Chapter One

INTRODUCTION

1.1 Background of the Study

Owing to its impact on the industrial economy, the job shop scheduler and controller are vital algorithms for modern manufacturing processes. This dissertation is concerned with the modeling of agent-based job shop scheduling and control system.

In this research work, AloAluminum Company was used as the case study. The Companyis concerned with the production of aluminum roofing sheets using the same raw material but with three main finishing types leading to three types of products.The three finishing product types are referred to as finishing type 1(Metro couple corrugated sheet), finishing type 2(Step tile corrugated sheet) and finishing type 3(Regular corrugated sheet).

In the present day, the market is highly competitive, dynamic, and customer driven. This has led to increasing rates of new product introduction (i.e., decreasing product life cycle) and dynamic variations in demand patterns across product mixes. As a result, customers have become harder to satisfy and manufacturing enterprises are facing greater pressures to be responsive and flexible in response to market changes. This is to enable them compete with business rivals with the same market focus. The competitive advantage is now largely dependent upon rapid responsiveness to the dynamic changes in product mixes and demand patterns, as well as to new opportunities in the market (i.e., market shifts). The urgent need for high responsiveness and flexibility in coping with the dynamic market changes has been demonstrated by the study carried out by Zhang and Sharifi(2001) involving a case with 12 companies and a questionnaire survey with 1000 companies. The analysis of the study indicates that, in order to achieve high responsiveness, one of the operational issues to be focused on is production planning and control, particularly process planning and production scheduling, which must be dynamically and cost-effectively integrated. Conventional control strategies for manufacturing systems were not designed to achieve such responsiveness.

In the United States alone, there are over 40,000 factories producing metal-fabricated parts (Albert and Luis, 2009). These parts end up in a wide variety of products sold in the US and elsewhere. These factories employ roughly over 3 million people and ship close to $ 7 billion worth of products every year. The vast majority of these factories are called "job shops", meaning that the flow of raw and unfinished goods through them is completely random.Over the years, the behavior and performance of these job shops have been the focus of considerable attention in Operations Research (OR) literature.

Manufacturing industries are facing a growing and rapid change. Major trends like globalization, customer orientation and increasing market dynamics lead to a shift in both managerial and manufacturing principles: enterprises have to become more flexible, open, fast, effective, self-organized, decentralized, to sum it up: agile (Eric, 2002). Manufacturing serves as a basic function for any agile enterprise. The call for agility challenges the shop floor with several problems, such as dominating customer demand, management of manufacturing processes and coordination of machines and materials.(Eric, 2002).

An important issue in a manufacturing environment is the improvement of resource utilization. A classical way of achieving improved resource utilization is by using scheduling algorithms (Philippe et al, 1995). As defined by Baker(1974), scheduling is concerned with the problem of assigning a set of jobs to resources over a period of time. Performance Criteria such as machine utilization, manufacturing lead times, inventory costs, meeting due dates, customer satisfaction, and quality of products are all dependent on how efficiently the jobs are scheduled in the system (Akeela et al, 2013). Hence, it becomes increasingly important to develop effective scheduling approaches that help in achieving the desired objectives.

The diversity of products, increased number of orders, the increased number and size of workshops and expansion of factories have made the issue of scheduling production orders more complicated, hence the traditional methods of optimization are unable to solve them (Othman et al, 2007), (Raya et al, 2008).

They typically do not scale with problems size, suffering from an exponential increase in computation time. A production scheduling and control that performs reactive scheduling and can make decision on which job to process next based solely on its partial view of the plant becomes necessary. This requirement puts the problem in the class of agent based model (ABM). Hence this work adopts an alternative view on job-shop scheduling problem where each resource is equipped with adaptive agent that, independent of other agents makes job dispatching decision based on its local view of the plant.

1.2 Statement of the Problem

This work explores the well-known n-by-m Job Scheduling Problem (**JSP**), in which n jobs must be processed exactly once on each of m machines. Each job i ($1 \leq I \leq n$) is routed through each of the m machines in a predefined order π_i where $\pi_i(j)$ denotes the jth machine ($1 \le j \le m$)in the routing order. The processing of job i on machine $\pi_i(j)$ is denoted O_{ij} and is called an operation. The scheduling objective is makespan minimization, i.e., to minimize the completion time of the last operation of any job.

Existing deterministic shop floor schedulers work well for situation where n job must pass through m machine in any order while in the case study company, the n job must pass through the m machine in a given sequence which makes the job shop scheduling more complicated.Also given the fact that agent-based modeling (ABM) is proven to be an effective way of modeling complex systems that are not easy to characterize analytically, this dissertation is focused on addressing the JSP by developing an agent-based model in which the stochastic impact on the dynamics of the schedule is formulated as a Markov chain.

1.3 **Aim and Objectives of the Study**

The aim of this study is to model an agent-based job shop scheduling and control using Markov Chain for steady state probabilities. Hence the work focuses on achieving the following objectives.

1. To develop an order agent to handle incoming orders and in due time to shift the orders to the scheduler.

2. To develop a scheduler agent to schedule the incoming jobs and handle makespan optimization
3. To develop a production agent to produce the scheduled jobs, adding slack as necessary to ensure that each finishing time lasts for an exact number of days; thus avoiding machine ideal time.
4. To develop an order release agent to:
   i. Forecast when the order is likely to be ready and inform customers.
   ii. Work out the cost per kilogram of order.
   iii. Pass the cost to customer.
   iv. Inform the customer when the order is ready to come and make payment and collect the order.
5. Simulate the developed model of objective 1 to 4 using Monte Carlo technique, to simulate customer order arrival.
6. To validate the optimized makespan using D.G. Kendall classical poisson queuing technique.

1.4 Significance of the Study

Efficient shop floor scheduling is very vital in a production system that relies heavily on the tight integration of the upstream supplier of parts, the midstream manufacturer and assembler of components, and the downstream distributor of finished goods. The successful outcome of this work should be of great prospect to raising the performance of this sort of supply chain that relies heavily on the shop floor scheduling and control mechanism of the middle manufacturer.

Globalization and strong competition in the current marketplace have forced companies to change their ways of doing business. Manufacturers have been compelled to adopt strategies such as Build-to-order (BTO) or Configuration-to-order (CTO) services. These all geared towards harnessing Just-in-Time (JIT) and Total Quality Management (TQM) strategies in order to realize greater plant productivity, improved processes and products, lower cost and higher profits. The methodical leverage of the contributions of this work would help remove the bottleneck currently inherent at the shop floor towards the effective exploitation of these production management strategies.

1.5 **Scope of the Study**

This work covers the modeling of the scheduling and control system that sequences jobs on machines used at AloCompany. The study also includes the modeling of optimization algorithm for the job shop scheduler. The objective of the optimization is makespan minimization. However, the work does not delve into issues relating to production line job routing, process planning and computerized numerical control (CNC) machine part programming.

1.6 Overview of research stages

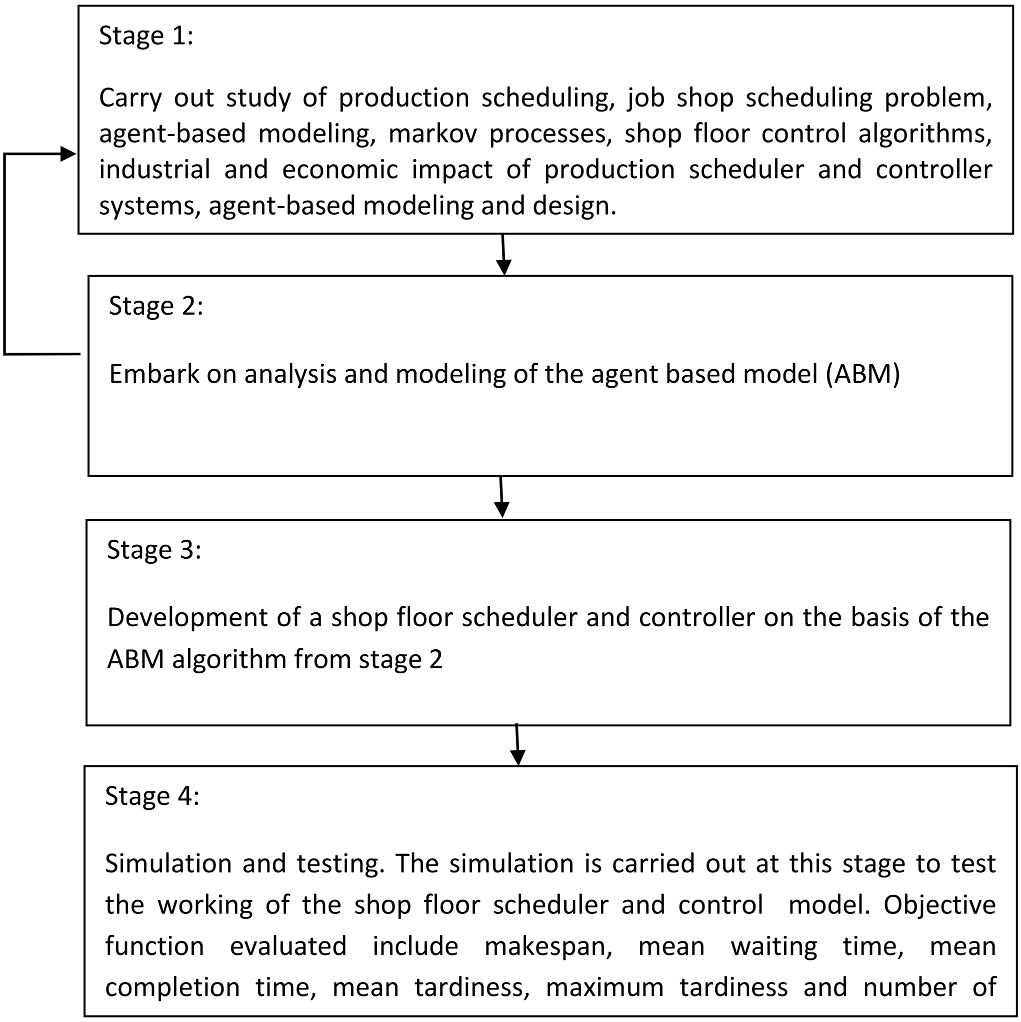


Fig.1.1:Block Diagram Overview of Research stages

The block diagram of fig.1.1, presents an overview of the project stages which started with general research on the problem areas to the final stage of simulation and testing the performance of the shop floor scheduler and control model developed.

Chapter Two

Literature Review

2.1 Job Shop Scheduling

Scheduling is an important tool for manufacturing and engineering, where it can have a major impact on the productivity of a process (Blazewic et al., 2001). In manufacturing, the purpose of scheduling is to minimize the production time and cost, by telling a production facility what to make, with which staff, and on which machine.

Survey of literature indicates that the job shop scheduling problem (or job-shop problem) is at least 70 years old. In the publications by Pinedo(2002), Tsai(2008), job shop scheduling is reported as an optimization problem in computer engineering and operations research in which ideal jobs are assigned to resources at particular times. The most basic version is described as follows (Pinedo, 2002):

Given n jobs J1, J2, …… Jn of varying sizes, which need to be scheduled on m identical machines, the task is to work out the scheme for assigning job i to machine $m_i$in order to minimize the makespan. The makespan is the total length of the schedule (that is, when all the jobs have finished processing). In the literature nowadays, the problem is presented as an online problem (dynamic scheduling), that is, each job is presented, and the online algorithm needs to make a decision about that job before the next job is presented. This problem is one of the best known online problems, and was the first problem for which competitive analysis was presented, by Graham (Graham, 1966). Best problem instances for basic model with makespan objectives are due to Taillard (Taillard, 1972).

2.1.1 **Job Shop Scheduling: Problem Variations**

Pinedo(2002) and Othman et al.(2007) reported existence of variations of the job scheduling problem, which include the following:

- Machines can be related, independent, equal
- Machines can require a certain gap between jobs or no idle-time
- Machines can have sequence-dependent setups
- Objective function can be to minimize the makespan, the linear programming (Lp) norm, tardiness, maximum lateness etc. It can also be multi-objective optimization problem
- Jobs may have constraints, for example, job ineed to finish before job j can be started. Also, the objective function can be multi-criteria (Malakoot, 2013).
- Jobs and machines have mutual constraints, for example, certain jobs can be scheduled on some machines only.

- Set of jobs can relate to different set of machines.
- Jobs may have deterministic (fixed) processing times or probabilistic processing times
- There may also be some other side constraints.

Just as it is known that the travelling salesman problem (TSP)is non deterministic polynomial-time (NP)hard, then the job-shop problem is clearly also NP-hard, since the TSP with m = 1 (the salesman is the machine and the cities are the jobs) (Pinedo, 2002).

2.1.2 **Job Shop Scheduling: Problem Representation**

Available literature indicate the disjunctive graph (Roy and Sussmann, 1964) as one of the popular models used for describing the job shop scheduling problem (JSP) instances (Jacek et al, 2000).

A mathematical statement of the problem can be as follows:

Let m $=\{M_1, M_2, \ldots\ldots M_m\}$ and $J= \{J_1, J_2, \ldots\ldots J_n\}$ be two finite sets. On account of the industrial origins of the problem, the Mi are called machines and the J_j are called jobs.

Let x denote the set of all sequential assignments of jobs to machines, such that every job is done by every machine exactly once; element$x \in \chi$ may be written as n x m matrices, in which column i lists the jobs that machine M_i will do, in order. For example, the matrix

$$x = \begin{pmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 1 \end{pmatrix}$$

Means that machine M_1 will do the three jobs J_1, J_2, J_3, in the order J_1, J_2, J_3, while machine M_2 will do the jobs in the order J_2, J_3, J_1. Suppose also that there is some cost function $C: \chi \rightarrow [0, +\infty]$.The cost function may be interpreted as a "total processing time", and may have some expression in terms of time.

Cij: M x J $\rightarrow [0, +\infty]$, the cost /time for machine M_i to do job J_j.

The job-shop problem is to find an assignment of jobs $x \in \chi$ such that $C(x)$ is a minimum, that is, there is no $y \in \chi$ such that $C(x) > C(y)$.

The reference Pinedo(2002) noted that one of the first problems that must be dealt with in the JSP is that many proposed solutions have infinite cost: i.e., there exists $x_\infty \in \chi$ such that $C(x_\infty) = +\infty$. Infact, it is quite simple to concoct examples of such x_∞ by ensuring that two machines will deadlock, so that each waits for the output of the other's next step.

Graham had already provided the list scheduling algorithm, which is $(2-{}^{1}/_{m})$ – competitive, where m is the number of machines (Graham, 1966). Also, it was proved that list scheduling is optimum online algorithm for 2 and 3 machines. The Coffman – Graham (1972) algorithm for uniform – length jobs is also optimum for two machines, and is $(2-{}^{2}/_{m})$ – competitive.Bartal et al. (1992) presented an algorithm that is 1.986 competitive. Kanger et al.(1994) reported that a 1.945 – competitive algorithm was presented by Kanger, Philips and Torry. Albers et al. (1992) provided a different algorithm that is 1.923 – competitive. The best known result is an algorithm given by Fleischer and Rudolf, which achieves a competitive ratio of 1.9201(Fleischer and Rudolf, 2009).Competitive ratio is an asymptotic approximation ratio applied to online algorithm. The performance of algorithms is measured by the competitive ratio. Applying machinelearning to job scheduling is an emerging approach now (Rosemarin et al 2017). In this approach, artificialintelligence determines optimizations without the need for human programmers to create an algorithm for them or to fully understand the complex causation that drives them (Goodhill, 2017).

2.1.3 **Offline Makespan Minimization**

The simplest form of the offline makespan minimization problem deals with atomic jobs; which is jobs that are not subdivided into multiple operations. It is equivalent to packing a number of items of various different sizes into a fixed number of bins, such that the maximum bin size needed is as small as possible.

Hochbaum and Shmoys (1987) presented a polynomial-time approximation scheme that finds an approximate solution to the offline makespan minimization problem with atomic jobs to any desired degree of accuracy.

2.1.4 **Job Consisting of Multiple Operations**

The basic form of the problem of scheduling jobs with multiple (m) operations over m machines, such that all of the first operations must be done on the first machine, all of the second operations on the second machine, etc., and a single job cannot be performed in parallel, is known as the open shop scheduling problem. Various algorithms are reported (Khuri and Miryala, 2001) in the literature.

A heuristic algorithm by Johnson (2003) can be used to solve the case of a 2 machine N job problem when all jobs are to be processed in the same order. The steps of the algorithm are as follows:

- Job P_i has two operations, of duration P_{i1}, P_{i2}, to be done on machines M_1, M_2 in that sequence.
- Step 1. List A = {1,2, ….., N}, List L_1 = { }, List L_2 = { }

    Where List A is the list of jobs 1…..N.

    List L_1 and $L_2$is the time for the two operations.

List the jobs and their times at each work center.

- Step 2, from all available operation durations, pick the minimum job.

  If the minimum belongs to P_{k1},

remove K from list A; Add K to end of list L_1

if minimum belongs to P_{k2},

remove K from list A; Add K to beginning of list L_2.

P_{k1} is the first machine center

P_{k2} is the second machine center

Select the job with the shortest activity time. If that activity time is for the first work center, then schedule the job first. If that activity time is for the second work center, then schedule the job last.

Step 3. Repeat step 2 until list A is empty

- Step 4. Join list L_1, list L_2. This is the optimum sequence.

Johnson's method only works optimally for two machines. However, since it is optimal, and easy to compute, some researchers have tried to adopt it for M machines, (m>2).

The idea is as follows: imagine that each job requires m operations in sequence, on m_1, m_2, ….Mm,the first $^m/_2$ machines are combined into an (imaginary) machining center, MC_1, and the remaining machines into a machining center MC_2. Then the total processing time for a job P on MC_1 = Sum (operation times on first $^m/_2$ machines), and processing time for job P on MC_2 = sum (operation times on last $^m/_2$ machines). By doing so, the m – machine problem is said to be reduced to a two machining center scheduling problem. This can then be solved by using the Johnson's method.

2.1.5 **Kendall's Notation**

In queueing theory, a discipline within the mathematical theory of probability, Kendall's notation is the standard system used to describe and classify a queueing node. D.G. Kendall proposed, describing queueing models using three factors written A/S/c in 1953 (Kendall, 1953) where A denotes the time between arrivals to the queue, S the size of jobs and c the number of servers at the node. It has since been extended to A/S/c/K/N/D where K is the capacity of the queue, N is the size of the population of jobs to be served, D is the queueing discipline (Lee, 1966) (Taha, 1968).

When the final three parameters are not specified (i.e. M/M/1 queue), it is assumed K=∞, N=∞ and D=FIFO (Gautam, 2007).

2.2 **Markov Chain**

A Markov Chain (Norris, 1998) named after Andrew Markov, is a mathematical system that undergoes transition from one state to another on a state space. A Markov Chain is a stochastic process with the Markov property. The term "Markov Chain" refers to the sequence of random variables such a process moves through, with the Markov property defining serial dependence only between adjacent periods (as in a "Chain"). It can thus be used for describing systems that follow Chain-of linked events, where what happens next depends only on the current state of the system. In the literature, different Markov processes are designated as "Markov Chains". Usually however, the term is reserved for a process with a discrete set of times (i.e. a discrete-time Markov Chain (DTMC) (Everitt, 2002), although some authors use the same terminology to refer to a continuous – time Markov Chain without explicit mention (Parzen, 1962) and (Dodge, 2003). While the time parameter definition is mostly agreed upon to mean discrete-time, the Markov Chain state space does not have an established definition: the term may refer to a process on an arbitrary general state space (Meyn and Tweedie, 2011). However, many uses of Markov Chain employ finite or countable (discrete on the real line) state space, which has more straightforward statistical analysis.

The changes of state of the system are called transitions, and the probabilities associated with various state changes are called transition probabilities. The process is characterized by a state space, a transition

Matrix describing the probabilities of particular transitions, and an initial state (or initial distribution) across the state space. By convention, it is assumed all possible states and transitions have been included in the definition of the process. So there is always a next state and the process does not terminate.

A discrete-time random process involves a system which is in a certain state at each step, with the state changing randomly between steps. The steps are often thought of as moments in time, but they can equally refer to physical distance or any other discrete measurement. Formally, the steps are the integers or natural numbers, and the random process is a mapping of these to states. The Markov property states that the conditional probability distribution for the system at the next step (and infact all future steps) depends only on the current state of the system, and not additionally on the state of the system at previous steps. Since the system changes randomly, it is generally impossible to predict with certainty the state of a Markov Chain at a given point in the future. However, the statistical properties of the system's future can be predicted. In many applications, it is these statistical properties that are important.

2.2.1 **Formal Definition**

A Markov Chain is a sequence of random variable X_1, X_2, X_3,…… with the Markov property, namely that, given the present state, the future and past states are independent. Formally,

P_r $(X_{n+1} = x/X_1, = x_1, X_2 = x_2 \ldots\ldots, X_n = x_n) = P_r (X_{n+1} = x/X_n = x_n) = P_{ij}$

Where P_{ij} is one-step transition probability ie. The probability that the chain, whenever in state i, moves next (one unit of time later) into state j, P_r is the probability, $X_{n+1}$is the next state, $X_1$is the present state, $\{X_0, \ldots, X_{n-1}\}$ is the past state and n is the present time. if both conditional probabilities are defined, i.e. if P_r $(X_1 = x_1, \ldots\ldots, X_n = x_n) > 0$ the possible values of X_i form a countable set S called the state space of the Chain.

Markov Chains are often described by a sequence of directed graphs, where the edges of graph n are labeled by the probabilities of going from one state at time n to other states at time n+1, P_r $(X_{n+1} = x/X_n = x_n)$. The same information is represented by the transition matrix from time n to time n+1. However, Markov Chains are frequently assumed to be time-homogenous, in which case the graph and matrix are independent of n and so are not presented as sequences.

These descriptions highlight the structure of the Markov Chain that is independent of the initial distribution P_r (X_1 = x_1). When time-homogenous, the Chain can be interpreted as a state machine assigning a probability of hopping from each vertex or state to an adjacent one. The probability P_r (X_n = x/X_1 = x_1) of the machines state can be analyzed as the statistical behavior of the machine with an element x_1 of the state space as input, or as the behavior of the machine with the initial distribution P_r (X_1 = y) = [x_1 = y] of states as input, where [P] is the Iverson bracket. The stipulation that not all sequences of states must have nonzero probability of occurring allows the graph to have multiple connected components. Suppressing edges encoding a zero (O) transition probability, as if a has a nonzero probability of going to b but a and b lie in different connected components, then P_r (X_{n+1} = b/X_n = a) is defined, while P_r (X_{n+1} = b/X_1 = x, …..x_n = a) is not (Meyn and Tweedie, 2011)

Variations

- Continuous –time Markov processes have a continuous index
- Time-homogenous Markov Chains (or stationary Markov Chains) are processes where P_r (X_{n+1} = x/X_n = y) = P_r (X_n = x/X_{n-1} = y) for all n. The probability of the transition is independent of n.

- A Markov Chain of order m (or a Markov Chain with memory m), where m is finite, is a process satisfying

  $P_r (X_n = x_n/X_{n-1} = x_{n-1}, X_{n+2} = x_{n-2}, \ldots\ldots X_1 = x_1)$

  $= P_r (X_n = x_n/X_{n-1} = x_{n-1}, X_{n-2} = x_{n-2}, \ldots\ldots X_{n-m} = x_{n-m})$ for $n > m$ 2.1

In other words, the future state depends on the past m states. It is possible to construct a Chain (Y_n) from (X_n) which has the 'classical' Markov property by taking as state space the ordered m – tuples of x values, i.e. $Y_n = (X_n, X_{n-1}, \ldots\ldots, X_{n-m+1})$.

Transient Evolution

The probability of going from state i to state j in n time steps is

$P_{ij}^{(n)} = P_r (X_n = j /X_0 = i)$ 2.2

and the single – step transition is

$P_{ij} = P_r (X_1 = j /X_0 = i)$2.3

For a time-homogenous Markov Chain:

$P_{ij}^{(n)} = P_r (X_{k+n} = j /X_k = i)$ 2.4

and

$P_{ij} = P_r (X_{k+1} = j /X_k = i)$2.5

The n-step transition probabilities satisfy the Chapman –Kolmogrov equation, that for any K such that 0<k<n,

$$P_{ij}^{n} = \sum_{r \epsilon s} P_{ir}^{(k)} P_{rj}^{(n-k)} \quad 2.6$$

where S is the state space of the Markov Chain.

The marginal distribution Pr $(X_{n = x})$ is the distribution over states at time n. The initial distribution is P_r $(X_{0=x})$. The evolution of the process through one-time step is described by

$$P_r (X_n = j) = \sum_{r \epsilon s} P_{rj} P_r (X_{n-1} = r) = \sum^{(n)} P_{rj} P_r (X_0 = r)$$
2.72.7

Note: The subscript (n) is an index and not an exponent.

Properties

A state j is said to be accessible from a state i (written i→j) if a system started in state i has a non-zero probability of transitioning into state j at some point. Formally, state j is accessible from state i

if there exists an integer $n_{ij} \geqslant 0$ such that

$P_r (X_{nij} = j / X_0 = i) = P_{ij}^{(nij)} > 0$.2.8

This integer is allowed to be different for each pair of states hence the subscripts in n_{ij}. Allowing n to be zero means that every state is defined to be accessible from itself.

A state i is said to communicate with state j (written i↔j) if both i⟶ j and j ⟶ i. A set of states C is a communicating class if every pair of states in C communicates with each other, and no state in C communicates with any state not in C. It can be shown that communication in this sense is an equivalence relation and thus that communicating classes are the equivalent classes of this relation. A communicating class is closed if the probability of leaving the class is zero, namely that if i is not in j, then j is not accessible from i.

A state i is said to be essential or final if for all j such that i⟶ j it is also true that j ⟶ i. A state i is inessential if it is not essential (Asher, 2009).

A Markov Chain is said to be irreducible if its state space is a single communicating class, in other words, if it is possible to get to any state from any state.

Periodicity

A state i has period k if any return to state i must occur in multiple of k time steps. Formally, the period of a state is defined as:

$K = \gcd\{n : P_r(X_n = i/X_0 = i) > 0\}$

(where "gcd" is the greatest common division). Note that even though a state has period k, it may not be possible to reach the state in k steps. For example, suppose it is possible to return to the state in {6,8,10,12….} time steps; k would be 2, even though 2 does not appear in this list.

If k = 1, then the state is said to be aperiodic: returns to state i can occur at irregular times, in other words, a state i is aperiodic if there exists n such that for all $n^1 \geqslant n$,

$P_r(X_n^1 = i/X_0 = i) > 0$

Otherwise (k>1), the state is said to be periodic with period k. a Markov Chain is aperiodic if every state is aperiodic. An irreducible Markov Chain only needs one aperiodic state to imply all states are aperiodic. Every state of a bi partite graph has an even period.

Recurrence

A state i is said to be transient if, given that the system start in state i, there is a non-zero probability that the system will never return to i. Formally, let the random variable Ti be the first return time to state i (the "hitting time"): $T_i = \inf\{n \geqslant 1 : X_n = i / X_0 = i\}$

the number $f_{ii}^{(n)} = P_r\ (T_i = n)$ is the probability that state i is returned to for the first time after n steps. Therefore, state i is transient if

$$P_r\ (T_i < \infty) = \sum_{n=1}^{\infty} f_{ii}^{(n)} < 1 \qquad 2.9$$

Statei is recurrent (or persistent) if it is not transient. Recurrent states are guaranteed to have a finite hitting time.

Even if the hitting time is finite with probability 1, it need not have a finite expectation. The mean recurrence time at state i is the expected return time M_i.

$$M_i = E\,[T_i] = \sum_{n=1}^{\infty} n.f_{ii}^{(n)} \qquad 2.10$$

State i is positive recurrent (or non-null persistent) if M_i is finite; otherwise, state i is null recurrent (or null persistent).

It can be shown that a state i is recurrent if and only if the expected number of visits to this state is infinite, i.e.

$$\sum_{n=0}^{\infty} P_{ii}^{(n)} = \infty \qquad 2.11$$

A state i is called absorbing if it is impossible to leave this state. Therefore, the state i is absorbing if and only if

$P_{ii} = 1$ and $P_{ij} = 0$ for $i \neq j$

2.2.2 **Markov chain Monte Carlo**

In statistics, Markov chain Monte Carlo (MCMC) methods comprise a class of algorithms for sampling from a probability distribution. By constructing a Markov chain that has the desired distribution as its equilibrium distribution, one can obtain a sample of the desired distribution, by observing the chain after a number of steps. The more steps there are, the more closely the distribution of the sample matches the actual desired distribution. Markov chain Monte Carlo methods are primarily used for calculating numerical approximations of multi-dimensional integrals, for example in Bayesian statistics, computational physics (Gupta et al., 2014). In Bayesian statistics, the recent development of Markov chain Monte Carlo methods has been a key step in making it possible to compute large hierarchical models that require integrations over hundreds or even thousands of unknown parameters (Banerjee et al., 2012). Random walk Monte Carlo methods are a kind of random simulation or Monte Carlo method. However, whereas the random samples of the integrand used in a conventional Monte Carlo integration are statistically independent, those used in Markov chain Monte Carlo methods are correlated. A Markov chain is constructed in such a way as to have the integrand as its equilibrium distribution.

Interacting Markov chain Monte Carlo methodology are a class of mean field particle methods for obtaining random samples from a sequence of probability distributions with an increasing level of sampling complexity (Del-Moral, 2013). These probabilistic models include path space state models with increasing time horizon, posterior distributions sequence of partial observations, increasing constraint level sets for conditional distributions, decreasing temperature schedules associated with some Boltzmann-Gibbs distributions, and many others. In principle, any Markov chain Monte Carlo sampler can be turned into an interacting Markov chain Monte Carlo sampler. These interacting Markov chain Monte Carlo samplers can be interpreted as a way to run in parallel a sequence of Markov chain Monte Carlo samplers.

2.3 **Agent-Based Modeling (ABM)**

ABMs are system models specified in terms of intelligent agents. Intelligent agent is briefly discussed in the next subsection. Agent-based modeling (ABM) is a modeling approach reported to have gained increasing attention over the past 18 years or so (Charles and Michael, 2011). This growth trend is evidenced by the increasing number of applications, articles appearing in modeling and applications journals, funded programs that call for agent-based models incorporating elements of human and social behavior, the growing number of conferences on or that have tracks dedicated to agent-based modeling, the demand for ABM courses and instructional programs, and the number of preparations at conferences such as the Winter simulation conference(WSC) that references agent-based modeling. Some authors Axelrod(1997) and Law (1998) contend that ABM "is a third way of doing science" and could augment traditional deductive and inductive reasoning as discovered methods.

Based on survey of the literature, it can be said that agent-basedmodeling is being applied to many areas, spanning human social, physical and biological systems. It is reported that applications range from modeling ancient civilizations that have been gone for hundreds of years, to designing new markets for products that do not exist right now. Heath et al.(2009) provides a review of agent-basedmodeling applications. Selected applications that use the Repast agent-basedmodeling toolkit are listed in table 2.1. All of the cited publications make the case for agent-based modeling as the preferred approach against other modeling techniques for the problem addressed. These cited publications (refer to table 2.1) argue that agent-basedmodeling is used because only agent-basedmodel can explicitly incorporate the complexity arising from individual behaviors and interactions that exist in the real-world.

Table 2.1: A sample of recent Agent Based applications available on the web (all applications use the Repast Agent-Based Modeling toolkit)

Application Area	
Agriculture	A spatial individual-based model prototype for assessing potential exposure of farm workers conducting small-scale agricultural production (Leyk et al., 2009)
Air Traffic Control	Agent-Based Model of air traffic control to analyze control policies and performance of an air traffic management facility (Conway, 2006)
Anthropology	Agent Based Model of prehistoric settlement patterns and political consolidation in the lake Titicaca basin of peru and Boliria (Griffin and Standish, 2007)
Biomedical	The Basic Immune Simulator, an agent-based Model

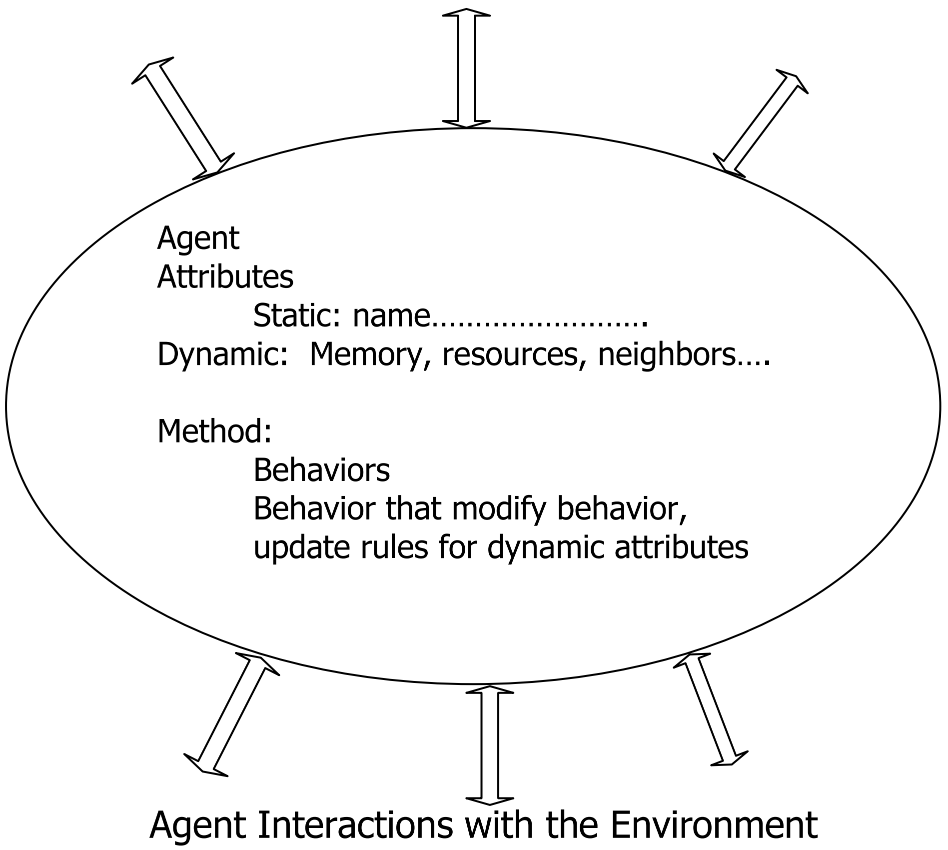
Research	to study the interactions between inmate and adaptive immunity (Folcik and Orosz 2007)
Crime Analysis	Agent-Based Model that uses a realistic virtual urban environment, populated with virtual burglar agent (Malleson, 2010)
Ecology	Agent-Based Model to investigate the trade-off between road avoidance and salt pool spatial memory in the movement behavior of more in the Laurentides wild life Reserve (Grosman et al., 2011) • Agent-based Model of predator-prey relationships between transient killer whales and other marine mammals (Mock and Testa, 2007) • A risk-based approach for analyzing the intentional introduction of non-native oysters on the USeast coast (Opalvch et al., 2005)
Energy Analysis	Agent-Based Model to identify potential intervention for the uptake of wood-pellet heating in Norway (Sopha, 2011)
Epidemiology	Synthetic age-specific contact matrices are computed through simulation of a simple individual based model (Lozzi, 2010)
Evacuation	A simulation of tsunami evacuation using a modified form of Helbing's social-force model applied to agent (Puckett, 2000)
Social Networks	An Agent-Based Model of email-based social networks, in which individuals establish, maintain and allow atrophy of links through contact lists and emails (Menges et al., 2008)

2.3.1 **Agents**

The understanding is that there is no universal agreement on the precise definition of the term agent in the context of ABM. It is the subject of much discussion and occasional debate. The issue is more than an academic one, as it often surfaces when one makes a claim that one's model is agent – based or when one is trying to discern whether such claims made by others have validity. However for want of definition, agents can be defined as autonomouse entities that act within an environment (Jennings, 2000). That is, agents are free to choose their own actions. An agent is often referred to generally as an entity (piece of software) that accomplishes some tasks on behalf of its user. There are important implications of the term agent-based when used to describe a model in terms of the model's capabilities or potential capabilities that could be attained through relatively minor modification. Some modelers consider any type of independent component, whether it be a software component or a model to be an agent (Bmabeau, 2001). Some authors insist that a component's behavior must also be adaptive in order for it to be considered an agent. Casti(1997) argues that agents should contain both base-level rules for behavior as well as a higher-level set of "rules to change the rules". The base-level rules provide response to the environment, while the rules-to-change-the-rules provide adaptation. Jennings(2000) provides a view of agent that emphasizes the essential characteristic of autonomous behavior.

For practical modeling purposes, agents are often considered to have certain properties and attributes, as follows (fig. 2.1):

Agent Interactions with Other Agents



Agent Interactions with the Environment

Fig. 2.1: A Typical Agent

A typical agent-based model has three elements:

1. Agent, their attributes and behaviors
2. Agent relationships and methods of interaction. An underlying topology of connectedness defines how and with whom agents interact.
3. Agents environment; agent live in and interact with their environment in addition to other agents.

Autonomy

An agent is autonomous and self-directed. An agent can function independentlywithin it's environment and in its interactions with other agents, generally from a limited range of situations that are of interest and that arise in the model. When we refer to an agents' behavior, we refer to a general process that links the information the agent senses from its environment and interactions to its decisions and actions.

Modularity

Agents are modular or self-contained. An agent is an identifiable, discrete individual with a set of characteristics or attributes, behaviors, and decision-making capabiliity. The modularity requirement implies that an agent has a boundary, and one can easily determine whether something (that is, an element of the model's state) is part of an agent, is not part of an agent, or is a characteristic shared among agents.

Social

An agent is social, interacting with other agents. Common agent interaction protocol include contention for space and collision avoidance, agent recognition, communication and information exchange, influence, and other domain-or application-specific mechanisms.

Conditionality

An agent has a state that varies over time. Just as a system has a state consisting of the collection of its state variables, an agent also has a state that represents its condition, the essential variables associated with its current situation. An agent's state consists of a set or subset of its attributes. The state of an agent-based model is the collective states of all the agents along with the state of the environment. An agent's behaviors are conditioned on its state. As such, the richer the set of an agent's possible states, the richer the set of behavior's that an agent can have.

2.4**Designing Agent-Based Model**

Modern software practices are based on a template design approach in which recurring elements are codified and reused for new applications; this approach has proven very valuable in designing model's as well as software. Several formats have been proposed for describing agent-based designs. Chief among these standards is Grimm et al's "Overview, Design concepts and Detail (ODD) protocol (Grimm et al., 2006). ODD describes models using a three-part approach: overview, concepts, and details. The model overview includes a statement of the model's intent, a description of the main variables, and a discussion of the agent activities and timing. The design concepts include a discussion of the foundations of the model, and the details include the initial setup configuration, input value definitions, and description of any embedded models (Grimm et al., 2006).

North and Macal(2011) discussed product design patterns for agent-based modeling. For example, design patterns that have proven themselves useful for agent-based modeling include:

- Scheduler scramble: The problem addressed is when two or more agents from the ABM pattern can schedule events that occur during the same clock tick. Getting to execute first may be an advantage or disadvantage. How do you allow multiple agents to act during the same clock tick without giving a long-term advantage to any one agent?

- Context and projection Hierarchy: The problem addressed is how to organize complex space into a single unified form such that individual agents can simultaneously exist in multiple spaces and the spaces themselves can be seamlessly removed and added.
- Strategy: The problem addressed is how to let clients invoke rules that may be defined long after the clients are implemented. There are a set of rules that need to be dynamically selected while a program is running. There is a need to separate rule creation from rule activation.
- Learning: The problem addressed is how to model agents that adapt or learn. There is need for agents to change their behavior over time based on their experiences.

2.4.1 **Markov Chain Approach for Agent-Based Modeling**

Sven et.al (2012) analyzed the dynamics of agent-based models from a Markovian perspective and derived explicit statements about the possibility of linking a microscopic agent model to the dynamical processes of macroscopic observables that are useful for a precise understanding of the model dynamics. These authors strongly argue that it is in this way the dynamics of collective variables may be studied, and a description of macro dynamics as emergent properties of micro dynamics, in particular during transient times, is possible. The work by Sven et al.(2012) is a contribution to interweaving two lines of research that have developed in almost separate ways;Markov Chains and agent-based models. The former represents the simplest form of a stochastic process while the later puts a strong emphasis on heterogeneity and social interactions.

The usefulness of the Markov Chain formalism in the analysis of more sophisticated ABM has been discussed by Izuquiredo et al.(2009), who looked at ten well-known social simulation models by representing them as a time-homogeneous Markov Chain. Among these models are the Schelling segregation model (Schelling, 1971), the Axelrod model of cultural dynamics (Axelrod, 1997) and the sugar scape model from Epstein and Axtell (Epstein and Axtell, 1996). The main idea of Izquiredo et al(2009) is to consider all possible configurations of system as the state space of the Markov Chain. Despite the fact that all the information of the dynamics on the ABM is encoded in a Markov Chain, it is difficult to learn directly from this fact, due to the huge dimension of the configuration space and its corresponding Markov transition matrix.

The work done in Izquiredo et al(2009) mainly relies on numerical computations to estimate the stochastic transition on metrices of the models.

Consider an ABM defined by a set N of agents, each one characterized by individual attributes that are taken from a finite list of possibilities. We denote the set of possible attributes by S and we call the configuration spaceΣ the set of all possible combination of attributes of the agents, i.e., $\Sigma = S^N$. This also incorporates models where agents move on a lattice (e.g., in the sugarscape model) because we can treat the sites as "agents" and use an attribute to encode whether a site is occupied or not. The updating process of the attributes of the agents at each time step typically consists of two parts. First, a random choice of a subset of agents is made according to some probability distribution w. Then the attributes of the agents are updated according to a rule, which depends on the subset of agents selected at this time. With this specification, ABM can be represented by a so-called random map representation which may be taken as an equivalent definition of a Markov Chain (Levin et al., 2009). Hence, ABM are Markov Chains on Σ with a transition matrix $\acute{p}$.For a class of ABM, the transition probabilities $\acute{p}(x,y)$ can be computed for any pair $x, y \epsilon \Sigma$ of agent configurations. The process $(\Sigma,\acute{p})$ is referred to as micro chain. When performing simulations of an ABM the actual interest is not in all the dynamical details but rather in the behavior of variables at the macroscopic level (mean job completion time, mean waiting time, mean tardiness, etc.). The formulation of an ABM as a Markov Chain $(\Sigma, \acute{p})$ enables the development of a mathematical framework for linking the Micro-description of an ABM to a Macro-description of interest. Namely, from the Markov Chain perspective, the transition from the micro to the macro level is a projection of the Markov Chain with state space Σ onto a new state space X by means of a (projection) map π from Σ to X. The meaning of the projection π is to lump sets of Micro configuration in Σ according to the macro property of interest in such a way that, for each $x \in X$, all the configurations of Σ in π^{-1} (x) share the same property.

2.5 Review of Related Literature

Scheduling, understood to be an important tool for manufacturing and engineering, has a major impact on productivity of a process (Blazewic et al., 2002). In manufacturing, the purpose of scheduling is to minimize the production time and cost, by telling a production facility what to make with which staff, and on which machine. Cited publications argued that agent-based modeling is used because only agent-based model can explicitly incorporate the complexity arising from individual behavior and interactions that exist in the real-world.

Low et al., (2004) formulated the JSP using integer programming. The study investigates the application of lot splitting in a job shop production system with set up times, which cannot be omitted. A disjunctive graph is first used to describe the addressed scheduling problem, and an integer programming model is then constructed to obtain an optimal solution. This technique involved assuming that each job consists of m operations and must pass through each machine exactly once. All machines are available at time zero. Furthermore, the total number of sub lots is given and consistent sub lot sizes are considered. Concerning the limitation of this technique, Buscher and Shen, (2009) pointed out that the problem formulation used does not recognize the physical environment of the shop floor domains where interference not only leads to readjustment of schedules but also imposes physical conditions to minimize them. Still, difficulties in the formulation of material flow constraints as mathematical inequalities and the development of generalized software solutions have limited the use of these approaches.

Taillard, (1999) demonstrated the effectiveness of tabu search algorithm for the job-shop scheduling problem. Since then, researchers have introduced numerous improvements to Taillard's original algorithm (Jain and Mecran, 2002). Yet, little progress was made in developing a theoretical understanding of these algorithms. Specifically, little is known about why tabu search is so effective for the JSP and under what conditions strong performance can be expected (Jain and Mecran, 2002).

Lawler and Wood, (2000) proposed branch-and-bound and lagrangian relaxation for solving the JSP. The basic idea of branching is to conceptualize the problem as a decision tree. This branching process continues until leaf nodes, that cannot branch any further are reached. These leaf nodes are solutions to the scheduling problem. Albert and Rebelo(2007) reported that the approach has a number of shortcomings. First, the model was developed and validated using small problem instances, leaving open the question of scalability. Second, despite good overall accuracy, model errors frequently exceed $^{1}/_{2}$, an order of magnitude in search cost, and the model is least accurate for the most difficult problem instances.

Furthermore, although efficient bonding and pruning procedures have been developed to speed up the search, this is still a very computationally intensive procedure for solving large scheduling problems. Davis and Jones, (2001) proposed a methodology for solving the job shop problem based on the decomposition of mathematical programming problems that used both Benders-type (Benders-, 1960) and Dantzig/Walfe-type (Dantzig and Walfe, 1960) decompositions. The methodology was part of closed-loop, real-time, two-level hierarchical shop floor control system. The top-level scheduler (i.e., the supremal) specified the earliest start time and the latest finish time for each job. The lower level scheduling modules (i.e., the infimals) would define these limit times for each job by detailed sequencing of all operations. A multi-criteria objective function was specified that include tardiness, throughput, and process utilization cost. The limitations of this methodology stem from the inherent stochastic nature of job shops. The presence of multiple, but often conflicting, objectives made it difficult to express the coupling constraints using exact mathematical relationships. This made the schedule not to converge. Furthermore, the rigid centralization of the scheduler made it not able to adjust to disturbances at the shop floor.

Bauer et al., (1991) evaluated the use of manufacturing resource planning(MRP or MRP-11) to create a medium-range scheduler. MRP system's major disadvantages are not only rigidity and the lack of feedback from the shop floor, but also the tremendous amount of data that have to be entered in the bill of materials and the fact that the model of the manufacturing system and its capacity are excessively simple. As can be deduced from these techniques, most approaches to tackle job-shop scheduling problem assume complete task knowledge and search for a centralized solution. These techniques typically do not scale with problem size, suffering from an exponential increase in computation time. The centralized view of the plant coupled with the deterministic algorithms characteristic of these schedulers do not allow the manufacturing processes to adjust the schedule (using local knowledge) to accommodate disturbances such as machine break downs.

Izquiredo et al., (2009) worked on and represented ten well-known simulation models as a time homogenous Markov Chain. The author's idea is the formulation of the system stochastic transition as the state space of the Markov Chain. Despite the fact that all the information of the dynamics on the ABM is encoded in a Markov chain, it is difficult to learn directly from this fact, due to the huge dimension of the configuration space and its corresponding Markov transition matrix. The work of Izquierdo and co-workers mainly relies on numerical computations to estimate the stochastic transition matrices of the models.

Sven et al., (2012) contributed to interweaving Markov Chains and agent-based modeling (ABM). The former represents the simplest form of a stochastic process while the latter puts a strong emphasis on heterogeneity and social interactions. In the model by Sven et. al. (2012), homogeneous mixing leads to a macroscopic Markov chain which underlines the theoretical importance of homogeneous mixing. An important prospect that is not exploited by Sven et. al. (2012) concerns the measure of practical emergence or discrepancy, the gap between the macro-structural properties of a system and internalized rules or intentions of the individual agents. The measure of this gap should lead to more elaborated gauges whose dynamics themselves call for new specific investigation.

Mareen and Carsten (2016), showed how to construct Markov state models that approximate the original Markov process by a Markov chain on a small finite state space and represent well the longest time scales of the original model. The approach extracts the aggregated long term dynamics of reversible Markov chains. The macrostates as well as transition probabilities between them can be estimated on the basis of short-term trajectory data. Apparent advantages of a reduced state space are that it is easier to compute eigenvalues and eigenvectors as well as other properties such as waiting times.One limitation to the

Mareen and Carsten(2016), is that the approach and its analysis depends on the original Markov chain that represents an agent-based model of interest, to be reversible. In general, it will be difficult to say whether it is reasonable to assume that an agent-based model results in a reversible Markov chain. One reason for this difficulty is that, if we estimate the transition matrix from simulated trajectory data, it does not need to fulfill the detailed balanced equation, even if the underlying Markov chain is reversible (Jan-Hendrik et al., 2011) and (Noe, 2008).

Beyond that, an approach that applies also to non-reversible Markov dynamics need to be exploited. There are few approaches that apply to non-reversible Markov chains (Marco and Christof, 2014).

Banisch (2015) suggested a graph-theoretical framework for constructing reversible surrogates of non-reversible dynamics, based on a cycle decomposition of the underlying Markov chain. However, the application to agent-based models was not treated. Therefore, the construction of Markov state models for general agent-based models is still an open problem.

Yih and Thesen (1991) formulated the scheduling problem as semi-Markov decision problems and used a non-intrusive 'knowledge acquisition' method to reduce the size of the state space. The idea was to identify and update dynamically the states and transition probabilities that are used by an expert system to solve real time scheduling issue. However, the reduced state-space and the estimated parameters cannot fully represent the original problem but an approximation. It is possible for a state to appear which is not included in the reduced state space during the operation if the simulation process does not exhaust all the possible states which can result from user decisions.

Gabel and Riedmiller (2007) modeled the job shop scheduling problem by means of a multi-agent reinforcement learning and attached to each resource an adaptive agent that makes its job dispatching decisions independently of the other agents and improves its dispatching behavior by trial and error employing reinforcement learning algorithm. Gabel and Riedmiller (2007) gave some suggestions of state feature selection, but did not consider whether these features are memoryless. The embedded Markov chain is also not mentioned in their work.

Tao et al. (2017) modeled a real-time job shop scheduling based on simulation and Markov decision processes. The main task is to decide which job in a queue should be processed next. The model uses two algorithms, simulation-based value iteration and simulation-based Q-learning were introduced to solve the scheduling problem from the perspective of a Markov decision process. The real-time job shop scheduling model by Tao et al.(2017) is a sequential decision making optimization technique. The system contains five (5) machines and produces two (2) products with two (2) operation flows. The operation flow in this model is not constrained to pass through each machine in series.

RESEARCH GAP

The flow of job in the existing models is not constrained to pass through each machine in a defined order; making it impossible for such system as seen in literature to handle efficiently job scheduling in a sequential order.

2.6 Summary ofRelated Literature

The research work done on job shop scheduling involvingnon-sequential and sequential machines were reviewed, it reveals that there is growing interest among researchers in this field.

Limited research is reported in sequential-dependent batch setup problems. When comparison of the findings is made, some gap in the current research becomes evident. These observations lead to the conclusion that there is much room for further research in this area. The following are areas that require attention of researchers;

1. There is a need to address process flexibility, operation flexibility and product flexibility as flexibility has been recognized to improve system performance. (Bauer et al.,1991), (Izquiredo et al.,2009)
2. There is need to develop setup-oriented dispatching (releasing) rule. (Sven et al., 2012)
3. Majority of reviewed articles considered only JSSPs on machine in series or parallel setup. No research is reported in JSSP on machine in series followed by one-out-of-n parallel output machine. (Tao et al., 2017)

Hence a production scheduling and control that performs reactive (not deterministic) scheduling and can make decision on which job to process next based solely on its partial (not central) view of the plant becomes necessary. This requirement puts the problem in the class of agent based model (ABM) where each job must be processed on three machines in series and the semi-processed product is passed on a one-of-three parallel finishing machine. Hence, this work adopts an alternative view on job shop scheduling problem where each resource is equipped with adaptive agent.

Chapter Three

MATERIALS AND METHOD

This chapter presents the steps taken in this research to acheive an optimal solution to the problem statement. Here a combination of Markovian process and agent-oriented analysis is used in the analysis of the proposed agent-based model for the job shop scheduling.

Classical queuing model by D. G. Kendall was used for carrying out model validation for the ABM because the order arrival is actually a queuing process.

3.1 **Methodology**

A combination of Markov Chain instruments and agent oriented analysis are used in the analysis of the proposed agent based model (ABM) for the job shop scheduling problem. The Markov Chain approach allows a rigorous analysis of key aspects of the ABM. It provides a general framework of aggregation in agent-based and related computational models.

Some of the conditions for asymptotic convergence, as for instance, the infinite length of Markov Chains, cannot be met in practice (Albert and Luis, 2009). Hence, in any finite implementation, choice have to be made with respect to the following parameters:

- The length of the Markov Chain
- The initial value of the control parameter
- The decrement rule of the control parameter

- The final value of the control parameter

Such a choice is usually referred to as cooling scheduler (Albert and Luis, 2009).

3.2 Materials

In order to achieve this research work, two major materials were used; Hardware systems and software systems

Hardware System

The hardware system used in this research work are;

i. Weighing scale used in measuring the quantity of raw aluminum in the production line. This helps to determine the acceptable quantity of semi-processed aluminum in each stage of the machine.

ii. A personal computer (PC). The PC was used in the analysis of the data, review of related works on scheduling technique, generation of secondary data, typing of the report etc. The following are the hardware requirements or configuration of the PC used in this research work.

Processing power………Intel Pentium with 1.8GHz frequency

Memory……………………..2GB RAM

Secondary storage………320GB Hard disk drive

Display adapter …………Intel (R) HD Graphics family

Peripherals………………….Keyboard, Mouse, MODEM etc

Software System

Software requirements deal with defining software resource and pre-requisites that need to be installed on a computer (PC) to provide optimal functioning of an application. For the research work carried out, the following are the software requirements used.

Operating system…………..Windows server 2012

AP/s and Drivers……………. .NET framework 3.5

Platform………………………… MATLAB R2016a (professional and student version)

Web browser…………………..Mozilla Firefox, Internet Explorer and Google Chrome.

3.3 Data Collection

The quantity of aluminum scrap (raw material) used for production in Alo Aluminum for a period of one month was measured in kilogram each day using a weighing scale, the weight was recorded. Some fraction of the scrap was rejected, while the accepted aluminum scrap was sent to the caster machine for processing. The same process was done on the molten aluminum from the caster machine where some impurity was sieved out and the accepted quantity of the semi-processed aluminum was weighed and recorded before entering the rolling mill machine. The record of the accepted semi-processed aluminum was done on each stage of the machine before it enters the next machine.The quantity of aluminum accepted in caster, rolling mill, paint line and corrugation machines in the production line on each of the days was recorded and tabulated in Table 3.1.

Table 3.1: Weight of raw material flow at the production line

Production Dates	Quantity of Aluminum Scrap Ordered (kg)	Accepted Quantity in Caster Machine (1) (kg)	Accepted Quantity in Rolling Machine (2) (kg)	Accepted quantity on Point Line Machine (3) (kg)	Accepted Quantity on the Corrugation Machine (4) (kg)
2/1/16	114	113.8	110.4	104.8	102.7
3/1/16	78	77.8	75.5	71.7	70.3
4/1/16	65	64.9	62.9	59.8	58.6
5/1/16	119	118.8	115.2	109.4	107.3
6/1/16	98	97.8	94.9	90.1	88.3
6/1/16	50	49.9	48.4	45.9	45.1
8/1/16	56	55.9	54.2	51.5	50.5
8/1/16	91	90.8	88.1	83.7	82
9/1/19	92	91.8	89.1	84.6	82.9
10/1/16	67	66.9	64.8	61.6	60.4
11/1/16	40	39.9	38.7	36.8	36.1
12/1/16	24	23.9	23.2	22.1	21.6
13/1/16	45	44.9	43.6	41.4	40.6

15/1/16	60	59.9	58.1	55.2	54.1
16/1/16	42	41.9	40.6	38.6	37.9
17/1/16	18	17.9	17.4	16.6	16.2
17/1/16	57	56.9	55.2	52.4	52.4
18/1/16	69	68.8	66.8	63.5	62.2
18/1/16	94	93.8	90.9	86.4	84.7
19/1/16	107	106.8	103.6	98.4	96.4
20/1/16	112	111.8	108.4	103	100.9
22/1/16	48	47.9	46.5	44.1	43.3
23/1/16	36	35.9	34.9	33.1	32.4
23/1/16	32	31.9	30.9	29.4	28.8
24/1/16	35	34.9	33.9	32.2	31.5
25/1/16	22	21.9	21.3	20.2	19.8
26/1/16	115	114.8	111.3	105.8	103.6
27/1/16	57	56.8	55.2	52.4	51.4
29/1/16	20	19.9	19.4	18.4	18
30/1/16	30	29.9	29	27.6	27

Source: Alo Aluminum production line

3.4Analysis of the Existing System

It was found that the case study company(Alo Aluminum) manufactures three types of roofing sheets namely metro couple corrugated sheet, step tile corrugated sheet and regular corrugated sheet, but identified in this work as finishing type 1, finishing type2 and finishing type 3 respectively. The probability of an order being either for finishing type 1 is equal to $^{1}/_{3}$; the probability of an order being of finishing type 2 is $^{1}/_{3}$; and the probability of an order being of finishing type 3 is also $^{1}/_{3}$. These facts were made use of by the order agent in the proposed system. Typically, (based on the records at Alo Aluminum company) the order size ranges from 1kg to 110kg.About thirty (30) different orders arrive within a space of a month. This fact was also made use of by the order agent when generating the orders in the proposed system. It was also found in the case study company that the production line can produce 15kg of finishing type1 per day; 19kg of finishing type 2 per day; and 30kg of finishing type 3 per day.

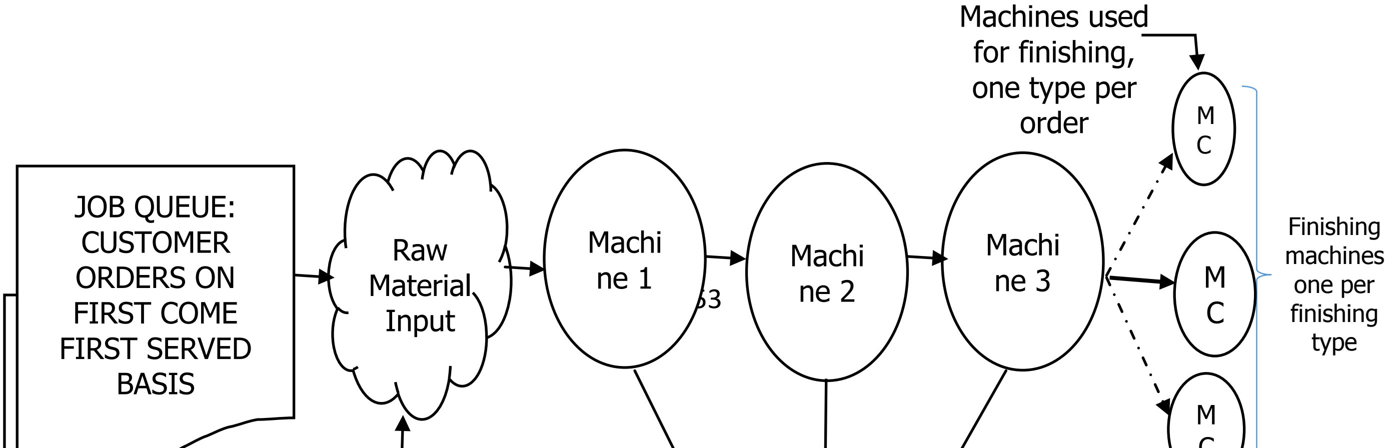


Fig. 3.1: Architecture of the Existing System. (Alo Aluminum Company)

The production process requires three machines in sequential order through which every raw material input must be processed and one-out-of three finishing machine used one per type of product. This is so because the production arrangement cannot be in parallel as the order must first be processed in machine one (1) before moving over to machine two (2). The same process has to be done on machine two (2) before machine three (3) in sequential order.

The case study company uses first come first served basis to schedule their jobs. This means that a small job may take unduly long before it may be delivered if it is positioned at the back of the queue and bigger jobs are at the front. This suggests that orders may be sorted in increasing order of size to accommodate smaller jobs first and increase the proportion of jobs that would be processed quickly. This was achie ng s a te the proposed system architecture ov ur

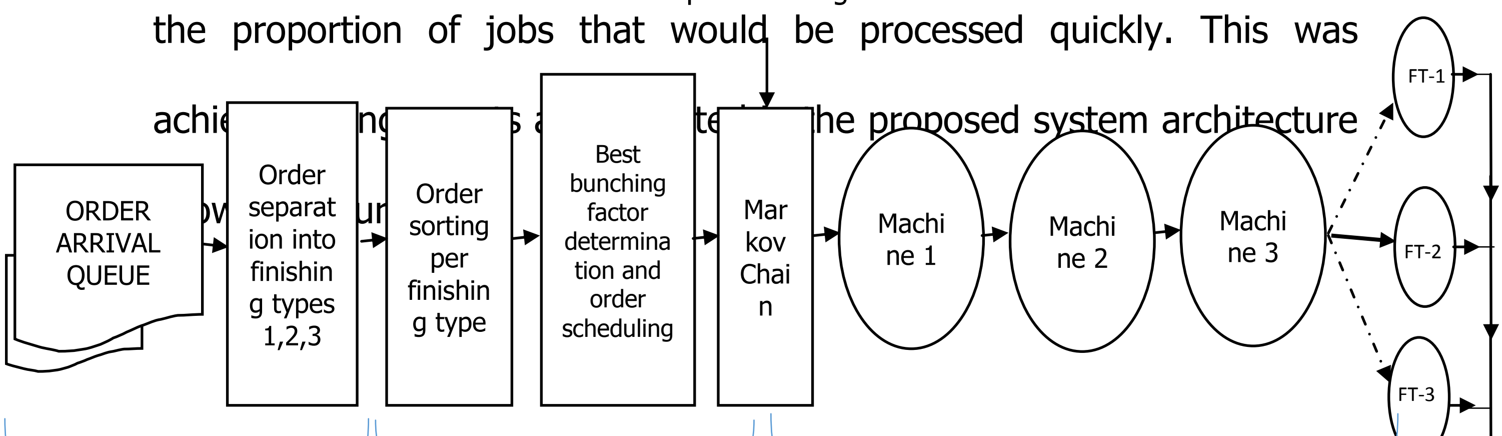


Fig. 3.2: Architecture for the proposed System.

3.5 **Limitations of the Existing System**

- The system operation obtainable with the existing first come first served method of scheduling is not flexible enough to accommodate the interest of customers with small jobs who may have joined the queue late. Thus, the first-come-first-served scheduling approach does not give an optimal result.
- The existing system does not have the engineering or scientific calculations (i.e. Markov chain method) used to determine the extra raw materials at the input needed to make up for wastages during production and yet meet the output target.

- The scheduling and control operations of the existing system do not have any opportunity for man/machine interaction often needed to accommodate certain critical contractual obligations.

3.6 Proposed System

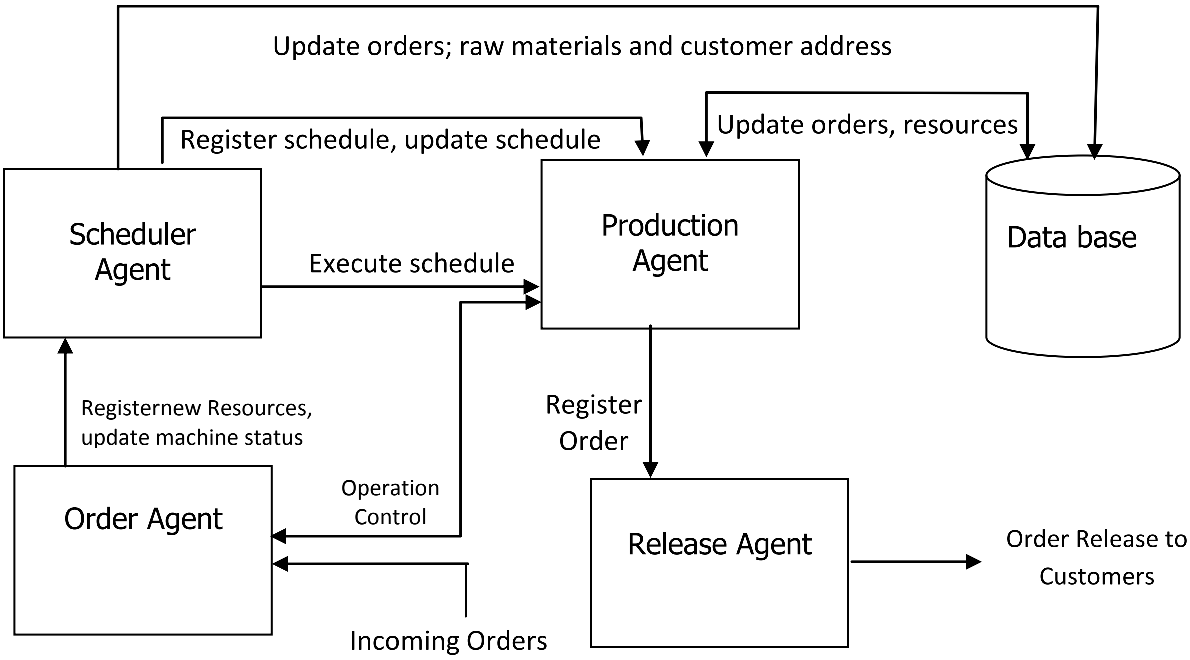


Fig. 3.3: Block Diagram of the Proposed ABM Framework for Solving the JSP

Figure 3.3 presents the block diagram of the proposed ABM and shows the interaction (information flow) among the various agents in a scheduling operation.

In order to correct the anomalies in the existing system, the model does the following job:

1. Using Markov chains the machine states were evaluated and cost of repairs and general machine maintenance were factored into the production costs.
2. The order agent receives incoming orders from customers. This is a stochastic process. However, when it is time for scheduling the received orders are passed to the scheduler in an organized form.
3. Using Markov chains, the amount of wastes in the production process were ascertained. The amount of extra raw material to add at the input end so as to obtain the desired output quantity even after the wastes was determined per kilogram of desired output.
4. Every job irrespective of the finishing type must pass through the first three machines in a sequential order. Thereafter the semi-processed output is assigned to one-of-three machines responsible for finishing types. The machine selected is the one for the finishing type required by the order being processed. The scenario informed the steps taken by the scheduler.
5. The scheduler handles the process of scheduling of a given order. It receives the order records from the order agent and then proceeds to schedule the order.Order scheduling allows order to come in up to a day before the previously scheduled order is to be

completed. It then scheduled the orders that had come in so far for the next production period while at the same time allowing fresh orders to start coming in for the period beyond the next production period.

6. The output of the scheduler is passed to the production agent. The production agent produces according to the schedule except when interrupted by routine maintenance or machine breakdown which introduces some delays.
7. The order release agent is responsible for the release of each order as its production is completed. It also prepares a bill for the owner according to the type of finish and the number of kilograms produced. It receives information about any delay from the production unit and factors them into its release timing which is passed to the customers. Such delays include those caused by routine maintenance, machine breakdown, public holidays, or any other unforeseen contingency. The release agent uses the delay information from the production agent to determine by how many days the expected delivery of an order is extended. The release agents ensure the earliest possible release of processed order to reducestock holding time to the bearest minimum. This saves

space required to hold processed stock and therefore reduces cost of space.

8. In order to validate this carefully worked out scheduling process, the same orders are passed to a classical queuing model by D.G. Kendall (1953) and the job release dates of this proposed model is compared with those achieved by this classical queuing model and compared under the following headings;

i. Order release date
ii. The date the last order was released
iii. Machine utilization, ideal time and cost.
iv. The number of customers whose expected order due date were not met.

   This is done for each of the finishing types and performance of this proposed system is then compared with the classical queuing model.

   Because of the stochastic nature of order arrival and order types, many production runs are done to show which method is producing acceptable results consistently.

3.7Model Design: Important Consideration

Based on the statement of the problem of this work, as it relates to the JSSP, the work is divided into four sub-sections; Order agent section, Scheduler agent section, Production agent section and Order release agent section.

The order agent receives incoming orders from customers, and on request passes the order records to the scheduler in an organized form. The order agent receives order in terms of;

a. Order ID
b. Time of order
c. Order size
d. Type of finish required

Scheduler agentthat uses a carefully crafted algorithm to schedule incoming orders for production was developed at the second section. Scheduler agent carries out bunching of jobs either in 1 or 2 or 3 days per finishing type and selects the best out of the three approaches. Bunching is a scheduling technique adopted in this model to schedule an order in a queue. This technique allows a certain product type to be produced for 1 or 2 or 3…..n days before changing to another product type. Thus either a finishing type is done for only one day before changing to another order in sequence which typically is of another finishing type, or one finishing type is produced for 2 or 3 continuous days before changing over to another finishing type. The bunching is not fixed at 1 or 2 or 3 days for each finishing type but the best performing bunching type is selected for each set of orders being scheduled.

The production agent produces according to the schedule except when interrupted by routine maintenance or machine breakdown which introduces some delays.

The release agent section was developed which ensures that orders are released as fast as possible.The release agent considers the earliest event dates and latest event dates of processing order to ensure that stock holding time is reduced.

The discrete event systems in terms of Markov processes in discrete or continuous time is described in this chapter. The manufacturing system is described as a finite state system, and the Markov chain describes the transitions between these states.

Consider the manufacturing operationof the existing system under study (Alo Aluminum company) that is made up of three sequential machines followed by 1 out of 3 finishing machine, fig 3.4 a.

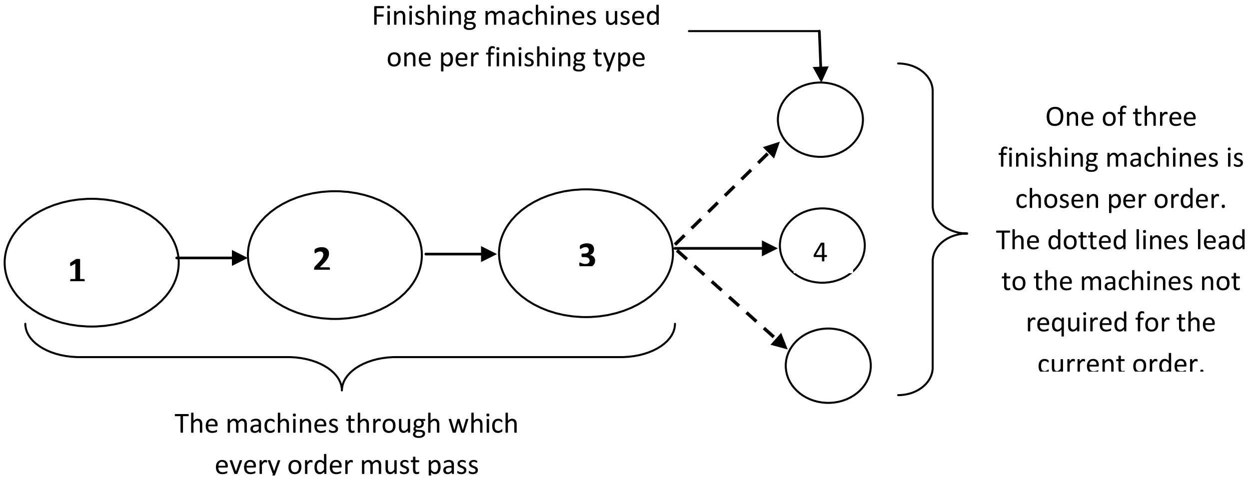


Fig 3.4 (a): Three sequential machines followed by 1 out of 3 finishing machine.

Figure 3.4 (a) presents the three sequential machines through which every order must pass and 1 out of 3 finishing machines used one per finishing type. Thus every order must pass through a total of four machines. Here, one of three finishing machines is chosen per order. The dotted line lead to machines not required for the current order finishing type.

3.8 Material flow and State Machine as Markov Chain

3.8.1Description of Material Flow as a Markov Chain

The system consists of four machines and a material flow structure. However, the fourth machine is one out of the three types used for finishing product. The exact one is determined by the customer order. A quality control measure is conducted after each operation and a product is rejected if it does not pass the quality control. After each machine the workpiece is inspected, and the piece will be rejected with a certain probability, fig 3.4b. It is assumed that the state transitions are made at time instants, so the system is modelled as a Markov chain. A sample analysis of one-month production done in the company under study (see Table 3.1) shows that the average raw material flow during inspection was 99.8% of the input pieces accepted in machine 1. 97% of the pieces operated in machine 1 was accepted in machine 2. From machine 3 and 4 there was 95% and 98% acceptancerespectively. Therefore:

Percentage Rejection

Input raw material	0.2%
Machine 1	3%
Machine 2	5%
Machine 3	2%

The system is described as a state graph in figure 3.4(b). The diagram shows the material flow through the system.

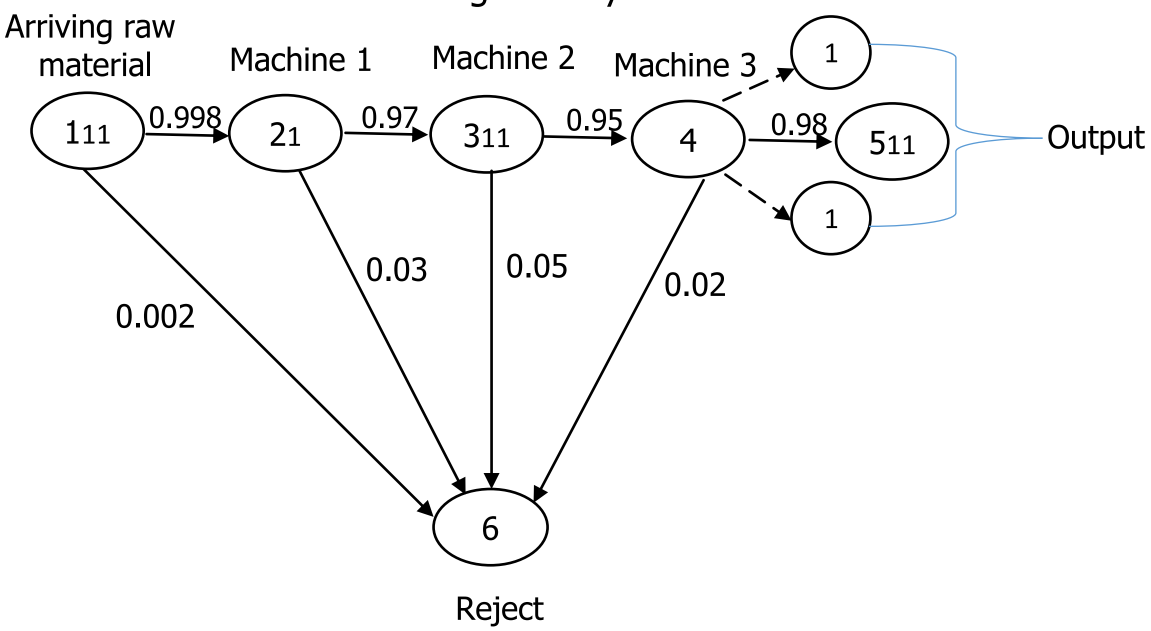


Fig. 3.4(b): State graph for the material flow

The numbers in figure. 3.4b denote the probabilities to go from one state to another. This can be written as a transition matrix in table 3.1:

Table 3.2: **Transition matrix for the material flow**

State 1	State 2	State 3	State 4	State 5	State 6
0	0.998	0	0	0	0.002
0	0	0.97	0	0	0.03
0	0	0	0.95	0	0.05
0	0	0	0	0.98	0.02
0	0	0	0	1.0	0
0	0	0	0	0	1.0

From table 3.1, the material flow end in some of the states 5 or 6. The two states are absorbing, while the other states are transient. The absorbing state probability can be calculated thus; (see fig.3.4b)

- From state 1 to 5:

$T_{1\text{ to }5}$=0.998 × 0.97 × 0.95 × 0.98 = 0.901

- From state 1 to 6:

$T_{1\text{ to }6}$= 0.002 + 0.998 × {0.03 + 0.97 (0.05 + 0.95 × 0.02)} = 0.0987

- From state 2 to 5:

$T_{2\text{ to }5}$= 0.97 × 0.95 × 0.98 = 0.903

- From state 2 to 6:

$T_{2\ to\ 6}$= 0.03 + 0.97 (0.05 + 0.95 × 0.02) = 0.097

Where T is the absorbing state probability.

Extra raw materials needed to allow for wastages using Markov chain

From Fig 3.4b, 99.8% of the input pieces was accepted in machine 1; 97% of the pieces operated in machine 1 was accepted in machine 2. From machine 3 and 4 there was 95% and 98% acceptance respectively. Therefore, out of 100Kg raw material units, we will only accept, 100 x 0.998 x 0.97 x 0.95 × 0.98 = 90.13Kg units (at an average). In order to produce 100Kg units, $\frac{100}{0.901} = 111$kg raw material input is needed. Therefore, one has to enter 111Kg raw material pieces into the system.

Therefore, the amount of raw materials per Kilogram of output is

$$\frac{100}{90} = 1.111$$

If 1Kg of input raw material cost N Naira, then

- (1.111 x 15) = 16.67Kg of raw materials must be supplied per day to the machine when producing finishing type 1 at 15kg per day.

- (1.111×19) = 21.11Kg of input raw materials is needed per day when producing finishing type 2 at 19Kg per day.
- $(\frac{100}{90}\ x\ 30)$ = 33.33Kg of input raw materials is needed per day when producing finishing type 3 at 30Kg per day.

Mathematical Expression for the Raw Material Flow using Markov Chain

Recall the general expression of a typical Markov chain;

$P_r(X_{n+1}=x/X_n=X_n) = P_{ij}$

Where P_r is the probability

$P_{ij}$is one-step transition probability; the probability that the chain, whenever in state i, moves next (one unit of time later) into state j

And $(X_{n+1}=x/X_n=X_n)$ is the state value (Quantity of accepted raw material)

Therefore, the expression for the raw material flow is;

$P_r\ (0.901) = P_{ij}$

Costing of finished goods.

i. If X kg of output is desired at a unit cost of N Naira per kilogram, then the proportion of cost of the finished goods attributable at material consumption is therefore (X . N) Naira.

ii. Other cost factors for finished goods includes the following

- Cost of labour or machine operators
- Cost of machine time consumed and markup added by the company. For example, if Y Naira is the cost per day for machine operators in a process producing C kg per day then the cost of operator to produce x kg of output would be $Y * \frac{X}{C}$

3.8.2Description of the State Machine as a Markov Chain

The manufacturing operation uses three machines in sequence through which every job must pass and one out of three finishing machine depending on the type of order. A machine that suffers a major breakdown twice a year while being used for continuous production is deemed to be 99% in good shape. When the number of major breakdown rises to three (3) times per year when in continuous use the machine is deemed to be 98% efficient. A machine that suffers up to four (4) breakdowns in a year while being in continuous use is deemed to be 94% efficient. Note that every machine used for production is serviced once a month as preventive maintenance. These are not counted as breakdown. A machine that is less than 96% efficient is considered unusable in a major continuous production process because the wastages willbe too high for comfort. This is the view of the experts in the production line (Alo Aluminum). Such a machine is due to be sold at a scrap value and replaced. The diagram of figure 3.5 presents the state graph for the production machine. The state machine probability can be estimate. Assume that the state transitions are made at time instants, so the system is modelled as a Markov chain.

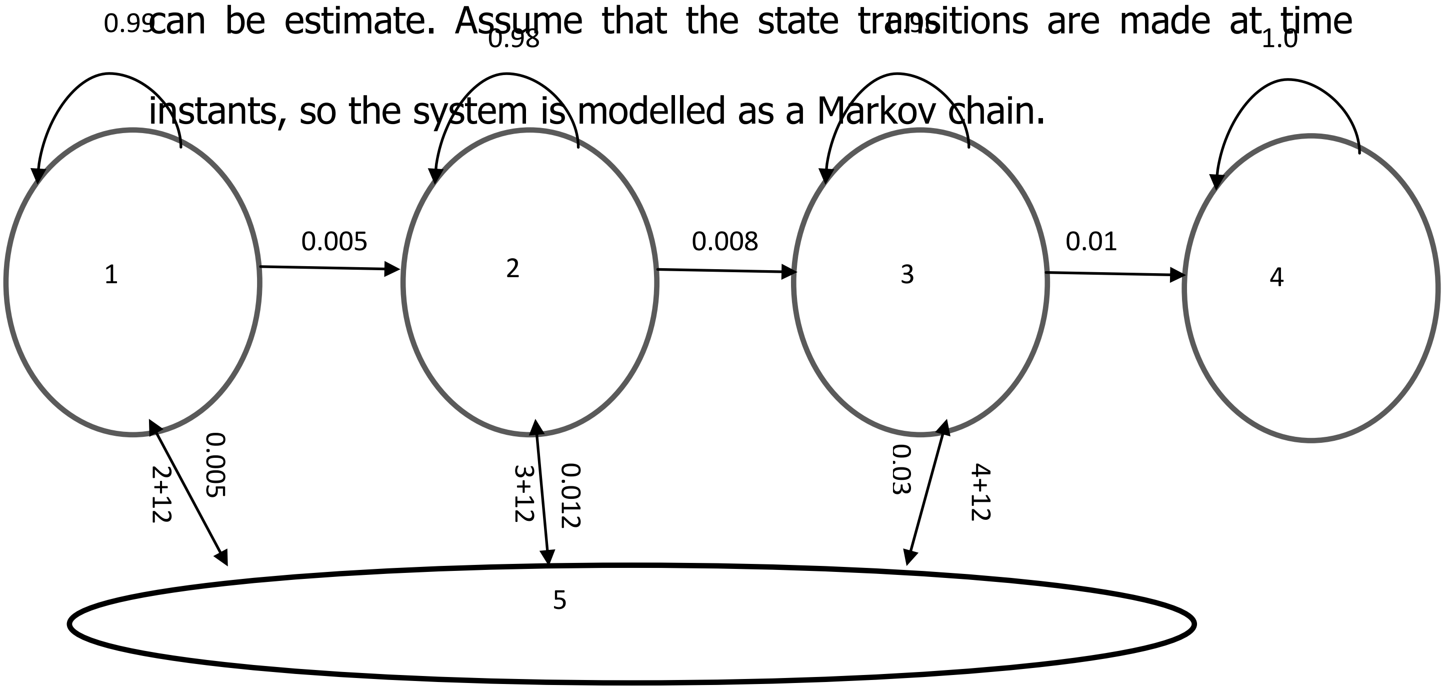


Fig. 3.5: State graph for the Machine

The machine goes for servicing at state 5 (Buffer or Sink) after a major breakdown or during routine maintenance. 0.99 is the proportion of fitness of the machine at the first year, while the deficiency in the machine that affect the output is 0.005 and the proportion for the machine maintenance is 0.005. Deferent proportions of fitness and unfitness were gotten for next four years of machine operation as presented in figure 3.5.

Table 3.3: Transition matrix for the machine state

	State 1	State 2	State 3	State 4	State 5
State 1	0.99	0.005	0	0	0.005
State 2	0	0.98	0.008	0	0.012
State 3	0	0	0.96	0.01	0.03
State 4	0	0	0	1.0	0
State 5	0.005	0.012	0.03	0	0

The machine state Probability can be derived using

vP =v

Where v = machinesteady-state vector

P = Transition probability

For the five states, let v = $\begin{bmatrix} a \\ b \\ c \\ d \\ e \end{bmatrix}$

Where;a is the state probability representing the proportion of time that the machine would be in state 1,

b is the state probability representing the proportion of time that the machine would be in state 2, c is the state probability representing the proportion of time that the machine would be in state 3,

d is the state probability representing the proportion of time that the machine would be in state 4 and

e is the state probability representing the proportion of time that the machine would be in state 5

Therefore, v = Pv

$$= \begin{pmatrix} a \\ b \\ c \\ d \\ e \end{pmatrix} \quad \begin{bmatrix} 0.99 & 0.005 & 0 & 0 & 0.005 \\ 0 & 0.98 & 0.08 & 0 & 0.01 \\ 0 & 0 & 0.96 & 0.01 & 0.03 \\ 0 & 0 & 0 & 1.0 & 0 \\ 0.005 & 0.012 & 0.03 & 0 & 0 \end{bmatrix} \begin{pmatrix} a \\ b \\ c \\ d \\ e \end{pmatrix}$$

Multiplying through, we have

$$0.99a + 0.005b + 0c + 0d + 0.005e = a \qquad 3.1$$

$$0a + 0.98b + 0.008c + 0d + 0.01e = b \qquad 3.2$$

$$0a + 0b + 0.96c + 0.01d + 0.03e = c \qquad 3.3$$

$$0a + 0b + 0c + 1.0d + 0e = d \qquad 3.4$$

$$0.005a + 0.012b + 0.03c + 0d + 0e = e \qquad 3.5$$

From equation 3.1;

$$0.01a = 0.005b + 0.005e = 0.005\ (b + e)$$

$$\therefore a = 0.5(b + e)$$

From equation 3.2;

$0.02b = 0.008c + 0.01e$ $\Rightarrow b = 0.4c + 0.5e$

From equation 3.3;

$0.04c = 0.01d + 0.03c$ $\Rightarrow c = 0.25d + 0.75e$

From equation 3.5;

$e = 0.005a + 0.012b + 0.03c$ 3.6

Substituting the values of a, b and c in equation 3.6, we have;

$e = 0.0025b + 0.0025e + 0.0048c + 0.006e + 0.0075d + 0.00225e$

$e = 0.0025b + 0.0048c + 0.0075d + 0.031e$

$0.969e = 0.0025b + 0.0048c + 0.0075d$

$= 0.0025 (0.4c + 0.5e) + 0.0048c + 0.0075d$

$0.96775e= 0.0058c + 0.0075d$

$= 0.0058 (0.25d +0.75e) + 0.0075d$

$0.96775e = 0.00145d + 0.00435e + 0.0075d$

$0.9634e = 0.00895d$

$e= 0.00929d$ or $d = 107.64246e$

$c = 26.910615e + 0.75e = 27.660615e$

$b = 11.064246e + 0.5e = 11.564246e$

$a = 0.5 (11.564246e + e) = 6.282123e$

$e = 0.00929d = 0.9999985e$

$$V = \begin{bmatrix} 6.282123e \\ 11.564246e \\ 27.660615e \\ 107.64246e \\ 0.9999985e \end{bmatrix} \qquad 3.7$$

From equation 3.7, the matrix is equal to 1;

6.282123e + 11.564246e+27.660615e+107.64246e + 0.9999985e = 1

154.14945e = 1

$\therefore e = 0.0064872$

d= 0.6982982
c = 0.1794399
b = 0.0750196
a = 0.0407534

$$\therefore V = \begin{bmatrix} 0.0408 \\ 0.0750 \\ 0.1794 \\ 0.6983 \\ 0.0065 \end{bmatrix}$$

Therefore, the required steady-state probabilities (i.e. The state probability representing the proportion of time that the machine would be in state 1, 2, 3, 4 and 5 respectively) are,

a= 0.0408, b = 0.0750, c = 0.1794, d = 0.6983, e = 0.0065

Machine Maintenance Costing

If it cost x Naira to overhaul the machine (including lost time) on the average and y Naira as production lost if the machine is found inoperative in n years, then the expected cost of maintenance per day will be; the steady state probability of the machine in state 1 multiply by the overhaul cost plus the steady state probability of state 5 (Buffer or Sink) multiply by the production lost during maintenance or routine servicing (Sharma, 2013).

Therefore; $0.0408 \times x + 0.0065 \times y$

But the production cost (y) includes →Raw material cost (R)

Labour Cost (L)

Machine cost (M)

Extra raw material cost (E)

Recall that the production line can produce 15kg of finishing type 1 per day; 19kg of finishing type 2 per day; and 30kg of finishing type 3 per day. Therefore;

1. The raw material cost (R) for finished type 1 = 15Kg X

for finished type 2 = 19KgX

for finished type 3 = 30KgX

2. The extra raw material for finished type 1 = (16.95 – 15)Xkg

for finished type 2 = (21.59 – 19)Xkg

for finished type 3 = (34.09 – 30)Xkg

3. The averagelabour cost per day for an operator from investigation in the company is = ₦2,500
4. Equipment cost per day is (considering three years of service for each machine)

$$= \frac{Capital\ \ (or\ purc\ hase\ )cost - scrap\ \ value\ \ at\ the\ end\ \ of\ \ t\ years + Running\ \ cost\ \ for\ \ t\ years}{52.Wk \times 5yrs \times 6days}$$

Where 52wk is the total number of weeks in a year,

5yrs is the acceptable number of years the machine will be used before selling it as scrap and

6days is the number of working days in a week.

The cost of overhaul per day is; X (Average number of breakdown for machines in state {1+2+3}).

Where X is the equipment cost per day.

3.9 The Schedule Processing

As part of the schedule Agent list of intentions, is the execution of the algorithm (i.e Run Schedule Algorithm intentions) that satisfy all constraints. The production agent uses the projected distributions (worked out with Markov Chain) i.e. intention of the production agent to initiate the production process. The agent continues to run this process in the background while reacting to disturbances from the factory floor. For example, when a new order arrives, when a job or operation is preempted or a machine becomes unavailable, the agent updates the schedule and re-iterates the process. Also when backtracking is detected (based on constraint checking), the agent adapts by either running another schedule from its schedule list or dumping the entire schedule and then re-computing the sequence. The objective of the algorithm is to schedule N jobs on M machines (while taking stochastic conditions into effect) so that the makespan MS is minimized.

Since multiple job operations are incorporated, the original job i with operation j will be called job (i,j). In addition, a job (i,j) can be processed on any machine k in set M_{ij} and processing times P_{ijk} may be different. The process scenario adopted for scheduling of jobs here uses three sequential machines followed by one out of three finishing machine. In this algorithm,jobs of varying sizes and levels are scheduled on the first three machines sequentially, then the output or the semi-processed product is passed on to any of the finishing (fourth) machine on a one out of three bases depending on the type of order. The algorithm allows for bunching of job in either three or two or one day, where the best bunch is selected except where the need for preferential consideration (i.e. government Job) is highly needed, human/machine interaction will be invoked. Each complete run of the algorithm terminates once all jobs have been scheduled.

A detailed description of the algorithm is now given.

- **Schedule Algorithm**

A carefully worked out procedure used to achieve the set objectives isas follows;

a. The system sorts the entire order into three parts according to the type of finish desired of each. All the orders of finishing type one are together, finishing type two are together and finishing type three are together after the sorting.
b. Each finishing type is again sorted in ascending order of size of order in kilograms.
c. Thereafter the scheduler agent schedules the orders as follows; the first of type one is followed by the first of type two, followed by the first of type three. Then the second of type one follows in the schedule and after that, the second of type two and the second of type three, and so on.
d. Because the machine must be kept as busy as possible, slacks are introduced into each finishing type to ensure that the production

of that type occupies only full days. Thus, once the machines start producing a particular finishing type, it must continue with that type throughout the working day before it can change to another type at the beginning of the next day if need be.

e. Simulate a schedule with bunching factor of 1, this means one type of finish is done each day, example;

Day 1 = finishing type 1

Day 2 = finishing type 2

Day 3 = finishing type 3

Day 4 = finishing type 1 and so on.

f. Simulating the scheduling using a bunching factor of 2 that is;

Finishing type 1 = first two days

Finishing type 2 = days 3 and 4

Finishing type 3 = days 5 and 6

Finishing type 1 = days 7 and 8 and so on.

g. Simulate the scheduling using a bunching factor of 3 this means;

Finishing type 1 = days 1, 2 and 3

Finishing type 2 = days 4, 5 and 6

Finishing type 3 = days 7, 8 and 9

Finishing type 1 = days 10, 11 and 12 and so on.

h. Select the bunching factor that yields the earliest finishing date for the order and use that bunching factor for scheduling the order.

3.10 Software Sub-System Model

The flow chart showing the agent-based model for job shop scheduling and control is presented in figures 3.6, 3.7, 3.8 and 3.9. It presents the order agent activity flow chart, the scheduler agent activity flow chart, production agent activity flow chart and the release agent activity flow chart.Matlab control program for the agent activity is presented in this section.

3.10.1 Order Agent Flow Chart and the Control Program

Figure 3.6,is the activity flow chart for the order agent. The Order Agent receives all incoming orders and on request passes the order records to the scheduler.

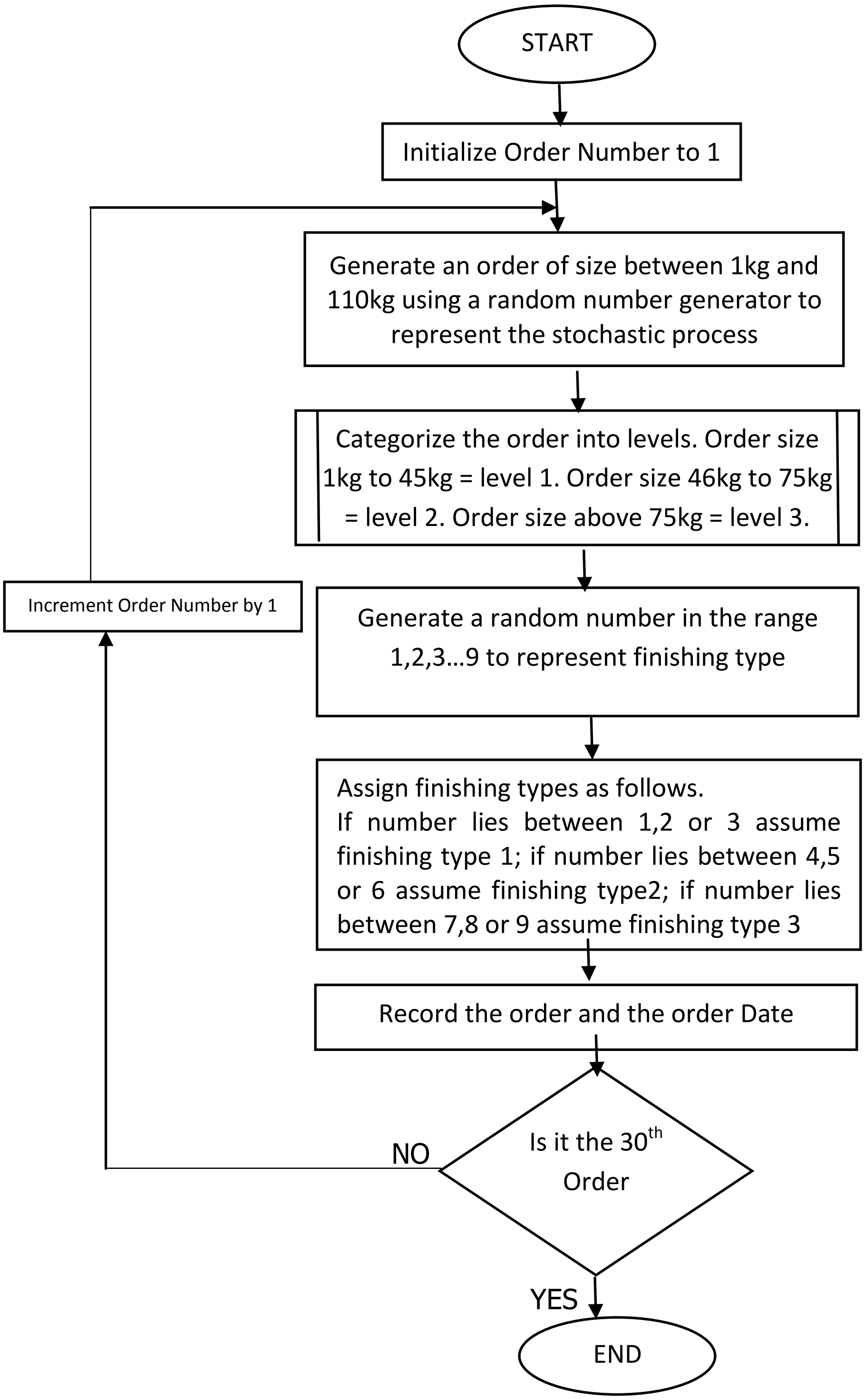

START
Initialize Order Number to 1
Generate an order of size between 1kg and 110kg using a random number generator to represent the stochastic process
Categorize the order into levels. Order size 1kg to 45kg = level 1. Order size 46kg to 75kg = level 2. Order size above 75kg = level 3.
Increment Order Number by 1
Generate a random number in the range 1,2,3…9 to represent finishing type
Assign finishing types as follows.
If number lies between 1,2 or 3 assume finishing type 1; if number lies between 4,5 or 6 assume finishing type2; if number lies between 7,8 or 9 assume finishing type 3
Record the order and the order Date
NO
Is it the 30th Order
YES
END

Fig. 3.6:Order agent's activity flow chart

The Matlab program for the order generation that follows the order arrival pattern of the case study company is presented in appendix A.

3.10.2 **Scheduler Agent Flow Chart and the Control Program**

Figure 3.7,shows the activity flow chart for the scheduler agent. The scheduler follows this established algorithm to schedule the orders for production.

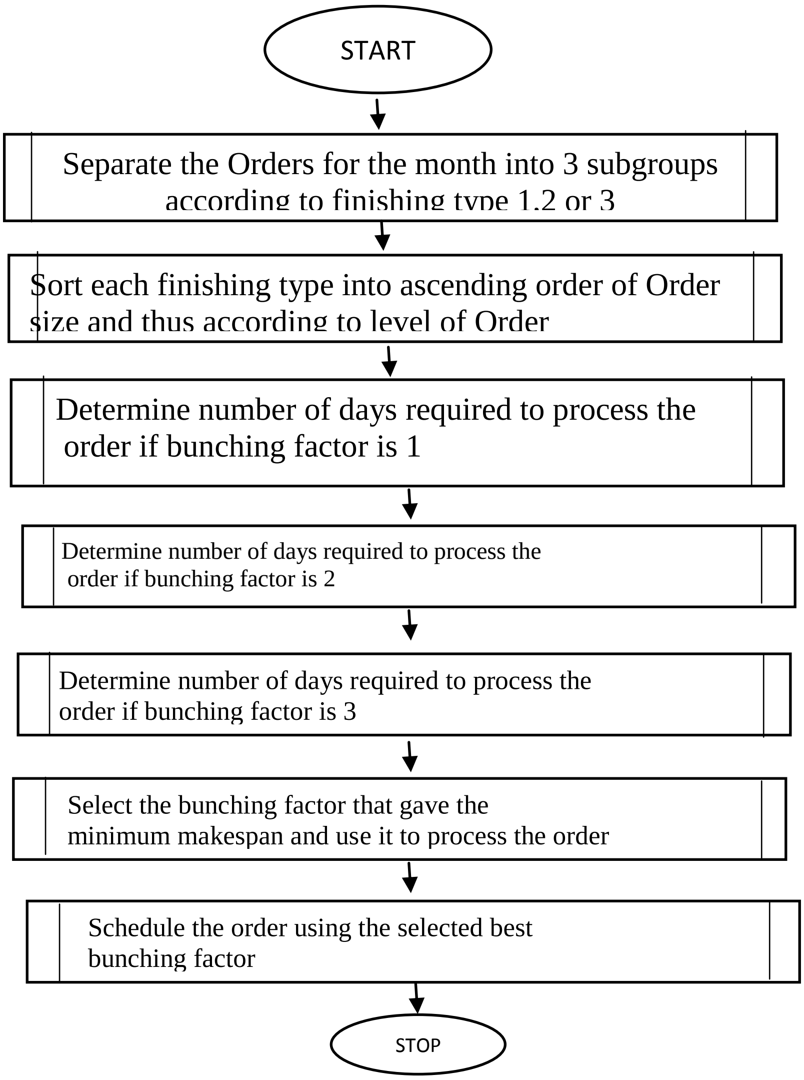


Fig. 3.7: Scheduling agent's activity flow chart

The scheduler agent program codes that follows the bunching technique used in the model for the scheduling of jobs on machines is presented in appendix B.

3.10.3 **Production Agent Flow Chart and the Control Program**

The activity flow chart for the production agent is shown in figure 3.8.

The Production Agent handles the production process. It also notifies the order release agent of any unforeseen delays in the production timing.

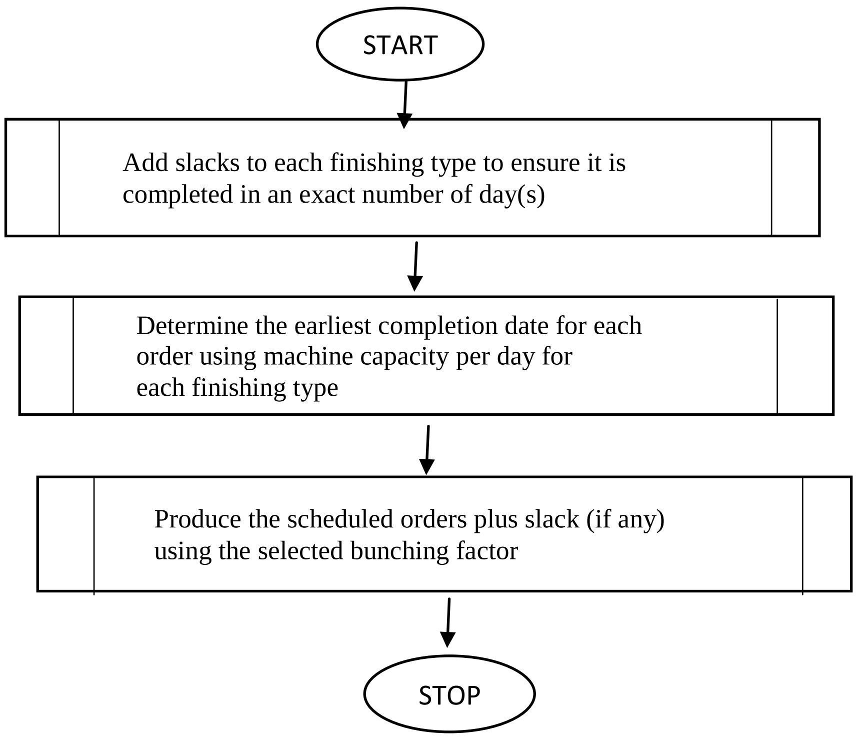


Fig. 3.8: Production agent's activity flow chart

The production agent control program that produces the schedule order based on the best bunching factor is shown in appendix C**.**

3.10.4 **Order Release Agent Flow Chart and the Control Program**

The activity flow chart for the order release agent is shown in figure 3.9. The Order Release Agent computes appropriate charges for each order, determines due date for each order and notifies the owner, and releases each order at the earliest event time.

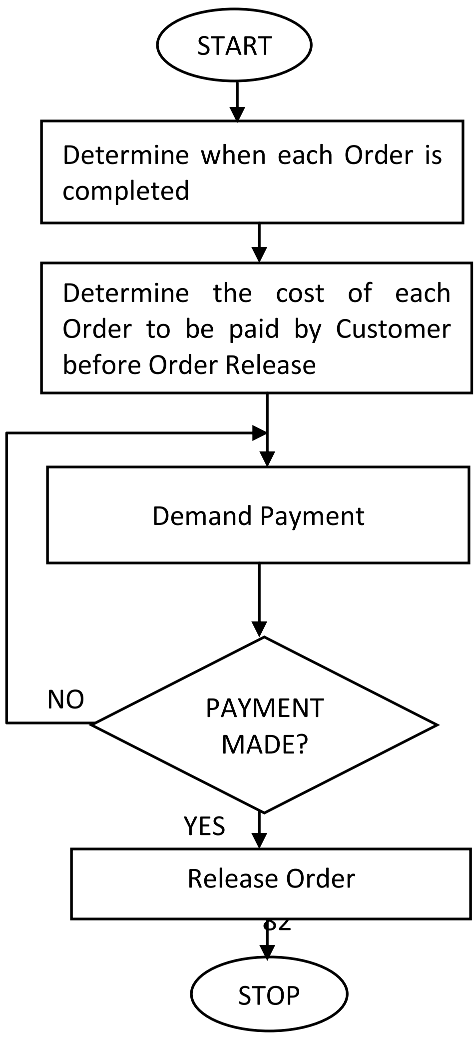


Fig. 3.9: Order release agent's activity flow chart

The order release program code for the finished jobs is presented in appendix D.

3.10.5 **Order Release Agent Costing of Finished Order**

The cost factors considered and factored into the order release agent for the release of finished order to customer is worked out as follows;

Let X be the size of order from the customers.

C_1 be the raw material cost per kilogram. The data from the case study companies, the cost per kilogram of raw materials is ₦400.

C_2 be the operator cost per kilogram.

C_2= salary/kilogram.

The monthly salary for the operator is ₦75000.

The average kilogram of finished product for the three finishing types per day is $(15 + 19 + 30)/3 = 21$kg.

Therefore, $C_2 = (75000)/(30 \times 21)$

C_3 be the cost of machine depreciation per kilogram.

From the information gathered from the company under study, the case study companyworksfor six (6) days in a week (i.e. 312 days in a year).

The machines used by the company, works efficiently for five (5) years. The capital cost + Running cost – Scrap value for the machines is ₦120.5million (based on information of the last scraped machine from the company)

C_3 = (120500000)/5 × 312 × 21) = ₦3678.26 per kilogram

Subtotal is X (C_3 + C_3+ C_3)

The costing worked out is presented in table 3.4.

Table 3.4: **Costing of Finished Product**

Order(kg)	**Raw material cost per kg (₦)**	**Operator cost per kg (₦)**	**Machine depreciation per kg (₦)**	**Subtotal (₦)**	**Mark up 30% of sub total**	**Total cost (₦)**
16	400	119	3678	67152	20145.6	87297.6
13	400	119	3678	54561	16368.3	70929.3
13	400	119	3678	54561	16368.3	70929.3
13	400	119	3678	54561	16368.3	70929.3
36	400	119	3678	151092	45327.6	196419.6
27	400	119	3678	113319	33995.7	147314.7
20	400	119	3678	83940	25182	109122
22	400	119	3678	92334	27700.2	120034.2
38	400	119	3678	159486	47845.8	207331.8
32	400	119	3678	134304	40291.2	174595.2
29	400	119	3678	121713	36513.9	158226.9
40	400	119	3678	176880	50364	218244
50	400	119	3678	209850	62955	272805
43	400	119	3678	180471	54141.3	234612.3
51	400	119	3678	214047	64214.1	278261.1
54	400	119	3678	226638	67991.4	294629.4
59	400	119	3678	247623	74286.9	321909.9
51	400	119	3678	214047	64214.1	278221.1
62	400	119	3678	260214	78064.2	338278.2
82	400	119	3678	344154	103246.2	447400.2
60	400	119	3678	251820	75546	327366
83	400	119	3678	348351	104505.3	452856.3

104	400	119	3678	436488	130946.4	567434.3
85	400	119	3678	356745	107023.5	463768.5
70	400	119	3678	293790	88137	381927
88	400	119	3678	369336	110800.8	480136.8
102	400	119	3678	428094	128428.2	556522.2
96	400	119	3678	402912	120873.6	523785.6
107	400	119	3678	449079	134723.7	583802.7
101	400	119	3678	423897	127169.1	551066.1

3.11. **Agent Interaction Model**

An agent activity diagram (agent event diagram) models the interaction among the agents of a system (Champ et. Al.,2003). The message should also be understood as events. The agent's desires are event triggered.

The activity diagram of the system is presented in figure 3.10.

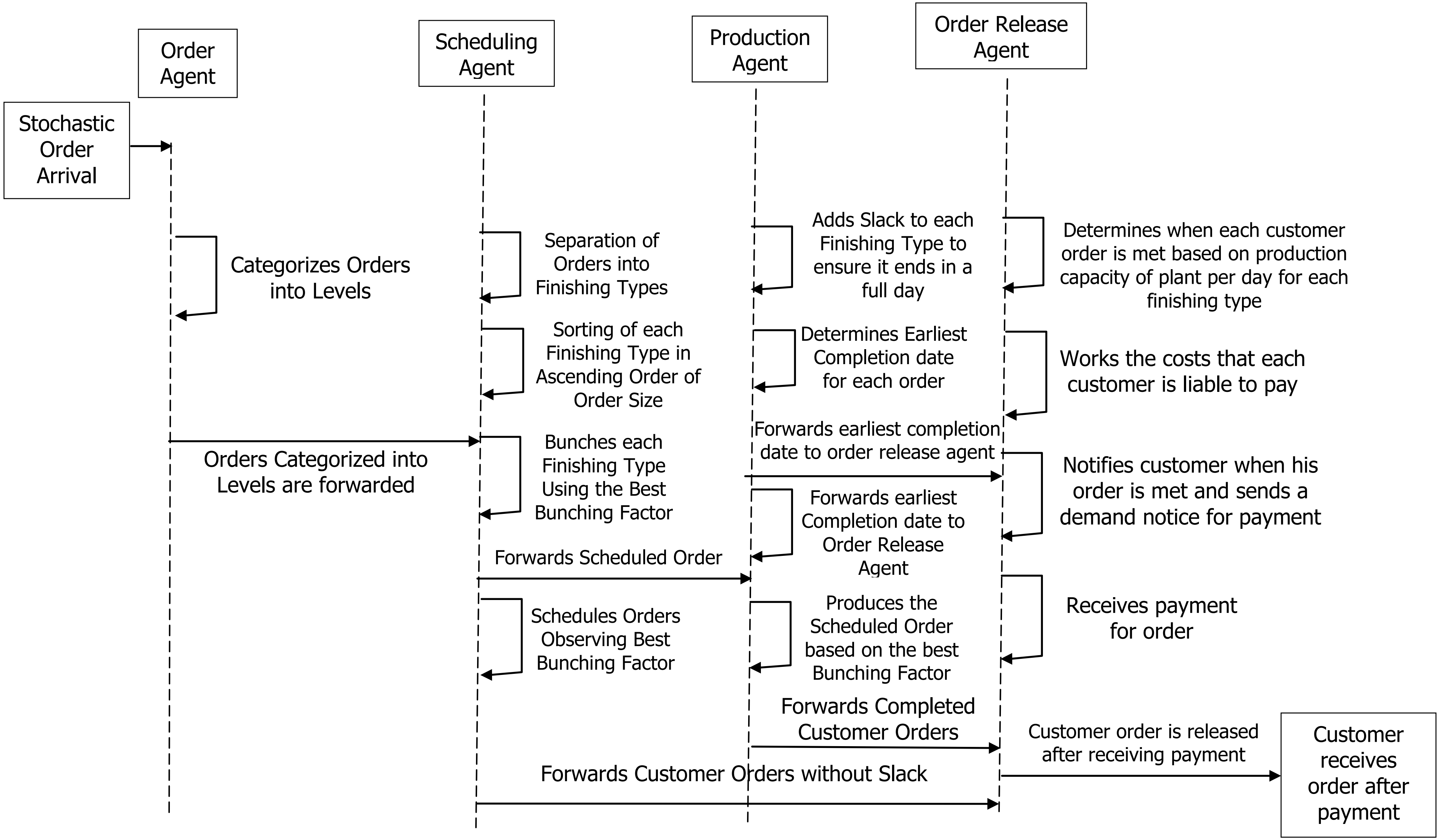


Fig. 3.10 Sequence diagram for the Agent Based Job-Shop Scheduling

Figure 3.10, shows the sequence diagram for the agent based job shop scheduling. Order arrival to the order agent is a stochastic process. When an agent does a job that is not forwarded to another agent, the arrow showing that process points back to the agents itself. When the outcome of an agent's process is communicated to another agent the arrow indicating that process points forward to the receiving agent. The processes carried out by an agent are placed under that agent in the order in which they are done (figure 3.10).

Before the customer receives an order, he is expected to have paid for the order. It is the responsibility of the order release agent to notify each customer of their order completion and at the same time to send to them a demand notice for the cost.

3.12 Design of Database Tables

The structure of database tables is shown below. In designing the database table, the field name is chosen to match the type of item being stored in the field for example, order ID, order date, order size and finishing type all representing the type of item stored in the field. The field type can either be numeric, integer only, alpha numeric or standard date type. The field width is determined by checking the number of character spaces, the largest value that fieldwould occupy. Field description gives fuller detail about the field name and primary key is the one that can be used for sorting. For example, one may sort an order according to ID or according to finishing type.

Table 3.5: Unsorted Customer Order Table

Field Name	**Field Type**	**Field Width**	**Description**	**Primary Key**
Order ID	Integer	4	Identifies an order	Yes
Order Date	Date	8	Order date	No
Order Size (kg)	Integer	3	Size of order (Kg)	No
Finishing Type	Integer	1	Type of finish	Yes

Field Name	**Field Type**	**Field Width**	**Description**	**Primary Key**
Order ID	Integer	4	Identifies an order	Yes
Order Date	Date	8	Order date	No
Order Size (kg)	Integer	3	Size of order (Kg)	No
Order level	Integer	1	How big an order is	No
Finishing Type	Integer	1	Type of finish	Yes

Table 3.6: Sorted Customer Order Table

Field Name	**Field Type**	**Field Width**	**Description**	**Primary Key**
Order Month	Integer	2	Month order was placed	Yes
Finishing Type	Integer	1	Type of Finish	Yes
Bf1 (days)	Integer	3	Days to Finish order Type with Bunching Factor 1	No
Bf2 (days)	Integer	3	Days to Finish order Type with Bunching Factor 2	No
Bf3 (days)	Integer	3	Days to Finish order Type with Bunching Factor 3	No
Best Bf	Integer	1	Best Bunching Factor	No

Table 3.7: Simulated Result for Ten (10) Different Order with Bf1, Bf2 and Bf3

Field Name	**Field Type**	**Field Width**	**Description**	**Primary Key**
Order Day	Integer	2	Order Day (1 to 30)	Yes
Order Month	Integer	2	Month of Order	Yes
Best Bf	Integer	1	Best Bunching Factor	No
Finishing Type	Integer	1	Type of Finish	Yes
Order Size (kg)	Integer	3	Size of Order (kg)	No
Release Days	Integer	3	How Long it takes to Release Order	No
Earliest Event Date	Character	8	Earliest Time for Release	No
Latest Event Date	Character	8	Latest Time for Release	No

Table 3.8: Scheduling Result for the Best Bunching Factor

Table 3.9: Release Date for Unsorted Order 1-10

Field Name	**Field Type**	**Field Width**	**Description**	**Primary Key**
Order Day	Integer	3	Day Order was Placed	No
Order Release Days for Month 1	Integer	3	Order Release Day in Month 1	Yes
Order Release Days for Month 2	Integer	1	Order Release Day in Month 2	Yes
Order Release Days for Month 3	Integer	1	Order Release Day in Month 3	Yes
Order Release Days for Month 4	Integer	3	Order Release Day in Month 4	Yes
Order Release Days for Month 5	Integer	3	Order Release Day in Month 5	Yes
Order Release Days for Month 6	Integer	3	Order Release Day in Month 6	Yes

Order Release Days for Month 7	Integer	3	Order Release Day in Month 7	Yes
Order Release Days for Month 8	Integer	3	Order Release Day in Month 8	Yes
Order Release Days for Month 9	Integer	3	Order Release Day in Month 9	Yes
Order Release Days for Month 10	Integer	3	Order Release Day in Month 10	Yes

Table 3.10: Release Date for sorted Order 1-10

Field Name	**Field Type**	**Field Width**	**Description**	**Primary Key**
Order Day	Integer	3	Day Order was Placed	No
Order Release Days for Month 1	Integer	3	Order Release Day in Month 1	Yes
Order Release Days for Month 2	Integer	1	Order Release Day in Month 2	Yes
Order Release Days for Month 3	Integer	1	Order Release Day in Month 3	Yes
Order Release Days for Month 4	Integer	3	Order Release Day in Month 4	Yes
Order Release Days for Month 5	Integer	3	Order Release Day in Month 5	Yes
Order Release Days for Month 6	Integer	3	Order Release Day in Month 6	Yes
Order Release Days for Month 7	Integer	3	Order Release Day in Month 7	Yes
Order Release Days for Month 8	Integer	3	Order Release Day in Month 8	Yes
Order Release Days for Month 9	Integer	3	Order Release Day in Month 9	Yes
Order Release Days for Month 10	Integer	3	Order Release Day in Month 10	Yes

CHAPTER FOUR

RESULTS AND DISCUSSION

4.0 Preamble

This chapter presents the implementation of the proposed agent-based job shop schedulingmodel to achieve the aim and stated objectives. The analysis was done using ten (10) monthorder obtained from the case study companyto achieve a model result as conceived. The model result agreed with the classical model during validation.

4.1 Implementation

The proposed modelseeks to obtain an agent-based scheduler that is optimized for handling job shop scheduling that ensures efficient and profitable manufacturing automation.The activities carried out to achieve the aim of the research work are;

i. The orders gotten from the customers for thirty (30) days are grouped into three different finishing types, see table 4.1(a-j).
ii. Each finishing type is sorted in ascending order of job size with respect to the finishing type before scheduling, see table 4.2(a-j).
iii. Bunching of each finishing type of job with a bunching factor (Bf) of 1 or 2 or 3 was used to schedule the job.
iv. Selecting the best bunching factor for each order, this means the bunching factor that gives the earliest finishing time for all the orders. See table 4.3.
v. Test running the carefully crafted algorithm on ten (10) separate orders.
vi. Scheduling with the best bunching factor (Bf) for each of the ten different orders and that lead to the latest finishing dates at the bottom of the table 4.3a.

4.1.1 Results for Grouping of Order into Finishing Types and Levels

The order gotten from the customers are grouped into three different product finishing types as demanded by the customers, with each type having different order size. Table 4.1a shows the listing of order from the customers for a month.

Table 4.1a: Order for month 1 from customers

Order ID	Order TIME	Order SIZE (kg)	Finishing Type
1001	2/01/16	102	3
1002	3/01/16	70	1
1003	4/01/16	59	3
1004	5/01/16	107	3
1005	6/01/16	88	3
1006	6/01/16	13	3
1007	8/01/16	50	3
1008	8/01/16	82	3
1009	9/01/16	83	3
1010	10/01/16	60	1
1011	11/01/16	36	3
1012	12/01/16	22	2
1013	13/01/16	40	3
1014	15/01/16	54	3
1015	16/01/16	38	3
1016	17/01/16	16	1
1017	17/01/16	51	2
1018	18/01/16	62	2
1019	18/01/16	85	3
1020	19/01/16	96	1
1021	20/01/16	101	1
1022	22/01/16	43	1
1023	23/01/16	32	1
1024	23/01/16	29	2
1025	24/01/16	13	2
1026	25/01/16	20	2
1027	26/01/16	104	2
1028	27/01/16	51	1
1029	29/01/16	13	2
1030	30/01/16	27	1

Table 4.1b: Order for month 2 from customers

Order ID	Order TIME	Order SIZE (kg)	Finishing Type
1031	1/02/16	38	2
1032	2/02/16	78	2
1033	2/02/16	55	2
1034	3/02/16	87	3
1035	4/02/16	109	3
1036	4/02/16	45	1
1037	6/02/16	23	1
1038	6/02/16	30	2
1039	7/02/16	56	1
1040	8/02/16	23	1
1041	9/02/16	39	1

1042	11/02/16	63	2
1043	11/02/16	35	2
1044	13/02/16	101	2
1045	14/02/16	34	1
1046	15/02/16	75	1
1047	16/02/16	47	3
1048	16/02/16	60	2
1049	17/02/16	62	3
1050	18/02/16	30	2
1051	20/02/16	71	1
1052	21/02/16	98	3
1053	22/02/16	79	3
1054	23/02/16	22	1
1055	24/02/16	46	1
1056	24/02/16	51	1
1057	25/02/16	25	1
1058	26/02/16	83	1
1059	27/02/16	104	2
1060	28/02/16	72	3

Table 4.1c: Order for month 3 from customers

Order ID	Order TIME	Order SIZE (kg)	Finishing Type
1061	1/03/16	69	1
1062	2/03/16	102	2
1063	3/03/16	55	1
1064	3/03/16	63	1
1065	4/03/16	108	3
1066	6/03/16	99	3
1067	7/03/16	42	2
1068	7/03/16	57	2
1069	8/03/16	46	1
1070	9/03/16	106	3
1071	10/03/16	51	3
1072	11/03/16	65	2
1073	13/03/16	103	3
1074	13/03/16	13	3
1075	14/03/16	87	3
1076	15/03/16	15	3
1077	16/03/16	85	3
1078	17/03/16	24	2
1079	18/03/16	103	1
1080	20/03/16	67	1
1081	20/03/16	77	3
1082	21/03/16	27	1
1083	22/03/16	47	3
1084	23/03/16	76	1
1085	24/03/16	30	1
1086	25/03/16	76	2

1087	27/03/16	22	1
1088	28/03/16	87	1
1089	29/03/16	27	1
1090	30/03/16	54	3

Table 4.1d: Order for month 4 from customers

Order ID	**Order TIME**	**Order SIZE (kg)**	**Finishing Type**
1091	1/04/16	26	3
1092	3/04/16	57	1
1093	3/04/16	85	3
1094	4/04/16	29	1
1095	5/04/16	49	3
1096	6/04/16	104	3
1097	7/04/16	29	3
1098	8/04/16	22	2
1099	8/04/16	46	3
1100	10/04/16	52	3
1101	11/04/16	70	3
1102	12/04/16	101	3
1103	13/04/16	59	2
1104	14/04/16	98	2
1105	15/04/16	77	1
1106	17/04/16	70	3
1107	17/04/16	59	3
1108	18/04/16	74	1
1109	18/04/16	32	1
1110	19/04/16	67	1
1111	20/04/16	59	1
1112	21/04/16	75	2
1113	22/04/16	22	1
1114	22/04/16	88	3
1115	24/04/16	98	3
1116	25/04/16	20	1
1117	26/04/16	71	2
1118	27/04/16	51	2
1119	28/04/16	30	3
1120	29/04/16	55	1

Table 4.1e: Order for month 5 from customers

Order ID	**Order TIME**	**Order SIZE (kg)**	**Finishing Type**
1121	1/05/16	41	1
1122	2/05/16	107	1
1123	3/05/16	86	3
1124	4/05/16	39	1
1125	5/05/16	19	2
1126	6/05/16	30	2

1127	8/05/16	16	1
1128	8/05/16	90	1
1129	9/05/16	38	1
1130	10/05/16	68	3
1131	11/05/16	64	3
1132	12/05/16	82	3
1133	13/05/16	88	3
1134	15/05/16	107	3
1135	16/05/16	74	1
1136	17/05/16	38	3
1137	17/05/16	47	3
1138	18/05/16	18	1
1139	18/05/16	75	1
1140	19/05/16	36	1
1141	20/05/16	83	1
1142	22/05/16	85	3
1143	23/05/16	38	1
1144	23/05/16	17	1
1145	24/05/16	48	1
1146	25/05/16	15	1
1147	26/05/16	72	2
1148	27/05/16	24	1
1149	29/05/16	86	3
1150	30/05/16	86	3

Table 4.1f: Order for month 6 from customers

Order ID	Order TIME	Order SIZE (kg)	Finishing Type
1151	1/06/16	23	3
1152	2/06/16	105	2
1153	3/06/16	89	1
1154	3/06/16	96	2
1155	5/06/16	46	2
1156	6/06/16	13	1
1157	7/06/16	53	3
1158	7/06/16	27	3
1159	8/06/16	22	3
1160	9/06/16	30	3
1161	10/06/16	82	3
1162	12/06/16	52	3
1163	13/06/16	102	3
1164	14/06/16	38	2
1165	15/06/16	87	3
1166	16/06/16	98	3
1167	16/06/16	51	3
1168	17/06/16	37	3
1169	19/06/16	52	1
1170	19/06/16	62	3
1171	20/06/16	41	1

1172	21/06/16	47	3
1173	22/06/16	98	3
1174	23/06/16	63	2
1175	24/06/16	101	3
1176	26/06/16	20	3
1177	27/06/16	99	2
1178	28/06/16	12	1
1179	29/06/16	110	2
1180	30/06/16	73	2

Table 4.1g: Order for month 7 from customers

Order ID	Order TIME	Order SIZE (kg)	Finishing Type
1181	1/07/16	78	2
1182	3/07/16	18	2
1183	4/07/16	95	2
1184	5/07/16	88	2
1185	6/07/16	81	3
1186	6/07/16	37	1
1187	7/07/16	98	1
1188	8/07/16	52	1
1189	8/07/16	84	2
1190	10/07/16	107	2
1191	11/07/16	60	2
1192	12/07/16	50	1
1193	13/07/16	102	1
1194	14/07/16	89	1
1195	15/07/16	47	2
1196	17/07/16	80	2
1197	17/07/16	14	3
1198	18/07/16	76	3
1199	18/07/16	38	2
1200	19/07/16	92	1
1201	20/07/16	59	2
1202	21/07/16	102	1
1203	22/07/16	79	1
1204	22/07/16	35	3
1205	24/07/16	24	3
1206	25/07/16	41	3
1207	26/07/16	79	2
1208	27/07/16	88	1
1209	28/07/16	93	3
1210	29/07/16	55	3

Table 4.1h: Order for month 8 from customers

Order ID	Order TIME	Order SIZE (kg)	Finishing Type
1211	1/08/16	74	2

1212	2/08/16	87	3
1213	3/08/16	110	3
1214	4/08/16	100	2
1215	5/08/16	57	1
1216	5/08/16	94	3
1217	7/08/16	37	3
1218	8/08/16	95	3
1219	9/08/16	94	3
1220	10/08/16	106	3
1221	11/08/16	39	2
1222	12/08/16	72	3
1223	12/08/16	99	1
1224	14/08/16	76	3
1225	15/08/16	81	3
1226	16/08/16	94	1
1227	17/08/16	64	2
1228	18/08/16	85	3
1229	18/08/16	24	1
1230	19/08/16	27	1
1231	21/08/16	66	3
1232	22/08/16	91	1
1233	23/08/16	99	3
1234	23/08/16	66	3
1235	24/08/16	55	1
1236	25/08/16	46	3
1237	26/08/16	59	3
1238	28/08/16	88	3
1239	29/08/16	82	3
1240	30/08/16	39	1

Table 4.1i: Order for month 9 from customers

Order ID	**Order TIME**	**Order SIZE (kg)**	**Finishing Type**
1241	1/09/16	63	3
1242	2/09/16	11	2
1243	4/09/16	76	3
1244	5/09/16	62	1
1245	6/09/16	110	3
1246	6/09/16	68	2
1247	7/09/16	106	1
1248	8/09/16	96	2
1249	9/09/16	69	1
1250	9/09/16	59	1
1251	11/09/16	31	1
1252	12/09/16	25	2
1253	13/09/16	41	1
1254	15/09/16	32	1
1255	16/09/16	35	3

1256	16/09/16	52	1
1257	18/09/16	81	3
1258	19/09/16	38	2
1259	20/09/16	54	3
1260	20/09/16	49	2
1261	21/09/16	82	2
1262	22/09/16	61	2
1263	23/09/16	30	1
1264	23/09/16	42	1
1265	25/09/16	65	3
1266	26/09/16	68	3
1267	27/09/16	12	3
1268	28/09/16	31	3
1269	29/09/16	41	3
1270	30/09/16	56	1

Table 4.1j: Order for month 10 from customers

Order ID	**Order TIME**	**Order SIZE (kg)**	**Finishing Type**
1271	2/10/16	46	3
1272	3/10/16	70	1
1273	4/10/16	43	3
1274	5/10/16	51	1
1275	6/10/16	35	3
1276	6/10/16	39	3
1277	7/10/16	83	1
1278	9/10/16	13	1
1279	9/10/16	36	3
1280	10/10/16	46	1
1281	11/10/16	99	2
1282	12/10/16	22	3
1283	13/10/16	43	3
1284	14/10/16	19	1
1285	16/10/16	39	2
1286	17/10/16	73	3
1287	17/10/16	22	2
1288	18/10/16	19	1
1289	18/10/16	68	1
1290	19/10/16	16	3
1291	20/10/16	102	3
1292	21/10/16	58	3
1293	23/10/16	70	1
1294	23/10/16	108	1
1295	24/10/16	109	2
1296	25/10/16	53	3
1297	26/10/16	110	3
1298	27/10/16	46	1
1299	28/10/16	68	2
1300	30/10/16	34	2

The order, as it came from the customer for processing, is shown in table 4.1(a-j) above with different order sizes and type of finish. The agent model proposed,sorts the order according to level and finishing type. Table 4.2(a-j) shows the order grouped in ascending order (order size) with respect to their finishing type. An order has a level depending upon its size. Orders of size 1kg up to size 45kg are deemed as level one (1) job; orders of size 46kg up to size 75kg are level two (2) jobs, while orders of size 76kg and above are level three (3) jobs. The grouping was done to reduce long queue and for customer satisfaction. The smaller orders with small lead time to finish should be processed first before large orders with larger lead time. This gives higher customer satisfaction as the lead time for every job is met with this approach.

Table 4.2a: Sorted Order 1 for ScheduleAccording to Size with Respect to the Finishing Type

Order ID	Order Time	Order Size (kg)	Order Level	Finishing Type
1016	17/01/16	16	1	1
1025	24/01/16	13	1	2
1029	29/01/16	13	1	2
1006	6/01/16	13	1	3
1011	11/01/16	36	1	3
1030	30/01/16	27	1	1
1026	25/01/16	20	1	2
1012	12/01/16	22	1	2
1015	16/01/16	38	1	3
1023	23/01/16	32	1	1
1024	23/01/16	29	1	2
1013	13/01/16	40	1	3
1007	8/01/16	50	2	3
1022	22/01/16	43	1	1

1017	17/01/16	51	2	2
1014	15/01/16	54	2	3
1003	4/01/16	59	2	3
1028	27/01/16	51	2	1
1018	18/01/16	62	2	2
1008	8/01/16	82	3	3
1010	10/01/16	60	2	1
1009	9/01/16	83	3	3
1027	26/01/16	104	3	2
1019	18/01/16	85	3	3
1002	3/01/16	70	2	1
1005	6/01/16	88	3	3
1001	2/01/16	102	3	3
1020	19/01/16	96	3	1
1004	5/01/16	107	3	3
1021	20/01/16	101	3	1

Table 4.2b: Sorted Order 2 for Schedule According to Size with Respect to the Finishing Type

Order ID	**Order Time**	**Order Size (kg)**	**Order Level**	**Finishing Type**
1054	23/02/16	22	1	1
1050	18/02/16	30	1	2
1047	16/02/16	47	2	3
1037	6/02/16	23	1	1
1038	6/02/16	30	1	2
1049	17/02/16	62	2	3
1040	8/02/16	23	1	1
1043	11/02/16	35	1	2
1057	25/02/16	25	1	1
1031	1/02/16	38	1	2
1060	28/02/16	72	2	3
1045	14/02/16	34	1	1
1033	2/02/16	55	2	2
1053	22/02/16	79	3	3
1041	9/02/16	39	1	1
1034	3/02/16	87	3	3
1048	16/02/16	60	2	2
1036	4/02/16	45	1	1
1052	21/02/16	98	3	3
1055	24/02/16	46	2	1
1042	11/02/16	63	2	2
1035	4/02/16	109	3	3
1056	24/02/16	51	2	1
1032	2/02/16	78	3	2
1039	7/02/16	56	2	1

1044	13/02/16	101	3	2
1051	20/02/16	71	2	1
1059	27/02/16	104	3	2
1046	15/02/16	75	2	1
1058	26/02/16	83	3	1

Table 4.2c: Sorted Order 3 for Schedule According to Size with Respect to the Finishing Type

Order ID	Order Time	Order Size (kg)	Order Level	Finishing Type
1087	27/03/16	22	1	1
1078	17/03/16	24	1	2
1074	13/03/16	13	1	3
1076	15/03/16	15	1	3
1089	29/03/16	27	1	1
1067	7/03/16	42	1	2
1083	22/03/16	47	2	3
1082	21/03/16	27	1	1
1071	10/03/16	51	2	3
1090	30/03/16	54	2	3
1085	24/03/16	30	1	1
1068	7/03/16	57	2	2
1072	11/03/16	65	2	2
1081	20/03/16	77	2	3
1069	8/03/16	46	2	1
1077	16/03/16	85	3	3
1063	3/03/16	55	2	1
1086	25/03/16	76	3	2
1075	14/03/16	87	3	3
1064	3/03/16	63	2	1
1066	6/03/16	99	3	3
1062	2/03/16	102	3	2
1079	18/03/16	103	3	3
1080	20/03/16	67	2	1
1070	9/03/16	106	3	3
1061	1/03/16	69	2	1
1065	4/03/16	108	3	3
1084	23/03/16	76	3	1
1088	28/03/16	87	3	1
1073	13/03/16	103	3	1

Table 4.2d: Sorted Order 4 for Schedule According to Size with Respect to the Finishing Type

Order ID	Order Time	Order Size	Order Level	Finishing Type

		(kg)		
1116	25/04/16	20	1	1
1098	8/04/16	22	1	2
1091	1/04/16	26	1	3
1097	7/04/16	26	1	3
1113	22/04/16	22	1	1
1118	27/04/16	51	2	2
1119	28/04/16	30	1	3
1094	4/04/16	29	1	1
1099	8/04/16	46	2	3
1095	5/04/16	49	2	3
1109	18/04/16	32	1	1
1103	13/04/16	59	2	2
1100	10/04/16	52	2	3
1107	17/04/16	59	2	3
1120	29/04/16	55	2	1
1117	26/04/16	71	2	2
1101	11/04/16	70	2	3
1092	3/04/16	57	2	1
1112	21/04/16	75	2	2
1106	17/04/16	70	2	3
1093	3/04/16	85	3	3
1111	20/04/16	59	2	1
1104	14/04/16	98	3	2
1114	22/04/16	88	3	3
1110	19/04/16	67	2	1
1115	24/04/16	98	3	3
1108	18/04/16	74	2	1
1102	12/04/16	101	3	3
1096	6/04/16	104	3	3
1105	15/04/16	77	3	1

Table 4.2e: Sorted Order 5 for Schedule According to Size with Respect to the Finishing Type

Order ID	Order Time	Order Size (kg)	Order Level	Finishing Type
1146	25/05/16	15	1	1
1127	8/05/16	16	1	1
1125	5/05/16	19	1	2
1126	6/05/16	30	1	2
1136	17/05/16	38	1	3
1137	17/05/16	47	2	3
1144	23/05/16	17	1	1
1138	18/05/16	18	1	1
1148	27/05/16	24	1	1
1131	11/05/16	64	2	3

1140	19/05/16	36	1	1
1130	10/05/16	68	2	3
1143	23/05/16	38	2	1
1132	12/05/16	82	3	3
1129	9/05/16	38	1	1
1142	22/05/16	85	3	3
1124	4/05/16	39	1	1
1123	3/05/16	86	3	3
1121	1/05/16	41	1	1
1149	29/05/16	86	3	3
1145	24/05/16	48	2	1
1150	30/05/16	86	3	3
1147	26/05/16	72	2	2
1133	13/05/16	88	3	3
1134	15/05/16	107	3	3
1135	16/05/16	74	2	1
1139	18/05/16	75	3	1
1141	20/05/16	83	3	1
1128	8/05/16	90	3	1
1122	2/05/16	107	3	1

Table 4.2f: Sorted Order 6 for Schedule According to Size with Respect to the Finishing Type

Order ID	Order Time	Order Size (kg)	Order Level	Finishing Type
1178	28/06/16	12	1	1
1156	6/06/16	13	1	1
1164	14/06/16	38	1	2
1176	26/06/16	20	1	3
1159	8/06/16	22	1	3
1151	1/06/16	23	1	3
1158	7/06/16	27	1	3
1171	20/06/16	41	1	1
1155	5/06/16	46	2	2
1160	9/06/16	30	1	3
1168	17/06/16	37	1	3
1169	19/06/16	52	2	1
1174	23/06/16	63	2	2
1172	21/06/16	47	2	3
1167	16/06/16	51	2	3
1180	30/06/16	73	2	2
1162	12/06/16	52	2	3
1153	3/06/16	89	3	1
1157	7/06/16	53	2	3
1170	19/06/16	62	2	3
1154	3/06/16	96	3	2

1161	10/06/16	82	3	3
1165	15/06/16	87	3	3
1177	27/06/16	99	3	2
1173	22/06/16	98	3	3
1152	2/06/16	105	3	2
1166	16/06/16	98	3	3
1175	24/06/16	101	3	3
1179	29/06/16	110	3	2
1163	13/06/16	102	3	3

Table 4.2g: Sorted Order 7 for Schedule According to Size with Respect to the Finishing Type

Order ID	Order Time	Order Size (kg)	Order Level	Finishing Type
1186	6/07/16	37	1	1
1182	3/07/16	18	1	2
1199	18/07/16	38	1	2
1197	17/07/16	14	1	3
1205	24/07/16	24	1	3
1204	22/07/16	35	1	3
1192	12/07/16	50	2	1
1195	15/07/16	47	2	2
1206	25/07/16	41	1	3
1210	29/07/16	55	2	3
1201	20/07/16	59	2	2
1198	18/07/16	76	3	3
1188	8/07/16	52	2	1
1191	11/07/16	60	2	2
1185	6/07/16	81	3	3
1203	22/07/16	79	3	1
1209	28/07/16	93	3	3
1181	1/07/16	78	3	2
1208	27/07/16	88	3	1
1207	26/07/16	79	3	2
1194	14/07/16	89	3	1
1196	17/07/16	80	3	2
1189	8/07/16	84	3	2
1200	19/07/16	92	3	1
1184	5/07/16	88	3	2
1187	7/07/16	98	3	1
1183	4/07/16	95	3	2
1190	10/07/16	107	3	2
1193	13/07/16	102	3	1
1202	21/07/16	102	3	1

Table 4.2h: Sorted Order 8 for Schedule According to Size with Respect to the Finishing Type

Order ID	Order Time	Order Size (kg)	Order Level	Finishing Type
1229	18/08/16	24	1	1
1217	7/08/16	37	1	3
1230	19/08/16	27	1	1
1221	11/08/16	39	1	2
1236	25/08/16	46	2	3
1240	30/08/16	39	1	1
1227	17/08/16	64	2	2
1237	26/08/16	59	2	3
1234	23/08/16	66	2	3
1235	24/08/16	55	2	1
1211	1/08/16	74	2	2
1231	21/08/16	66	2	3
1222	12/08/16	72	2	3
1215	5/08/16	57	2	1
1214	4/08/16	100	3	2
1224	14/08/16	76	3	3
1225	15/08/16	81	3	3
1232	22/08/16	91	3	1
1239	29/08/16	82	3	3
1228	18/08/16	85	3	3
1226	16/08/16	94	3	1
1212	2/08/16	87	3	3
1238	28/08/16	88	3	3
1223	12/08/16	99	3	1
1216	5/08/16	94	3	3
1219	9/08/16	94	3	3
1218	8/08/16	95	3	3
1233	23/08/16	99	3	3
1220	10/08/16	107	3	3
1213	3/08/16	110	3	3

Table 4.2i: Sorted Order 9 for Schedule According to Size with Respect to the Finishing Type

Order ID	Order Time	Order Size (kg)	Order Level	Finishing Type
1263	23/09/16	30	1	1

1242	2/09/16	11	1	2
1252	12/09/16	25	1	2
1267	27/09/16	12	1	3
1268	28/09/16	31	1	3
1258	19/09/16	38	1	2
1255	16/09/16	35	1	3
1269	29/09/16	41	1	3
1251	11/09/16	31	1	1
1259	20/09/16	54	2	3
1254	15/09/16	32	1	1
1260	20/09/16	49	2	2
1241	1/09/16	63	2	3
1253	13/09/16	41	1	1
1262	22/09/16	61	2	2
1264	23/09/16	42	1	1
1265	25/09/16	65	2	3
1246	6/09/16	68	2	2
1266	26/09/16	68	2	3
1256	16/09/16	52	2	1
1243	4/09/16	76	3	3
1261	21/09/16	82	3	2
1257	18/09/16	81	3	3
1270	30/09/16	56	2	1
1248	8/09/16	96	3	2
1245	6/09/16	110	3	3
1250	9/09/16	59	2	1
1244	5/09/16	62	2	1
1249	9/09/16	69	2	1
1247	7/09/16	106	3	1

Table 4.2j: Sorted Order 10 for Schedule According to Size with Respect to the Finishing Type

Order ID	**Order Time**	**Order Size (kg)**	**Order Level**	**Finishing Type**
1271	2/10/16	13	1	1
1287	17/10/16	22	1	2
1290	19/10/16	16	1	3
1282	12/10/16	22	1	3
1284	14/10/16	19	1	1
1288	18/10/16	19	1	1
1300	30/10/16	34	1	2
1275	6/10/16	35	1	3
1279	9/10/16	36	1	3
1285	16/10/16	39	1	2
1276	6/10/16	39	1	3
1280	10/10/16	46	2	1
1273	4/10/16	43	1	3

1283	13/10/16	43	1	3
1298	27/10/16	46	2	1
1299	28/10/16	68	2	2
1271	2/10/16	46	2	3
1296	25/10/16	53	2	3
1274	5/10/16	51	2	1
1281	11/10/16	99	3	2
1292	21/10/16	58	2	3
1286	17/10/16	73	3	3
1289	18/10/16	68	2	1
1295	24/10/16	109	3	2
1291	20/10/16	102	3	3
1272	3/10/16	70	2	1
1297	26/10/16	110	3	3
1293	23/10/16	70	2	1
1277	7/10/16	83	3	1
1294	23/10/16	108	3	1

4.2 Scheduling of Job Order Using Bunching Factors 1, 2 & 3

Bunching technique was adopted in this model to schedule job for processing. Bunching of the whole order queue with bunching factor (bf) of 1 or 2 or 3 to determine the best bunching that gives the earliest finishing time or minimum makespan for all the orders. Table 4.3 shows the schedule result for ten (10) different orders, scheduled using Bf1, Bf2 and Bf3.

Because of the stochastic nature of the order arrival, the best bunching factor may change with each order, for example, in an empirical study involving ten (10) different sets of orders (Table 4.3) the bunching factor of two (2) gave the best result in eight out of the ten (10) sets of orders while the bunching factor of three (3) gave the best result in two

(2) out of the ten (10) sets of orders. The bunching factor of one (1) is consistently the worst case scenario in all the ten (10) sets of order.

To make the matter clearer, consider order one (1) in Table 4.3, the table shows that all the orders that need finishing type one (1) will be finished in 100 days using bunching factor 1, but 98 days using bunching factor 2 and 102 days using bunching factor 3. Also all the orders that require finishing type 2 will be completed in 50 days using bunching factor 1, or 52 days using bunching factor 2 and 51 days using bunching factor 3. Similarly, all the orders requiring finishing type 3 will be completed in 84 days using bunching factor 2 but will take as much as 90 days if bunching factor 3 were used. In this scenario the best bunching factor is the one with the least number of days for completing the last job in a given order queue. Because the different finishing types in one set of orders do not have the same number of jobs, a finishing type may finish before others. For example, for order number 1 using bunching type (Bf1), finishing type 1 was the last to be processed up to the 100^{th} day, finishing type 2 finished on the 50^{th} day while finishing type 3 finished on the 84^{th} day. In order 1 therefore, a bunching factor of 2 that finished the work in an order queue in 98 days is superior to bunching factor 1 that finished the work in an order queue in 100 days. The bunching factor of 3 gave the worst case scenario for this order requiring 102 days to complete the order. Thus the best is bunching factor 2 as shown in table 4.3.

Consider another example from table 4.3 where bunching factor 3 is the best out of the three possible bunching factors. Consider order number 7, the latest finishing time to complete the order for bunching factor 1 is 157 days, that of bunching factor 2 is 152 days but the bunching factor 3 will get the work done in 147 days. Thus the bunching factor to use when scheduling order number 7 is bunching factor 3.

Table 4.3: Schedule result for Ten (10) Different order with Bf1, Bf2 & Bf3

ORDER	FINISHING TYPE	Bf_1 (days)	Bf_2 (days)	Bf_3 (days)	BEST
1	1	100	98	102	2
	2	50	52	51	
	3	84	84	90	
2	1	118	116	120	2
	2	95	94	96	
	3	57	60	63	
3	1	134	128	129	2
	2	59	58	60	
	3	87	90	90	
4	1	97	92	93	2
	2	59	58	60	
	3	93	90	99	
5	1	166	164	156	3
	2	8	10	6	
	3	84	84	90	
6	1	40	38	39	2
	2	101	100	105	
	3	102	102	108	
7	1	157	152	147	3
	2	131	130	132	
	3	42	42	45	
8	1	97	92	93	2
	2	44	46	42	
	3	147	144	153	
9	1	115	110	111	2

	2	68	70	69	
	3	66	66	72	
10	1	118	116	120	2
	2	59	58	60	
	3	69	72	72	

The result of 10 different set of orders (table 4.3) shows that bunching factor two (bf2) has the smallest finishing times for orders 1,2,3,4,6,8,9 and 10; bunching factor three (bf3) just had a better result in 5 and 7 while bunching factor one (bf1) had none. A careful application of this bunching technique will help save time and cost in everyindustry that receives stochastic order. The graphs of figure 4.1 to 4.10 shown below were used to illustrate the performances of the three bunching factors.

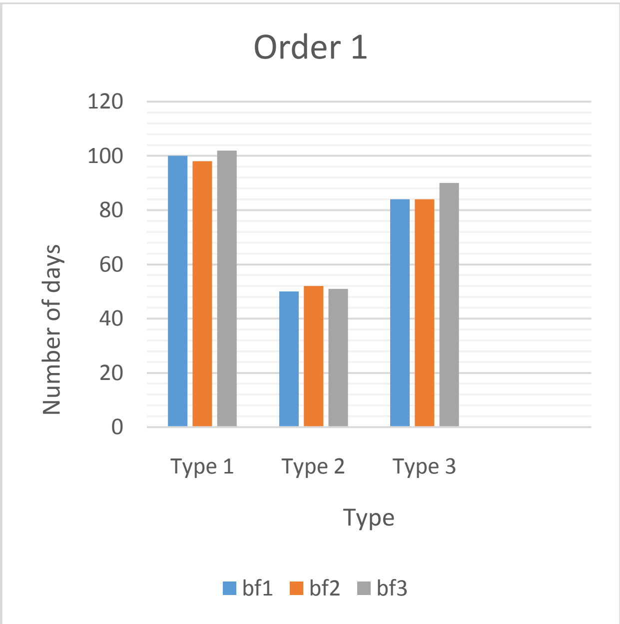


Fig 4.1 Bar chart of Order 1 as bf varies from 1-3

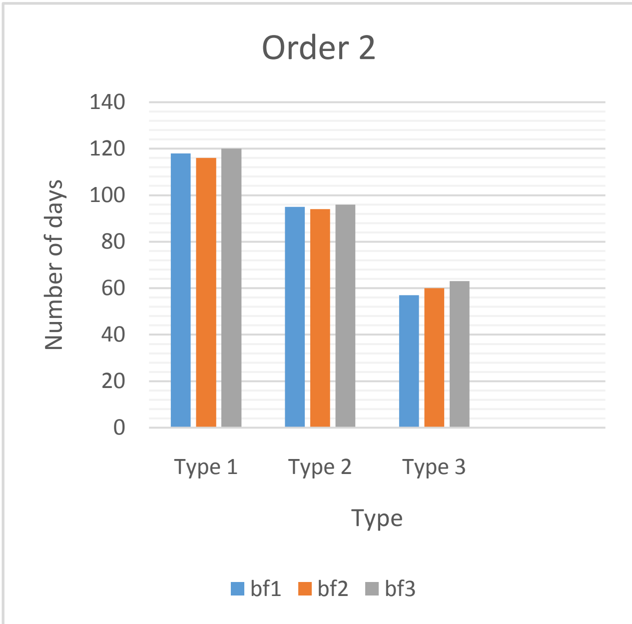


Figure 4.2 Bar chart of Order 2 as bf varies from 1-3

The release dates for orders received in one month and then scheduledis shown in figure 4.1. Referring to table 4.3,the orders are scheduled using three different bunching factors (bf1, bf2 and bf3) for the three finishing types. From the bar chart, the finishing type 1 has bf1 as 100 days, bf2 as98 days and bf3 as 102 days;finishing type 2 has bf1 as 50 days, bf2 as52 days and bf3 as 51 days, while finishing type 3 has bf1 as 84 days, bf2 as84 days and bf3 as 90 days. The result shows that bf2 had the earliest due date to complete the last operation, with the latest due date for the last release as 98 days while bf1 has 100 days and bf3 has 102 days. Similar thing happened in the second order of figure 4.2, with bf2 having earliest due date for the complete process as 116 days while bf1 uses 118 days and bf3 uses 120 days to complete the process.

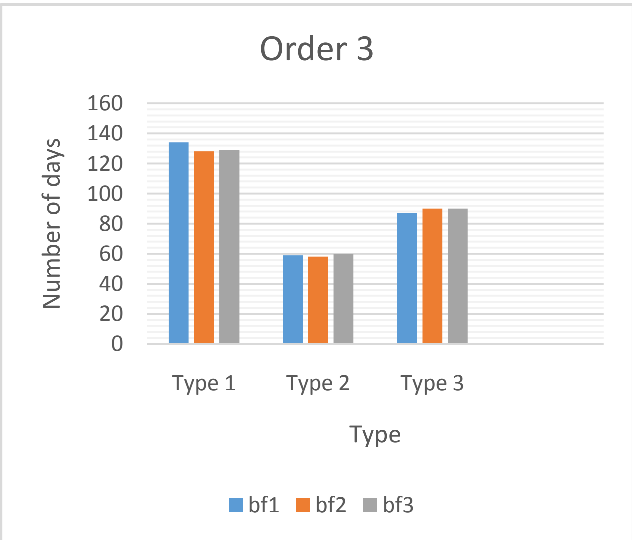


Figure 4.3 Bar chart of Order 3 as bf varies from 1-3

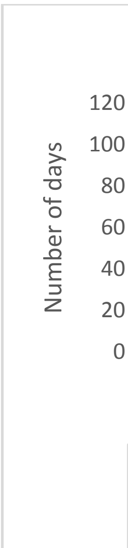
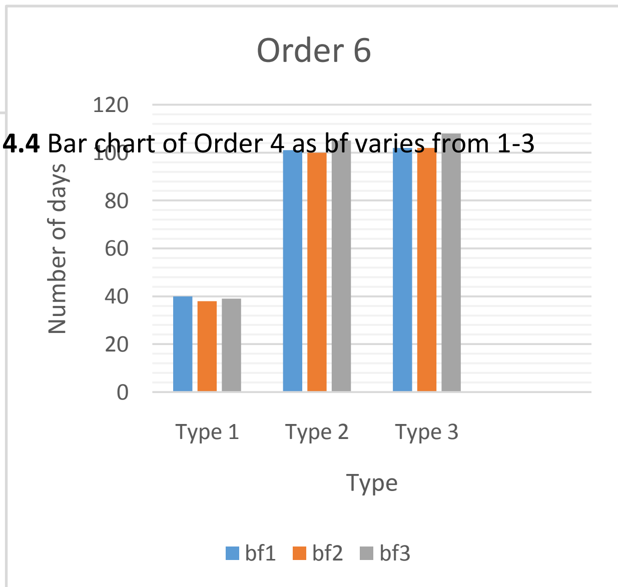


Figure 4.4 Bar chart of Order 4 as bf varies from 1-3



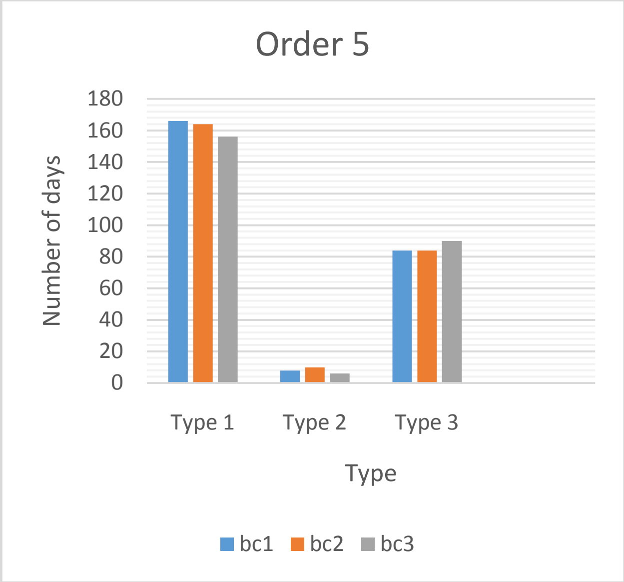
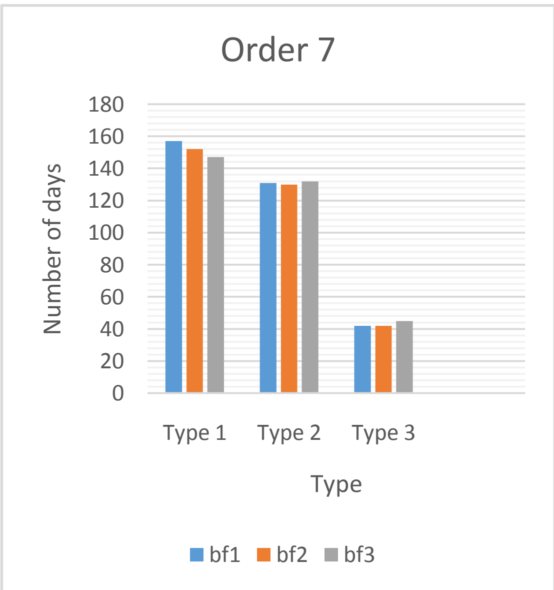


Figure 4.5 Bar chart of Order 5 as bf varies from 1-3 **Figure 4.6** Bar chart of Order 6 as bf varies from 1-3

The situation was the same for eight out of the ten different orders except order five where the latest release date for type 1 is bf1 =166, bf2=164 and bf3=156 as can be seen in figure 4.5. The result in this case shows that bunching factor 3 (Bf3) had earliest due date for the complete order.



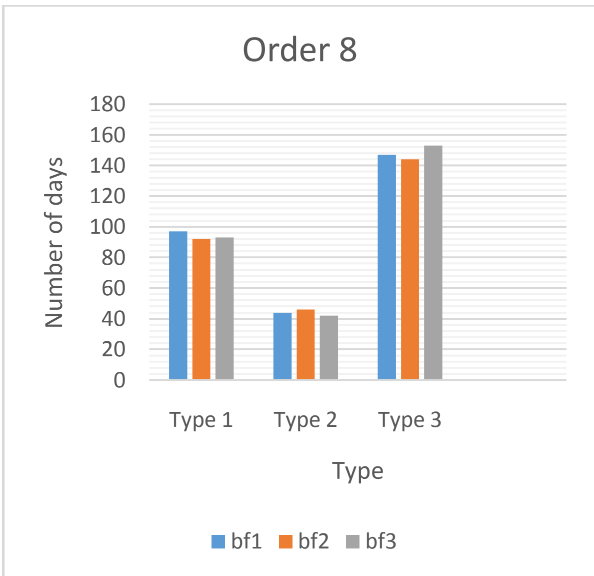
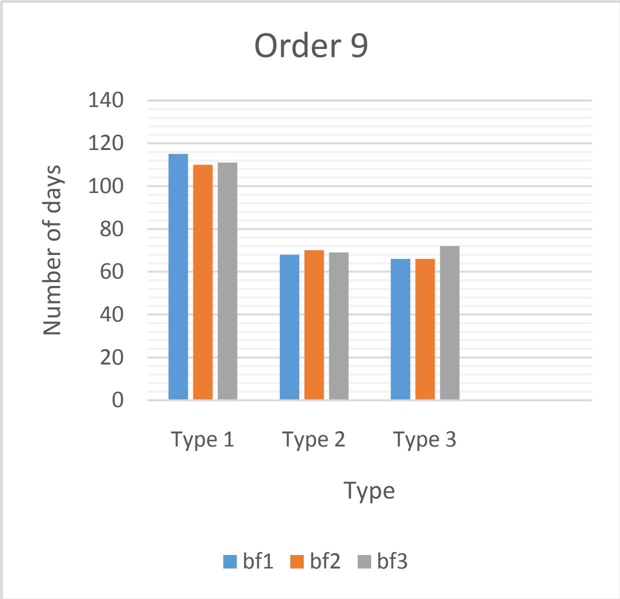


Figure 4.7 Bar chart of Order 7 as bf varies from 1-3**Figure 4.8** Bar chart of Order 8 as bf varies from 1-3



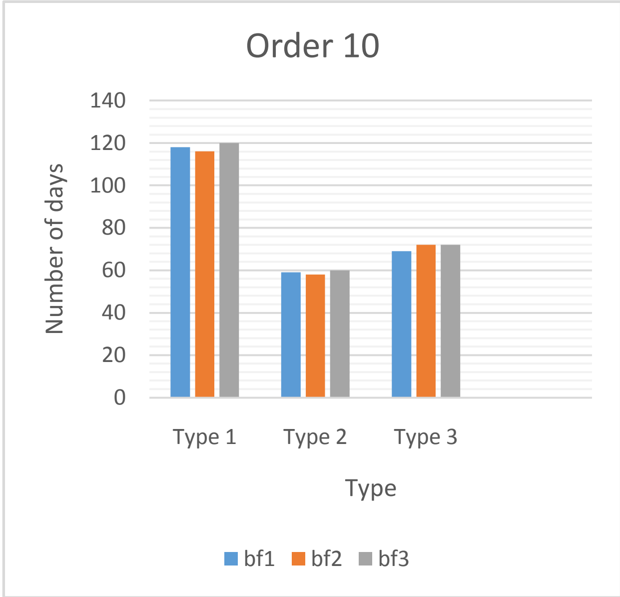


Figure 4.9 Bar chart of Order 9 as bf varies from 1-3.**Figure 4.10** Bar chart of Order 10 as bf varies from 1-3

The results obtained with this scheduling technique shows that, bunching factor 2 gives better result in eight different order while bunching factor 3 is better in just two order while bunching factor 1gave poor schedule result in all.

4.3 Results of Ten Different Orders Scheduled with Best Bunching Factor

The adoption of a particular bunching factor is dependent on the earliest finishing time for all orders, therefore after simulating the order with different bunching factors, the best bunching factor was selected for scheduling of the batch of ordersand the result for ten different batch of orders for ten months is presented in table 4.4(a-j).

Table 4.4a: Schedule result using the best bunching factor for Month 1

S/NO	ORDER NO	BEST BUNCHING FACTOR	FINISHING TYPE	ORDER SIZE (KG)	RELEASE DAYS	EARLIEST EVENT DATE	LATEST EVENT DATE
1	1	2	1	16	2	2nd Feb	4th Feb
2	1	2	2	13	3	3rd Feb	6th Feb
3	1	2	2	13	4	4th Feb	7th Feb
4	1	2	3	13	5	6th Feb	8th Feb
5	1	2	3	36	6	7th Feb	9th Feb
6	1	2	1	27	7	8th Feb.	10th Feb
7	1	2	2	20	8	9th Feb	11th Feb
8	1	2	2	22	9	10th Feb	13th Feb
9	1	2	3	38	10	11th Feb	14th Feb
10	1	2	1	32	12	14th Feb	16th Feb
11	1	2	2	29	14	16th Feb	18th Feb
12	1	2	3	40	16	18nd Feb	21st Feb
13	1	2	3	50	17	20th Feb	22nd Feb
14	1	2	1	43	20	23rd Feb	25thFeb
15	1	2	2	51	22	25th Feb	28th Feb
16	1	2	3	54	24	28th Feb	2nd Mar
17	1	2	3	59	26	2nd Mar	4th Mar
18	1	2	1	51	30	7th Mar	9th Mar
19	1	2	2	62	34	11th Mar	14th Mar
20	1	2	3	82	37	15th Mar	17th Mar
21	1	2	1	60	41	20th Mar	22nd Mar
22	1	2	3	83	44	23rd Mar	25th Mar
23	1	2	2	104	49	29th Mar	31st Mar
24	1	2	3	85	51	31st Mar	3rd April
25	1	2	1	70	55	5th April	7th April
26	1	2	3	88	58	8th April	11th April
27	1	2	3	102	62	13th April	15th April
28	1	2	1	96	69	21th April	24th April
29	1	2	3	107	72	25th April	27th April
30	1	2	1	101	79	3rd May	5th May

Table 4.4b: Schedule result using the best bunching factor for Month 2

S/NO	ORDER NO	BEST BUNCHING FACTOR	FINISHING TYPE	ORDER SIZE (KG)	RELEASE DATE	EARLIEST EVENT DATE	LATEST EVENT DATE
1	2	2	1	22	1	4 th May	6th May
2	2	2	2	30	3	6thMay	9thMay
3	2	2	3	47	5	9thMay	11thMay
4	2	2	1	23	7	11thMay	13thMay
5	2	2	2	30	8	12thMay	14thMay

6	2	2	3	62	10	14tt May	17thMay
7	2	2	1	23	11	16thMay	18thMay
8	2	2	2	35	13	18thMay	20thMay
9	2	2	1	25	15	20stMay	23rdMay
10	2	2	2	38	17	23rdMay	25thMay
11	2	2	3	72	19	25thMay	27thMay
12	2	2	1	34	21	27thMay	30thMay
13	2	2	2	55	24	31st May	2nd Jun
14	2	2	3	79	27	3rd Jun	6thJun
15	2	2	1	39	30	7th Jun	9thJun
16	2	2	3	87	33	10thJun	13thJun
17	2	2	2	60	36	14thJun	16thJun
18	2	2	1	45	39	17thJun	20thJun
19	2	2	3	98	42	21stJun	23rdJun
20	2	2	1	46	45	24thJun	27thJun
21	2	2	2	63	48	28thJun	30thJun
22	2	2	3	109	52	2ndJul	5thJul
23	2	2	1	51	55	6thJul	8th Jul
24	2	2	2	78	59	11thJul	13thJul
25	2	2	1	56	63	15rdJul	18thJul
26	2	2	2	101	69	22ndJul	25thJul
27	2	2	1	71	73	27thJul	29thJul
28	2	2	2	104	78	2ndAug	4thAug
29	2	2	1	75	83	9thAug	1thAug
30	2	2	1	83	89	16th Aug	19th Aug

Table 4.4c: Schedule result using the best bunching factor for Month 3

S/NO	ORDER NO	BEST BUNCHING FACTOR	FINISHING TYPE	ORDER SIZE (KG)	RELEASE DATE	EARLIEST EVENT DATE	LATEST EVENT DATE
1	3	2	1	22	1	17th Aug	19th Aug
2	3	2	2	24	3	19thAug	22ndAug
3	3	2	3	13	3	20thAug	23rdAug
4	3	2	3	15	4	20thAug	23rdAug
5	3	2	1	27	6	24thAug	26thAug
6	3	2	2	42	8	26thAug	29thAug
7	3	2	3	47	9	27thAug	30thAug
8	3	2	1	27	11	31st Aug	2nd Sep
9	3	2	3	51	13	3rdSep	6thSep
10	3	2	3	54	15	5thSep	7thSep
11	3	2	1	30	17	7thSep	9thSep
12	3	2	2	57	20	8thSep	10thSep
13	3	2	2	65	23	16thSep	19thSep
14	3	2	3	77	25	17thSep	20thSep
15	3	2	1	46	28	20thSep	22ndSep
16	3	2	3	85	31	26thSep	28thSep

17	3	2	1	55	35	28thSep	30thSep
18	3	2	2	76	39	30thSep	3rd Oct
19	3	2	3	87	42	8th Oct	11th Oct
20	3	2	1	63	46	12th Oct	14th Oct
21	3	2	3	99	49	17th Oct	19th Oct
22	3	2	2	102	55	21st Oct	24th Oct
23	3	2	3	103	59	5thNov	8thNov
24	3	2	1	67	63	7thNov	9thNov
25	3	2	3	106	66	18thNov	21stNov
26	3	2	1	69	71	22ndNov	24thNov
27	3	2	3	108	75	1stDec	4thDec
28	3	2	1	76	80	11thDec	13thDec
29	3	2	1	87	86	27thDec	29thDec
30	3	2	1	103	93	19th Jan	21st Jan

Table 4.4d: Schedule result using the best bunching factor for Month 4

S/NO	ORDER NO	BEST BUNCHING FACTOR	FINISHING TYPE	ORDER SIZE (KG)	RELEASE DATE	EARLIEST EVENT DATE	LATEST EVENT DATE
1	4	2	1	20	1	21stJan	23rdJan
2	4	2	2	22	2	23rdJan	26thJan
3	4	2	3	26	3	25thJan	27thJan
4	4	2	3	29	4	26thJan	28thMar
5	4	2	1	22	6	27thJan	29thJan
6	4	2	2	51	8	30thJan	2nd Feb
7	4	2	3	30	9	1st Feb	3rd Feb
8	4	2	1	29	11	3rd Feb	5th Feb
9	4	2	3	46	12	8th Feb	10th Feb
10	4	2	3	49	14	9th Feb	11th Feb
11	4	2	1	32	16	10th Feb	12th Feb
12	4	2	2	59	19	12th Feb	15th Feb
13	4	2	3	52	21	16th Feb	18th Feb
14	4	2	3	59	23	23rd Feb	25th Feb
15	4	2	1	55	26	24th Feb	26th Feb
16	4	2	2	71	30	26th Feb	28th Feb
17	4	2	3	70	32	6thMar	8thMar
18	4	2	1	57	36	8thMar	10thMar
19	4	2	2	75	40	10thMar	13thMar
20	4	2	3	70	42	13thMar	15thMar
21	4	2	3	85	45	21stMar	23rdMar
22	4	2	1	59	49	22ndMar	24thMar

23	4	2	2	98	54	25thMar	28thMar
24	4	2	3	88	57	3rd April	5thApril
25	4	2	1	67	62	5thApril	7thApril
26	4	2	3	98	65	11thApril	13thApril
27	4	2	1	74	69	20thApril	22ndApril
28	4	2	3	101	73	24thApril	26thApril
29	4	2	3	104	76	2nd May	4thMay
30	4	2	1	77	81	4thMay	6thMay

Table 4.4e: Schedule result using the best bunching factor for Month 5

S/NO	ORDER NO	BEST BUNCHING FACTOR	FINISHING TYPE	ORDER SIZE (KG)	RELEASE DATE	EARLIEST EVENT DATE	LATEST EVENT DATE
1	5	3	1	15	1	5thMay	8thMay
2	5	3	1	16	2	6thMay	9thMay
3	5	3	2	19	3	9thMay	11thMay
4	5	3	2	30	5	11thMay	13thMay
5	5	3	3	38	7	13thMay	16thMay
6	5	3	3	47	8	15thMay	17thMay
7	5	3	1	17	10	16thMay	18thMay
8	5	3	1	18	11	17th May	19thMay
9	5	3	1	24	12	18thMay	20thMay
10	5	3	3	64	14	24thMay	26thMay
11	5	3	1	36	17	29thMay	31stMay
12	5	3	3	68	19	3rd Jun	6th Jun
13	5	3	1	38	21	9thJun	12thJun
14	5	3	3	82	24	15thJun	17thJun
15	5	3	1	38	27	20thJun	22ndJun
16	5	3	3	85	30	26thJun	28thJun
17	5	3	1	39	32	26thJun	2nd Jul
18	5	3	3	86	35	30thJun	7th Jul
19	5	3	1	41	38	5th Jul	11th Jul
20	5	3	3	86	41	9th Jul	18th Jul
21	5	3	1	48	44	16th Jul	21st Jul
22	5	3	3	86	47	19th Jul	28th Jul
23	5	3	1	72	52	26th Jul	5th Aug
24	5	3	3	88	55	3rd Aug	10th Aug
25	5	3	3	107	58	8th Aug	21st Aug
26	5	3	1	74	63	18th Aug	25th Aug
27	5	3	1	75	68	12th Aug	14th Aug
28	5	3	1	75	73	19th Aug	22nd Aug
29	5	3	1	90	79	25th Aug	28th Aug
30	5	3	1	107	86	9th Sep	11th Sep

Table 4.4f: Schedule result using the best bunching factor for Month 6

S/NO	ORDER NO	BEST BUNCHING FACTOR	FINISHING TYPE	ORDER SIZE (KG)	RELEASE DATE	EARLIEST EVENT DATE	LATEST EVENT DATE
1	6	2	1	12	1	10th Sep	13th Sep
2	6	2	1	13	2	11th Sep	14th Sep
3	6	2	2	38	4	14th Sep	16th Sep
4	6	2	3	20	5	15th Sep	17th Sep
5	6	2	3	22	6	16th Sep	18th Sep
6	6	2	3	23	6	22nd Sep	24th Sep
7	6	2	3	27	7	23rd Sep	25th Sep
8	6	2	1	41	10	24th Sep	27th Sep
9	6	2	2	46	12	27th Sep	29th Sep
10	6	2	3	30	13	29th Sep	31st Sep
11	6	2	3	37	14	30th Sep	2nd Oct
12	6	2	1	52	17	2nd Oct	4th Oct
13	6	2	2	63	21	4th Oct	6th Oct
14	6	2	3	47	23	5th Oct	7th Oct
15	6	2	3	51	25	12th Oct	14th Oct
16	6	2	2	73	28	18th Oct	20th Oct
17	6	2	3	52	29	19th Oct	21st Oct
18	6	2	1	89	35	22nd Oct	25th Oct
19	6	2	3	53	37	26th Oct	28th Oct
20	6	2	3	62	39	4th Nov	6th Nov
21	6	2	2	96	45	9th Nov	11th Nov
22	6	2	3	82	48	11th Nov	13th Nov
23	6	2	3	87	51	19th Nov	21st Nov
24	6	2	2	99	56	24th Nov	26th Nov
25	6	2	3	98	59	2nd Dec	4th Dec
26	6	2	2	105	64	14th Dec	16th Dec
27	6	2	3	98	67	15th Dec	17th Dec
28	6	2	3	101	71	23rd Dec	27th Dec
29	6	2	2	110	77	5th Jan	7th Jan
30	6	2	3	102	80	7th Jan	9th Jan

Table 4.4g: Schedule result using the best bunching factor for Month 7

S/NO	ORDER NO	BEST BUNCHING FACTOR	FINISHING TYPE	ORDER SIZE (KG)	RELEASE DATE	EARLIEST EVENT DATE	LATEST EVENT DATE
1	7	3	1	37	2	11th Jan	13th Jan
2	7	3	2	18	3	12th Jan	14th Jan
3	7	3	2	38	5	14th Jan	16th Jan
4	7	3	3	14	6	15th Jan	18th Jan
5	7	3	3	24	6	16th Jan	19th Jan

6	7	3	3	35	8	19th Jan	21st Jan
7	7	3	1	50	12	22nd Jan	25th Jan
8	7	3	2	47	14	26th Jan	28th Jan
9	7	3	3	41	15	27th Jan	29th Jan
10	7	3	3	55	17	29th Jan	1st Mar
11	7	3	2	59	21	5th Mar	8th Mar
12	7	3	3	76	24	9th Mar	11th Mar
13	7	3	1	52	27	10th Mar	12th Mar
14	7	3	2	60	30	16th Mar	18th Mar
15	7	3	3	81	33	18th Mar	20th Mar
16	7	3	1	79	39	23rd Mar	25th Mar
17	7	3	3	93	42	29th Mar	1st April
18	7	3	2	78	46	7th April	9th April
19	7	3	1	88	51	16th April	18th April
20	7	3	2	79	55	18th April	21st April
21	7	3	1	89	61	6th May	8th May
22	7	3	2	80	65	7th May	10th May
23	7	3	2	84	70	19th May	21st May
24	7	3	1	92	76	27th May	29th May
25	7	3	2	88	80	8th Jun	10th Jun
26	7	3	1	98	87	17th Jun	19th Jun
27	7	3	2	95	92	21st Jun	23rd Jun
28	7	3	2	107	98	10th Jul	13th Jul
29	7	3	1	102	105	16th Jul	19th Jul
30	7	3	1	102	112	29th Jul	31st Jul

Table 4.4h: Schedule result using the best bunching factor for Month 8

S/NO	ORDER NO	BEST BUNCHING FACTOR	FINISHING TYPE	ORDER SIZE (KG)	RELEASE DATE	EARLIEST EVENT DATE	LATEST EVENT DATE
1	8	2	1	24	1	31st Jul	3rd Aug
2	8	2	3	37	2	4th Aug	6th Aug
3	8	2	1	27	4	7th Aug	10th Aug
4	8	2	2	39	6	9th Aug	11th Aug
5	8	2	3	46	8	11th Aug	13th Aug
6	8	2	1	39	11	14th Aug	17th Aug
7	8	2	2	64	15	17th Aug	19th Aug
8	8	2	3	59	17	18th Aug	20th Aug
9	8	2	3	66	19	25th Aug	27th Aug
10	8	2	1	55	22	28th Aug	1st Sep
11	8	2	2	74	26	1st Sep	3rd Sep
12	8	2	3	66	28	3rd Sep	5th Sep
13	8	2	3	72	31	10th Sep	12th Sep
14	8	2	1	57	35	12th Sep	15th Sep
15	8	2	2	100	40	21st Sep	23rd Sep
16	8	2	3	76	42	23rd Sep	28th Sep

17	8	2	3	81	45	2nd Oct	5th Oct
18	8	2	1	91	51	6th Oct	8th Oct
19	8	2	3	82	54	11th Oct	13th Oct
20	8	2	3	85	56	23rd Oct	26th Oct
21	8	2	1	94	62	27th Oct	29th Oct
22	8	2	3	87	65	1st Nov	3rd Nov
23	8	2	3	88	68	13th Nov	16th Nov
24	8	2	1	99	75	17th Nov	19th Nov
25	8	2	3	94	78	22nd Nov	24th Nov
26	8	2	3	94	81	6th Dec	9th Dec
27	8	2	3	95	85	15th Dec	17th Dec
28	8	2	3	99	88	27th Dec	30th Dec
29	8	2	3	106	91	10th Jan	13th Jan
30	8	2	3	110	95	19th Jan	21st Jan

Table 4.4i: Schedule result using the best bunching factor for Month 9

S/NO	ORDER NO	BEST BUNCHING FACTOR	FINISHING TYPE	ORDER SIZE (KG)	RELEASE DATE	EARLIEST EVENT DATE	LATEST EVENT DATE
1	9	2	1	30	2	21st Jan	23rd Jan
2	9	2	2	11	2	22nd Jan	24th Jan
3	9	2	2	25	4	23rd Jan	26th Jan
4	9	2	3	12	5	24th Jan	27th Jan
5	9	2	3	31	6	26th Jan	28th Jan
6	9	2	2	38	8	30th Jan	3rd Feb
7	9	2	3	35	9	1st Feb	4th Feb
8	9	2	3	41	10	3rd Feb	5th Feb
9	9	2	1	31	12	4th Feb	6th Feb
10	9	2	3	54	14	10th Feb	12th Feb
11	9	2	1	32	16	11th Feb	13th Feb
12	9	2	2	49	18	13th Feb	15th Feb
13	9	2	3	63	20	17th Feb	19th Feb
14	9	2	1	41	23	18th Feb	20th Feb
15	9	2	2	61	27	21st Feb	24th Feb
16	9	2	1	42	30	26th Feb	28th Feb
17	9	2	3	65	32	29th Feb	1st May
18	9	2	2	68	35	4th Mar	7th Mar
19	9	2	3	68	37	5th Mar	8thMar
20	9	2	1	52	40	9thMar	11thMar
21	9	2	3	76	43	12th Mar	15th Mar
22	9	2	2	82	47	18thMar	21stMar

23	9	2	3	81	50	21stMar	23rdMar
24	9	2	1	56	54	22ndMar	24thMar
25	9	2	2	96	59	1st Apr	3rd Apr
26	9	2	3	110	62	5th Apr	7th Apr
27	9	2	1	59	66	6th Apr	8th Apr
28	9	2	1	62	70	20th Apr	22nd Apr
29	9	2	1	69	75	4th May	6th May
30	9	2	1	106	82	25th May	27th May

Table 4.4j: Schedule result using the best bunching factor for Month 10

S/NO	ORDER NO	BEST BUNCHING FACTOR	FINISHING TYPE	ORDER SIZE (KG)	RELEASE DATE	EARLIEST EVENT DATE	LATEST EVENT DATE
1	10	2	1	13	1	26th May	27th May
2	10	2	2	22	3	30th May	1st Jun
3	10	2	3	16	4	31st May	2nd Jun
4	10	2	3	22	5	1st Jun	3rd Jun
5	10	2	1	19	6	2nd Jun	4th Jun
6	10	2	1	19	7	3rd Jun	6th Jun
7	10	2	2	34	8	4th Jun	7th Jun
8	10	2	3	35	9	7th Jun	9th Jun
9	10	2	3	36	10	8th Jun	10th Jun
10	10	2	2	39	12	11th Jun	14th Jun
11	10	2	3	39	13	14th Jun	16th Jun
12	10	2	1	46	16	16th Jun	18th Jun
13	10	2	3	43	18	21st Jun	23rd Jun
14	10	2	3	43	19	22nd Jun	24th Jun
15	10	2	1	46	22	24th Jun	27th Jun
16	10	2	2	68	26	25th Jun	28th Jun
17	10	2	3	46	28	29th Jun	1st Jul
18	10	2	3	53	29	6th Jul	8th Jul
19	10	2	1	51	33	7th Jul	9th Jul
20	10	2	2	99	38	11th Jul	13th Jul
21	10	2	3	58	40	13th Jul	15th Jul
22	10	2	3	73	43	20th Jul	22nd Jul
23	10	2	1	68	47	22nd Jul	25th Jul
24	10	2	2	109	53	1st Aug	3rd Aug
25	10	2	3	102	56	2nd Aug	4th Aug
26	10	2	1	70	61	11th Aug	13thAug
27	10	2	3	110	65	16th Aug	18th Aug
28	10	2	1	70	69	25th Aug	27th Aug
29	10	2	1	83	75	16th Sep	18th Sep
30	10	2	1	108	82	25th Sep	27th Sep

Table 4.3(a-j) shows result of the best bunching factor for ten different orders with their release days. The earliest event date and latest event dates were presented for the customer to know on demand the likely time to expect delivery of goods.

4.4 Release Dates for Sorted and Unsorted order

Tables 4.5(a & b) shows the results obtained when Scheduling ten batch of orders that came on ten different months in an unsorted form (i.e., conventional method used by the company) and in a sortedform (i.e., using the agent based approach). The finishing times for the unsorted orders represent what will happen if the orders were processed on first come first served (FCFS) basis. The finishing times for the sorted orders represent what the proposed ABM would achieve when the most favorable bunching factor is applied. This comparison is important because the company used as case study is at the moment using first come first served approach which is now improved upon by the introduction of sorting and bunching factor in this model. Each finishing type machine has a capacity per day in kilograms. For example, finishing type one machine has a capacity of 15kg per day, finishing type two machine has a capacity of 19kg per day and finishing type three machine has a capacity of 30kg per day. By looking at the finishing type demanded by the customer and the capacity per day of the machine that produces it, the number of days each customer order will take to process is determined.

Table 4.5a: Release Date for order 1-10 Scheduled on First come First Served (FCFS)

S/NO	Release days for orders in Month1	Release days for orders in Month2	Release days for orders in Month3	Release days for orders in Month4	Release days for orders in Month5	Release days for orders in Month6	Release days for orders in Month7	Release days for orders in Month8	Release days for orders in Month9	Release days for orders in Month10
1	4	2	5	1	2	1	4	4	2	2
2	9	6	11	5	10	6	5	7	2	6
3	11	9	14	8	13	12	10	11	5	7
4	14	11	19	10	15	17	15	16	9	11
5	17	15	23	12	16	20	18	19	12	12
6	18	18	26	15	18	21	20	22	16	14
7	19	19	28	16	19	23	27	23	23	19
8	22	21	31	17	25	24	30	26	28	20
9	25	25	34	19	29	25	34	29	33	21
10	29	26	37	20	32	26	40	33	37	24
11	30	29	39	33	34	28	43	35	39	30
12	32	32	42	36	36	30	47	37	40	31
13	34	34	46	39	39	33	54	44	42	32
14	35	39	46	44	43	35	60	47	45	33
15	37	41	49	49	48	38	63	49	47	35
16	38	46	49	51	49	42	67	59	52	39
17	40	48	52	53	51	43	67	59	52	39
18	44	51	54	58	52	44	70	62	54	41
19	47	53	60	60	57	47	72	64	56	45
20	54	55	65	64	59	50	78	66	59	45
21	60	60	68	68	65	53	81	68	63	49
22	63	63	70	72	68	54	87	74	66	50
23	65	66	71	75	70	57	93	78	68	55
24	66	67	76	77	71	60	94	80	71	62
25	67	70	78	80	75	64	95	83	73	67
26	68	74	82	81	76	64	96	85	76	69
27	73	76	83	85	80	69	100	86	76	73
28	77	81	89	87	82	70	106	89	77	76
29	78	86	91	88	84	74	109	92	78	80
30	80	88	93	92	87	78	111	95	82	82

The interesting point with sorting of order by the agent according to levels is on the area of customer satisfaction. The release dates for the sorted order meets the lead time. The sorting arrangement helps to clear undue delay of small order beyond the lead time.

Table 4.5b: Release date for ABM Scheduled order 1-10

S/NO	Release days for orders in Month1	Release days for orders in Month2	Release days for orders in Month3	Release days for orders in Month4	Release days for orders in Month5	Release days for orders in Month6	Release days for orders in Month7	Release days for orders in Month8	Release days for orders in Month9	Release days for orders in Month10
1	2	1	1	1	1	1	2	1	2	1
2	3	3	3	2	2	2	3	2	2	3
3	4	5	3	3	3	4	5	4	4	4
4	5	7	4	4	5	5	6	6	5	5
5	6	8	6	6	7	6	6	8	6	6
6	7	10	8	8	8	6	8	11	8	7
7	8	11	9	9	10	7	12	15	9	8
8	9	13	11	11	11	10	14	17	10	9
9	10	15	13	12	12	12	15	19	12	10
10	12	17	15	14	14	13	17	22	14	12
11	14	19	17	16	17	14	21	26	16	13
12	16	21	20	19	19	17	24	28	18	16
13	17	24	23	21	21	21	27	31	20	18
14	20	27	25	23	24	23	30	35	23	19
15	22	30	28	26	27	25	33	40	27	22
16	24	33	31	30	30	28	39	42	30	26
17	26	36	35	32	32	29	42	45	32	28
18	30	39	39	36	35	35	46	51	35	29
19	34	42	42	40	38	37	51	54	37	33
20	37	45	46	42	41	39	55	56	40	38
21	41	48	49	45	44	45	61	62	43	40
22	44	52	55	49	47	48	65	65	47	43
23	49	55	59	54	52	51	70	68	50	47
24	51	59	63	57	55	56	76	75	54	53
25	55	63	66	62	58	59	80	78	59	56
26	58	69	71	65	63	64	87	81	62	61
27	62	73	75	69	68	67	92	85	66	65
28	69	78	80	73	73	71	98	88	70	69
29	72	83	86	76	79	77	105	91	75	75
30	79	89	93	81	86	80	112	95	82	82

The graphs of ten different orders for ABM and that of FCFS are shown in figs. 4.11(a-j). From the graph of fig.4.11a, it is observed that the release dates for the ABM scheduled order is smaller, maintaining the lead time expected to release each order for the set of jobs. The flow of the graph shows the ABM scheduled order having smaller release date (i.e. less time taken to release the order) while the FCFS scheduled order had higher release date (i.e. higher time taken to release the same order as that of ABM scheduled order).

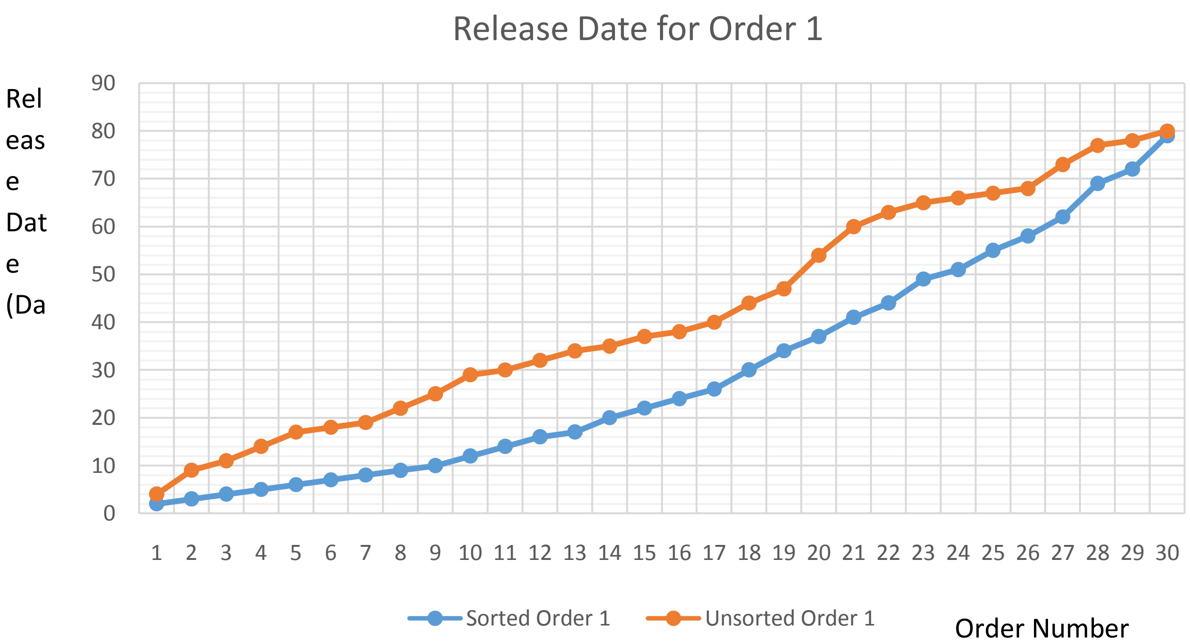


Fig. 4.11a: The Graph of the Release Date for the ABM scheduled Job versus the FCFSscheduled Job for Order 1

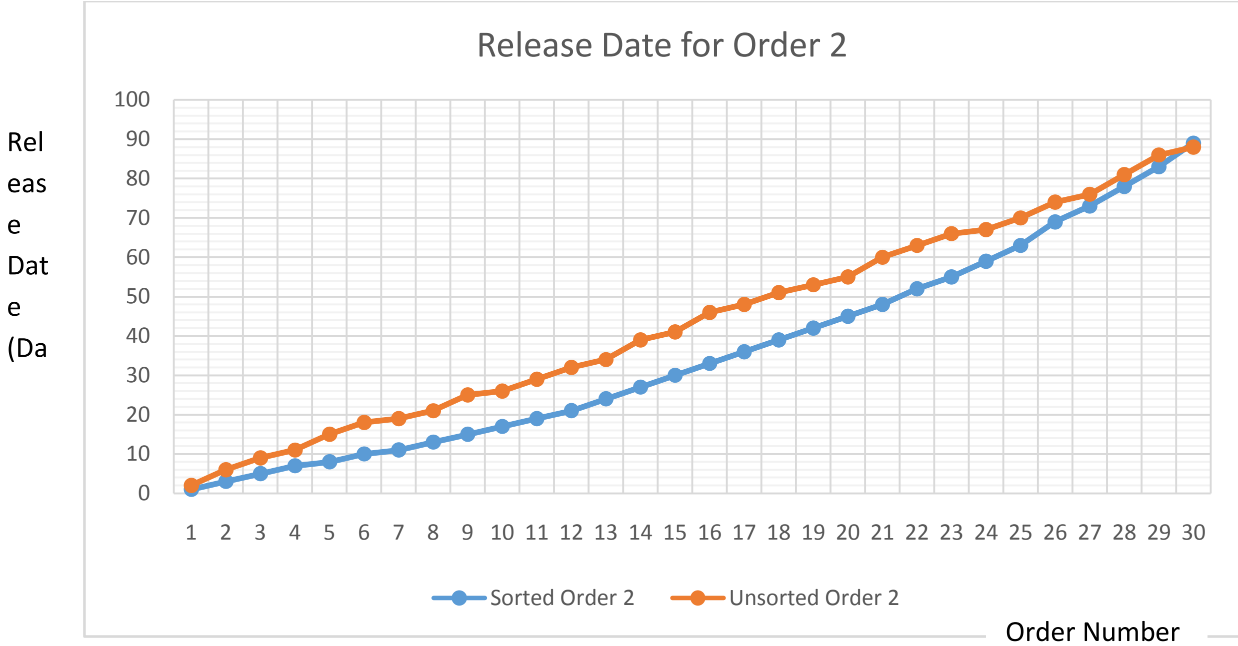


Fig. 4.11b: The Graph of the Release Date for the ABM scheduled Job versus the FCFSscheduled Job for Order 2

Figures. 4.11 (a-b) give clear picture of the above explanation with ABM scheduled order having earlier release date to finish to that ofFCFS scheduled order as can be seen on the graphs of figures. 4.11 (a-b).

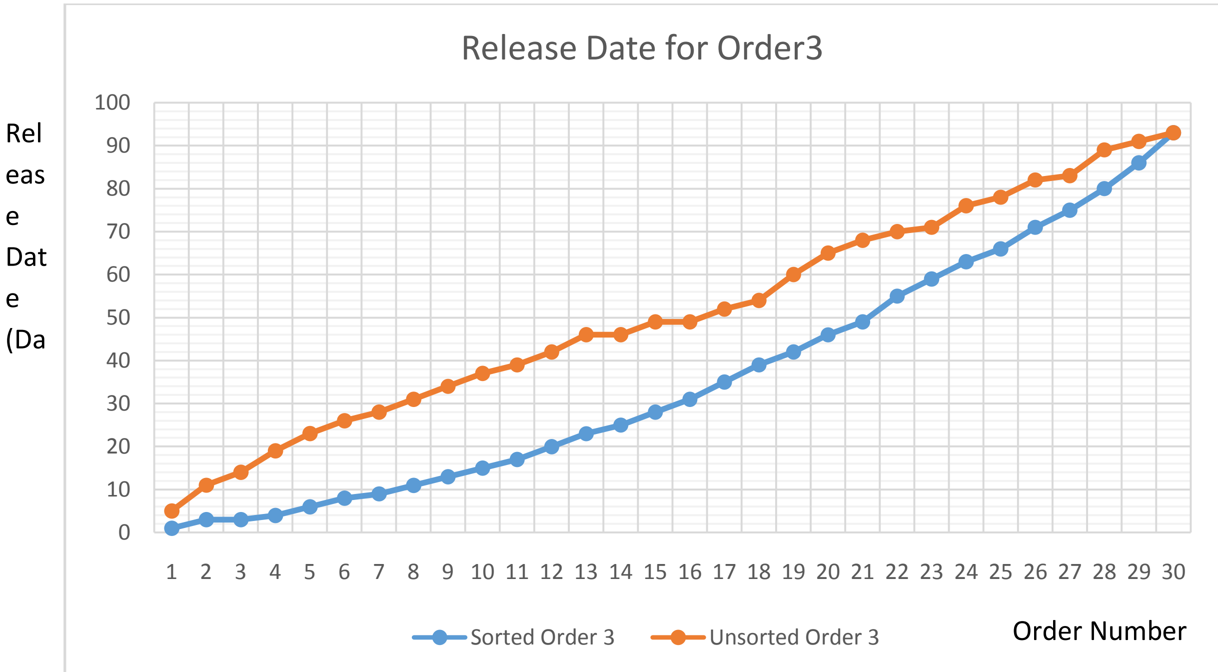


Fig. 4.11c: The Graph of the Release Date for the ABM scheduled Job versus the FCFSscheduled Job for Order3

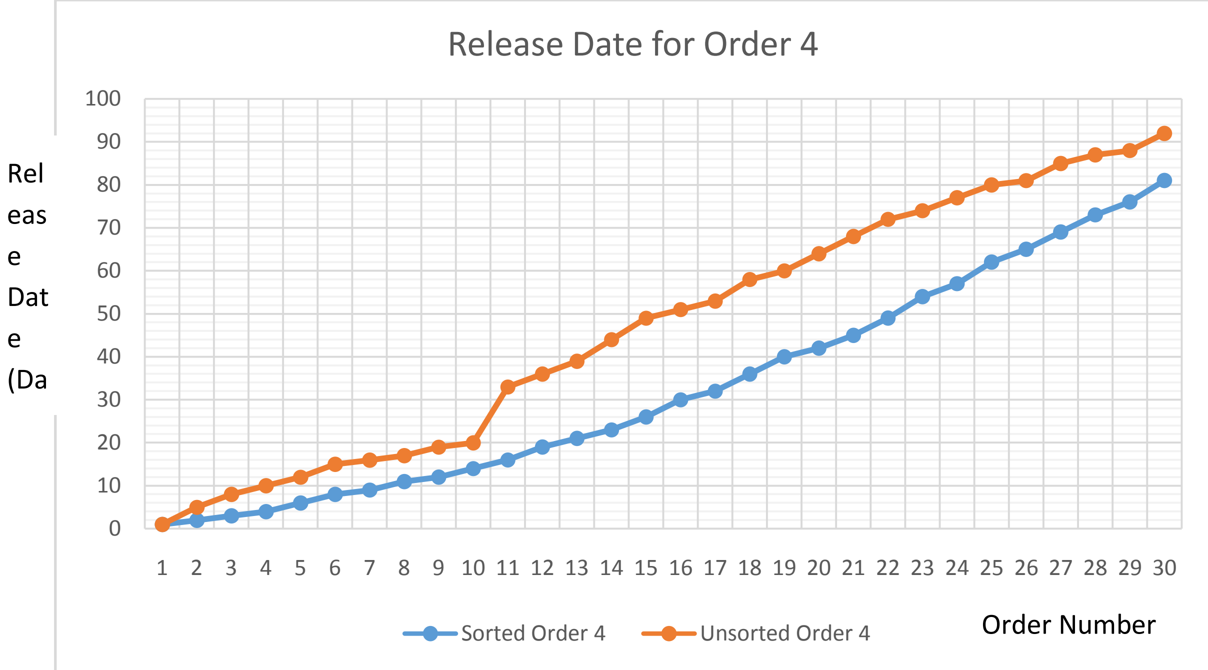


Fig. 4.11d: The Graph of the Release Date for the ABM scheduled Job versus the FCFSscheduled Job for Order 4

The same scenario can be seen in figures. 4.11 (c-d), with ABM scheduled order having earlier release date to finish a set of monthly order.The remaining graphs clearly support the need for ABM as a solution to industrial operations scheduling.

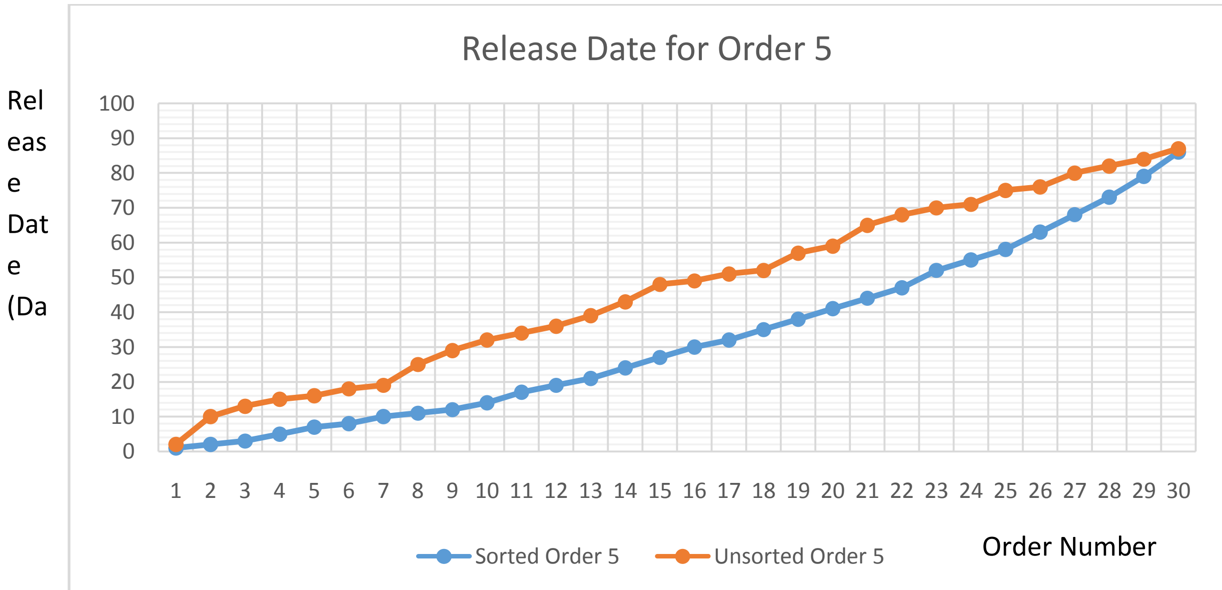


Fig. 4.11e: The Graph of the Release Date for the ABM scheduled Job versus the FCFS

scheduled Job for Order5

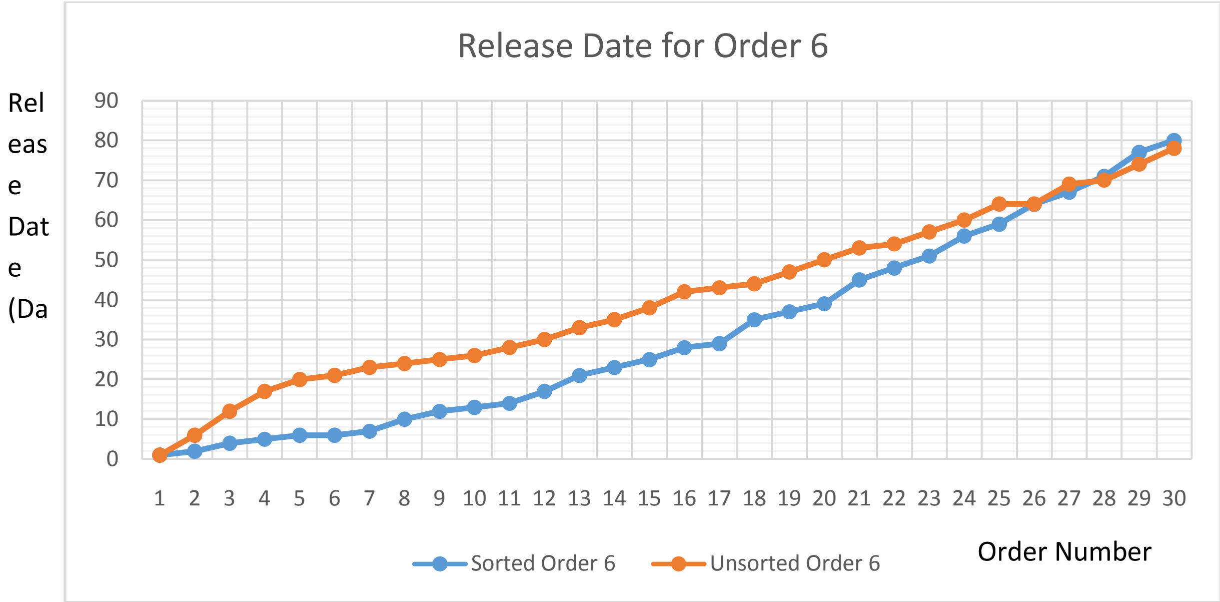


Fig. 4.11f: The Graph of the Release Date for the ABM scheduled Job versus the FCFS scheduled Job for Order 6

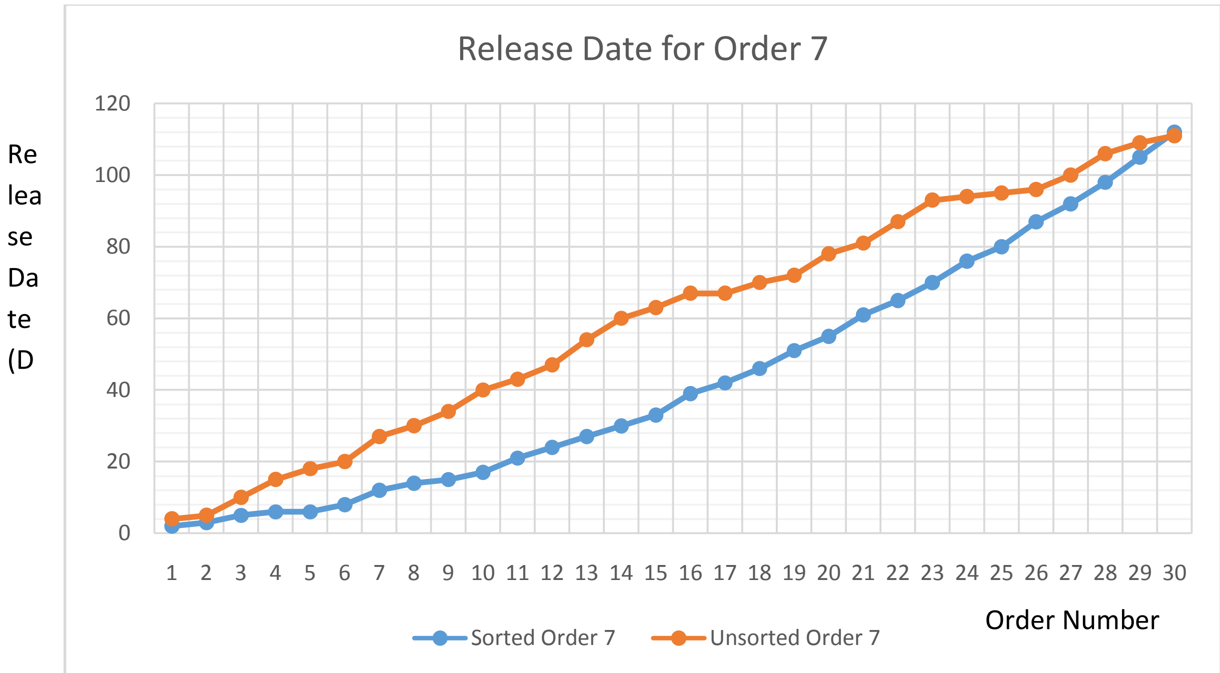


Fig. 4.11g: The Graph of the Release Date for the ABM scheduled Job versus the FCFS scheduled Job for Order7

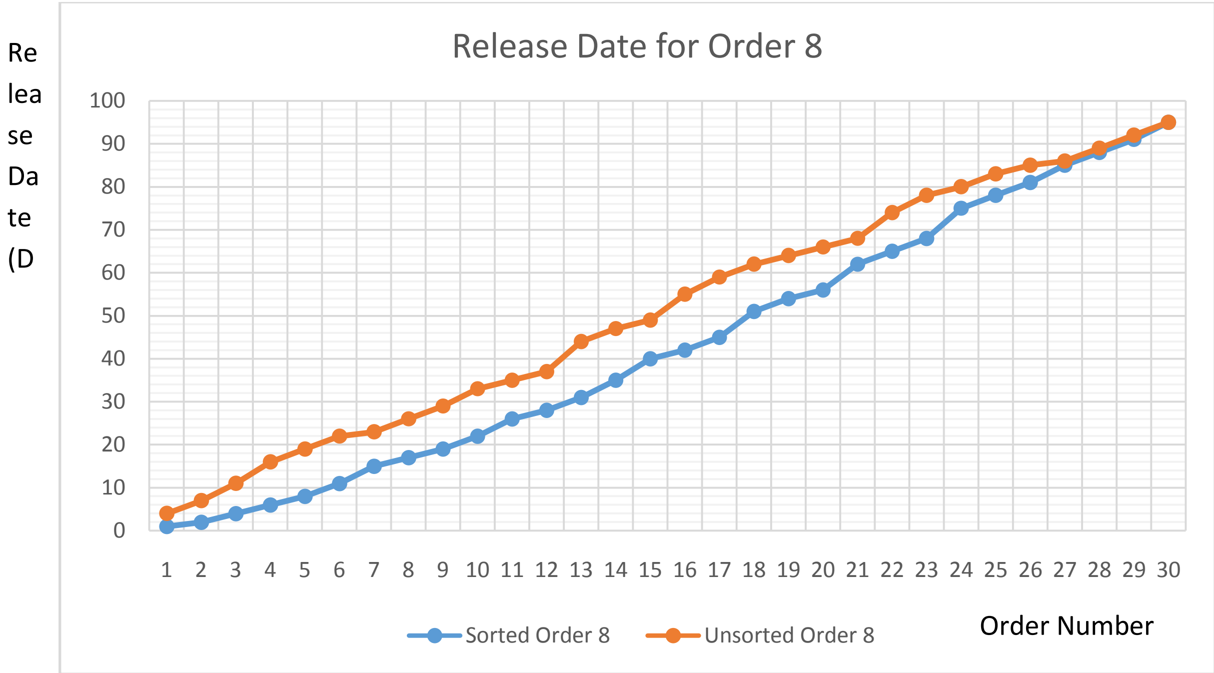


Fig. 4.11h: The Graph of the Release Date for the ABM scheduled Job versus the FCFS scheduled Job for Order8

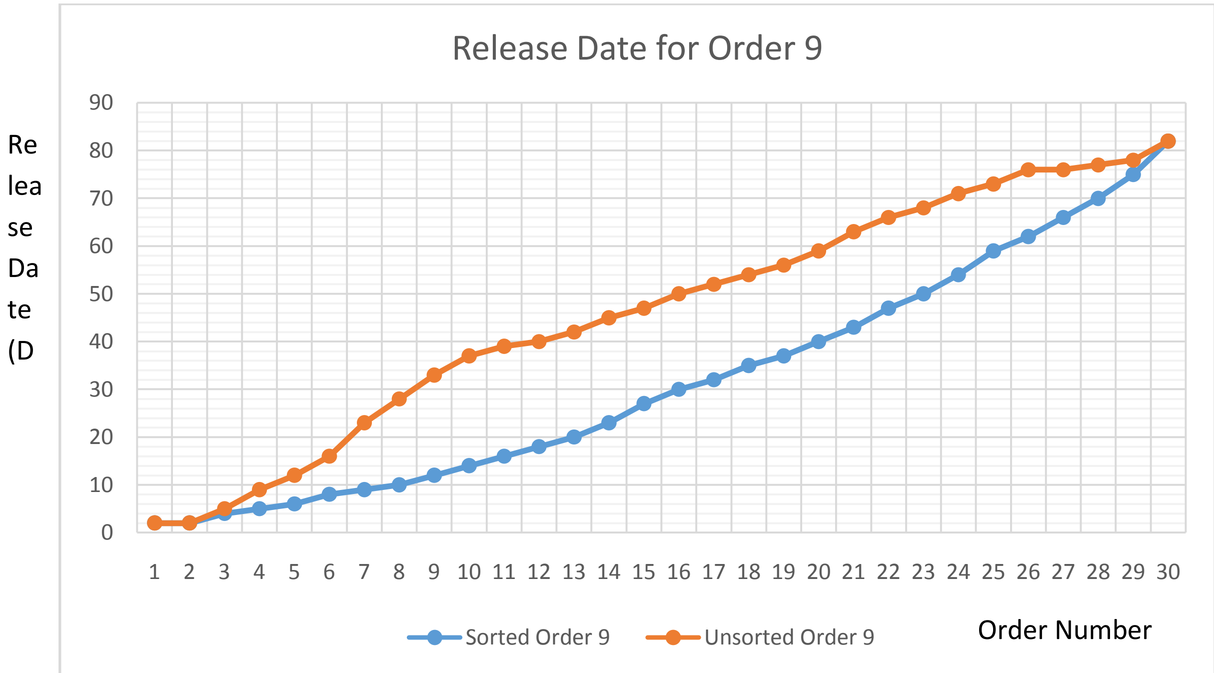


Fig. 4.11i: The Graph of the Release Date for the ABM scheduled Job versus the FCFS scheduled Job for Order 9

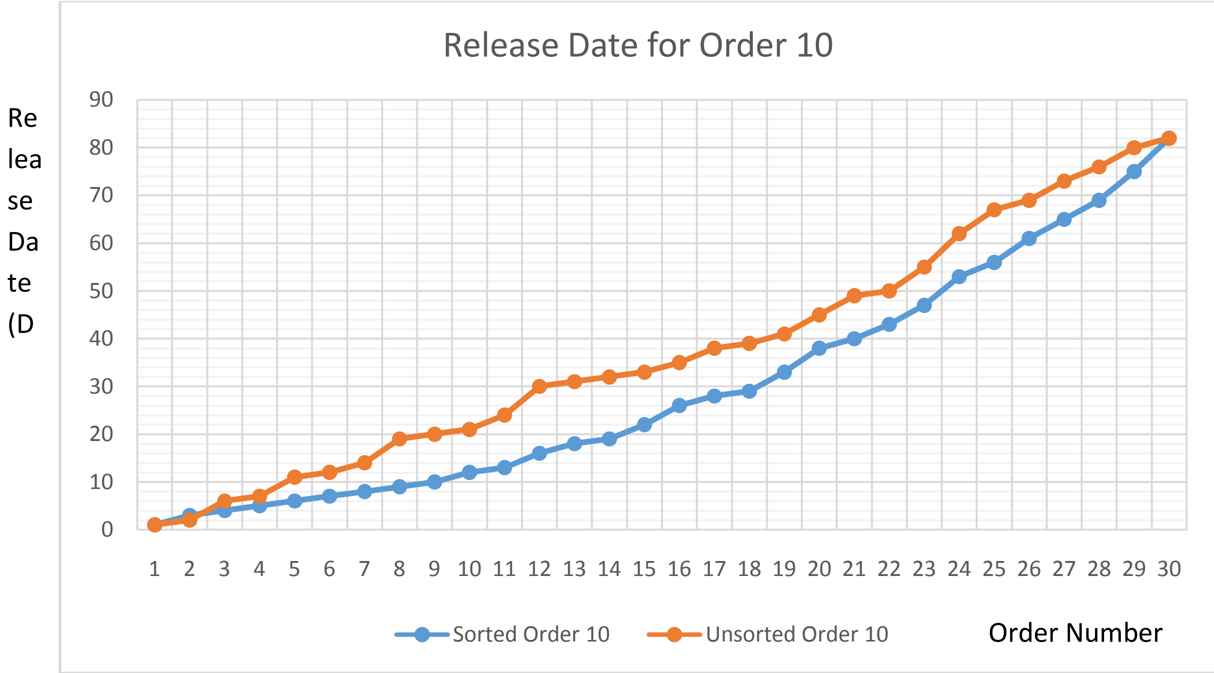


Fig. 4.11j: The Graph of the Release Date for the ABM scheduled Job versus the FCFS scheduled Job for Order 10

Figures 4.11(a-j) shown above are the graphs of the modeled agent based job shop scheduler proposed in this research work with the conventional job scheduling process obtained from the case study companies. The release date for the ABM scheduled order shows thatthe agent-based model has a betterresult compared to the initial schedule process used by the companiesin terms of customer satisfaction. Here, jobs are scheduled with respect to their type of finish and order size, which clearsorder queue. Smaller orders whichtheir due date can be met in one day are processed first and released before large orders with acceptable large lead time.

4.5 Model Validation using D.G. Kendall queuing System

For the purpose of validation and testing of the agent-based job shop scheduling model, a classical method for poisson arbitrary distribution with nonpreemptive discipline by Kendall (1953) was used.

4.5.1 D.G. Kendall Queue Model Result for Ten Order

The mathematical model by D.G. Kendall is stated thus;$(\boldsymbol{M_i/G_i/1}):(\boldsymbol{NPRP}/\propto/\propto)$**,** thesymbol NPRP is used with the Kendall notation to represent the nonpreemptive discipline; $M_i$and G_i stand for poisson and arbitrary distributions. (Taha, 1968)

Let $F_i(t)$ be the CDF of the arbitrary service time distribution for the ith queue (i=1,2,… M), and let $E_i\{t\}$ and $Var_i\{t\}$ be the mean and variance, respectively; let λ_i be the arrival rate at the ith queue per unit time. Define $Lq^{(k)}$, $Wq^{(k)}$,$Ws^{(k)}$and $Ls^{(k}$as;

Ls= expected number of customers in system

Lq= expected number of customers in queue

Ws = expected waiting time in system

Wq = expected waiting time in queue

Except that they now represent the measures of the kth queue.

Then the results of this model are given by

$Wq^{(k)} = \frac{\sum_{i=1}^{n} \lambda_i (\mathrm{E}_i^2\{t\} + Var_i\{t\}}{2(1-S_{k-1})(1-S_k)}$(Kendall, 1953)

$Lq^{(k)} = \lambda_k Wq^{(k)}$

$Ws^{(k)} = Wq^{(k)} + E_k\{t\}$

$Ls^{(k)} = Lq^{(k)} + P_k$

Where $P_k = \lambda_k E_k\{t\}$

$S_k = \sum_{i=0}^{k} P_i < 1$ K=1,2,… M

$S_0 \equiv 0$

Where, $\mathrm{E}_i^2\{t\}$= mean

$Var_i\{t\}$= variance

S = time interval

λ_k = constant service rate per a day

P_k= probability distribution.

Using the sorted order of Table 4.2a, the values for the first order (first month) are given below;

SL_1= 16 27 32 43 51 60 70 96 101

SL_2 = 13 13 20 22 29 51 62 104

SL_3 = 13 36 38 40 50 54 59 82 83 85 88 102 107

Where SL_1 is finishing type 1

SL_2 is finishing type 2

SL_3 is finishing type 3

Therefore, the mean for SL_1 is

$$\frac{16 + 27 + 32 + 43 + 51 + 60 + 70 + 96 + 101}{9}$$

$$\text{Mean} = \frac{496}{9} = 55.11$$

$$\lambda_1 = \frac{mean}{3} = \frac{55.11}{3} = 18.37$$

For SL_2

Mean = $\frac{13+13+20+22+29+51+62+104}{8} = \frac{314}{8}$

Mean = 39.25

$$\lambda_2 = \frac{39.25}{3} = 13.08$$

For SL_3

Mean = $\frac{13+36+38+40+50+54+59+82+83+85+88+102+107}{13}$

Mean = 64.38

$$\lambda_3 = \frac{64.38}{3} = 21.46$$

But $P_i = \lambda_i Ei\{t_i\}$

$$\therefore P_1 = \lambda_1 E\{t\} = 18.37 \left(\frac{1}{15}\right) = 1.2247$$

$P_2 = 13.08 \left(\frac{1}{19}\right) = 0.6884$

$P_3 = 21.46 \left(\frac{1}{30}\right) = 0.7153$

Where 15kg, 19kg and 30kg are the maximum production capacity for product type 1, 2 and 3 per normal production day respectively.

$S_1 = P_1 = 1.2247$

$S_2 = P_1 + P_2 = 1.2247 + 0.6884 = 1.9131$

$S_3 = P_1 + P_2 + P_3 = 1.9131+0.7153 = 2.6284$

The due date for the complete schedule for order 1 is 2.6284 X 30 = 78.852

For the second order (order 2)

$SL_1 = \frac{22+23+23+25+34+39+45+46+51+56+71+75+83}{13}$

Mean = 45.61

$$\therefore \lambda_1 = \frac{45.61}{3} = 15.21$$

For SL_2

Mean = $\frac{30+30+35+38+55+60+63+78+101+104}{10}$

Mean = 59.4

$$\lambda_2 = \frac{59.4}{3} = 19.8$$

For SL_3

Mean = $\frac{47+62+72+79+87+98+109}{7} = \frac{554}{7}$

Mean = 79.14 $\therefore \lambda_3 = \frac{79.14}{3} = 26.38$

$P_1 = \lambda_1 E\{t_1\} = 15.21 \left(\frac{1}{15}\right) = 1.014$

$P_2 = 19.8\left(\frac{1}{19}\right) = 1.042$

$P_3 = 26.38(\frac{1}{30}) = 0.879$

$S_1 = P_1 = 1.014$

$S_2 = P_1 + P_2 = 2.056$

$S_3 = P_1 + P_2 + P_3 = 2.052+0.879 = 2.935$

The Due date for the complete schedule for order 2 is 2.935 x 30 = 88.06

The complete value of the last release date for a queue of ten different orders using D.G.Kendall model is shown in table 4.6, in comparison to that of agent-based job shop scheduling model.

Table 4.6: Comparison for the last release date for the proposed ABM and D. G. Kendal classical model

Order No	**Release Date (Days) for Agent Model (Table 4.5b)**	**Release Date (Days) for Classical Model**
1	79	78.85
2	89	88.06
3	93	92.92
4	81	87.38
5	86	85.338
6	80	88.51
7	112	106.59
8	95	103.67
9	82	82.72
10	82	85.82

Table 4.6 presents the latest completion time for that of agent-based model and the classical model by D.G. Kendall. The result of the agent based model shows a better result in comparison to that of classical model. The graph of figure 4.13 shows the comparison of agent-based model to that of classical model.

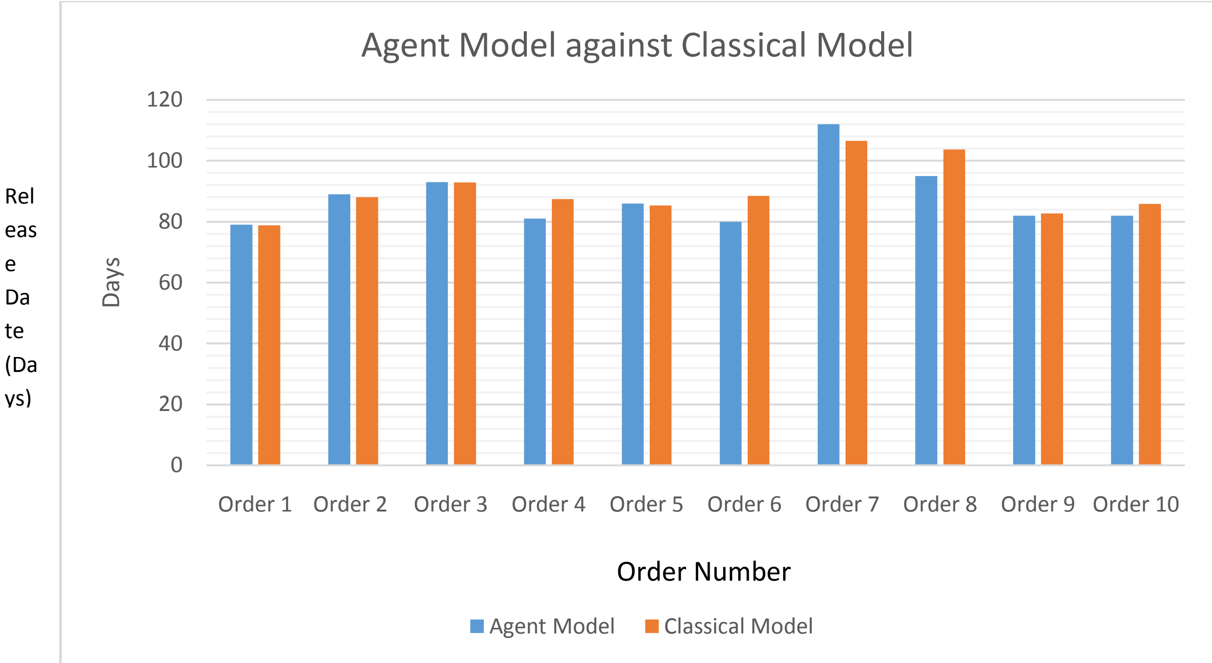


Figure 4.12: Graph showing the comparison between Agent Model against Classical Model

Figure 4.12, shows clearly the performance of the ABM model, with the latest due date for the complete job out performing that of the classicalmodel in orders 4,6,8,9 and 10 while it still relatively close to that of classical method in other once, as can be seen in order 1 that has the agent based model result as 79 dayswhile classical model had 78.85approximately 79 days. The classical model seems better in order 7 with about 106 days against ABM's 112 days.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1 Summary of Achievements

The following accomplishments have been made as a result of this research:

- An application of Markov chain for the model was used to work out the extra raw material needed to allow for wastages and the factor for the given order, which includes the cost of machine maintenance and the raw material cost for a given set of order.
- The model adopted bunching technique with different bunching factors to ascertain the best bunching factor that gives the minimum makespan to be used to schedule the given job.
- Four important agents useful in the factory floor were developed to handle every activity from order reception to the release of the processed scheduled job to the client.
- The developed ABM was successfully validated by comparing the results for scheduled order with the different result obtained from the classical queuing method by D.G. Kendall.

5.2 Problems Encountered

Some of the major problems encountered in this research included:

i. Obtaining real data and the production scheduling approach used by theindustries under study.These were considered business secret and there was fear of divulging their method of production to rival company.

ii. Other problems encountered include funding of the research especially for data gathering from the case study company.

iii. Sourcing information from reputable sources such as high impact factor journals.

5.3 Contribution to Knowledge

In this research work, the following contributions to the body of knowledge have been made:

- The model introduced an important technique that can choose, out of the several factors, the best factor that will give the minimum makaspan for scheduling a given set of orders. The bunching factor as can be seen from the result gotten in chapter four.
- The application of Markov chain to work out the extra raw material required at the input to make up for wastages such that output remains as required, also used to work out the machine maintenance cost to charge the customer.
- A well-crafted scheduler agent algorithm was developed.
- The developed model has a human/machine interaction that can adjust to the best schedule algorithm to take care of important jobs

requiring preferential treatment. This made the model flexible and a better option for scheduling stochastic processes.

5.4 Suggestions for Further Improvement

Further research is advocated in the area of machine arrangement; an investigation of the gains accruing from using three sets of series machines with only two sets of parallel output finishing machine types is suggested. It is envisaged that a better arrangement that would reduce machine cost could be worked out. Theissues relating to production line job routing, process planning and machine part programming should also be worked on.

5.5 Conclusion

Modern software practices are based on a template design approach in which recurring elements are codified and reused for new applications; this approach has proven very valuable in designing model's as well as software. Scheduling, understood to be an important tool for manufacturing and engineering, has a major impact on productivity of a process. In manufacturing, the purpose of scheduling is to minimize the production time and cost, by telling a production facility what to make with which staff and on which machine. The methodical leverages of the ABM technique modelled in this research work will give the minimum production time and a reduction in production cost for any stochastic order in complex manufacturing industries

5.2 Recommendation

The following recommendations were generated as a result of this research:

- Deployment of agent based schedule that incorporate Markov chain to determine the cost factor of the equipment, material cost and wastages for cost effective manufacturing automation.
- The application of bunching factor in the scheduling (assignment) of order for production in every industrial setupto help reduce time spent in production of a certain job.
- The need to apply this model in mostindustries for customer satisfaction as it guarantees orders grouped in ascending order of magnitude and scheduled so that the lead time is always met.

REFERENCES

AkeelaM.Al-Atroshi, Sama T. Azez, Baydaa S. Bhnam (2013) "An Effective Genetic Algorithm for job shop scheduling with fuzzy Degree of satisfaction," University of Dohuk, Irag pp. 180

Albers, Susanne; TorbenHagerup (1992) "Improved parallel integer sorting without concurrent writing" proc. Of the third annual ACM-SIAM Symposium on Discrete algorithms Pp. 463-472.

Albert Jones, Luis C. Rabelo(2009) "Survey of job shop scheduling techniques" MIT, USA, pp 232

Albert Jones, C. Rebelo (2007) "Survey of job shop scheduling techniques" National institute of standard pp: 201-223.

Asher Levin, David (2009). "Markov Chains and mixing times" , UK PP 16-23.ISBN 97.8-0-8218-47-39-8

Axelrod, R. (1997). "The complexity of cooperation: Agent-Based-Models of competition and collaboration" Princeton, NJ: Princeton University Press

Axelrod, R. (1997). "The dissemination of culture: A model with local convergence and global polarization". The journal of conflict resolution, 41(2): 203-226.

Banerjee, Sudipo; Carlin, Bradley P. and Gelfrand, Alanp (2012). "Hierarchical modeling and analysis for spatial data (second ed). CRC press Pp. xix. ISBN 978-1-4398-1917-3.

Baker, K.R [1974]. "Introduction to sequencing and scheduling, Wiley & sons.

Bartal Y.; A. Fiat, H. Karloff; R. Vohra (1992) "New Algorithms for an Ancient scheduling problem" proc. 24th ACM symp. Theory of computing pp. 51-58.doi:10.1145/129712.129718

Bauer, A, Bowden, R., Brown, I. Duggen J en Lyons, G.,(1991) "shop floor control systems, design and implementation, "chapman & Hall, London.

Benders, J. (1960) "Partitioning procedures for solving mixed-variables mathematical programming problems" Numberschemathematik, 4(3): 238-252.

Blazewicz J., Domschke W., and Pesch E. (1996) "The job shop scheduling problem: conventional and new solution-techniques" European journal of operational research, 93:1-33,.

Blazewic, S. Ecker, K. H., Pesch. E., Schmodt, G. Ind J. Weglerz, (2001) "Scheduling Computer and Manufacturing processes", Berlin (springer) , ISBN 3-540-41931-4. PP. 322, 323.

Bmabeau E (2001) "Agent-Based modeling: Methods and techniques for simulating Human systems" in proc. National academy of science 99(3):7280-7287.

Buscher U. and Shen L..(2009) "An integrated tabu search algorithm for the lot streaming problem in job shops" European journal of operational research, vol. 19 pp 385-399.

Champ – Hyun Ju, Guobin Chen and Jemes Choi. (2003) "A framework for BDI Agent – Based Software Engineering" SIU Pp. 303.

Casti, J. (1997)." Would-be worlds: How simulation is changing the world of science". New York: Wiley.

Charles M. Macal, Michael J. North (2011) "Introductory tutorial: Agent-based modeling and simulation" proc. Of the 2011 winter simulation conference, Argonne, IL 60439 USA, pp. 1456

Coffman, E-G., Jr., Graham, R.L. (1972) "optimal scheduling for two-processor systems", Acts Informatica 1: pp. 200-213.

Conway, S. R. (2006). "An Agent-Based Model for Analyzing Control Policies and the Dynamic Service-Time performance of a capacity-constrained Air Traffic Management Facility", In processing of ICAS

2006-25th congress of the International Council of the Aeronautical sciences. Hambing, Germany, 3-8 Sept. 2006.

Dantzig, G. and P. walfe (1960) "Decomposition principles for linear programs", 8(1): 101-111

Davis, W. and A. Jones (2001). "A real-time production scheduler for a stochastic manufacturing environment". 1(2): 101-112.

Del-Moral, Pierre (2013). "Mean field simulation for Monte Carlo integration". Chapman and Hall/CRC press Pp 626. Monographs on statistics and applied probability.

Dodge, Y. (2003) "Markov Chains" the Oxford Dictionary of statistical terms, OUP 0-19-92 0613-9

Dorit S. Hochbaum and Shmoys B. David (1987) "using dual approximation algorithms for scheduling problems: theoretical and practical results" journal of the ACM, 134 (1): PP 144-162

Eric Scherer (2002) "shop floor control – An integrative framework from static scheduling models towards an agile operations management" **, CH8092** Zurich Switzerland 2002, pp. 1,2.

Everitt, B. S. (2002) "The Cambridge Dictionary of statistics. CUP.ISBN 0-521-81099-x.

Epstain, J. M and Axtell, R. (1996). "Growing artificial societies: social science from the bottom up". The Brooking Institution, Washinton, DC, USA.

Fleischer, Rudolf (2009) "Algorithms" ESA 2009 prof. ISAM conf. on advanced scheduling algorithms. Pp. 14-15.

Folcik, V., G. C. Hn, and C. G. Orosz (2007). "The Basic immune Simulator: An Agent – Based Model to study the Interactions between Innate and Adaptive Immunity", Theoretical Biology and Medical Modeling, 4(39).

Frank Noe (2008) "Probability distributions of molecular observables computed from Markov models. The journal of chemical physics, 128:244103

Gabal, T.and M. Riedmiller (2007) "Adaptive reactive job-shop scheduling with reinforcement learning agents." International journal of information technology and intelligent computing 24(4):1-30.

Goodhill, Geoff (2017) "No more playing games: Alpha go AI to tackle some real world challenges" singularity hub, retrieved 2018-06-05

Graham, R (1966). "Bourds for certain multiprocessing anomalies" http://www.math.ucsd.edu/% ofErmspub166-0-41 multiprocessing (PDF). Bell system Technical journal 45 pp 1563-1581

Grimm, V.,U. Berger, F. Bastiansen, S. Eliassen, V. Ginot, J. Giske, J. Goss-Custard, T. Grand, S.K.HeinZ,G.Huse, A Huth, J.U.Depson, C. Jorgansen, W.M.Mooij.B. Muller, G. Pe'er, C piou, S.F. Railsback, A.M.Robbings, M.M Robbins, E.Rossmanth, N. Ruger, E. Strad, S. Souissi, R.A. Stillman. R. Vabo, U. Visser, and D.L. De Angelis. (2006). "A standard protocol for describing individual-based and Agent-Based models" Ecological modeling 198(1-2): 115-126.

Griffin, A. F., and C. Standish. (2007). "An Agent-Based Model of prehistoric settlement patterns and political consolidation in the Lake Titicaca Basin of peru and Bolivia, structure and Dynamics",Journal of Anthropological and Related sciences, 2(2).

Grosman, P. D. , J. A. G. Jaeger, P. M. Biron, C. Dussault, and J. –P. Ouellet (2011). "A Trade-off Between Road Avoidance and

Attraction by Roadside salt pools in Moose: An Agent-Based Model to Assess measures for Reducing moose vehicle collision", in Ecological Modeling 222(8): 1423-1435.

Gupta, Ankur and Rawlings, James (2014) "Comparison of parameter estimation methods in stochastic chemical kinetic models: examples in systems". AIChE journal 60(4) 1253-1268

Heath, B. L., F. Ciarallo, and R. R. Hill. (2009) "A survey of Agent-Based-Modeling Practices" (January 1998 to July 2008). Journal of Artificial Societies and social simulation 12(4) October.

Izquiredo, L.R., Izquiedo, S.S., Galan, J.M., and Santos, J.I.(2009). "Techniques to understand computer simulations: Markov chain analysis". Journal of Artificial societies and social simulation, 12(1):6.

JacekBlazewicz, Erwin Pesch, Malgorzata, sterna, (2000) "The disjunctive graph machines representation of the job shop scheduling problem", European Journal of operational research, volume 127, issue 2,1 December 2000, pp.317-331, ISSN 0377-2217

Jain A. Z. and Mecran S. (2002) "Deterministic job shop scheduling: past, present and future". European journal of operational research, 113:390-434.

Jennings, N. R. (2000) "On Agent-Based software Engineering". Artificial intelligent. Aug. 2000: 117:277-296.

Johnson S. M., (2003) "Optimal two – and three-stage production schedules with setup times included "Naval Ros. N. Y. USA, pp. 61-66

Kanger, D., S. Phillips; E. Torng (1994) "A Better Algorithm for an Ancient Scheduling Problem", proc. Fifth ACM Symp. Discrete Algorthms. Pp.18-23

Kendall, D.G.(1953) "Stochastic processes occurring in the theory of queues and their analysis by the method of the embedded Markov chain". The annals of mathematical statistics. 24(3) 338.

Khuri, Sami and Miryala, Sowmy a Rao (2001) "Genetic Algorithm for solving open shop scheduling problem ",proceeding of the 9thPortuguese conference on Artificial Intelligence: progress in Artificial Intelligences.

Law, A. M. (1998) "Agent-Based Modeling: A new approach to systems modeling" ASSS-journal of Artificial societies and Social Simulation of (4): pp.100,101

Lawler, E and D. Wood (2000) "Branch and bond method for modeling the job shop scheduling problem" operations research. 14, 699-719, pp: 123-128.

Lee, Alec Miller (1966) "A problem of standards of service (chapter 15)" Applied queueing theory. New York Macmillan. ISBN 0-333-04079-1

Levin, D.A., Peres, Y, and Wilmer, E.L. (2009). "Markov chains and mixing times" American Mathematical society.

Leyk, S., C. R. Binder, and J. R. Nuckols (2009) "Spatial Modeling of Personalized Exposure Dynamics: The case of pesticide use in small-scale Agriculture Production Landscapes of the Developing World", International journal of Health Geographic's, 8:17-doi:10.1186/1476-072x-8-17

Low C., Hsu C. M and Huang K. I., (2004) "Lot splitting in job shop scheduling using integer programming". International Journal of advanced manufacturing technology, vol. 24, pp773-780.

Lozzi J., F. Trusiano, M. Chinazzi, F. C. Bilkri, E. Zapheni, S. Merber, M. Aselli, E. Del. Feva, and P. man fredi. (2010). "Little Haley" : An Agent Based Approach to the Estimation of contact patterns-fitting predicted matrices to Serological Data" PlosComputBiol 2 el001021.

Malakoot, B (2013) "Operations and Production Systems with multiple Objectives". John Wiley and sons. ISBN 978-1-11-58537-8 PP.63-69.

Malleson, N. (2010). "Agent-Based Modelling of Burglany"Ph.D Thesis, school of Geography University of leads.

Marco Sarich and ChristofSciitte (2014) "Utilizing hitting times for finding metastable sets in non-reversible Markov chains". Technical report 14-32, Konrad-zuse-zentrum fir information stechnik Berlin.

MareenHallier and Carsten Hartman (2016) "Constructing Markov state models of reduced complexity from agent-based simulation data" BTU Cottbus senffenberg.

Menges, F., Mishra B. And Narzisi G. (2008). "Modeling and Simulation of E-mail social networks: A new stochastic agent-based approach". Proceeding of the 2008 winter simulation conference. 2792-2800.

Meyn, Sean P., and Richard L. Tweedic (2011) "Markov Chains and stochastic stability " Cambridge University Press. (reference, pp.iii).

Michael Pinedo, (2002) "scheduling theory, Algorithms, and systems" prentice hall, Pp. 1-6.

Mock, K. J., and J. W. Testa (2007). "An agent-based model of predator-prey relationship between transient killer whales and other marine mammals". University of Alaska Anchorage, Anchorage, AK, may 31, 2007.

Norris, James R. (1998) "Markov Chains", Cambridge University Press.

North, M.J, and C.M.Macal, (2011) "Product design pattern for Agent-Based modeling". In precedings of the 2011 winter simulation conference, edited by S. Jain, R.R. Creasey, J. Himmelspach,k.

Opalvch, J.J., J. L. Anderson, and K. Schnierce (2005) "A Risk – Based Approach to managing the international introduction of Non-native Species",Paper presented at the AEREworkshopNatural Resources at Risk, Grand Teton National Park, Wyoming, 12-14.

Othman Z., Subari K., and Morad N., (2007) "job shop scheduling with Alternative machines using Genetic Algorithm," journal Teknolosi, 41 (D). 2007.pp. 67-78.

Parzen, E. I (1962) "Stochastic Processes ", Holden-Day ISBN 0-8162-6664-6.

Philippe Baptiste, claude le pepe and Wim Nuijten. (1995)"Constraint-Based optimization and Approximation for job-shop scheduling" ILOG S.A., France.

Puckett, R.R. (2000) "Multi-Agent Crowd Behavior Simulation for Tsunami Evacuation". Master thesis, Department of information and computer science, University of Hawaii, May 2000.

Ralf Banisch (2015) "Markov processes beyond equilibrium and optimal control. Ph.D thesis, Fachbereichmathematikundinformatik, F U Berlin.

Raya K., Saravaran R., and Selladurai V., (2008)"Mult-Objective Optimization of parallel machine scheduling (PMS) using Genetic Algorithm and fuzzy Logic Approach" IE(I). journal-PR,vol. 87, 2008, pp. 26-31.

Rosemarin, Hanan; Dickerson, John and Kraus, Sarit (2017) "learning to schedule deadline and operator sensitive tasks"

Roy B., and Sussmann B., (1964) "The problems and theory of constraint disjunctives " semiral notes, D.S. No. 9 Paris, 1964.

Schelling, T. (1971). "Dynamic models of segregation" Journal of mathematical sociology 1(2) pp 143-186.

Sharma J.K (2013) "Operations Research Theory and Applications" Macmillan Publishers India Ltd. ISBN 978-9350-59336-3 Pp 664-665

Sopha, B. M., C.N. Klockner, and E. G. Hertwich (2011)"Exploring Policy for a transition to sustainable Heating System Deffusion using an Agent-Based Simulation". Energy policy (in press).

Sven Banisch, Ricardo Lima and Tanya Araujo (2012)"Aggregation and Emergence In Agent-Based models: A markov chain approach" ISEG, Lisbon (2012). ISSN 0874-4548.

Taha, Hamdy A. (1968). "Operations research: an introduction.

Taillard Instances (1972)"model the job shop scheduling problem", ordonnencement. Berlin Heidelberger, Springer pp.431-440.

Taillard E.D. (1999) "Parallel tabu search technique for the job shop scheduling problem" Technical report ORWP 89/11, DMA, EcolePolytechnique. Federale de lausenne, Switzland.

Tao Zhang, ShufangXie and Oliver Rose (2017) "Real-time job shop scheduling based on simulation and Markov decision processes".

Proceeding of the 2017 winter simulation conference. Pp 3899-3907.

Ten-Hendrik Prinz, Hao Wu, Marco Sarich, Bettina Keller, Martin Senne, Martin Held, John D. Chodera, ChristofShiitte and Frank Noe (2011) "Markov models of molecular kinetics: Generation and validation". The journal of chemical physics, 134(174105).

Tsai J. T., (2008)"An improved genetic algorithm for job shop scheduling problems using taguchi-based crossover", pp. 22-27

YihYuehwern and Arne Thesen (1991) "Semi-Markov decision models for real-time scheduling". International journal of production research. 29:11, 2331-2346.

Zhang, Z., Sharifi, H., (2001)"A methodology for achieving agilityin manufacturing organisations". Int. J. of Operationsand Production Management, 20 (4), 496-512

APPENDICES

APPENDIXA

```
% ORDER AGENT PROGRAM
typef = [1 2 3 4 5 6 7 8 9];
n = 30;
order = zeros (30,8);
fori = 1:n;
    b = rand(1,1);
    p = fix(b(1)*10)+1;
if p>9
        l = p-1;
        p = l;
end
if p<4
        type = 1;
elseif p>4 & p<7
        type = 2;
elseif p>6
        type = 3;
end

    b1 = rand(1,1);
    b2 = rand(1,1);
    d1 = fix(b1(1)*10+1);
    d2 = fix(b2(1)*10+1);
    size = d1*10+d2;
    b1 = rand(1,1);
    b2 = rand(1,1);
    b3 = rand(1,1);
    d1 = fix(b1(1)*10+1);
    d2 = fix(b2(1)*10+1);
    d3 = fix(b3(1)*10+1);
orderid = d3+d2*10+d1*100;
if size<46
ldtm = 14;
        t = 1.0e+03 *2.0170;
        level = 1;
end
if size>45 & size<76
ldtm = 28;
        t = 1.0e+03 *0.0160;
```

```
        level = 2;
end
if size>75
ldtm = 42;
        t = 1.0e+03 *0.0540;
        level = 3;
end
    order(i,1) = orderid
    order(i,2) = t
    order(i,3) = size
    order(i,4) = level
    order(i,5) = type
    order(i,6) = p
    order(i,7) = ldtm
    order(i,8) = 0
end
```

APPENDIXB

```
% SCHEDULING AGENT PROGRAM
f1=zeros(30,6);
f2=zeros(30,6);
f3=zeros(30,6);
l1=1;
l2=1;
l3=1;
% Read orders and seperate according to type of
finish
fori=1:30
if order(i,5)== 1
f1(l1,1)=order(i,1);
f1(l1,2)=order(i,2);
f1(l1,3)=order(i,5);
f1(l1,4)=order(i,3);
f1(l1,5)=order(i,6);
f1(l1,6)=order(i,4);
        l4=l1;
        l1=l4+1;
end
if order(i,5)== 2
f2(l2,1)=order(i,1);
f2(l2,2)=order(i,2);
f2(l2,3)=order(i,5);
f2(l2,4)=order(i,3);
f2(l2,5)=order(i,6);
f2(l2,6)=order(i,4);
        l4=l2;
        l2=l4+1;
end
if order(i,5)== 3
f3(l3,1)=order(i,1);
f3(l3,2)=order(i,2);
f3(l3,3)=order(i,5);
f3(l3,4)=order(i,3);
f3(l3,5)=order(i,6);
f3(l3,6)=order(i,4);
        l4=l3;
        l3=l4+1;
end
end
```

```
type1=f1
type2=f2
type3=f3
      sl1=0;
      sl2=0;
      sl3=0;
% Find cumulative order for each type
cum1=0;
cum2=0;
cum3=0;
rcum1=zeros(3,l1-1);
rcum2=zeros(3,l2-1);
rcum3=zeros(3,l3-1);
ul1=zeros(l1-1);
ul2=zeros(l2-1);
ul3=zeros(l3-1);
fori=1 : l1-1
    ul1(i) = f1(i,4)
end
fori=1 : l2-1
    ul2(i) = f2(i,4)
end
fori=1 : l3-1
    ul3(i) = f3(i,4)
end
sl1=sort(ul1)
sl2=sort(ul2)
sl3=sort(ul3)
fori=1:l1-1
    cum4=cum1;
    cum1=cum4 + sl1(i);
rcum1(1,i)=cum1;
end
fori=1:l2-1
    cum4=cum2;
    cum2=cum4 + sl2(i);
rcum2(1,i)=cum2;
end
fori=1:l3-1
    cum4=cum3;
    cum3=cum4 + sl3(i);
rcum3(1,i)=cum3;
end
```

APPENDIXC

```
% PRODUCTION AGENT PROGRAM

%note total number of entries in finish type1=l1-1
%note total number of entries in finish type2=l2-1
%note total number of entries in finish type3=l3-1
%cummulative order in type1=cum1
%cummulative order in type2=cum2
%cummulative order in type3=cum3

% Now decide on how many sub groups each finishing
type will be split

l1Schedule = zeros(1,l1-1);
l2Schedule = zeros(1,l2-1);
l3Schedule = zeros(1,l3-1);

l1tot = cum1;
l2tot = cum2;
l3tot = cum3;

l1Rem = cum1 - fix (l1tot/15)*15;
l2Rem = cum2 - fix (l2tot/19)*19;
l3Rem = cum3 - fix (l3tot/30)*30;

l1slack = 15 - l1Rem;
l2slack = 19 - l2Rem;
l3slack = 30 - l3Rem;

l1Sch = cum1 + l1slack; %this makes l1Sch dvisible by
15
l2Sch = cum2 + l2slack; %this makes l2Sch dvisible by
19
l3Sch = cum3 + l3slack; %this makes l3Sch dvisible by
30

l1days = l1Sch/15;
l2days = l2Sch/19;
l3days = l3Sch/30;
bc = 1;

n1x = l1days - fix(l1days/bc)*bc;
```

```
n2x = l2days - fix(l2days/bc)*bc;
n3x = l3days - fix(l3days/bc)*bc;

%n1x = 1 or 2 to bc - 1 if bc = 3
%n2x = 1 or 2 to bc - 1 if bc = 3
%n3x = 1 or 2 to bc - 1 if bc = 3

n1 = (l1days - n1x)/bc;
n2 = (l2days - n2x)/bc;
n3 = (l3days - n3x)/bc;

fori = 1:n1
    l1Schedule(i) = bc*15;
ifi == 1
rcum1(2,i) = bc*15;
else
        t1 = rcum1(2,i-1);
rcum1(2,i)=t1+bc*15;
end
end
l1Schedule(n1+1) = n1x * 15;

fori = 1:n2
    l2Schedule(i) = bc*19;
ifi == 1
rcum2(2,i) = bc*19;
else
        t2 = rcum2(2,i-1);
rcum2(2,i)=t2+bc*1;
end
end
l2Schedule(n2+1) = n2x * 19;

fori = 1:n3
    l3Schedule(i) = bc*30;
ifi == 1
rcum3(2,i) = bc*30;
else
        t3 = rcum3(2,i-1);
rcum3(2,i)=t3+bc*30;
end
end
l3Schedule(n3+1) = n3x * 30;
```

```
finish1 = l1Schedule
finish2 = l2Schedule
finish3 = l3Schedule
```

APPENDIXD

```
% RELEASING AGENT PROGRAM
% Shop floor Agent release of finished jobs
fori = 1:15
ifi == 1
rcum1(3,i) = bc;
rcum2(3,i) = bc*2;
rcum3(3,i) = bc*3;
else
        t1 = rcum1(3,i-1);
        t2 = rcum2(3,i-1);
        t3 = rcum3(3,i-1);
rcum1(3,i) = t1+bc*3;
rcum2(3,i) = t2+bc*3;
rcum3(3,i) = t3+bc*3;
end
end

rfin1 = rcum1
rfin2 = rcum2
rfin3 = rcum3
```