International Journal of Trend in Scientific Research and Development (IJTSRD)

International Open Access Journal

ISSN No: 2456 - 6470 | www.ijtsrd.com | Volume - 2 | Issue – 1

Agent-Based Modeling: Methods and Techniques for Scheduling Industrial Machine Operation

Chiagunye Tochukwu
Department of Computer Engineering, Michael Okpara University of Agriculture, Umudike

Inyiama Hyacinth
Department of Electronic and Computer Engineering, Nnamdi Azikiwe University, Awka

Aguodoh Patrick
Department of Computer Engineerin, Michael Okpara University of Agriculture, Umudike

ABSTRACT

Owing to its impact on the industrial economy, job shop scheduler and controller are vital algorithms for modern manufacturing processes. 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). 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. A combination of Markov Chain instruments and agent oriented analysis is 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 the ABM. 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. Simulated annealing technique is used for carrying out the optimization modeling for the ABM.

Keywords: *Agent-Based, Scheduling, Industry, Machine Operation, Markov Chain*

1.1 Introduction

In the United States alone, there are over 40,000 factories producing metal-fabricated parts [1]. 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 [2]. Manufacturing serves as a basic function for any agile enterprise. 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 [2].

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 [3]. As defined by Baker [4] scheduling is concerned with the problem of assigning a set of jobs to resources over a period of time. Effective scheduling plays a very important role in today's competitive manufacturing environment. 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 [5]. Hence, it becomes increasingly important to develop effective scheduling approaches that help in achieving the desired objectives.

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 more complicated, hence the traditional methods of optimization are unable to solve them [6][7].

As can be deduced from these techniques, most approaches to job-shop scheduling 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 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 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 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.

2.0 Literature Review

Scheduling, understood to be an important tool for manufacturing and engineering, has a major impact on productivity of a process [8]. 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. [9] contributed to interweaving Markov Chains and agent-based modeling. [10] worked on and represented ten well-known simulation models as a time homogenous Markov Chain. The author's main idea is the formulation of the system configuration as the state space of the Markov Chain.

[11] formulated the JSP using integer programming. This technique involved assuming that each job consist 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, [12] 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.

Furthermore, although efficient bonding and pruning procedures have been developed to speed up the search, this is still a very computational intensive procedure for solving large scheduling problem. [13] proposed a methodology for solving the job shop problem based on the decomposition of mathematical programming problems that used both Benders-type [14] and Dantzig/wolfe-type [15] 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 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.

[16] evaluated the use of 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. 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). 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.

3.0 Research methodology:

A combination of Markov Chain instruments and agent oriented analysis is 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 the ABM. 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.

The starting point is a Microscopic Markov Chain description of the dynamical process in complete correspondence with the dynamic behavior of the agent model which is obtained by considering the set of all possible agent configurations as the state space of a huge Markov Chain. This is referred to as micro chain. The explicit micro formulation enables the application of the theory of Markov Chain aggregation-namely, lumpability-in order to reduce the state space of the micro chain and relate microscopic deceptions to a macroscopic formulation of interest.

Simulated annealing technique is used for carrying out the optimization modeling for the ABM. Some of the conditions for asymptotic convergence, as for instance, the infinite length of Markov Chains, cannot be met in practice [1]. Hence, in any finite implementation, choice have to be made with respect to the parameter:

- 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 or an annealing schedule [2], hence the use of simulated annealing for the global optimization of the ABM for the proposed JSP.

3.2 Proposed System

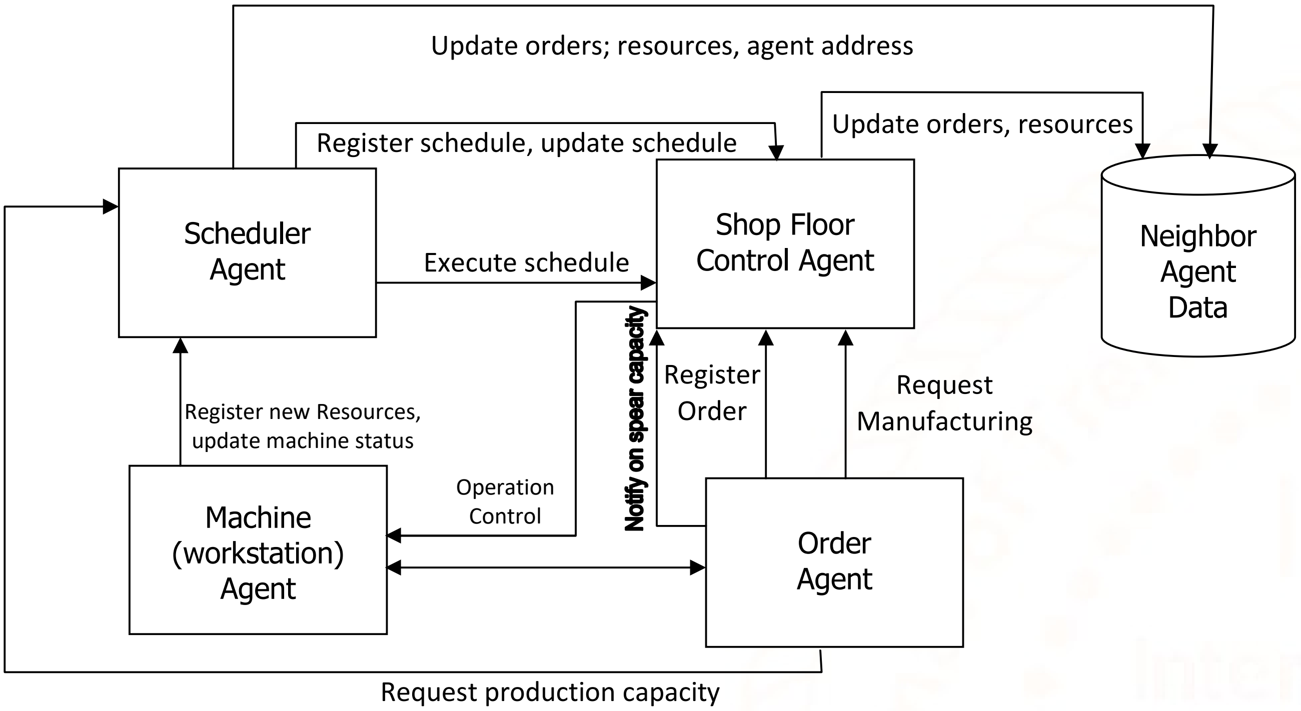

Fig. 1: Block Diagram of the Proposed ABM Framework for Solving the JSP

Fig. 1 presents the shop for scheduler agent, shop floor control (SFC) agent, order agent and the machine (or work station) agent. This figure shows the interaction between the scheduler agent and other agents.

First, the agents mentioned above needs a model of the surrounding agents. Agents have to register themselves with the SFC agent. Second, resources like a machine (or workstation) need to notify the shop control agent when they have spear productivity capability. The order agent sends request for production capacity (to the scheduler agent or SFC agent). The function of the shop floor control agent is to match the request of the order agent with the offers of the resources on a certain instance of time. The SFC agent therefore uses the schedules it receives regularly from one or more schedules. Order agents and resources (or workstation) are not obliged to wait with their request and offers until the operation is executable or the resource is free. This enables the agents to foresee future events and consider the consequences of it. This will be used to have, e.g. an idle machine (workstation), waiting for an important order, even when work is available.

In addition to the global scheduler, each machine (workstation) is equipped with a local scheduling agent that makes decision on which job to process next based solely on its partial view of the plant. For executing a schedule, the machine (workstation) agent needs to register itself with the shop floor scheduler agent. Further on, the machine agent shall inform the scheduler agent and SFC agent when it is available, occupied, or unavailable in the (near) future. When it becomes available again, it notifies this event to the scheduler agent. The scheduler agent can explicitly ask for the availability of the machine. The SFC agent, considering all available information, will allocate a machine (workstation) to a task. It will inform both the machine agent and the order agent requesting that task. The order agent and machine agent will then autonomously settle the details of the task to be executed. This includes setting up the machine i.e. the workstation, co-ordinating their activities in time, loading the necessary auxilliary resources and exchanging information like NC-programs. The machine (workstation), in principle, has the possibility to schedule the tasks. This can be done to account for small durations in the predicted processing time; to react to resource breakdown; or to optimize setups that have not been foreseen in sufficient detail on higher level.

4.0 Agent Structure Modeling

In this work, the Belief - Desire – Intention (**BDI**) agent model is adopted to model the system. The BDI software model is one of the software models developed for modeling and programming intelligent agents [17].

In BDI terms Beliefs represent knowledge of the world. It represents information state of the agent. In computational terms, Beliefs represent the state of the world, such as the value of a variable (i.e. constraint variable in the proposed job shop scheduling problem (JSSP) model), a mathematical or logical expression (in this design constraint inqualities, **JSSP** constraint expressions, and scheduling constraint equations), a relational database and belief includes inference rules used by the agents that allow forward chaining that leads to new beliefs.

Desire (or goals) form another essential component of the system state. Desire represents the motivational state of the agent, objectives or goal that the agent would like to accomplish [17]. In computational terms, a desire may simply be the value of a variable, a record structure, or a symbolic expression in some logic. The important point is that a goal represents some end state.

Intentions represent the deliberative state of the agent, the agents chosen actions, which means the agent has begun executing a plan. Computationally, intentions may simple be a set of executing threads in a process that achieve the goals (desires) of the system.

4.1. Specifying the BDI components of the proposed agent based model.

Referring to the proposed system block diagram and narrative of section 3.2:

➢ **Scheduler Agent BDI structure**

❖ **Belief**

The agent has the following belief;

- **Scheduling Constraints** (Resource Constraints, task constraints, temporary constraints):

This constraint, are specified in this model design in form of mathematical expressions, inequalities, logical expressions that the computed schedule parameter must satisfy.To this effect the scheduler agent, uses combinatorics (i.e a search) to generate scheduling values (start time for jobs, and machine) that ensures that the set constraints are satisfied. It uses constraint propagation computation to adjust generated values.

- **Data on available machines (i.e. available resources):**

The machine agent from time to time updates the scheduler agent on the availability of machines (in other words data on free machines).

- **Request Order:**

A data structure that list and stores request received from the order agent.

- **Job state space probability vector**:

This is data of stochastic projection received from the shop floor agent. The shop floor agent uses Markov chain to arrive at this projection.

❖ **Desire**

The scheduler agent has the desire

- **Makespan Minimization**:

The objective (i.e desired end state) of this agent is to apply it's scheduling heuristics in order to minimize the manufacturing makespan.

- **Assignment of jobs (unscheduled jobs, preempted jobs) to machines:**

The agents want to arrive at the state whereby all unscheduled jobs, pre empted jobs, waiting jobs are completely assigned to machines.

- **Constraint Satisfaction:**

The agent objective is for all mathematical constraint, that would be presented in detail in this chapter later are satisfied for whatever value of variable (e.g. start time, processing time etc) are computed by the scheduler algorithm as a possible solution to the JSSP.

❖ **Intention**

The scheduler agent carries out the following action (intention); these actions are methods or threads in software systems. This explains why the representation is in form of program function (methods).

- **Run scheduling Algorithm ()**

The scheduler agent runs the algorithm that schedules (i.e. assigns) jobs to machines. This algorithm is specified in this chapter.

- **Probe Resource Availability ()**

The scheduler agents continuously run the code to get information on available resource (machines) from the order agent and machine agent. This is necessary for it to compute the next schedule, to update the current schedule or discard the current schedule.

- **Send schedule to shop floor Agent ()**

This scheduler agent intention computes and sends schedule to the shop floor agent. It is the shop floor agent that executes the schedule.

- **Execute Constraint Propagation ()**

This intention is specified to run verification code by evaluating the various constraints that have been specified for the JSSP. These constraints are given as mathematical expression in this design. By evaluating constraint satisfaction for any possible solution for the JSSP, the agent constructs new constraints (temporary constraints) and updates its belief.

➢ **Shop floor control Agent BDI structure**

❖ **Beliefs**

The agent has the following beliefs

- **Machine Status**

It gets knowledge (i.e data) on the status of machines (idle, breakdown etc) from the machine agent. It gets this knowledge in form of updates.

- **Manufacturing Schedule**

It gets knowledge of completed schedules from the scheduler agent. The schedule is the data on the assignment of jobs to machines.

- **Job Status**

This is the states of all jobs at time t. It has knowledge of the state of all jobs in the system at any time. The status of the job could be: Unscheduled, scheduled, waiting, processing, preempted (i.e stop or interrupted), finished. It constitutes this belief as its state space (Σ) at time t.

- **Manufacturing Request**

The encoded instruction from the order agent on the details of requested manufacturing. What sequence of operations should be carried out on a job, the sequence of machining required to deliver the job.

- **Order Request**

A registry data structure that constitutes a listing of order received.

❖ **Desires**

The agent desires:

- **Global optimization (Markov chain)**

The agent wants to keep the schedule optimized at every time t. it uses the Markov chain to project a forecast state space that enables the schedule agent to adopt its Runscheduling Algorithm() intentions.
This is a key part of the proposed system.

- **Job and Machine (Resource) Alignment**

This agent wants to actualize the objective of matching job(ij) with the right machine. Its Execute Schedule () intentions is the action it carries out to realize this objective.

- **Keep Scheduler Agent Update on Job State Transient**

This desire is from the perspective of Markov Chain. The agent wants to let the schedule agent to be aware of the probability distribution of the non-absorbing states of the systems at every scheduling loop or scheduling window or upon request from the scheduler agent.

❖ **Intentions**

The shop floor control agent carries out the following actions (operations or plan execution):

- **Initiate Markov Chain ()**

The agent uses the job status updates from the order agent to constitute the job state space vector at time t, it evaluates the mathematics to obtain the Markov transition matrix. It uses the transition matrix to project the initial system state space to incorporate stochasticity in the schedule computation of the schedule agent. The outcome of this intention gives the schedule agent the initial distribution (the highest probability likelihood) from which the schedule is computed.

- **Execute Schedule ()**

It executes the schedule from the schedule agent. The schedule gives this agent the required instructions on the allocation of operations to machines. The execution of the schedule enables it to pass control instructions to the machine agent.

- **Run Update on Allocation ()**

It updates the scheduler agent belief, the machine agent belief and the order agent belief on machine (i.e resource) allocation. It also updates the scheduler agent on job state space at time t.

4.2 Modeling the Scheduling Constraints

The core of the algorithm is the agent BDI computing of the schedule and Markov process. In a continuous and iterative process, the agent based modeling is used to compute and track the state of jobs to produce initial state distribution which is then used to initialize the Markov Chain to produce the optimized target schedule. The scheduler agent uses constraints programming technique to produce the schedule.

Most scheduling problems can easily be represented as instances of the constraints satisfaction problem [18]: given a set of variables, a set of possible values (domain) for each variable, and a set of constraints between the variables, assign a value to each variable, so that all the constraints are satisfied. Hence the principle observed by the Scheduler Agent is that given a set of resources (i.e machines) with given capacities, a set of operations or activities (to be carried out at the machines) with given processing times and resource, requirements, and a set of temporal constraints between activities, the scheduling problem consists of deciding when to execute each activity or operations O, so that both temporal constraints and resource constraints are satisfied.

- **Makespan Objective Function**

The scheduler agent's desire (goals) is to:
Minimize the manufacturing completion time, or makespan (MS) for processing all jobs. Mathematically, the problem of minimizing the manufacturing makespan is equivalent to the following formulation (adopted from mixed integer programming) [17]

$$\text{Min Ms} = f\,(P_{ijk}),\ \text{Min } T_{max}$$

Subject to (i.e. the constraints):

$$B_{ij} + \Sigma\, P_{ijk}\, X_{ijk} \leq B_{i,\,j+1} \quad i=1,2,....N;\ j = 1, 2, ...j(i) -1 \text{ .... } (4.1)$$

$$K \in M_{ij}$$

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.

$$B_{i,J(i)} + \Sigma \quad P_{i,J(i)k} - MS \leq 0,\ i = 1, 2, \text{........}, N \text{ ....}(4.2)$$

$$K \in M_{i,J(i)}$$

constraint set (4.2) 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).

$$\Sigma\ \ X_{ijk} = 1,\ i = 1, 2,......, N;\ j = 1, 2, ... J(i) \text{ .... } (4.3)$$

$$K \in M_{ij}$$

Equation (4-3) 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.

$$X_{ghk} + X_{ijk} -1 \leq Y_{ijghk} \ +\ Y_{ghijk}, \text{ .... } (4.4)$$

$i = 1, 2, \ldots\ldots\ldots.., N;\ g = 1, 2, \ldots\ldots.,N:\ i‡g;\ j = 1,2, \ldots\ldots, J(i);$

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

During scheduling decision or search, the Agent uses constraint set (4.4) 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.

$$Y_{ijghk} + Y_{ghijk} \leq 1,$$

$i = 1, 2, \ldots\ldots., N ;\ g = 1, 2, \ldots\ldots.. ,\ i‡g;\ j = 1, 2, \ldots\ldots J(i)\ \ldots\ldots\ldots..(4.5)$

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

$$(B_{ij} + P_{ijk} X_{ijk}) – (B_{gh} + P_{ghk} X_{ghk}) + Y (1\text{-}Y_{ghijk})\geq P_{ijk} X_{ijk}$$

$i = 1, 2, \ldots\ldots.N,\ g = 1, 2, \ldots\ldots.. N,\ i‡g;\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots..\ (4.6)$

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

$$(B_{gh} + P_{ghk} X_{ghk}) – (B_{ij} + P_{ijk} X_{ijk}) +Y(1\text{-}Y_{ijghk})\geq P_{ghk} X_{ghk}\ \ldots\ldots.(4.7)$$

$i = 1, 2,\ldots\ldots\ldots N;\ g = 1, 2, \ldots\ldots.,N;\ i ‡ g;\ j = 1, 2, \ldots\ldots.J(i);$

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

The agents of constraint set (4.5) 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, then the two operations cannot be performed simultaneously. Agent implements constraint set (4.6) 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 (4.7) 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.

$B_{i1} \geq R_i,\ \ i = 1, 2, \ldots\ldots\ldots\ldots N\ \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..\ (4.8)$

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

$B_{ij} \geq 0\ \ \ i = 1, 2,\ldots\ldots\ldots.. N;\ j = 2, \ldots\ldots\ldots J(i)\ \ldots\ldots\ldots\ldots\ldots..\ (4.9)$

$MS \geq 0\ \ \ \ldots\ldots\ldots\ldots\ldots..\ (4.10)$

Using the non-negativity constraints (4.9) and (4.10) the agent ensures that all starting times for the remaining operations and the manufacturing makespan are positive.

$Xijk \in \{0, 1\}, i = 1, 2, \ldots\ldots\ldots, N; j = 1, 2, \ldots\ldots\ldots J(i);$ ……(4.11)
$k = 1, 2, \ldots\ldots\ldots\ldots, m$
$Y_{ijghk} \in \{0, 1\}, i = 1, 2, \ldots\ldots\ldots\ldots N, j = 1, 2 \ldots\ldots\ldots\ldots J(i);$
$g = 1, 2, \ldots\ldots\ldots\ldots\ldots N; h = 1, 2, \ldots\ldots, J(g)$
$k = 1, 2, \ldots\ldots\ldots\ldots\ldots, m$

Constraints (4.11) and (4.12) show the integer constraints for the 0-1 variables.

$$\mathrm{Bi, j(i)} + \sum_{k \epsilon M i, J(i) k}^{n} \mathrm{X}_{\mathrm{i, J(i)k}} = \mathrm{Ci}, \qquad ¡ = 1, 2, \ldots\ldots \mathrm{N} \ldots\ldots (4.13)$$

$C_i - 0_i - Tmax \leq 0 \quad i = 1, 2, \ldots\ldots\ldots, N$ …………………… (4.14)
$Tmax \geq 0$ ………………………………………(4.15)

Constraint set (4.13) ensures that the starting time and processing time of the last operation J(i) for job i, i = 1, 2, …….N is equivalent to the manufacturing completion time, while constraint set (4.14) ensures that the tardiness of job i, i = 1, 2, ………………N is less than or equivalent to the maximum tardiness. With constraint (4.15) the scheduler agent ensures the maximum tardiness value will be non negative.

$$\mathrm{I}_{\mathrm{ik}} + \sum_{i=i}^{\mathrm{n}} \mathrm{X}_{\mathrm{ijk}} + {}_{1}\mathrm{P}_{\mathrm{ijk}} + \mathrm{B}_{\mathrm{ij}} + 1, \mathrm{k} = \mathrm{I}_{\mathrm{j},}\mathrm{k} + 1 + \sum_{i=1}^{n} \mathrm{X}_{\mathrm{ijk}} \mathrm{P}_{\mathrm{ijk}} + 1 + \mathrm{B}_{\mathrm{j},}\mathrm{k} + 1 \qquad \ldots\ldots\ldots\ldots\ldots\ldots (4.16)$$

Constraint (4.16) 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.

5.1 Conclusion

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 of the two operations. The algorithm specified in this work for the scheduler agent is based on the earliest time heuristics [19]. 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 combinatorics). For the scheduler and shop floor agents' deliberations (intentions) to factor in stochasticity, Markov Chain is integrated into the solution of the JSSP in the proposed model. 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.

REFERENCES

1. Albert Jones, Luis C. Rabelo "Survey of job shop scheduling techniques" MIT, USA (2009), pp 232
2. Eric scherer "shop floor control – An integrative framework from static scheduling models towards an agile operations management" , **CH8092** Zurich Switzerland 2002, pp. 1,2.
3. Philippe Baptiste, claude le pepe and Wim Nuijten. Constraint-Based optimization and

Approximation for job-shop scheduling" ILOG S.A., France (1995).

4. Baker, K.R [1974]. "Introduction to sequencing and scheduling, Wiley & sons.
5. Akeela M.Al-Atroshi, Sama T. Azez, Baydaa S. Bhnam An Effective Genetic Algorithm for job shop scheduling with fuzzy Degree of satisfaction," University of Dohuk, Irag (2013), pp. 180
6. 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
7. 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
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. 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.
10. 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.
11. C. Low, C.M. Hsu and K.I. Huang, " Lot sputting in job shop scheduling using integer programming". International Journal of advanced manufacturing technology, vol. 24, pp773-780, 2004.
12. U. Buscher and L Shen. "An integrated tabu search algorithm for the lot streaming problem in job shops" European journal of operational research, vol. 19 pp 385-399, 2009.
13. Davis, W. and A. Jones (2001), " A real-time production scheduler for a stochastic manufacturing environment". 1(2): 101-112.
14. Benders, J. "Partitioning procedures for solving mixed-variables mathematical programming problems" Number sche mathematik, 4(3): 238-252.
15. Dantzig, G. and P. walfe "Decomposition principles for linear programs", 8(1): 101-111
16. Bauer, A, Bowden, R., Brown, I. Duggen J en Lyons, G., "shop floor control systems, design and implementation, "chapman & Hall, London, 1991.
17. 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.
18. Nuijten, W. P. M & Aarts, E. H. L. (1994) "Constraint satisfaction for multiple capacity job-shop scheduling" proc. 11th European, Conference or Artificial Intelligence, pp 635-639. John Wiley and Sons.
19. 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.