# Markov Chain Approach to Agent Based Modelling (ABM) of an Industrial Machine Operation and Control

**Tochukwu Chiagunye**
Computer Engineering Department,
Michael Okpara University of Agriculture, Umudike,
Abia State, Nigeria
Email: tchiagunye2@yahoo.com

***Abstract*—This research work is aimed at modeling an agent based scheduler that is optimized for handling job scheduling in a way that ensures efficient and profitable manufacturing automation. An alternative view on job shop scheduling problem (JSP) was adopted in this research work where each machine is equipped with adaptive agent that independent of other agents, makes job dispatching decision based on its local view of the plant. A combination of Markovian process and agent oriented analysis are used in the analysis of the proposed agent based model (ABM) for the industrial machine operations. The model optimization was carried out using simulated annealing technique. It provides a general framework of aggregation in agent based and related computational models by making use of Markov chain aggregation and lumpability theory in order to link the micro and the macro level of observation.**

***Keywords—Markov chain, ABM, Optimization, Operation, Scheduling, Dispatching.***

## 1.0 INTRODUCTION.

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. The call for agility challenges the shop floor with several problems. With an increasing occurrence of changes and dominating customer demand, management of manufacturing processes and the coordination of the multifarious resources, i.e., machines, materials, information, knowledge and humans, becomes a core task of shop floor scheduling and control algorithm [1].

The scheduling and planning of production order have an important role in the manufacturing system. 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, hence the traditional methods of optimization are unable to solve them [2][3].

With respect to related studies, [4] proposed a methodology for solving the job shop problem based on the decomposition of mathematical programming problems that used both Benders-type [5] and Dantzig/wolfe-type [6] 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 refine these limit times for each job by detailed sequencing of all operations. A multi-criteria objective function was specified to include tardiness, throughput, and process utilization cost. The limitations of this methodology stem from the inherent stochastic nature of job shops and 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.

[7] evaluated the use of MRP or MRP-11 to create a medium-range scheduler. MRP system’s major disadvantages are 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 job-shop schedulingg assume complete task knowledge and search for a centralized solution. These techniques typically do not scale with problems size, suffering from an exponential increase in computation time. The centralized view of the machine 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 breakdowns. 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 machine 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 machine.

## 2.0 LITERATURE REVIEW

Scheduling is an important tool for manufacturing and engineering, where it can have a major impact on the productivity of a process [8]. In manufacturing, the purpose of scheduling is to minimize the production time and cost, by telling a production facility when to make, with which staff, and on which machine. Survey of the literature indicates that the job shop scheduling problem (or job-shop problem) is at least 70 years old. In the publications [9][10][11], job shop scheduling is reported as an optimization problem in industrial engineering and operations research in which ideal jobs are assigned to resources at particular times. The most basic version is described [9] as follows:

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.

Sven et al [12] 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. This work [12] 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 Izquierds [13], 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 [14], the Axelrod model of cultural dynamics [15] and the sugar scape model from Epstein and Axtell [16]. The main idea of Izquierdo et al [13] 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 [13] mainly relies on numerical computations to estimate the stochastic transition on metrices of the models.

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 breakdown and arrival of new job. 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 machine 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 (machine) is equipped with adaptive agent that independent of other agents, makes job dispatching decision based on its local view of the machine.

## 3.0 Designing Agent-Based Model

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 [17]. 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 [17].

### 3.1 Markov Chain Approach for Agent-Based Modeling (ABM)

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. The set of possible attributes is denoted by S and is called the configuration space $\Sigma$ the set of all possible combinations of attributes of the agent, 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 [18]. Hence, ABM are Markov Chains on $\Sigma$ with a transition matrix P for a class of ABM the transition probabilities P(x,y) can be computed for any pair $x, y \epsilon$N of agent configurations. The process ($\Sigma$,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 ($\Sigma$,ṕ) 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 (protection) map $\pi$ from Σ to X. The meaning of the projection $\pi$ is to lump sets of Micro configuration in Σ according to the macro property of interest in such a way that, for each xϵX, all the configurations of Σ in $\pi^{-1}$ (x) share the same property.

### 3.2 Mathematical Model for the Makespan Objective Function

The scheduler agent's desire (goals) is to minimize the manufacturing completion time, or makespan (MS) for processing all jobs. The problem of minimizing the manufacturing makespan is equivalent to the following formulation [19].

Min Ms = f ($P_{ijk}$), Min Tmax

Subject to (i.e. the constraints):

$$B_{ij}+\sum_{k\epsilon M_{ij}} P_{ijk}X_{ijk}\le B_{i,j+1} \quad i=1,2, \ ,N;\ j=1,2,\ldots j(i)-1\ldots \quad (3.1)$$

Constraint set (4.1) ensures that an operation or activity j + 1 cannot start before the previous operation j of the same job i has been completed.

$$B_{i,J(i)} + \sum_{k\epsilon M_{i,J(i)}} P_{i,J(i)k} - MS \le 0,\ i = 1, 2,\ldots,N \quad (3.2)$$

constraint set (3.2) ensures that the starting time and processing time of the last operation J(i) for job i, i = 1, 2, ……., N is less than or equal to the makespan (MS).

$$\sum_{k\epsilon M_{ij}} X_{ijk} = 1,\ i = 1, 2,\ldots, N;\ j = 1, 2,\ldots J(i) \quad (3.3)$$

Equation (3.3) ensure that one operation j of job i can only be performed on only one machine k at a time. In essence, this constraint guarantees that each job i takes only one path through the system.

$$X_{ghk} + X_{ijk} -1 \le Y_{ijghk} + Y_{ghijk}, \quad (3.4)$$

i = 1, 2,..., N; g = 1, 2, ….,N: i ǂg; i = 1,2, …, J(i);

h = 1, 2, ….,J(g); k ϵ $M_{ij} \cap M_{gh}$.

During scheduling decision or search, the Agent use constraint set (3.4) to restrict two operations of two different jobs that are scheduled on the same machine from being performed at the same time. Thus, only one operation of one job is always performed before the other operation of the second job.

$$Yi_{jghk} + Y_{ghijk} \le 1,$$

i = 1, 2,…., N ; g = 1, 2,…., iǂg;j = 1, 2, …J(i) (3.5)

h = 1, 2, ……., J(g); k ϵ $M_{ij} \cap M_{gh}$.

$$(B_{ij} + P_{ijk} X_{ijk}) - (B_{gh} + P_{ghk} X_{ghk}) + Y\,(1-Y_{ghijk}) \ge P_{ijk} X_{ijk}$$

i = 1, 2, …….N, g = 1, 2, ……. N, iǂg; . (4.6)

j = 1, 2, …., J(g), h = 1, 2,….J(g); k ϵ $M_{ij} \cap M_{gh}$

$$(B_{gh} + P_{ghk} X_{ghk}) - (B_{ij} + P_{ijk} X_{ijk}) + Y(1-Y_{ijghk}) \ge P_{ghk} X_{ghk} \quad (3.7)$$

i = 1, 2,…N; g = 1, 2, ….,N; i ǂ g; j = 1, 2, ….J(i);

h = 1, 2, …., J(g); k ϵ $M_{ij} \cap M_{gh}$

The agents are of constraint set (3.5) in its belief guarantees that if operation j and h from jobs i and g, respectively, are to be performed on the same machine k, than the two operations cannot be performed simultaneously. Agent implements constraint set (3.6) to ensure that if operation j of job i is chosen to be processed before operation h of job g, the starting time and processing time of operation j of job i must be less than the starting time of operation h of job g. The same logic applies to constraint set (3.7) for the reverse case when operation h of job g is chosen to be processed before operation j of job i. Agent, these constraints reinforce that one job is always processed before a second job on a given machine to avoid contacts.

$$B_{i1} \ge R_i,\ i = 1, 2, \ldots\ldots N \quad (3.8)$$

Constraint set (4.8) ensures that the first operation of a job i cannot start before it is ready.

$$B_{ij} \ge 0\ \ i = 1, 2,\ldots N;\ j = 2, \ldots J(i) \quad (3.9)$$

$$MS\ge 0 \quad (3.10)$$

Using the non-negativity constraints (3.9) and (3.10) the agent ensures that all starting times for the remaining operations and the manufacturing makespan are positive.

Xijk ϵ {0, 1}, i = 1, 2, …, N; j = 1, 2, ….J(i); (3.11)

k = 1, 2, …………,m

$y_{ijghk}$ ϵ {0, 1}, i = 1, 2,…….N, j = 1, 2……J(i);(3.12)

g = 1, 2, ……….N; h = 1, 2, ……,J(g)

k = 1, 2, …….,m

Constraints (3.11) and (3.12) show the integer constraints for the 0-1 variables.

$$Bi, j(i) + \sum_{k\epsilon Mi,J(i)k}^{n} X_{i,J(i)k} = Ci, i = 1,2,\ldots N \quad (3.13)$$

$$C_i - 0_i - Tmax \le 0\ \ i = 1, 2, \ldots,N \quad (3.14)$$

$$Tmax \ge 0 \quad (3.15)$$

Constraint set (3.13) ensures that the starting time and processing time of the last operation and J(i) for job $_i$, $_i$ = 1, 2, …….N is equivalent to the manufacturing completion time, while constraint set (3.14) ensures that the tardiness of job $_i$, $_i$ = 1, 2, ……………….N is less than or equivalent to the maximum tardiness. With constraint (3.15) the scheduler agent ensures the maximum tardiness value will be non-negative.

$$I_{ik} + \sum_{i=i}^{n} X_{ijk} + {}_1P_{ijk} + B_{ij} + 1, \quad k = 1_{j,k+1} + \sum_{i=1}^{n} X_{ijk} P_{ijk+1} + B_{j,k+1} \qquad (3.16)$$

Constraint (3.16) established the relationships required to keep the consistency between machine's idle time and machine blocking times.

The scheduler agent makes scheduling decisions. This decision involves searches for time values (i.e start time of operations i on machine $M_k$) that satisfy all constraints as presented. For example the scheduler agent searching for time intervals over which two activities or operations ($O_{ij}$ $O_{i'j'}$) requiring the same resource (machine $M_k$) cannot overlap. Considering the disjunctive constraints states that either $O_{ij}$ precedes $O_{i'j}$' or $O_{i'j}$ precedes $O_{ij}$. Constraint propagation consists in determining cases where only one of the two orderings is feasible. When making scheduling decision, the scheduler agent continuously carry out constraint propagation in order to determine conditions (resulting in temporary constraints) that a schedule must satisfy (as it relates to operation ordering) to meet all the considered constraints.

This results in the scheduler activating the intentions that updates the time-bounds of each operations.

**3.3 Model for the Scheduler Optimization.**

To enable the incorporation into the shop floor agents inference (i.e intention) the stochastic effect of events at the shop floor, there has to be a way to make stochastic projections from which the scheduler agents intention (**Run scheduling Algorithm**) can optimize the schedule computation (which is more of a combinations). For the scheduler and shop floor agents deliberations (intentions) to factor in stochasticity, Markov Chain is integrated into the solution of the job scheduling problem in the proposed model.

Let jobs within the production system be at any six states.

(1) Finished state; (2) unscheduled state; (3) scheduled state

(4) Waiting state; (5) Processing state; (6) Preempted state.

This means some jobs have been completed. Some jobs may have been ordered through the order agent but not scheduled. Some jobs have been scheduled i.e. have been assigned start time on a machine $M_k$. Some jobs have been scheduled, but could not start as scheduled because they are experiencing wait for some resource (maybe $M_k$ or NC program file load or tool bit) some jobs might be currently being processed. Processing for some jobs was interrupted (i.e the job was preempted) for whatever reason.

The probabilities of these states give the initial distribution for the Markov Chain. However, the initial distribution change over time, as the entire state of the system changes, new jobs arriving, machine breakdown or becoming unavailable, more or less job experiencing waits, jobs being preempted etc.

Let state 1 be denoted by $FN_{state}$

Let state 2 be denoted by $U_{state}$

Let state 3 be denoted by $SC_{state}$

Let state 4 be denoted by $WT_{state}$

Let state 5 be denoted by $PRO_{state}$

Let state 6 be denoted by $PRM_{state}$

Let set $FN_{state}$ be the set of jobs in state 1

Let set $U_{state}$ be the set of jobs in state 2

Let set $SC_{state}$ be the set of jobs in state 3

Let set $WT_{state}$ be the set of jobs in state 4

Let set $PRO_{state}$ be the set of jobs in state 5

Let set $PRM_{state}$ be the set of jobs in state 6

Let $P_{(FN)(t)}$ be the probability of a job being in state 1 at time t

Let $P_{(US)(t)}$ be the probability of a job being in state 2 at time t

Let $P_{(SC)(t)}$ be the probability of a job being in state 3 at time t

Let $P_{(WT)(t)}$ be the probability of a job being in state 4 at time t

Let $P_{(PRO)(t)}$ be the probability of a job being in state 5 at time t

Let $P_{(PRM)(t)}$ be the probability of a job being in state 6 at time t

$FN_{state}$ is an absorbing state. In Markov Chain modeling an absorbing state is an end state. When a process enters an absorbing state it does not leave. $FN_{state}$ is an absorbing state since when a job(ij) enters state 1 it does not leave. That is it does not transition to any of the other state.

Let the set $U_{Trn}$ be defined such that at any time t

$U_{Trn}$ = {{set $U_{state}$}, {set $SC_{state}$}, {set $WT_{state}$}, {set $PRO_{state}$},{set $PRM_{state}$}}

That is;

$\cup_{Trn}$ ={set $U_{state}$} ∪ {set $SC_{state}$} ∪ {set $WT_{state}$} ∪ {set $PRO_{state}$}

or

$\cup_{Trn}$ = {∪: set $U_{state} \supset \cup_{Trn}$, set $SC_{state} \supset \cup_{Trn}$, set $WT_{state} \supset \cup_{Trn}$, set $PRO_{state} \supset \cup_{Trn}$}

Set $FN_{state}$ is not part of the universal set since state $FN_{state}$ is an absorbing state at time t.

## 4.0 **Numerical Results of the Markov Chain Algorithm.**

The discussions so far seem to be a little more abstract. Hence in order to motivate and exemplify the point of view expressed so far, consider this scenario: a time t the entire system has a stochastic pattern. At time t the shop-floor Agent evaluates its belief, the state space of job configuration (i.e. the state of all jobs at time t with their probabilities at time t), this includes updates from the **machine agent** and **order agent** arrival of new order from order agent. It evaluates the probability distribution of the job states at time t. It constitutes the probability distribution vector. It uses transition matrix relation in Chain for each state to constitute the Markov transition matrix x. It uses the Markov matrix to project the state space (i.e. the probability distribution vector) at time t, to arrive at a forecast (i.e. stochastic trend) at time t. From there it picks the **state set** that would have the most influence (i.e. highest, impact) on the schedule. It updates the **scheduler agent** belief with this from the state set picked the scheduler agent evaluates the earliest completion time of each job, the one having the smallest earliest possible completion time provided it satisfies all constraints is scheduled on the contemporary machine. The agent checks if its solution satisfies all constraints within its belief.

Assuming at time t, based on updates from the **machine agent**, and **order agent**, the **shop floor** agent evaluates the various states transition probabilities as:

$P_{(US)_{(t)}} = 0.3800$; $P_{(PRO)} = 0.3500$ . $P_{(Sc)}=0.1500$

$P_{(WT)} = 0.0800$; $P_{(PRM)} = 0.0400$

The finished state is taken here as the Markov Chain absorbing state, hence it transition probability is 1. Hence initial dynamic probability vector (i.e. initial distribution) at time t is:

$Xo = [i_1\ i_2\ i_3 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ i_\delta]$

= [1 0.3800 0.1500 0.0800 0.3500 0.04] constituting the transition matrix X

The matrix in this case should be a δ x δ square matrix i.e. 6x6 to match the state configuration first row.

[1 0 0 0 0 0]

row before last row

$[0\ 0\ P^{(N-2)}\ q^{(N-2)}\ P^{(N-2)}\ 0]$

N = number of jobs in the transient state at time t i.e. $|\cup_{Trn}|$.

Assuming at time t, scheduler agent evaluates $|\cup_{Trn}|$ as 60.

$P^{(k)} = \frac{k(N-k)}{N_2}$.

$P^{(N-2)} = P^{(58)} \frac{56(60-58)}{60^2} = 0.0322$ using equation (4k)

$q^{(N-2)} = q^{(58)} = 1 - 2x0.03 = 1 - 0.06$

= 0.9356

hence

[0 0 0 .0322 0.9356 0.0322]

last row

$[0\ 0\ 0\ P^{(N-1)}\ q^{(N-1)}\ P^{(N-1)}]$

$P^{(N-1)} = 0.0250$

$q^{(N-1)} = 0.9500$

hence

[0 0 0 0.0250 0.9500 0.0250]

Second row

$[P^{(1)}\ q^{(1)}\ P^{(1)}\ 0\ 0\ 0]$

$P^{(1)} = 0.0163$

$q^{(1)} = 0.0326$

hence

[0.0163 0.0326 0.0163 0 0 0]

Third row

$[0\ P^{(2)}\ q^{(2)}\ P^{(2)}\ 0\ 0]$

$P^{(2)} = 0.322$

$q^{(2)} = 0.9356$

hence

[0 0.0322 0.9356 0.3222 0 0]

Forth row

$[0\ 0\ P^{(k)}\ q^{(k)}\ P^{(k)}\ 0]$

Here k is the number of job at a particular state at time t. Hence the different elements of this row are needed for each of the transient states. The transition probabilities at each state and $[\cup_{Trn}]$ at time t would be used to estimate a value for k at each state. The value of $[\cup_{Trn}]$, hence k depends on the update (i.e. agent gossip) the shop floor agent receives from the order agent in terms of number of jobs being processed, being preempted etc, the machine agent in terms of number of jobs finished being preempted etc. The number k is estimated by multiplying $|\cup_{Trn}|$ and the transition probability of the particular job state at time t. In order to minimize rounding errors, the product of the multiplication is passed to the floor function. In the

**floor function**, i.e. **floor (x)** means the longest integer not greater than x. [20]

(1) For the unscheduled state ($U_{state}$)

K = 0.3800 x 60 = 22.8

Floor (22.8) = 22

Hence for USstate

$P^{(k)} = 0.2322$

$q^{(k)} = 0.5356$

Hence row four for $U_{state}$ is [0 0 0.2322 0.5356 0.2322 0]

(II) For the $PRO_{state}$ Floor (k) = floor(0.3500x60) = 21

$P^{(k)} = 0.2275$

$q^{(k)} = 0.5450$

Hence row for state $PRO_{state}$ is [0 0 0.2275 0.5450 0.2275 0]

(III) For the $SC_{state}$

Floor(k) = floor (0.1500x60) = 9

$P^{(k)} = 0.1275$

$q^{(k)} = 0.7450$

Hence row four of the transition matrix for $SC_{state}$ :

[0 0 0.1275 07450 0.1275 0]

(IV) For the $WT_{state}$

Floor(k)= floor (0.0800x60) = 4

$P^{(k)} = 0.0622$

$q^{(k)} = 0.8756$

Row four for $WT_{state}$:

(0 0 0622 0.8756 0.622 0)

(V)For $PRM_{state}$

Floor(k) = floor (0.0400x60) = 24

$P^{(k)} = 0.2400$

$q^{(k)} = 0.5200$

Row for $PRM_{state}$

[0 0 0.2400 0.5200 0.2400 0]

Transition matrix for each transient state, the following notation is used

Let transition matrix for $U_{state} = T_U$

,, ,, ,, ,, $CS_{state} = T_{CS}$

,, ,, ,, ,, $WT_{state} = T_{WT}$

,, ,, ,, ,, $PRO_{state} = T_{PRO}$

,, ,, ,, ,, $PRM_{state} = T_{PRM}$

Note the $FN_{state}$ is the absorbing state, its influence in the arriving at the overall system is not factored in. Putting rows 1,2,3,4,5 and 6 together as computed.

The transition matrix associated with state is multiplied to give the system transition matrix with which the initial state probability distribution is projected.

## 5.0. CONCLUSION

Based on the statement of the problem of this work, as it relates to the job shop scheduling problem (JSSP), the main point of consideration in the model design carried out as relates to schedule computation (scheduling algorithm) and the schedule optimization (the markov chain) is to model an agent based scheduler that is optimized for handling the job shop scheduling in a way that ensures efficient and profitable manufacturing automation. This puts the focus mainly in the scheduling agent and the shop floor control agent. The remaining agents provides support services to these two agents in the solution of the JSSP. The formulation of the proposed Agent based model for job shop scheduling problem as a Markov Chain enables the development of a mathematical framework for linking the micro-description of an Agent based model to a Macro-description of manufacturing plant.

## REFERENCES

1. Eric scherer “shop floor control – An integrative framework from static scheduling models towards an agile operations management” **, CH8092** Zurich Switzerland 2002, pp. 1,2.

2. Z. Othman, K. Subari, and N. Morad, “job shop scheduling with Alternative machines using Genetic Algorithm,” journal Tek nolosi, 41 (0). 2007.pp. 67-78

3. K. Raya, R. Saravaran, and V. Selladurai,” 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

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

5. Benders, J. “Partitioning procedures for solving mixed-variables mathematical programming problems” Number sche mathematik, 4(3): 238-252.

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

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

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

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

10. Z Othman, K. Subari, and N. Morad," job shop scheduling with alternative machines using Genetic Algorithms," journal Teknolosi, 41(D). 2007.PP.67-78.

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

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

13. 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.

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

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

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

17. Grimm, V.,U. Berger, F. Bastiansen, S. Eliassen, V. Ginot, J. Giske, J. Goss-Custard, T. Grand, S.K.Hein Z,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.

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

19. Ten Noranis Mohd Aris "Object-oriented programming semantics representation utilizing agents" journal of theoretical and applied information technology sep. 2011.vol. 3, No. 1 Pp. 10, 11, 12.

20. Baptists, Ph. LePape, C, & Nuijten, W. (1999) "Satisfiability test and time – Bound Adjustment for cumulative scheduling problems "Annals of Operations Research pp, 92, 305-333.