ECONOMIC-LOT-SIZE FORMULAS IN MANUFACTURING USING MATLAB

By

AYAD ABDULAZIZ MAHMOOD

A thesis submitted to the

Graduate School-New Brunswick

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Master of Science

Graduate Program in Operations Research

Written under the direction of

Michael Katehakis

And approved by

____

____

____

New Brunswick, New Jersey

January 2015

ABSTRACT OF THE THESIS

Economic-Lot-Size Formulas in Manufacturing Using MATLAB

By AYAD ABDULAZIZ MAHMOOD

Thesis Director:

Prof. Michael Katehakis

Economic-lot-size formulas enable us to get number of orders that reduce costs of each of: inventory, setup, and ordering. In manufacturing problems, formulas of economic-lot-size may give rise in unrealizable schedules, in some production periods the shop could not satisfy demands, whereas at other times slack periods that lead to idleness may occur. This research depicts a technique that allows the computation of order quantities in order that demands are satisfied with available labor and machines. The technique includes a simple step-by-step computation, particularly adaptable to computer programming (MATLAB).

Acknowledgement

I would express my deepest gratitude to my advisor, Prof. Michael N. Katehakis, for his outstanding guidance, suggestions, and caring.

I would like to thank my sponsor, the Higher Committee for Education Developments in Iraq (HCED), for their Financial and Academic supports, Advices, and all other facilities they offered to let me get the degree. Also, I would like to acknowledge, with much appreciation, the crucial role of my friends and my brother (Bashar) for their support during my scholarship duration.

Last, but not least, many thanks to my family and relatives for their great support, prayers, and their best wishes to me.

Table of contents

Abstract ii

Acknowledgement iii

Table of Contents iv

List of Tables v

List of Illustrations vi

Chapter One: Introduction 1

Chapter Two: Mathematical Model of the Problem 5

- **Feasible Solution Computing** 10
- **The General Method of Obtaining Feasible Solution** 18
- **The Inclusion of Set-Up Time** 23

Chapter Three: Practical Project (MATLAB) 25

- **First Method Code** 25
- **Second Method Code** 28

Chapter Four: Conclusion 31

References 34

List of Tables

Table 1: Demand matrix [*d*] 11

Table 2: Cumulative demand matrix [*D*] 11

Table 3: Run-time matrix [*T*] 11

Table 4: Available-hours matrix [*H*] 11

Table 5: Hours imposed by P_1 or matrix [$y_{n,1}^m$] 14

Table 6: Available hours for P_2, P_3, P_4, or matrix [$z_{n,1}^m$] 14

Table 7: The order-quantity matrix [*x*] 17

Table 8: The cumulative-order-quantity matrix [*X*] 17

Table 9: Labor-hours-imposed matrix [*h*] 17

List of Illustrations

Figure 1: A hypothetical production plan 2

Chapter One

Introduction

Consider a hypothetical corporation making a particular part to satisfy the demands shown in (Figure 1). By the end of February the demand is 30 parts, by the end of March the demand is 40 parts, by the end of April the demand is 65 parts, by the end of May the demand is 100 parts, and finally by the end of June the demand is 120 parts. The figure displays two possible production plans. The first plan determines that 80 parts be made in January and 40 parts be made in April; whereas the second plan proposes these parts should be made each month, 30 in February, 10 in March, 25 in April, 35 in May, and 20 in June. Actually, both of these production plans satisfy the demands, but Plan 1 includes only two set-ups, in January and April, whereas Plan 2 includes set-ups in each of the six months. Each set-up includes a cost (not only in setting up the machine, but also in paperwork). Obviously, Plan 1 includes less set-up cost than Plan 2. In other hand, Plan 1 includes a larger inventory and, as a result, a larger inventory-carrying cost. If we regard combined inventory and set-up cost, which of these two plans represent the lower cost? Also, does either of these plans represent the lowest possible cost and, if not, how could one determine the most economic production plan?

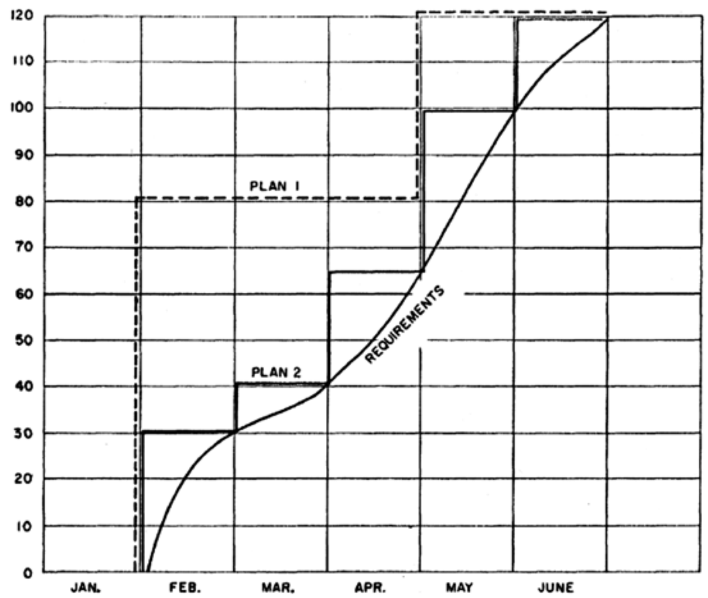

Figure 1: A hypothetical production plan.

The traditional economic-lot-size formula answers this question on condition that demands form a straight line. The formula is as follows

Q = $\sqrt{2DCs/pC}$

Where *Q* = order quantity (economic lot size)

D = yearly demand

Cs = set-up cost, C is the cost of the part

pC = annual inventory carrying cost

Indeed, in many manufacturing corporations, using this economic-lot-size formula is limited by the fact that the corporation participates in manufacturing many parts and not only one single part. As a result of this, the economic-lot-size formula may lead to an addition condition in some production periods, whereas in other production periods there may not be enough work. In most production cases, there are just a limited number of machines and workers available and, as a result, production must be planned within available machine capacity and workers. The purpose behind this research is to determine economic-lot-size formula that satisfies this case.

Our aim of this research, obtaining a solution to the following problem: Suppose a manufacturing corporation produces many parts to satisfy a given set of demands, and plans are made for future production periods. It is supposed that the available productive hours on each machine are known, that is the time required to make each part (including set-up time) is known, and that cost of both set-up and carrying inventory are known. The problem is that we have to determine order quantities for each part in order that the demands are satisfied and the cost of both set-ups and inventory carrying is minimized.

This formulation of the problem involves a particular case of traditional economic-lot-size formula. If there is an increment of machines and workers, then it is possible to plan for each part separately. If the demands for each part are given by a straight line, then the traditional economic-lot-size formula minimize costs of both carrying inventory and setting up. On the other hand, under these conditions the economic-lot-size formula gives a production plan that is achievable. As a result, our problem we are dealing with here is a generalization of economic-lot-size formula.

We shall display the problem firstly by defining its mathematical statement. Then we shall show that the problem results in a very large nonlinear, mathematical-programming problem. This nonlinear programming problem offers such an enormous suggestion that it is not yet possible to improve a general solution. However, we shall prove that it is possible to get a solution to this problem which satisfies demands and stay within available production capacities, but it may not necessarily minimize costs of both inventory and set-up. The computation method is simple and especially suitable for a computer programming. Although the fact that the computed solution may not be optimal, it may be possible to perform a series of computations and then get a relatively good solution. This research is to generalize the economic-lot-size formulas.

Chapter Two

Mathematical Model of the Problem

Consider a manufacturing corporation produces parts $P_1, P_2, P_3, \ldots$ on machines M_1, M_2, M_3, … . Our problem is that we have to make plans for future production periods π^1, π^2, π^3, … . Demands for each part in each production period are given, and we denote by d_i^m the demands for part P_i in production period π^m. Our problem is to compute the quantity of parts P_i to be made in production period π^m. Let us denote x_i^m to the order quantity of P_i to be made in production period π^m, then our problem is to evaluate this unknown x_i^m for every part in every production period. (We have to know that in some production periods some of these order quantities could be zero.) In order to facilitate the improvement of the mathematical statement of the problem, it will be helpful to deal with cumulative demands and order quantities. We denote by D_i^m the cumulative demands for part P_i up to and including period π^m, and by X_i^m the cumulative order quantity for part P_i. In a mathematical statement, we have

$$D_i^m = \sum_{k=1}^{m} d_i^m \quad \ldots\ldots\ldots. (1)$$

$$X_i^m = \sum_{k=1}^{m} x_i^m \quad \ldots\ldots\ldots. (2)$$

The demands must be satisfied by our production plan, this means that the cumulative order quantities must be larger or not less than the cumulative demands. In a mathematical statement, this can be written as follows inequality

$$X_i^m \geq D_i^m \quad \ldots\ldots\ldots. (3)$$

Now we have to deal with a mathematical statement of condition that the production plan must stay within available machine capacities. To put this condition into a mathematical form, we have to compute the machine-hour requirements enjoined by production plan. Making of each part P_i on machine M_n needs a particular number of hours, let us denote this by $T_{n,i}$ (This is run-time related to part P_i and machine M_n.) Also, to make this part on the machine, the machine requires set-up; let us denoted the set up time by $\sigma_{n,i}$. According to this notation, we can compute how large a load is enjoined on machine M_n by the making of a lot of P_i . x_i^m denotes the quantity of P_i must be made in π^m ; this quantity of parts needs the following number of hours to making :

Machine-hour needs = $T_{n,i}\, x_i^m + \sigma_{i,m}$

There is a remark about this formula that has to be watched. If we do not make any part in this period, then we have x_i^m=0. The formula would tell us that the machine-hour needs in this situation is $\sigma_{i,n}$, but this is in fact incorrect, since if there is no manufacturing for any part then there is no need for machine setting up. Therefore, we have to add this statement

Machine-hour needs = 0 if $x_i^m = 0$

Now, we introduce now a special notation for the "unit function" so that the above two statements will put in a single one. We say that the unit function $U(x)$ is given by

$$U(x) = 1 \text{ when } x>0 \text{ , } U(x) = 0 \text{ when } x=0$$

With the above notation we can say that

$$\text{Machine-hour needs} = T_{n,I}\, x_i^m + \sigma_{n,I}\, U(x_i^m)$$

Now we denote h_n^m to the number of hours needed for the machine M_n in production period π^m . To get these machine-hour needs, we have to consider all parts, thus we can write that

$$h_n^m = \sum_i T_{n,i}\, x_i^m + \sum_i \sigma_{n,i}\, U(x_i^m) \quad \text{.........(4a)}$$

Now we denote by H_n^m the number of hours available for machine M_n in period π^m . According to the production plan, the machine-hour needs must be within available hours, thus we have the following inequality:

$$h_n^m \leq H_n^m \quad \text{.........(4b)}$$

These last two equations can now be written as the following :

$$\sum_i T_{n,i}\, x_i^m + \sum_i \sigma_{n,I}\, U(x_i^m) \leq H_n^m \quad \text{.........(5)}$$

In other words, the equations (3) confirms that the production plan satisfies demands and equations (5) shows that the production plan must be within available machine hours. A production plan satisfying these conditions can be called a feasible production plan. To complete the mathematical statement of the problem, we shall determine the cost related to the production plan.

At first, we shall compute the inventory cost. The production plan determine that a cumulative quantity of X_i^m is made by production period π^m. Demands are just D_i^m. Which means that there will be inventory X_i^m - D_i^m . This includes an inventory carrying cost, which computed as follows

pC_i $(X_i^m - D_i^m)$ ……….(6)

Where pC_i now represents the carrying cost per part *i* for one production period. We can get the total inventory carrying cost by adding the inventory carrying cost of each part for each period. We denote by *Ic* the total inventory carrying cost, so we have the following equation

$Ic = p \sum_i \sum_m C_I (X_i^m - D_i^m)$ ……….(7)

Now, we shall compute set-up cost. Let $C_{n,i}$ denote the set-up cost of manufacturing part P_i on machine M_n . (This set-up cost involves the labor cost and the paperwork cost.) The total set-up cost is computed by following equations

$S_c = \sum_i \sum_n C_{n,I}\, U(x_i^m)$ ……….(8)

As we know, the unit function takes the value of 1 when there is a production order, and the value 0 when there is none. We obtain the total cost related to the production plan, by adding up inventory carrying and set-up costs, as follows

$Z = p\sum_i \sum_m C_I (X_i^m - D_i^m) + \sum_i \sum_n C_{n,i}\, U(x_i^m)$ ……….(9)

In mathematical language, we have to minimize the total cost Z, subject to the inequalities as expressed by equations (2), (3), and (5). This result in a linear-programming problem, except that the equation (5) is nonlinear as well as cost function. The reason is that these equations contain the unit function, which is a nonlinear function. So we have here a problem in nonlinear programing.

In order to estimate the size of the problem, we will consider a pragmatic example. Assume that we have to make 500 parts on 50 machines, and we are having a production plan for 10 production periods. In our equations system we have 500 unknowns in each production period, as a result we have 5000 unknowns. Equation (2) pertains the cumulative production quantities to the monthly production quantities. This equation have to be written for each part and each production period, as a result, the equation (2) represents 5000 equations. In the same manner, the equation (3) represents another 5000 equations. The condition of staying within available machine hours as written by equation (5) must be considered for each machine and each production period, thus we have here 500 inequalities. The total cost, as written by equation (9), consists of two terms for each part in each production period, thus, a total of 10000 terms. Now, we have 5000 unknowns, 10500 inequalities, and our problem is to minimize a cost function consisting of 10000 terms. Now, we know that even if our equations were linear we will have programming problem that can be solved by capacity of the largest electronic computer. But we have nonlinear equations, and we don't have a general known method to deal with such kind of equations. This implies that the solution of our problem, which minimize the cost function, is outside the available capabilities, and this should not inhibits us because in many production cases the most important thing is to get a feasible solution (one that satisfies demands and stay within available machine abilities). The motivating point is that such a feasible solutions can be got with relative case. Now we shall describe the computation related to getting a feasible solution to problem of production planning.

Feasible Solution Computation

Computation method can be best explained by hypothetical example. Consider a manufacturing corporation that is manufacturing four parts,P_1, P_2, P_3, and P_4 on three machines M_1, M_2, and M_3 . A production plan is to be done for five future production periods π^1, π^2, π^3, π^4, and π^5. Table 1 illustrates the demands for each part in each production period. It is shown that there are no demand in π^1 and π^2 for part $P_{1,}$ and for example, the demand is 25 for P_2 in production period π^4. The cumulative demands are given in Table 2. The production time (run-time) for each part in each machine is given in Table 3. The available hours on each machine for each production period are shown in Table 4. (It has to be known that hours available on individual machine do not change through periods, because in this plant the machines remain the same and have not changed. It is obvious that different machines may have different hours available. Generally, we might mean by M_1 a group of similar machines, by M_2 another group of similar machines, and so on.)

K \ J	π^1	π^2	π^3	π^4	π^5
P_1	0	0	30	10	20
P_2	0	5	10	25	20
P_3	0	20	15	15	20
P_4	5	35	0	0	0

Table 1: Demand matrix [*d*]

K \ J	π^1	π^2	π^3	π^4	π^5
P_1	0	0	30	40	60
P_2	0	5	15	40	60
P_3	0	20	35	50	70
P_4	5	40	40	40	40

Table 2: Cumulative demand matrix [*D*]

K \ i	P_1	P_2	P_3	P_4
M_1	2	3	0	1
M_2	4	0	2	2
M_3	0	5	5	2

Table 3: Run-time matrix [*T*]

K \ i	π^1	π^2	π^3	π^4	π^5
M_1	100	100	100	100	100
M_2	150	150	150	150	150
M_3	200	200	200	200	200

Table 4: Available-hours matrix [*H*]

Before we start the computation of a production plan which satisfies demands and stay within capacities, we have to know the set- up time for each part on each machine. However, we ignore these set-up times because it may be easier for computation when set-up times are ignored. Later, we will include the set-up times into the computation procedure which needs just a slight adjustment. The supposition of ignoring set-up time can be considered in another way. One could have experience from past rendering to foretell the number of machine hour available for production, and thus a particular number of hours may be specified for set-up.

Now we have to compute the unknown production quantities x_i^m to satisfy the demands as shown in Table 1 and we stay within available hours as shown in Table 4.

We maybe think of a question, why cannot we make the order quantities equal the demands? That is, why cannot we have:

$$x_i^m = d_i^m \quad ? \quad \text{..........} \ (10)$$

These order quantities satisfy demand, but do they stay within available productive machine hours? We cannot consider that this is the case. As an explanation, we will compute the number of hours required by M_1 in production period π^5. We have to use the equation (4) to obtain h_1^5;

$$h_1^5 = T_{1,1}\, d_1^5 + T_{1,2}\, d_2^5 + T_{1,3}\, d_3^5 + T_{1,4}\, d_4^9 \quad \text{..........}(11)$$

(Remark that we are neglecting set-up time). By substituting the numerical values we obtain a machine load of 140 hours. But we have just 100 hours available and, as a result, we are not within the abilities of M_1 (Therefore, we have to make some of parts earlier than the fifth production period to satisfy the demands and stay within available capacities.)

We suggest a method to computing order quantities. This method consists of a step-by-step computation beginning at the last production period. Now, consider only the schedule for P_1. We need 60 P_1's by the end of the fifth production period, whereas we need 40 P_1's by the end of the fourth production period. If we do not make anything just P_1's, how many of these parts can we manufacture in the fifth production period? To answer this question, we have to know the machine loads enjoined in the fifth production period, which are given as follows

$$h_1^4 = 2\, x_1^4,\ h_2^4 = 4\, x_1^4,\ h_3^4 = 0 \quad \text{..........(12)}$$

We must be within capacities and, therefore, we must have

$$h_1^4 \leq 100,\ h_2^4 \leq 150,\ h_3^4 \leq 200 \quad \text{..........(13)}$$

From above equations, it follows that

$$x_1^4 \leq 50\ ,\ x_1^4 \leq 150/4 = 37 \quad \text{..........(14)}$$

It is shown that we cannot make more than 37 P_1's in the fourth production period. However, we want just 20, and therefore, we manufacture 20 or put $x_1^4 = 20$. Now, we perform the same reasoning for making P_1's in the fourth production period. It is obvious to show that we could make 37 parts, but we want just 10; so we manufacture 10, therefore $x_1^4 = 10$. The same situation in the third production period, we could manufacture 37 of P_1's but we want just 30, so we put $x_1^3 = 30$. Now, all order quantities for P_1 are known, and so we satisfy the demand. Now, we are going to deal with P_2 and its computation.

We have to consider that we have come to an agreement to manufacture P_1's and therefore we do not have the productive hours available as shown in Table 4. In order to compute how many hours we still have, we will determine the hours enjoined by P_1. This is shown in Table 5 and it was obtained with the help of first column of Table 3, which shows machine-hour needs (run-time) for P_1. Now we subtract the hours expended from Table 5 from the hours available in Table 4, so we obtain the hours still available in Table 6.

K / i	π^1	π^2	π^3	π^4	π^5
M_1	0	0	60	20	40
M_2	0	0	120	40	80
M_3	0	0	0	0	0

Table 5: Hours imposed by P_1 or matrix $[y_{n,1}{}^{m}]$.

K / i	π^1	π^2	π^3	π^4	π^5
M_1	100	100	40	80	60
M_2	150	150	30	110	70
M_3	200	200	200	200	200

Table 6: Available hours for P_2, P_3, P_4, or matrix $[z_{n,1}{}^{m}]$.

Now, there is a question: how many P_2's can we manufacture in the fifth production period? The load enjoined on M_1 must be within capacity, such that

$$3\, x_2^5 \leq 60 \quad \text{..........(15)}$$

P_2 does not include Machine 2, so there is no limitation there. But we must stay within limitation on M_3 and, thus, we must have

$$5\, x_2^5 \leq 200 \quad \text{..........(16)}$$

From equation (15) we conclude that we cannot make more than 20 of P_2 in the fifth production period. Now, we will check Table 2, the cumulative demands for P_2. We see that we want 60 parts of P_2 by the end of the fifth production period, whereas we want 40 of P_2 by the end of the fourth production period. Therefore, we have to make 20 of P_2 in the fifth production period. This mean that $x_2^5 = 20$, and the cumulative order quantity for the fifth and fourth periods are

$$X_2^5 = 60 \quad \text{..........(17)}$$

$$X_2^4 = 40 \quad \text{..........(18)}$$

Now, we will consider the order quantities P_2 in fourth production period. We must not overtake machine capacities on M_1

$$3\, x_2^4 \leq 80 \quad \text{..........(19)}$$

And the same on M_3

$$5\, x_2^4 \leq 200 \quad \text{..........(20)}$$

This implies that we can manufacture 26 of P_2 in the fourth production period, but we need only 25. Therefore, $x_2^4 = 25$ ……….(21)

The cumulative order quantity for P_2 is

$X_2^4 = 40$ ……….(22)

With the same manner, we can determine the rest of order quantities for P_2 , and obtain

$x_2^3 = 10$, $X_2^3 = 15$, $x_2^2 = 5$, $X_2^2 = 5$

Now, we have completed the computations of order quantities for P_1 and P_2. Next, we will compute the machine load enjoined by P_2 and subtract from the original available productive hours the combined load enjoined by P_1 and P_2. This gives us the hours available for making P_3 and P_4 . After that, we continue and determine step-by-step the order quantities for P_3. Afterwards, we again compute the available machine hours for P_4 by subtracting the load imposed by P_1, P_2, and P_3. Then we end our computation by determining step-by-step the order quantities for P_4 . The computed order quantities for the different parts are shown Table7, and, the cumulative order quantities are shown in Table 8. It is clear that see that these order quantities satisfy the demand. Table 9 determines the total labor imposed and it is observed that these productive machine hours are staying within available productive hours as shown in Table 4.

Indeed, it is possible that there is no schedule that satisfies the demands within available capacities. On the other hand, with some other computation methods, it is possible that a feasible schedule could be obtained. This problem cannot be resolved by our method and there is no substitution in this case except to go back to the gigantic non-linear-programming problem as explained in our research.

J \ K	π^1	π^2	π^3	π^4	π^5
P_1	0	0	30	10	20
P_2	0	5	10	25	20
P_3	0	20	15	15	20
P_4	5	35	0	0	0

Table 7: The order-quantity matrix [*x*].

J \ K	π^1	π^2	π^3	π^4	π^5
P_1	0	0	30	40	60
P_2	0	5	15	40	60
P_3	0	20	35	50	70
P_4	5	40	40	40	40

Table 8: The cumulative-order-quantity matrix [*X*].

i \ K	π^1	π^2	π^3	π^4	π^5
M_1	5	50	90	95	100
M_2	10	110	150	70	120
M_3	10	195	125	200	200

Table 9: Labor-hours-imposed matrix [*h*].

Now, we have completed a procedure of getting a solution to our problem (under condition that we ignored set-up time and that there is no need for initial inventory). If the problem is not too large, then a solution can be gotten by hand computation. But if the problem gets really large, for example, if we make 500 parts, in 10 production period, and 50 machines, then hand computation will be difficult. Fortunately, the solution of this problem can be obtained through a program in MATLAB (we will present this program). Also, we will present a more general mathematical formulation of this computation technique.

The General Method of Obtaining Feasible Solution

At first, we want to commute equation (5) by an equation which deals with the cumulative schedules (we still are neglecting set-up time) or

$$\sum_i T_{n,i} (X_i^m - X_i^{m-1}) \leq H_n^m \quad ....(23)$$

Now, our problem is to get a positive solution to the equation (23) under condition that

$$X_i^m \geq X_i^{m-1} \quad ....(24)$$

(The last equation implies that the cumulative schedule does not lessen.) We begin our method by neglecting all other parts, but P_1 and by taking the port of the equation (23) which relates to P_1. Thus, we get the following inequality

$$T_{n,1} (X_1^m - X_1^{m-1}) \leq H_n^m \quad ....(25)$$

Suppose that we do not design to have a final inventory:

$$X_i^m = D_i^m \quad \text{..........(26)}$$

(If we wish to have a final inventory, we have to change the last equation.) Now, we will compute X_1^{m-1}. We compute the maximum number of P_1's that could be manufactured by determining the hours enjoined on the machines. We get the following inequality:

$$X_1^m - X_1^{m-1} \leq H_n^m / T_{n,1} \quad (n = 1,2,\ldots,N) \quad \text{..........(27)}$$

Last equation can be written in a this form

$$X_1^m - X_1^{m-1} \leq \min_n [H_n^m / T_{n,1}] \quad \text{..........(28)}$$

This means that the left-hand side must be smaller than the smallest of the different terms on the right-hand side. Now, we compute this with our previous numerical example. In equation (13) we had 100, 150, and 200 hours for each of the machines, dividing them by run-time 2, 4, and 0 we got 50, 37, and infinity. According to equation (28), we picked the smallest of these numbers. We can rewrite equation (28) in a different form:

$$X_1^{m-1} \geq X_1^m - \min_n [H_n^m / T_{n,1}] \quad \text{..........(29)}$$

Last inequality implies that X_1^{m-1} has to be larger than the number shown on the right-hand side. According to the demands X_1^{m-1} must also be larger than D_1^{m-1}, therefore we have

$$X_1^{m-1} \geq D_1^{m-1} \quad \text{..........(30)}$$

In our method, we usually picked X_1^m, the largest possible; this implies that we take X_1^{m-1}, the smallest possible. In other words, we obtained X_1^{m-1} by picking the larger of the two numbers appearing on the right-hand side of equations (29) and (30). In mathematical statement, this can be written as follows

$$X_1^{m-1} = \max \{ [X_1^m - \min_n (H_n^m/T_{n,1})] ; D_1^{m-1} \} \quad \text{..........(31)}$$

In our particular numerical problem we had

$$\min (H_n^m / T_{n,1}) = 37 \quad \text{..........(32)}$$

The cumulative demand for P_1 at the end of the fifth period was 60, and thus we have

$$X_1^m = D_1^m = 60 \quad \text{..........(33)}$$

The cumulative demand for P_1 at the end of the fourth period was 40, and thus we have

$$D_1^{m-1} = 40 \quad \text{..........(34)}$$

We get the cumulative schedule by the help of equation (31), which in the present situation, we have

$$\text{Max} \{ [60 - 37] ; 40 \} = 40 \quad \text{..........(35)}$$

As a result, we obtained $X_1^{m-1} = 40$ ....(36)

In fact, we see then, that our equation (31) is the same as the method we have already used. Exactly similarly, we can determine, step-by-step, the schedule of P_1 for the remaining of the production periods by working backward with the help of the following relation

$$X_1^{m-1} = \max \{ [X_1^m - \min (H_n^m / T_{n,1})] ; D_1^{m-1}\}$$

$$(m = m, m-1, m-2, \ldots) \quad \ldots\ldots\ldots.(37)$$

Now we have a general mathematical expression to get the schedule for P_1. In order to compute the schedule for $P1$. We have to compute the machine hours enjoined by P_1. Let $Y_{n,1}^m$ denote the load enjoined on machine M_n in production period π^m . (The second index 1 means that this is the load enjoined by schedule only part one.) Therefor, we have

$$Y_{n,1}^m = T_{n,1}\, x_1^m \quad \ldots\ldots\ldots.(38)$$

Now, we can determine the hours that are still available with the help of the formula

$$Z_{n,1}^m = H_n^m - Y_{n,1}^m \quad \ldots\ldots\ldots.(39)$$

Now, we can employ equation (37), but we have to change the originally available hours by the balance of the hours. Thus, we obtained

$$X_2^{m-1} = \max \{ [X_2^m - \min (z_{n,1}^m / T_{n,2})] ; D_2^{m-1} \} \quad \ldots\ldots\ldots.(40)$$

This is the schedule for P_2. Next, we determine the hours enjoined on the machines by P_1 and P_2 with the help of the formula

$$Y_{n,2}^m = T_{n,1}\, x_1^m + T_{n,2}\, x_2^m \quad \ldots\ldots\ldots.(41)$$

Now, the balance of available hours can be determined as follows

$z_{n,2}^{m} = H_{n}^{m} - Y_{n,2}^{m}$ ……….(42)

By similar way, we can compute the schedule for P_3 with the help of the formula

$X_3^{m-1} = \max \{ [X_3^{m} - \min(z_{n,2}^{m} / T_{n,3})] ; D_3^{m-1} \}$ ……….(43)

For generalization, after k steps (that is, after determining the schedule for P_1, P_2, …, P_k) we impose on the machine the load

$Y_{n,k}^{m} = \sum_{j=1}^{j=k} T_{n,j}\, x_j^{m}$ ……….(44)

Now, the available productive hours are computed as following

$z_{n,k}^{m} = H_{n}^{m} - Y_{n,k}^{m}$ ……….(45)

The schedule for P_{k+1} is determined by

$X_{k+1}^{m-1} = \max \{ [X_{k+1}^{m} - \min (z_{n,k}^{m} / T_{n,k+1})] ; D_{k+1}^{m-1} \}$ ……….(46)

In brief, we have here a complete system of equation, which, step by step, offer a solution to our scheduling problem. This system of equations is called in mathematics an algorithm. The problem of putting machine-shop scheduling on a computer can be solved this algorithm on a computer program (by MATLAB).

In the solution method we have presented, we determined first the schedule of P_1, then the schedule of P_2, and so on. Also, we can use an alternative method of computation which is to begin with the schedule of P_1 for the last production period, then obtained the schedule for P_2 for the last production period, and so on, and complete the schedules for all the parts in the last production period. Then we proceed to the last but one production period, and determine the schedule there, and so on. The amount of computation needed in both methods is the same. However, there may be some workable reason to prefer one of the two methods.

The Inclusion of Set-Up Time

Finally, we will turn our attention to the problem of how to include set-up times in our computation. Indeed, since our method of determining order quantities through step-by-step computation, we can make a slight modification of our method to include set-up time. As an explanation, let us go back to the hypothetical corporation making four parts on three machines. We will again compute the order quantity x_1^5. Assume that the set-up time for P_1 is 0.5 hours on M_1 and 1.2 hours on M_2. In order to stay within capacity we need to change equations (12) and (13) by

$$2x_1^5 + 0.5 \leq 100\,, \quad 4x_1^5 \leq 150 \quad \text{..........(47)}$$

We know that, according to Table 1 we should make 20 of P_1 in fifth production period. Similarly, we can compute x_1^4 and x_1^3, and in each case apposition must be made for set-up time. Next step is that, we must subtract the time required to make P_1 from the available machine hours. This again must involve both run-time and set-up time. Then we have to compute step-by-step the order quantities for P_2, followed by computation for P_3, and finally for P_4.

We can include this modification of computation technique into the general formulation of our algorithm. For example, equation (25) can be changed by

$$\sigma_{n,1} + T_{n,1} (X_i^m - X_i^{m-1}) \leq H_n^m \quad \text{..........(48)}$$

And the equation (31) can be changed by

$$X_1^{m-1} = \max \{ [X_1^m - \min(H_n^m - \sigma n,1) / T_{n,1}] ; D_1^{m-1} \} \quad \text{..........(49)}$$

A similar modification can be done in equation (37).

We must also modify the formulas that give the load enjoined on different machines. For example, equation (38) is replaced by

$$Y_{n,1}^m = \sigma_{n,1} + T_{n,1}\, x_1^m \quad \text{..........(50)}$$

Indeed the inclusion of set-up times in determining order quantities can be done easily.

Chapter Three

Practical Project (MATLAB)

This chapter presents two MATLAB codes to solve our previous numerical example as follows:

```
%First Method Code
D = [ 0 0  30 10 20 ;
     0 5  10 25 20 ;
     0 20 15 15 20 ;
     5 35 0  0  0 ];
CD = [ 0 0  30 40 60 ;
      0 5  15 40 60 ;
      0 20 35 50 70 ;
      5 40 40 40 40 ];
T = [ 2 3 0 1 ;
     4 0 2 2 ;
     0 5 5 2 ];
H = [ 100 100 100 100 100 ;
     150 150 150 150 150 ;
     200 200 200 200 200 ];
H_org = H;
x = zeros(4,5);

for k = 1 : 4
v = [999 999 999];
for j = 5 : -1 : 1
for i=1:3
  if (T(i,k) ~= 0 && H(i,j) ~= 0 )
     v(i) = floor(H(i,j)/T(i,k));
  end
end
minValue = min(v)
   if(minValue >= D(k,j))
      x(k,j) = D(k,j)
   end
end

H1 = T(:,k)*x(k,:);
H = H - H1
end
H_Laber_Hours_Imposed = H_org - H
```

After run the above code, we got the following results:

```
minValue =

   37

x =
    0    0    0    0   20
    0    0    0    0    0
    0    0    0    0    0
    0    0    0    0    0

minValue =
   37

x =
    0    0    0   10   20
    0    0    0    0    0
    0    0    0    0    0
    0    0    0    0    0

minValue =
   37

x =
    0    0   30   10   20
    0    0    0    0    0
    0    0    0    0    0
    0    0    0    0    0

minValue =
   37

x =
    0    0   30   10   20
    0    0    0    0    0
    0    0    0    0    0
    0    0    0    0    0

minValue =
   37

x =
    0    0   30   10   20
    0    0    0    0    0
    0    0    0    0    0
    0    0    0    0    0

H =
  100  100   40   80   60
  150  150   30  110   70
  200  200  200  200
200

minValue =
   20

x =
    0    0   30   10   20
    0    0    0    0   20
    0    0    0    0    0
    0    0    0    0    0

minValue =
   26

x =
    0    0   30   10   20
    0    0    0   25   20
    0    0    0    0    0
    0    0    0    0    0

minValue =
   13

x =
    0    0   30   10   20
    0    0   10   25   20
    0    0    0    0    0
    0    0    0    0    0

minValue =
   33

x =
    0    0   30   10   20
    0    5   10   25   20
    0    0    0    0    0
    0    0    0    0    0

minValue =
   33

x =
    0    0   30   10   20
    0    5   10   25   20
    0    0    0    0    0
    0    0    0    0    0

H =
  100   85   10    5    0
  150  150   30  110   70
  200  175  150   75  100

minValue =
   20

x =
    0    0   30   10   20
    0    5   10   25   20
    0    0    0    0   20
    0    0    0    0    0

minValue =
   15

x =
    0    0   30   10   20
    0    5   10   25   20
    0    0    0   15   20
    0    0    0    0    0

minValue =
   15

x =
    0    0   30   10   20
    0    5   10   25   20
    0    0   15   15   20
    0    0    0    0    0

minValue =
   35

x =
    0    0   30   10   20
    0    5   10   25   20
    0   20   15   15   20
    0    0    0    0    0
```

```
minValue =
   40

x =
    0    0   30   10   20
    0    5   10   25   20
    0   20   15   15   20
    0    0    0    0    0

H =
  100   85   10    5    0
  150  110    0   80   30
  200   75   75    0    0
minValue =
   15

x =
    0    0   30   10   20
    0    5   10   25   20
    0   20   15   15   20
    0    0    0    0    0

minValue =
    5

x =
    0    0   30   10   20
    0    5   10   25   20
    0   20   15   15   20
    0    0    0    0    0

minValue =
   10

x =
    0    0   30   10   20
    0    5   10   25   20
    0   20   15   15   20
    0    0    0    0    0

minValue =
   37

x =
    0    0   30   10   20
    0    5   10   25   20
    0   20   15   15   20
    0   35    0    0    0

minValue =
   75

x =
    0    0   30   10   20
    0    5   10   25   20
    0   20   15   15   20
    5   35    0    0    0

H =
   95   50   10    5    0
  140   40    0   80   30
  190    5   75    0    0


H_Laber_Hours_Imposed
=
    5   50   90   95  100
   10  110  150   70  120
   10  195  125  200  200
```

```
%Second Method Code

CD = [ 0 0  30 40 60 ;
       0 5  15 40 60 ;
       0 20 35 50 70 ;
       5 40 40 40 40 ];

T = [ 2 3 0 1 ;
      4 0 2 2 ;
      0 5 5 2 ];

H = [ 100 100 100 100 100 ;
      150 150 150 150 150 ;
      200 200 200 200 200 ];

H_org = H;

X = zeros(size(CD));

for k = 1 : 4
    v = [999 999 999];
    for j = 5 : -1 : 2
        X(k,5) = CD(k,5)
        for i=1:3
            if (T(i,k) ~= 0 && H(i,j) ~= 0 )
                v(i) = floor(H(i,j)/T(i,k));
            end
        end
        minValue = min(v)
        X(k,j-1) = max(X(k,j)-minValue, CD(k,j-1));
    end
  x = [X(k,1) X(k,2)-X(k,1) X(k,3)-X(k,2) X(k,4)-X(k,3) X(k,5)-X(k,4)]
    H1 = T(:,k)*x;
    H = H - H1
end
X
H_Laber_Hours_Imposed = H_org - H
```

After run the above code, we got the following results:

```
X =
    0    0    0    0   60
    0    0    0    0    0
    0    0    0    0    0
    0    0    0    0    0

minValue =
   37

X =
    0    0    0   40   60
    0    0    0    0    0
    0    0    0    0    0
    0    0    0    0    0

minValue =
   37

X =
    0    0   30   40   60
    0    0    0    0    0
    0    0    0    0    0
    0    0    0    0    0

minValue =
   37

X =
    0    0   30   40   60
    0    0    0    0    0
    0    0    0    0    0
    0    0    0    0    0

minValue =
   37

x =
    0    0   30   10   20

H =
  100  100   40   80   60
  150  150   30  110   70
  200  200  200  200  200
```

```
X =
    0    0   30   40   60
    0    0    0    0   60
    0    0    0    0    0
    0    0    0    0    0

minValue =
   20

X =
    0    0   30   40   60
    0    0    0   40   60
    0    0    0    0    0
    0    0    0    0    0

minValue =
   26

X =
    0    0   30   40   60
    0    0   15   40   60
    0    0    0    0    0
    0    0    0    0    0

minValue =
   13

X =
    0    0   30   40   60
    0    5   15   40   60
    0    0    0    0    0
    0    0    0    0    0

minValue =
   33

x =
    0    5   10   25   20

H =
  100   85   10    5    0
  150  150   30  110   70
  200  175  150   75  100
```

```
X =
    0    0   30   40   60
    0    5   15   40   60
    0    0    0    0   70
    0    0    0    0    0

minValue =
   20

X =
    0    0   30   40   60
    0    5   15   40   60
    0    0    0   50   70
    0    0    0    0    0

minValue =
   15

X =
    0    0   30   40   60
    0    5   15   40   60
    0    0   35   50   70
    0    0    0    0    0

minValue =
   15

X =
    0    0   30   40   60
    0    5   15   40   60
    0   20   35   50   70
    0    0    0    0    0

minValue =
   35

x =
    0   20   15   15   20

H =
  100   85   10    5    0
  150  110    0   80   30
  200   75   75    0    0
```

```
X =
    0    0   30   40   60
    0    5   15   40   60
    0   20   35   50   70
    0    0    0    0   40

minValue =
   15

X =
    0    0   30   40   60
    0    5   15   40   60
    0   20   35   50   70
    0    0    0   40   40
minValue =
    5

X =
    0    0   30   40   60
    0    5   15   40   60
    0   20   35   50   70
    0    0   40   40   40

minValue =
   10

X =
    0    0   30   40   60
    0    5   15   40   60
    0   20   35   50   70
    0   40   40   40   40

minValue =
   37

x =
    5   35    0    0    0

H =
   95   50   10    5    0
  140   40    0   80   30
  190    5   75    0    0

X =
    0    0   30   40   60
    0    5   15   40   60
    0   20   35   50   70
    5   40   40   40   40
```

```
H_Laber_Hours_Imposed =
    5   50   90   95  100
   10  110  150   70  120
   10  195  125  200  200
```

Chapter Four

Conclusion

In this research, we have formulated the non-linear programming problem of how to compute production order quantities that satisfies demands, stay within machine abilities, and minimize costs of both set-up and inventory. Also, we have depicted a computation technique that computes the production order quantities that satisfies demands and stay within machine hours, but that could not necessarily minimize the combined costs of set-up and inventory. We want to say that these production order quantities still could not necessarily form a realizable production schedule. At first, it may be possible that because of the specific sequence in which these parts have to be made not all the available hours on machines can be used. If we have 40 hours of work in a week for specific machine, and all this work has to be carried out during Thursday and Friday, then we would have unrealizable schedule. The problem of scheduling includes a detailed report of when each specific part goes on each specific machine. This scheduling problem is extremely difficult. There are no general method prepared for solving scheduling problem for job-shop kind of production.

It is significant to appreciate also that the solution of scheduling problem can be formulated just if the statistical nature of the problem is known. Indeed, it is known for practical production people that schedules are seldom satisfied as they are determined. There are usually troubles that happen in a plant, which makes it unwieldy to deal with schedules in detail. Machines wrack, workers may be truant, raw materials may be not available or may be late, the paperwork necessary for performing production may not be ready, or the tools needed to make apart may not be available. These different troubles form a fundamental element of the real problem of scheduling and must be involved in a study of the scheduling problem. We know that the scheduling problem is actually a problem in statistical-decision theory, and can be considered as a widely generalized form of the so-called Theory of Waiting Lines. There has been some progress here.

In this research, we have dealt with the cost of both carrying inventory and set-ups. However, we could know that these are just some of costs included in production, and also may be these cost are not the most paramount ones. Productive people have to improve schedules that include stable work on both machines and people. This is very important because overtime or night-shift operations are expensive and because, also, even during slack periods manpower must usually be kept intact.

One more important remark is the ability to quote schedules and satisfy these schedules. The cost of long lead-time, and the cost of not satisfying schedules is difficult to evaluate and even more difficult to integrate into a computation technique. We know that the entire question of costs is an open one and that more action will have to be done in this field before important result are gotten. It will be possible to improve a technique that leads to optimum production schedules only after a true measurement of efficacy of production plans has been obtained.

We know that the conventional economic-lot-size formula has just a limited use in production. In this research, we made a simple step in expanding the conventional economic-lot-size formula by involving some of the ability limitations of production and programming the problem by MATLAB.

References

1. A. Vazsonyi. Economic-Lot-Size in manufacturing. Opns. Res. 5, 28-44, 1957.

2. A. Vazsonyi. Operations Research in Production Control. Opns. Res. 4, 19-31, 1956.

3. C. C. Holt, F. Modiglioni, and H. A. Simon. Linear Decision Rule For Production and Employment Scheduling. Management Science 2, 1-30, 1955.

4. W. Karush and A. Vazsonyi. Employment Scheduling Over Time.