BINARY ILP FORMULATION FOR THE SIMULTANEOUS MACHINE LOADING PROBLEMS OF TARAKU MILLS LIMITED

BY

M. E. Meliga and L. T. Tuleun
Mechanical Engineering Department,
University of Agriculture,
Makurdi.

ABSTRACT

The integer linear programming (ILP) in production scheduling provides optimal solutions for the desired objective function in sequential scenario. A similar formulation for Taraku Mills Limited is presented where processors are capable of accommodating more than one job (Products) at the same time.

The algorithm used in the computation, utilizes two other algorithms, the Cambell, Dudex and Smith algorithms and the nearest neighbour method of the assignment problems. The simultaneous approach performs extremely better than the sequential approach. There is great reduction in the total flow time of products considered.

The solution to the Taraku Mills Limited formulation is presented in three parts with the objective of minimizing the total flow time. The first part schedules the jobs (products) unto the first processor. The second part finds the total flow time of the four products while the third part schedules the production times of these jobs unto a ganth chart.

The simultaneous approach performs entirely better than the sequential approach and the method like other operations research models can be applied to real life problems.

1. INTRODUCTION:

Modern society requires that business and industry supply its needs and desires at reasonable cost in adequate quantities, and with goods and services of satisfactory quality. Operations planning objective is to make the "best" effort to provide the desired goods and services at the right time and at a reasonable cost. Many management decisions involve maximizing the amount of profit whilst keeping the level of usage of labour, raw materials and equipment within certain limits.

Taraku Mills Limited, established in 1985 in Benue State of Nigeria is an integrated factory designed to process so many kinds of seeds like Soya beans, maize, ground-nuts, sun flower seeds and rape seeds. As a manufacturing company, it is expected to meet the market demand of its high quality products which obviously poses a lot of problems for the management.

Integer linear programming (ILP) is a programme in which some or all the decision variables are restricted to be integer values. The reason for constructing integer programmes is that many practical problems such as the limited number of machines in the shops require integers and machines operate in whole, hence integers. By having binary variables, the essence is modeling all or nothing, or go-no-go situations. These types of problems are very important because all or nothing situations are very common in such areas as scheduling.

The flow-shop scheduling theory has been traditionally based on the sequential machine loading assumptions. Such an approach is justified since the old fashioned production machine/processor transform work parts one-on-one basis. However, spectacular advances in micro-processor based manufacturing systems and ingenuity in adaptation of production systems have provided simultaneous (many-on-one) processing capabilities built in them.

Now, processors can accommodate more than one work part at the same time in their ork heads. This adaptation of the systems to simultaneous approach can bring about a dramatic reduction in manufacturing lead times.

Another simultaneous loading example is the heat treatment shop in a production facility, where a number of work parts share the same bath or furnace at the same time. The heat treatment shop more often provide the simultaneous processing than the machine shops.

Although, scheduling theory deals with identical processors, in most situations the simultaneous processing of jobs sharing the same equipment is not mentioned as such. In this paper, the simultaneous, loading/processing of jobs is adapted and discussed in terms of Taraku Mills Limited loading problems. The processors are adapted to have the ability to accommodate a certain number of jobs at the same time.

With some assumptions and borrowing notations, N/1/PF/s-g problem of the first part of the adaptation is considered first in the formulation. Where N/1/PF is a familiar term in scheduling theory, the expression s-g denotes simultaneous processing and g indicates the number of jobs sharing the same processor simultaneously.

Then, the second part of the adaptation is termed the N/2/PF/H-g, problem is considered. Again, the expression N/2/PF indicates a familiar scheduling scenario, that is, two processors scheduling problem with permutation flow shop. The last portion H-g indicates a hybrid of processors, that is, first processor would allow simultaneous loading and the second processor would allow the classical sequential loading.

2. INTEGER LINEAR PROGRAMMING:

Walukięwicz (1991) defines an integer linear programming problem as a programming in which some or all the decision variables are restricted to be integer valued. In general, there are three types of integer programmes namely: Pure integer, mixed integer and binary integer, Philip (1976). A pure integer programme is one in which all the variables are restricted to be integers, while a mixed integer programme restricts some of the variables to be integers while others can assume continuous (fractions) values and of course, binary integer programmes are those linear programmes in which all the variables can only assume the values of either zero or one (0 or 1).

The integer programming formulation of the problem provides a general purpose solution approach. As in other operations research models, the integer programming approach can accommodate a variety of performance measures. The big draw back is the computational demand that becomes severe for a modest problem, because of large number of variables and constraints. The optimization led Baker to keep improving the ILP formulations (1). All the work made use of the sequential loading approach. A similar formulation was attempted for the identical processor case. In all the models a binary variable Z i,j has been defined to acquire a value of 1 if the job I is assigned position j in the sequence and a value of 0 otherwise.

2.1 TERMS AND NOTATIONS:

- n - Number of jobs to be processed.
- m - Processor/machine number, particularly, m=1 and 2
- i - Job/work part number.
- g - Maximum number of jobs that can be processed simultaneously.
- G - Number of job-groups to be processed simultaneously, it is the smallest integer greater than or equal to n/g.
- k - Job loading sequence position on simultaneous processor 1.
- I - Job index used in combination with K.
- i - Job unloading position on processor 1 job unloading/loading sequence position for processor.
- $t_{i,m}$ - Process time on mth machine for ith job.
- $F_{i,m}$ - Flow (completion) time of ith job on machine/processor in particular, Fi, 1 = Fi.
- Wi/1- ith job waiting time after the completion of work on 1 and the start of work on machine/processor 2.
- Ii,2- Idle time of machine 2 before it starts processing ith job.

3. TARAKU MILLS LIMITED.

Taraku Mills Limited is supposedly one of the leading Soya beans, maize and ground-nut processing companies in Nigeria. The factory remains one of the most integrated and is designed to process other kinds of seeds like sunflower, beniseed and rape seed. The company was established in 1987 and is located at kilometer seventy (km 70) along Makurdi - Otukpo road. The main finished products of the company include golden soya (Vegetable oil), Benolina (Semolina), corn oil, groundnut oil and benofeeds (livestock feeds).

3.1 PROCESS FLOW IN TARAKU MILLS LIMITED:

The entire process flow in Taraku Mills factory is as shown in figure 1. The process includes groundnut, maize, soya in take, cleaning, cracking, conditioning, flaking and solvent extraction.

3.2 THE PROCESS MODEL:

The Model under consideration is as shown in Figure 1. The pre-extraction process including the extraction process up to the distillation unit is considered to be a processor, processing a particular job (Product) and which have a definite process time. The process is almost the same for each of the four products under consideration. However,. due to the nature of each of the grains and the slight difference in their routing processes, coupled with the higher oil content in others, the process times are not the same. Figure 2, shows the process of production for each of the oils and the soya meal. From the figure, it can be seen that, even though the products have almost the same process of production, some will take longer time than others.

Based on the following assumptions, the factory process model is reduced to the model shown in Figure 3:

(i) The process of production from grain intake to flaking (distiller) including the extraction process up to the distillation unit it considered a processor, processing each of the jobs (products) and which have a definite process time though different for all the products (jobs);

(ii) The process after the distillation process through, the processing of the products in the refinery is considered as a second processor which also has a definite process time; and

(iii)Each of the products under considerations is considered a particular job being processed independently.

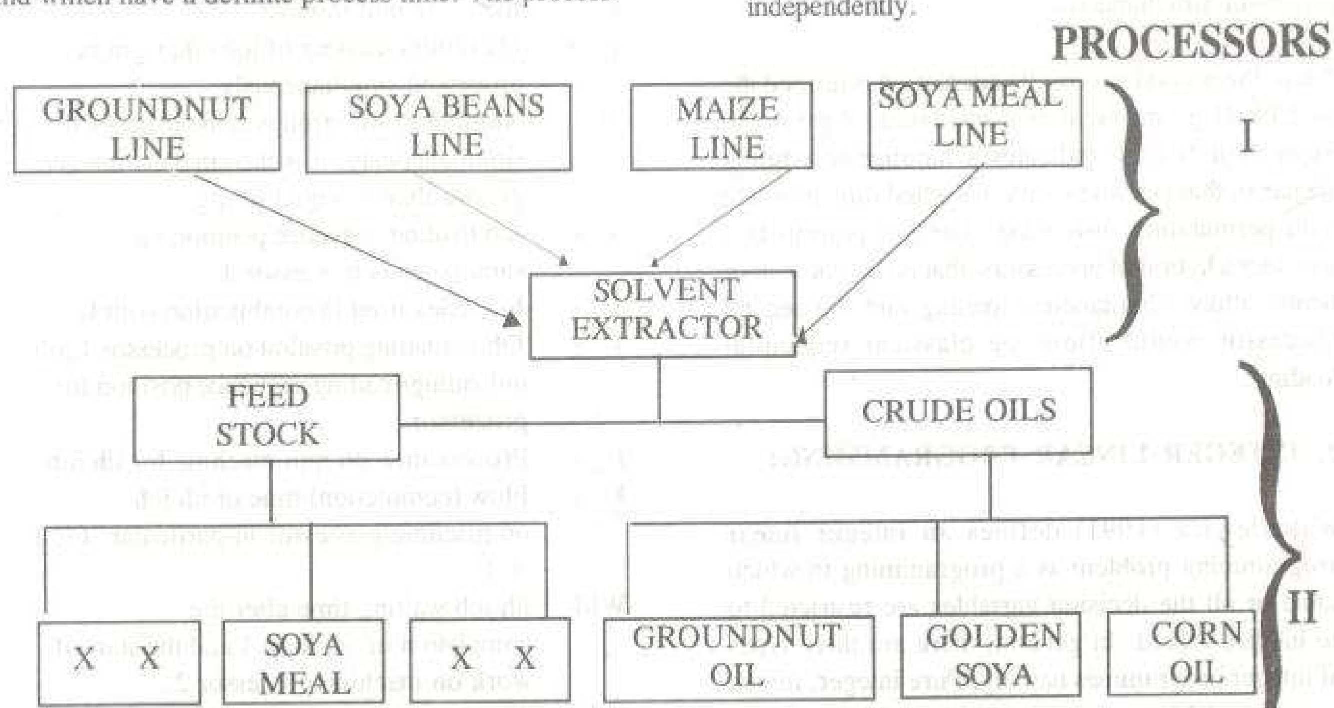

Figure 3: The model showing process of the products under consideration.

Based on the model, four jobs are to be processed on two processors. The first processor is capable of loading jobs simultaneously while the second processor loads jobs sequentially. The products to be processed are as shown in Table 1.

Table 1:Serial number of products under consideration.

PROD.	Groundnut Oil	Corn Oil	Soya Oil	Soya Meal
S/N	1	2	3	4

4. INTEGER LINEAR PROGRAMMING FORMULATION

4.1 THE FORMULATION:

Consider the products shown in Table 1 as jobs to be processed at different periods, which are termed positions.

Z_{1j} = processing of groundnut oil in positions j

Z_{2j} = processing of corn oil in position j

Z_{3j} = processing of soya oil in prosition j

Z_{4j} = processing of soya meal in position j

Where j = 1, ..., 4.

The problem therefore, is to formulate the programme in order to find the schedule, which minimizes the total flow time (completion time) for all the four jobs. Let ti, denote the process time of each of the products I on each of the processors; That is:

Zij = {1 if job I is selected in position i, 0 otherwise}

Invariably, $t_i z_{ij}$ denotes the time it takes to process job i in position j.

4.1.1 Objective Function:

The static flow-shop problem with single processor allowing simultaneous (many on one) loading offers an opportunity to device a multitude of alternative objective function. With data available on jobs process duration only, minimizing the mean flow time will also minimize the make span with additional information about the delivery date or due date, the objective of minimizing average lateness can be achieved. However, in this paper, the emphasis is on scheduling to minimize the flow time of the jobs. which is also the time when the last job is available to exit the production system.

It may therefore be formulated as the total of the flow time of the individual jobs in sequence. This can be written as:

Minimize $(F_1 + F_2 + F_3 +, ..., + F_n)$

Depending on the number of jobs to be processed, Where F_i = flow time of job i, i-1, ..., n and n = number of jobs.

This could be summarized as:

$$\text{Min} \sum_{i=1}^{i=n} F_{i,m} \qquad (1)$$

Where, m = 1 for single processor and m = 1 or 2 depending on the flow time computations on the corresponding processor.

4.2 SIMULTANEOUS PROCESSING.

4.2.1 Single Processor:

The flow time F_i of any job I on the processor is also the completion time C_i and, it is also the time at which the job is removed from the processor.

Using this relation between j,k, and I, the flow time of a job to be loaded at sequence position k as 1th member is as follows:

$$F_{(k-1)g} + \sum_{i=n}^{i=n} \{t_i z_i (k-1)g^{+1}\} = F_{(k-1)}g+1 \qquad (2)$$

$$F_{(k-1)g} + 1 >= F_{(k-1)g} + I-1$$

When k=1,2,..., 6-i

And I=1,2,...., g $\qquad (3)$

For the Gn (last) sequence position, the flow times for the jobs would be

$$F_{(G-1)g} + \sum_{i=1}^{i=n} \{t_i z_i (G-1)g^{+1}\} = F_{(G-1)}g+1 \qquad (4)$$

$$F_{(G-1)g} + > F_{(G-1)}g+1\text{-I; when I}=1, ...n\text{-Ng} \qquad (5)$$

Interpreting i as the index for unloading position, binary variable Zi,j=1 implies that job i is to be loaded at the jth position provided it was loaded at position k as Ith member on the processor.

The constraints may be rewritten in terms of j as follows:

$$F_{i-}1 + \sum_{i=n}^{i=n} \{t_i z_i\} = F_i \quad - \quad (6)$$

$$F_i > F_{i-}1 \quad - \quad (7)$$

Where: $j=1,2, \ldots, n$ and $I= 1, 2, \ldots, g$

Additional, each job i must be allocated to sequence position, j that is,

$$\sum_{j=n}^{j=n} Zij = 1_i, \quad i = 1, 2, \ldots, n \quad - \quad (8)$$

Similarly, each sequence j must be allocated to job I

$$\sum_{i=n}^{i=n} Zij = 1_i, \quad j = 1, 2, \ldots, n \quad - \quad (9)$$

4.2.2 Two Processors:

The job sequencing on two processors when the first processor allows g-jobs to be loaded simultaneously and, the second processor takes jobs sequentially is a hybrid problem.

4.3 THE CONSTRAINTS:

In the case of the simultaneous processing: Two processors, the constraint function will increase. It can be noticed that so many other factors will affect the process time of the second processor for each of the jobs and so these factors will affect the total flow time.

The constraint of equations 8 and 9 will remain. However, there will be three other sets of constraints as given below:

(i) The first set of constraints would be constraints regarding the flow times on the simultaneous processors I. These constraints can still be expressed by equations 6 and 7 with a little modification in the flow time expression. That is, use Fi,1 instead of fi for the first processor. That is:

$$F_{i-}1 + \sum_{i=1}^{n} \{t_i z_{i,} j = F_{1j}$$

$$F_{(k-1)g,} 1 + \sum_{i=1}^{n} \{t_i z_{i,} (k-j) + F\} = F_{(k-1)} gI,1 \quad - \quad (10)$$

$$F_{(k-1)g,} > F_{(k-1)} g + I\text{-}1.\ 1, \text{ when } k\text{-}1, 2, \ldots, 6\text{-}1$$

$$\text{And } I = 1, 2, \ldots g \quad - \quad (11)$$

(ii) The second constraint set would establish the principal consistency constraints. These constraints set provides equation balancing the processing time with waiting times and machine idle time for the jobs occupying adjacent positions in a sequence. For two processors, the jobs would wait for their turn after the completion of processing on processor I. Hence, all Wj,2 = 0. Similarly processor I would not remain idle at any moment, so all Ij, 1 = O. These can simply be written in the form

$$\sum_{i=1}^{n} \{t_i z_{i,} j+1) + Wj,+1,1 = Wj,1 + \sum_{i=n}^{n} \{(tii,+ti2) Z_{ij}\} + I_j +_{,j} 2\text{-} \quad (12)$$

for $j=1,2, \ldots, n\text{-}1$ and $j <> Ixg$.

(iii) The third set of constraints include constraints regarding the jobs flow time on the second (sequential) processor. It has the following relationships:

$$\sum_{i=1}^{n} \{t_i 1 + TI,2) Wi,j = Z1,2 \text{-} \quad (13)$$

$$Fj\text{-}i,_i + Ij,2 + \sum_{i=1}^{n} \{t_i 2 \ \ Zi,j\} = Fj,2 \quad - \quad (14)$$

Where $j = 2,3,\ldots, n$

5. PROCESS TIME MEASUREMENT

The process time for Taraku Mills Limited have over the times been perfected and under normal circumstances/production conditions without stoppages are as given for 90 tons of each of the products (jobs) under consideration in Table 2.

It is a flow shop production, which under normal circumstances should have the same process times. However, this is not so due to the slight differences in their routing processes and the higher oil content in others.

Table 2:Process Times of various jobs under consideration

(Table 2: 4/1/PF/problem)

Job I	Products	Process Time ti (Hrs.)
1.	Groundnut Oil	8
2.	Corn oil	9
3.	Soya oil	6
4.	Soya meal	4

6. SUBSTITUTION AND EXPANSION OF FORMULATION

6.1 SIMULTANEOUS PROCESSING:

6.1.1 Single Processor:

The integer linear programming formulation of the 4/1/PF problem, shown in Table 2 is as follows:
Objective function:

$$\text{Minimize } \sum_{i=1}^{n} F_{i,x} = \text{Min } (F_1+F_2+F_3+F_4) \quad - \quad (15)$$

Subject to:

$$F_i - 1 + \sum_{i=n}^{n} \{^{m}i,j\} = F_j \quad - \quad (16)$$

Expanding equations 15 and 16 and substitution of values shown in Table 2 yield.

$$8Z_{11}+9Z_{21}+6Z_{31}+4Z_{41} = F1 \quad - \quad (17)$$
$$8Z_{12}+9Z_{22}+6Z_{32}+4Z_{42} = F2 \quad - \quad (18)$$
$$8Z_{13}+9Z_{23}+6Z_{33}+4Z_{43} = F3 \quad - \quad (19)$$
$$8Z_{14}+9Z_{24}+6Z_{34}+4Z_{44} = F4 \quad - \quad (20)$$

$$\sum_{i=n}^{n} Z_{ij} = 1, \quad i=1,2,3,4 \quad - \quad (21)$$

$$\sum_{i=n}^{n} Z_{ij} = 1, \quad i=1,2,3,4 \quad - \quad (22)$$

6.1.2 Two Processors:

The time measurement for the two processors is the same as that of single processor and the processor time for the second processor is as shown in Table 3.

Table 3:PROCESS TIME OF THE FOUR JOBS ON BOTH PROCESSORS 1 AND 2 (TABLE 2, 4/2/PF/H-Z)

Job I	Process Time (Hours) Processor 1 (ti,1)	Processor 2 (ti,2)
1.	8	7
2.	9	5
3.	6	11
4.	4	6

The ILP formulation of the 4/Z/PF/H-2 problem shown in Table 3 is as follows:

6.2 OBJECTIVE FUNCTION

$$\text{Minimize } \sum_{i=1}^{n} F_{i,x} = \text{Min } (F_{1,2}+F_{2,2}+F_{3,2}+F_{4,2})$$

Where $F_{1,2}$ = total flow of the first job on processor 2; $F_{2,2}$ = total flow time of the second job on processor 2;$F_{3,2}$ and $F_{4,2}$ are similarly total flow time of third and forth job on processors 2.

6.3 CONSTRAINTS

Three new sets of constraints will be included to the nominal two constraints and modeled for the Taraku Mills Limited problem as shown below:

(i) The first constraints are the flow times on the simultaneous processor I and is given below as for the process times above.

$$8Z_{11}+9Z_{21}+6Z_{31}+4Z_{41} = F1 \quad - \quad (23)$$
$$8Z_{12}+9Z_{22}+6Z_{32}+4Z_{42} = F2 \quad - \quad (24)$$
$$8Z_{13}+9Z_{23}+6Z_{33}+4Z_{43} = F3 \quad - \quad (25)$$
$$8Z_{14}+9Z_{24}+6Z_{34}+4Z_{44} = F4 \quad - \quad (26)$$

(ii) The second set of constraints are the principal consistency constraints. These constraints simply enforces the technological sequence, it shows the relationship of the various jobs as they are being processed through the two processors.

(a) While job 2 is processing on processor 1, job 1 has moved to machine 2. Using the process times in Table 2 yield the following:

$$8Z_{12}+9Z_{22}+6Z_{32}+4Z_{42}+W_{2,1} = 15Z_{11}+8Z_{11}+14Z_{21}+17Z_{31}+10Z_{41}+I_{22} \quad - \quad (26)$$

(b) While job 3 is processing on machine 1, job 2 is also processing on machine/processor 2. Using the process times in Table 2 gives:

$$8Z_{13}+9Z_{23}+6Z_{33}+4Z_{43}+W_{2,1} = 7Z_{12}+5Z_{22}+11Z_{32}+6Z_{42}+I_{32} \quad - (27)$$

$$8Z_{14}+9Z_{24}+6Z_{34}+4Z_{44}+W_{3,1} = +15Z_{13}+14Z_{23}+17Z_{33}+10Z_{43}+I_{42} \quad - (28)$$

(iii) The third set of constraints is the flow times on the sequential processor 2 which is a direct extension of the other processor except that they have different process times.

Flow time on processor 2 is as given below:

$$15Z_{11}+14Z_{21}+17Z_{31}+10Z_{41} = F_{12} \quad - (29)$$

$$F_{12}+I_{22}+7Z_{12}+5Z_{22}+11Z_{23}+6Z_{42} = F_{22} - (30)$$

$$F_{22}+I_{32}+7Z_{13}+5Z_{23}+11Z_{33}+6Z_{43} = F_{32} - (31)$$

$$F_{32}+I_{42}+7Z_{14}+5Z_{24}+11Z_{34}+6Z_{44} = F_{42} - (32)$$

$$\sum_{i=n}^{n} Z_{i,j}=1, \qquad j=1,\ldots\ldots,n \quad - \quad (33)$$

$$\sum_{j=n}^{n} Z_{i,j}=1, \qquad i=1,2,\ldots\ldots, n \quad - \quad - \quad (34)$$

$F_{i,m} 7,0,\ Z_{ij}=$ binary (0-1); $1<i>4j\ 1<j<4$

6.4 TARAKU MILLS LIMITED FORMULATION, RESULTS AND DISCUSSIONS:

6.4.1 Formulation:

The Taraku Mills Limited job sequencing problem in which four jobs are to be processed on two processors in series with the first processor loading jobs simultaneously and the second hybrid processor loading jobs sequentially has its formulation results as follows:

Objective Function: The objective of Taraku Mills in formulating a feasible schedule is to minimize the total flow of the four products (jobs).

That is,

$$\text{Min} \sum_{i=n}^{n} Z_{i,j}=1, \qquad i=1,2,\ldots\ldots, n \quad -- \quad (35)$$

Subject to equations 23-28 for the first processor, and subject to equations 29-34 for the second processor.

The solutions are obtained with the aid of a computer.

The heuristic algorithm for the solution of the problem is composed of three parts. In part I, the sequence of all the operations is determined. Part II determines the total flow times of all the jobs in Part II of the algorithm, the operations are scheduled unto a Ganth chart based on the obtained sequence.

6.4.2 Results:

The results of the formulation of Taraku Mills Limited are presented in Table 4, which is the sequence of loading and the total flow time of the jobs. Figures 2 (a) and (b) show the schedule of the jobs on a Ganth Chart.

Table 4: Result of the scheduling problem of Taraku Mills Limited.

Part I (Phase I)	$Z_{41} = Z_{32} = Z_{13} + Z_{24} = 1$ $F_{11} = 4_j = F_{12} = 6_j = F_{31} = 14 F_{41} = 14$
Part II (Phase II)	$Z_{41} = Z_{32} = Z_{13} + Z_{24} = 1$ $F_{12} = 10_j = F_{22} = 21_j = F_{32} = 28_j F_{42} = 33$

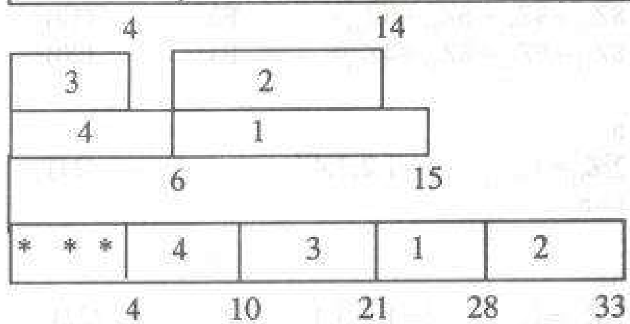

Figure: 2a: Gantt Chart showing the result of the ILP proposed system.

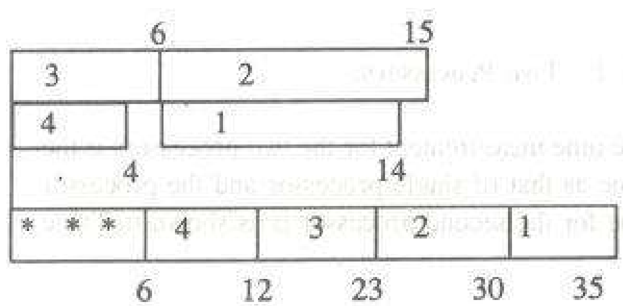

Figure :2b: Gantt Chart showing the result of the TML loading system.

6.4.3 **Discussion.**

The problem studied in this work is that of sequencing 4 (four) products in a permutation flow-shop on two (2) processors working in series, with one processor capable of processing more than one job at the same time and the second processor, processing jobs sequentially. Taraku Mills Limited intend to achieve the following (i) minimizing of lead times, (ii) maximum capacity utilization, (iii) realistic due date achievements (iv) customer satisfaction and above all, high productivity.

From the result, it could be seen that Taraku Mills Limited has higher flow times for the test cases. This is because Taraku Mills Limited (TML) adopts the first-come-first-serve (FCFS) dispatching rule, which is considered to be a fair approach to jobs scheduling. This actually reflects TML objective policy which is to maximize profit. Most orders are for soya oil and its by products and highest priority is given to soya line. However, the FCFS approach by far gives worse result when other performance measures (criteria) and objectives are considered, since it always lead to high job lateness which might be very severe. The algorithm for ILP formulation utilizes the shortest processing time (SPT) approach of Campbell Dudek and Smith method and the neighbourhood search approach of the assignment problems and the result of the TML problem is presented in table 4 and figures 2(a) and (b) above.

This result suggests that the ILP formulation yields higher variance of time in the que for a substantive time, thereby causing severe due date problem for only a few jobs. However, since the algorithm completes the average job in a relatively short time, it produces a much smaller average job lateness. Hence the overall lateness performance will be greatly reduced. These algorithm will also be better for the company (TML) especially under those conditions where due dates are set a bit less "rationally", that is where due dates are set without regard to the work content involved in processing the jobs.

Now, since the model is a "reasonable" mathematical representation of the real problem, the plans based on the production schedule will not lead Taraku Mills Limited too far astray and in fact will allow TML to plan its operations in a more efficient and profitable manner. If the ILP model and its solution represents the untarnished ideal right procedure (the difference between the optimal solution and what is actually obtained on the shop floor). It then gives production supervisors (and their managers) some goal to be measured against and indications where improvement can be made.

With this result, the primary objectives of goals of due dates, flow time and work centers utilization that typically apply to scheduling in the flow shops can be achieved. Effective due date means completing and making the jobs at hand ready as at when due. At minimized flow times, enough production is possible at minimum capacity utilization, since within a short time more quantity of the product is produced. The result of this is that, the company can now set realistic due dates for all the products which could be achieved. Also, if extra jobs are released to the shop, they will tend to have longer times, but capacity can be better utilized and perhaps due date performance can be improved. At minimized flow times, due date might simply be specified on a yes-or-no basis as more typically, due date performance is based on lateness, which is minimized by the total flow time reduction based on high capacity utilization.

Work center utilization means that, the capacity of expensive equipment and personnel are fully utilized. With effective minimization of flow time, the utilization of capacity have been achieved. It will be discovered that the reduced total flow time, the work center and personnel will be under utilized for the production of the same quantity of the product. But if the work content of the shop is increased, with the same capacity, more product will be produced, which results in productivity improvement and therefore increase capacity utilization. This is achieved easily where the orders are dynamic in nature as in Taraku Mills Limited.

Having successfully utilized the work center and expensive personnel, the result is improved productivity and this inversely results in effective cost minimization/ utilization since cost also includes the running of expensive equipment and personnel. Also, with improved productivity all the orders will be met at the right time, since due date setting are more realistic. At high productivity, both the supply from vendors and customers demand will be at the right state. The company will therefore be operating at the right economic order quantity (EOQ) which leads to low cost of inventory and the result is high cost utilization/ minimization.

With the improved productivity resulting from total flow time minimization, due dates are realistic and as a result customers are satisfied and subsequently higher demand. Financial buoyancy and self-sufficiency is achieved.

7.0 CONCLUSION

The general zero-one (0-1) ILP formulation for Taraku Mills Limited scheduling problem is shown in section 6.3.1. This formulation which is a special type of the integer linear programme (ILP) provides a general purpose solution approach. It could be seen that, as in other operations research models, the integer linear programme approach can accommodate a variety of performance measures. Specifically, an intuitive definition for the binary variable Zij was given in the context of simultaneous processing as in Taraku Mills Limited. The definition for j and subsequently Zij as regards simultaneous processes are clearly of high importance to the scheduling formulation in the flow shop scheduling criteria.

With additional information, the objective of due date setting or rather, the minimization of late due date can be achieved. The process times gotten from the factory were used to measure the performance of the formulation even though, the algorithm for solution is based on the deterministic times utilizing heuristics scheduling procedures, the results is still important. The algorithm for the ILP formulation yields a higher variance of time in the system and allows other jobs with long processing times to wait in queue for a substantial time, thereby causing severe due date problem for only a few jobs. However, since the algorithm completes the average job in a relatively short time, it produces a much smaller average job lateness, hence the overall lateness performance will be greatly reduced.

Finally, although the computational requirements are of exponential time complexity, the formulation provides a deep insight into the problem variables and the allied inter-relationship among them. Such an understanding of the problem helps in paving way for evolving more smart and efficient heuristics, since the 0-1 ILP is commonly used in practice but the topic has not received the desired attention in scheduling research. It is hoped that this research may stimulate more future research in this direction.

REFERENCES

Baker R. K. and S. Wabster, 1995 "Scheduling groups of jobs on a single machine". Journal of Computer and Operations Research Vol. 22.

Hussain I. R. Jared and A. Khan, 1993, "Integer Linear programming formulation for the simultaneous machine loading problems:. Journal of Engineering and Applied Sciences. Vol. 10

Joseph A. 1993, "Parametric formulation of the general integer linear programme problem; Computer and Operations Research. Vol. 22.

Meliga, M. E. 1997, "Binary Integer Linear Programme Formulation for the Simultaneous Machine loading Problems of Taraku Mills Limited", Unpublished, M. Eng. Thesis, University of Agriculture Makurdi, Nigeria.

Wagner, H. M. 1959, An Integer Programming model for machine scheduling; Naval Base Logistics Operations No. 6.