An Agent Neural Network-based Automated Guided Vehicle Scheduling, Routing and Control for Flexible Manufacturing System

Henrietta Udeani[1], Hyacinth Inyiama[2] and *Tochukwu Chiagunye[1]*
Computer Engineering Department Michael Okpara University of Agriculture, Umudike.[1]
Electronic and Computer Engineering Department, Nnamdi Azikiwe University Awka.[2]

***Abstract*—To manage the increasing complexity and to shorten design time, industry is forced to consider system-level specification and design methods that enable analyzing design alternatives in an early phase of the design process. In this work, a flexible manufacturing system (FMS) in which material transfer between machines is performed by a number of identical Automated Guided Vehicle (AGVs) is considered and the problem of routing and scheduling the AGVs and machines is addressed using Agent-neural network technique. It focuses on the intersection of agent technology and machine learning (artificial neural network) techniques for producing intelligent agents. This means more intelligent AGVs able to flexibly adapt to changes in a working environment while performing the material handling task. Given a number of AGVs and a set of transportation request, the main problem tackled by this research is to frame an agent-neural network algorithm for scheduling and routing control of AGVs with a view to avoiding collisions and deadlocks and achieving minimum journey motion times in FMS.**

Keywords—Flexible, Scheduling, Manufacturing, Automation, Agent, Machine.

1.0 INTRODUCTION.

To achieve flexibility in manufacturing systems, the concept of an autonomous decentralized flexible manufacturing system (AD-FMS) is very vital. The flexible manufacturing, system (FMS) is an advanced production system in which different standards may be used and different product types can be produced in the same line, and is controlled by computers and equipped with a material handling (transportation) system that will deliver any work piece to any machine in any sequence [1][2][3][4]. Flexible manufacturing systems aim to combine the productivity of flow lines with the flexibility of job shops to attain very versatile manufacturing units achieving high operational efficiencies. They are particularly designed for low volume, high variety manufacturing, and good decision making and management are crucial to maximize the benefit that they offer. An FMS consists of a set of cells, material handling system (Automated Guided Vehicles), service centers, etc. Automated Guided Vehicles (AGVs) are the most flexible means to transport materials among workstations of a flexible manufacturing system [5].

The major difference between an FMS and a conventional job shop is that the human tasks are automated in the FMS. In an FMS, an AGV functions as an unmanned, computerized system that is capable of undertaking external guidance to destination [6]. The advantages of the AGV system include improvements in flexibility, space utilization, safety and overall operating cost of the FMS. The AGV systems are highly flexible, since their route can usually be changed quickly, and vehicles can be dynamically rerouted. Regarding space utilization, the AGV does not create physical barriers on the factory floor as conveyors do, and they can also share aisle space with other users.

In comparison with conventional manufacturing system, FMS, are more efficient (faster production, less human involvement), more flexible (lower setup and change over times), and have a higher level of integration (more than one operation can be performed on flexible automated machines). However, for these advantages to be fully realized, certain vital issues related to the FMS have to be addressed. In order to gain the desired objective, the planning of the FMS decision making is crucial because it influences subsequent decision planning such as scheduling and control. Accordingly, one of the major problems encountered in the FMS is the AGV scheduling and control. An effective AGV controller is needed to monitor the equipment status and regulate work piece movement, so that the right material can be moved to the right place at the right time which is the basis of just in time (JIT) philosophy [7]. Furthermore, since the concept of FMS has been recognized to imply the ability to accommodate change, it is an essential aspect in FMS to adopt flexibility. The shortcomings of the above mentioned FMS AGV scheduling and control approaches indicate that a new generation of AGV is necessary to meet the dynamism required in current FMSs. There is the necessity for more intelligent vehicles able to tackle problems being imposed with the demand for more effective, reliable, faster, collision and deadlock free and efficient AGVs required for today's FMS. This means more intelligent AGVs able to flexibly adapt to changes in a working environment while performing the material handling task.

2.0 LITERATURE REVIEW

Daniels [8] first introduced an algorithm to route AGVs in a bidirectional flow path network, in which the PSP algorithm [9] is applied to find the shortest path for an AGV. The correctness and feasibility (in terms of time/space requirements) off the algorithm are theoretically proven. The algorithm can find a conflict and shortest time route for a newly added AGV without changing the existing routes of the other vehicles the computational complexity of finding a route for every AGV is O (n x a), where n is the number of nodes and a is the number of areas in the path network (areas are the path segments and nodes and P/D stations or junctions of paths). The limitation of this algorithm is that it does not allow vehicles to use path resource that could otherwise be shared during different time-window. Consequently, sometimes the algorithm may not find a path even if there exits one for a vehicle. Hence, the algorithm is only suitable for system with a small path network and a small number of AGVs.

Huang et al [10] proposed a labeling algorithm to find a shortest time path for routing a single AGV in a FMS in a bidirectional path network. A graph G is obtained from a given path network by representing each physical path segment as a node in G; two nodes in G are linked if and only if the corresponding path segments are adjacent to each other. By comparing the labels of every node, a shortest time path could be obtained if it exists. The main disadvantage of the algorithm as reported is the unacceptably large amount of computation.

Kim and Tanchoco [11] also presented a conflict-free and shortest time algorithm for routing AGVs in a FMS bidirectional path network. Their algorithm is based on Dijkstra's shortest path algorithm. The algorithm has the disadvantage that is takes a large amount of time to get an optimal path specifically; it requires O (V^4 n^2) computation in the worst case, were V is the number of vehicles and n is the number of nodes. Therefore, it will be more suitable for a small system with few vehicles.

Potts and White head [12] derived a three phase integer programming model to solve the combined scheduling and machine layout problems in a FMS. This research model was applied to a proposed FMS where tow plastic products (e.g a chemical badge and a microchip box) were to be manufactured with ten distinct operation. Through this algorithm showed some improvement in FMS layout solutions, if has the shortening of not maximizing throughout by balancing workloads and minimizing the movement of work between machines.

Goetz and Egbelu [13] the modeled and solved the noble of AGV routing and scheduling in a FMS as well as the location of pick-up/drop (P/D) stations an integer linear programming problem. The objective is to minimize the total distance travelled by both loaded and unloaded AGVs. A heuristic algorithm is used in the study to reduce the size of the problem. The limitation of the solution is that the path studied was unidirectional, which to some extent results in low path utilization [14].

Co et al [15] formulated a O-l MLP model to address FMS batching, machine scheduling and loading, and tool magazine configuration problems, simultaneously. Since this formulation contained a large number of variables, they realized that the model would not be useful in actual applications as it is NP-hond. On the other hand, the MILP model provides a useful structure that allows for the continuation of FMS research. One to the complexity of the model, a four-pass heuristic was developed that used sub-models of the original MILP model. Through computational results show that the heuristic approach was able to find the optimum values much faster than first the original MILP model. It does a poor job of managing uncertainties and is not suitable for global optimization of AGV scheduling and routing in the FMS.

Gaskins and Tanchoco [16] first formulated the path layout problem as a O-I inter programming model with consideration of the given facility layout and P/D stations. The objective is to find an optimal path network which will minimize the total travelling distance of loaded vehicles. However, the paper only considers unidirectional path network, which has lower utilization than bidirectional ones do [17]. The distance travelled by unloaded vehicles is not taken into consideration, which may affect the routing control and system through put. The main limitation of the study is that it only considers a fleet of AGVs with the same origin and destruction every time. These AGVs run along the same route. Therefore, in this case routing control is trivialized since issues such as congestion, deadlocks and conflicts will never occur. For the formulated O-I integer programming model here, another limitation is that it has a low probability of obtaining a non-empty solution set. Moreover, for practical problems the numbers of O-I variables needed for the model tend to be very large and computational efficiency becomes a critical issue.

Most of the FMS control algorithm as hinted above treat the routing problem as a shortest path problem in graph theory, to find a route for a vehicle, the algorithm usually have to search for every node and are of the path network graph [17]. Some of the algorithm may also miss the optimal solutions because certain constraints are overly enforced. The complexities of some of the algorithms especially O-I MILP and the O-I integer programming makes them unviable for real-time shop floor conditions including the inability to adapt to uncertain shop floor conditions like AGVs or CNC machines breaking down unexpectedly. Consequently the need arises for more intelligent algorithm for automating a FMS material handling system. Hence this work proposes more intelligent digital control algorithm for the FMS based on the combination of intelligent agent and neural network technologies for optimal AGV routing and scheduling.

3.0 Materials and Mothods

A flexible manufacturing system (FMS) in which material transfer between machines is performed by a number of identical Automated Guided Vehicle (AGUs) is considered and the problem of routing and scheduling the AGVs and machines is addressed using Agent-neural network technique.

The methodology focuses on the intersection of agent technology and machine learning (artificial neural network) techniques for producing intelligent agents.

Algorithms used

A neural network technique is used to improve the reasoning mechanism of the agents, supplying to the agent, a new behavior which it did not possess from the beginning. This machine learning algorithms (the Artificial Neural Network ANN) allow for the agents to adequately respond to changes at the FMS shop floor and improve the behavioral rules.

Evolutionary programming is used for developing the ANN. To meet this goal particle swarm optimization (PSO) is used for the training of the neural network.

Note: The most widely used method of training ANN is back propagation (BP) algorithm. Similar to genetic algorithm (GA), the PSO algorithm is a global algorithm, which has a strong ability to find a global optimistic result. The BP algorithm, on the contrary, has a strong ability to find local optimistic result, but its ability to find global optimistic result is weak [18]. In addition, the efficiency of BP method depends on the selection of appropriate learning parameters. In the other hand PSO acceleration draining speed of the ANN single its search process starts from initiating a group of particles. Particle swarm optimization is a stochastically global optimization method that belongs to the family of swarm intelligence and Artificial life [19] [20].

Neural network technique is used for the generation of the agent rules-base. Agent oriented software engineering methodology is exploited for the analysis and design of the intelligent agents.

FMS description

The zone blocking technique [21] [22] is used for the layout of the FMS. The zone blocking technique permits only one vehicle in a given path segment at a time. It is suitable for multiple AGVs under centralized traffic control. In such a system, the guide path is divided into a set of many small path segments; these segments are termed as a link of the network. The link from node X is one of its adjacent nodes Y is denoted as (X, Y). For this link, node X is said to be adjacent to node Y and vice versa. For each pair of adjacent nodes, link (X, Y) is aid to be unique.

This physical layout of the FMs shop floor is presented in chapter five of this report.

Assumptions

In this work, some implicit assumptions are adopted to map the FMs shop-floor Scenario: AGVs can start and stop only at nodes. In order to avoid collision, temporary stay at some nodes is permitted. A spin turn of AGV on a guide path is assumed to be avoided. It is also assumed that a vehicle path cannot contain any loops or partial paths, whose start node is the same as its goal node. A trapezoidal velocity profile is used, and the maximum speed for each profile is fixed at its maximum vehicle speed V_{max} multiplied by velocity parameter $Л_{xy}$ (O ≤ $Л_{xy}$≤1) that is assigned to each link. Acceleration and deceleration of AGVs are assumed to be constant and can be denoted as ∂ac and ∂dec.

3.1 Description of the basic 0-1 MILP model for AGV routing and scheduling control.

The MILP model was developed for routing and sequencing a set of N jobs over a limited set of M machines in an FMS. The following is a listing of the subscripts, variables and parameters used in the MILP algorithm. Parameters such as processing time are known before hand and have been randomly generated or extracted from given manufacturing process plans.

Subscripts:

i or g = 1,2, …, N, index for a job in FMS, where N is the total number of jobs.

j or h = 1,2,…,j (i), index for processing operations in the FMS, where j(i) is the final operation of job i, i = 1,2, …, N.

k = 1,2,…, M, index for a machine in the FMS, where M is the total number of machines.

Mij (or Mgh) = set of valid optional machines for operating of job i, j = 1,2,…j = 1, 2, …, j(i); i = 1, 2, …N. for example, M_{32} = {1,3} denotes that the second operation of job 3 can be performed on either machine 1 or machine 3.

3.2 Variables and parameters:

Bij = manufacturing starting time of operation j of job i, j = 1, 2,… j(i); ii = 1, 2, …, N

Pijk = manufacturing processing time required for operating j of job I

on machine k.

j = 1, 2,…, j(i); i = 1, 2,…, N; k = 1, 2,… M.Yγ = large positive integer value used in the disjunctive constraints which help to order jobs which use the same machine into a specific sequential order.

X_{ijk} = {1, if operation j of job I is performed on machine k, 0,

Otherwise}.

K € M_{ij}, M_{ij} : i = 1, 2,…, N; j = 1, 2,…, j(i)

Y_{ijghk} ={1, if operation j of job I is performed before operation h of

job g on the same machine k, 0, otherwise}.

Ii ≠ g; k € M_{ij} n M_{gh}, M_{ij} : i = 1, 2,…, N; j = 1, 2,…, j(i); M_{gh} :

g = 1, 2,…, N;

h = 1, 2,…, j(g).

R_i = ready time of job i

MS = max {C_i : i = 1, 2,…,N}, make span is defined as the maximum completion time for all jobs.

Make span Objective Function:

The objective is to minimize the manufacturing completion time or make span (MS) for processing all jobs of a batch (or an order). Mathematically, the problem of minimizing the manufacturings make span is equivalent to the following formulation.

Min Ms = f(P_{ijk}) (3.1)

Constraints:

$B_{ij} + \sum P_{ijk} X_{ijk} \leq$ Bi, j + 1, (3.2)

K € M_{ij}i = 1, 2,…, N : j = 1, 2,…j(i)

Constraint set (3.2) ensures that an operation j+1 cannot start before the previous operation j of the same job i has been completed.

$B_{i,j(i)} + \sum P_{ij(i)k} X_{ij(i)k} - Ms \leq 0$, i= 1,2, …N (3.3)

K € M_{ij}

Constraint set (3.3) 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 make span (Ms).

$\sum X_{ijk} = 1$, i = 1,2,…,N; j = 1,2,…, j(i). (3.4)

K € M_{ij}

Equation (3.4) ensures that one operation j of job i can only be performed on one machine k at a time. In essence, this constraint generates that each job i takes only one path through the system.

$X_{ghk} + X_{ijk} - 1 \leq Y_{ijghk} + Y_{ghijk}$ (3.5)

i = 1,2,…,N; g = 1,2,…,N; ii ≠ g; j = 1,2,…,j(i);

h = 1,2,…, j(g); K € M_{ij} n M_{gh}

constraints set 3.5 restricts 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.

$Y_{ijghk} + Y_{ghijk} \leq 1$. (3.6)

i = 1,2,…,N; g = 1,2,…,N; i ≠ g; j = 1,2,…,j(i);

h = 1,2,…, j(g); K € M_{ij} n M_{gh}

$(B_{ij} + P_{ijk}X_{ijk}) - (B_{gh} + P_{ghk}X_{ghk}) + \gamma(1 - Y_{ghijk}) \geq P_{ijk}X_{ijk}$ (3.7)

i = 1,2,…,N; g = 1,2,…,N; i ≠ g; j = 1,2,…,j(i);

h = 1,2,…, j(g); K € M_{ij} n M_{gh}

$(B_{gh} + P_{ghk}X_{ghk}) - (B_{ij} + P_{ijk}X_{ghk}) + \gamma(1 - Y_{ijghk}) \geq P_{ghk}X_{ghk}$ (3.8)

i = 1,2,…,N; g = 1,2,…,N; i ≠ g; j = 1,2,…,j(i);

h = 1,2,…, j(g); K € M_{ij} n M_{gh}

Constraint set (3.6) guarantees that if operations j and h from job ii and g, respectively, are to be performed on the same machine K, then the two operations cannot be performed simultaneously. Constraint set (3.7) ensures that if operation j of job i is chosen to be processed before operation h of job g, the starting time and the processing time of the 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 (3.8) for the reverse case when operation h of job g is chosen to be processed before operation j of job i. Again, these constraints reinforce that one job is always processed before a second job on a given machine to avoid conflicts.

$B_{i1} \geq R_i$ i = 1,2,…,N …. .(3.9)

Constraint set (3.9) ensures that the first operation of a job i cannot start before it is ready.

$B_{i1} \geq 0$ i = 1,2,…,N; j = 2,…,j(i) …. (3.10)

MS ≥ 0 . (3.11)

Non negativity constraints (3.10) and (3.11) ensure that all starting times for the remaining operations and the manufacturing make span are positive.

X_{ijk} € {0,1}, i = 1,2,…,N; j = 1,2,…,j(i); k = 1,2,…,M; . (3.12)

Y_{ijghk} € {0,1}, i = 1,2,…,N; j = 1,2,…,j(i); g = 1,2,…,N;. (3.13)

h = 1,2,…, j(g); k = 1,2,…, M.

Constraints (3.12) and (3.13) show the integer constraints for the 0-1 variables.

3.3 Extension of the Basic Model

A 0-1 MILP formulation was intended to give an optional solution, as well as provide a basic understanding and foundation for a given FMS scheduling problem. Thus, it lends itself to be easily extended to a number of performance measures that are dependent on what a scheduler needs to resolve for a specific manufacturing environment. With minimal change of basic model, some of the regular performance measures that could be used with this MILP formulation are total completion time, mean completion time, total flow time, mean flow time, maximum lateness, maximum tardiness, or even the number of tardy jobs.

The minimum changes that are required to extend this model for two additional performance measures and one scheduling conditions are presented as follows:

Additional notation for model extensions

C_i = manufacturing completing time of job i = 1,2,…,N

D_i = due date of job i = 1,2,…,N.

E_i = max {0, Di - (i)}, the earliness of job i, i = 1,2,…,N.

T_i = max {0, i - Di}, the tardiness of job i, i = 1,2,…,N.

Tmax = $\max_i \{T_i\}$, the maximum tardiness.

αi = the unit earliness penalty for job i where αi > 0 i = 1,2,…,N.

βi = the unit tardiness penalty for job i where βi > 0 i = 1,2,…,N.

Maximum Tardiness Problem

In order to minimize tardiness, due date parameters are required in addition to the processing times and machine routing information that is usually provided in process plans. To use the basic MILP model, the make span minimization objective formed is the most to be changed to the following:

Min Tmax .(3.14)

This represents that the new objective is to minimize the maximum tardiness. In addition, constraint set (3.3) must be replaced with the following set of constraints:

Constraint set (3.3) must be replaced with the following set of constraints:

B_i j (i) + Σ $P_{i,j(i)\ k}$ $X_{i,j(i)k}$ = $(_{I,\ i21,2....N}$- (3.15)

$KEM_{ik(i)}$

The following additional set of constraints must be added:

$(_i$-D_i- T_{max} ≤ O, $_{I=1,\ 2,\ ....N.}$ (3.16)

Lastly constraint (2.11) must be replaced with the following constraint:

T_{max} ≥ O - (3.17)

Constraint set (2.15) ensures that the starting time and process time of the last operation J (i) for job I,I =1,2,----N is equivalent to the manufacturing completion time, while constraint set (3.16) ensures that the tardiness of job i, I, =1,2, --------- N is less than or equivalent to the maximum tardiness. The combination of (3.14), (3.15) and (3.16), that is the new objective friction with the new constraints, will ensure that the maximum tardiness objective is minimized white (3.17) ensures that the maximum tardiness value will be non-negative.

However it has been determined [23] that the basic O-1 MILP model presented above is NP—lord. Hence the two-stage routing and regarding MILP (2-MILP) model was proposed to solve the FMS routing and scheduling problem more efficiently. The full model is split into two MILP sub problems. This two stage procedure relaxes the basic O—1 MILP model's procedure constraints set (3.2) and (3.3) in the first sub-problem (Stage 1) to determine the routing of the jobs, while the second sub-problem (stage 2) uses the results from stage 1 to determine the sequence of the jobs.

For stage 1, the problem of minimizing the manufacturing make span is equivalent to using the following objective friction and constraints sets:

(Stage 1):

Min Ms (3.18)

S.t

Σ X $_{ijk}$ = 1, $_{i=1,\ 2,\ ---------------Nij=1,2,\ ----------\ j(i)}$ - (3.19)

$K€M_{ij}$

ΣΣ P_{ijk} X $_{ijk}$ --- MS ≤ O, $_{K=1,2,\ -------M}$ - (3.20)

i€N j€j (i)

MS ≥ O, (3.21)

X_{ijk} € {O, i} (3.21)

i = 1,2 -------, N, j = 1,2 -------,) (i) j k=1,2, ------,M

in this stage 1 sub-problem, only one new constraint set (3.20) was added. This new constraint set ensures that for every machine k, which is loaded with selected operation jEj (i) of jobs iE N, the total processing time on each machine is less than or equal to the overall make span. The remaining equations (3.18), (3.19), (3.21) and (3.22) are taken from the original formulation of the basic O-1 MILP model.

Once the stage 1 model is formulate and solved, the X_{ijk} routing variables are fixed (i.e, the routes are fixed for all of the jobs to be processed). The X_{ijk} variables that are assigned a value of one indicate that operation j of job I is assigned to machine k, while those assigned a value of zero are not assigned to any machines. This, a new subscript $\acute{M}_{ij}$ (or Ḿgh) is introduced where $\acute{M}_{is}$ = {K: X_{ijk} = 1, K € M_{ij}}. This represents the selection of the single machine K that is chosen (i.e., the job assigned to the machine) in stage 1 from the set of optional machines M_{ij} from operation j or job I. once this imported information has been established, the stage 2 sub-problem can now be formulated with the following objective friction and constraints sets:

(Stage 2):

M_{in} M_s (3.23)

s.t

B_{ij} + $P_{i,k}$ ≤ $B_{ij+1,\ I=1,2,-----N,}$ j=1,2---j(i) -1, K€M --- (3.24)

$B_{ij(i)}$+ $P_{i,j\ (i)jk}$ ----Ms ≤ O $_{i=1,2------N;\ KE}$ $\acute{M}_{ij\ (i)}$ -- (3.25)

Y_{ijghk} + Y_{ghijk} = 1 ----- (3.26)

i =1,2,---N, g = 1,2,-----,N i g j =1,2, =1,2, -----J (i)

$h= 1,2, -----, j\ oj\ ; KE\ \acute{M}_{ij\ n}\ \acute{M}_{gh}$

$(B_{ij} + P_{ijk}) – (B_{gh} + P_{ghk}) + Y\ (1-Y_{ghijk}) \geq P_{ijk}, -----$ (2.27)

$i= 1,2, --------- N; g =1,2, ------N, ig; J= 1,2-----J\ (i)$

$h = 1,2, -------- J\ (g)\ ; K \text{€}\ \acute{M}_{ij\ n}\ \acute{M}_{gh.}$

$Bi1 \geq P_i, i =1,2,-----------,N$ ------------ (3.28)

$Bij \geq O\ i=1,2,-----,N\ j=2, ------,J\ (i)$---- (3.29)

$MS \geq O$ ------------ (3.30)

$Y_{ijghk\ \text{€}} \{O, 1\}$ ----------------- (3.31)

$i = 1,2,---------N; j=1,2---------j(i); g=1,2,----------,Nj$

$h=1,2,----------j(g); k=1,2,---------,M$

These changes reflect that the stage 2 model uses the new X_{VE} routing variables from stage1, thus reducing the total number of integer variables used in this model compared with the amount used in the original O-1 MILP model. Once stage 2 has been solved, all hobs that have been previously routed are now in machine sequence, and a final make span value is determined.

4.0 OVERVIEW OF PROPOSED SYSTEM

.

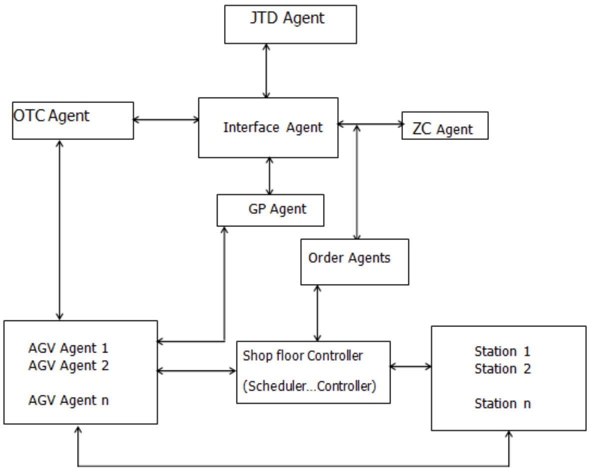

Fig.1: Overview of Proposed System

The block presents the proposed system. It consists of society of agents to resolve the issue related to conflict, deadlock and interruption occurring in a FMS. Six types of agent have been proposed to develop the architecture of the control mechanism of AGVs in FMS. Each agent is associated with modules and these modules follow some rule bases, heuristics and algorithms. The generation of the rule base is automated using evolutionary neural network to negotiate the conflict and interruption related to operating control of AGVs in FMSs.

Guide Path (GP) Agent: The guide path agent is responsible for finding the possible paths through which the AGVs can be routed. A guide path agent receiver's information forms an AGV agent regarding its present location, location of pick-up, and drop-off work station. GP agent generates shortest feasible path based on the high level reasoning capability of the artificial neural network associated with it. Based on this link occupation time is estimated by JTD agent.

Journey Time Database (JTD) Agent: The journey time database agent enumerates the link occupation time of each AGVs. The JTD agent generates link occupation time data according to vehicle speed. Link occupation time is the interval between the entry time and exit time of a vehicle on a link. It also includes the response time that is nothing but the time from the start of the algorithm execution until start of the vehicle movement. Link occupation time for every link is stored in the firm of link occupation table (LOT). Hence, after dispatching of a vehicle, the link occupation scheduled of the vehicle is stored in LOT. Zone controller (ZC) agent utilizes LOT to determine collision free trajectory of a vehicle.

Zone Controller (ZC) Agent: The zone controller agent is responsible for determining the conflict and interruption free path/route associated with other moving vehicle within the horizon of journey time schedule. The zone controller agent utilizes link occupation table data to determine a collision and deadlock free trajectory of a vehicle. Neural network automatically generates rules associated with ZC agent that ensures that overlapping of link occupation time is avoided.

Online Traffic Controller (OTC) Agent: The OTC agent determines the overall motion planning of AGVs. This agent is also the decision maker. The OTC agent on the basis of communication with other agent and the neural network generated rule base associated with it. After deciding the shortest feasible path (based on the neural net generated rule base), OTC agent instructs the AGV agents to initiate its motion and continuously governs its movement. If any problems related to the AGVs control (probable location to be head on collision of AGVs, breakdown of AGVs) arise, it reports the shop-floor controller to heal up the trouble.

Order Agent (OA): As generation of new order to transfer supplies from one station to another station on entry of new supplies in the system arises the shop-floor controller detects the requirement of AGVs to transport the supplies form station to station. The shop floor controller instructs the order agent to develop a plan for transportation of the supplies. The order agent passes the information to other agents of the system for finding and solving the transportation demand. According to the OTC agent instruction AGVs can load and dispatch the supplies.

AGV Agent: Each AGV is associated with an AGV agent. AGV agent manages AGV moment. These AGV agents manage an AGV by initiating enquires with other agents and by negotiating with other AGV agent. An AGV agent make a decision on the basis of message sent by OTC agent (routing plain is managed by OTC agent). AGV agents communicate with OTC agent at each incident such as AGVs cross the node, receives supplies and unloading the supplies etc. if any AGV breaks down or AGVs come into location to be head-on collision, OTC agent indicates to the shop-floor controller to recover the AGV or land to reschedule the movement plan of the remained journey of the supplies.

Shop floor controller: The shop floor controller is a neural network controller that schedules and reschedules the movement of AGVs at FMS shop floor (i.e shop floor scheduling). It leads manufacturing control instructions from programs loaded from operators into the shop floor computer. From this it detects the requirements of AGVs to transport supplies from station to station. Based on this it instructs the relevant agents. It reacts to signal from agents in order to clear AGVs from the shop floor to avoid collision, dead lock and it reacts and fends alert messages to operators when AGVs breakdown, it reacts to such uncertainties as AGV breakdown, CNC machine becoming unavailable by dynamically rescheduling and interacting with the agents to reroute the AGVs in a flexible manner.

5.0. CONCLUSION

Most of the FMS control algorithm as hinted above treat the routing problem as a shortest path problem in graph theory, to find a route for a vehicle, the algorithm usually have to search for every node and are of the path network graph [17]. Some of the algorithm may also miss the optimal solutions because certain constraints are overly enforced. The complexities of some of the algorithms especially O-l MILP and the O-l integer programming makes them unviable for real-time shop floor conditions including the inability to adapt to uncertain shop floor conditions like AGVs or CNC machines breaking down unexpectedly. Consequently the need arises for more intelligent algorithm for automating a FMS material handling system. Hence this work proposes more intelligent digital control algorithm for the FMS based on the combination of intelligent agent and neural network technologies for optimal AGV routing and scheduling.

REFERENCES

1. S. Takukawa (1997), “the Use of simulation in Activity Based costing for flexible manufacturing systems” Proc of the 1997 winter simulation conference, PP 793-800.

2. G.G. Kosi, and R. Zdanwowicz (2005), “Modeling of manufacturing systems and robot motions”, Journal of materials processing Technology, Vol. 164-165, PP 1369-1378.

3. M. Kaighodadi and K. Verkatesh (1994), “Flexible Manufacturing systems: an overview” International Journal of Operations of production management, Vol. 14. PP 26-49.

4. M. P. Grover (1987), “Automation Production system and computer Integrated Manufacturing New York: Prentice Hall.

5. Steeke KE, Solberg JI (1983) Loading and control policies for a flexible manufacturing system. Int J Prod Res 19 (5); 481-490.

6. P.E. Guy, and A. Castleberry. The AGV Handbook Ann Arbor, Michigan: Brain-Brum field, Inc. 1991, PP. 378, 379, 380.

7. T. Ohno (1958), Toyota Production system: Bey and Large scale production, Productivity press.

8. Daniels, S.C., (1988) Real-time conflict resolution in automated guided vehicles scheduling. PhD thesis, Department of industrial engineering, Pennsylvania State University, USA PP 167-169.

9. Glover, F., K. lingman, D.D. and Phillips, N.V (1935), “A new Polynomial bounded shortest path algorithm”. Operations Research, 33 (1), 65-73.

10. Huang, J. Palekar, US and Kapoor, S.G. (1989)” A labeling algorithm for the navigation of automated guided vehicle” In advances in manufacturing systems Engineering, Proceeding of the ASME winder Annual meetings, San Francisco, CA, PED Vol. 37. PP 181-193.

11. Kim, C.W and Tanchoco, J.M.A (1991), “Conflict free shortest-time bi-directional AGV routing” International Journal of production research, 29,2377-2391.

12. Potts, C.N. and Whitehead, J.D. (2001), “Workload balancing and loop layout in the design of a flexible Manufacturing system” European Journal of Operational Research, 129 (2), 326-336.

13. Goetz, W.G. and Egbelu, P.J. (1990), “Guide path design and location of load pick-up/SWA point for an automated guided vehicle system.” International Journal of Production Research, 78, 927-841.

14. Egbelu. P.J & Tanchoco, J.M.A (1986), “Potential for bi-directional guide path for automated guided vehicle based systems” International Journal of production Research, Vol. 24. No.5 PP 1075-1097.

15. Co, H.C., Biermann, J.S., and Chen, S.K (1990), “A methodical approach to flexible manufacturing-system batching loading and tool configuration problem”. International journal of production Research, 28 (12), 2171-2186.

16. S. Takukawa (1997), “the Use of simulation in Activity Based costing for flexible manufacturing systems” Proc of the 1997 winter simulation conference, PP 793-800.

17. Goswami, M. and Tiwari, M.K (2006), “A reallocation based heuristic to solve a machine loading problem with material handling constraint in a flexible manufacturing system”. International Journal of production research, 44 (3), 569-588.

18. Diptain Dulta, Argha Roy, Kaustav Choud hury. “Training Artificial Network Using particle Swarm optimization algorithm” International Journal of Advance Research in Computer science an software

engineering Vol.3 Issue 3, March 2013 ISSN 2277-123 x PP 430-432.

19. Kennedy. J: Eberhort, R. (1995), “Particle Swarm Optimization” Proceeding of IEEE international Conference on Neural Network.

20. Kennedy J: Eberhort R.C. (2001) “Swarm Intelligence” Morgam Kanufarms.

21. Huang, J. Palekar, US, Kapoor, S.G. (1997)” A labeling algorithm for the navigation of automated guided vehicle” Trans ASME J by Ind. 115:315-321.

22. Miller RK (1987), “Automated Guided vehicles and automated manufacturing” Dearborn, MI, Soc Maufact Eng.

23. Goswami, M. and Tiwari, M.K (2006), “A reallocation based heuristic to solve a machine loading problem with material handling constraint in a flexible manufacturing system”. International Journal of production research, 44 (3), 569-588.