form to a less difficult form. The lagrangian of the problem is formed by multiplying the constraints (AX^T-b) by a set of non negative multipliers (u^T) and then subtracting their sum from the objective function.

Let $\overline{u} = (u^1, u^2 \ldots u^T)$ be the set of non negative multiplier vectors for the capacity-blending constraints of the scheduling problem. Then the lagrangian of problem 6.1 is:

$$\text{Max} \quad C^1 X^1 + C^2 X^2 + C^T X^T - u^1 (AX^1 - b^1) - u^2 (AX^2 - b^2) \ldots u^T (AX^t - b^t)$$

subject to:

$$\begin{array}{lrll} EX^1 & & < 0 & \\ EX^1 + EX^2 & & < 0 & (6.2) \\ EX^1 + EX^2 & EX^T & < 0 & \\ & \sum_{m=1}^{2} \sum_{t=1}^{T} X_n^{mt} & < 1 & \\ & X_n^{mt} & > 0 & \forall \; m,t \;\&\; n \end{array}$$

Let:

$\overline{C}$ = 1 x T vector of time period cost coefficients (C^1, C^2, C^3 ... C^T).

$\overline{X}$ = 1 x T vector of variables X^t representing different time periods. $\overline{X} = (X^1, X^2 \ldots X^T)$,

$\overline{b}$ = 1 x T vector of right hand side coefficients in each time period $\overline{b} = (b^1, b^2, \ldots b^T)$

$\overline{u}$ = 1 x T vector of lagrange multipliers assign to capacity and blending constraints of each time period $\overline{u} = (u^1, u^2, \ldots u^T)^t$

Γ = the set of feasible solutions for the sequencing constraints.

Using the previous notation the production scheduling problem of (6.1) can be written in abstract form as follows:

$$Z\ (X) = \max\ \bar{C}\ \bar{X}$$

Subject to:

$$A\bar{X} < \bar{b}$$

$$\bar{X} \in \Gamma \qquad (6.3)$$

the Lagrangian of problem (6.3) is;

$$L\ (\bar{u}) = \underset{u,x}{\text{Max}}\ (C\bar{X} - \bar{u}\ (A\bar{X} - b))$$ b? ?

$$\bar{X} \in \Gamma. \qquad (6.4)$$

6.1.1 Relationships Between The Original Problem and the Lagrangian

There are a number of important relationships between the original production scheduling problem and its lagrangian. The following theorems will show these.

Theorem 6.1. For any $\bar{u} > 0$, if the optimal solution for the lagrangian problem (6.4) is feasible to the original problem (6.3), then this optimal solution to the lagrangian will always be greater than or equal to the optimal solution of the original problem, i.e., $L(\bar{u}) > Z(\bar{X})$.

Proof. Since the solution for the lagrangian problem is also feasible for problem (6.3), the condition $(A\bar{X} < \bar{b})$ holds. Hence $(A\bar{X} - \bar{b}) < 0$. Since $\bar{u} > 0$, the product $\bar{u}\ (A\bar{X} - \bar{b}) < 0$. Therefore, this gives

$$\text{Max } \bar{C}\ \bar{X} < \text{Max } (\bar{C}\ \bar{X} - u\ (A\bar{X} - \bar{b}))$$

and

$$Z(\bar{X}) < L(\bar{u})$$

6.1.2 Optimality Conditions

In theorem 6.1 we proved that for any set of non-negative multipliers, the solution to the lagrangian which is feasible for the original problem gives an upper bound on the solution to the original problem. A natural question to ask is: "When is a zero-one solution $\bar{X}$ obtained from the lagrangian optimal for the original production scheduling problem?"

Theorem 6.2. For a given $\overline{u}^* > 0$ if a vector $\overline{X}^*$ satisfies the three conditions

(i) $\overline{X}^*$ is optimal in lagrangian $L(\overline{u})$

(ii) $A\overline{X}^* < \overline{b}$ $b \leq \widehat{b}$

(iii) $\overline{u}^* (A\overline{X}^* - b) = 0$

then $\overline{X}^*$ is an optimal solution to the original production scheduling problem (6.3).

Proof. The solution $\overline{X}^*$ is clearly feasible in problem (6.3) since $\overline{X}^* \varepsilon \Gamma$ and $A\overline{X}^* < \overline{b}$ by condition (ii). By condition (i)

$$L(\overline{u}^*, \overline{X}) = \text{Max } C\overline{X}^* - \overline{u}^* (A\overline{X}^* - b)$$

but by condition (iii), $u(AX^*-b) = 0$, therefore $L(\overline{u}^*, \overline{X}) = Z(\overline{X}^*)$ and this completes the proof.

6.2 THEORETICAL BASIS FOR THE SOLUTION ALGORITHM

In this section, the theoretical basis for the solution algorithm for the production scheduling problem will be given. First, the dual problem will be discussed; based on this a solution strategy will be defined and lastly the subgradient optimization algorithm will be presented.

6.2.1 Strategy for the Solution Algorithm

Theorems 6.1 and 6.2 give an important insight about the original problem (6.3) and its lagrangian (6.4). These two theorems tell us that the solution obtained by the lagrangian for any $\overline{u} > 0$ will be the upper bound of discounted profits for the original problem. When a vector $\overline{u}^* > 0$ is found such that a solution of the lagrangian satisfies the conditions of theorem 6.2 then one obtains the smallest upper bound to the original production scheduling problem. Hence, the optimal solution to the production scheduling problem is the smallest value to be obtained by the lagrangian.

Hence, for different $\overline{u} > 0$ values we will solve different problems of the following types:

$$L(\overline{u},\overline{X}) = \underset{X \,\varepsilon\, \Gamma}{\text{MAX}}\ C\overline{X} - \overline{u}_i\ (A\overline{X} - \overline{b}) \qquad (6.6)$$

The optimum solution to the original problem (6.3) will be the solution obtained from problem (6.6) that gives the minimum value of $L(\overline{u}_i)$ and for which the optimality conditions hold (theorem 6.2). In other words, the desired solution can be found by solving the following dual problem.

$$L^* = \text{Min } L(\overline{u},\overline{X}) \qquad (6.6)$$

$$\text{s.t.} \quad \overline{u} > 0$$

Corrolary 6.3. If $(\overline{X}^*, \overline{u}^*)$ satisfies the optimality conditions of theorem 6.2 for problem (6.3), then $\overline{u}^*$ is optimal in the dual problem.

Proof: We have $L(\overline{u}^*,\overline{X}) = C\overline{X}^* - \overline{u}^* (A\overline{X}^* - b) = C\overline{X}^* = Z(\overline{X}^*)$ by theorem 6.2. Since $Z(\overline{X}) < L(\overline{u}, \overline{X})$ for all $\overline{u} > 0$, and all feasible X then by theorem (6.1) $L^* < L(\overline{u}, \overline{X})$ for all > 0.

Thus the critical problem in solving the production scheduling problem becomes finding an optimum vector $\overline{u}^*$ where $\overline{u}^* > 0$ such that $L(u^*,\overline{X})$ is minimum.

Hence an ideal strategy for this purpose would be to design an iterative algorithm such that at each iteration a vector $\overline{u}^{k+1}$ is obtained such that the solution to dual

$L(\bar{u}^{k+1},\bar{X}^k) < L(\bar{u}^k,\bar{X}^k)$. This can be demonstrated by the following:

u^k		$L(u^k, \bar{X}^k)$
u^1		$L(u^1, \bar{X}^1)$
$\vdots$		$\vdots$
u^k		$L(u^k, \bar{X}^k)$
u^{k+1}		$L(u^{k+1}, \bar{X}^{k+1})$
$\vdots$		$\vdots$
$\bar{u}^*$	Optimum	L^*

Unfortunately there is no algorithm in existence which will converge to the optimum in a non-increasing sequence such as the one above.

6.2.2 Determination of Lagrange Multipliers ($\bar{u}$)

There are a number of ways to attack problem (6.6). These can be categorized as (1) the subgradient method (2) various versions of the Simplex Method implemented using column generation techniques (3) multiplier adjustment methods.

A few of the methods in category (2) and (3) are already applied to the production scheduling problem.

Johnson's (1968) application of the decomposition principle is one way to come up with multipliers and falls into category (2). Application of the dual simplex algorithm by Benham (1978) was another attempt to determine the multipliers. This also falls into category (2). The incremental adjustments of one or two multipliers as applied to production scheduling problem by Davis and Williams (1973) can be considered as type (3).

During the last ten years, a number of articles in the operations research literature have reported on the successful application of Held and Karp's subgradient optimization technique for adjusting multipliers in solving problems similar to problem (6.6) (Held and Karp 1971; Held, Wolfe, and Shapiro 1974; Geofrian 1974; Fisher and Shapiro 1974; Fisher, Northup and Shapiro 1975; Etcheberry 1977; Shapiro 1979; and Fisher 1981). The computational experience of this algorithm with large-scale problems reported in the literature indicates the superior performance of this approach over other linear and non linear methods. (Fisher, Northup and Shapiro 1975, Etcheberry 1977, Fisher 1981).

6.2.3 Subgradient Optimization

The subgradient optimization algorithm starts with a vector of multipliers $\overline{u}^0$. It then computes a direction and a step size to get a new vector of multipliers $\overline{u}^1$. This process is repeated, obtaining a sequence of vectors $\{\overline{u}^k\}$ that converges to a vector $\overline{u}^*$ that minimizes $L(\overline{u},\overline{X})$.

An m-vector S is called a subgradient of $L(\overline{u},\overline{X})$ at $\overline{u}^*$ if it satisfies (see Figure 16)

$$L(\overline{u}^*,\overline{X}^*) + (\overline{u} - \overline{u}^*) \, . \, S < L(\overline{u},\overline{X}) \text{ for all } \overline{u}$$

It can be shown that the half space $\{\overline{u}/(\overline{u} - \overline{u}^*) \, S > 0\}$ contains all solutions to dual with lower values of L. In other words, any subgradient appears to point in the direction of descent of L at $\overline{u}^*$.

It is shown in theorem 6.1 that

$$Z(\overline{X}^*) + \overline{u} \, (A\overline{X}^* - \overline{b}) < L(\overline{u},\overline{X}) \text{ for all } u$$

and by 6.2

$$L(\overline{u}^*,\overline{X}^*) - Z(\overline{X}^*) - \overline{u}^* \, (A\overline{X}^* - \overline{b}) = 0$$

Adding these relations yields

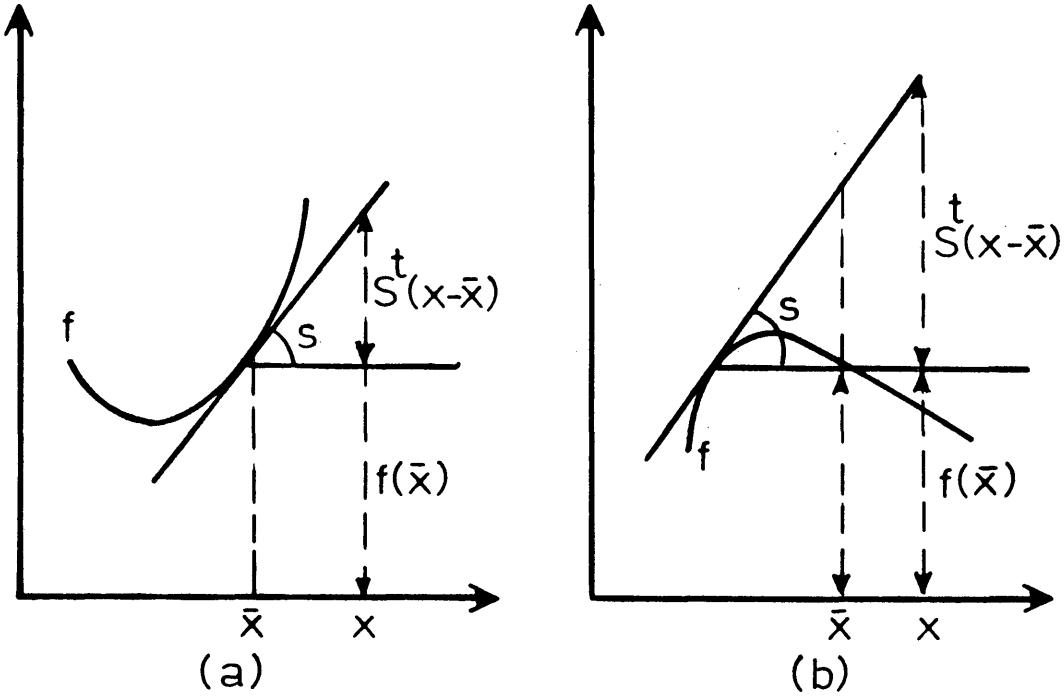


Figure 16. The Subgradient S of a Convex (a) and Concave Function (b)

$$L(\bar{u}^*,\bar{X}^*) + (\bar{u} - \bar{u}^*)\ (A\bar{X}^* - \bar{b}) < L(\bar{u},\bar{X}) \qquad \text{for all } \bar{u}$$

Comparing this relationship with the definition of a subgradient, it is apparent that a readily available subgradient is $S = (A\bar{X}^* - \bar{b})$.

Given an initial value $\bar{u}^o$ a sequence $\{\bar{u}^k\}$ is generated by the rule

$$u^{k+1} = u^k + t_k\ (A\bar{x}^k - \bar{b})$$

where $\bar{x}^k$ is an optimal solution to $L(u^k,X^k)$ and t_k is a positive scalar step size.

The hyperplane through $\bar{u}$ having $(A\bar{x}^k - b)$ as its normal determines a closed half space containing all points $\bar{u}^*$ such that $L(\bar{u},\bar{X}) - L(\bar{u}^*,\bar{X})) < 0$ and at each iteration, $\bar{u}$ moves into the half-space along the normal $S^k = (A\bar{X}^k - \bar{b})$. In particular, this half-space includes any point where L(.) assumes its maximum value, and a sufficiently small step produces a point closer than $\bar{u}$ to any such maximum point. In Held and Karp (1971), the limit on the appropriate step size is given as:

$$t_k = \frac{\alpha_k\ (\bar{L}(\bar{u}^*,\bar{X}^*) - L(\bar{u}^k,\bar{X}^k)}{||\ S^k\ ||^2}$$

where $||\ S^k\ ||$ denotes an Euclidean norm and $0 < \alpha_k < 2$.

The theoretical development and the convergence properties of this algorithm are given in Held and Karp (1971), Held, Wolfe and Crowder (1974) and in their references. The fundamental theoretical result is that $L(u^k, X^k) \text{ ---> } L^* \text{ if } t_k \text{ ---> } 0$ and $\sum_{i=0}^{k} t_i \text{ ---> } \infty$.

Although it can be shown that $\overline{u}^k$ is closer to $\overline{u}^*$ in the norm $||\overline{u}^* - \overline{u}||$ at each iteration, the values of $L(u^k,\overline{X}^k)$ do not form a non-increasing sequence. The sequence $L(u^k,\overline{X}^k)$ is guaranteed to converge to $L(\overline{u}^*,\overline{X}^*)$ only asymtotically. However, in practice, the problems solved by this method have proved to converge in a very few iterations to values $L(\overline{u}^k,\overline{X}^k)$ that are close to $L(\overline{u}^*,\overline{X}^*)$ (Fisher and Shapiro 1974, Etcheberry 1977). The computational results obtained in this dissertation also supports this assertion.

6.3 CONDITION OF NON-CONVERGENCE

Although the convergence of this algorithm is achieved in many cases, there is still a possibility that this algorithm may not converge to the optimum solution of the problem for a given set of constraint requirements. This non convergence can take place when there is no vector consisting of the lagrange multipliers which will give a solution feasible to the constraint of the problem.

When it is not possible to find a set of linear multipliers to satisfy the original primal problem's feasibility conditions, then it is said that the condition of gaps exists for the problem being solved (Everett 1963, Bazaara 1979).

The mathematical explanation for the existing gaps is that the feasible region of the dual space is not convex and as a result by using linear multipliers one can only determine the solutions which lie on the convex hull of the feasible region. This is graphically shown on Figure 17. In Figure 17, if the right hand side requirements of the problem being solved is b*, then the optimal solution is said to exist in the gap region and cannot be found by adjustments of the multipliers. The only solutions which can possibly be found by the modification of the multipliers are the ones corresponding to the right hand sides b_1 and b_2 which lie on the convex hull shown in Figure 17.

Although there might be a case where the exact solution cannot be found, it is proven by Everett (1963) that the solutions which can be found by the lagrange multiplier approach will still be optimum for the set of right hand sides which the existing solution satisfies. The following theorem proves this:

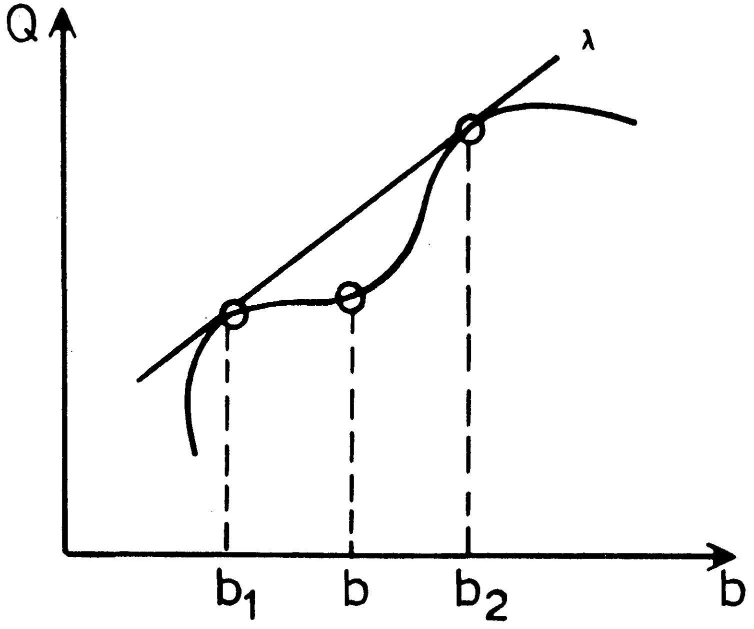


Figure 17. Condition of Gaps

Theorem 6.4 (Everett)

Given the following two conditions are true:

1. $\overline{u}^k$, k = 1, n are non-negative multipliers

2. $\overline{X}^* \varepsilon \Gamma$ maximizes the function $Z(\overline{X}) - (A\overline{X} - \overline{b})u^k$ over all $\overline{X} \varepsilon \Gamma$

then:

3. $\overline{X}^*$ maximizes $Z(\overline{X})$ over all those $X \varepsilon \Gamma$ such that $A\overline{X} \leq A\overline{X}^*$

Proof: For the solution $\overline{X}^*$ of lagrangian to be optimum to the original problem it must satisfy the optimality condition of theorem 6.2. Since replacing the right hand side of the relaxed constraint ($A\overline{X}^* \leq b$) with $\overline{b} = AX^*$ will satisfy the conditions (ii) and (iii) and since $\overline{X}^*$ is the optimum solution to, (i) $\overline{X}^*$ is optimum to the original problem for the case where $\overline{b}$ is replaced by $\overline{b} = A\overline{X}^*$.

In the case of the example of Figure 17, theorem 6.4 says that if one can accept the solution with the right side equal to b_1 for example, instead of b^*, the solution obtained from the lagrange multiplier approach will be optimum for this case. Certainly for the mine production scheduling problem the constraints are not considered to be very rigid, and requirements of blending and capacities do not have to be satisfied exactly. The production scheduling problem will in many situations tolerate the conditions of gaps if they exist. When the results coming from the lagrange multiplier approach do not exactly satisfy the constraint requirement, it is also possible to measure the difference from the true optimum. The following theorem due to Everett determines the deviations of the solution from the true optimum under the condition of gaps.

Theorem 6.5 (Everet)

If $\overline{X}^*$ comes within ε of minimizing the dual, (problem 6.6) for all $X \varepsilon \Gamma$

$$Z\ (\overline{X}^*) < L(\overline{u},\overline{X}) - \varepsilon$$

then $\overline{X}^*$ is an ε-optimal solution to the original problem with $\varepsilon = -\overline{u}\ (A\overline{X} - \overline{b})$

Proof: From theorem 6.2 $Z\ (\overline{X}^*) = L(\overline{u}^*,\overline{X}^*)$; otherwise from theorem 6.1 $Z(\overline{X}) < L(\overline{u},\overline{X})$, let's say $L(\overline{u},\overline{X})$ is ε amount bigger than $Z(\overline{X})$ then

$$Z(\overline{X}) = L(\overline{u},\overline{X}) - \varepsilon$$

but $L(\overline{u},\overline{X}) = Z(\overline{X}) - \overline{u}\,(A\overline{X} - \overline{b})$. Substituting this equation into the above equation

$$Z(\overline{X}) = Z(\overline{X}) - \overline{u}\,(A\overline{X} - \overline{b}) - \varepsilon$$

which gives $\varepsilon = \overline{u}\,(A\overline{X} - \overline{b})$ completing the proof.

CHAPTER 7
SOLUTION ALGORITHM FOR T-PERIOD SEQUENCING PROBLEM

7.1 INTRODUCTION

In this chapter an algorithm for the lagrangian subproblems of Chapter 6 will be given. This algorithm is based on linear programming and network concepts. Johnson's earlier work (1968) on the solution algorithm for the single time period problem and subsequent discussions provided the necessary intuition for the development of this algorithm.

7.1.1 The Ultimate Pit Limit Problem

The algorithm developed to solve the multi time period sequencing problem involves solving T (where T is a number of scheduling periods) ultimate pit limit problems with a modified block values. As such, in order to understand the development of the algorithm, a brief review of the ultimate pit limit problem, its network structure and available solution methods will be given in this section.

The formulation of the ultimate pit limit problem involves maximization of block values subject to sequencing constraints:

Max CX

Subject to:

$\min \sum_{n} P_n$

$\underline{st's}$

$$\begin{aligned} &EX < 0 \quad && U \\ &IX < 1 \quad \forall n && P \\ &\boxed{X_n > 0 \quad \forall n} \end{aligned} \qquad UE + PI \geq C \qquad (7.1)$$

$z = \pi b$

Where X_n indicates the fraction of the block n to be mined. C = vector of block values, E represents the coefficients of sequencing constraints (see Chapters 3 and 4) and I is the identity matrix. Problem (7.1) has a network structure because of the structure of the E matrix. As discussed in Chapter 4, the E matrix consists of 0, -1 and +1 values and the matrix is unimodular. As a result, the solution satisfying the set of these linear equations will be either 0 or 1. Hence any solution obtained by solving problem (7.1) will necessarily be an integer. This network structure becomes more apparent if one takes the dual of 7.1 (Johnson 1968). The dual problem is:

Minimize $P = \sum_{n} P_n$

subject to:

$$\begin{aligned} UE + IP &> C \ \forall n \\ U, P &> 0 \end{aligned} \qquad (7.2)$$

In order to see the network structure of the above problem, consider the cross sectional three block open pit model given in Figure 18. Figure 18(a) gives the block number and figure 18(b) gives the block values. The primal problem (7.1) for this example can be written as:

$$\text{Max } Z = X_1 - 2X_2 + 2X_3$$

Subject to:

$$\begin{array}{lcccl}
-X_1 + X_2 & & < & 0 & \left.\right\} U_{21} \\
& -X_2 + X_3 & < & 0 & \quad U_{32} \\
X_1 & & < & 1 & \\
& X_2 & < & 1 & \\
& \quad X_3 & < & 1 & \\
\multicolumn{4}{c}{X_1 > 0,\ X_2 > 0,\ X_3 > 0} &
\end{array} \qquad (7.1\text{-a})$$

The dual of the problem is

$$\text{Min } P = P_1 + P_2 + P_3$$

Subject To:

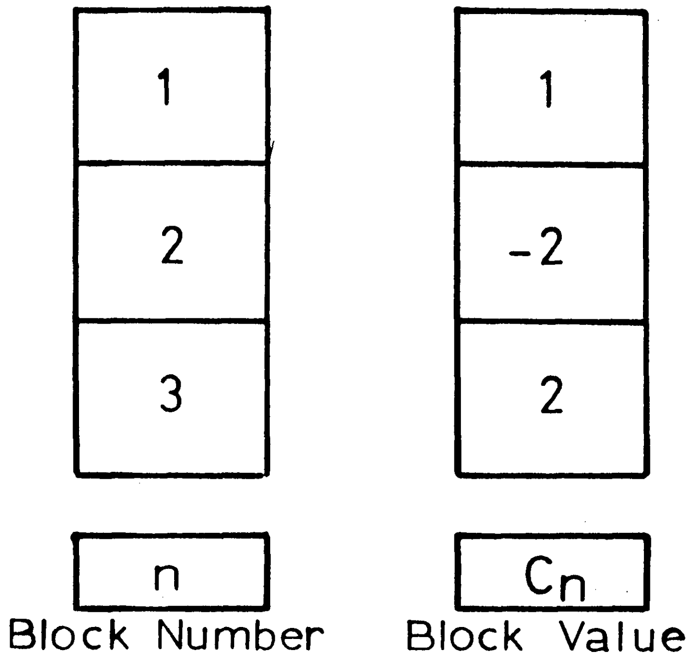


Figure 18. Simple 2-D Cross-Sectional Block Model

$$\begin{array}{rcrcrcrcr} -U_{21} & & & + & P_1 & & & > & 1 \\ U_{21} & - & U_{32} & & & + P_2 & & > & -2 \\ & & U_{32} & & & & + P_3 & > & 2 \\ & & & & & U_{21}, & U_{32}, P_n & > & 0 \end{array} \qquad (7.1\text{-b})$$

The network representation of this problem is given in Figure 19.

It can be seen from the network shown in Figure 19, that the dual variables U_{ij} represent the flows in arcs connecting block i to its restricting block j in order to satisfy the sequencing constraints of the ultimate pit limit problem. Furthermore, the flow is induced into the network if $c_n > 0$ and distributed around the network by arcs represented by the dual variables U_{ij} and taken out of the network either by $c_n < 0$ or P_n arcs. The dual constraints as shown in problems (7.2) and (7.1-b) may be interpreted as conservation of flow relations, i.e., they maintain the condition that flow in equals flow out around each node. The objective of the dual problem (7.2) is to send the minimum amount of flow through the P_n arcs of the network of Figure 19 such that the flow requirements of each node are satisfied.

Since one needs to send the **maximum** amount of flow possible through the negative gain arcs ($c_n < 0$) in order to

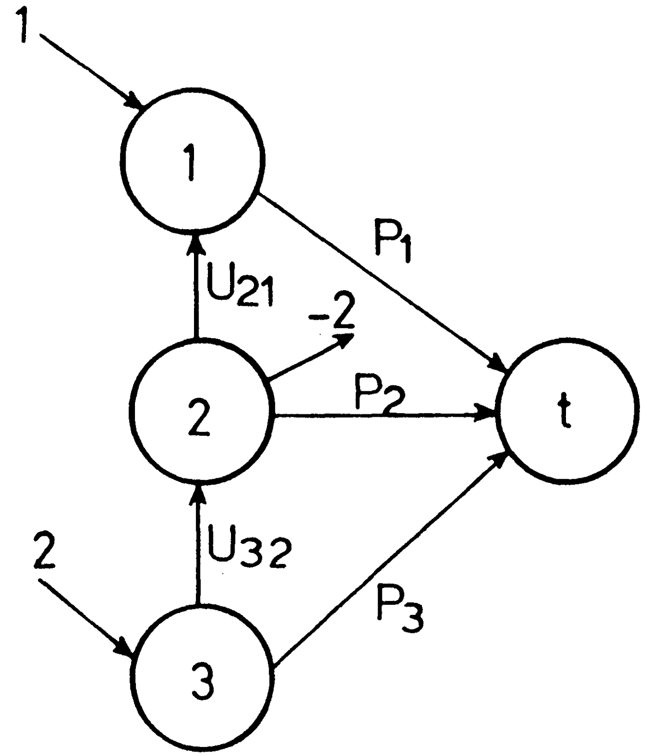


Figure 19. Network Representation of the Dual Problem (8.1-b)

send the minimum amount of flow through the P_n arcs, Johnson (1968) reformulated this as a maximum flow problem and proposed an efficient algorithm to solve the original ultimate pit limit problem (7.1). This algorithm is based on a labeling algorithm developed by Ford and Fulkerson (1963) and is known as the max flow ultimate pit limit algorithm (see Johnson 1968).

Another algorithm used to solve ultimate pit limit problems was developed by Lerchs and Grossmann (1965) and is known as the 3-dimensional graph theoritic tree algorithm. The equivalency between this algorithm and Johnson's maxflow algorithm was established by Barnes (1982). Therefore any of these algorithms can be used to solve the subproblems of the sequencing problem if they are equivalent to the ultimate pit limit problem discussed above.

7.2 PRIMAL AND DUAL OF THE LAGRANGIAN SUBPROBLEM

The lagrangian subproblems of the production scheduling problem for a given $\overline{u}$ vector as discussed in Chapter 6 will be of the following form.

$$\text{Max } Z = c^1 X^1 + c^2 X^2 \quad + c^T X^T$$

Subject to

$$\begin{aligned} &EX^1 && \leq 0 \\ &EX^1 + EX^2 && \leq 0 \qquad 7.3 \\ &EX^1 + EX^2 + \ldots EX^T && \leq 0 \\ &IX^1 + IX^2 + \ldots IX^T && \leq 1 \\ & & X_n^t & \geq 0 \qquad \forall n, t \end{aligned}$$

Because of the special structure, it appears that the above problem is easier to solve than the original scheduling problem which includes the capacity and blending constraints. Yet the subproblem (7.3) is still a rather large problem with variables and constraints in the hundreds of thousands, and is not easily solved in a straight forward manner.

Because it is known how to solve a problem of the form:

$$\begin{aligned} &\text{Max} \quad CX \\ &\text{s.t.:} \\ &\qquad EX \leq 0 \\ &\qquad IX \leq 1 \\ &\qquad X \geq 0 \end{aligned}$$

which is the ultimate pit limit problem, it appears useful to employ a substitution of variables in order to transform problem 7.1 into the form closer to this type of structure.

$$\boxed{\text{Let } W^t = \sum_{k=1}^{t} X^k}$$

By Variable . . .

and substitute W^t for the variables of problem (7.3). (This substitution was suggested by Johnson 1968). The problem 7.3 then becomes:

$$\text{Max } Z = (c^1 - c^2)\, W^1 + (c^2 - c^3)\, W^2 + \quad + c^t W^t$$

Preced ?

$$\begin{array}{lr}
\text{Subject to:} & \\
E\,W^1 & \leq 0 \\
\qquad + E\,W^2 & \leq 0 \\
\qquad\qquad\qquad\qquad E\,W^t & \leq 0 \qquad 7.4 \\
I\,W^1 - I\,W^2 & \leq 0 \\
\qquad I\,W^2 - I\,W^3 & \leq 0 \\
\qquad\qquad I\,W^{t-1} - W^t & \leq 0 \\
I\,W^t & \leq 1 \\
W^t & \geq 0
\end{array}$$

~~Different~~ Same blks

The structure of the E matrix was shown in Chapter 3 (p. 61). It has the characteristics of the transpose of the node-arc incidence matrix of a network or graph with at most two non-zero elements per row and they are either -1 or

+1. This network structure should become more apparent in the dual problem.

The dual of problem 7.4 is:

$$\text{MIN } D = \sum_{n=1}^{N} P_n$$

Subject to: (7.5)

$$\begin{array}{llllll}
EU^1 & & + IV^{12} & > (c^1 - c^2) & & (a) \\
\quad EU^2 & & - IV^{12} + IV^{23} & > (c^2 - c^3) & & (b) \\
\qquad +EU^{T-1} & & - IV^{j(T-1)} + IV^{(T-1)T} & > (C^j - C^{T-1}) & & (c) \\
 & + EU^T & - IV^{(T-1)T} + IV_n{}^T + P_n & > C^T & & (d) \\
 & & U^t,\ V^{it}{}_n,\ P_n > 0 & & &
\end{array}$$

7.3 NETWORK REPRESENTATION OF THE DUAL

A close examination of the dual problem (7.4) shows that this problem is a network. Each constraint represents flow into and out of a node.

The arcs corresponding to the dual variables U^t, V^{ij} are incident into a node (n, t) if the coefficient of U^t and V^{ij} in the E and I matrix respectively are -1, and incident out from the node (n, t) if the coefficient of U^t and Vn^{ij} in the E and I matrix respectively are +1.

As mentioned before, for the case of the ultimate pit limit problem, the E matrix in the dual problem represents the coefficients of the sequencing constraints. Thus the dual variables U_t again represents the flows in arcs connecting a given block i to the overlying restricting blocks n. The dual variables $V_n^{t,\ t+1}$ represent flow in arcs connecting a given block in time period t to the same block in time period t + 1.

The arcs corresponding to P_n are incident out of the node (n, T) where T indicates the last time period.

The set of constraints in problem 7.5 thus represent flow into and out of each node or block (Figure 20).

As seen from Figure 20 the right hand sides $(c^t - c^{t+1})$, and c^T of problem 7.3 correspond to either flow into a node of time period t if $(c^t - c^{t+1}) > 0$ (positive gain) or flow out of a node of time period t if $(c^t - c^{t+1}) \leqslant 0$ (negative gain). c^T represents the flow in or out of the nodes of only the last time period depending on the sign of c^T.

7.3.1 Numerical Example of a Dual Network

To clarify the relations discussed so far, consider a system of three blocks, as shown in Figure 21. Block values for three consecutive time periods are given in figure 22.

The primal formulation of this problem is

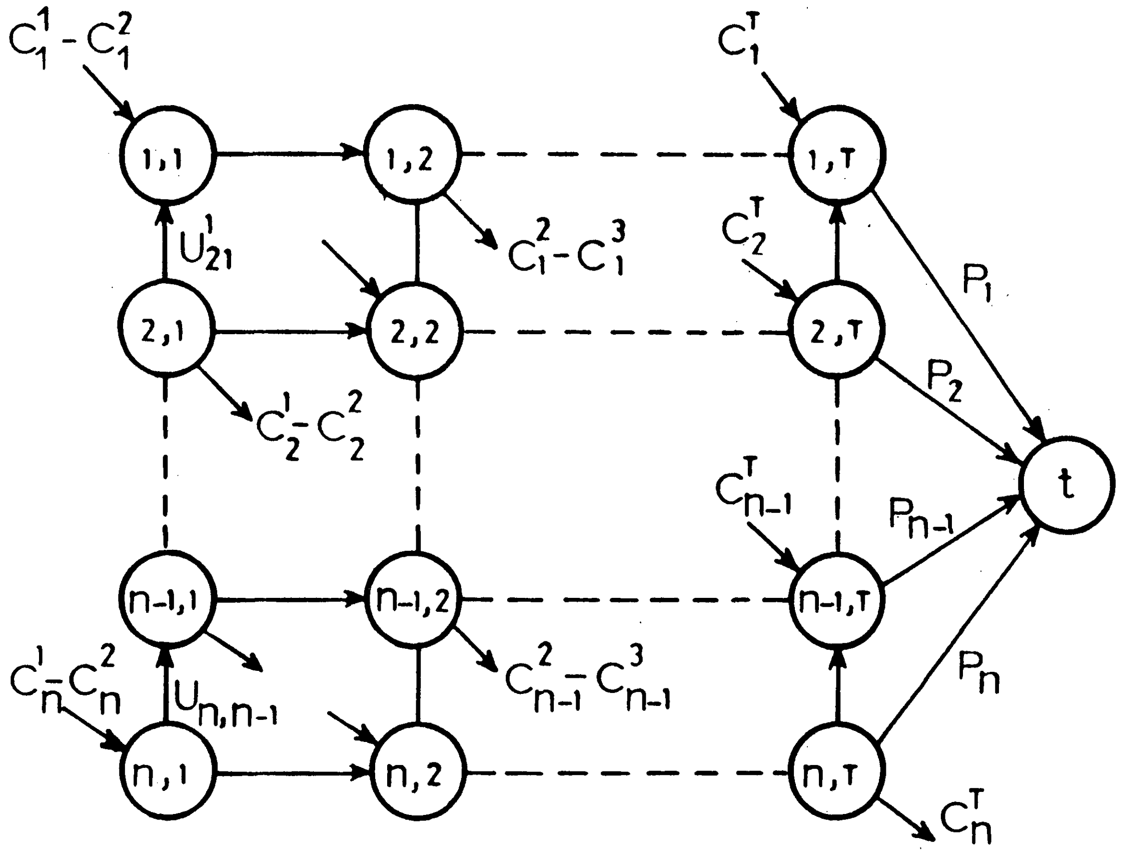


Figure 20. Node Representation of the Dual Problem

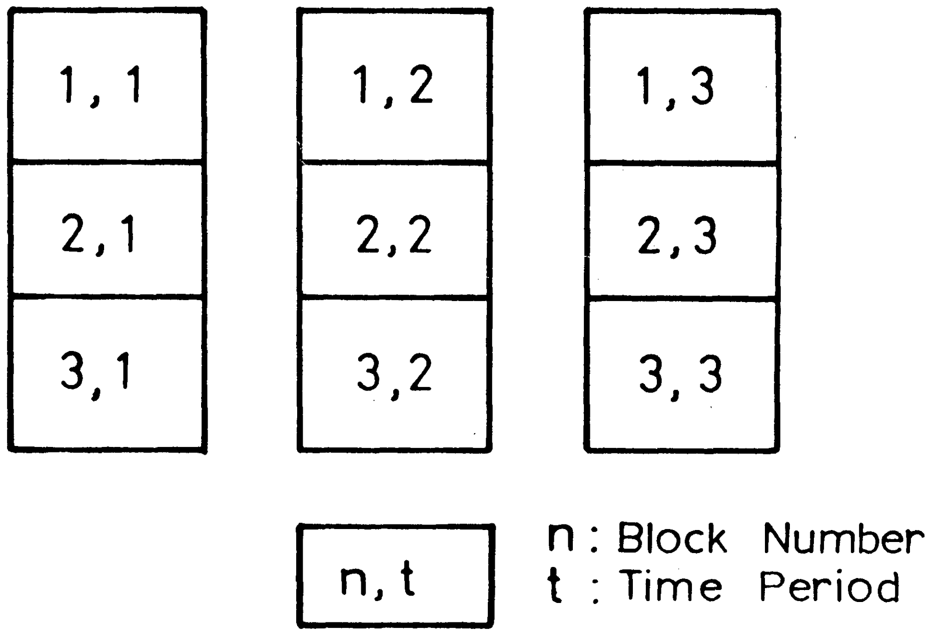
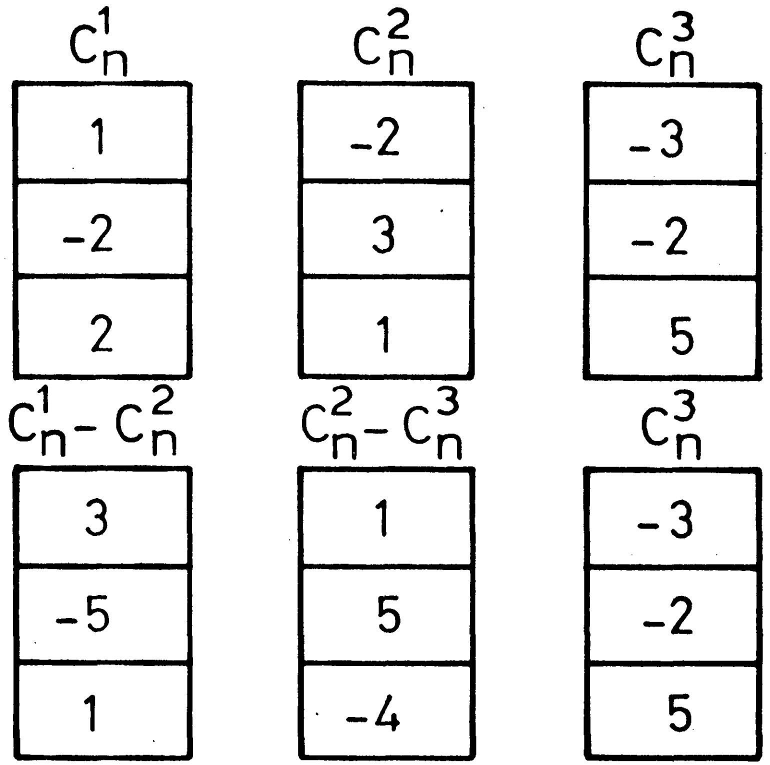


Figure 21. Node Number, Time Presentation of 2-Dimensional Block Number



C_n^1	C_n^2	C_n^3
1	-2	-3
-2	3	-2
2	1	5

$C_n^1 - C_n^2$	$C_n^2 - C_n^3$	C_n^3
3	1	-3
-5	5	-2
1	-4	5

Figure 22. (a) Block Values for Each Time Period
(b) Opportunity Values for Each Time Period
$(C_n^t - C_n^{t+1})$

$$\text{Max } Z = 1X^1{}_1 - 2X^1{}_2 + 1X_3{}^1 - 2X_1{}^2 + 3X_2{}^2 + 1X_3{}^2 - 3X_1{}^3 - 2X_2{}^3 + 5X_3{}^3$$

Subject to

$$\begin{array}{lllllll}
-X_1{}^1 + X_2{}^1 & & & & & & \le 0 \\
 & -X_2{}^1 + X_3{}^1 & & & & & \le 0 \\
-X_1{}^1 + X_2{}^1 & & - X_1{}^2 + X_2{}^2 & & & & \le 0 \\
 & X_2{}^1 - X_3{}^1 & & - X_2{}^2 + X_3{}^2 & & & \le 0 \\
X_1{}^1 - X_2{}^1 & & - X_1{}^2 + X_2{}^2 & & + X_1{}^3 - X_2{}^3 & & \le 0 \\
 & X_2{}^1 - X_3{}^1 & & - X_2{}^2 + X_3{}^2 & & - X_2{}^3 - X_3{}^3 & \le 0
\end{array} \tag{7.3-a}$$

$$\begin{array}{lllllllll}
X_1{}^1 & & & + X_1{}^2 & & & + X_1{}^3 & & & \le 1 \\
 & X_2{}^1 & & & + X_2{}^2 & & & + X_2{}^3 & & \le 1 \\
 & & + X_3{}^1 & & & + X_3{}^2 & & & + X_3{}^3 & \le 1
\end{array}$$

$$X_n{}^t \ge 0 \ \forall n, \ \& \ t$$

Transforming the problem by substituting

use by Voors...

$$W_n{}^t = \sum_{k=1}^{t} X^k$$

$$W_n{}^1 = X_n{}^1$$

$$W_n{}^2 = X_n{}^1 + X_n{}^2$$

$$W_n{}^3 = X_n{}^1 + X_n{}^2 + X_n{}^3$$

results in;

$$\text{Max } Z = 3W_1{}^1 - 5W_2{}^1 - \overset{1}{\cancel{0}}W_3{}^1 + 1W_1{}^2 + 5W_2{}^2 - 4W_3{}^2 - 3W_1{}^3 - 2W_2{}^3 + 5W_3{}^3 \qquad (7.4a)$$

Subject to:

$$\begin{array}{llllllllll}
-W_1{}^1 & + W_2{}^1 & & & & & & & & \le 0 \\
 & - W_2{}^1 & + W_3{}^1 & & & & & & & \le 0 \\
 & & & - W_1{}^2 & + W_2{}^2 & & & & & \le 0 \\
 & & & & - W_2{}^2 & + W_3{}^2 & & & & \le 0 \\
 & & & & & & - W_1{}^3 & + W_2{}^3 & & \le 0 \\
 & & & & & & & - W_2{}^3 & + W_3{}^3 & \le 0 \\
W_1{}^1 & & & - W_1{}^2 & & & & & & \le 0 \\
 & W_2{}^1 & & & - W_2{}^2 & & & & & \le 0 \\
 & & W_3{}^1 & & & - W_3{}^2 & & & & \le 0 \\
 & & & W_1{}^2 & & & - W_1{}^3 & & & \le 0 \\
 & & & & W_2{}^2 & & & - W_2{}^3 & & \le 0 \\
 & & & & & W_3{}^2 & & & - W_3{}^3 & \le 0 \\
 & & & & & & W_1{}^3 & & & \le 1 \\
 & & & & & & & W_2{}^3 & & \le 1 \\
 & & & & & & & & W_3{}^3 & \le 1
\end{array}$$

$$W_n{}^t \ge 0 \;\forall\; n \text{ and } t$$

In order to state the dual of this problem (7.4-a) let U_{it}^{t} be the dual variable corresponding to the first six constraints, $V_n^{t,t+1}$ be the dual variable corresponding to constraints 6 through 12 and P_n be the dual variables corresponding to last three constraints of problem (7.4). Then the dual of problem (7.4) is:

$$\text{Min } D = P_1 + P_2 + P_3 \qquad (7.5a)$$

Subject to:

$$\begin{array}{llllllll}
-U_{21}^{1} & +V_1^{12} & & & & & & \geq 3 \\
U_{21}^{1}-U_{32}^{1} & +V_2^{12} & & & & & & \geq -5 \\
U_{32}^{1} & & +V_3^{12} & & & & & \geq 1 \\
-U_{21}^{2} & -V_1^{12} & & +V_1^{23} & & & & \geq 1 \\
U_{21}^{2}-U_{32}^{\boxed{1}}{}^{2} & -V_2^{12} & & & +V_2^{23} & & & \geq 5 \\
U_{32}^{2} & & -V_3^{12} & & & +V_3^{23} & & \geq -4 \\
U_{21}^{3} & & & -V_1^{23} & & & +P_1 & \geq -3 \\
U_{21}^{3}-U_{32}^{3} & & & & -V_2^{23} & & +P_2 & \geq -2 \\
U_{32}^{3} & & & & & -V_3^{23} & +P_3 & \geq +5
\end{array}$$

U_{nj}^{t}, $V^{t,t+1}$, $P_n \geq 0$ ∀ all dual variables

7.3.2 Network Representation of the Example Problem

The network representation of this dual problem (7.5) is shown in Figure 23. This network not only shows the nature of the sequencing problem but also plays an important role in the development of the solution algorithm for the T-period sequencing problem (7.3).

The nodes of the network in figure 23 corresponds to constraint equations of the dual problem (7.5). (Each node of the network may be thought of as a given block n in time period t.) The flows in the arcs of the network correspond to the dual variables of problem (7.5):

U_{ij}^{t} = Dual variable representing the flow in the arcs between block i and block j in time period t. (An arc between block i and j will be present only if block j must be mined to mine block i).

$V_{n}^{t,t+1}$ = Dual variable representing the flow on arcs between time period (t) and time period (t+1) block n.

P_{n} = Dual variable representing the arcs between all the nodes of the last time period and the termination node X.

Figure 23

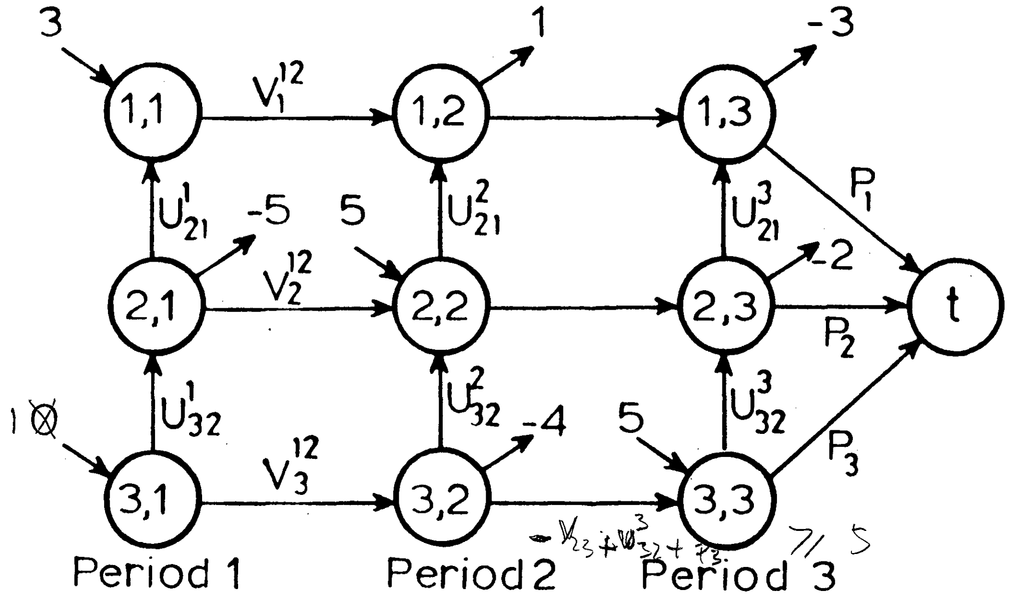


Figure 23. Network Representation of the Dual Problem (8.6)

It can be seen from the network of Figure 23 that the flows will be induced into the network if $(C_n{}^t - C_n{}^{t+1}) > 0$ or $C_n{}^T > 0$ and will be taken out of the network if $(C_n{}^t - C_n{}^{t+1}) < 0$, or $C_n{}^T < 0$ and through the Pn arcs. All the other arcs serve to distribute the flows through the network.

The objective of the dual problem (7.5) is to send the minimum amount of flow through the P_n arcs of the network in such a way that the requirements of $(C_n{}^t - C_n{}^{t+1})$ or $C_n{}^T$ are satisfied at each node.

It can be seen from the network in Figure 23 that the only way to send the minimum amount of flow through the P_n arcs of the network 6 is to take as much flow as possible out through the negatives gain arcs. Looking at the dual problem from the maximum flow perspective provides an effective strategy for the solution algorithm (Johnson 1968).

7.4 THE EQUIVALENT FORMULATION OF THE DUAL PROBLEM (7.3)

The ~~maximum flow~~ equivalent of this dual problem (7.3) can be established with a slight modification to the network of Figure 23. By connecting all the nodes with positive gains to a source node and all the nodes with negative gains to a sink node as shown in Figure 24, it can be intuitively seen that it is possible to minimize the flow going through

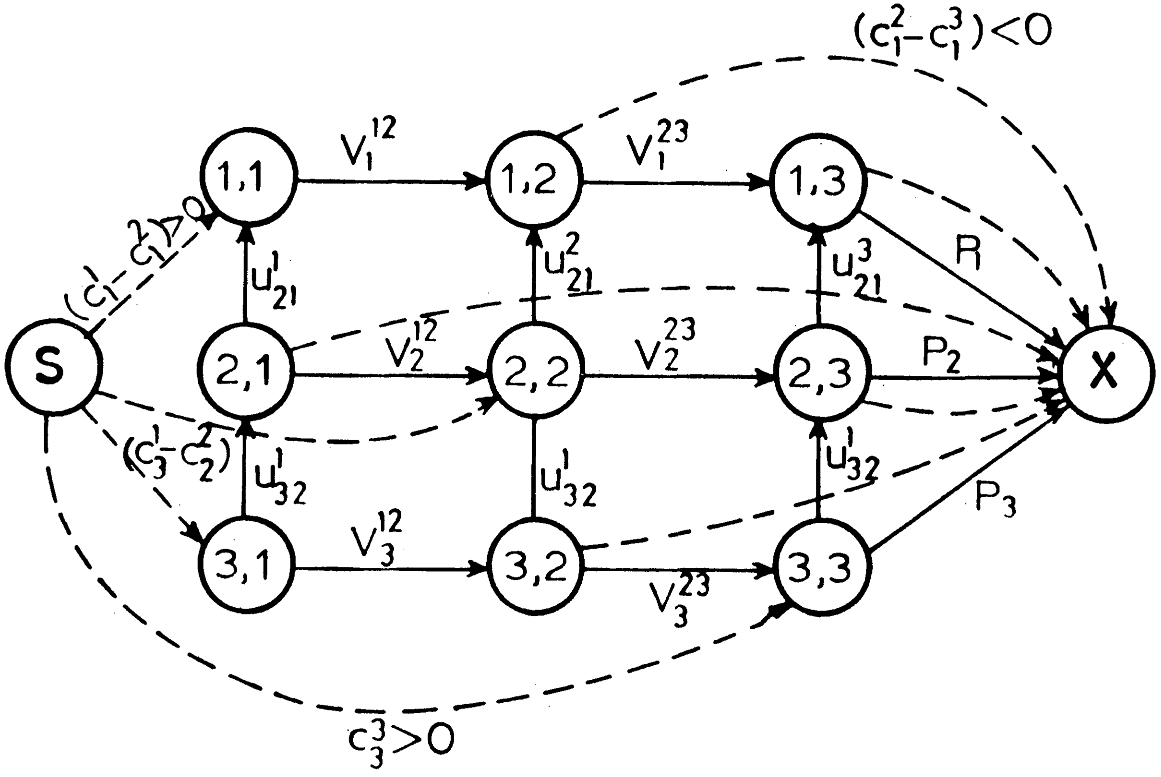


Figure 24. Modified Network for the Dual Problem 8.3

the P_n arcs by sending as much flow as possible through the negative gain arcs of different nodes. Hence, the dual problem (7.5) can be formulated as a Maximum Flow problem. Let

f^t_{sn} = The flow going from source node to node n of time period t with $c_n^t - c_n^{t+1} > 0$ or $c_n^T > 0$. (See Figure 25).

f^t_{mx} = The flow going from a node m of time period t with $c_n^t - c_n^{t+1} < 0$ or $c_n^T < 0$ to the sink node. (See Figure 25).

(+)

N_t^{t} = The set representing the nodes with positive gains ($c_n^t - c_n^{t+1} > 0$, or $c_n^T > 0$) i.e. the nodes connected to sources node in Figure 24.

N_t^- = The set representing the nodes with negative gains ($c_n^t - c_n^{t+1} < 0$, or $c_n^T < 0$).

Then the max flow formulation of the problem can be written as:

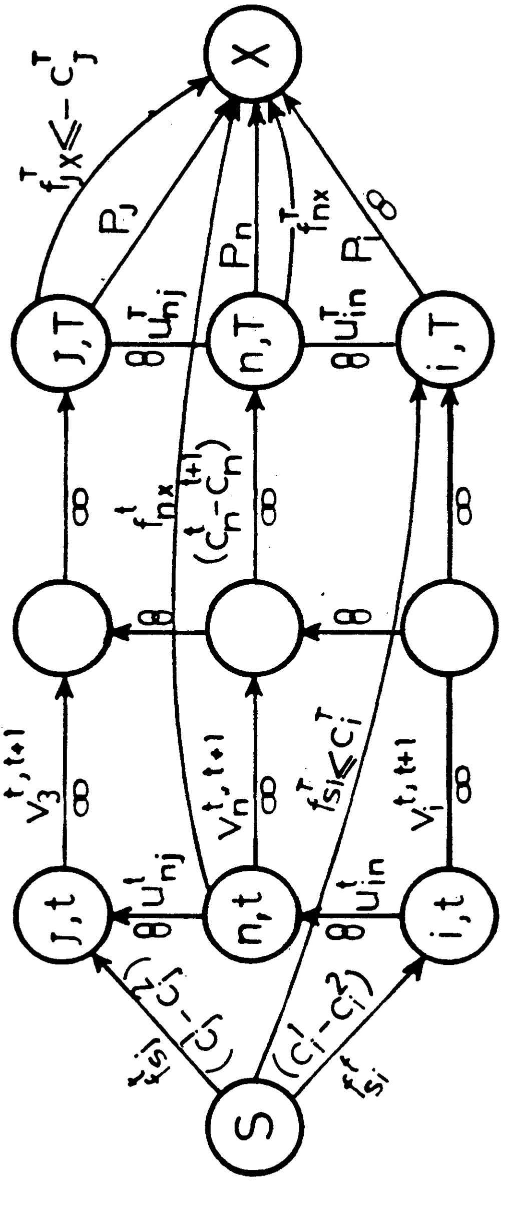


Figure 25. Network Representation of the Max Flow Problem

Problem (7.9):

$$\text{Max} \quad \sum_{t=1}^{1} \quad \sum_{m \varepsilon N_t^-} f_{mx}^t \qquad (7.9)$$

Subject to:

the constraint set for the nodes in time period 1

$$-f_{sn}^1 - u_{in}^1 + u_{nj}^1 + v_n^{12} = 0 \qquad n \ \varepsilon \ N_1^+ \quad (a)$$

$$-f_{mx}^1 - u_{im}^1 + u_{mj}^1 + v_m^{12} = 0 \qquad m \ \varepsilon \ N_1^- \quad (b)$$

$$f_{sn}^1 \quad < (c_n^1 - c_n^2) \qquad n \ \varepsilon \ N_1^+ \quad (c)$$

$$f_{mx}^1 \quad < -(c_n^1 - c_n^2) \qquad m \ \varepsilon \ N_1^- \quad (d)$$

the constraint set for nodes in the time periods between the first and the final time period (T) (t = 2...T-1)

$$-f_{sn}^t - u_{in}^t + u_{nj}^t - v_n^{t-1,t} + v_n^{t,\ t+1} = 0$$

$$+f_{mx}^t - u_{in}^t + u_{nj}^t - v_n^{t-1,t} + v_n^{t,t+1} = 0$$

$$f_{sn}^t \quad < (c_n^t - c_n^{t+1})$$

$$f_{mx}{}^{t} \qquad < -(c_m{}^{t} - c_m{}^{t+1})$$

$$\forall\ n\ \varepsilon\ N_t^+\ ,\ (t = 2\ \ldots\ T\text{-}1)$$
$$\forall\ m\ \varepsilon\ N_t^-,\ (t = 2\ \ldots\ T\text{-}1)$$

the constraint set for the last time period (T)

$$-f_{sn}{}^{T} - u_{in}{}^{T} + u_{nj}{}^{T} - v_n{}^{T-1,T} + Pn = 0 \qquad \forall n\ \varepsilon\ N_T^+$$
$$+f_{mx}{}^{t} - u_{im}{}^{T} + u_{nj}{}^{T} - v_m{}^{T-1,T} + Pm = 0 \qquad \forall m\ \varepsilon\ N_T^-$$

$$f_{sn}{}^{T} \qquad < (c_n{}^{T}) \qquad \forall\ n\ \varepsilon\ N_T^+$$
$$f_{mx}{}^{T} \qquad < -(c_n{}^{T}) \qquad \forall\ m\ \varepsilon\ N_T^-$$

$$f_{sn}{}^{t},\ f_{mx}{}^{t},\ u_{in}{}^{t},\ u_{nj}{}^{t},\ v_n{}^{t,t+1},\ Pm > 0 \qquad \forall n,\ m,\ t.$$

7.4.1 The Dual of the Maximum Flow Problem (7.9)

The equivalence of the max flow problem to problem (7.4) can be seen by taking the dual of this max flow problem. Let:

$a_n{}^{t}$ = The dual variable corresponding to the conservation of flow equations 7.9 (a) and (b) for time period t.

$g_{sn}{}^{t}$ = The dual variable corresponding to the flow upper bound constraint equations 7.9(c), in time period t for the positive nodes $n \in N_t^+$.

$h_{mx}{}^{t}$ = The dual variable corresponding to the flow upper bound constraint equations 7.9(d) in time period t for the negative gain nodes $m \in N_t^-$.

Then the dual of this problem (7.9) is

$$\text{Min} \sum_t \sum_{n \varepsilon N_t^+} (c_n{}^t - c_n{}^{t+1})\, g_{sn}{}^t - \sum_t \sum_{m \varepsilon N_t^-} (c_m{}^t - c_m{}^{t+1})\, h_{mx}^t$$

subject to:

$$\begin{array}{rlll}
-a_k{}^t + a_i{}^t & > 0 & \text{(a)} & \\
a_k{}^t - a_j{}^t & > 0 & \text{(b)} & \\
-a_k{}^t - a_k{}^{t-1} & > 0 & \text{(c)} & (7.10) \\
a_k{}^t - a_k{}^{t+1} & > 0 & \text{(d} & \\
a_k{}^T & > 0 & \text{(e)} &
\end{array}$$

equation (a) through (e) are for all k and t

$$\begin{array}{rlll}
-a_n{}^t + g_{sn}{}^t & > 0 & \forall t,\ n \ \varepsilon\ N_t^+ & \text{(f)} \\
a_m{}^t + h_{mx}{}^t & > 1 & \forall t,\ m \ \varepsilon\ N_t^- & \text{(g)}
\end{array}$$

$a_k{}^t$ is unrestricted

$g_{sn}{}^t,\ h_{mt}{}^t > 0$

The equivalent of this dual problem to the original primal transformed problem (7.4) can be shown by a substitution of variables.

Let:

$g_{sn}{}^t = 1 - W_n{}^t$

$h_{mx}{}^t = W_m{}^t$

substituting these in the objective function of problem (7.10)

$$\text{Min} \sum_t \sum_{n \varepsilon N_t^+} (c_n{}^t - c_n{}^{t+1})(1 - W_n{}^t) - \sum_t \sum_{m \varepsilon N_t^-} (c_m{}^t - c_m{}^{t+1}) W_m{}^t$$

rearranging the terms

$$\text{Min} \sum_t \sum_{n \varepsilon N_t^+} (c_n{}^t - c_n{}^{t+1}) - \sum_t \sum_{n \varepsilon N_t^+} (c_n{}^t - c_n{}^{t+1})(W_n{}^t)$$

$$- \sum_t \sum_{m \varepsilon N_t^-} (c_m{}^t - c_m{}^{t+1})\ W_m{}^t$$

minimizing the above function is equivalent to

$$\text{Max} \sum_t \sum_{n \varepsilon N_t^+} (c_n{}^t - c_n{}^{t+1}) w_n{}^t + \sum_t \sum_{m \varepsilon N_t^-} (c_m{}^t - c_m{}^{t+1}) w_m{}^t$$

and this is exactly the same as the objective function of problem (7.4).

What remains is to show that the optimal solution to the max flow problem is also feasible to the constraint sets of problem (7.4).

At the optimality for the max flow problem (7.9) if for some nodes n ε N^+ the following is true

$$f_{sn}{}^t < (c_n{}^t - c_n{}^{t+1}) \text{ and } f_{sn}{}^t > 0$$

then for all nodes $m \varepsilon N-$ in the cone of node n

$$f_{mx}{}^t = -(c_m{}^t - c_m{}^{t+1}) \text{ and } f_{mx}{}^t > 0$$

from the complimentary slackness conditions (see Dantzig 1963), if $f_{sn}{}^t > 0$, then

$$-a_n{}^t + g_{sn}{}^t = 0 \Rightarrow a_n{}^t = g_{sn}{}^t \qquad (7.11)$$

and again by the complimentary slackness condition if $f_{mx}{}^t > 0$ then

$$a_m^t + h_{mx}^t - 1 = 0 \qquad (7.12)$$

this gives

$$a_m^t = 1 - h_{mx}^t$$

substituting $g_{sn}^t = 1 - W_n^t$, $h_{mx}^t = W_m^t$ into equations (7.11) and (7.I2)

$$\begin{array}{ll} a_n^t = 1 - W_n^t & n \varepsilon \; N+ \\ a_m^t = 1 - W_m^t & m \; \varepsilon \; N- \end{array}$$

substituting the values of $a_n^t = 1 - W_n^t$ into all constraints but the last two of dual problem (7.10) one obtains:

$$\begin{array}{ll} -\, W_k^t + W_i^t & < 0 \\ W_k^t - W_j^t & < 0 \\ W_k^{t-1} - W_k^t & < 0 \\ W_k^t - W_k^{t+1} & < 0 \\ W_k^T & < 1 \end{array}$$

Since the problem being solved has network structure with all columns or rows containing at most two non-zero elements and they are (± 1) then the matrix is unimodular and when the right hand sides are also integer then the values of the variables in the optimal solution are also integer. If the right hand sides are either 0 or 1 then the values of the variables in the optimal solution are also 0 or 1, (See Jensen and Barnes 1980). Hence:

$a_n{}^t = 0$ or 1 $\forall n$ and t

$g_{sn}{}^t = 0$ or 1 $\forall t,\ n \varepsilon N_t^+$

$h_{mx}{}^t = 0$ or 1 $\forall t,\ m \varepsilon N_t^-$

since these conditions guaranty that

$W_k{}^t > 0 \quad \forall\ k,\ t$ Q.E.D.

therefore, the optimal solution to the max flow problem will also satisfy the feasibility conditions of problem (7.4). As a result, solving the max flow problem is equivalent to solving the transformed problem (7.4)

7.5 THE DECOMPOSITION OF THE DUAL PROBLEM (7.3)

It has been shown that solving the maximum flow problem (7.9) is equivalent to solving the transformed problem (7.4). Hence, problem (7.9) can be solved by an efficient algorithm such as the labeling algorithm developed by Ford and Fulkerson (1963). The solution to the original problem (7.4) can be recognized from the complimentary slackness conditions as follows:

Case 1: If in the final solution some of the positive gain arcs have slacks:

$$[f_{sn}{}^{t} - c_{n}{}^{t} - c_{n}{}^{t+1}] > 0$$

then by the complimentary slackness conditions corresponding dual variable $g_{sn}{}^{t} = 0$, and this requires $W_{n}{}^{t} = 1$.

Case 2: If in the final solution some of the negative gain arcs have slacks:

$$[f_{mx}{}^{t} + (c_{m}{}^{t} - c_{m}{}^{t+1})] > 0$$

then again by the complimentary slackness conditions $h_{mx}{}^{t} = 0$ and this requires $W_{m}{}^{t} = 1$.

Although the maximum flow problem (7.9) can be solved by the labeling algorithm and the solution to the original problem can be identified as shown above from the complimentary slackness conditions, the maximum flow problem (7.9) is still a considerably large problem. By studying the dual problem (7.5) and its network representation as shown through the example problem in Figure 23, it is possible to decompose the dual problem (7.5) further into smaller subnetworks each consisting of nodes for a given time period.

It can be seen from the network in Figure 23, that the nodes in a given time period are connected by u_{ni}^{t} arcs which enforce the sequencing requirements in space. These arcs have no interactions with other time periods. The only connection between the time periods are through the $v_n^{t,t+1}$ arcs which enforce the sequencing requirements in time. (See Figure 23.)

Therefore, the network structure of the dual problem indicates that the dual problem (7.5) is made up of T subproblems; each subproblem corresponds to a time period. In a given subproblem, the flow is induced into the network by way of positive gain arcs $(c_n^t - c_n^{t+1}) > 0$ and taken out of the network by way of a negative gain arc $(c_n^t - c_n^{t+1}) < 0$. The $v_n^{t,t+1}$ arcs carry the excess flow into the next period.

When the dual problem (7.5) is solved, optimality of the dual problem (7.5) requires that at each time period as much flow as possible should be taken out through the negative gain arcs, $(c_n^t - c_n^{t+1}) < 0$, and only the excess flow should be passed onto the next time period. Since it is never necessary and never improves the dual problem objective to send flow across the time lines, one needs to send out as much flow as possible in the same period through the negative gain arcs. If this is not the case in a given solution the flows going through the Pn arcs would not be minimum and this will contradict the optimality assumption.

Therefore it is possible to solve the dual problem (7.3) by decomposing it into smaller max flow problems each corresponding to a given time period.

By solving the dual problem (7.3) this way, the only interaction needed between the maxflow problems of individual time periods will be in passing the net flows (remainder) across the time line.

The logical way to pass the excess flow from the preceding period nodes into the next time period node is to taking the node potential of individual nodes, which are labeled in the preceding max flow solutions to the node potential of the corresponding nodes in the succeeding time period, and then solving the succeeding time period max flow problem.

7.5.1 Decomposed Method of Solution

The proposed method of solution for the dual problem is to solve the max flow problem by considering the time period 1 nodes first. This method involves solving the following problem:

$$\text{Max} \sum_{m \varepsilon N_1^-} f_{mx}^{1}$$

subject to:

$$-f_{sn}^{1} - u_{in}^{1} + u_{nj}^{1} = 0 \qquad \forall n,\ n \ \varepsilon\ N_1^{+}$$

$$f_{mx}^{1} - u_{im}^{1} + u_{mj}^{1} = 0 \qquad \forall m,\ m \ \varepsilon\ N_1^{-}$$

$$f_{sn}^{1} \qquad \leq (c_n^{1} - c_n^{2}) \qquad \forall n,\ n \ \varepsilon\ N_1^{+}$$

$$f_{mx}^{1} \qquad \leq -(c_n^{1} - c_n^{2}) \qquad \forall m,\ m \ \varepsilon\ N_1^{-}$$

$$f_{sn}^{1},\ f_{mx}^{1},\ u_{in}^{1},\ u_{nj}^{1} \geq 0 \quad \forall\ n,\ m$$

By solving the above problem one will be able to determine those nodes which have the potential to be mined in time period 1 (n ε sl).

The nodes which are identified to be in the solution set (n ε s^1) will also be the nodes with the excess flow potential. In actuality, the overall total value of the optimum pit for the time period 1 max flow problem will be the total excess flows which will need to be passed on to the next time period.

The logical way to pass this excess flow from the preceding period nodes into the next time period nodes is to take the node potential of individual nodes $n \varepsilon S^1$ and to add them to the node potential of the corresponding nodes in the succeeding time period.

This process of addition when carried out on a node to node basis for all the nodes which are candidates to be in the solution $(n \varepsilon s^1)$ is identical to solving the max flow problem (7.7) by the max flow labeling routine. This is mainly because, around each node, $u_{ij}{}^t$ arcs serve only to distribute the induced flows <u>within</u> the time period, and the effects of $u_{ij}{}^t$ arcs in passing the excess flows from one period to the next cancel out in the addition process.

Once addition of the node potentials for nodes identified in the solution set in time period 1 to those in time period two is completed then for the time period 2 max flow problem one solves

$$\text{Max} \sum_{m \varepsilon N_{\bar{2}}} f^2_{mx}$$

subject to:

$$-f^2_{sn} - u^2_{in} + u^2_{nj} = 0 \qquad \forall n,\ n \in N^+_2$$
$$f^2_{mx} - u^2_{in} + u^2_{nj} = 0 \qquad \forall m,\ m \in N^-_2$$

for those nodes n, $n \in s^1$

$$f^2_{sn} \leq C^1_n - C^3_n) \qquad \forall n,\ n \in N^+_1$$
$$f^2_{mx} \leq -(C^1_n - C^3_n) \qquad \forall n,\ n \in N^-_1$$

and for the rest of the nodes,

$$f^2_{sn} \leq (C^2_n - C^3_n) \qquad \forall n,\ n \in N^+_1$$
$$f^2_{mx} \leq -(C^2_n - C^3_n) \qquad \forall n,\ n \in N^-_1$$

$$f^2_{sn},\ f^2_{mx},\ u^2_{in},\ u^2_{nj} \geq 0\ \forall n,\ m$$

and if this process is continued all the way through the last time period, one will solve the original dual problem (7.3) from which the solution to the original problem 7.2 can easily be identified.

7.6 STEPS OF THE ALGORITHM

The proposed method for solving the original multi period sequencing problem (7.3) is to transformed it into problem (7.4) and transform this into problem (7.5) and solve problem (7.5) by an iterative process; at each iteration solving a max flow problem which is equivalent to solving a single time period utimate pit limit problem.

At a given iteration, the steps of the algorithm are as follows:

Step 1: Solve the max flow problem of type (7.8) with only nodes $n \varepsilon N^t$, when N^t is the set of nodes included in time period t. Use the modified node potentials (if t=1 use the original node potentials $c_n^1 - c_n^2$), to identify those blocks which can be labeled (i.e. possibly mined) in time period t. Let $k \varepsilon s^t$ be the set of those blocks.

Step 2: Modify the node potentials of nodes $k \varepsilon N^{t+1}$ such that $k \varepsilon s^t$ by adding their current node potential values in time period t to the node potentials of their corresponding nodes in the next time period t+1. If t=T skip this step and go to step 3. If t<T, set t = t+1 and go to step 1.

Step 3: Terminate with the optimal solution. Mine those blocks $\varepsilon s^t ns^{t+1} ..s^T$ in time period t.

At each iteration the algorithm solves an ultimate pit limit problem of type (7.10). This problem can either be solved by Johnson's Max Flow Ultimate Pit Limit algorithm (1968) or by Lerchs-Grossmann 3-dimensional Graph Theoretic Tree algorithm (Lerchs and Grossmann 1965). The algorithm is so simple between the iterations, that the only requirement is the modification of the node potentials for those blocks which are identified in the solution set of previous time period problems.

7.6.1 Numerical Example of the Algorithm

As an example to demonstrate the algorithm, consider the 2-dimensional block diagram in Figure 26. The values assigned to each block with respect to the three time periods of the scheduling duration are given in Figure 27.

The transformation of the original problem (7.3) into the form of problem (7.4) requires the calculation of $(C_n^1 - C_n^2)$ and $(C_n^2 - C_n^3)$. The modified block values shown in Figure 27. Note the $(C_n^1 - C_n^2)$ values are assigned to time period 1, the $(C_n^2 - C_n^3)$ values are assigned to time period 2, and time period 3 values remain as they were (see the objective function of the transformed problem (7.3)).

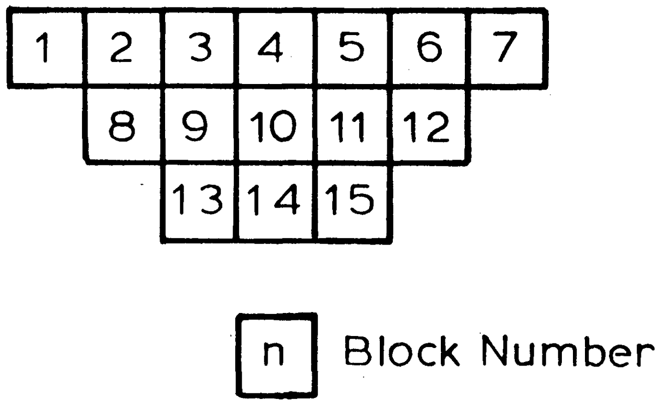
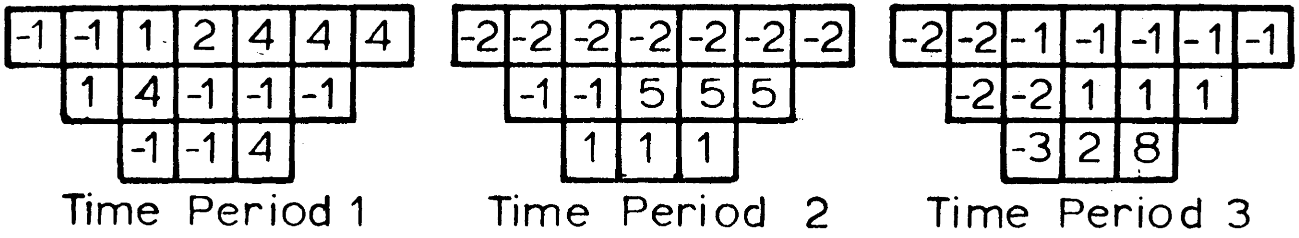


Figure 26. Cross-Sectional Representation of the Two Dimensional Block Model



-1	-1	1	2	4	4	4
	1	4	-1	-1	-1	
		-1	-1	4		

Time Period 1

-2	-2	-2	-2	-2	-2	-2
	-1	-1	5	5	5	
		1	1	1		

Time Period 2

-2	-2	-1	-1	-1	-1	-1
	-2	-2	1	1	1	
		-3	2	8		

Time Period 3

Figure 27. Original Block Values for Different Time Periods (C_n^t)

Iteration 1:

Step 1: Solve the max flow problem, i.e., find the ultimate pit limit for the transformed block values corresponding to time period 1 as given in Figure 27(a). The set of blocks which are in the optimal pit limit are:

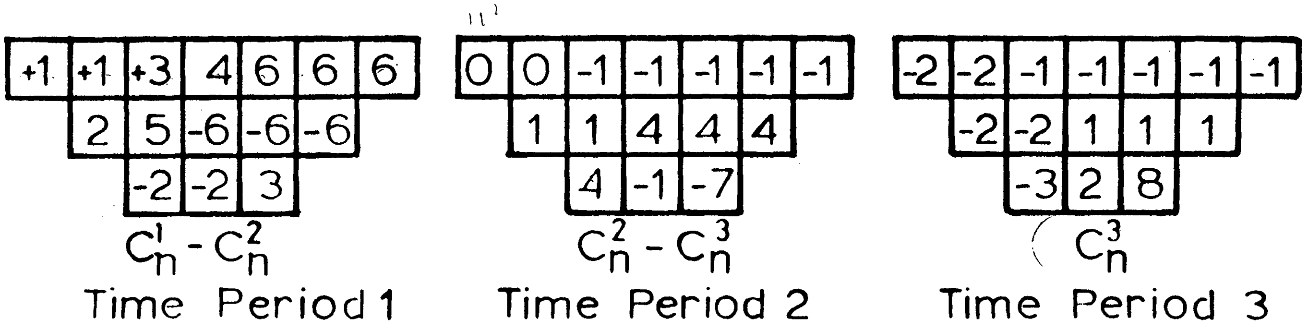
$$s^1 = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

and this is shown in Figure 28.

Step 2: Now add the transformed block values (node potentials) of those blocks which are in the optimal pit of first time period ($n \varepsilon s^1$) to their corresponding block values in time period 2 (to the values of Figure 27 (b)) and keep the rest of the blocks values in figure 27(b) as they are. The resulting block values are shown in Figure 29. Set t=2 and go to iteration 2.

Iteration 2:

Step 1: Solve the ultimate pit limit problem corresponding to time period 2 with the modified block values as shown in



+1	+1	+3	4	6	6	6
	2	5	-6	-6	-6	
		-2	-2	3		

$C_n^1 - C_n^2$

Time Period 1

0	0	-1	-1	-1	-1	-1
	1	1	4	4	4	
		4	-1	-7		

$C_n^2 - C_n^3$

Time Period 2

-2	-2	-1	-1	-1	-1	-1
	-2	-2	1	1	1	
		-3	2	8		

C_n^3

Time Period 3

Figure 28. Transformed Block Values for Different Time Periods $(C_n^t - C_n^{t+1})$ or C_n^T

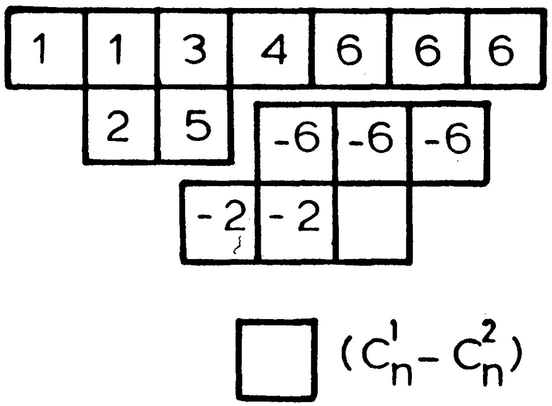


Figure 29. Optimal Pit Limit Solution for Time Period 1

Figure 28. The set of blocks which are in the optimal pit (see Figure 30) are:

$$S^2 = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13\}$$

Step 2: For those blocks identified to be in the optimal solution of the time period 2 as shown in Figure 30, add their current block values as shown in Figure 30 to the corresponding block values in time period 3 (that is, the corresponding values of Figure 27(c)). Keep the rest of the Figure 27(c) values as they are. The resulting block values are shown in Figure 31. Set t=3 and go to iteration 3.

Iteration 3:

Step 1: Solve the ultimate pit limit problem with the block values as given in Figure 31. The set of blocks which are in the optimal pit (see Figure 32) are:

$$S^3 = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15\}$$

Step 2: Since t = T = 3 go to step 3.

Figure 30

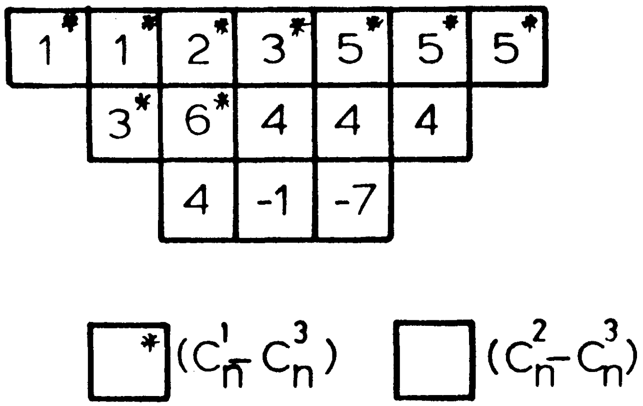


Figure 30. Modified Block Values of Time Period 2

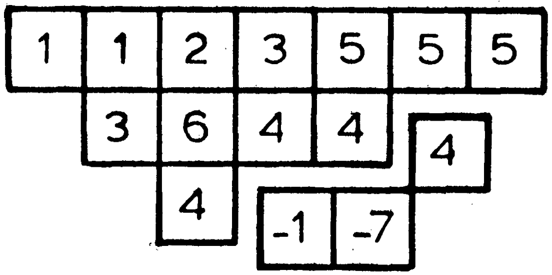


Figure 31. Optimal Pie Limit Solution for Time Period 2

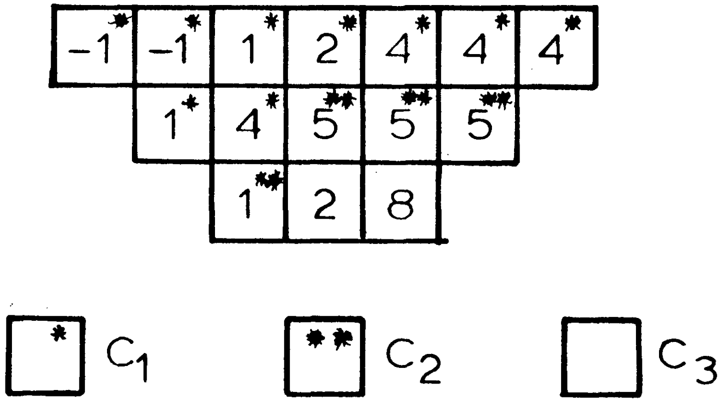


Figure 32. Modified Block Values for Time Period 3

Step 3: Mine those blocks $m^1 = s^1 \cap s^2 \cap s^3$ in period 1:

$$M^1 = \{1, 2, 3, 4, 5, 6, 7, 8, 9,\}$$

Mine those blocks $M^2 = S^2 \cap S^3$ in time period 2:

$$M^2 = \{10, 11, 12, 13\}$$

Mine those blocks:

$$M^3 = S^3 \cap (\overline{S^1 U S^2}) = \{14, 15\}$$

Figure 33 gives the periods in which each block is to be mined.

Figure 33

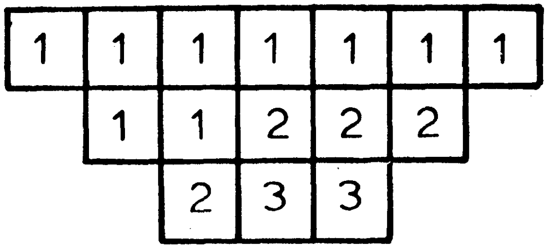


Figure 33. Time Periods in Which Individual Blocks to be Mined

CHAPTER 8

THE APPLICATION OF THE PROPOSED METHOD

8.1 INTRODUCTION

The open pit mine production scheduling problem as defined and formulated in this thesis is difficult to solve by the direct application of standard algorithms. The primary reason for this difficulty being the immense size of the problem. Therefore, the problem is decomposed into a smaller and more structured problem by relaxing difficult side constraints consisting of the blending and capacity requirements of the mining system. The procedure of relieving side constraints by the lagrangian relaxation techniques is discussed in Chapters 5 and 6. The relaxed lagrangian problem is then solved by further decomposing it into a series of equivalent single time period ultimate pit limit problems. This procedure was discussed as Chapter 7.

The overall strategy used in solving the production scheduling problem is therefore an iterative approach. At a given iteration one determines values for a set of multipliers assigned to the side constraints. The determination of multipliers in a given iteration are based on how much the constraints are violated in the previous iteration. Based on this new set of multipliers, the coefficients of the objective function to the lagrangian are modified and the multi time period sequencing problem is solved. The constraint violations are checked again to determine a new set of multipliers. This process is continued until convergence to the optimum solution is achieved. (In the case where gaps exist, the algorithm stops with an optimum solution to the modified constraint requirement). An overall flow chart of the algorithm is shown in Figure 34.

8.2 THE COMPUTER PROGRAMS FOR THE MULTI TIME PERIOD SCHEDULING

In order to apply the algorithm developed, one computer program was written in FORTRAN 77 and executed on a Digital Equipment Corporation VAX 11-750 model computer using the VMS operating system. This program is included in Appendix A. One of the subprograms performs multiplier adjustments based on the subgradient optimization algorithm of Held and Karp (1972). Another of the subprograms reads the initial multipliers estimates block grades, determines the block values, adjusts the objective function according to the multipliers, and performs the steps of the multi time period

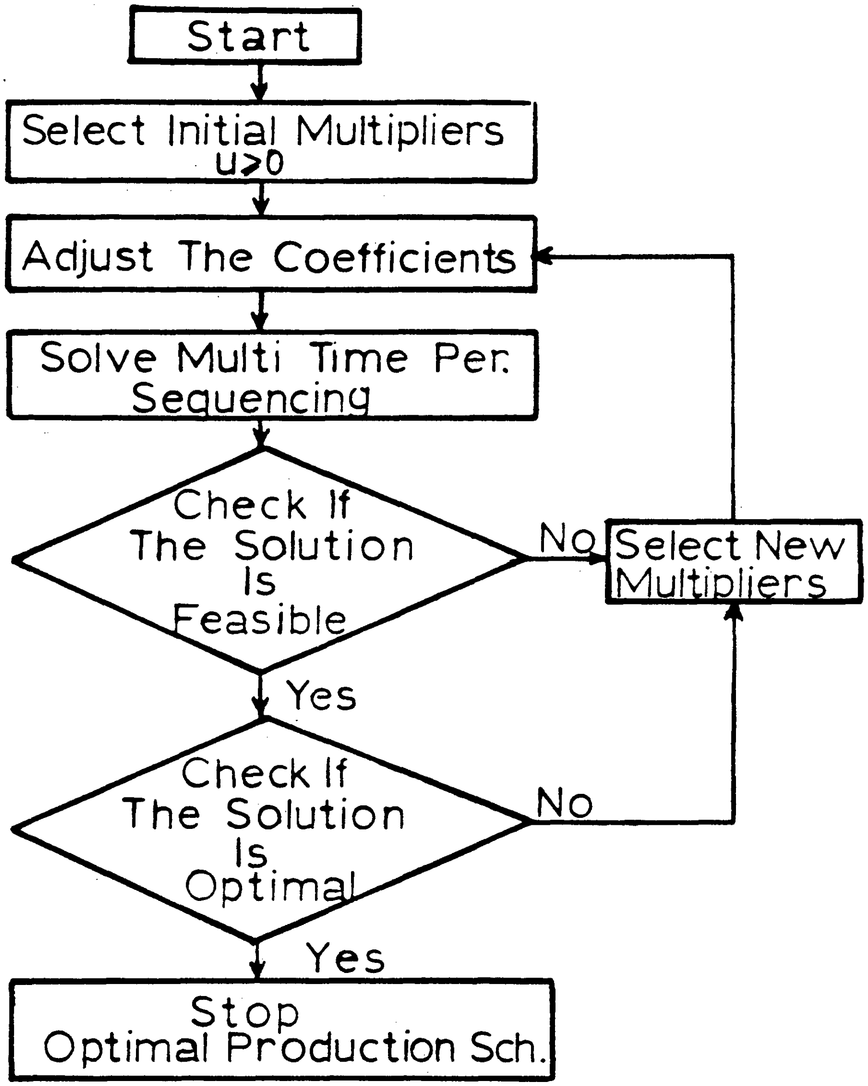

Start
Select Initial Multipliers u>0
Adjust The Coefficients
Solve Multi Time Per. Sequencing
Check If The Solution Is Feasible
No
Select New Multipliers
Yes
Check If The Solution Is Optimal
No
Yes
Stop Optimal Production Sch.

Figure 34. Logical Flow Chart of the Algorithm

sequencing algorithm. An important feature of this program is that it includes the Lerchs and Grossmann (1965) three-dimensional tree algorithm to solve the single time period ultimate pit limit problem.

The scheduling programs were written in a form conducive to direct reading of the code. The system was designed in a modular form, and a large number of comments were included to explain the purpose of the particular segment.

While the code in this scheduling system was written in a user oriented fashion, it was also written for a specific case on a specific computer. As a result these programs can not be directly generalized nor immediately applied to different circumstances without going through modification. Modifications are needed in the areas of dimension statements and their subsequent usage in the Do loop statements. The program is currently dimensioned such that the total number of blocks in the model must not exceed 5400 and the scheduling periods for any given time is limited to 10. The input-output routines also need to be modified to fit the specific conditions.

8.3 PHYSICAL EXAMPLE

8.3.1 The Model

To demonstrate and test the algorithms developed in this investigation, a small high grade copper deposit was selected.

The block model to which the computer programs were applied contained 5400 blocks each measuring 100 ft in length and width and 45 ft in height. The geologic block model showing block grades (% Cu) in benches 1 through 6 of this deposit is given in Appendix B.

The level maps of the economic block model showing original and discounted block values (in dollars) for different time periods are given in Appendix C. The economic assumptions made in the development of this economic block model were:

Price of Copper	:	\$ .70 per lb.
Mining Costs	:	\$ .85 per ton of material
Milling Costs	:	\$2.20 per ton of ore
Smelting and Marketing	:	\$ .25 per lb of Cu
G & A	:	\$ .15 per ton of ore
Discount Rate	:	12.5%

The ultimate pit limit contour obtained by using Lerchs and Grossmann's 3-D algorithm based on a 45 degree slope constraint is given in Appendix D. The total mineral reserve as indicated by the ultimate pit contour has a value of \$57,663,750* (profit before tax). This reserve includes 8,416,666 cubic yards (505 blocks) of waste and 2,362,500 tons (63 blocks) of ore with an average grade of 3.7% Cu (see Table 4).

Table 4. Level by Level Statistics of Reserves in the Ultimate Pit Limit Contour

Level	Number of Ore Blks	Ave. Grade	Number of Waste Blks
1	1	6.1	196
2	9	2.9	133
3	10	3.8	95
4	13	4.3	55
5	16	3.5	25
6	~~14~~ 15	~~3.6~~ 3.4	0
Total	63	3.7	505

* (153,770¢ given by the program is calculated by assuming each block weighs a ton. Since in a typical copper operation the tonnage factor is around 12.5, and as a result a block on average will contain 37,500 tons of material, the real value of the ultimate pit is 57,663,750.)

8.3.2 SCHEDULING EXAMPLES

As a first example of the production scheduling methodology developed in this thesis, the deposit described in the previous section will be scheduled to be mined over the time horizon of three years. The restrictions on the mining system are only on ore mining capacity. Specifically, the constraints are:

1. Mine 19 ore blocks in period 1
2. Mine 21 ore blocks in period 2
3. Mine 23 ore blocks in period 3
4. Satisfy the precedence and slope constraints as described in section 3.

The problem is now to schedule the operation such that the above restrictions are not violated and, at the same time, the total discounted profits before tax are maximized.

The optimum schedule obtained after nine iterations by using discounted block values is given in Figures 35 through 40. The numbers in these figures indicate the period in which each block is scheduled to be mined.

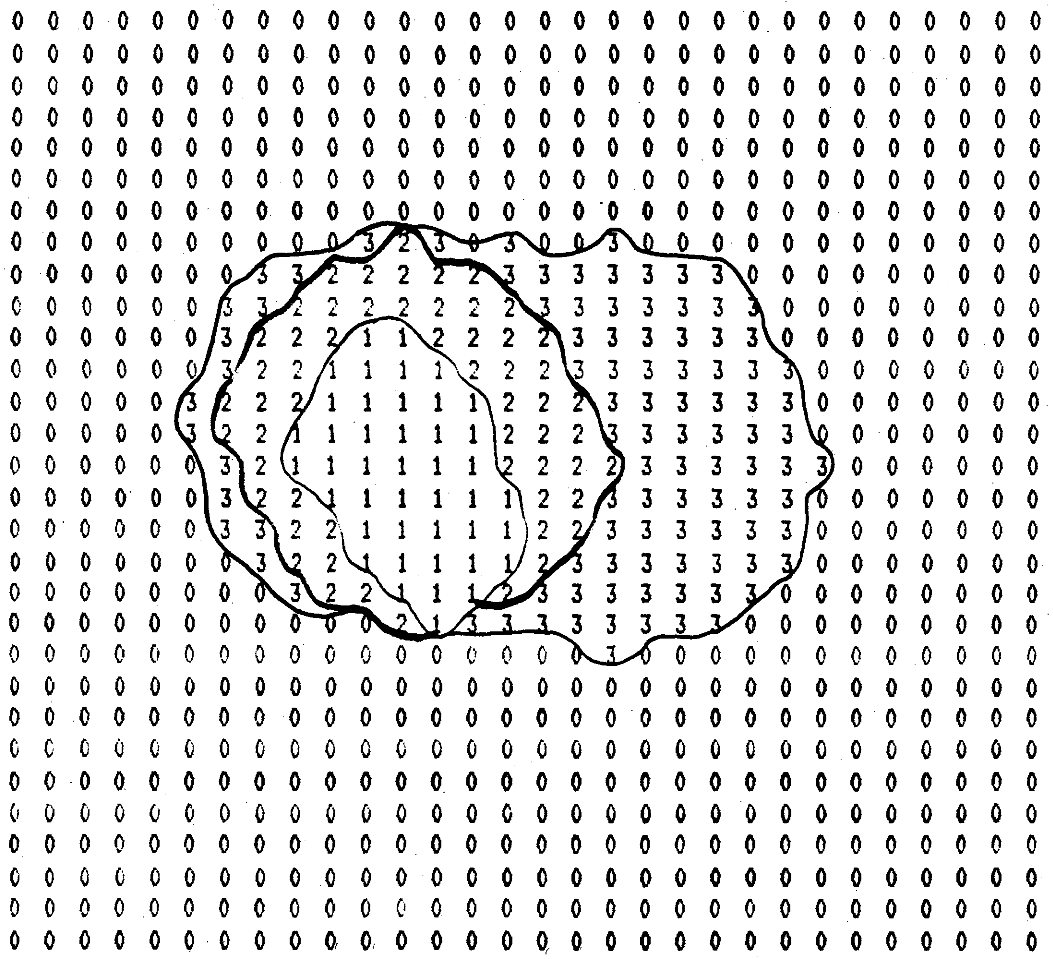
The summary statistics given in Figure 36 and Table 5 show that the net present value of this schedule is

Table 5. Level by Level Statistics of the Optimum Schedule Found for Example 1.

	Time Period 1			Time Period 2			Time Period 3		
Level	Ore Blks	Ave. Grade	Waste Blks	Ore Blks	Ave. Grade	Waste Blks	Ore Blks	Ave. Grade	Waste Blks
1	1	6.1	42	0	0	50	0	0	104
2	9	2.9	15	0	0	44	0	0	75
3	9	4.2	0	1	0.6	31	0	0	64
4	0	0	0	10	4.2	13	3	4.6	42
5	0	0	0	10	3.3	0	6	3.7	25
6	0	0	0	0	0	0	14	3.6	0
Total	19	3.7	57	21	4.4	146	23	3.0	302

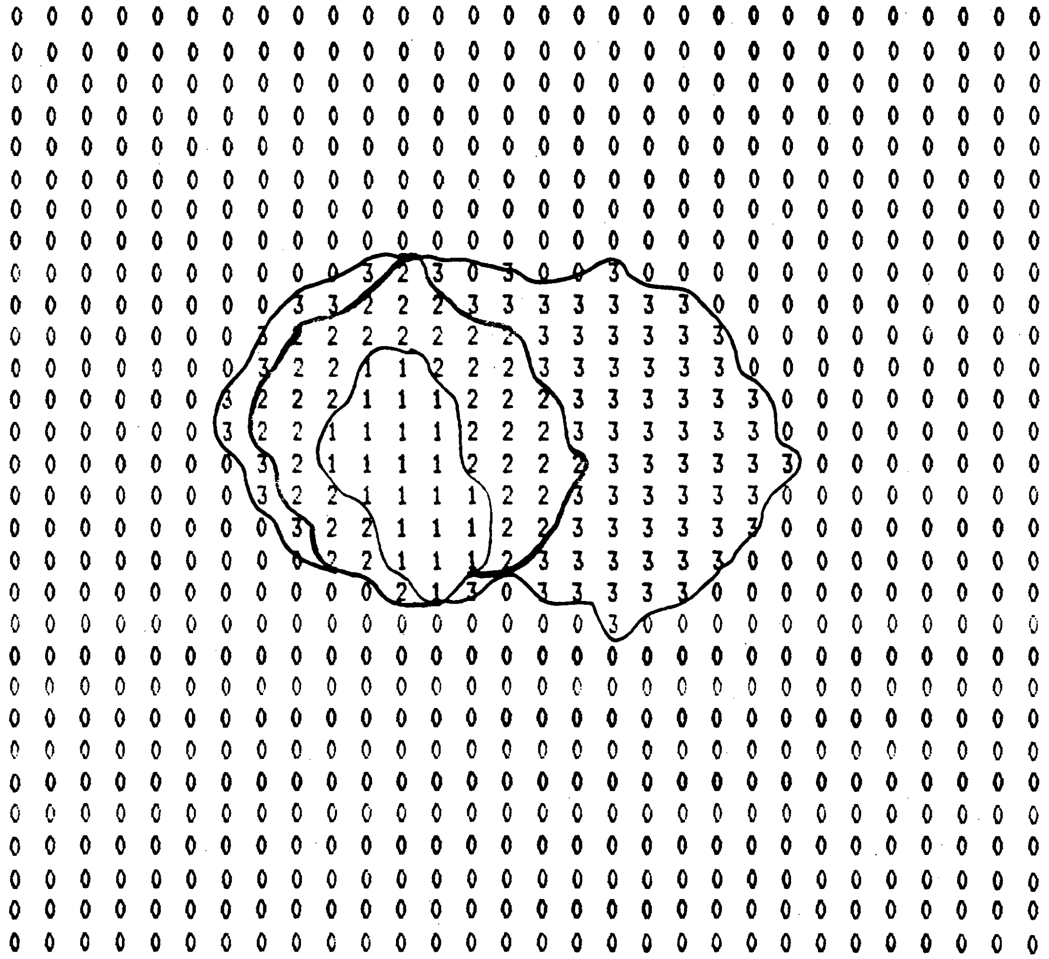
```
TY FOR041.DAT

            NPVT OF THE SCHEDULE:  139389
                        TOTAL TONS      ORE TONS      AVERA GRADE
            PERIOD1         57              19              3.7
            PERIOD2        146              21              4.4
            PERIOD3        302              23              3.0
$ TY FOR043.DAT
```



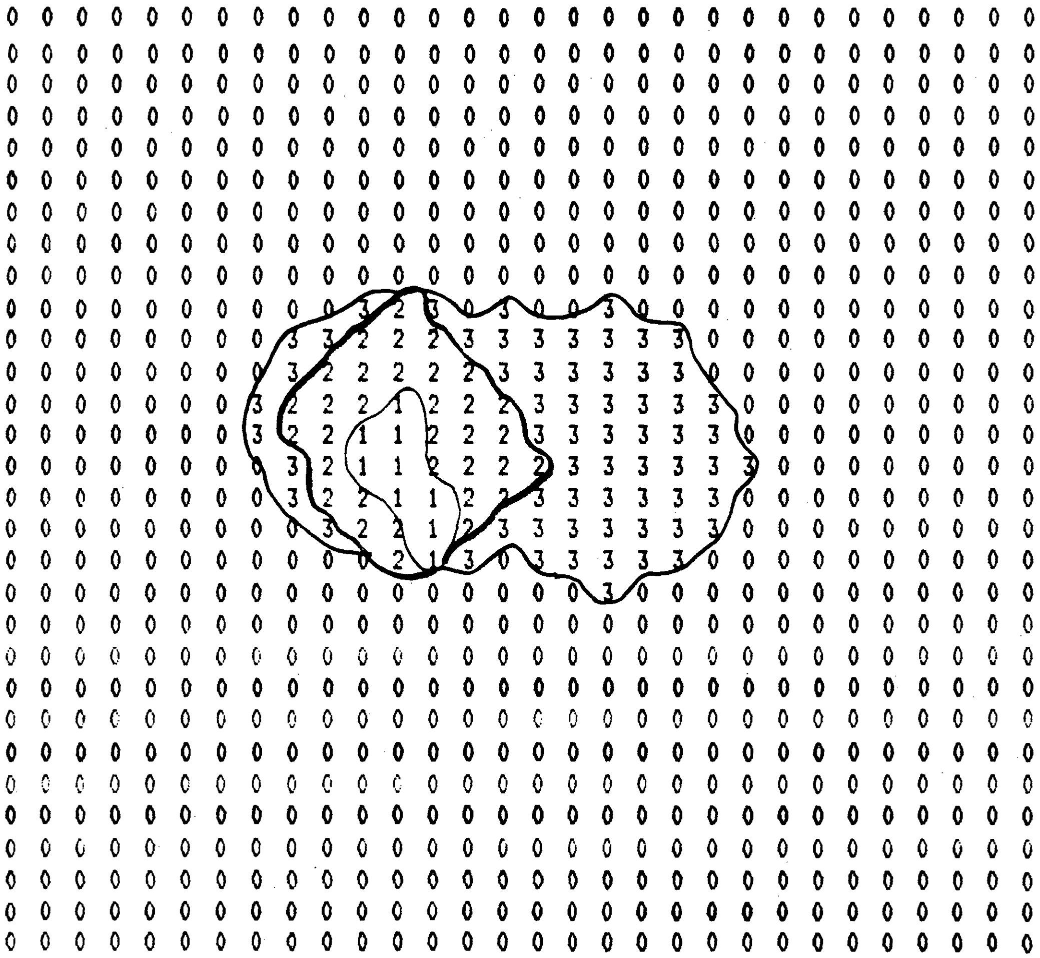
```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 3 2 3 0 3 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 3 3 2 2 2 2 2 3 3 3 3 3 3 3 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 3 3 2 2 2 2 2 2 2 3 3 3 3 3 3 3 0 0 0 0 0 0 0 0
0 0 0 0 0 0 3 2 2 2 1 1 2 2 2 2 3 3 3 3 3 3 0 0 0 0 0 0 0 0
0 0 0 0 0 0 3 2 2 1 1 1 1 2 2 2 3 3 3 3 3 3 3 0 0 0 0 0 0 0
0 0 0 0 0 3 2 2 2 1 1 1 1 1 2 2 2 3 3 3 3 3 3 0 0 0 0 0 0 0
0 0 0 0 0 3 2 2 1 1 1 1 1 1 2 2 2 3 3 3 3 3 3 0 0 0 0 0 0 0
0 0 0 0 0 0 3 2 1 1 1 1 1 1 2 2 2 2 3 3 3 3 3 3 0 0 0 0 0 0
0 0 0 0 0 0 3 2 2 1 1 1 1 1 1 2 2 3 3 3 3 3 3 0 0 0 0 0 0 0
0 0 0 0 0 0 3 3 2 2 1 1 1 1 1 2 2 3 3 3 3 3 3 0 0 0 0 0 0 0
0 0 0 0 0 0 0 3 2 2 1 1 1 1 1 2 3 3 3 3 3 3 3 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 3 2 2 1 1 1 2 3 3 3 3 3 3 3 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 2 1 3 3 3 3 3 3 3 3 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Figure 35. Summary Statistics and Level 1 of the Optimum Schedule for Example 1.



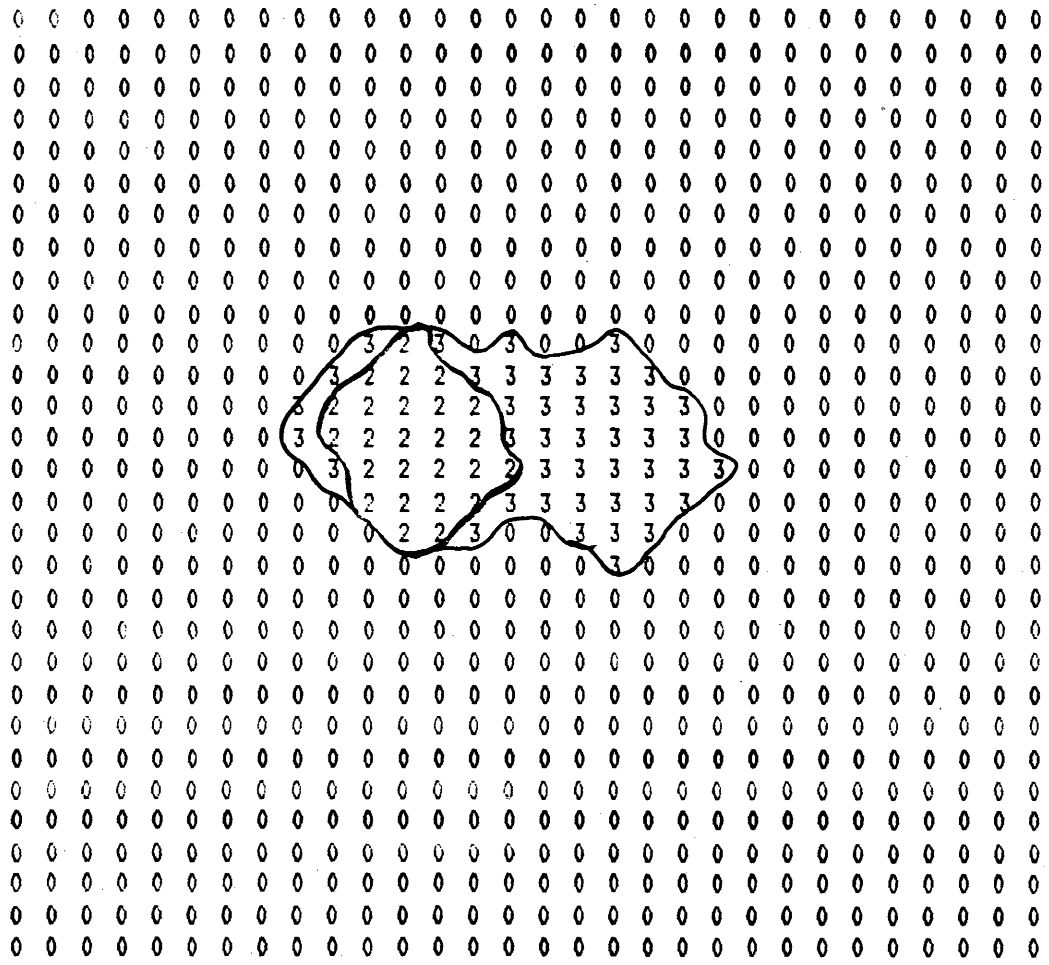
```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 3 2 3 0 3 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 3 3 2 2 2 3 3 3 3 3 3 3 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 3 2 2 2 2 2 2 2 3 3 3 3 3 3 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 3 2 2 1 1 2 2 2 3 3 3 3 3 3 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 3 2 2 2 1 1 1 2 2 2 3 3 3 3 3 3 0 0 0 0 0 0 0 0
0 0 0 0 0 0 3 2 2 1 1 1 1 2 2 2 3 3 3 3 3 3 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 3 2 1 1 1 1 2 2 2 2 3 3 3 3 3 3 0 0 0 0 0 0 0
0 0 0 0 0 0 0 3 2 2 1 1 1 1 2 2 3 3 3 3 3 3 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 3 2 2 1 1 1 2 2 3 3 3 3 3 3 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 2 2 1 1 1 2 3 3 3 3 3 3 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 2 1 3 0 3 3 3 3 3 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Figure 36. Level 2 of the Optimum Schedule for Example 1.



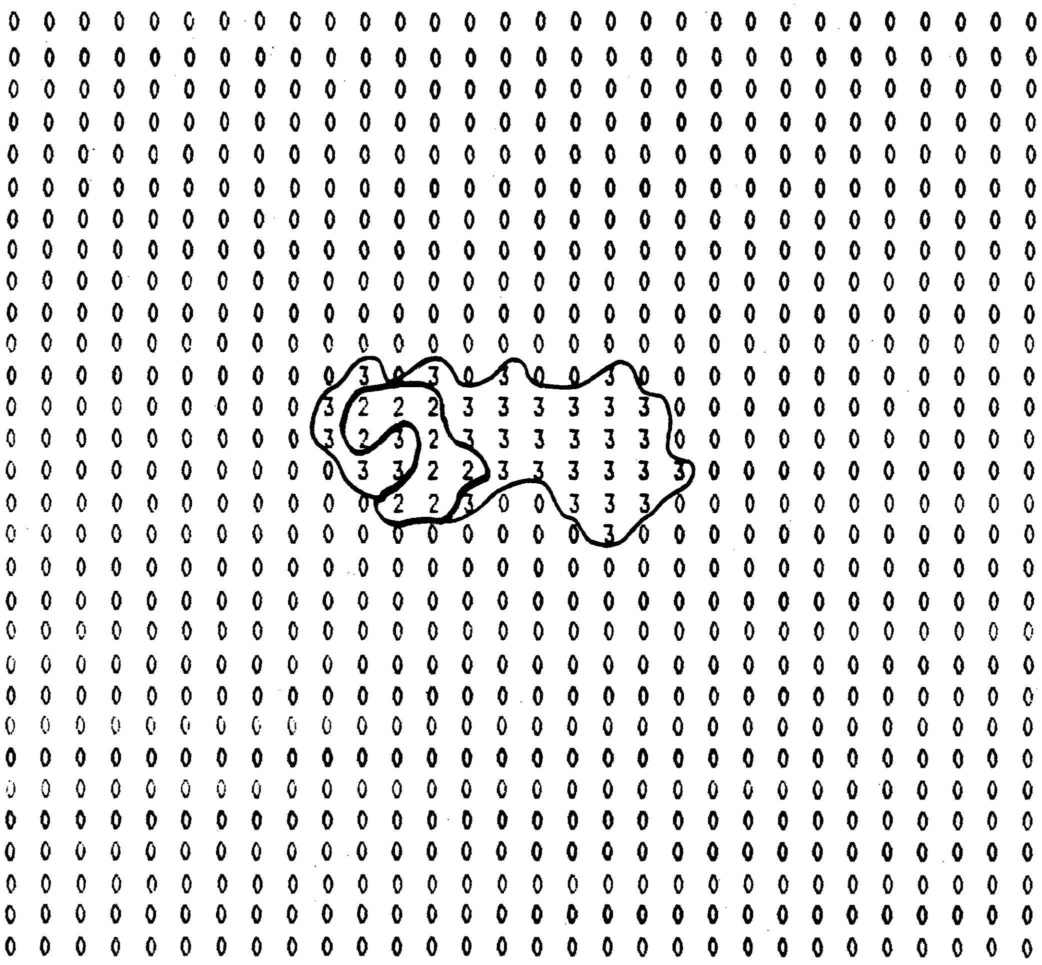
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	3	2	3	0	3	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	3	3	2	2	2	3	3	3	3	3	3	3	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	3	2	2	2	2	2	3	3	3	3	3	3	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	3	2	2	2	1	2	2	2	3	3	3	3	3	3	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	3	2	2	1	1	2	2	2	3	3	3	3	3	3	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	3	2	1	1	2	2	2	2	3	3	3	3	3	3	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	3	2	2	1	1	2	2	3	3	3	3	3	3	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	3	2	2	1	2	3	3	3	3	3	3	3	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	2	1	3	0	3	3	3	3	3	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 37. Level 3 of the Optimum Schedule for Example 1.



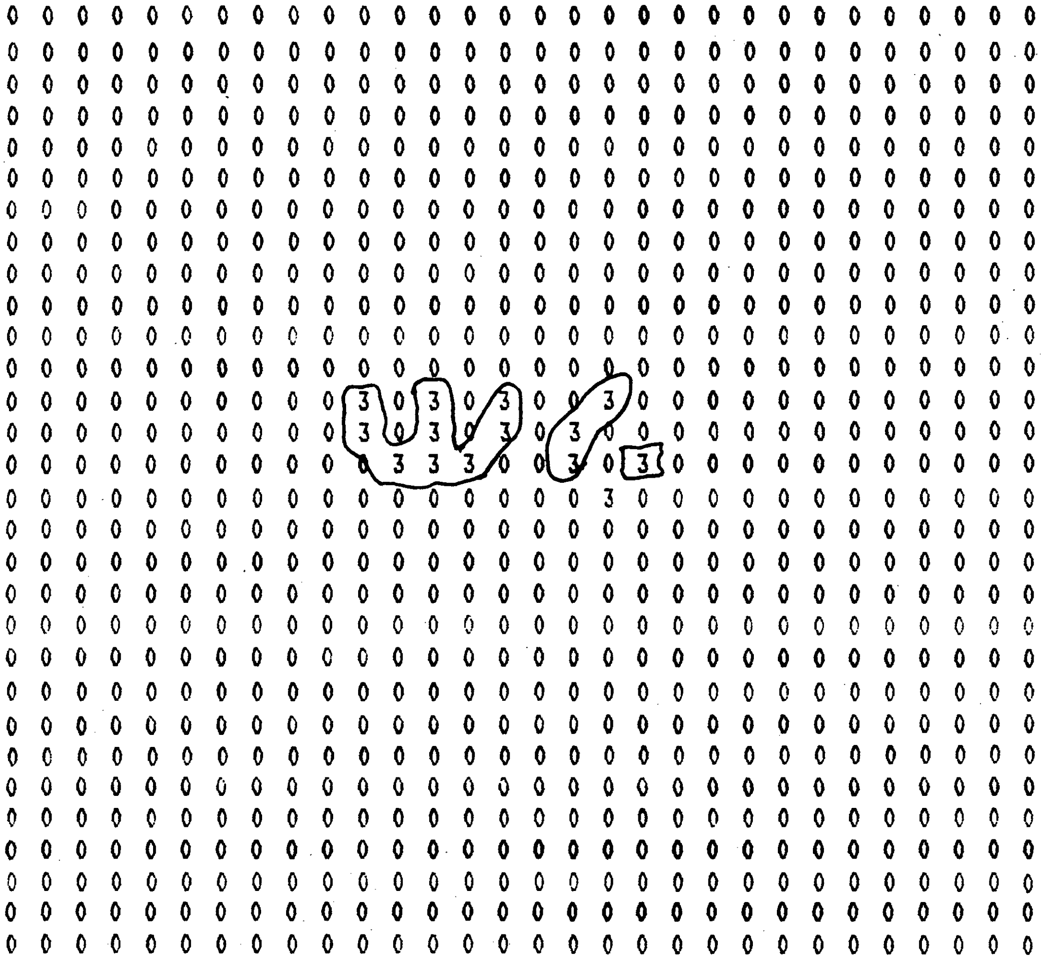
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	3	2	3	0	3	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	3	2	2	2	3	3	3	3	3	3	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	3	2	2	2	2	2	3	3	3	3	3	3	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	3	2	2	2	2	2	3	3	3	3	3	3	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	3	2	2	2	2	2	3	3	3	3	3	3	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	2	2	2	2	3	3	3	3	3	3	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	2	2	3	0	0	3	3	3	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 38. Level 4 of the Optimum Schedule for Example 1.



```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 3 0 3 0 3 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 3 2 2 2 3 3 3 3 3 3 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 3 2 3 2 3 3 3 3 3 3 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 3 3 2 2 3 3 3 3 3 3 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 2 2 3 0 0 3 3 3 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Figure 39. Level 5 of the Optimum Schedule for Example 1.



```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 3 0 3 0 3 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 3 0 3 0 3 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 3 3 3 0 0 3 0 3 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Figure 40. Level 6 of the Optimum Schedule for Example 1.

$52,270,875. Notice that this value is less than the total profits of $57,663,750 obtained for the ultimate pit. Imposing ore tonnage constraints on the system together with discounting reduced the total cash flows by $5,930,625.

The average grade of the ore mined in each period starts at 3.7% Cu in the first year, goes up to 4.4% Cu in the second year and decreases to 3.0% Cu in year 3. This schedule indicates that mining commences in a high grade area and progresses toward the low grade area. The required stripping to be done in each period shows a pattern of gradual increase. According to the schedule, very little stripping is required in time period 1; the amount of stripping is more then doubled in period 2; and most of the stripping is done in period 3 (see Table 5).

In order to illustrate to a greater extent the effects and costs of the system constraints on cash flows, a second example was generated by imposing further restriction on the example model. For this example, an additional restriction on mill feed grade was imposed together with the ore mining capacity constraints of the previous example; specifically:

1. Mine 19 ore blocks averaging 3.7% Cu in period 1.
2. Mine 21 ore blocks averaging 3.7% Cu in period 2.
3. Mine 23 ore blocks averaging 3.7% Cu in period 3.

As indicated in the discussion of the ultimate pit reserves, the average grade of the total reserve within the ultimate pit contour was 3.7% Cu. Hence, the second restriction on each period is to force the operation to mine as close to this average grade as possible.

The optimum solution to the above mining system is depicted in Figures 41 - 46. The numbers on different levels again indicate the time periods in which the block will be mined.

From the summary statistics given in Figure 41 and Table 6, total discounted profits from this schedule is $51,733,125, and all the constraints for different time periods are satisfied except the average grade requirement in period 2. Average grade of the schedule for this period is 3.6% as compared to required 3.7%. The most likely reason for this lower grade is a lack of availability of the blocks in the deposit to give the average grade of 3.7%.

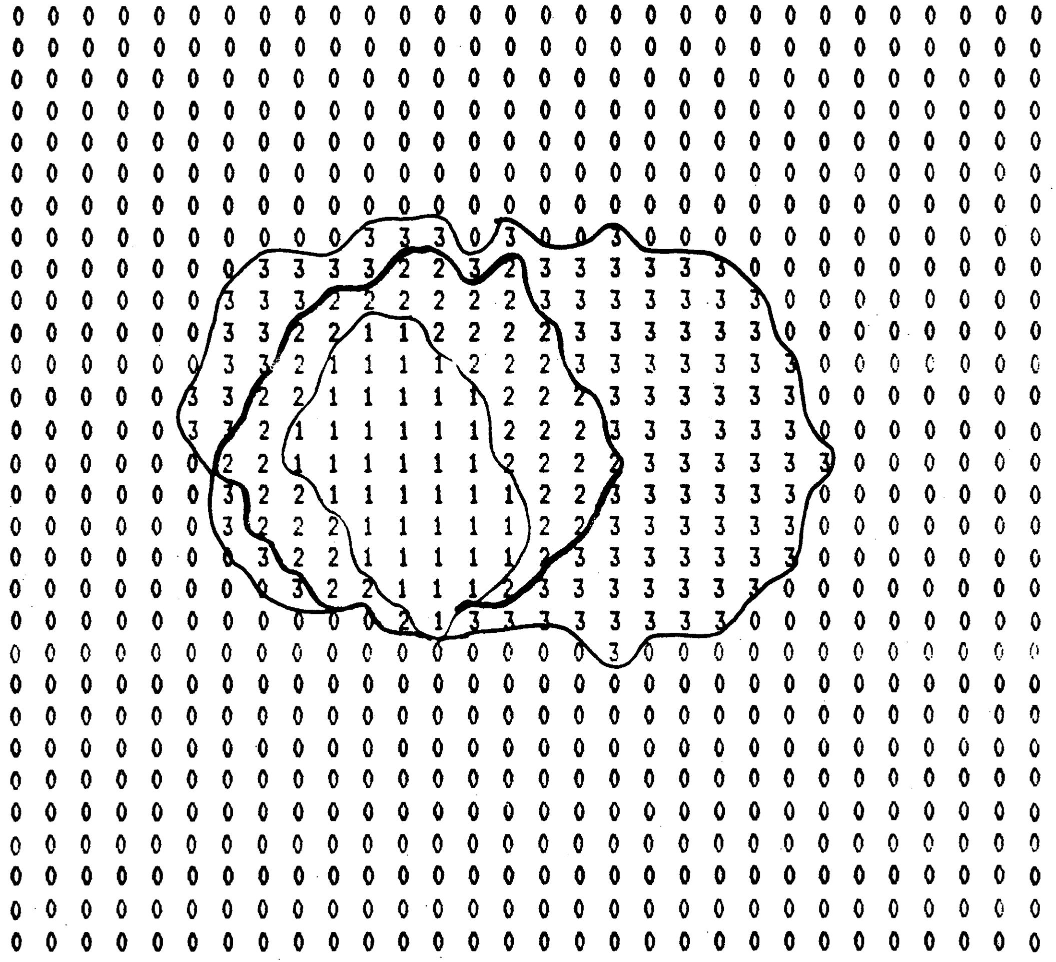
The total discounted value of the schedule for this case is less than the total discounted cash flow obtained for the previous example. That is, because of the additional restrictions on the mill feed grade, the total discounted profits are further reduced $537,750. As can be seen, this last reduction is caused by blending some of the low grade material with high grade in period 2. By

Table 6. Level by Level Statistics of the Optimum Schedule Found for Example 2.

	Time Period 1			Time Period 2			Time Period 3		
Level	Ore Blks	Ave. Grade	Waste Blks	Ore Blks	Ave. Grade	Waste Blks	Ore Blks	Ave. Grade	Waste Blks
1	1	6.1	42	0	0	50	0	0	104
2	9	2.9	15	0	0	44	0	0	75
3	9	4.2	0	1	0.6	31	0	0	64
4	0	0	0	10	4.2	13	3	4.6	42
5	0	0	0	10	3.3	0	6	3.7	25
6	0	0	0	0	0	0	14	3.6	0
Total	19	3.7	57	21	3.6	138	23	3.7	310

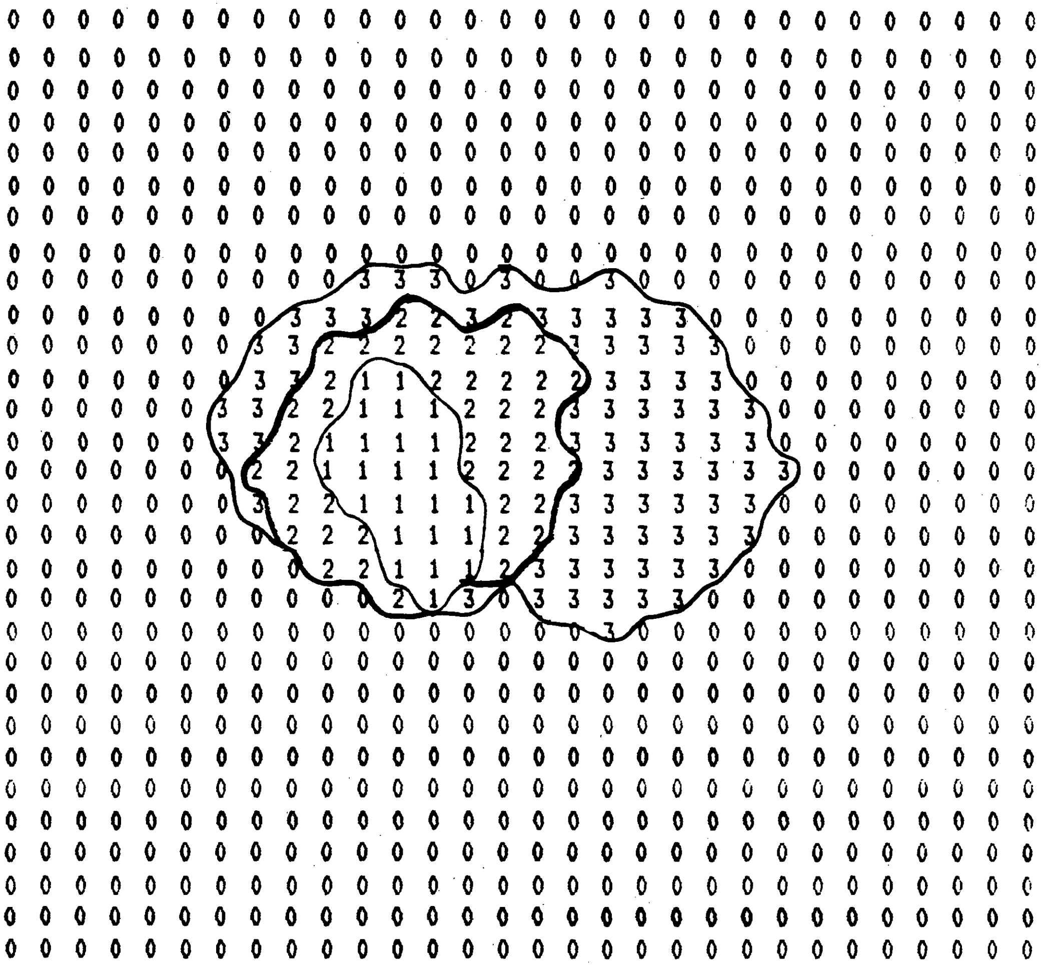
```
TY FOR041.DAT


        NPVT OF THE SCHEDULE:  137955
                    TOTAL TONS       ORE TONS      AVERA GRADE
        PERIOD1        57               19               3.7
        PERIOD2       138               21               3.6
        PERIOD3       310               23               3.7
$ TY FOR043.DAT
```



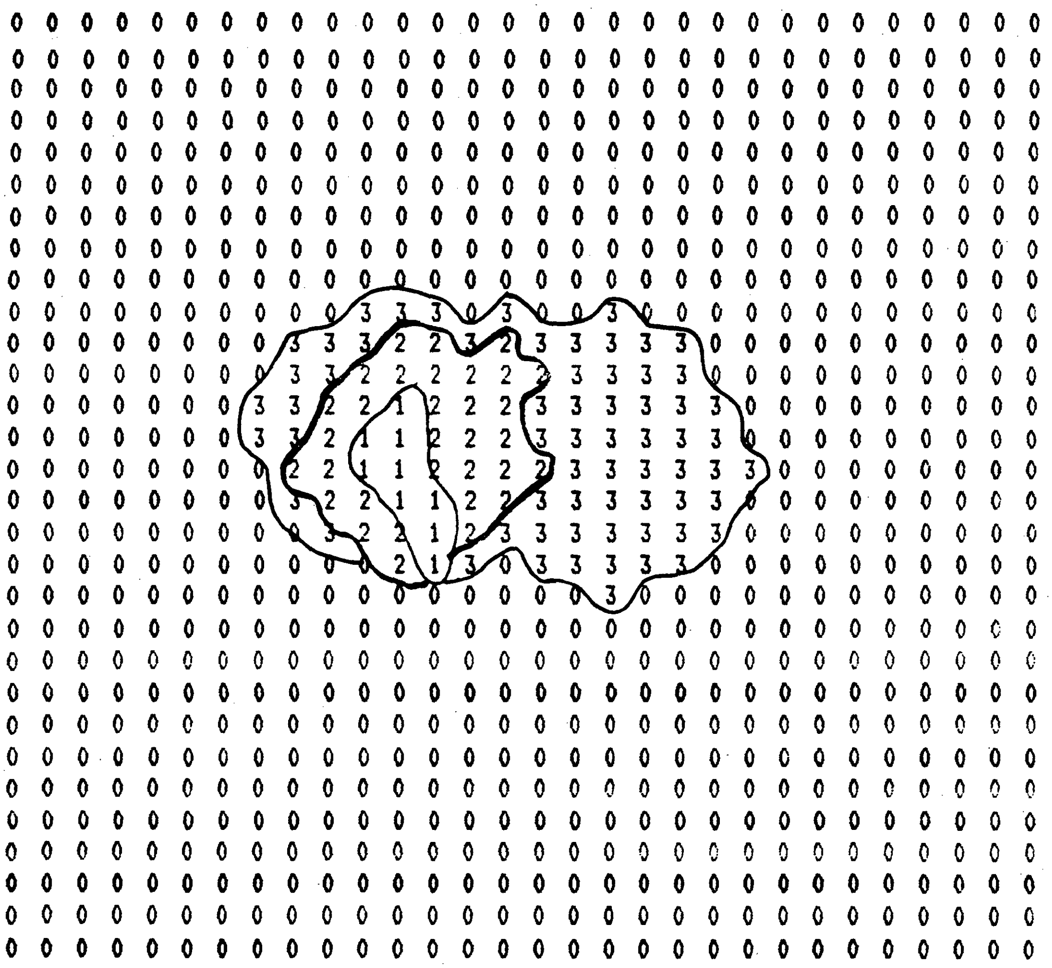
```
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 3 3 3 0 3 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 3 3 3 3 2 2 3 2 3 3 3 3 3 3 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 3 3 3 2 2 2 2 2 2 3 3 3 3 3 3 3 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 3 3 2 2 1 1 2 2 2 2 3 3 3 3 3 3 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 3 3 2 1 1 1 1 2 2 2 3 3 3 3 3 3 3 0 0 0 0 0 0 0
 0 0 0 0 0 3 3 2 2 1 1 1 1 1 2 2 2 3 3 3 3 3 3 0 0 0 0 0 0 0
 0 0 0 0 0 3 3 2 1 1 1 1 1 1 2 2 2 3 3 3 3 3 3 0 0 0 0 0 0 0
 0 0 0 0 0 0 2 2 1 1 1 1 1 1 2 2 2 2 3 3 3 3 3 3 0 0 0 0 0 0
 0 0 0 0 0 0 3 2 2 1 1 1 1 1 1 2 2 3 3 3 3 3 3 0 0 0 0 0 0 0
 0 0 0 0 0 0 3 2 2 2 1 1 1 1 1 2 2 3 3 3 3 3 3 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 3 2 2 1 1 1 1 1 2 3 3 3 3 3 3 3 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 3 2 2 1 1 1 2 3 3 3 3 3 3 3 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 2 1 3 3 3 3 3 3 3 3 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Figure 41. Summary Statistics and Level of the Optimum Schedule for Example 2.



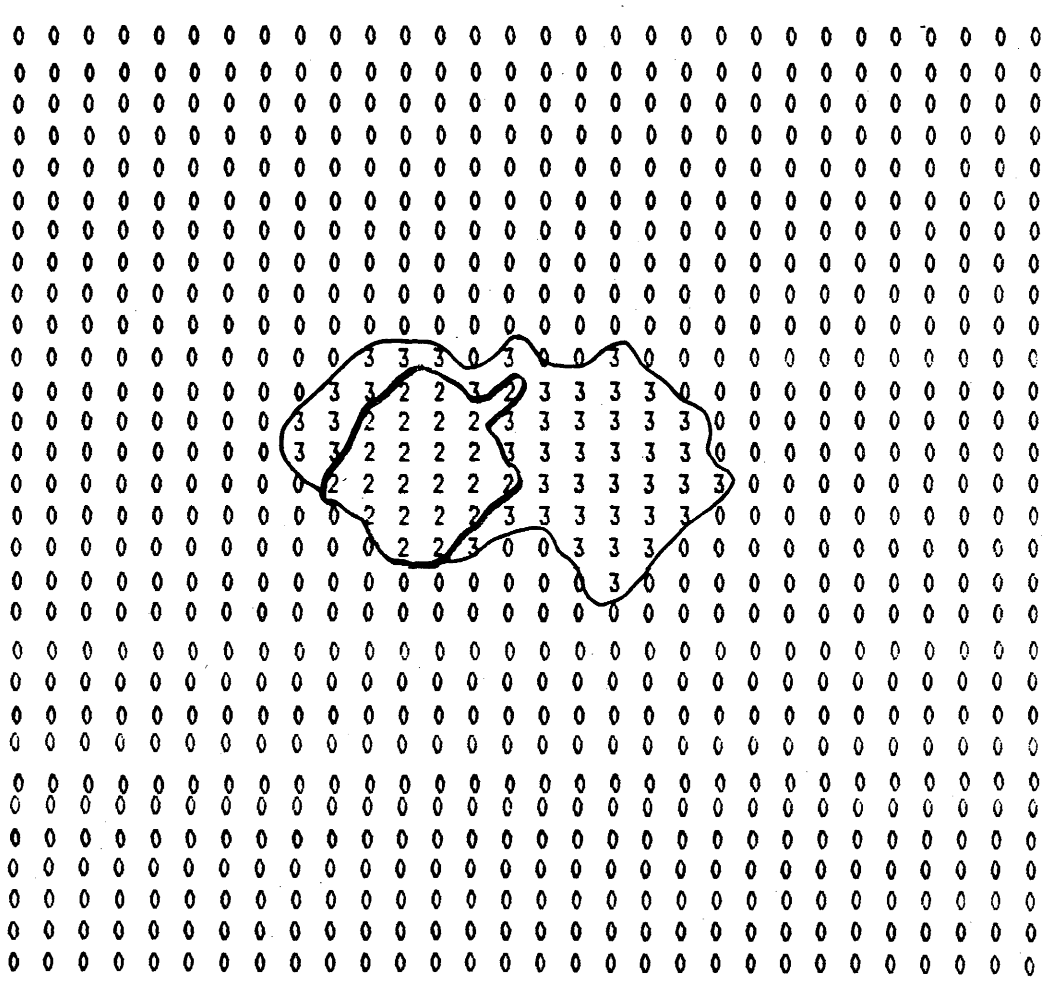
```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 3 3 3 0 3 0 0 3 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 3 3 3 2 2 3 2 3 3 3 3 3 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 3 3 2 2 2 2 2 2 2 3 3 3 3 3 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 3 3 2 1 1 2 2 2 2 2 3 3 3 3 0 0 0 0 0 0 0 0
0 0 0 0 0 0 3 3 2 2 1 1 1 2 2 2 3 3 3 3 3 3 0 0 0 0 0 0 0
0 0 0 0 0 0 3 3 2 1 1 1 1 2 2 2 3 3 3 3 3 3 0 0 0 0 0 0 0
0 0 0 0 0 0 0 2 2 1 1 1 1 2 2 2 2 3 3 3 3 3 3 0 0 0 0 0 0
0 0 0 0 0 0 0 3 2 2 1 1 1 1 2 2 3 3 3 3 3 3 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 2 2 2 1 1 1 2 2 3 3 3 3 3 3 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 2 2 1 1 1 2 3 3 3 3 3 3 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 2 1 3 0 3 3 3 3 3 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Figure 42. Level 2 of the Optimum Schedule for Example 2.



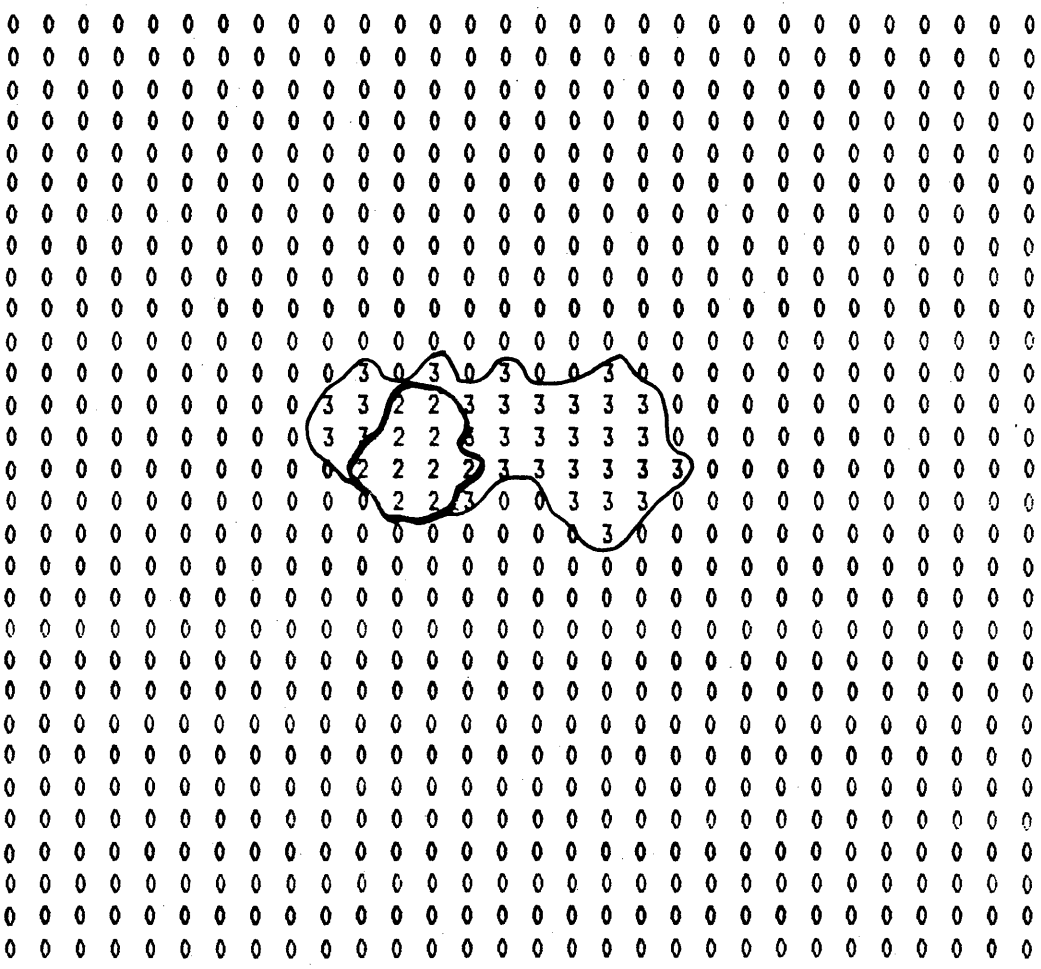
```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 3 3 3 0 3 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 3 3 3 2 2 3 2 3 3 3 3 3 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 3 3 2 2 2 2 2 2 3 3 3 3 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 3 3 2 2 1 2 2 2 3 3 3 3 3 3 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 3 3 2 1 1 2 2 2 3 3 3 3 3 3 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 2 2 1 1 2 2 2 2 3 3 3 3 3 3 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 3 2 2 1 1 2 2 3 3 3 3 3 3 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 3 2 2 1 2 3 3 3 3 3 3 3 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 2 1 3 0 3 3 3 3 3 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Figure 43. Level 3 of the Optimum Schedule for Example 2.



```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 3 3 3 0 3 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 3 3 2 2 3 2 3 3 3 3 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 3 3 2 2 2 2 3 3 3 3 3 3 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 3 3 2 2 2 2 3 3 3 3 3 3 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 2 2 2 2 2 2 3 3 3 3 3 3 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 2 2 2 2 3 3 3 3 3 3 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 2 2 3 0 0 3 3 3 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Figure 44. Level 4 of the Optimum Schedule for Example 2.



```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 3 0 3 0 3 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 3 3 2 2 3 3 3 3 3 3 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 3 3 2 2 3 3 3 3 3 3 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 2 2 2 2 3 3 3 3 3 3 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 2 2 3 0 0 3 3 3 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Figure 45. Level 5 of the Optimum Schedule for Example 2.

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 3 0 3 0 3 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 3 0 3 0 3 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 3 3 3 0 0 3 0 3 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

$

Figure 46. Level 6 of the Optimum Schedule for Example 2.

the average grade requirement, high grading of the deposit is prevented. Although the reduction of revenues by mill feed grade requirements is not significant in this example, the results interestingly reinforce the common economic notion that high grading gives higher discounted profits than a schedule which blends high with low grades.

These example studies can be expanded to include other conditions to determine effects and costs of different constraints on the system. Clearly, with the proposed methods, many different mining scenarios can be evaluated.

CHAPTER 9

CONCLUSIONS AND RECOMMENDATIONS

9.1 CONCLUSIONS

Sound mine planning should contribute greatly to converting a deposit into an economically profitable mining operation. If the corporate objective is the attainment of maximum net present value, scientific decision making tools similar to those proposed in this dissertation need to be substituted for present mine planning practices.

A number of significant contributions are made in this dissertation. These are:

1. A clear, concise and structured definition and formalization of open pit production scheduling problems are given.
2. The most important stumbling block in solving the production scheduling problem by mathematical optimization tools has been the multi-period sequencing problem. In this investigation this problem is solved and a very efficient algorithm is presented.

3. One of the significant developments in open pit planning in the 1970's was the concept of parametrization. The concept of parametrization is formalized in terms of the lagrangian theory of mathematics and, as a result, is enriched into a broader, and more useful form.
4. A new interpretation of cutoff grades is developed which gives a framework for understanding the dynamic nature of cutoff grades.

In addition, numerous benefits will be gained from the application of the proposed method. These can be summarized as:

1. Application of the model and the optimum mine production scheduling algorithm presented in the previous chapters will not only provide an open pit mine plan which maximizes the return on investment, it will also yield other benefits in providing answers to many "what if" questions. Having a model similar to one developed here, and the necessary computer programs, the sensitivity of mine economics to various mining and plant capabilities can be carried out rapidly by varying right hand side

requirements. This process of analysis; when combined with optimum production schedules, can be of great interest to management in evaluation and optimization of the complete mining system.

2. The mining industry is known for its fluctuating market conditions. By utilizing the model developed here, it is possible to perform sensitivity analysis with respect to various economic factors such as metal prices and costs.
3. The procedure presented in this study may also affect the traditional cutoff grade concept used in production scheduling. The production scheduling obtained as a result of the model will not require a cutoff grade decide what is ore and what is waste. This decision is automatically made by the model.
4. The other advantages of the methodology presented here will be in efficient utilization of planning engineer's time. With this method, the engineer is freed from doing tedious year to year calculations by hand. He is put in a position where he can view the entire planning horizon through the model, and consider other possibilities such as the interaction of demands of yearly schedules on one another.

In conclusion, this research study was undertaken with the objective of bringing the decision maker closer to making decisions which result in more profitable operations. This objective can be attained only when the proposed method is accepted by the mining industry as an effective tool and applied to the scheduling of numerous mining properties.

9.2 RECOMMENDATIONS FOR FUTURE RESEARCH

The model and solution algorithm developed in this investigation made considerable progress in obtaining a production schedule which will maximize the net present value of the operation. Yet more work should be done in a number of areas concerning the production scheduling problem.

The modification of lagrange multipliers, as discussed in Chapter 6, needs to be improved upon. This step of the overall algorithm may be improved by further investigating the relationship between lagrange multipliers and the cutoff grades. This interesting area deserves more study and can have powerful implications.

Another area for further research is the investigation into the "condition of gaps". This area is actively being researched in the domain of mathematics. This research needs to be followed closely, and later combined with the existing knowledge of the production scheduling problem. Only through the continuing interest in both operations research and its application to mine production scheduling can this problem be totally solved.

REFERENCES

1. Barnes, R.J., and Johnson T.B., 1980, "New Ideas in the Optimum Ultimate Pit Limit Problem: Bounding The Optimum", Proceedings of the Applications of Computers in Mining Symposium, Moscow, U.S.S.R., October 1980.

2. Barnes, R.J., and Johnson, T.B., 1982, "Bounding Techniques for the Ultimate Pit Limit Problem", 17th APCOM, Colorado School of Mines, Golden, Colorado, April 1982.

3. Barnes, R.J., "Optimizing the Ultimate Pit", Master of Science Thesis, Colorado School of Mines, Golden, Colorado, May 1982.

4. Bazaraa, M.S., Shetty, C.M., 1979, Nonlinear Programming, John Wiley & Sons, New York, 1979.

5. Benham, D., 1978, "Open Pit Production Scheduling", M.Sc. Thesis, Colorado School of Mines, Golden, Colorado 1978.

6. Blackwell, M.R.L., 1971, "Some Aspects of the Evaluation and Planning of the Bougainville Copper Project", Proceedings of Decision-Making in the minerals industry symposium, Montreal, Canada, Published by C.I.M. as C.I.M. Special Vol. no. 12, pp. 261-269.

7. Braticevic, D.B., 1983, "Open Pit Optimization Method", 18th APCOM, Institute of Mining and Metallurgy, London, England, March 1984.

8. Brackebush, F.W., 1971, "To design Lakeshore Open Pit Hecla Thinks Big With Small Computer", EMJ - July 1971.

9. Carlson, T.R., J.P. Erickson, D.T. O'Brian, et al., 1966, "Computer Techniques in Mine Planning", Mining Engineering, v. 18, n. 5, May 1966, pp. 53-56, 80.

10. Cousins, T.R., 1979, "Aspects of Production Planning: Operating Layout and Phase Plans", in Open Pit Mine Planning and Design, Crawford and Hustrulid (eds.), SME-AIME, New York, 1979.

11. Carlisle, D., 1954, "The Economics of a Fun Resource With Particular Reference to Mining", American Economic Rev., 44, pp. 575-616, Sept. 1955.

12. Chen, T., 1976, "3-D Pit Design With Variable Wall Slope Capabilities", 14th Apcom, Pennsylvania State University, State College, Pennsylvania, October 1976, p. 615-625.

13. Crawford, John T., 1979, "Open Pit Limit Analysis - Some Observations On Its Use", 16th APCOM, University of Arizona, Tucson, Arizona, October 1979, pp. 625-634.

14. Crawford, J.T., and R.K. Davey, 1979, "Case Study in Open Pit Limit Analysis", Computer Methods for the 80's in the Mineral Industry, Alfred Weiss (ed.), SME-AIME, pp. 310-318.

15. Dagdelen, K., Francois-Bongarcon, D., 1982, "Towards the Double Parametrization of Reserves", 17th APCOM, Golden, Colorado, Oct. 1982.

16. David, M., 1975, Geostatistical Ore Reserve Estimation, Elsevier Scientific Publishing Company, Amsterdam, 364 p.

17. Davis, R.E., and Williams, C.E., 1973, "Optimization Procedures for Open Pit Mine Scheduling", 11th APCOM, University of Arizona, Tucson, Arizona, Vol. 1C., pp. 1-18, 1973.

18. Decker, J., Garg, O.P., Setele, J.E., 1984, "Production Planning System at the Hanna Mining Company", SME-AIME Fall Meeting preprint 84-372, October 1984.

19. Douglas, E.J., 1971, "How to Make the Most of a Mining Investment", Mining Engineering, Oct. 1971, pp. 64-67.

20. Dantzig, G.B., 1963, Linear Programming and Extensions, Princeton University Press, Princeton, New Jersey.

21. Dantzig, G.B. and Wolfe, P., 1960, "Decomposition Principal for Linear Programs", Operations Research, Vol. 8, No. 1, Jan. 1960.

22. Erickson, J.D., 1968, "Long Range Open Pit Mine Planning", Mining Engineering, Vol. 20, No. 4, April 1968, pp. 75-78.

23. Etcheberry, J., 1977, "The Set Covering Problem: A new Implicit Enumeration Algorithm", Operations Research, Vol. 25, pp. 760-772, 1977.

24. Everett, H., 1963, "Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources", Operations Res., Vol. II, pp. 399-417, 1963.

25. Fairfield, J.D., and R.W. Leigh, 1968, "A Computer Program for the Design of Open Pits", 7th APCOM, Golden, Colorado, April 1968, Colorado School of Mines Quarterly, Vol. 64, No. 3.

26. Fisher, M.L., and Shapiro, J.F., 1974, "Constructive Duality In Integer Programming", SIAM J. Appl. Math., Vol. 27, pp. 31-52, 1974.

27. Fisher, M.L., Northup, W.D., and Shapiro, J.F., 1975, "Using Duality to Solve Discrete Optimization

Problems: Theory and Computational Experience", Math. Programming Study, Vol. 3, pp. 56-94, 1975.

28. Ford, L.R. and D.R. Fulkerson, 1962, Flows in Networks, Princeton University Press, Princeton, New Jersey.

29. Fox, B.L., 1978, "Data Structures and Computer Science Techniques in Operations Research", Operations Research, Vol. 26, No. 5, Sept. 1978.

30. Francois-Bongarcon, D., 1978, "Le Parametrage Des Contours Optimaux D'uve Exploitation a Ciel Ouvert", Ph.D. Thesis, Institut National Polytechnique De lorraine, France, April 1978.

31. Francois-Bongarcon, D., and Marechal, A., 1976, "A New Method for Open Pit Design: Parametrization of the Final Pit Contour", 14th APCOM AIME, New York, pp. 573-583.

32. Gangwar, A., 1973, "A General Binary Stochastic Model for Optimum Pit Production Scheduling", Ph.D. Dissertation, School of Engineering and Applied Science, Columbia University, New York, 1973.

33. Gauthier, F.J., Gray, K.G., 1971, "Pit Design by Computer at Gaspe Copper Mines, Limited", CIM Bulletin, Nov. 1971, pp. 95-102.

34. Geoffrion, A.M., 1966, "Integer Programing by Implicit Enumeration and Balas Method", Rand Corp. RM-4783-RP, Santa Monica, California, Feb. 1966.

35. Geoffrion, A.M., 1974, "Lagrangian Relaxation and its Uses in Integer Programming, "Math. Programming Study, Vol 2, pp. 82-114, 1974.

36. Gerschon, M., 1982, "A Linear Programming Approach to Mine Scheduling Optimization", 17th APCOM, Johnson and Barnes (eds.), Colorado School of Mines, Golden, Colorado, Apr. 1982.

37. Gilbert, J.W., 1966, "A Mathematical Model for the Optimal Design of Open Pit Mines", M.Sc. Thesis, Department of Industrial Engineering, University of Toronto, Canada, Sept. 1966.

38. Gomory, R.E., 1967, "All Integer Programming Algorithm", Industrial Scheduling, J.F. Math and G.L. Thompson eds., Prentice Hall, New Jersey, pp. 193-206, 1967.

39. Grinald, R.C., 1970, "Lagrangian Subgradients", Management Sci. 17, pp. 185-188, 1970.

40. Halls, J.L., Bellum, D.P., and Lewis, C.K., 1969, "Determination of Optimum Ore Reserves and Plant Size by Incremental Financial Analysis", Transactions of Institute of Mining and Metallurgy, Sect. A: Mining Industry, Vol. 78, 1969, p. A25.

41. Hartman, R.J., and G.C. Varma, 1966, "3D OPT - A Three Dimensional Optimum Pit Program and a Basis for a Mining Engineering System", 6th APCOM, Pennsylvania State University, State College, Pennsylvania, Apr. 1966.

42. Held, M. and Karp, R.M., 1970, "The Traveling Salesman Problem and Minimum Spanning Trees", Operations Research, Vol. 18, pp. 1138-1162, 1970.

43. Held, M., and Karp, R.M., 1971, "The Traveling Salesman Problem and Minimum Spanning Trees: Part II", Mathematical Programming, Vol. 1, pp. 6-25, 1971.

44. Held, M., Wolfe, P. and Crowder, H.D., 1974, "Validation of Subgradient Optimization", Mathematical Programming, Vol. 6, pp. 62-88.

45. Jensen, P.A., Barnes, J.W., 1980, Network Flow Programing, John Wiley & Sons, New York, 1980.

46. Johnson, T.B., 1968, "Optimum Open-Pit Mine Production Scheduling", Ph.D. Thesis, University of California, Berkeley, Operations Research Department, May 1968, 120 p.

47. Johnson, T.B., 1973, "A Comparative Study of Methods for Determining Ultimate Open-Pit Mining Limits", 11th APCOM, University of Arizona, Tucson, Arizona, Apr. 1973.

48. Johnson, T.B., and D.G. Mickle, 1970, "Optimum Design of an Open Pit, an Application in Uranium", 9th APCOM,

Montreal, Canada, June 1970, CIM Special Volume 12, pp. 331-337.

49. Johnson, T.B., and W.R. Sharp, 1971, "A Three-Dimensional Dynamic Programming Method for Optimal Open Pit Design", U.S.B.M. RI 7553, 1971.

50. Journal, A.G., and Huijbregts, C.J., 1978, Mining Geostatistics, Academic Press, New York, 1978.

51. Kim, Y.C., 1978, "Ultimate Pit Limit Design Methodologies Using Computer Models - the State of the Art", Mining Engineering, Vol. 30, pp. 1454-1459.

52. Kim, Y.C., 1979, "Production Scheduling: Technical Overview", Computer Methods for the 80's in the Mineral Industry, Alfred Weiss (ed.), SME-AIME, 1979, pp. 297-303.

53. Korobov, Sergei, 1974, "Method for Determining Optimal Open Pit Limits", Rappart Technique ED 74-R-4, Department of Mineral Engineering Ecole Polytechnique Montreal, Canada, Feb. 1974.

54. Koskiniemi, Benjamin C., 1979, "Hand Methods, Open Pit Mine Planning and Design", J.T. Craford and W.A. Hustrulid (ed.), SME-AIME, pp. 187-195.

55. Lane, K.E., 1964, "Choosing the Optimum Cutoff Grade", Colorado School of Mines Quarterly, Vol. 59, 1964, pp. 811-829.

56. Lasdon, L.S., 1970, *Optimization Theory For Large Systems*, MacMillan, New York, 1970.

57. Lawler, E.L., and Bell, M.D., 1966, "A Method for Solving Discrete Optimization Problems", Operations Research, Vol. 15, pp. 892-914, 1967.

58. Lemieux, M., 1979, "Moving Cone Optimization Algorithm", *Computer Methods for the 80's in the Mineral Industry*, Alfred Weiss (ed.), SME-AIME, 1979, pp. 329-345.

59. Lemke, C.E. and Spielberg, K., 1967, "Direct Search Algorithms for Zero-One and Mixed Integer Programming", Operations Research, Vol. 15, pp. 892-914, 1967.

60. Lerchs, H., and Grossmann, I.F., 1965, "Optimum Design of Open Pit Mines", C.M.I. Bulletin, Vol. 58, No. 633, Jan. 1965, pp. 47-54.

61. Larson, E.R., and Wells, C.H., (?), "Optimum Open Pit Mine Scheduling", Technical Note OXD-70-1, Systems Control, Inc., Palo Alto, California 94306.

62. Lewis, C.K., 1969, "An 'Economic Life' for Property Evaluation", EIMJ, Oct. 1969, pp. 78-79.

63. Lilico, T.M., 1973, "How to Maximize Return on Capital When Planning Open Pit Mines", World Mining, June 1973, pp. 22-27.

64. Lipkewich, M.P., and Borgman, L., 1969, "Two and Three Dimension Pit Design Optimization Technique, A Decade in Digital Computing in the Mineral Industry, Alfred Weiss (ed.), SME-AIME, 1969, pp. 505-523.

65. Marino, J.M. and J.P. Slama, 1973, "Ore Reserve Evaluation and Open Pit Planning", 10th APCOM, Johannesburg, Republic of South Africa, April 1973.

66. Mathieson, G.A., 1982, "Open Pit Sequencing and Scheduling", Preprint #82-368, SME-AIME Fall Meeting, Honolulu, Hawaii, September 1982.

67. Mason, P.M., 1984, "Capital and Operational Planning for Open-Pits in a Modern Economy", 18th APCOM, London, England, Mar. 1984, pp. 791-801.

68. Milner, T.E., 1977, "Long Range Mine Planning at Gibraltar Mines Limited", C.I.M. Bulleting, Oct. 1977, p. 89-94.

69. Muckstadt, J., and Koeniz, S.A., 1977, "An Application of Lagrangian Relaxation to Scheduling in Power Generation Systems", Operations Res., Vol. 25, pp. 387-403.

70. Murty, Katta, 1979, Linear and Combinatorial Programming, John Wiley & Sons, New York, 1979.

71. Pana, M.J., 1965, "Simulation Approach to Open Pit Design", 5th APCOM, Tucson, Arizona, Mar. 1965.

72. Pana, M.T., Carlson, T.R., 1966, "A Description of a Computer Technique Used in Mine Planning of the Utah Mine of Kennecott Copper Corp.", 6th APCOM, State College, Pennsylania, April 1966.

73. Phillips, D.A., 1973, "Optimum Design of and Open Pit", 10th APCOM, Johannesburg, Republic of South Africa, Apr. 1973.

74. Phillips, Don T., A Ravindran, and James J. Solberg, 1976, Operations Research, New York, John Wiley & Sons Inc.

75. Robinson, R.H., 1975, "Programming the Lerchs Grossman Algorithm for Open Pit Design", 13th APCOM, Clausthal, Federal Republic of Germany, Oct. 1975.

76. Robinson, R.H., and W.B. Prenn, 1973, "An Open Pit Design Model", 10th APCOM, Johannesburg, Republic of South Africa, Apr. 1973.

77. Roman, Ronald J., 1974, "The Role of Time Value of Money in Determining an Open Pit Mining Sequence and Pit Limits", 12th APCOM, Golden, Colorado, April 1974.

78. Ross-Brown, D.M., 1979, "Analytical Design", Open Pit Mine Planning and Design, J.T. Crawford and W.A. Hustrulid, (ed.), SME-AIME, pp. 161-184.

79. Rychkum, E.A. and T. Chen, 1979, "Open Pit Mine Feasibility Method and Application at Placer Development", Computer Methods for the 80's in the Mineral Industry, Alfred Weiss (ed.), SME-AIME, 1979, pp. 304-309.

80. Shapiro, J.F., 1971, "Generalized Lagrange Multipliers in Integer Programming", Operations Res., Vol 19, pp. 68-76, 1971.

81. Shapiro, J.F., 1979, "A Survey of Lagrangian Techniques for Discrete Optimization", Ann. Discrete Math., Vol. 5, pp. 113-138, 1979.

82. Sims, D.E., 1979, "Open-Pit Long-Range Mine Planning Using O.R.E.", Computer Methods for the 80's in the Mineral Industry, Alfred Weiss (ed.), SME-AIME, 1979, pp. 359-370.

83. Soderberg, A., Rausch, D.O., 1968, "Pit Planning and Layout", Surface Mining", E.P. Pfleider, (ed.), SME-AIME, New York, 1968.

84. Stanley, B., 1979, "Mineral Model Construction: Principles of Ore-Body Modeling", Open Pit Mine Planning and Design, J.T. Crawford, and W.A. Hustrulid (eds.), SME-AIME, pp. 43-50.

85. Suriel, Roberto, 1984, "Preliminary Economic Evaluation of Open Pit Mines", Master of Science Thesis, Colorado School of Mines, Golden, Colorado, 1984.

86. Taylor, H.K., 1972, "General Background Theory of Cutoff Grades", Transactions of I.M.M., Sect. A, Vol. 1, pp. A160-179.

87. Vickers, E.L., 1961, "Marginal Analysis, Its Applications in Determining Cutoff Grade", Mining Engineering, Vol. 13, 1961, pp. 578-581.

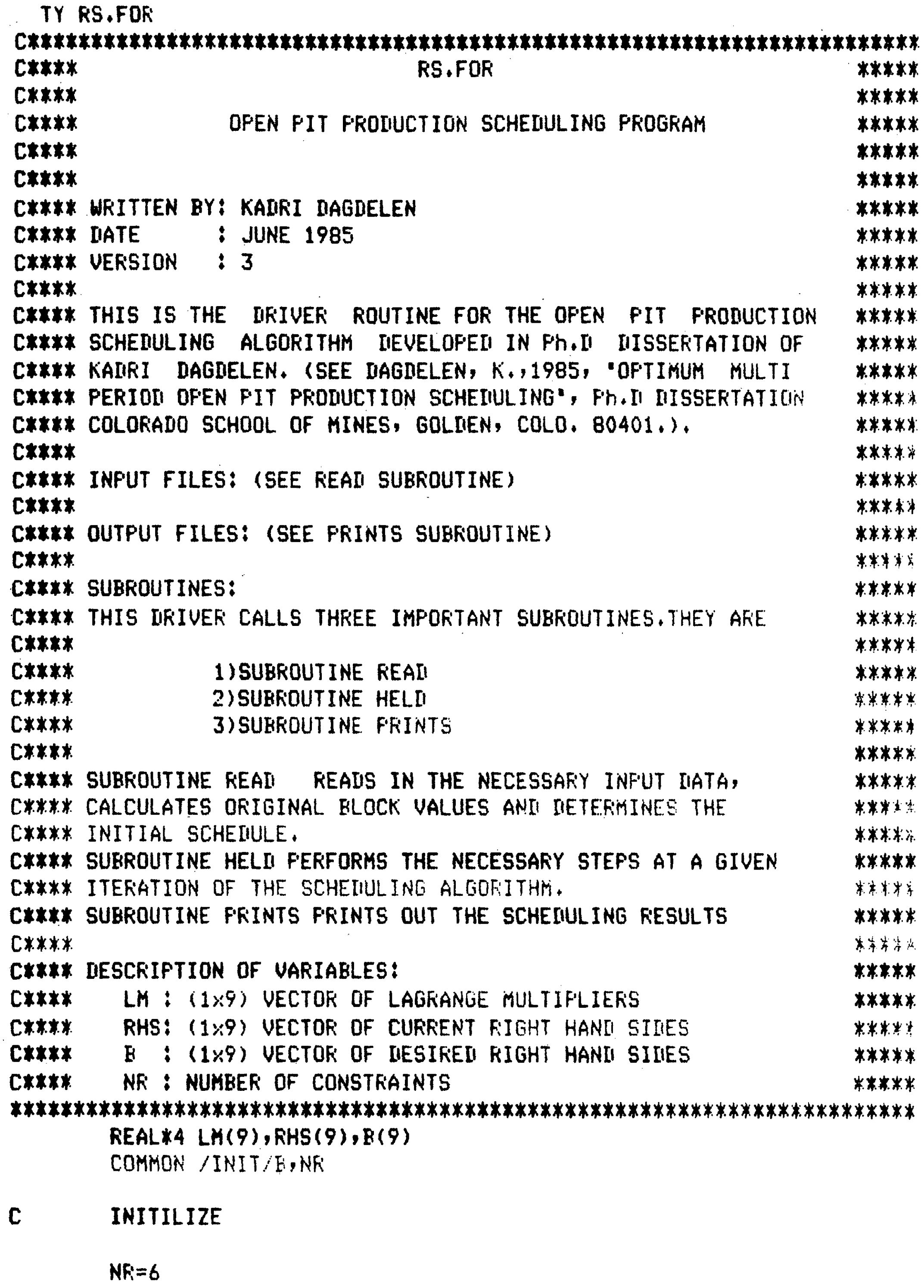
88. Wells, H., 1978, "Optimization of Mining Engineering Design in Mineral Valuation", Mining Engineering, December 1978, pp. 1576-1584.

89. Williams, C., 1974, "Computerized Year-By-Year Open-Pit Mine Scheduling", SME-AIME Transaction, Vol. 256, Dec. 1974, pp. 309-317.

90. Wilke, F.L., Mueller, K., Wright, E., "Ultimate Pit and Production Scheduling Optimization", 18th APCOM, Institution of Mining and Metallurgy, London, England, March 1984.

APPENDIX A

Listing of the Computer Codes



```
 TY RS.FOR
C*************************************************************************
C****                              RS.FOR                           *****
C****                                                               *****
C****            OPEN PIT PRODUCTION SCHEDULING PROGRAM             *****
C****                                                               *****
C****                                                               *****
C**** WRITTEN BY: KADRI DAGDELEN                                    *****
C**** DATE      : JUNE 1985                                         *****
C**** VERSION   : 3                                                 *****
C****                                                               *****
C**** THIS IS THE  DRIVER  ROUTINE FOR THE OPEN  PIT  PRODUCTION    *****
C**** SCHEDULING  ALGORITHM  DEVELOPED IN Ph.D  DISSERTATION OF     *****
C**** KADRI  DAGDELEN. (SEE DAGDELEN, K.,1985, 'OPTIMUM  MULTI      *****
C**** PERIOD OPEN PIT PRODUCTION SCHEDULING', Ph.D DISSERTATION     *****
C**** COLORADO SCHOOL OF MINES, GOLDEN, COLO. 80401.).              *****
C****                                                               *****
C**** INPUT FILES: (SEE READ SUBROUTINE)                            *****
C****                                                               *****
C**** OUTPUT FILES: (SEE PRINTS SUBROUTINE)                         *****
C****                                                               *****
C**** SUBROUTINES:                                                  *****
C**** THIS DRIVER CALLS THREE IMPORTANT SUBROUTINES.THEY ARE        *****
C****                                                               *****
C****            1)SUBROUTINE READ                                  *****
C****            2)SUBROUTINE HELD                                  *****
C****            3)SUBROUTINE PRINTS                                *****
C****                                                               *****
C**** SUBROUTINE READ   READS IN THE NECESSARY INPUT DATA,          *****
C**** CALCULATES ORIGINAL BLOCK VALUES AND DETERMINES THE           *****
C**** INITIAL SCHEDULE.                                             *****
C**** SUBROUTINE HELD PERFORMS THE NECESSARY STEPS AT A GIVEN       *****
C**** ITERATION OF THE SCHEDULING ALGORITHM.                        *****
C**** SUBROUTINE PRINTS PRINTS OUT THE SCHEDULING RESULTS           *****
C****                                                               *****
C**** DESCRIPTION OF VARIABLES:                                     *****
C****    LM : (1x9) VECTOR OF LAGRANGE MULTIPLIERS                  *****
C****    RHS: (1x9) VECTOR OF CURRENT RIGHT HAND SIDES              *****
C****    B  : (1x9) VECTOR OF DESIRED RIGHT HAND SIDES              *****
C****    NR : NUMBER OF CONSTRAINTS                                 *****
*************************************************************************
        REAL*4 LM(9),RHS(9),B(9)
        COMMON /INIT/B,NR

C       INITILIZE

        NR=6
```

```
C       INITILIZE THE LAMDA AND PHI MATRICES

        CALL READ(LM,RHS)

C       PERFORM THE STEPS OF THE ALGORITHM

        DO I=1,15
           CALL HELD(LM,RHS)
        ENDO

C       PRINT THE RESULTS AFTER DESIRED NUMBER OF ITERATIONS

        CALL PRINTS

        STOP
        END
```

```
C*************************************************************************
C****                                                                *****
C****                    SUBROUTINE READ                             *****
C****                                                                *****
C****  THIS SUBROUTINE PERFORMS THE INITIL STEPS OF THE OVERALL      *****
C****  PROGRAM. THIS ROUTINE READS IN THE INPUT DATA, DETERMINES     *****
C****  THE ORIGINAL BLOCK VALUES AND CALLS THE SCHEDULING            *****
C****  SUBROUTINE TO DETERMINE THE STARTING SCHEDULE.                *****
C****                                                                *****
C****  INPUT FILES :                                                 *****
C****  THE (30x30) ARRAY of  ESTIMATED BLOCK GRADES ARE INPUT TO     *****
C****  THE PROGRAM FROM A FILE NAMED 'FOR030.DAT'. THE FORMAT OF     *****
C****  THIS FILE IS: FORMAT(30(30F4.1,/),/).                         *****
C****                                                                *****
C****  THE (1x6) ARRAY OF THE INITIAL LAGRANGE MULTIPLIER VECTOR     *****
C****  IS INPUT FROM A FILE NAMED 'FOR025.DAT'. THE FORMAT OF        *****
C****  THIS FILE IS : FORMAT(F).                                     *****
C****                                                                *****
C****  THE (1x6) ARRAY OF THE DESIRED RIHGT HAND SIDES ARE INPUT     *****
C****  FROM A FILE NAMED 'FOR026.DAT'. THE FORMAT OF THIS FILE       *****
C****  IS : FORMAT(F).                                               *****
C****                                                                *****
C****  SUBROUTINES:                                                  *****
C****  THIS SUBROUTINE CALLS FOLLOWING  SUBROUTINES:                 *****
C****                                                                *****
C****           1)SUBROUTINE FVALUE                                  *****
C****           2)SUBROUTINE DOIT                                    *****
C****           3)SUBROUTINE PRINTS                                  *****
C****                                                                *****
C****  SUBROUTINE FVALUE CALCULATES THE ORIGINAL BLOCK VALUES ($).   *****
C****  SUBROUTINE DOIT FINDS THE INITIAL SCHEDULE BASED ON           *****
C****  THE CALCULATED BLOCK VALUES AND STARTING MULTIPLIERS.         *****
C****  SUBROUTINE PRINTS PRINTS OUT THIS INITIAL SCHEDULE.           *****
C****                                                                *****
C****  DESCRIPTION OF VARIBLES:                                      *****
C****  IC         : THE COUNTER FOR THE NUMBER OF ITERATIONS         *****
C****  LM(I)      : LAGRANGE MULTIPLIER FOR THE I'TH CONSTRAINT      *****
C****  MBT(I,J)   : ON-OFF INDEX OF BLOCK J IN PERIOD I              *****
C****  OVAL(I,J)  : ORIGINAL DOLLAR VALUE OF BLOCK J IN PERIOD I     *****
C****  RHS(I)     : THE CURRENT RIGHT HAND SIDE OF CONSTRAINT I      *****
C****  TTP        : NUMBER OF TIME PERIODS IN A SCHEDULING HORIZON   *****
C****  VAL(I,J)   : ADJUSTED DOLLAR VALUE OF BLOCK J IN PERIOD I     *****
C****  X(I)       : ESTIMATED GRADE OF BLOCK I                       *****
C****  WOI(I,J)   : ORE- WASTE INDICATOR OF BLOCK J IN PERIOD I      *****
C****  WASTE(I,J) : STRIPPING COST OF BLOCK J IN PERIOD I            *****
C****                                                                *****
C*************************************************************************
```

```
        SUBROUTINE READ(LM,RHS)
        REAL*4 LM(9),RHS(9),B(9)
        REAL*4 X(5400)

        INTEGER*2 TTP,MBT(0:5,5400),WOI(0:5,5400)
        INTEGER*4 VAL(0:5,5400),WASTE(0:5,5400),OVAL(5,5400)

        COMMON TTP,IC
        COMMON /BLK6/VAL,MBT,WOI
        COMMON /BLK8/X
        COMMON /BLK30/WASTE
        COMMON /OVAL/OVAL
        COMMON /INIT/B,NR

        NR=6
        TTP=3

        TYPE *,'INITIL'

C       READ IN ESTIMATED BLOCK GRADES

        READ(30,5)(X(I),I=1,5400)
5       FORMAT(30(30F4.1,/),/)

C       READ IN THE ORIGINAL MULTIPLIERS

        READ(25,10)(LM(I),I=1,NR)
10      FORMAT((F))

C       READ IN THE DESIRED RIGHT HAND SIDES

        READ(26,20)(B(I),I=1,NR)
20      FORMAT((F))

C       CALCULATE THE ORIGINAL BLOCK VALUES

        CALL PVALUE

C       CALL SCHEDULER

        CALL DOIT(LM,RHS)

C       WRITE OUT ORIGINAL PIT LIMITS

        CALL PRINTS

        RETURN
        END
```

```
C***********************************************************************
C****                                                              *****
C****                   SUBROUTINE PVALUE                          *****
C****                                                              *****
C****   THIS SUBROUTINE DETERMINES THE BLOCK VALUES ACCORDING TO  *****
C****   SOME ASSUMED COSTS AND PRICES.                             *****
C****   THE FOLLOWING ECONOMIC ASSUMPTIONS ARE MADE IN CALCULATING *****
C****   BLOCK VALUES :                                             *****
C****           PRICE            :  .70 $/Lb. OF CU.               *****
C****           MINING COST      :  .85 $/TON OF MATERIAL.         *****
C****           MILLING COST     : 2.20 $/TON OF ORE.              *****
C****           SMELTING & MARK. :  .25 $/LB OF CU.                *****
C****           ADMINISTRATIVE   :  .15 $/TON OF ORE.              *****
C****           RECOVERY         :  100 %                          *****
C****                                                              *****
C****   INPUT FILES: NONE                                          *****
C****                                                              *****
C****   SUBROUTINES: NONE                                          *****
C****                                                              *****
C****   DESCRIPTION OF VARIABLES: (SEE SUBROUTINE READ)            *****
C***********************************************************************
        SUBROUTINE PVALUE
        LOGICAL*1 READI
        INTEGER*2 TTP,MBT(0:5,5400),WOI(0:5,5400)
        INTEGER*4 VAL(0:5,5400),WASTE(0:5,5400),OVAL(5,5400)
        REAL*4 X(5400)
        COMMON TTP,IC
        COMMON /BLK6/VAL,MBT,WOI
        COMMON /BLK8/X
        COMMON /BLK30/WASTE
        COMMON /OVAL/OVAL
        READI=.TRUE.

C       INITIALIZE TIME THE BLOCKS WILL BE MINED TO 0
!
!
                DO J=0,TTP
                        DO I=1,5400
                                MBT(J,I)=0
                                WOI(J,I)=1
                        ENDDO
                ENDDO
!
!
!
!
                DO K=1,6
                        DO I=1,30
```

c G2A

```
                           DO J=1,30
                             N=(K-1)*900+(I-1)*30+J
                             VAL(0,N)=X(N)*20*(70-25)-65-220-15
                             IF(X(N).LE.0)VAL(0,N)=-65-220-15
                             IF(K.EQ.1)WASTE(0,N)=-65
                             IF(K.EQ.2)WASTE(0,N)=-70
                             IF(K.EQ.3)WASTE(0,N)=-75
                             IF(K.EQ.4)WASTE(0,N)=-80
                             IF(K.EQ.5)WASTE(0,N)=-85
                             IF(K.EQ.6)WASTE(0,N)=-90
                           ENDDO
                        ENDDO
                ENDDO
!
C       DETERMINE UNMODIFIED BLOCK VALUES WITH RESPECT TO TIME PER.

                DO J=1,TTP
                        DO I=1,5400
                                IF(J.LE.1)THEN
                                        VAL(J,I)=VAL(J-1,I)
                                        WASTE(J,I)=WASTE(J-1,I)
                                ELSE
                                        VAL(J,I)=VAL(J-1,I)/1.125
                                        WASTE(J,I)=WASTE(J-1,I)/1.125
                                ENDIF
                        ENDDO
                ENDDO

        DO KK=1,TTP
           DO I=1,5400
                IF(VAL(KK,I).LE.WASTE(KK,I))THEN
                   VAL(KK,I)=WASTE(KK,I)
                ENDIF
                   OVAL(KK,I)=VAL(KK,I)
           ENDDO
        ENDDO

C       WRITE THE VALUES OUT

        DO J=1,3
        WRITE(32,20)((OVAL(J,I)),I=1,5400)
        ENDDO
20      FORMAT(30(1X,30I4,/),/)
C       DO J=1,3
C       WRITE(31,20)(VAL(J,I),I=1,5400)
C       ENDDO
!
!
```

```
!
         RETURN
         END
!
```

```
C***********************************************************************
C****                                                              *****
C****                    SUBROUTINE HELD                           *****
C****                                                              *****
C****  THIS SUBROUTINE PERFORMS THE NECESSARY STEPS IN A GIVEN      *****
C****  ITERATION OF THE  ALGORITHM. IN A GIVEN  ITERATION, THE      *****
C****  CONSTRAINT VIOLATIONS OF THE SCHEDULE OBTAINED IN THE        *****
C****  PREVIOUS ITERATION IS DETERMINED FIRST. BASED ON DEGREE OF   *****
C****  THE VIOLATIONS A NEW SET OF MULTIPLIERS ARE CALCULATED TO    *****
C****  GET AN IMPROVED SCHEDULE. AT THE FINAL STEP A NEW SCHEDULE   *****
C****  IS DETERMINED BY USING THE MULTIPLIERS FOUND.                *****
C****  THE MULTIPLIER ADJUSTMENTS ARE BASED ON THE SUBGRADIENT      *****
C****  OPTIMIZATION METHOD SUGGESTED BY HELD AND KARP. FOR THE      *****
C****  DETAILS OF THIS METHOD SEE :                                 *****
C****      1) HELD, M. AND KARP, R.M., 1970, 'THE TRAVELING         *****
C****         SALESMAN PROBLEM AND MINIMUM SPANNING TREES',         *****
C****         OPERATIONS RESEARCH, VOL. 18, PP. 1138-1162,          *****
C****         1970.                                                 *****
C****      2) HELD, M., AND KARP, R.M., 1971, 'THE TRAVELING        *****
C****         SALESMAN PROBLEM AND MINIMUM SPANNING TREES:          *****
C****         PART II'. MATHEMATICAL PROGRAMING,VOL.1,              *****
C****         PP. 6-25, 1971                                        *****
C****      3) HELD. M., WOLFE, P., AND CROWDER, H.D.,1974           *****
C****         'VALIDATION OF SUBGRADIENT OPTIMIZATION ',            *****
C****         MATHEMATICAL PROGRAMING, VOL. 6, PP.62-88.            *****
C****      4) MURTY, KATTA, 1979, LINEAR AND COMBINATORIAL          *****
C****         PROGRAMING, JOHN WILEY AND SONS, NEWYORK,1979.        *****
C****                                                              *****
C****  INPUT FILES: NONE                                            *****
C****                                                              *****
C****  SUBROUTINES:                                                 *****
C****  THIS ROUTINE CALLS THE 'DOIT' SUBROUTINE TO OBTAIN           *****
C****  A SCHEDULE FOR A GIVEN SET OF MULTIPLIERS.                   *****
C****                                                              *****
C****  DESCRIPTION OF VARIABLES:                                    *****
C****  MULT(I)  : ARRAY CONTAINING THE CURRENT MULTIPLIER VECTOR    *****
C****  STEP     : SIZE OF THE STEP TAKING IN THE IMPROVING DIR.     *****
C****  TVI      : SUM OF THE SQUARES OF THE VIOLATIONS.             *****
C****  VIOL(I)  : ARRAY CONTAINING VIOLATIONS IN EACH CONSTRAINT    *****
C****             OF TIME PERIODS.                                  *****
C****  VIOL(1)  : VIOLATION OF CONSTRAINT ONE IN TIME PERIOD ONE.   *****
C****  VIOL(2)  : VIOLATION OF CONSTRAINT TWO IN TIME PERIOD ONE.   *****
C****  VIOL(3)  : VIOLATION OF CONSTRAINT THREE IN PERIOD ONE ETC.  *****
C****                                                              *****
C***********************************************************************
      SUBROUTINE HELD(LM,RHS)

      REAL*4 LM(9),RHS(9),B(9),VIOL(9),MULT(9),VIOLO(9)
```

```
         COMMON /INIT/B,NR

         IW=0

         TYPE *, 'HELD'

C        DETERMINE VIOLATION VECTOR
         TVI=0.
         DO I=1,NR
            VIOL(I)=RHS(I)-B(I)
            VIOLO(I)=VIOL(I)
            TVI=TVI+VIOL(I)**2
         ENDDO

C        STORE TOTAL VIOLATIONS

         TVIO=TVI

C        USE HELD'S MEDHOD TO DETERMINE A STEP

C        XMOBJ=NPVT-500.
         ROW=2
C        STEP=ROW*(PVAL-XMOBJ)/TVI
         STEP=4.
         TYPE *,STEP

C        DETERMINE THE NEW SET OF MULTIPLIERS

         DO I=1,NR
            MULT(I)=LM(I)+STEP*VIOL(I)
         ENDDO
         TYPE *,MULT

C        INITILIZE LOWER AND UPPER VALUES OF THE STEP

         STEPL=0
         STEPU=STEP

C        GENERATE A NEW SOLUTION

10       CALL DOIT(MULT,RHS)

C        DETERMINE VIOLATIONS OF THE NEW SOLUTION

         TVI=0
         DO I=1,NR
            VIOL(I)=RHS(I)-B(I)
```

```
           TVI=TVI+VIOL(I)**2
        ENDDO

C       IF THE NEW SOLUTION IS BETTER THAN THE OLD ONE RETURN

        IF((TVI-150.) .LT.TVIO)GO TO 20

C       IF THE NEW SOLUTION IS WORSE, THEN ADJUST THE STEP

        IF(TVI.GT.TVIO)THEN

                IW=1
                STEPU=STEP
                STEP=(STEPL+STEP)/2

        ELSEIF(IW.EQ.1)THEN

                STEPL=STEP
                STEP=(STEPL+STEPU)/2.0
        ELSE

                STEPL=STEP
                STEP=STEPL+STEP
        ENDIF

C       CALCULATE A NEW SET OF MULTIPLIERS

        DO I=1,NR
           MULT(I)=LM(I)+STEP*VIOLO(I)
        ENDDO
        TYPE *,MULT

        GOTO 10

20      DO I=1,NR
           LM(I)=MULT(I)
        ENDDO

        RETURN
        END
```

```
C***********************************************************************
C****                                                              *****
C****                   SUBROUTINE DOIT                            *****
C****                                                              *****
C****  THIS SUBROUTINE IS A DRIVER ROUTINE FOR THE SCHEDULING ALG. *****
C****  THIS ROUITINE SUMS THE MULTIPLIERS CORRESPONDING TO EACH    *****
C****  BLOCK, ADJUSTS THE BLOCK VALUES AND CALLS THE MULTI PERIOD  *****
C****  SCHEDULING SUBROUTINE.                                      *****
C****                                                              *****
C****  INPUT FILES: NONE                                           *****
C****                                                              *****
C****  SUBROUTINES:                                                *****
C****  THIS ROUTINE CALLS THE FOLLOWING SUBROUTINE:                *****
C****           1)SUBROUTINE ADJUST                                *****
C****           2)SUBROUTINE MULTSH                                *****
C****  SUBROUTINE ADJUST ADJUSTS THE VALUES OF THE BLOCKS BY       *****
C****  SUBSTRACTING THE PENALTIES DETERMINED BY MULTIPLIERS        *****
C****  FROM THE ORIGINAL BLOCK VALUES.                             *****
C****  SUBROUTINE MULTSH DETERMINES THE OPTIMUM SCHEDULE BY        *****
C****  USING THE MULTI PERIOD SCHEDULING DEVELOPED IN DAGDELEN'S   *****
C****  Ph.D DISSERTATION.                                          *****
C****                                                              *****
C****  DESCRIPTION OF VARIABLES:                                   *****
C****   AVGR(I) : ARRAY CONTAING AVERAGE GRADE OF THE ORE MINED    *****
C****             IN  PERIOD I                                     *****
C****   LM(I)   : LAGRANGE   MULTIPLIER  CORRESPONDING  TO         *****
C****             CONTSRAINT I                                     *****
C****   RHS(I)  : RIHGT HAND SIDE OF CONSTRAINT I IN THE CURRENT   *****
C****             SCHEDULE.                                        *****
C****   SUM(I)  : ARRAY CONTAINING THE VALUE OF TOTAL NUMBER OF    *****
C****             WASTE BLOCKS MINED IN PERIOD I                   *****
C****   SUMO(I) : ARRAY CONTAINING THE VALUE OF TOTAL NUMBER OF    *****
C****             ORE BLOCKS MINED IN  PERIOD I                    *****
C****   TLM(I,J): TOTAL PENALTIES ASSOCIATED TO BLOCK J IN PERIOD  *****
C****             I                                                *****
C****   TTP     : NUMBER OF TIME PERIODS                           *****
C****                                                              *****
C***********************************************************************
        SUBROUTINE DOIT(LM,RHS)
        INTEGER*2 TTP
        INTEGER*2 WOI(0:5,5400),MBT(0:5,5400)
        INTEGER*4 VAL(0:5,5400),WASTE(0:5,5400)
        INTEGER*4 SUM(20),SUMO(20),NPV(20)

        REAL*4 X(5400),TGR(20),AVGR(20)
        REAL*4 TLM(0:5,5400),LM(9),RHS(9)
```

```
      COMMON TTP,IC
      COMMON /BLK6/VAL,MBT,WOI
      COMMON /BLK7/SUM,SUMO,NPV,TGR,AVGR
      COMMON /BLK8/X
      COMMON /BLK29/TLM
      COMMON /BLK30/WASTE

C     INITILIZE THE TOTAL MULTIPLIERS

      DO J=1,3
      DO I=1,TTP
        TLM(J,I)=0.0
      ENDDO
      ENDDO

C     DETERMINE TOTAL PENALTY PER TIME PERIOD

      DO I=1,5400
      TLM(1,I)=LM(1)+LM(2)*X(I)
C     IF(TLM(1).LT.0)TLM(1)=0.0
      TLM(2,I)=LM(3)+LM(4)*X(I)
      TLM(3,I)=LM(5)+LM(6)*X(I)
      ENDDO

C     CALL SUBROUTINE TO ADJUST THE BLOCK VALUES

      CALL ADJUST

C
C     CALL SUBROUTINE TO DO MULTI TIME PERIOD PRODUCTION
C     SCHEDULING

      CALL MULTSH

C     SEND BACK THE NEW RIGHT HAND SIDES

      RHS(1)=SUMO(1)
      RHS(2)=AVGR(1)
      RHS(3)=SUMO(2)
      RHS(4)=AVGR(2)
      RHS(5)=SUMO(3)
      RHS(6)=AVGR(3)

C     END OF SUBROUTINE DOIT
      RETURN
      END
```

```
C***********************************************************************
C****                   SUBROUTINE ADJUST                          *****
C****                                                              *****
C**** THIS SUBROTINE ADJUSTS THE ORIGINAL BLOCK VALUES BY          *****
C**** SUBSTRACTING TOTAL MULTIPLIER VALUES. THIS SUBROUTINE ALSO   *****
C**** DETERMINES IF A BLOCK TO BE MINED AS ORE OR AS WASTE BY      *****
C**** CALLING SUBROUTINE COMPARE.                                  *****
C****                                                              *****
C**** INPUT FILES: NONE                                            *****
C****                                                              *****
C**** SUBROUTINES:                                                 *****
C****                                                              *****
C**** THIS PROGRAM CALLS SUBROUTINE 'COMPARE' TO DETERMINE WASTE   *****
C**** ORE INDICATOR.                                               *****
C****                                                              *****
C**** DESCRIPTION OF VARIABLES: (SEE READ SUBROUTINE)              *****
C****                                                              *****
C***********************************************************************
        SUBROUTINE ADJUST
        INTEGER*2 TTP
        INTEGER*2 WOI(0:5,5400),MBT(0:5,5400)
        INTEGER*4 VAL(0:5,5400),WASTE(0:5,5400)
        INTEGER*4 OVAL(5,5400)
        REAL*4 TLM(0:5,5400)
        REAL*4 X(5400)
!
        COMMON TTP,IC
        COMMON/BLK6/VAL,MBT,WOI
        COMMON/BLK8/X
        COMMON/BLK29/TLM
        COMMON/BLK30/WASTE
        COMMON/OVAL/OVAL
!
!
C       MODIFY THE BLOCK VALUES
!
!
        DO J=1,TTP
                DO I=1,5400
                   IF(X(I).GT.0)THEN
                IF(J.EQ.1.AND.I.GT.2700)TLM(J,I)=5000.
                IF(J.EQ.2.AND.I.GT.4500)TLM(2,I)=5000.
                 VAL(J,I)=OVAL(J,I)-TLM(J,I)
                   ENDIF
                ENDDO
        ENDDO
!
!
```

```
C       COMPARE THE ORE VS WASTE VALUE AND CHOOSE THE
C       HIGHEST
!
!
        CALL COMPARE
!
!
C       DO J=1,3
C       WRITE(33,20)((VAL(J,I)),I=1,5400)
C       ENDDO
C20     FORMAT(30(1X,30I4,/),/)

        RETURN
        END
```

```
C*************************************************************************
C****                     SUBROUTINE COMPARE                         *****
C****                                                                *****
C**** THIS SUBROUTINE COMPARES THE ORE VALUES TO STRIPPING COSTS     *****
C**** IN DETERMINING WHICH BLOCK TO BE MINED AS ORE AND WHICH ONES   *****
C**** AS WASTE                                                       *****
C****                                                                *****
C**** INPUT FILES: NONE                                              *****
C****                                                                *****
C**** DESCRIPTION OF VARIABLES: (SEE SUBROUTINE READ)                *****
C****                                                                *****
C*************************************************************************
        SUBROUTINE COMPARE
        INTEGER*2 TTP
        INTEGER*4 VAL(0:5,5400),WASTE(0:5,5400)
        INTEGER*2 WOI(0:5,5400),MBT(0:5,5400)
        COMMON TTP,IC
        COMMON /BLK6/VAL,MBT,WOI
        COMMON /BLK30/WASTE
!
!
C       SET WASTE ORE INDICATOR TO BE 1 FOR THE WASTE BLOCKS
!
!
        DO KK=0,TTP
                DO I=1,5400
                        IF(VAL(KK,I).LE.WASTE(KK,I))THEN
                                VAL(KK,I)=WASTE(KK,I)
                        ELSE
                                WOI(KK,I)=0
                        ENDIF
                ENDDO
        ENDDO

C       DO J=1,3
C       WRITE(31,20)(VAL(J,I),I=1,5400)
C       ENDDO
C20     FORMAT(30(1X,30I4,/)/)
!
        RETURN
        END
```

```
C*************************************************************************
C****                     SUBROTINE MULTSH                          *****
C****                                                               *****
C**** GIVEN THE BLOCK VALUES IN EACH TIME PERIOD THIS SUBROUTINE    *****
C**** DETERMINES WHICH BLOCKS SHOULD BE MINED IN A GIVEN PERIOD.    *****
C**** THE MULTI PERIOD SCHEDULING IS BASED ON THE ALGORITHM GIVEN   *****
C**** BY KADRI DAGDELEN. (SEE DAGDELEN, K., 1985, ' OPTIMUM MULTI   *****
C**** PERIOD OPEN PIT SCHEDULING ', Ph.D DISSERTATION, COLORADO     *****
C**** SCHOOL OF MINES.                                              *****
C****                                                               *****
C**** INPUT FILES: NONE                                             *****
C****                                                               *****
C**** SUBROUTINES:                                                  *****
C**** FOLLOWING SUBROUTINES ARE CALLED TO DO VARIOUS STEPS OF THE   *****
C**** MULTI PERIOD SEQUENCING ALGORITHM:                            *****
C****            1) SUBROUTINE OPCP1                                *****
C****            2) SUBROUTINE OPCP2                                *****
C****            3) SUBROUTINE OPCP3                                *****
C****            4) SUBROUTINE TIMING                               *****
C****            5) SUBROUTINE SUMMARY                              *****
C****                                                               *****
C*************************************************************************

!
        SUBROUTINE MULTSH
!
!
C       DO ULTIMATE PIT LIMIT ANALYSIS WITH TIME PERIOD 1 OPORTUNITY
C       COSTS.
!
!
        CALL OPCP1
!
!
C       DO ULTIMATE PIT LIMIT ANALYSIS WITH TIME PERIOD 2 OPORTUNITY
C       COSTS.
!
!
        CALL OPCP2
!
!
C       DO ULTIMATE PIT LIMIT ANALYSIS WITH TIME PERIOD 3 OPORTUNITY
C       COSTS. (SINCE THIS IS THE LAST TIME PERIOD DO THIS STEP WITH
C       TIME PERIOD 3 COSTS)
!
!
        CALL OPCP3
!
```

```
!
C       DETERMINE WHAT BLOCKS WILL BE MINED IN A GIVEN PERIOD OPTIMALY
!
!
        CALL TIMING
!
!
C       DETERMINE TONS OF ORE, TONS WASTE, AND THE AVERAGE GRADE
C       MINED IN A GIVEN TIME PERIOD
!
!
        CALL SUMMARY
!
!
C       END THE SUBROUTINE
!
!
        RETURN
        END
!
```

```
!
C***********************************************************************
C****                                                              *****
C****                   SUBROUTINE OPCP1                           *****
C****                                                              *****
C****   THIS SUBROUTINE SOLVES ULTIMATE OPEN PIT LIMIT PROBLEM     *****
C****   BY USING THE OPPORTUNITY COSTS BETWEEN PERIOD 1            *****
C****   AND PERIOD 2.                                              *****
C****                                                              *****
C****   INPUT FILES: NONE                                          *****
C****                                                              *****
C****   SUBROUTINES: PIT                                           *****
C****                                                              *****
C****   DESCRIPTION OF VARIABLES:                                  *****
C****   OC1(I)  : ARRAY CONTAINING OPPORTUNITY VALUE OF BLOCK I    *****
C****             IN TIME PERIOD ONE.                              *****
C****   OC2(I)  : ARRAY CONTAINING OPPORTUNITY VALUE OF BLOCK I    *****
C****             IN TIME PERIOD 2.                                *****
C****   OC3(I)  : ARRAY CONTAINING OPPORTUNITY VALUE OF BLOCK I    *****
C****             IN TIME PERIOD 3.                                *****
C****                                                              *****
C***********************************************************************
        SUBROUTINE OPCP1
!
        INTEGER*2 MBT(0:5,5400),WOI(0:5,5400),MBT1(5400)
        INTEGER*4 VAL(0:5,5400)
        INTEGER*4 OC1(5400),OC2(5400),OC3(5400)

        COMMON /BLK6/VAL,MBT,WOI
        COMMON /BLK10/OC1,OC2,OC3

!
!
C       CALCULATE THE OPPORTUNITY COST OF  MINING A BLOCK IN TIME
C       PERIOD 1 INSTEAD OF IN TIME PERIOD 2.
!
!
                DO I=1,5400
                        OC1(I)=VAL(1,I)-VAL(2,I)
                ENDDO
!
!
C       CALL THE ULTIMATE PIT LIMIT SUBROUTINE
!
!
                CALL PIT(OC1,MBT1)
!
!
```

```
                DO I=1,5400
                MBT(1,I)=MBT1(I)
                ENDDO
!
C               NLEV=6
C               LEV=0
C               LBN1=1
C               LBN2=900
C               DO WHILE (LEV.LT.NLEV)
C                       WRITE(47,50)(MBT(1,I),I=LBN1,LBN2)
C50                     FORMAT(1X,30(/1X,30('+---'),'+',/1X,'I ',30(I1,' I
C     +                         1X,30('+---'),'+'///)
C                       LEV=LEV+1
C                       LBN1=LEV*900+1
C                       LBN2=LBN1+899
C               END DO
!
!
C       END THE SUBROUTINE
!
!
        RETURN
        END
```

```
!
C***********************************************************************
C****                                                              *****
C****                   SUBROUTINE OPCP2                           *****
C****                                                              *****
C****   THIS SUBROUTINE SOLVES ULTIMATE OPEN PIT LIMIT PROBLEM     *****
C****   BY USING THE OPPORTUNITY COSTS DETERMINED FOR PERIOD 2     *****
C****                                                              *****
C****   INPUT FILES: NONE                                          *****
C****                                                              *****
C****   SUBROUTINES: SUBROUTINE PIT                                *****
C****                                                              *****
C****   DESCRIPTION OF VARIABLES:                                  *****
C****                                                              *****
C***********************************************************************
        SUBROUTINE OPCP2
!
        INTEGER*2 MBT(0:5,5400),WOI(0:5,5400),MBT2(5400)
        INTEGER*4 VAL(0:5,5400)
        INTEGER*4 OC1(5400),OC2(5400),OC3(5400)

        COMMON /BLK6/VAL,MBT,WOI
        COMMON /BLK10/OC1,OC2,OC3

!
!
C       CALCULATE THE OPPORTUNITY COST OF  MINING A BLOCK IN TIME
C       PERIOD 1 INSTEAD OF IN TIME PERIOD 2.
!
!
                DO I=1,5400
                        IF(MBT(1,I).EQ.1)THEN
                                OC2(I)=VAL(2,I)-VAL(3,I)+OC1(I)
                        ELSE
                                OC2(I)=VAL(2,I)-VAL(3,I)
                        ENDIF
                ENDDO
!
!
C       CALL THE ULTIMATE PIT LIMIT SUBROUTINE
!
!
                CALL PIT(OC2,MBT2)
!
                DO I=1,5400
                MBT(2,I)=MBT2(I)
                ENDDO
!
```

```
C           NLEV=6
C           LEV=0
C           LBN1=1
C           LBN2=900
C           DO WHILE (LEV.LT.NLEV)
C                 WRITE(47,50)(MBT(2,I),I=LBN1,LBN2)
C50               FORMAT(1X,30(/1X,30('+---'),'+',/1X,'I ',30(I1,' I
C    +                  1X,30('+---'),'+'///)
C                 LEV=LEV+1
C                 LBN1=LEV*900+1
C                 LBN2=LBN1+899
C           END DO
!
C     END THE SUBROUTINE
!
!
      RETURN
      END
!
```

```
!
C************************************************************************
C****                                                               *****
C****                   SUBROUTINE OPCP3                            *****
C****                                                               *****
C****   THIS SUBROUTINE SOLVES ULTIMATE OPEN PIT LIMIT PROBLEM      *****
C****   BY USING THE OPPORTUNITY COSTS OF PERIOD 3                  *****
C****                                                               *****
C****   INPUT FILES: NONE                                           *****
C****                                                               *****
C****   SUBROUTINES: SUBROUTINE PIT                                 *****
C****                                                               *****
C****   DESCRIPTION OF VARIABLES:                                   *****
C****   OC1(I)  : ARRAY CONTAINING OPPORTUNITY VALUE OF BLOCK I     *****
C****             IN TIME PERIOD ONE.                               *****
C****   OC2(I)  : ARRAY CONTAINING OPPORTUNITY VALUE OF BLOCK I     *****
C****             IN TIME PERIOD 2.                                 *****
C****   OC3(I)  : ARRAY CONTAINING OPPORTUNITY VALUE OF BLOCK I     *****
C****             IN TIME PERIOD 3.                                 *****
C****                                                               *****
C************************************************************************
        SUBROUTINE OPCP3
!
!
        INTEGER*2 MBT(0:5,5400),WOI(0:5,5400),MBT3(5400)
        INTEGER*4 VAL(0:5,5400)
        INTEGER*4 OC1(5400),OC2(5400),OC3(5400)

        COMMON /BLK6/VAL,MBT,WOI
        COMMON /BLK10/OC1,OC2,OC3

!
!
C       CALCULATE THE OPPORTUNITY COST OF  MINING A BLOCK IN TIME
C       PERIOD 1 INSTEAD OF IN TIME PERIOD 2.
!
!
                DO I=1,5400
                        IF(MBT(2,I).EQ.1)THEN
                                OC3(I)=VAL(3,I)+OC2(I)
                        ELSE
                                OC3(I)=VAL(3,I)
                        ENDIF
                ENDDO
!
!
C       CALL THE ULTIMATE PIT LIMIT SUBROUTINE
!
!
```

```
            CALL PIT(DC3,MBT3)
!
            DO I=1,5400
            MBT(3,I)=MBT3(I)
            ENDDO
!
C           NLEV=6
C           LEV=0
C           LBN1=1
C           LBN2=900
C           DO WHILE (LEV.LT.NLEV)
C                 WRITE(47,50)(MBT(3,I),I=LBN1,LBN2)
C50               FORMAT(1X,30(/1X,30('+---'),'+',/1X,'I ',30(I1,' I
C    +                  1X,30('+---'),'+'///)
C                 LEV=LEV+1
C                 LBN1=LEV*900+1
C                 LBN2=LBN1+899
C           END DO
!
!
C     END THE SUBROUTINE
!
!
      RETURN
      END
!
```

```
!
C***********************************************************************
C****                                                              *****
C****                   SUBROUTINE TIMING                          *****
C****                                                              *****
C**** THIS SUBROUTINE DETERMINE THE TIMING OF THE BLOCKS ACCORDING *****
C**** TO THE RULE GIVING IN CHAPTER 7 OF THE KADRI DAGDELEN'S      *****
C**** DISSERTATION.                                                *****
C****                                                              *****
C***********************************************************************

!
!
        SUBROUTINE TIMING
!
!
        INTEGER*4 VAL(0:5,5400)
        INTEGER*2 MBT(0:5,5400),WOI(0:5,5400)
!
        COMMON /BLK6/VAL,MBT,WOI
!
!
C       DETERMINE WHEN EACH BLOCK TO BE MINED BASED ON THE
C       PREVIOUS OPPORTUNITY COST SOLUTIONS
!
!
                DO I=1,5400
                        IF(MBT(3,I).EQ.1)THEN
                                        MBT(0,I)=3
                                        WOI(0,I)=WOI(3,I)
                        ENDIF
                        IF(MBT(2,I).EQ.1)THEN
                                        IF(MBT(3,I).EQ.1)THEN
                                                MBT(0,I)=2
                                                WOI(0,I)=WOI(2,I)
                                        ENDIF
                        ENDIF
                        IF(MBT(1,I).EQ.1 .AND. MBT(2,I).EQ.1)THEN
                                        IF(MBT(3,I).EQ.1)THEN
                                                MBT(0,I)=1
                                                WOI(0,I)=WOI(1,I)
                                        ENDIF
                        ENDIF
                ENDDO
!
```

```
.
!
C        END OF THE SUBROUTINE
!
!
         RETURN
         END
```

```
!
C***********************************************************************
C****                                                              *****
C****                     SUBROUTINE SUMMARY                       *****
C****                                                              *****
C**** THIS SUBROUTINE SUMMARIZES THE OVERALL PRODUCTION TONNAGES    *****
C**** AND GRADES MINED IN EACH TIME PERIOD.                         *****
C****                                                              *****
C**** SUBROUTINES:                                                 *****
C****           1) SUBROUTINE INISTA                               *****
C****           2) SUBROUTINE STAT                                 *****
C****           3) SUBROUTINE AVGRD                                *****
C****                                                              *****
C****  DEFINITION OF VARIABLES:                                    *****
C****  AVGR(I)  : ARRAY CONTAINING AVERAGE GRADE OF THE MATERIAL   *****
C****             MINED IN PERIOD I                                *****
C****  NPV(I)   : THE PRESENT WORTH OF THE ADJUSTED VALUES OF THE  *****
C****             BLOCKS MINED IN PERIOD I                         *****
C****  NPVA     : THE UNADJUSTED NET PRESENT VALUE OF BLOCKS       *****
C****             MINED IN PERIOD I (NPV OF THE ORIGINAL VALUES)   *****
C****  TGR(I)   : TOTAL METAL CONTENT OF ORE MINED IN PERIOD I     *****
C****                                                              *****
C***********************************************************************
        SUBROUTINE SUMMARY

        INTEGER*2 MBT(0:5,5400),WOI(0:5,5400)
        INTEGER*2 KK,TTP
        INTEGER*4 VAL(0:5,5400)
        INTEGER*4 SUM(20),SUMO(20),NPV(20),NPVA(20)

        REAL*4    TGR(20),AVGR(20)
!
        COMMON TTP,IC
        COMMON /BLK6/VAL,MBT,WOI
        COMMON /BLK7/SUM,SUMO,NPV,TGR,AVGR
        COMMON /BLK9/NPVT
        COMMON /BLK15/NPVA
!
!
C       INITIALIZE THE SUMMARY VARIABLES
!
!
                CALL INISTA
!
!
C       DO THE BOOKKEEPING
!
!
```

```
            DO I=1,5400
                  IF(MBT(0,I).EQ.1)THEN
                        KK=1
                        CALL STAT(KK,I)
                  ELSEIF(MBT(0,I).EQ.2)THEN
                        KK=2
                        CALL STAT(KK,I)
                  ELSEIF(MBT(0,I).EQ.3)THEN
                        KK=3
                        CALL STAT(KK,I)
                  ENDIF
            ENDDO
!
!
C     CALL AVERAGE GRADE SUBROUTINE
!
!
      CALL AVGRD
!

      NPVT=0
      DO I=1,3
      NPVT=NPVT+NPVA(I)
      ENDDO

C     WRITE(6,9)(NPVA(I),I=1,3)
C9    FORMAT(14X,'NPV OF THE ORIGINAL PROB.:',3(I8,2X))
      WRITE(6,10)NPVT
10    FORMAT(//,14X,'NPVT OF THE SCHEDULE:',I8/,
    1           4X,T26,'TOTAL TONS',6X,'ORE TONS',
    1           6X,'AVERA GRADE')
!
      DO I=1,TTP
            WRITE(6,20)I,SUM(I),SUMO(I),AVGR(I)
20          FORMAT(14X,'PERIOD',I1,T28,I4,12X,I4,14X,F5.1)
      ENDDO
!
C     END OF THE SUBROUTINE SUMARY
!

      RETURN
      END
```

```
C*************************************************************************
C****                                                                 *****
C****                    SUBROUTINE INISTA                            *****
C****                                                                 *****
C**** THIS SUBROUTINE  INITIALIZES VARIABLES USED IN SUMMARY           *****
C**** SUBROUTINE.                                                      *****
C****                                                                 *****
C*************************************************************************
        SUBROUTINE INISTA

        INTEGER*4 SUM(20),SUMO(20),NPV(20),NPVA(20)

        REAL*4    TGR(20),AVGR(20)
!
        COMMON /BLK7/SUM,SUMO,NPV,TGR,AVGR
        COMMON /BLK15/NPVA
!
!
C       INITIALIZE
!
!
                DO I=1,20
                        SUM(I)=0.
                        SUMO(I)=0.
                        NPV(I)=0.
                        NPVA(I)=0.
                        TGR(I)=0.
                        AVGR(I)=0.
                ENDDO
!
!
C       END OF THE SUBROUTINE 'INISTA'
!
!
        RETURN
        END
```

```
C***********************************************************************
C****                                                              *****
C****                   SUBROUTINE STAT                            *****
C****                                                              *****
C****   THIS SUBROUTINE DETERMINES THE ORE AND WASTE STATISTICS    *****
C****   OF DIFFERENT TIME PERIODS.                                 *****
C****                                                              *****
C**** DEFINITION OF THE VARIABLES: (SEE SUBROUTINE SUMMARY)        *****
C****                                                              *****
C***********************************************************************
        SUBROUTINE STAT(KK,I)
!
        INTEGER*2 MBT(0:5,5400),WOI(0:5,5400)
        INTEGER*2 KK
        INTEGER*4 VAL(0:5,5400)
        INTEGER*4 SUM(20),SUMO(20),NPV(20),NPVA(20)
        REAL*4 X(5400),TGR(20),AVGR(20)
!
        COMMON /BLK6/VAL,MBT,WOI
        COMMON /BLK7/ SUM,SUMO,NPV,TGR,AVGR
        COMMON /BLK8/X
        COMMON /BLK15/NPVA
!
!
C       DO THE BOOKKEEPING
!
!
                        IF(WOI(KK,I).EQ.1)THEN
                        SUM(KK)=SUM(KK)+1
                        ENDIF
                        IF(WOI(KK,I).NE.1)THEN
                                SUMO(KK)=SUMO(KK)+1
                                TGR(KK)=TGR(KK)+X(I)
                        ENDIF
                        NPV(KK)=NPV(KK)+VAL(KK,I)
                        IF(KK.EQ.1)THEN
                        NPVA(KK)=NPVA(KK)+VAL(0,I)
                        ELSEIF(KK.EQ.2)THEN
                        NPVA(KK)=NPVA(KK)+VAL(0,I)/1.125
                        ELSEIF(KK.EQ.3)THEN
                        NPVA(KK)=NPVA(KK)+VAL(0,I)/(1.125**2)
                        ENDIF
                        IF(WOI(KK,I).EQ.0)WOI(0,I)=0
!
```

```
!
C       END OF THE SUBROUTINE 'STAT(KK)'
!
!
        RETURN
        END
!
!
```

```
C***********************************************************************
C****                                                              *****
C****                   SUBROUTINE AVGRD                           *****
C****                                                              *****
C****  THIS SUBROUTINE DETERMINES THE AVERAGE GRADES OF THE ORE    *****
C****  MINED IN EACH TIME PERIOD.                                  *****
C****                                                              *****
C****  DESCRIPTION OF THE VARIABLES: (SEE SUBROUTINE SUMMARY)      *****
C****                                                              *****
C***********************************************************************
        SUBROUTINE AVGRD
!
        INTEGER*2 TTP
        INTEGER*4 SUM(20),SUMO(20),NPV(20),NPVT
        REAL*4 TGR(20),AVGR(20)
!
        COMMON TTP,IC
        COMMON /BLK7/SUM,SUMO,NPV,TGR,AVGR
        COMMON /BLK9/NPVT


        NPVT=0
!
!
                DO I=1,TTP
                        IF(SUMO(I).GT.0)THEN
                                AVGR(I)=TGR(I)/SUMO(I)
                        ELSE
                                AVGR(I)=0
                        ENDIF
                        NPVT=NPVT+NPV(I)
                ENDDO
!
!
C       END OF THE SUBROUTINE 'AVGR'
!
!
                RETURN
                END
!
!
```

```
:
C***********************************************************************
C****                                                              *****
C****                    SUBROUTINE PRINTS                         *****
C****                                                              *****
C**** THIS SUBROUTINE PRINTS THE DETERMINED SCHEDULE CORRESPONDING *****
C**** TO A GIVEN MULTIPLIERS.                                       *****
C****                                                              *****
C**** OUTPUT FILES:                                                 *****
C**** THE SUMMARY STATISTICS OF THE SCHEDULE IS WRITTEN INTO A      *****
C**** DISK FILE NAMED 'FOR041.DAT'.                                 *****
C****                                                              *****
C**** THE BENCH BY BENCH PRINTOUT OF THE SCHEDULE IS WRITTEN INTO  *****
C**** THE DISK FILE NAMED 'FOR043.DAT'.                             *****
C****                                                              *****
C**** THE BENCH BY BENCH ORE WASTE INDICATORS OF THE BLOKCS ARE    *****
C**** INTO THE DISK FILE NAMED 'FOR044.DAT'.                        *****
C****                                                              *****
C***********************************************************************
        SUBROUTINE PRINTS
!
        INTEGER*2 TTP
        INTEGER*2 MBT(0:5,5400),WOI(0:5,5400)
        INTEGER*2 IBLK1,IBLK2
        INTEGER*4 VAL(0:5,5400)
        INTEGER*4 SUM(20),SUMO(20),NPV(20),NPVT,NPVA(20)
        REAL*4 TGR(20),AVGR(20)
!
        COMMON TTP,IC
        COMMON/BLK6/VAL,MBT,WOI
        COMMON/BLK7/SUM,SUMO,NPV,TGR,AVGR
        COMMON/BLK9/NPVT
        COMMON/BLK15/NPVA
!
!
C       WRITE THE NECESSARY STATISTICS
!
!
        WRITE(42,9)(NPVA(I),I=1,3)
9       FORMAT(14X,'NPV OF THE ORIGINAL PROB.:',3(I8,2X))
        WRITE(41,10)NPVT
10      FORMAT(//,14X,'NPVT OF THE SCHEDULE:',I8/,
     1         4X,T26,'TOTAL TONS',6X,'ORE TONS',
     1         6X,'AVERA GRADE')
!
        DO I=1,TTP
                WRITE(41,20)I,SUM(I),SUMO(I),AVGR(I)
20              FORMAT(14X,'PERIOD',I1,T28,I4,12X,I4,14X,F5.1)
        ENDDO
```

```
!
!
CCC             K=70
C               NLEV=6
C               LEV=0
C               LBN1=1
C               LBN2=900
C               DO WHILE (LEV.LT.NLEV)
C                       WRITE(K,50)(MBT(0,I),I=LBN1,LBN2)
C50                     FORMAT(1X,30(/1X,30('+---'),'+',/1X,'I ',30(I1,' I '
C     +                         1X,30('+---'),'+'///)
C50                     FORMAT(30(30(I3),/))
C                       K=K+1
C                       LEV=LEV+1
C                       LBN1=LEV*900+1
C                       LBN2=LBN1+899
C               END DO
!
        WRITE(43,60)(MBT(0,I),I=1,5400)
60              FORMAT(30(1X,30(I3),/),//)
        WRITE(44,60)(WOI(0,I),I=1,5400)

!
C       THE END OF THE SUBROUTINE 'PRINTS'
!
!
        RETURN
        END
C*****************************************************************************
C****                                                                    *****
C****                     SUBROUTINE PIT                                 *****
C****                                                                    *****
C**** THIS IS THE MAIN SUBROUTINE TO DETERMINE ULTIMATE PIT LIMITS *****
C**** OF AN OPEN PIT MINE IN 3-D BY USING LERCH'S AND GROSSMAN'S    *****
C**** TREE(GRAPH THEORY) ALGORITHM. THE DETAIL DISCUSSION OF THE    *****
C**** ALGORITHM IS GIVEN BY THE FOLLOWING REFERENCE:                *****
C****                                                                    *****
C****   LERCHS, H. AND GROSSMANN, I.F., 1965, 'OPTIMUM DESIGN OF    *****
C****      OPEN PIT MINES', C.M.I. BULLETIN, VOL. 58, No. 633,      *****
C****      JAN. 1965, PP. 47-54.                                    *****
C****                                                                    *****
C**** THIS SUBROUTINE IS ALSO WRITTEN BY USING STRUCTURED NETWORK   *****
C**** FLOW CONCEPTS DESCRIBED BY MURTY (SEE MURTY, K., 1979,        *****
C**** LINEAR AND COMBINATORIAL  PROGRAMING, JOHN WILEY AND SONS,    *****
C**** NEW YORK, 1979 PAGES 371-375) AND ALSO BY JENSEN AND BARNES   *****
C**** (SEE JENSEN, P.A., BARENS, J.W., 1980, NETWORK FLOW           *****
C**** PROGRAMING, JOHN WILEY & SONS, NEW YORK, 1980)                *****
C****                                                                    *****
C**** SUBROUTINES:                                                  *****
C****   1) 'SUBROUTINE TEMPLT' TO DETERMINE TEMPLATE TO TAKE        *****
C****      VARYING PIT SLOPE CONSTRAINTS INTO ACCOUNT.              *****
C****   2) 'SUBROUTINE INIT' TO INITIALIZE THE ORIGINAL TREE INDEX *****
C****      VARIABLES.                                               *****
C****   3) 'SUBROUTINE STEP1' TO START THE STEPS OF THE ALGORITHM  *****
C****                                                                    *****
C**** DESCRIPTION OF THE VARIABLES:                                 *****
C****    CV(I)  : CUMULATIVE VALUE OF THE NODE I                    *****
C****    INX    : INDICATOR OF STRONG NODES AT FIRST PASS OF THE    *****
C****             3-D ALGORITHM.                                    *****
C****    IS     : FLAG TO STOP ITERATIONS IF NO NEW NODES ARE       *****
C****             INCLUDED IN THE ULTIMATE PIT IN THE PREVIOUS      *****
C****             ITERATION OF THE 3-D ALGORITHM.                   *****
C****    H      : ARRAY CONTAINING SLOPE TEMPLATE                   *****
C****    NTX    : NUMBER OF ITERATIONS OF 3-D ALGORITHM             *****
C****    P(I)   : PREDECESSOR INDEX OF NODE I                       *****
C****    S(I)   : SUCCESSOR INDEX OF NODE I                         *****
C****    V(I)   : THE BLOCK VALUE OF NODE I                         *****
C****    VA(I)  : THE BLOCK VALUES PASSED TO THIS SUBROUTINE        *****
C****    WS(I)  : STRONG WEAK INDICATOR OF BLOCK I                  *****
C****    WSI(I) : WEAK OR STRONG INDICATOR INDEX OF NODE I          *****
C****             IF THE INDICATOR IS 1 THE NODE IS                 *****
C****             STRONG AND THE THE BLOCK IS INCLUDED IN THE SET  *****
C****             BLOCKS WHICH CAN ECONOMICALLY BE MINED.           *****
C****    YB(I)  : YOUNGER BROTHER INDEX OF NODE I                   *****
C****                                                                    *****
C*****************************************************************************
```

```
        SUBROUTINE PIT(VA,WS)

!
        INTEGER*2 WSI(5400),WS(5400),INX(5400),IS(50)
        INTEGER*2 S(-1:5400),P(-1:5400),YB(-1:5400)
        INTEGER*4 CV(5400)
        INTEGER*4 V(5400),VA(5400)
!
        REAL*4 H(-6:6,-6:6)
!
        COMMON /BLK1/ V,CV,WSI
        COMMON /BLK2/S,P,YB
        COMMON /BLK3/H
        COMMON /BLK12/INX
        COMMON /BLK14/IS

C!      WRITE(6,2)
C!2     FORMAT(1X,/,'PIT',/)

C!      WRITE(6,2)(V(I),I=1,1800)
C!2     FORMAT(30(30I4,/),/)
C       V(1271)=5
C       V(2323)=20
C       V(2327)=1000
        DO I=1,5400
        V(I)=VA(I)
        ENDDO

C       WRITE(48,10)(V(I),I=1,5400)
C10     FORMAT(30(30I4,/),/)
!
!               DEFINE PIT LIMIT BOUNDARY TEMPLT.
!
                CALL TEMPLT

!
!               INITIALIZE THE TREE INDEX VARIABLES
!
                CALL INIT
!
!               DO LEARCH AND GROSSMAN ALGORITHM LEVEL BY LEVEL
!
                NLEV=6

!
!
                DO I=1,50
                  IS(I)=0
                ENDDO
```

```
!
          DO NTX=1,50
               IF(NTX.EQ.1)GOTO12
               IF(IS(NTX-1).NE.1)GOTO14
12             TYPE *,NTX
               DO K=2,NLEV
               CALL STEP1(K,NTX)
               ENDDO
14        ENDDO

     DO I=1,5400
     WS(I)=WSI(I)
     ENDDO
!
!         PRINT THE LEVEL MAPS WITH INDICATORS OF THE MINED BLOCK
!
          CALL OUTLINE(NLEV,WSI)

          CALL PRINT (NLEV,WSI)
!

C!        WRITE (6,2)(WSI(I),I=1,1800)
C!2       FORMAT((30I1))
!
     RETURN
     END
!
```

```
C***********************************************************************
C****                                                              *****
C****                   SUBROUTINE TEMPLATE                        *****
C****  THIS SUBROUTINE GENERATES THE TEMPLATE TO TAKE UN ISOTROPIC *****
C****  SLOPE CONSTRAINTS INTO ACCOUNT.                             *****
C****                                                              *****
C***********************************************************************
        SUBROUTINE TEMPLT

        REAL*4  H(-6:6,-6:6)
!
        COMMON /BLK3/H
!
!       DETERMINE THE ALLOWABLE HEIGHT A BLOCK (WHICH IS LOCATED (I,J)
!       DISTANCE AWAY FROM THE APEX BLOCK) CAN HAVE IN ORDER TO BE IN THE CO
!       OF THE APEX BLOCK
!

C!      WRITE(6,2)
C!2     FORMAT(1X,/,'TEMPLT',/)

                DO I=-6,6,1
                        DO J=-6,6,1

!
                                H(I,J)=(I**2+J**2)**.5
!
                        ENDDO
                ENDDO
!
        RETURN
        END
!
```

```
C************************************************************************
C****                                                               *****
C****                   SUBROUTINE INIT                             *****
C****  THIS SUBROUTINE INITIALIZES ALL THE INDECES OF THE NODES     *****
C****  OF THE INITIAL TREE WITH RESPECT TO DUMMY ROOT               *****
C****                                                               *****
C************************************************************************
        SUBROUTINE INIT

        INTEGER*4 V(5400),CV(5400)
        INTEGER*2 WSI(5400),INX(5400),WSII(5400)
        INTEGER*2 S(-1:5400),P(-1:5400),YB(-1:5400)
!
        COMMON /BLK1/V,CV,WSI
        COMMON /BLK2/S,P,YB
        COMMON /BLK12/INX,WSII
!
!       INITIALIZE  ALL THE INDEXES FOR THE ORIGINAL TREE
!

C!      WRITE(6,2)
C!2     FORMAT(1X,/,'INIT',/)

                DO I=1,5400
!
                        CV(I)=V(I)
!
                        IF(V(I).GT.0)THEN
                                WSI(I)=1
                                WSII(I)=1
                        ELSE
                                WSI(I)=0
                                WSII(I)=0
                        ENDIF
!
                        P(I)=0
!
                        S(I)=-1
!
                        YB(I)=-1

                        INX(I)=0
                ENDDO
!
        RETURN
        END
!
!
```

```
C***********************************************************************
C****                                                              *****
C****                   SUBROUTINE STEP1                           *****
C**** THIS SUBROUTINE SEARCHES FOR A STRONG NODE ON A GIVEN LEVEL  *****
C**** CALLS ANOTHER SUBROUTINE TO CONNECT THIS STRONG NODE TO AN   *****
C**** OVERLYING WEAK NODE                                          *****
C****                                                              *****
C**** DESCRIPTION OF VARIABLES:                                    *****
C****   LNUM : LEVEL NUMBER BEING CONSIDERED                       *****
C****   NTX  : ITERATION NUMBER INDICATOR OF THE TREE ALGORITHM    *****
C****                                                              *****
C***********************************************************************
        SUBROUTINE STEP1(LNUM,NTX)
!
        INTEGER*2 WSI(5400),INX(5400),IS(50),WSII(5400)
        INTEGER*2 BN
        INTEGER*4 V(5400),CV(5400)
!
        COMMON /BLK1/V,CV,WSI
        COMMON /BLK12/INX,WSII
        COMMON /BLK14/IS
!
!       ON A GIVEN LEVEL LNUM START SEARCHING FOR A STRONG BLOCK
!

C!      WRITE(6,2)
C!2     FORMAT(1X,/,'STEP1',/)


C!      WRITE(6,*)LNUM
                DO IX=1,30
                        DO IY=1,30

                        BN=(LNUM-1)*900+(IX-1)*30+IY
                                IF(WSI(BN).EQ.1)THEN
                                   IF(WSII(BN).EQ.1)THEN
CCCC
C       WRITE(6,10)BN,LNUM,IX,IY
C10     FORMAT('POSI. BLOCK :',I10,' LEVEL:',I1,' X COOR:'I2,' Y COOR:',I2)
                                        IF(NTX.EQ.1)THEN
                                                INX(BN)=1
                                                CALL TUPD(BN,LNUM,IX,IY,NTX)
                                        ELSEIF(INX(BN).EQ.1)THEN
                                                CALL TUPD(BN,LNUM,IX,IY,NTX)
                                        ENDIF
                                    ENDIF
                                ENDIF
                        ENDDO
                ENDDO
```

```
!
          RETURN
          END
```

```
!
C***********************************************************************
C****                                                              *****
C****                   SUBROUTINE TUPD                            *****
C****                                                              *****
C**** THIS SUBROUTINE CONNECTS PREVIOUSLY IDENTIFIED POSITIVE      *****
C**** BLOCK BN TO THE OVERLYING WEAK NODE.                         *****
C****                                                              *****
C**** SUBROUTINES :                                                *****
C**** THE FOLLOWING 5 SUBROUTINES ARE CALLED IN THIS ROUTINE.      *****
C****           1) SUBROUTINE CONE                                 *****
C****           2) SUBROUTINE ROOT                                 *****
C****           3) SUBROUTINE ADD                                  *****
C****           4) SUBROUTINE CUMVAL                               *****
C****           5) SUBROUTINE NORM                                 *****
C****                                                              *****
C**** SUBROUTINE CONE DEFINES THE BLOCKS WITHIN THE CONE OF A      *****
C**** STRONG BLOCK BN AND AS A RESULTS IDENTIFIES A WEAK BRANCH    *****
C**** SUBROUTINE ROOT DETERMINES THE ROOT NODE OF A STRONG BRANCH  *****
C**** SUBROUTINE ADD ADDS AN ARC FROM STRONG NODE THE WEAK NODE    *****
C**** SUBROUTINE CUMVAL CALCULATES THE CUMULATIVE VALUES OF THE    *****
C**** NODES ON THE PATH FROM ROOT NODE OF THE STRONG TREE TO THE   *****
C**** DUMMY ROOT SO THAT NORMALIZATION STEP OF THE LERCHS AND      *****
C**** GROSSMANN'S ALGORITHM CAN BE CARRIED OUT BY THE NEXT ROUTINE *****
C**** SUBROUTINE NORM DOES THE NORMALIZATION                       *****
C****                                                              *****
C**** DEFINITION OF THE VARIABLES:                                 *****
C****    BN     : STRONG NODE                                      *****
C****    IX     : ROW NUMBER OF THE BLOCK BN                       *****
C****    IY     : COLUMN NUMBER OF BLOCK BN                        *****
C****    LNUM   : LEVEL NUMBER ON WHICH THIS CONNECTING STRONG     *****
C****             IS FOUND.                                        *****
C****    NTX    : CURRENT ITERATION NUMBER OF THE ALGORITHM        *****
C****                                                              *****
C***********************************************************************
       SUBROUTINE TUPD(BN,LNUM,IX,IY,NTX)
!
       INTEGER*2 S(-1:5400),P(-1:5400),YB(-1:5400)
       INTEGER*2 WSI(5400)
       INTEGER*2 BN
       INTEGER*2 CLOS(5400),IS(50)
       INTEGER*4 V(5400),CV(5400)
!
       COMMON/BLK1/V,CV,WSI
       COMMON/BLK2/S,P,YB
       COMMON/BLK4/CLOS
       COMMON/BLK14/IS
```

```
C!      WRITE(6,2)
C!2     FORMAT(1X,/,'TUPD',/)


!
!       DETERMINE THE BLOCKS WHICH FALL INTO THE CONE OF THE STRONG NODE
!       PREVIOUSLY IDENTIFIED
!
                IF(P(BN).EQ.0)THEN
                  IF(S(BN).NE.-1)THEN
                        RETURN
                  ENDIF
                ENDIF


                CALL CONE(LNUM,IX,IY,MINI,MAXI,MINJ,MAXJ)
C!      WRITE(6,5)(CLOS(I),I=1,900)
C!5     FORMAT(30(30I2,/),/)
!
!       SEARCH FOR THE OVERLYING WEAK NODES AND CONNECT THE STRONG NODE TO
!       WEAK NODES UNTIL IT CAN NO LONGER SUPPORT THE WEIGHTS OF THE WEAK NO
!
                DO K=2,LNUM
                        DO I=MINI,MAXI
                                DO J=MINJ,MAXJ
                                        N=(LNUM-K)*900+(I-1)*30+J
                                        IF(N.GT.5400)WRITE(6,*)N
                                        IF(N.LT.0)WRITE(6,*)N
                                        IF(WSI(N).EQ.0)THEN

                                                IF(CLOS(N).EQ.1)THEN
                                                        IS(NTX)=1
CCCC
C       WRITE(6,7)BN,N
C7      FORMAT(1X,'STRONG NODE=',I10,4X,'OVERLAYING WEAK NODE=',I4)
                                                        CALL ROOT(BN,N,MNOD)
                                                        CALL ADD(BN,N,MNOD)
                                                        CALL CUMVAL(MNOD)
                                                        CALL NORM(MNOD)
                                                ENDIF
                                        ENDIF
                                        IF(WSI(BN).NE.1)GOTO 10
                                ENDDO
                        ENDDO
                ENDDO
!
!
10      RETURN
        END
```

```
!
C***********************************************************************
C****                                                              *****
C****                   SUBROUTINE ROOT                            *****
C****                                                              *****
C**** THIS SUBROUTINE DETERMINES THE ROOT NODE OF A STRONG BRANCH  *****
C****                                                              *****
C**** DESCRIPTION OF VARIABLES:                                    *****
C****   BN      : STRONG NODE                                      *****
C****   N       : CURRENT CONNECTING WEAK NODE TO BN               *****
C****   MNOD    : ROOT NODE OF THE STRONG BRANCH                   *****
C****                                                              *****
C***********************************************************************
        SUBROUTINE ROOT (BN,N,MNOD)

        INTEGER*2 S(-1:5400),P(-1:5400),YB(-1:5400),IPATH(-1:5400)
        INTEGER*2 BN
!
        COMMON /BLK2/S,P,YB
        COMMON /BLK5/IPATH
!

C!      WRITE(6,2)
C!2     FORMAT(1X,/,'ROOT',/)

                K=BN
                IPATH(BN)=N
!
                        DO WHILE(P(K).NE.0)
                                IPATH(P(K))=K
                                K=P(K)
                        END DO
                MNOD=K

CCCC
C       WRITE(6,10)MNOD
C10     FORMAT(1X,'***ROOT NODE OF THE STRONG TREE***  ',I4)
!
        RETURN
        END
!
```

```
C***********************************************************************
C****                                                              *****
C****                   SUBROUTINE ADD                             *****
C****                                                              *****
C**** THIS SUBROUTINE CONNECTS STRONG NODE BN OF A STRONG BRANCH   *****
C**** TO OVERLAYING NODE N OF A WEAK BRANCH.                       *****
C****                                                              *****
C**** DESCRIPTION OF VARIABLES:                                    *****
C****   BN      : CURRENT STRONG NODE                              *****
C****   N       : OVERLAYING WEAK NODE                             *****
C****   MNOD    : ROOT NODE OF THE STRONG BRANCH                   *****
C****                                                              *****
C***********************************************************************
        SUBROUTINE ADD(BN,N,MNOD)

        INTEGER*2 S(-1:5400),P(-1:5400),YB(-1:5400),IPATH(-1:5400)
        INTEGER*2 BN,A,B,C
!
        COMMON /BLK2/S,P,YB
        COMMON /BLK5/IPATH
!

CCCC
C       WRITE(6,2)
C2      FORMAT(1X,' ADD')

!
        P(0)=0
        S(0)=-1
        YB(0)=-1
        P(-1)=0
        S(-1)=-1
        YB(-1)=-1
!
                K=MNOD
!
                        DO WHILE (K.NE.BN)
                                A=P(K)
                                B=S(K)
                                C=YB(K)
!
!       MODIFY THE INDECES OF ROOT NODE OF THE STRONG BRANCH
!
                        IF(K.EQ.MNOD)THEN
                                IF(S(K).EQ.IPATH(K))THEN
                                IF(YB(B).NE.-1)THEN
                                        S(K)=YB(B)
                                ELSE
                                        S(K)=-1
                                ENDIF
                                P(K)=IPATH(K)
                                ELSE
                                KX=S(K)
                                        DOWHILE(YB(KX).NE.IPATH(K))
                                                KX=YB(KX)
                                        ENDDO
                                        YB(KX)=YB(IPATH(K))
                                        S(K)=B
                                        P(K)=IPATH(K)
                                ENDIF
                        ENDIF
!
                        IF(K.NE.MNOD)THEN
                                IF(S(K).EQ.IPATH(K))THEN
                                        S(K)=A
                                        YB(S(K))=YB(B)
                                        P(K)=IPATH(K)
                                ELSE
                                        KK=S(K)
                                                DO WHILE(YB(KK).NE.IPATH(K))
                                                        KK=YB(KK)
                                                ENDDO
                                        YB(KK)=YB(IPATH(K))
                                        S(K)=A
                                        P(K)=IPATH(K)
                                        YB(S(K))=B
                                ENDIF
                        ENDIF

C!      WRITE(6,*)IPATH(K)

!

                        K=P(K)

                        ENDDO
!
                A=P(BN)
                B=S(BN)
                C=YB(BN)
!
                P(BN)=N
                S(BN)=A
                IF(S(BN).EQ.0)S(BN)=-1
                YB(BN)=S(N)
                S(N)=BN
                YB(S(BN))=B
CCCC
C       WRITE(6,*)P(BN),S(BN),YB(BN)
C       WRITE(6,*)P(N),S(N),YB(N)
        RETURN
        END
```

```
C***********************************************************************
C****                                                              *****
C****                   SUBROUTINE CUMVAL                          *****
C****                                                              *****
C**** THIS SUBROUTINE DETERMINES THE CUMULATIVE VALUES OF THOSE    *****
C**** NODES WHICH ARE ON THE PATH FROM ROOT NODE OF STRONG TREE   *****
C**** TO THE DUMMY ROOT.                                           *****
C****                                                              *****
C**** DESCRIPTION OF VARIABLES:                                    *****
C****   MNOD: ROOT NODE  OF STRONG BRANCH                          *****
C****                                                              *****
C***********************************************************************
        SUBROUTINE CUMVAL(MNOD)

        INTEGER*2 WSI(5400)
        INTEGER*2 S(-1:5400),P(-1:5400),YB(-1:5400)
        INTEGER*4 V(5400),CV(5400)
!
        COMMON /BLK1/V,CV,WSI
        COMMON/BLK2/S,P,YB
!
CCCC
C       WRITE(6,2)
C2      FORMAT(1X,'CUMVAL')
!
        P(0)=0
        S(0)=-1
        YB(0)=-1
        P(-1)=0
        S(-1)=-1
        YB(-1)=-1
!
!
                K=MNOD
!
                DO WHILE(K.NE.0)
                        IF(S(K).NE.-1)THEN
                                KK=S(K)
                                CV(K)=V(K)+CV(S(K))
                                DO WHILE(YB(KK).NE.-1)
                                        CV(K)=CV(K)+CV(YB(KK))
                                        KK=YB(KK)
                                ENDDO
                        ELSE
                                CV(K)=V(K)
                        ENDIF
                        K=P(K)
```

```
                    ENDDO
!
          RETURN
          END
!
```

```
:
C***********************************************************************
C****                                                              *****
C****                   SUBROUTINE NORM                            *****
C****                                                              *****
C**** THIS SUBROUTINE PERFORMS THE NORMALIZATION STEP OF LERCHS    *****
C**** AND GROSSMANN' ALGORITHM.                                    *****
C****                                                              *****
C**** SUBROUTINES:                                                 *****
C**** THE FOLLOWING SUBROUTINE ARE CALLED IN ORDER TO NORMALIZE    *****
C**** THE CURRENT TREE.                                            *****
C****           1) SUBROUTINE MEDGE                                *****
C****           2) SUBROUTINE PEDGE                                *****
C****           3) SUBROUTINE REDGE                                *****
C****                                                              *****
C**** DESCRIPTION OF VARIABLES:                                    *****
C****    MNOD   : ROOT NODE OF THE STRONG BRANCH                   *****
C****                                                              *****
C***********************************************************************
        SUBROUTINE NORM(MNOD)

        INTEGER*2 WSI(5400),A
        INTEGER*2 S(-1:5400),P(-1:5400),YB(-1:5400)
        INTEGER*4 V(5400),CV(5400)
!
        COMMON /BLK1/V,CV,WSI
        COMMON /BLK2/S,P,YB
!
CCCC
C       WRITE(6,2)
C2      FORMAT(1X,'NORM')
!
        P(0)=0
        S(0)=-1
        YB(0)=-1
        P(-1)=0
        S(-1)=-1
        YB(-1)=-1
!
!
                K=MNOD
!
10              IF(P(K).NE.0)THEN
!
                        IF(K.GT.P(K))THEN
                                A=P(K)
                                CALL MEDGE(K)
                                K=A
                                GOTO 10
```

```
                      ELSE
                              A=P(K)
                              CALL PEDGE(K)
                              K=A
                              GOTO 10
                      ENDIF
!
              ELSE
!
                      CALL REDGE(K)
!
              ENDIF
!
      RETURN
      END
```

```
!
C***********************************************************************
C****                                                              *****
C****                   SUBROUTINE MEDGE                           *****
C****                                                              *****
C**** THIS SUBROUTINE NORMALIZES A CURRENT TREE WHEN A STRONG      *****
C**** M-EDGE IS FOUND ON THE PATH FROM ROOT NODE OF THE STRONG     *****
C**** BRANCH TO THE DUMMY ROOT. THIS IS DONE BY DELETING THE       *****
C**** ORIGINATING NODE OF THE M-EDGE AND THEN CONNECTING THIS NODE *****
C**** TO THE DUMMY ROOT.                                           *****
C****                                                              *****
C**** DESCRIPTION OF VARIABLES:                                    *****
C****    K1     : ORIGINATING NODE OF THE M-EDGE                   *****
C****                                                              *****
C***********************************************************************
        SUBROUTINE MEDGE(K1)

        INTEGER*2 WSI(5400)
        INTEGER*2 S(-1:5400),P(-1:5400),YB(-1:5400)
        INTEGER*2 A,A1,A2
        INTEGER*4 V(5400),CV(5400)
!
        COMMON /BLK1/V,CV,WSI
        COMMON /BLK2/ S,P,YB
!
CCCC
C       WRITE(6,2)
C2      FORMAT(1X,' MEDGE')
!
        P(0)=0
        S(0)=-1
        YB(0)=-1
        P(-1)=0
        S(-1)=-1
        YB(-1)=-1
!
!
!
                IF(CV(K1).LE.0)THEN
                        K2=P(K1)
                        DO WHILE(K2.GT.0)
                                CV(K2)=CV(K2)-CV(K1)
                                K2=P(K2)
                        ENDDO

                        A=P(K1)
                        IF(S(A).NE.K1)THEN
                                KK1=S(A)
                                DO WHILE(YB(KK1).NE.K1)
                                        KK1=YB(KK1)
                                ENDDO
                                YB(KK1)=YB(K1)
                                IF(YB(KK1).LE.0)YB(KK1)=-1
                        ELSE
                                S(A)=YB(K1)
                                IF(S(A).LE.0)S(A)=-1
                        ENDIF
                        P(K1)=0
                        YB(K1)=-1
                        WSI(K1)=0
                        IVAL=WSI(K1)
                        CALL RESET(K1,IVAL)
                ENDIF
!
10      RETURN
        END
```

```
!
C***********************************************************************
C****                                                              *****
C****                   SUBROUTINE PEDGE                           *****
C****                                                              *****
C**** THIS SUBROUTINE NORMALIZES A CURRENT TREE WHEN A STRONG      *****
C**** P-EDGE IS FOUND ON THE PATH FROM ROOT NODE OF THE STRONG     *****
C**** BRANCH TO THE DUMMY ROOT. THIS IS DONE BY DELETING THE       *****
C**** ENDING NODE OF THE P-EDGE AND THEN CONNECTING THIS NODE      *****
C**** TO THE DUMMY ROOT.                                           *****
C****                                                              *****
C**** DESCRIPTION OF VARIABLES:                                    *****
C****    K1     : ENDING NODE OF THE P-EDGE                        *****
C****                                                              *****
C***********************************************************************
        SUBROUTINE PEDGE(K1)

        INTEGER*2 WSI(5400)
        INTEGER*2 S(-1:5400),P(-1:5400),YB(-1:5400)
        INTEGER*2 A,A1,A2,AA
        INTEGER*4 V(5400),CV(5400)
!
        COMMON /BLK1/V,CV,WSI
        COMMON /BLK2/S,P,YB
!

CCCC
C       WRITE(6,2)
C2      FORMAT(1X,' PEDGE')
!
        P(0)=0
        S(0)=-1
        YB(0)=-1
        P(-1)=0
        S(-1)=-1
        YB(-1)=-1
!
!
                IF(CV(K1).GT.0)THEN
                        K2=P(K1)
                        DO WHILE(K2.NE.0)
                                CV(K2)=CV(K2)-CV(K1)
                                K2=P(K2)
                        ENDDO
!
                        A=P(K1)
CCCC
```

```
CCCC
C                     TYPE*,K1,S(K1),YB(K1),A,S(A),YB(A)
                      IF(A.LE.0)THEN
                      TYPE *, ' ERROR IN PEDGE'
                      RETURN
                      ENDIF
                      IF(S(A).NE.K1)THEN
                              KK1=S(A)
                              IF(KK1.LE.0)THEN
                              TYPE *,' KK1 IN PEDGE IS WRONG'
                              RETURN
                              ENDIF
                              DO WHILE(YB(KK1).NE.K1)
                                      KK1=YB(KK1)
CCCC
C                                     TYPE*,'CHECKING THE YB',KK1
                                      IF(KK1.LE.0)THEN
                                      TYPE *,' CAN NOT FIND K1'
                                      RETURN
                                      ENDIF
                              ENDDO
                              YB(KK1)=YB(K1)
CCCC
C                             TYPE*,'YB(K1) IS YB(KK1)',YB(KK1)
                              IF(YB(KK1).LE.0)YB(KK1)=-1
                      ELSE
                              S(A)=YB(K1)
                              IF(S(A).LE.0)S(A)=-1
                      ENDIF
CCCC
C                     TYPE*,S(A)
C                     S(A)=YB(K1)
C                     IF(S(A).EQ.0)S(A)=-1
                      P(K1)=0
                      YB(K1)=-1
                      WSI(K1)=1
                      IVAL=WSI(K1)
                      CALL RESET(K1,IVAL)
!
              ENDIF
!
      RETURN
      END

!
```

```
:
C***********************************************************************
C****                                                              *****
C****                   SUBROUTINE REDGE                           *****
C****                                                              *****
C**** THIS SUBROUTINE UPDATES THE INDECES OF A TREE AFTER A        *****
C**** NORMALIZATION STEP BY CALLING RESET SUBROUTINE.              *****
C****                                                              *****
C**** DESCRIPTION OF VARIABLES :                                   *****
C****    K1     : ROOT NODE OF A NORMALIZED TREE                   *****
C****                                                              *****
C***********************************************************************
        SUBROUTINE REDGE(K1)

        INTEGER*2 WSI(5400)
        INTEGER*2 S(-1:5400),P(-1:5400),YB(-1:5400)
        INTEGER*2 A,A1,A2
        INTEGER*4 V(5400),CV(5400)
!
        COMMON /BLK1/V,CV,WSI
        COMMON /BLK2/S,P,YB
!
CCCC
C       WRITE(6,2)
C2      FORMAT(1X,' REDGE')


                IF(CV(K1).GT.0)THEN
                        WSI(K1)=1
                ELSE
                        WSI(K1)=0
                ENDIF

C!      WRITE(6,*)K1,CV(K1),WSI(K1)
!
                IVAL=WSI(K1)
                CALL RESET(K1,IVAL)
!
        RETURN
        END
!
```

```
!
!
C***********************************************************************
C****                                                              *****
C****                   SUBROUTINE RESET                           *****
C****                                                              *****
C**** THIS SUBROUTINE UPDATES WEAK STRONG INDECES OF A NORMALIZED  *****
C**** TREE BASED ON THE VALUE OF THE ROOT NODE.                    *****
C****                                                              *****
C**** DESCRIPTION OF VARIABLES :                                   *****
C****   NODE    : ROOT NODE OF THE NEWLY CREATED TREE              *****
C****   IVAL    : STRONG WEAK INDICATOR OF THIS ROOT NODE          *****
C****                                                              *****
C***********************************************************************
        SUBROUTINE RESET(NODE,IVAL)

        INTEGER*2 WSI(5400)
        INTEGER*2 S(-1:5400),P(-1:5400),YB(-1:5400)
        INTEGER*4 V(5400),CV(5400)
!
        COMMON /BLK1/V,CV,WSI
        COMMON /BLK2/S,P,YB
!
CCCC
C       WRITE(6,2)
C2      FORMAT(1X,'RESET')

        ICOUNT=0
        ICOUN=0
        P(0)=0
        S(0)=-1
        YB(0)=-1
        S(-1)=-1
        P(-1)=0
        YB(-1)=-1
CCCC
C       WRITE(6,*)NODE,S(NODE),IVAL,V(NODE),CV(NODE)
                IF(S(NODE).EQ.-1)THEN
                        WSI(NODE)=IVAL
                        GOTO 30
                ENDIF
                K=S(NODE)
10              IF(K.LE.0)GO TO 30
                WSI(K)=IVAL
                ICOUN=ICOUN+1
                IF(ICOUN.GT.100000)THEN
                TYPE *,' THIS IS IT 1'
        WRITE(6,*)K,WSI(K),CV(K)
                ENDIF
```

```
CCCC
C       WRITE(6,*)K,WSI(K),CV(K)
                IF(S(K).NE.-1)THEN
                        K=S(K)
                ELSEIF(YB(K).NE.-1)THEN
                        K=YB(K)
                ELSE
20                      K=P(K)
                        ICOUNT=ICOUNT+1
                        IF(ICOUNT.GT.10000)THEN
                        TYPE *,' THIS IS IT 2'
        WRITE(6,*)NODE,K,P(K),S(K),YB(K)
                        ENDIF
                        IF(K.LE.0)GO TO 30
                        IF(K.EQ.NODE)GOTO 30
                        IF(YB(K).EQ.-1)GOTO 20
                        K=YB(K)
                ENDIF
                GOTO 10
!
30      RETURN
        END
!
!
```

```
:
C************************************************************************
C****                                                              *****
C****                  SUBROUTINE CONE                             *****
C****                                                              *****
C**** THIS SUBROUTINE DETERMINES THOSE BLOCKS WHICH FALL INTO THE  *****
C**** CONE OF A STRONG NODE.                                       *****
C****                                                              *****
C**** DESCRIPTION OF VARIABLES :                                   *****
C****   IX     : X COORDINATE OF STRONG NODE                       *****
C****   IY     : Y COORDINATE OF STRONG NODE                       *****
C****   MINI   : STARTING X COORDINATE AT THE TOP LEVEL            *****
C****   MINJ   : STARTING Y COORDINATE AT THE TOP LEVEL            *****
C****   MAXI   : ENDING X COORDINATE AT THE TOP LEVEL              *****
C****   MAXJ   : ENDING Y COORDINATE AT THE TOP LEVEL              *****
C****   LNUM   : Z COORDINATE OF THE STRONG NODE                   *****
C****                                                              *****
C************************************************************************
        SUBROUTINE CONE(LNUM,IX,IY,MINI,MAXI,MINJ,MAXJ)

        INTEGER*2 CLOS(5400),DELTI,DELTJ
        REAL*4 H(-6:6,-6:6),HEIGHT
!
        COMMON /BLK3/H
        COMMON/BLK4/CLOS

C!      WRITE(6,2)
C!2     FORMAT(1X,/,'CONE',/)


!
!       INITILIZE ALL THE BLOCKS AS NOT IN THE CONE
!
                DO I=1,5400
                        CLOS(I)=0
                ENDDO
!
!
!       CALL THE NECESSARY SUBROUTINE TO DEFINE THE OUETER LIMITS OF X AND Y
!       COORDINATES OF THE CONE AT THE TOP LEVEL.
!
                CALL TOPLMT(LNUM,IX,IY,MINI,MAXI,MINJ,MAXJ)
C!              WRITE(6,10)LNUM,IX,IY,MINI,MAXI,MINJ,MAXJ
C!10            FORMAT(1X,3I3,4I5)
'
```

```
!
!       DETERMINE WHICH BLOCKS FALL INTO THE CONE OF A STRONG NODE.
!
                DO K=1,LNUM-1
                        DO I=MINI,MAXI
                                DO J=MINJ,MAXJ
                                        HEIGHT=LNUM-K
                                        DELTI=IX-I
                                        DELTJ=IY-J
                                        IF(HEIGHT.GE.H(DELTI,DELTJ))THEN
                                                N=(K-1)*900+(I-1)*30+J
                                                CLOS(N)=1
                                        ENDIF
                                ENDDO
                        ENDDO
                ENDDO
!
        RETURN
        END
```

```
C***********************************************************************
C****                                                              *****
C****                   SUBROUTINE TOPLMT                          *****
C****                                                              *****
C**** THIS SUBROUTINE DETERMINES OUTER LIMITS OF THE CONE AT UPPER *****
C**** MOST LEVEL IN TERMS OF I AND J COORDINATE INDECES.           *****
C****                                                              *****
C**** DESCRIPTION OF VARIABLES :                                   *****
C****   IX      : X COORDINATE OF STRONG NODE                      *****
C****   IY      : Y COORDINATE OF STRONG NODE                      *****
C****   MINI    : STARTING X COORDINATE AT THE TOP LEVEL           *****
C****   MINJ    : STARTING Y COORDINATE AT THE TOP LEVEL           *****
C****   MAXI    : ENDING X COORDINATE AT THE TOP LEVEL             *****
C****   MAXJ    : ENDING Y COORDINATE AT THE TOP LEVEL             *****
C****   LNUM    : Z COORDINATE OF THE STRONG NODE                  *****
C****                                                              *****
C***********************************************************************
        SUBROUTINE TOPLMT(LNUM,IX,IY,MINI,MAXI,MINJ,MAXJ)

        INTEGER*2 DELTI,DELTJ
        REAL*4 H(-6:6,-6:6),HEIGHT
!

        COMMON/BLK3/H
!
C!      WRITE(6,2)
C!2     FORMAT(1X,/,'TOPLMT',/)



                MINI=30
                MINJ=30
                MAXI=0
                MAXJ=0
                HEIGHT=LNUM-1
!
!       DETERMINE THE OUTER LIMITS OF X,Y COORDINATES AT THE TOP LEVEL
!
C!      WRITE(6,4)((H(I,J),J=-6,6),I=-6,6)
C!4     FORMAT(13(13F4.1,/),/)
C!      WRITE(6,*)HEIGHT,IX,IY
                DO I=1,30
                        DO J=1,30
                                DELTI=IX-I
                                IF(DELTI.LT.-6 .OR. DELTI.GT.6)GOTO 20
                                DELTJ=IY-J
                                IF(DELTJ.LT.-6 .OR. DELTJ.GT.6)GOTO 10
                                IF(HEIGHT.GE.H(DELTI,DELTJ))THEN
```

```
                                        IF(I.GT.MAXI)THEN
                                                MAXI=I
                                        ENDIF
                                        IF(I.LT.MINI)THEN
                                                MINI=I
                                        ENDIF
                                        IF(J.GT.MAXJ)THEN
                                                MAXJ=J
                                        ENDIF
                                        IF(J.LT.MINJ)THEN
                                                MINJ=J
                                        ENDIF
                                ENDIF
10                      ENDDO
20              ENDDO
!
        RETURN
        END
```

```
!
C***********************************************************************
C****                                                              *****
C****                   SUBROUTINE PRINT                           *****
C****                                                              *****
C**** THIS SUBROUTINE PRINTS THE LEVEL MAPS SHOWING WHICH BLOCKS   *****
C**** ARE ECONOMICALLY MINED IN THE ULTIMATE PIT LIMIT.            *****
C****                                                              *****
C**** DESCRIPTION OF VARIABLES :                                   *****
C****   NLEV    : NUMBER OF LEVELS TO BE PRINTED                   *****
C****   BI(I)   : MINED OR NOT MINED INDEX OF BLOCK I              *****
C****                                                              *****
C***********************************************************************
        SUBROUTINE PRINT(NLEV,BI)

        INTEGER*2 BI(5400)
!
!
C!      WRITE(6,2)
C!2     FORMAT(1X,/,'PRINT',/)

C               LEV=0
C               LBN1=1
C               LBN2=900
C               DO WHILE (LEV.LT.NLEV)
C                       WRITE(10,10)(BI(I),I=LBN1,LBN2)
C10                     FORMAT(1X,30(/1X,30('+---'),'+',/1X,'I ',30(I1,' I ')
C    +                  1X,30('+---'),'+'///)
C                       LEV=LEV+1
C                       LBN1=LEV*900+1
C                       LBN2=LBN1+899
C               END DO
!
C       WRITE(47,20)(BI(I),I=1,5400)
C20     FORMAT(30(30I4,/),/)
!

        RETURN
        END
```

```
C***********************************************************************
C****                                                              *****
C****                   SUBROUTINE OUTLINE                         *****
C****                                                              *****
C**** THIS SUBROUTINE PRINTS OUT THE TOPO FILE OF THE ULTIMATE     *****
C**** PIT FOUND.                                                   *****
C****                                                              *****
C**** DESCRIPTION OF VARIABLES :                                   *****
C****   NLEV    : NUMBER OF LEVELS TO BE PRINTED                   *****
C****   WSI(I)  : MINED OR NOT MINED INDEX OF BLOCK I              *****
C****                                                              *****
C***********************************************************************
        SUBROUTINE OUTLINE(NLEV,WSI)
        INTEGER*2 WSI(5400),IOUTL(900)

        WRITE(6,2)
2       FORMAT(1X,/,' OUTLINE',/)

                DO I=1,30
                        DO J=1,30
                                L=0
                                DO K=1,NLEV
                                        N=(K-1)*900+(I-1)*30+J
                                        IF(WSI(N).EQ.1)THEN
                                                IF(N.LT.900)THEN
                                                        L=1
                                                ELSEIF(N.LE.1800)THEN
                                                        L=2
                                                ELSEIF(N.LE.2700)THEN
                                                        L=3
                                                ELSEIF(N.LE.3600)THEN
                                                        L=4
                                                ELSEIF(N.LE.4500)THEN
                                                        L=5
                                                ELSEIF(N.LE.5400)THEN
                                                        L=6
                                                ENDIF
                                        ENDIF
                                ENDDO
                                IOUTL((I-1)*30+J)= L
                        ENDDO
                ENDDO

C               WRITE(10,10)(IOUTL(I),I=1,900)
C10                     FORMAT(1X,30(/1X,30('+---'),'+',/1X,'I ',30(I1,' I ')
C     +                         1X,30('+---'),'+'///)

        RETURN
        END
!
!
$
```

APPENDIX B

The Block Grades in % Cu

L1

```
TY FOR030.DAT
1  0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
2  0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
3  0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
4  0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
5  0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
6  0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
7  0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 6.1 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
22 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
                  5                                                                                   25
```

(5,7,1) - 6.1

l2

```
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 4.2 5.1 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 3.8 1.0 0.8 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.8 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 2.9 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.6 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 5.7 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
```

6

8

4 5 5

(6,7,2) 4.2
(6,8,2) 5.1
(7,7,2) 3.8
(7,8,2) 1.0
(7,9,2) 0.8
(8,8,2) 1.8

(9,8,2) 2.9
(10,9,2) 0.6
(11,9,2) 5.7

L3

(7,8,3) - 7.3
(8,7, 3) - 4.0
(8,8,3) - 5.6
(" 9, ") - 0.6
(9,7,3) - 6.3
(9,8,3) - 1.1
(10,8,3) - 1.7
(10,9,3) - 3.4
(11,9,3) - 5.6
(12,9,3) - 2.8

```
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 7.3 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 4.0 5.6 0.6 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 6.3 1.1 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.7 3.4 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 5.6 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 2.8 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
```

3

8

4

5

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.2	1.1	5.1	0.0	2.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.7	4.8	6.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.5	6.5	0.0	0.0	0.0	2.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	7.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

(5,8,4) - 6.3
(6,7,4) - 5.2
(6,8,4) - 1.1
(6,9,4) - 5.1
(6,11,4) - 2.9
(7,7,4) - 3.7
(7,8,4) - 4.8
(7,9,4) - 6.7
(8,8,4) - 1.5
(8,9,4) - 6.5
(8,13,4) - 2.3
(9,8,4) - 2.0
(9,9,4) - 7.3

L5

6

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.2	3.6	4.6	0.0	2.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	7.5	1.4	4.8	0.0	0.0	0.0	1.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.3	1.1	6.7	3.4	0.0	2.5	2.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.8	2.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

8

4

5

(7,7,5) - 5.2
(7,8,5) - 3.6
(7,9,5) - 4.6
(7,11,5) - 2.5
(8,7,5) - 7.5
(8,8,5) - 1.4
(8,9,5) - 4.8
(8,13,5) - 1.8
(9,7,5) - 2.3
(9,8,5) - 1.1
(9,9,5) - 6.7
(9,10,5) - 3.4
(9,12,5) - 2.5
(9,13,5) - 2.9
(10,8,5) - 2.8
(10,9,5) - 2.7

46

```
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 3.4 0.0 2.8 0.0 3.7 0.0 0.0 3.2 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 3.5 0.0 3.0 0.0 3.4 0.0 4.9 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.6 1.0 2.4 3.4 0.0 0.0 5.5 0.0 5.4 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 4.4 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
```

9

8

4

5

(7,7,6) - 3.4
(7,9,6) - 2.8
(7,11,6) - 3.7
(7,14,6) - 3.2
(8,7,6) - 3.5
(8,9,6) - 3.0
(8,11,6) - 3.4
(8,13,6) - 4.9
(9,7,6) - 1.6
(9,8,6) - 1.0
(9,9,6) - 2.4
(9,10,6) - 3.4
(9,13,6) - 5.5
(9,15,6) - 5.4

(10,14,6) - 4.4

Total: 15 blks

APPENDIX C

Discounted dollar values of the block on different levels. First six pages show the values for time period 1; pages 7 through 12 are for time period 2; remaining are for time period 3.

```
TY FOR032.DAT
-65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65
-65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65
-65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65
-65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65
-65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65
-65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65
-65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65
-65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65
-65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65
-65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65
-65 -65 -65 -65 -65 -65 -65 -65 -65 -655190 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65
-65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65
-65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65
-65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65
-65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65
-65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65
-65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65
-65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65
-65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65
-65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65
-65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65
-65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65
-65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65
-65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65
-65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65
-65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65
-65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65
-65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65
-65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65
-65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65
```

6

8

4

5

6

-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70
-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70
-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70
-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70
-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70
-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70
-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70
-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70
-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70
-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70
-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70
-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	3480	4290	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70
-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	3120	600	420	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70
-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	1320	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70
-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	2310	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70
-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	240	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70
-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	4830	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70
-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70
-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70
-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70
-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70
-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70
-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70
-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70
-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70
-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70
-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70
-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70
-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70
-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70	-70

8

4

5

-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75
-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75
-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75
-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75
-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75
-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75
-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75
-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75
-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75
-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75
-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75
-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75
-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	6270	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75
-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	3300	4740	240	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75
-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	5370	690	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75
-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	1230	2760	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75
-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	4740	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75
-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	2220	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75
-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75
-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75
-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75
-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75
-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75
-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75
-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75
-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75
-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75
-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75
-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75
-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75	-75

-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80
-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80
-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80
-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80
-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80
-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80
-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80
-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80
-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80
-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80
-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	5370	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80
-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	4380	690	4290	-80	2310	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80
-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	3030	4020	5730	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80
-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	1050	5550	-80	-80	-80	1770	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80
-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	1500	6270	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80
-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80
-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80
-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80
-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80
-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80
-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80
-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80
-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80
-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80
-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80
-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80
-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80
-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80
-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80
-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80

```
-85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85
-85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85
-85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85
-85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85
-85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85
-85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85
-85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85
-85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85
-85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85
-85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85
-85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85
-85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85
-85 -85 -85 -85 -85 -85 -85 -85 -85 -85 4380 2940 3840 -85 1950 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85
-85 -85 -85 -85 -85 -85 -85 -85 -85 -85 6450 960 4020 -85 -85 -85 1320 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85
-85 -85 -85 -85 -85 -85 -85 -85 -85 -85 1770 690 5730 2760 -85 1950 2310 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85
-85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 2220 2130 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85
-85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85
-85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85
-85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85
-85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85
-85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85
-85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85
-85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85
-85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85
-85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85
-85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85
-85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85
-85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85
-85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85
-85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85 -85
```

```
-90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90
-90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90
-90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90
-90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90
-90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90
-90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90
-90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90
-90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90
-90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90
-90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90
-90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90
-90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90
-90 -90 -90 -90 -90 -90 -90 -90 -90 -902760 -902220 -903030 -90 -902580 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90
-90 -90 -90 -90 -90 -90 -90 -90 -90 -902850 -902400 -902760 -904110 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90
-90 -90 -90 -90 -90 -90 -90 -90 -90 -901140 60018602760 -90 -904650 -904560 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90
-90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -903660 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90
-90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90
-90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90
-90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90
-90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90
-90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90
-90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90
-90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90
-90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90
-90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90
-90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90
-90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90
-90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90
-90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90
-90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90 -90
```

```
-57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57
-57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57
-57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57
-57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57
-57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57
-57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57
-57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57
-57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57
-57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57
-57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57
-57 -57 -57 -57 -57 -57 -57 -57 -57 -574613 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57
-57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57
-57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57
-57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57
-57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57
-57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57
-57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57
-57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57
-57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57
-57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57
-57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57
-57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57
-57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57
-57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57
-57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57
-57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57
-57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57
-57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57
-57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57
-57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57 -57
```

```
-62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62
-62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62
-62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62
-62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62
-62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62
-62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62
-62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62
-62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62
-62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62
-62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62
-62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62
-62 -62 -62 -62 -62 -62 -62 -62 -62 -6230930813 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62
-62 -62 -62 -62 -62 -62 -62 -62 -62 -622773 533 373 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62
-62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -621173 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62
-62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -622053 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62
-62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 213 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62
-62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -624293 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62
-62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62
-62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62
-62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62
-62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62
-62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62
-62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62
-62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62
-62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62
-62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62
-62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62
-62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62
-62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62
-62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62 -62
```

```
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -665573 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -6629334213 213 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -664773 613 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -6610932453 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -664213 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -661973 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
```

```
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -714773 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -713893 6133813 -712053 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71269335735093 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 9334933 -71 -71 -711573 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -7113335573 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
```

```
-75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75
-75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75
-75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75
-75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75
-75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75
-75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75
-75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75
-75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75
-75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75
-75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75
-75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75
-75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75
-75 -75 -75 -75 -75 -75 -75 -75 -75 -75389326133413 -751733 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75
-75 -75 -75 -75 -75 -75 -75 -75 -75 -755733 8533573 -75 -75 -751173 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75
-75 -75 -75 -75 -75 -75 -75 -75 -75 -751573 61350932453 -7517332053 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75
-75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -7519731893 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75
-75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75
-75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75
-75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75
-75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75
-75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75
-75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75
-75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75
-75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75
-75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75
-75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75
-75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75
-75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75
-75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75
-75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75 -75
```

```
-80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80
-80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80
-80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80
-80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80
-80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80
-80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80
-80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80
-80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80
-80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80
-80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80
-80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80
-80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80
-80 -80 -80 -80 -80 -80 -80 -80 -80 -802453 -801973 -802693 -80 -802293 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80
-80 -80 -80 -80 -80 -80 -80 -80 -80 -802533 -802133 -802453 -803653 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80
-80 -80 -80 -80 -80 -80 -80 -80 -80 -801013 53316532453 -80 -804133 -804053 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80
-80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -803253 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80
-80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80
-80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80
-80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80
-80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80
-80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80
-80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80
-80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80
-80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80
-80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80
-80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80
-80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80
-80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80
-80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80
-80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80 -80
```

```
-50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50
-50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50
-50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50
-50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50
-50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50
-50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50
-50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50
-50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50
-50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50
-50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50
-50 -50 -50 -50 -50 -50 -50 -50 -50 -504100 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50
-50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50
-50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50
-50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50
-50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50
-50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50
-50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50
-50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50
-50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50
-50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50
-50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50
-50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50
-50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50
-50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50
-50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50
-50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50
-50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50
-50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50
-50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50
-50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50 -50
```

```
-55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55
-55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55
-55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55
-55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55
-55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55
-55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55
-55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55
-55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55
-55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55
-55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55
-55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55
-55 -55 -55 -55 -55 -55 -55 -55 -55 -5527493389 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55
-55 -55 -55 -55 -55 -55 -55 -55 -55 -552464 473 331 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55
-55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -551042 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55
-55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -551824 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55
-55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 189 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55
-55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -553816 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55
-55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55
-55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55
-55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55
-55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55
-55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55
-55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55
-55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55
-55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55
-55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55
-55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55
-55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55
-55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55
-55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55 -55
```

```
-58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58
-58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58
-58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58
-58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58
-58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58
-58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58
-58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58
-58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58
-58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58
-58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58
-58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58
-58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58
-58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -584953 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58
-58 -58 -58 -58 -58 -58 -58 -58 -58 -5826073744 189 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58
-58 -58 -58 -58 -58 -58 -58 -58 -58 -584242 544 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58
-58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 9712180 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58
-58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -583744 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58
-58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -581753 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58
-58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58
-58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58
-58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58
-58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58
-58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58
-58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58
-58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58
-58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58
-58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58
-58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58
-58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58
-58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58 -58
```

```
-63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63
-63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63
-63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63
-63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63
-63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63
-63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63
-63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63
-63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63
-63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63
-63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63
-63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -634242 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63
-63 -63 -63 -63 -63 -63 -63 -63 -63 -633460 5443389 -631824 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63
-63 -63 -63 -63 -63 -63 -63 -63 -63 -63239331764527 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63
-63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 8294384 -63 -63 -631398 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63
-63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -6311844953 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63
-63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63
-63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63
-63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63
-63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63
-63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63
-63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63
-63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63
-63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63
-63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63
-63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63
-63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63
-63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63
-63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63
-63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63
-63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63 -63
```

```
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66346023223033 -661540 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -665096 7583176 -66 -66 -661042 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -661398 54445272180 -6615401824 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -6617531682 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
-66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66 -66
```

```
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -712180 -711753 -712393 -71 -712038 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -712251 -711896 -712180 -713247 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 900 47314692180 -71 -713673 -713602 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -712891 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
-71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71 -71
```

APPENDIX D

The level maps showing those blocks which are in the ultimate pit contour. Those blocks which are in the ultimate pit limits are indicated by '1'.

```
 TY FOR041.DAT

              NPVT OF THE SCHEDULE:  153770
                         TOTAL TONS      ORE TONS      AVERA GRADE
              PERIOD1       505             63              3.7
              PERIOD2        0               0              0.0
              PERIOD3        0               0              0.0
$ TY FOR043.DAT
1    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
3    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
4    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
5    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
6    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
7 ~~8~~ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0
     0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0
     0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0
     0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0
     0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0
     0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
22 ~~18~~ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

8

16

~~4~~ 1   1 ~~4~~

5   ← 21 →   25

Y

x

← 21 →

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

↑ 16 ↓

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

(8,7)

(9,15

(10,14

i

j

13

0 9 10