Machine Shop Material Flow Analysis and Scheduling

Anil Kumar
M.E.(Manufacturing & Automation), Department of Mechanical Engineering
UIET MDU, Rohtak, Haryana, India

Sandeep Kumar
Asstt. Prof. , Department of Mechanical Engineering
UIET MDU, Rohtak, Haryana, India

Abstract - This paper considers the dynamic scheduling of material handling equipments (Automatic Guided Vehicles or AGVs) and manufacturing facilities (machines) in the production of products. The aim of the problem is to minimize the cumulative lead time of the overall production schedule for on-time delivery, and to reduce the travelling distance of vehicle. The problem of delivery time minimization is formulated as Rank-Order Clustering. The parts to the same machines are groped to reduce the travel distance of the vehicle and to reduce the lead time. For industrial sized problems, an effective solution is developed for manufacturing schedule and material handling operations by finding critical path of an integrated operations network.

I. INTRODUCTION

Dissertation Idea:

With drastically increase in number of Transporters in manufacturing industry efficient algorithms are needed to resolve the increased contention of resources (e.g. path, loading and unloading buffers) among Transporters.
Ex: Consider a shop floor with 20 AGVs transporting material from storage area to the machining area and back. Complexities involved are high with handling so many AGVs crossing each other. With this high complex work environment requires complex planning.

II. LITERATURE SURVEY

Scheduling is the allocation of start and finish time to each particular order. Therefore scheduling can bring productivity in shop floor by providing a calendar for processing a set of jobs. It is nothing but scheduling various jobs on a set of machines or resources such that certain performance measures are optimized[1].
In Job Shop Scheduling, assume that each job has m different operations. If some of the jobs are having less than m operations, required number of dummy operations with zero process times. By this assumption, the condition of equal number of operations for all the jobs is ensured. In job shop scheduling problem, the process sequences of the jobs are not the same[2].
Objective of the integrated problem is to minimize the cumulative lead time of the overall production schedule for on-time shipment, and to reduce material handling and inventory holding costs on the shop-floor [3]. A complexity boundary is presented for each model and related existing results are consolidated and also introduce some enumerative solution algorithms and heuristics and analyze their performance [4].
Mathematical programming has been applied widely to machine shop scheduling problems. Problems have been formulated using integer programming [5], mixed-integer programming and dynamic programming [6].
Davis and Jones [7] proposed a methodology based on the decomposition of mathematical programming problems that used both Benders-type [8] and Dantzig/Wolfe-type [9] decompositions.

Some of the terms used in scheduling are:

- ❖ *Loading:* It is a process of scheduling how much of load each centre must carry during the coming planning period. Loading assigns hours of work to work centers in accordance with the available capacity of the work centers.
- ❖ *Sequencing***:** It is the process of giving priorities for jobs in the waiting lines at the work centers.

http://dx.doi.org/10.21172/1.72.591

- *Scheduling:* The aim of transporter scheduling is to dispatch a set of transporters to achieve the goals for a batch of Pickup/Drop-off (or P/D for short) jobs under certain constraints such as deadlines, priority, etc. The goals are normally related to the processing time or utilization of resources, such as minimizing the number of transporters involved while maintaining the system throughput, or minimizing the total travel time of all vehicles.
- *Routing:* it is the process of assigning traveling path to AGV's.
- *Dispatching:* It is the process of selecting and sequencing jobs to be done at individual work centers and subsequently assigning the work to be done.
- *Set up time:* It is the time required to adjust a machine and mount proper tooling take to make a job.
- *Lead time:* It is time period between releasing an order and the completion of first unit. This includes waiting time, moving time, set up time, time in queue and run time.
- *Planned order:* It is a customer order that is on the books and planned for production but not yet has been launched into production.
- *Open order:* It is a customer order that has been launched into production and is in process.

Routing AGV routing problem is modelled using an evolutionary algorithm-based intelligent path planning model, which handles vehicle assignments to material handling requests and makes routing decisions with the objective of maximizing the system throughput [10].

Automated Guided Vehicles (AGVs) are state-of-the-art technology for optimizing large scale production systems and are used in a wide range of application areas. A standard task is to find efficient routing schemes, i.e., algorithms that route these vehicles through the particular environment. The productivity of the AGVs is highly dependent on the used routing scheme. The key feature of this algorithm is that it avoids collisions, deadlocks and livelocks already at the time of route computation [11].

For AGV routing problems, Genetic Algorithm (GA) or a distributed route planning method has been proposed by Nishi et al. [12]. The method is to derive the routing to minimize total transportation time for AGVs by repeating the generation of routing for each AGV and data exchanging between the routing for each AGV.

In a flexible manufacturing system there can be several alternative routes for a product [13]. They are:

- Fixed routing: Each product has only one possible route. This type of routing is used mainly in automated production.
- Static routing: The first currently feasible route in a priority list is selected.
- Local dynamic routing: The decision is based on some information of the current situation that the module has. This information can be based on queries to connected modules of the type.
- Global routing: The central controller has most information on the current global situation of the Manufacturing system, and should therefore be able to make optimal routing decisions.
- In some applications of Transporter, only a few vehicles and jobs are involved. In this case, simple scheduling algorithms can serve the purpose. Jobs are usually handled in a First-Come-First-Serve (FCFS) fashion, and the nearest idle vehicle is usually chosen to serve a new job. Therefore, in this case scheduling seems very trivial and the research emphases are focused on routing.
- In applications that involve many jobs, however, due to limitations of facility resources such as paths and loading/unloading buffers, a simple scheduling algorithm may not achieve high system efficiency. For example, when we have only a limited number of vehicles but many jobs, jobs have to be divided into several batches based on related factors such as traffic capacity or waiting time, and accordingly Transporters have to be dispatched and routed concurrently for each of the batches of jobs. In this case, the scheduling strategy must ensure that the conditions under which the given routing algorithm works are satisfied. For instance, no two vehicles are dispatched to a same destination in the same batch of jobs.

AGV Automating routine movement of goods frees up truck drivers for more demanding and flexible work, while reducing errors and accidents and rendering operations transparent.

The problem of AGV scheduling and routing and then differentiates it from several related problems and classifies major existing algorithms for the problem [14]. AGVs has been developed to make use of cassettes that can be detached from the AGV and developed an agent-based simulator for evaluating the cassette-based system and comparing it to a more traditional AGV system in order to find the most efficient configuration [15].

Free Ranging on Grid (FROG) has developed a path design tool, in which a vehicle path can be drawn by a human designer using NURBS (Non-Uniform Rational B-Splines) curves [16].

The routing efficiency is analyzed in terms of the distance traveled and the time required for AGVs to complete all pickup and drop-off jobs [17].AGV system based on a time-driven approach and execute the model on an efficient simulation engine implemented by using Cilk and then elaborate on the AGV routing algorithms and traffic control scheme and propose a deadlock-free decentralized routing scheme.

The concept of free-ranging AGVs is fairly recent; free-ranging AGVs do not travel on guide paths but are instead controlled by radio messages. In pioneering work, Taghaboni and Tanchoco describe a LISP-driven controller for scheduling free-ranging AGVs

The greatest benefits are gained in multi-shift work.

- *Cost effective*
- *Flexible*
- *Safe*
- *Expansion & Changes*
- *24 Hour Service*
- *Reduction in Labor costs*
- *Careful Handling*
- *Excellent trace & tracking*
- *High Availability*
- *Load Scheduling & Priorities*
- *Accessible Layout*

Applications of AGVs Automated Guided Vehicle systems are used in a growing number and variety of applications. The principal of AGVs applications in production and logistics are (1) driverless train operations, (2) storage and distribution, (3) assembly line applications and (4) flexible manufacturing systems.

A driverless train consists of towing vehicle pulling one or more trailers to form a train. It is used to move the large quantities of materials over long distances in warehouses or factories with or without intermediate pickup and drop-off points along the route.

In typical operation, load the parts to the AGV from the receiving department and then deliver the parts to the individual workstations in the system. When the AGV arrives at the assigned station, the pallet is transferred from the vehicle platform to the station. At the completion of processing, a vehicle returns to pick up the work and transport it to the next assigned station.

Plant-Simulation is a Discrete Event Simulation Software (DES Software) which is developed by Siemens PLM Software for modeling, simulating, analyzing, visualizing and optimizing production systems and processes, the flow of materials and logistic operations. Using Tecnomatix Plant-Simulation can optimize material flow, resource utilization and logistics for all levels of plant planning from global production facilities through local plants, to specific lines.

Plant-Simulation enables to create well-structured, hierarchical models of production facilities, lines and processes. This is achieved through powerful object-oriented architecture and modeling capabilities that enable to create and maintain even highly complex systems, including advanced control mechanisms.

Simulating System Performance In Plant-Simulation simulations are used to optimize throughput, relieve bottlenecks and minimize work-in-process. The simulation models take into consideration internal and external supply chains, production resources and business processes, allowing to analyzing the impact of different production variations. We can evaluate different line production control strategies and verify synchronization of lines and sub-lines.

Automatic Optimization: Optimization can also be done automatically by using the Plant Simulation-Genetic Algorithms module. This is especially useful if a variety of system parameters and constraints make it difficult to find the optimum. Genetic Algorithms optimize system parameters while considering multiple constraints such as throughput, inventory, utilization of resources and delivery dates.

Analyzing Simulation Results Plant Simulation analysis tools allow for easy interpretation of simulation results. Statistical analysis, graphs and charts display the utilization of buffers and machines etc. and also generate extensive statistics and charts to support dynamic analysis of performance parameters including line workload, breakdowns, idle and repair time and proprietary key performance factors. In Plant Simulation’s

- Bottleneck Analyzer shows the utilization of resources, thus indicating bottlenecks as well as underworked machines.

- ❖ Material flow may be visualized in a Sankey chart that, at a glance, shows transport volume in the context of the layout.
- ❖ Plant Simulation also generates a Gantt chart of the optimized production plans that can be modified interactively.

3D-Visualization In addition to the highly efficient 2D view, simulation may be visualized in a virtual, reality-like 3D environment. Using Plant Simulation's libraries or CAD data, impressive 3D models can be built and used in several ways, including:

- ❖ Easy-to-understand platform for internal and external discussions
- ❖ Management presentations
- ❖ Demonstration to customers
- ❖ Trade show displays.

III. PROBLEM DEFINITION AND SCOPE

Problem Definition:

In the manufacturing facility layout raw material, Work in Process, Finish goods are stored at different places and material handling equipments (AGV, Forklift, human etc.) used to move the materials when required. The current trend of market is so fast and advance that demand is very high with small product life cycle and huge competition with respect to cost and quality. Planning of the production schedule and routing for AGV is complex activity in the manufacturing facility which is producing verity of products.

Scope:

Discrete Event Simulation software is used to analyze the automatic routing of material handling equipments. The performance of the material handling system with delays and jobs in the queue before each work station is studied. The jobs are distributed according to the *work plan table* in Plant-Simulation software, which generates a set of Pickup and Deposit tasks. Initially, the vehicles are produced from the *AGV_pool* and then transfer the materials to the machines according to the table. When transferring all the materials rest at the Depot (where the vehicles reside after completing the job) When a P/D job is assigned to an AGV, the route for the AGV is determined beforehand.

Objectives:

The aim of this work is routing of transporter in a non-conflicting manner. To make the best route for the transporter which move along the path, such that the distance traveled will be minimized. A single vehicle is allowed to carry number of jobs at a time as per the transporters Capacity. Hence, the primary objectives of the research are:

- To study the routing and path layout design for a specific path topology, and to use the analogous ideas for transporter routing model.
- To design the bi-directional path layout in such a way that the transporter routing is achieved in a non-conflicting manner.
- To develop a criteria for conflict-free routing.

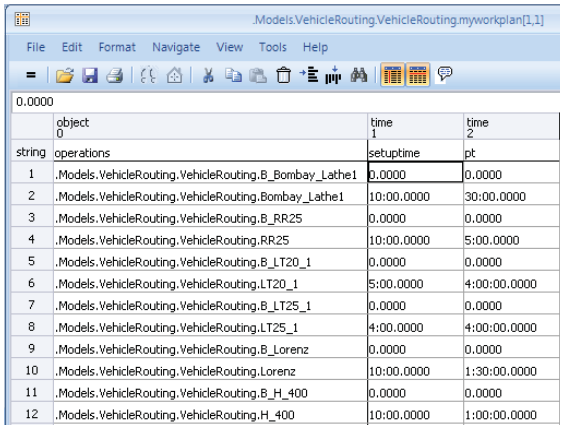
IV. SYSTEM DESIGN

- The model of Automatic Routing for Transporter can be done using Tecnomatix-Plant Simulation (Discrete-Event Simulation) software.
- This software is used for modeling, simulating, analyzing, visualizing and optimizing production systems and processes.
- First, create a new Frame in Plant-Simulation Software and insert the objects (Source, Buffer, Machine, Tracks, Transporter, Drain Methods, Tables, and Charts) which are related to the model.
- Transporter loads the materials from source and then transfer to the machines according to MySequence Table. After completing the finished products then moves on to the drain. These are the conditions which has been done using Method (Program).
- Next, group the parts by using Rank-Order Clustering to reduce set-up time and to reduce the travel distance of the vehicle. After grouping the parts, then put into the model.
- Compare the working, waiting and utilization of each machine before and after grouping.

Investigations on routing analysis with a case study:

Representative network architecture for cloud storage service architecture is illustrated in Figure 1. Three different network entities can be identified as follows:

Set-up time and processing time:



0.0000

	object 0	time 1	time 2
string	operations	setuptime	pt
1	.Models.VehicleRouting.VehicleRouting.B_Bombay_Lathe1	0.0000	0.0000
2	.Models.VehicleRouting.VehicleRouting.Bombay_Lathe1	10:00.0000	30:00.0000
3	.Models.VehicleRouting.VehicleRouting.B_RR25	0.0000	0.0000
4	.Models.VehicleRouting.VehicleRouting.RR25	10:00.0000	5:00.0000
5	.Models.VehicleRouting.VehicleRouting.B_LT20_1	0.0000	0.0000
6	.Models.VehicleRouting.VehicleRouting.LT20_1	5:00.0000	4:00:00.0000
7	.Models.VehicleRouting.VehicleRouting.B_LT25_1	0.0000	0.0000
8	.Models.VehicleRouting.VehicleRouting.LT25_1	4:00.0000	4:00:00.0000
9	.Models.VehicleRouting.VehicleRouting.B_Lorenz	0.0000	0.0000
10	.Models.VehicleRouting.VehicleRouting.Lorenz	10:00.0000	1:30:00.0000
11	.Models.VehicleRouting.VehicleRouting.B_H_400	0.0000	0.0000
12	.Models.VehicleRouting.VehicleRouting.H_400	10:00.0000	1:00:00.0000

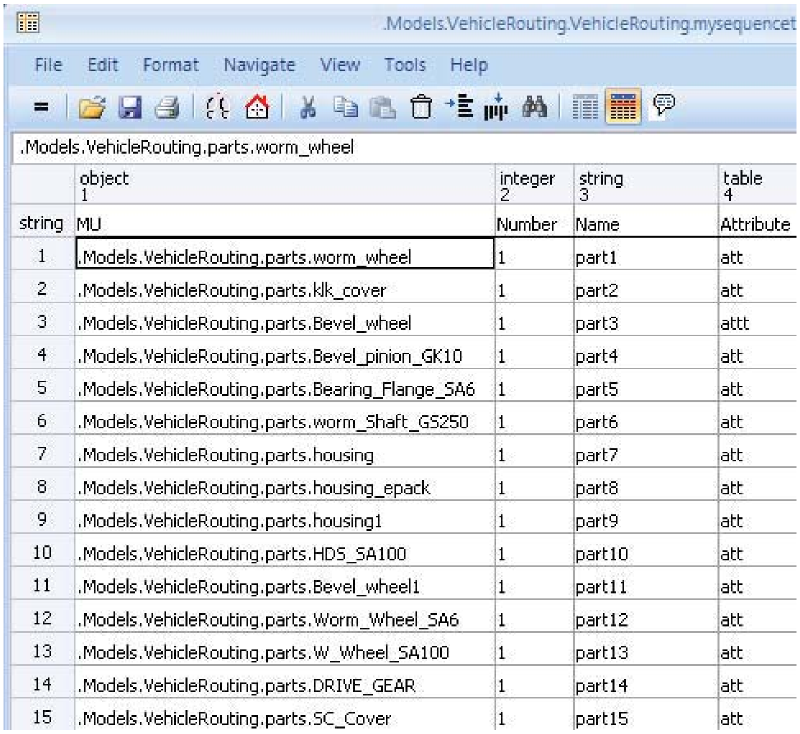
Fig 1. MyWorkPlan_Subtable of Operations for part1

We enter the set-up times of the stations and the processing times of the different parts into subtables of the work plan. To meet the needs of set up operations, the application object Station contains the object setup place. Set-up times are needed for changing the part type, the order or the operating step. Set-up times are either specified in the routing plan or in the station. There are three possibilities to define set up and processing times:

- Input in the operations plan
- Input in the dialog of the application object Station
- Input in the dialog of the basic object.

The dialog of the application object Station allows the user to define the set-up and processing time as follows: according to the operations plan (part specific) and to the station (station specific).

Mysequencetable:



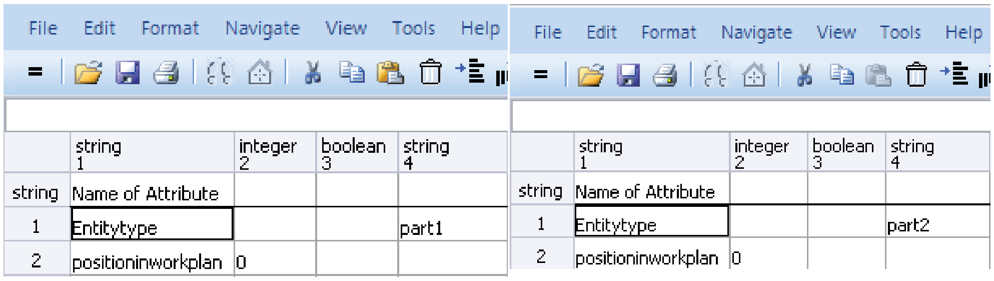
.Models.VehicleRouting.parts.worm_wheel

	object 1	integer 2	string 3	table 4
string	MU	Number	Name	Attribute
1	.Models.VehicleRouting.parts.worm_wheel	1	part1	att
2	.Models.VehicleRouting.parts.klk_cover	1	part2	att
3	.Models.VehicleRouting.parts.Bevel_wheel	1	part3	attt
4	.Models.VehicleRouting.parts.Bevel_pinion_GK10	1	part4	att
5	.Models.VehicleRouting.parts.Bearing_Flange_SA6	1	part5	att
6	.Models.VehicleRouting.parts.worm_Shaft_GS250	1	part6	att
7	.Models.VehicleRouting.parts.housing	1	part7	att
8	.Models.VehicleRouting.parts.housing_epack	1	part8	att
9	.Models.VehicleRouting.parts.housing1	1	part9	att
10	.Models.VehicleRouting.parts.HDS_SA100	1	part10	att
11	.Models.VehicleRouting.parts.Bevel_wheel1	1	part11	att
12	.Models.VehicleRouting.parts.Worm_Wheel_SA6	1	part12	att
13	.Models.VehicleRouting.parts.W_Wheel_SA100	1	part13	att
14	.Models.VehicleRouting.parts.DRIVE_GEAR	1	part14	att
15	.Models.VehicleRouting.parts.SC_Cover	1	part15	att

Figure 2. MySequenceTable

Insert a TableFile into Frame from the folder InformationFlow in the Class Library or from the tab Information Flow in the Toolbox.

Open the table and enter the type of part which we want to produce, how many parts required, their name of the part and the name of the subtable containing the attributes of the part.



	string 1	integer 2	boolean 3	string 4
string	Name of Attribute			
1	Entitytype			part1
2	positioninworkplan	0		

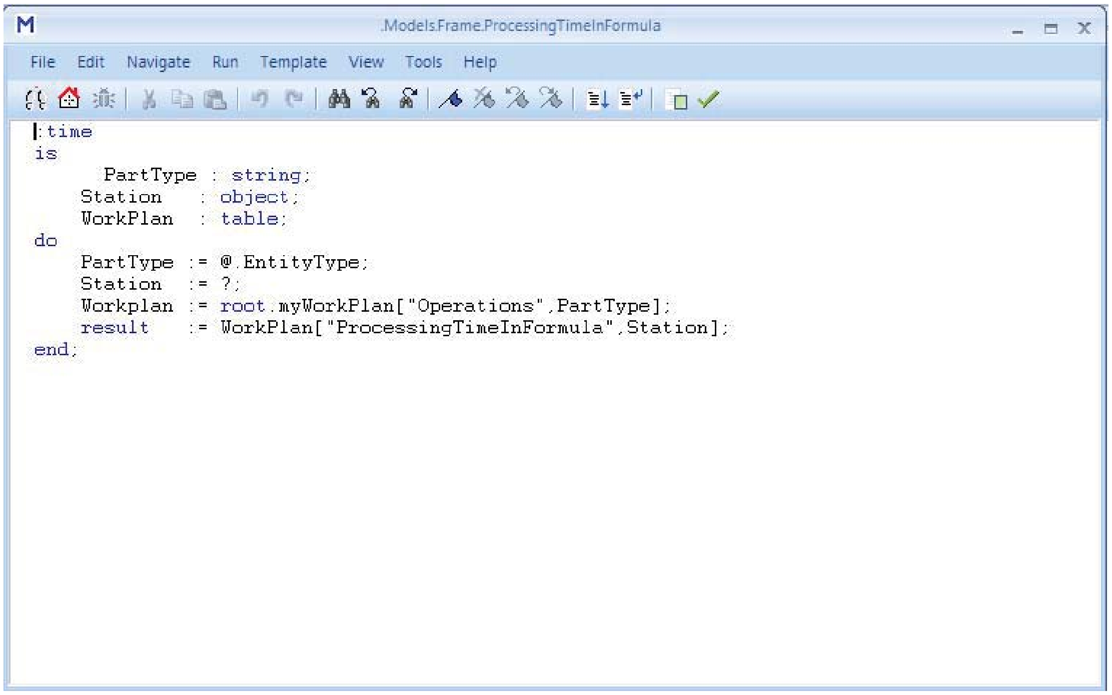
	string 1	integer 2	boolean 3	string 4
string	Name of Attribute			
1	Entitytype			part2
2	positioninworkplan	0		

Figure 3. MySequenceTable_SubTable for Part1 and Part2

To create a Subtable for the user-defined attributes of the parts, enter an identifier into the cells below Attributes. To open the Subtable, double-click on the Attribute cell. Here we enter the names and the values of two user-defined attributes: One set the name of the part, the EntityType is either Part1 or Part2 it depends on the desired Attributes. The other sets the position of the part in the work plan (PositionInWorkPlan).

V. METHOD

ProcessingTimeInFormula:



```
:time
is
      PartType : string;
    Station   : object;
    WorkPlan  : table;
do
    PartType := @.EntityType;
    Station  := ?;
    Workplan := root.myWorkPlan["Operations",PartType];
    result   := WorkPlan["ProcessingTimeInFormula",Station];
end;
```

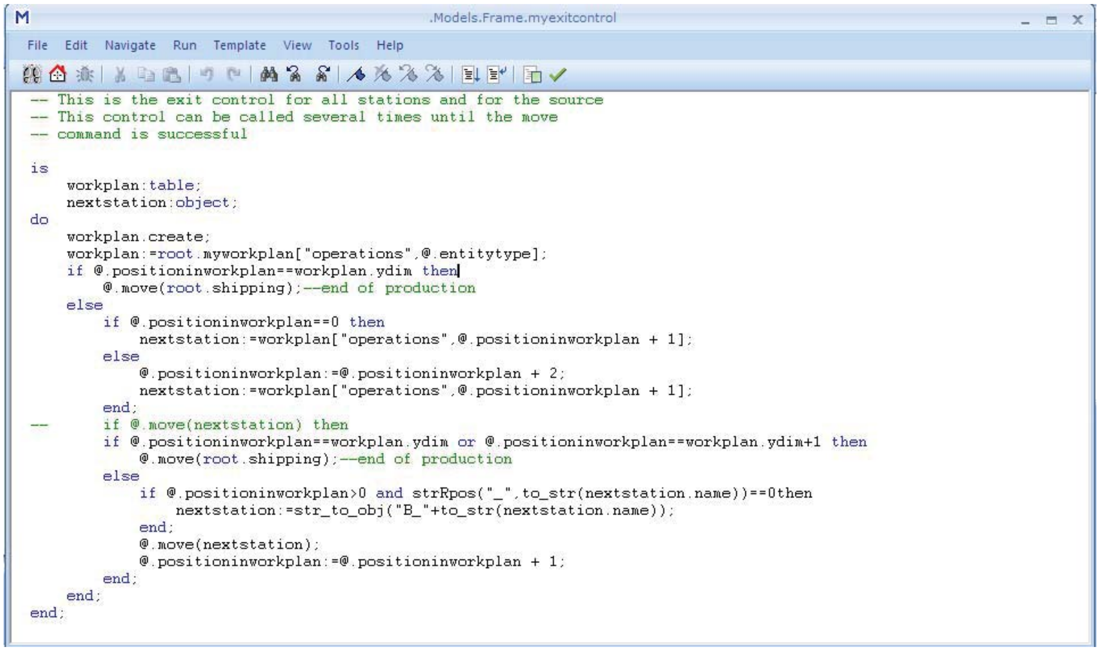
Figure 4. ProcessingTimeInFormula

The object Method is the container for miniature programs. To open it and enter the source code to be executed, double-click on Method [M] icon in the Frame into which we inserted it. Move the pointer after the keyword do, press the Enter key and enter the source code. Once program is written click Apply Changes and then click Run in the Method window. If the method expects arguments to be passed, Plant Simulation opens the Debugger, where we enter the arguments.

A method queries information from an object and returns a value. It calculates a value and starts one or several actions that control the behavior of the object. The stations to get the processing time of all parts from the MyWorkPlan Table we use a formula, which we programmed in the Method-ProcessingTimeInFormula. Select Formula as the Processing time in all stations and drag the Method- ProcessingTimeInFormula from Frame to the processing time of all stations.

MyExitControl:

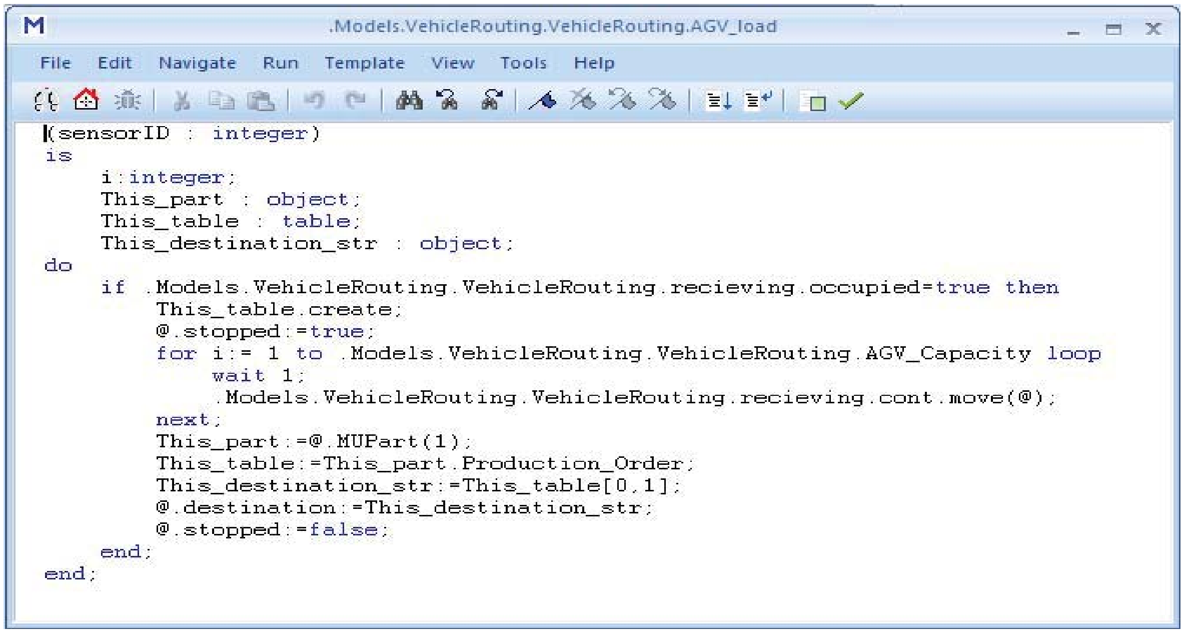
We program a Method, i.e., an Exit Control, which enables the processing station to find the next station in the sequence of operations after each processing step and moves the part onto this station. We place this method, which all stations access, into the Frame. If just the stations were going to use this control, we could have programmed it in a user-defined attribute of type method of the class of the station.



```
-- This is the exit control for all stations and for the source
-- This control can be called several times until the move
-- command is successful

is
    workplan:table;
    nextstation:object;
do
    workplan.create;
    workplan:=root.myworkplan["operations",@.entitytype];
    if @.positioninworkplan==workplan.ydim then
        @.move(root.shipping);--end of production
    else
        if @.positioninworkplan==0 then
            nextstation:=workplan["operations",@.positioninworkplan + 1];
        else
            @.positioninworkplan:=@.positioninworkplan + 2;
            nextstation:=workplan["operations",@.positioninworkplan + 1];
        end;
--      if @.move(nextstation) then
        if @.positioninworkplan==workplan.ydim or @.positioninworkplan==workplan.ydim+1 then
            @.move(root.shipping);--end of production
        else
            if @.positioninworkplan>0 and strRpos("_",to_str(nextstation.name))==0then
                nextstation:=str_to_obj("B_"+to_str(nextstation.name));
            end;
            @.move(nextstation);
            @.positioninworkplan:=@.positioninworkplan + 1;
        end;
    end;
end;
```

Figure 5. MyExitControl

AGV_Load:

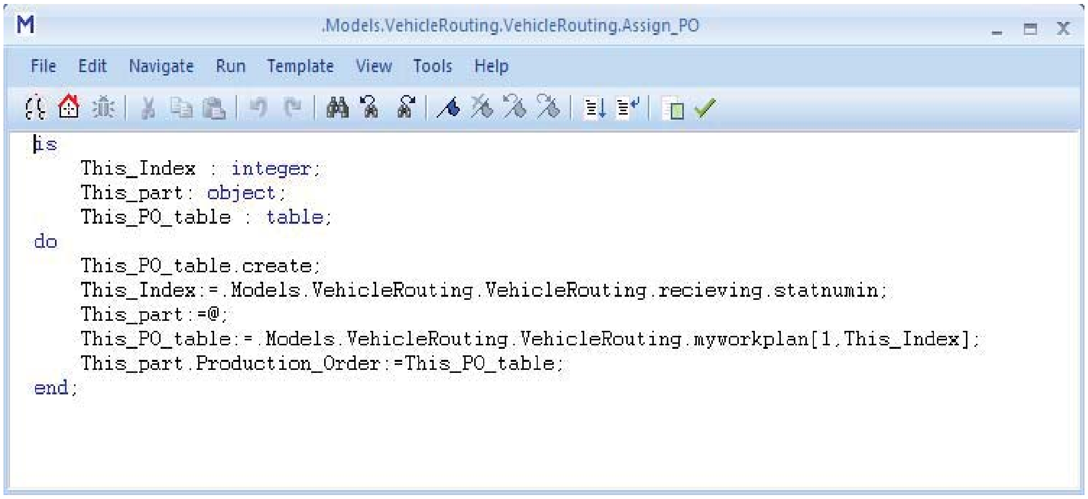


```
(sensorID : integer)
is
    i:integer;
    This_part : object;
    This_table : table;
    This_destination_str : object;
do
    if .Models.VehicleRouting.VehicleRouting.recieving.occupied=true then
        This_table.create;
        @.stopped:=true;
        for i:= 1 to .Models.VehicleRouting.VehicleRouting.AGV_Capacity loop
            wait 1;
            .Models.VehicleRouting.VehicleRouting.recieving.cont.move(@);
        next;
        This_part:=@.MUPart(1);
        This_table:=This_part.Production_Order;
        This_destination_str:=This_table[0,1];
        @.destination:=This_destination_str;
        @.stopped:=false;
    end;
end;
```

Figure 6. AGV_Load

This Method which indicates, to load the amount of parts or materials as per the sequence in the Mysequencetable from the receiving (where the parts are to be produced) to the AGVs according to the AGVs Capacity and then transport these materials from one machine to another machine.

Assign_PO:

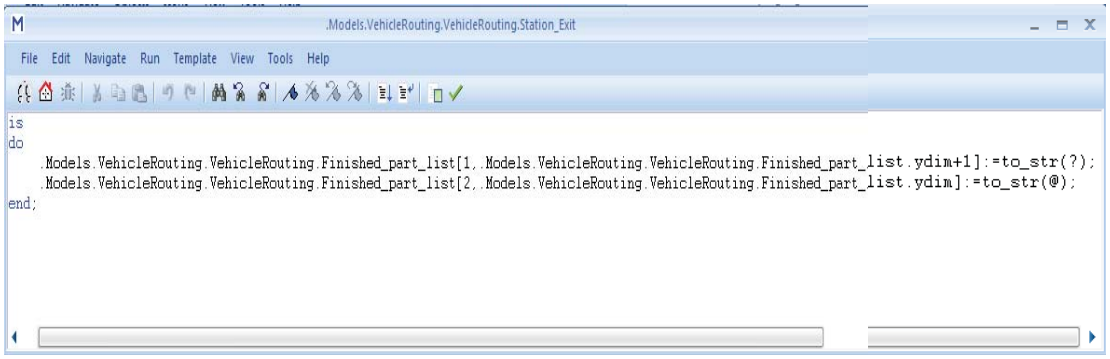


```
is
    This_Index : integer;
    This_part: object;
    This_PO_table : table;
do
    This_PO_table.create;
    This_Index:=.Models.VehicleRouting.VehicleRouting.recieving.statnumin;
    This_part:=@;
    This_PO_table:=.Models.VehicleRouting.VehicleRouting.myworkplan[1,This_Index];
    This_part.Production_Order:=This_PO_table;
end;
```

Figure 7. Assign_PO

This method triggers in the receiving (Source) department where the parts are produced according to the MyWorkPlan table.

Station_Exit:
This Method activates the exit controls, when an MU exits the object and then selects Front or Rear set.

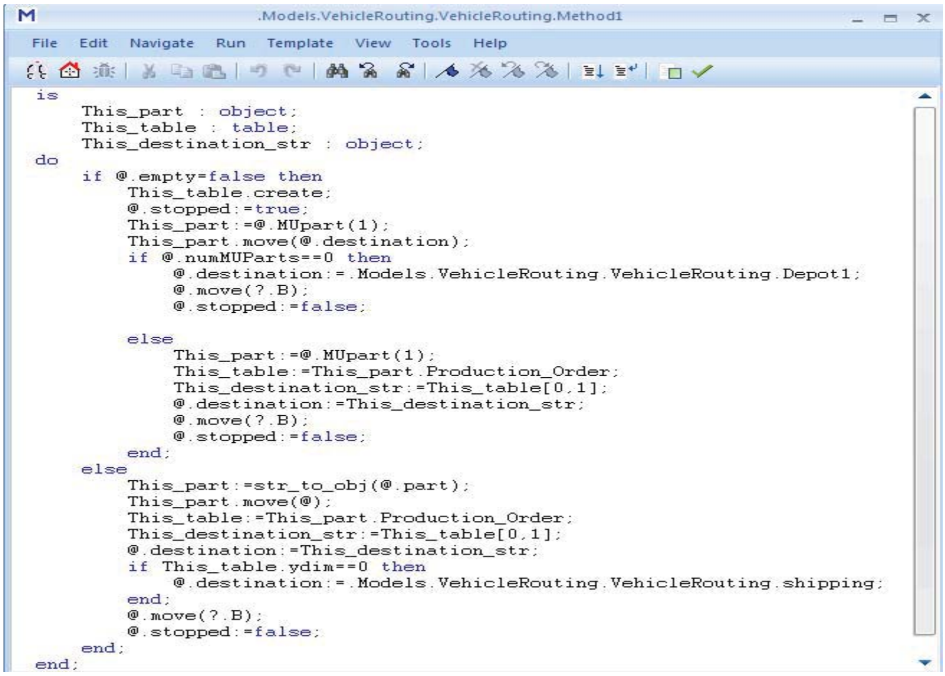

.Models.VehicleRouting.VehicleRouting.Station_Exit
File Edit Navigate Run Template View Tools Help

```
is
do
    .Models.VehicleRouting.VehicleRouting.Finished_part_list[1,.Models.VehicleRouting.VehicleRouting.Finished_part_list.ydim+1]:=to_str(?);
    .Models.VehicleRouting.VehicleRouting.Finished_part_list[2,.Models.VehicleRouting.VehicleRouting.Finished_part_list.ydim]:=to_str(@);
end;
```

Figure 8. Station_Exit

If select Front to activate the control as soon as the MU is ready to exit the object. The exit control then has to move the MU to another object, as the built-in behavior of the material flow object, i.e., moving the MU on to the successors along the Connectors, has been overridden by the exit control.
If select Rear to activate the control as soon as the rear of the MU has completely exit the object. This control does not override the built-in behavior of how MUs move to the successor of the present object.

Buffer_Load:


.Models.VehicleRouting.VehicleRouting.Method1
File Edit Navigate Run Template View Tools Help

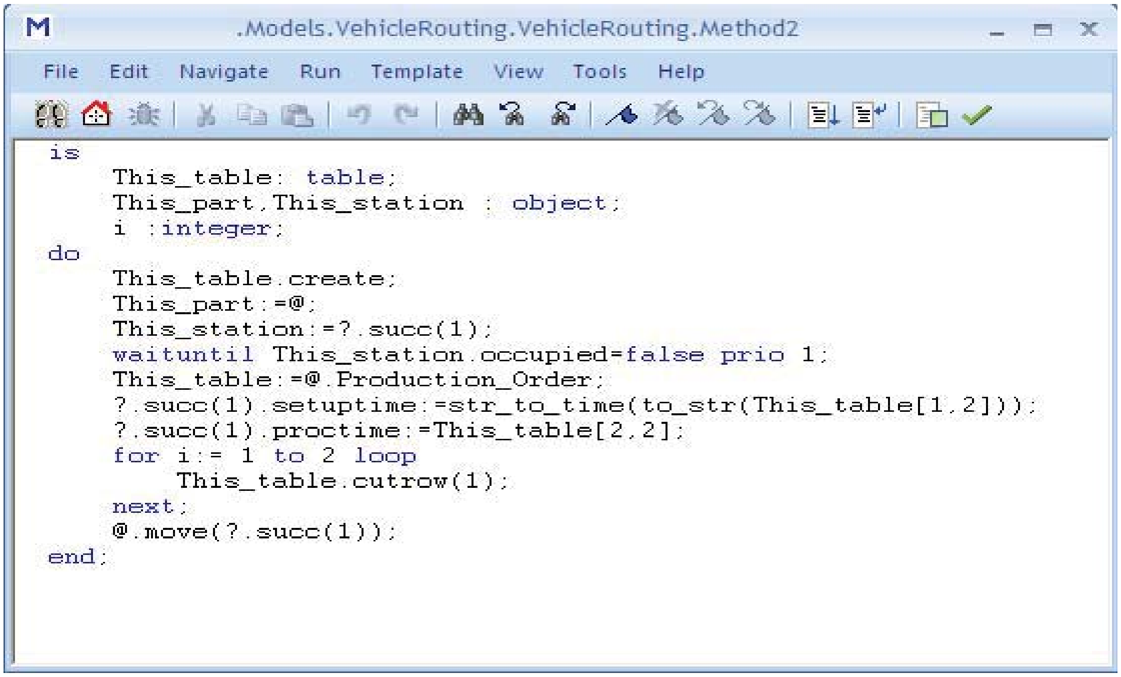
```
is
    This_part : object;
    This_table : table;
    This_destination_str : object;
do
    if @.empty=false then
        This_table.create;
        @.stopped:=true;
        This_part:=@.MUpart(1);
        This_part.move(@.destination);
        if @.numMUParts==0 then
            @.destination:=.Models.VehicleRouting.VehicleRouting.Depot1;
            @.move(?.B);
            @.stopped:=false;

        else
            This_part:=@.MUpart(1);
            This_table:=This_part.Production_Order;
            This_destination_str:=This_table[0,1];
            @.destination:=This_destination_str;
            @.move(?.B);
            @.stopped:=false;
        end;
    else
        This_part:=str_to_obj(@.part);
        This_part.move(@);
        This_table:=This_part.Production_Order;
        This_destination_str:=This_table[0,1];
        @.destination:=This_destination_str;
        if This_table.ydim==0 then
            @.destination:=.Models.VehicleRouting.VehicleRouting.shipping;
        end;
        @.move(?.B);
        @.stopped:=false;
    end;
end;
```

Figure 9. Buffer_Load

This Method triggers on the track near the machine to load and unload the materials from the AGV to the buffer and from machine to the AGV. When the AGVs moves on the track near the buffer this method activates and then transfer the materials from AGV to the buffer. Once all the materials transfer to the machines, moves back to the Depot (where the vehicles are stayed after completing the job).

This Method triggers on the track near the machine to load and unload the materials from the AGV to the buffer and from machine to the AGV. When the AGVs moves on the track near the buffer this method activates and then transfer the materials from AGV to the buffer. Once all the materials transfer to the machines, moves back to the Depot (where the vehicles are stayed after completing the job).

Machine_Load:



```
is
    This_table: table;
    This_part,This_station : object;
    i :integer;
do
    This_table.create;
    This_part:=@;
    This_station:=?.succ(1);
    waituntil This_station.occupied=false prio 1;
    This_table:=@.Production_Order;
    ?.succ(1).setuptime:=str_to_time(to_str(This_table[1,2]));
    ?.succ(1).proctime:=This_table[2,2];
    for i:= 1 to 2 loop
        This_table.cutrow(1);
    next;
    @.move(?.succ(1));
end;
```

Figure 10. Machine_Load

This Method (Machine_Load) which triggers on Buffer near to the Machine. When the parts load on the Buffer, the set-up time and processing time of the particular part is to be taken from the "Myworkplan" table and then move on to the Machine.

VI. ROUTING AND DRIVING PROPERTIES

The Transporter moves along the object Track with a speed we set, forward or in reverse. The Length of the Track and the Speed of the Transporter result in the time the Transporter remains on the object. Then, the Track moves the Transporter on to its successor.

Transporters may not pass each other on Tracks. If a faster Transporter collides with a slower one, it automatically reduces its speed to that of the slower Transporter. As soon as the obstacle is removed, the Transporter increases its speed again to the value we set.

During failures, the Transporter stops and continues on its own, when the failure does not exist any more. When a Track has several successors use any of the control methods to move the Transporter: The destination control, the destination lists, automatic routing, the driving control and the exit strategy.

Destination for the Transporter:

The Transporter automatically finds the destination object using the destination lists of the objects of type Track or TwoLaneTrack. If activate Automatic Routing in the tab Attribute, do not need to define destination lists. The Transporter will automatically find its destination object.

For the destination lists we have to enter a table or a list. This list or table can be a TableFile, a CardFile, a QueueFile or a StackFile, which we inserted into a Frame. It can also be a user-defined attribute or a Variable of type table, list, queue, or stack. The list or table has to have the column type object. The Transporter then covers all destination objects in the list or table one after the other.

TableFile (table): Plant Simulation cyclically processes the TableFile. The Transporter always heads for the destination object, which is located in the table row to which the attribute DestinationListRow of the Transporter points.

CardFile (list): Plant Simulation treats the CardFile as a TableFile with one column.

QueueFile or StackFile (queue or stack): The Transporter always drives to the first object in the QueueFile or the StackFile. As soon as the destination is reached, Plant Simulation removes the first object from the list. The Transporter then drives to the new first object in the list.

VII. RESULTS AND DISCUSSION

Chart:

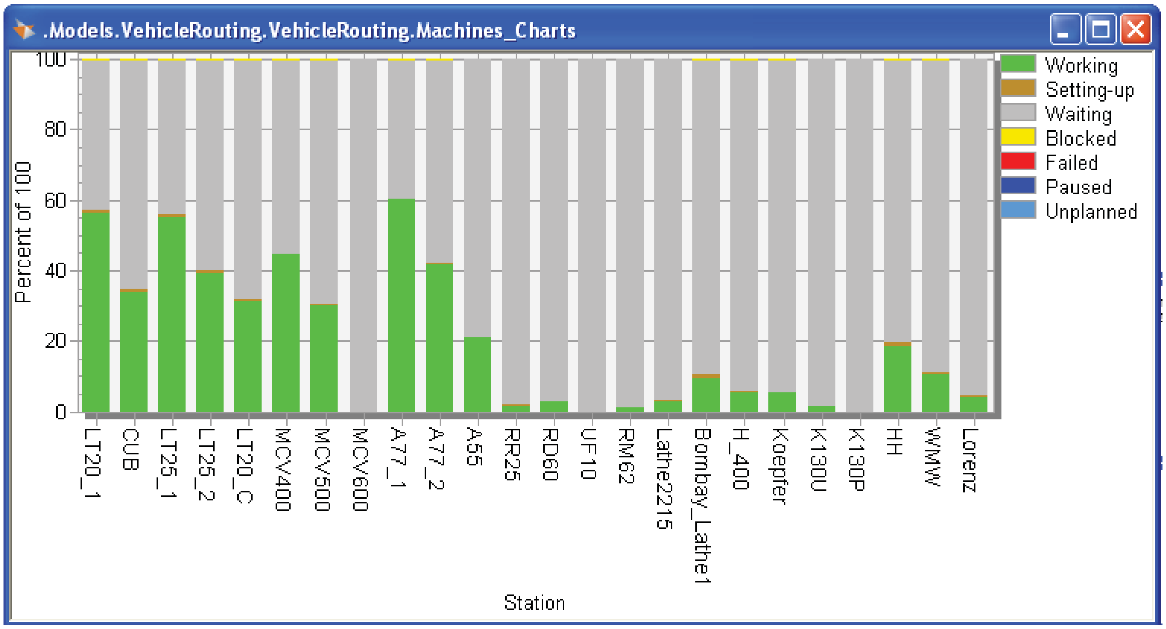


Figure 11. Machine Utilization_Chart

Sankey Diagram:

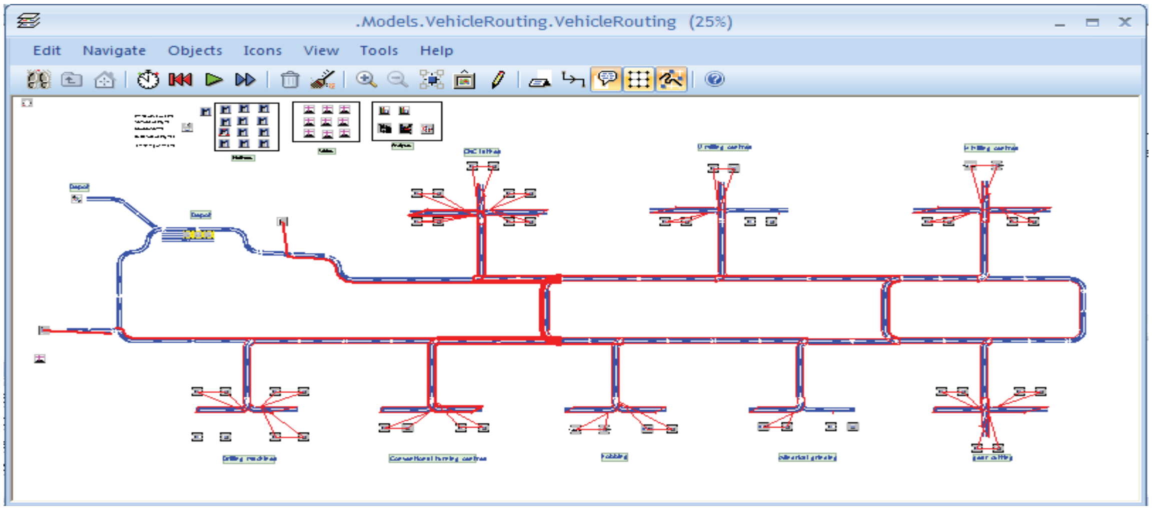


Figure 12. Sankey Diagram

Bottleneck Analyzer:

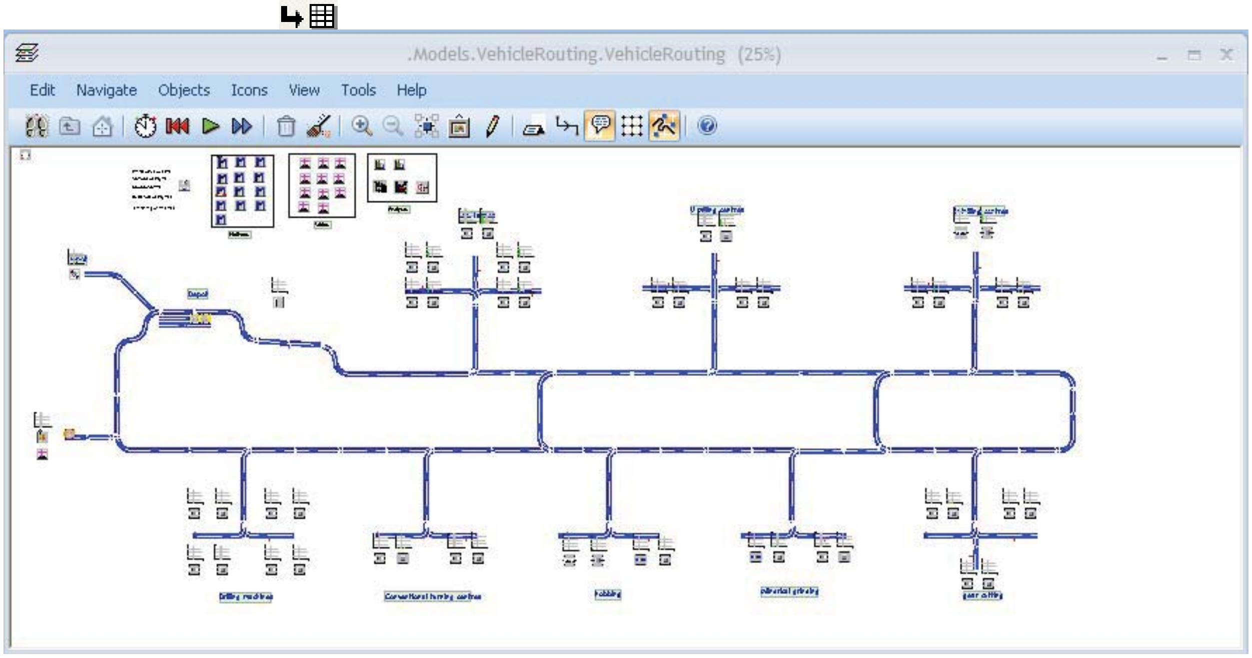


Figure 13. Bottleneck Analyzer

Loop and Line Layout:

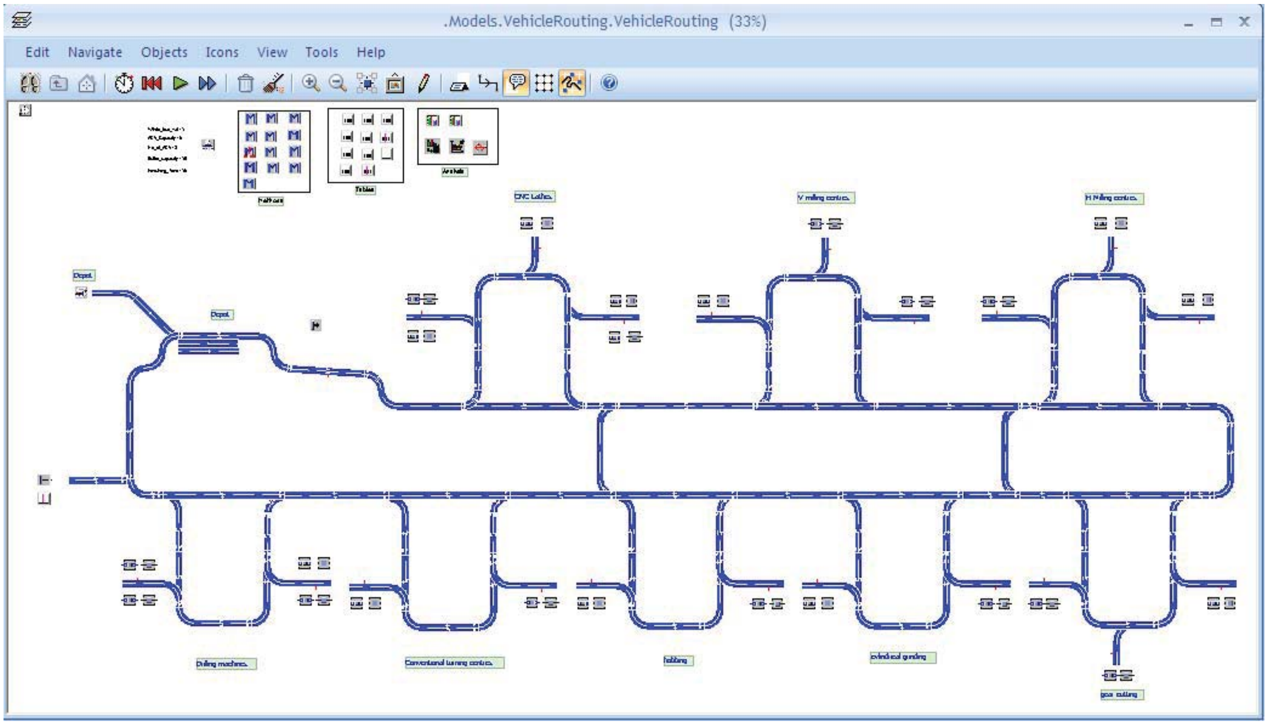


Figure 14. Loop and Line Layout

The above figure shows that material handling using loop and line layout. There are some changes which has been done compare to the previous layout. In this layout, added loop for each workstations to reduce the travel distance of the vehicle and to reduce the lead time.

Rank-Order Clustering:

Rank-Order clustering was developed by J.R.King (1980). This algorithm considers the following data:

- Number of components
- Component Sequences

This algorithm is used to obtain machine-component cells in group technology. Based on the component sequences, a machine-component incidence matrix is developed. The rows of the machine-component incidence matrix represent the machines which are required to process the components. The columns of matrix represent the component numbers.

VIII. ANALYSIS OF SIMULATION RESULTS BEFORE AND AFTER GROUPING

Working Time (%):

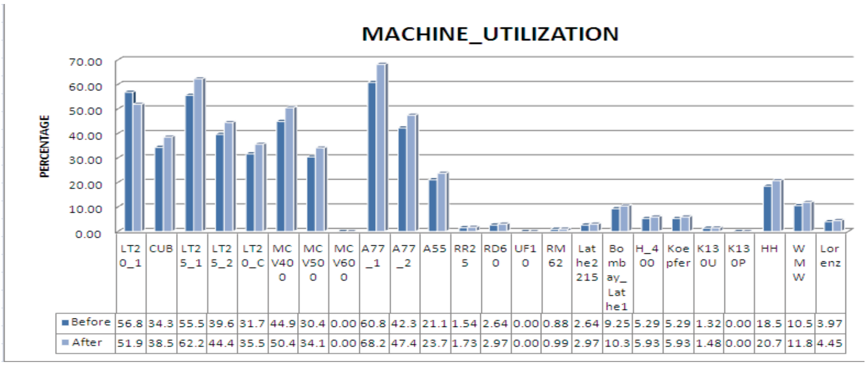


Figure 15. Before and After Group of each Machines_Working

Figure 15. shows the machine utilization (Working Time) before and after grouping. The Machine A77_1 has the highest utilization of 60.81 % and the machine RM62 has the lowest utilization of 0.88 % before grouping. For the same machines after grouping the utilization has changed to 68.22 % for A77_1 and 0.99 % for RM62.

Waiting Time (%):

Figure *16.* shows the machine utilization (Waiting Time) before and after grouping. The Machine RM62 has the highest utilization of 99.03 % and the machine A77_1 has the lowest utilization of 39.12 % before grouping. For the same machines after grouping the utilization has changed to 98.91 % for RM62 and 31.70 % for A77_1.

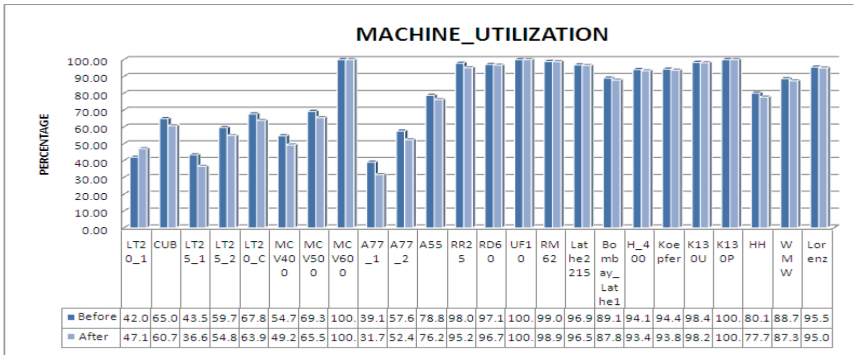


Figure 16. Before and After Group of each Machines_Waiting

Figure 16. shows the machine utilization (Utilization Time) before and after grouping. The Machine A77_1 has the highest utilization of 60.88 % and the machine RM62 has the lowest utilization of 0.97 % before grouping. For the same machines after grouping the utilization has changed to 68.30 % for A77_1 and 1.09 % for RM62.

IX. CONCLUSIONS AND FUTURE SCOPE

Analysis of machine shop layout and material flow is a complex activity in the industries. The use of suitable scheduling and routing technique improves the performance of the system. In the present work, using object oriented Discrete Event Simulation software with several optimization tools the problems associated with the machine shop layout and material flow have been systematically discussed. The method adapted identifies Bottleneck Stations, computes waiting times and station utilization under two different layouts. The resource statistics have been computed for with and without material handling.

The following work may advance current status of the problem:

- *Communication Problem:* In case of communication system fails and AGV cannot pass signal to central controller, AGV will block the path. This blockage causes the complete system blockage. Communication failure can be studied for further expansion.
- *AGV routing on a non-uniform path layout:* AGVs can be routed along the bidirectional path layout where the distance between the adjacent workstations may not be uniform. In this case, the speed of the vehicle must be synchronized, so that the time required in traveling a distance between any two adjacent workstations will be the same.
- *Continuous AGV Routing:* In the proposed model, if loading and unloading time is uniform across different P/D jobs, the times taken for these operations are characterized by the most time consuming job. This decreases the routing efficiency.
- *Routing AGVs on mesh topology:* In the model, AGVs are routed along the bidirectional path. The model can be extended to route the AGVs along a complex path layout like mesh topology, which is a rectangular array of linear path.

REFERENCES

[1] Ouelhadj D., Petrovic S, “Survey Of Dynamic Scheduling in Manufacturing Systems”, Automated Scheduling, Optimization and Planning Research Group, School of Computer Science and IT, University of Nottingham, Nottingham NG8 1BB, UK.

[2] Albert Jones, Luis C. Rabelo, “Survey of Job Shop Scheduling Techniques”, National Institute of Standards and Technology, Massachusetts Institute of Technology.

[3] Muhammad F. Anwar, Rakesh Nagi, “Integrated Scheduling of Material Handling and manufacturing Activities for Just-In-Time Production of Complex Assemblies”, Department of Industrial Engineering, 342 Bell Hall State University of New York at Buffalo, Buffalo, NY 14260.

[4] T. C. E. Cheng and Q. Dinga, European Journal of Operational Research, Department of Management, the Hong Kong Polytechnic University, Volume 152, Issue 1, 1 January 2004, Pages 1-13.

[5] Adams, J.,E.Balas and D.Zawack, “The shifting bottleneck procedure for job schedling”, Management Science, Page 391-401.

[6] Srinivasan.V, “A hybrid algorithm for the one machine sequencing problem to minimize total tardiness”, Naval Research Logistics Quarterly, Pages 317-327.

[7] Davis. W and A. Jones, “A real-time production schedule for a stochastic manufacturing environment”, international journal of Computer integrated Manufacturing, Page 101-112

[8] Benders. J , “Partitioning procedures for solving mixed-variables mathematical programming problems”, Numersche mathematic, Page 238-252.

[9] Dantzig. G and P. Wolf, “Decomposition principles for linear programs”, Naval research logistics quarterly, Page 101-111.

[10] Nebil Buyurgan, Lakshmanan Meyyappan, Can Saygin, Cihan H. Dagli, Journal of Manufacturing Technology Management, ISSN: 1741-038X, Volume: 18, Issue 2, 2007, Pages 169 – 181.

[11] Ewgenij Gawrilow, Ekkehard K¨ohler, Rolf H. M¨ohring, Bj¨orn Stenzel, Dynamic Routing of Automated Guided Vehicles in Real-time, October 10, 2007.

[12] Tatsushi Nishi, Distributed Optimization Method for Simultaneous Production Scheduling and Transportation Routing In Semiconductor Fabrication Bays, Graduate School of Engineering Science Osaka University, ISSN 1349-4198, Volume 4, Number 3, March 2008, page 559—575.

[13] Björn Johansson, Edward J. Williams, Tord Alenljung, Using Autonomous Modular Material Handling Equipment for Manufacturing Flexibility, 2004.

[14] Ling Qiu; Wen-Jing Hsu; Shell-Ying Huang; Han Wang, “Scheduling and routing algorithms for AGVs: a survey”, International Journal of Production Research, Volume 40, Issue 3 February 2002 , pages 745 – 760.

[15] Lawrence Henesey, 2 Paul Davidsson, 1 Jan A. Persson, “Evaluation of Automated Guided Vehicle Systems for Container Terminals Using Multi Agent Based Simulation”, Department of Systems and Software Engineering, Blekinge Institute of Technology, 1Box 214, 374 24 Karlshamn, Sweden, 2Box 520, 372 25 Ronneby, Sweden.

[16] Reijer Idema, “Optimal Path Synthesis for Automated Guided Vehicles, Preliminary Research”, February 15, 2005.

[17] Sanjay S. Gurav, *“Route Planning of Automated Guided Vehicles for Container Logistics”,* University of Pune, 2000 May 2004