University of Windsor
Scholarship at UWindsor

Industrial and Manufacturing Systems Engineering Publications

Department of Industrial and Manufacturing Systems Engineering

2007

An Approach to Line Balancing on Virtual Supervisor Induction Method and Intelligent Agents

J. Trujillo

P. Pawlewski

Zbigniew Pasek
University of Windsor

P. Golinska

E. Baeyens

See next page for additional authors

Recommended Citation

Trujillo, J.; Pawlewski, P.; Pasek, Zbigniew; Golinska, P.; Baeyens, E.; and Oleskov, J.. (2007). An Approach to Line Balancing on Virtual Supervisor Induction Method and Intelligent Agents. *19 th International Conference on Production Research*.
http://scholar.uwindsor.ca/industrialengpub/3

Authors
J. Trujillo, P. Pawlewski, Zbigniew Pasek, P. Golinska, E. Baeyens, and J. Oleskov

AN APPROACH TO LINE BALANCING ON VIRTUAL SUPERVISOR INDUCTION METHOD AND INTELLIGENT AGENTS

J.Trujillo*, P. Pawlewski, Z. Pasek***, P. Golinska**, E. Baeyens*, J. Oleskov****

*Dep.Automation and Systems, University of Valladolid, E47008, Valladolid, Spain

**Institute of Management Engineering. Poznan University of Technology, 60-695 Poznan, Poland

***Dep Mechanical Engineering,2250 Hayward street, University of Michigan, MI48109, USA

Abstract

This approach develops a method for solving the line-balancing problem, which is based on two stages. The works in a first stage is to identify the task of workstation, the assignment of the tasks to stations on the line and the recognized balance delay. In this stage we propose the induction VS method, which allows further identify the exact position between pieces, machine into a workstation and also between extern workstation, as well as intracellular and intercellular part. This way each task is identified and measured. In the second stage is to carry out a macro-approach to choose the resource to perform each of them. The hybrid intelligent agent architecture is proposed for this second stage, which has consideration of machining sequence. The integration between both technologies allows us to develop new hybrid architecture capable to reduce the computational time in the deliberative layers fundamentally. Finally, a reconfigurable testbed has been proposed for future experiments and results to evaluate this new balancing method. Some previous computational experiments provide that the proposed approach is efficient to solve practical transfer line design for balancing problems

Keywords:

Balancing line, Job-shop floor, Scheduling management, Intelligent Agents, Reconfigurable manufacturing

1 INTRODUCTION

The line balancing problem is one of the most important problems of preliminary design stage for flow line production systems [2,3,5], and a complete solution algorithm for solving it does not exist. This problem was generally studied for Hybrid assembly lines with a relatively simple structure and frequently with multiple objectives [8]. The works in a first stage is to identify the task of workstation, the assignment of the tasks to stations on the line and the recognized balance delay. In this stage we propose the induction VS method [8,9]. This method allow further identify the exact position between pieces [9], machine into a workstation and also between extern workstation. This form each task is identified and measured. In the second stage is to carry out a macro-approach to choose the resource to perform each of them (task of workstation).

The intelligent agent structure is proposed for this second stage, which has consideration of machining sequence [7]. We will present three approaches for agent encapsulation: (i) functional decomposition approach where agents are used to encapsulate modules assigned to functions such as planning, etc. There is no explicit relationship between agents and physical entities [10,11,12,13] (ii) physical decomposition approach where agents are used to represent entities in the physical world, such as workers, machines, tools, and operations, etc. There exists an explicit relationship between an agent and a physical entity.[14,15,16,17,18] (iii) process decomposition approach where agent are owners of key company processes. The number of agents is equal key processes and existing agents can be used by several process owners [19]. We show above three methods based on the implementation of VS induction method. These methods could be allowing the reconfigurability of process [1], because specific tasks are done for same family of product with equivalent machines. A Reconfigurable Manufacturing systems (RMS) is designed at the outset for rapid change in structure, as well as in hardware and software components, in order to quickly adjust production capacity and functionality within a part family in response to sudden change in market or regulatory requirements [1]. The final goal is to minimize the total cost of the line by integrating design [4,6], such as congestion, machine real cost principally, and operation issues as cycle time, precedence constraints and availability. Therefore the model sequencing problem for allocating the workstation on minimizing inter-workstations movement distance unit [9] will be investigated in this paper.

2 NOTATIONS AND PROBLEM FORMULATION

2.1 Notations and preliminaries

The generic balancing problem is extremely complex [3],

The following notation is used for modeling the design *N* is the set of all operations needed to manufacture a B is the set of blocks (spindle heads) which can be used *m* is the number of workstations in considered line for a *nk* is the number of blocks of workstation *k; Ql* is the basic cost of one Workstation; *Nu* is the set of operations of block I of workstation *k; Nk=(Nki,....Nknk)* is the set of blocks from B which problem considered new products for this line; design decision; are executed *at* the workstation *k,* Pr(NkI) is the set of operations which must be executed before any operation from Nis: *Q&Vd* is the cost of the additional block NN. We assume also that the designed line cannot involve more than mo workstations and each workstation at most *no* blocks.

2.2 Problem Formulation

The line cost for design decision ***P*** can be estimated by the following expression:

$$Q(P) = Q_l m + \sum_{k=1}^{m}\sum_{l=1}^{nk} Q_2(N_{kl}) \quad (1)$$

The consmints introduced in Section 1 can be represented as follows:

a) A partial order relation over the set N is represented by the acyclic digraph G=(N,D). An arc(I,j)єNxN belongs to the set D’ iff the operation j must be executed after the operation i.

b) Since all blocks of the same workstation are executed simultaneously, the blocks with the block time over required line cycle time can be excluded from B before optimization. Then the constraint (2) from previous section can be omitted using this transformation.

c) Exclusion conditions for the blocks of the same workstation can be represented by the graph. $\overline{G}^s = (B, \overline{D}^s)$ such that a pair $(N', N'') \in BxB$ belongs to the set $\overline{D}^s$ if blocks N’ and N” cannot be allocated to the same workstation.

d) Inclusion conditions for the operations of the same workstation can be represented by the graph $G^d = (N, D')$ such that a pair (i,j) $(i, j) \in NxN$ belongs to e the set D^s if operation I and j must be allocated to the same workstation.

Thus, the design problem can be reduced to finding a collection

$$\bigcup_{k=1}^{m}\bigcup_{l=1}^{nk} N_{kl} = N \tag{2}$$

$$N_{k'l'} \cap N_{k''l''} = \varphi, \ \ for\ k'l' \neq k''l'', \ \ k', \\ k'' = 1,\ldots,m, k' = l,\ldots,n_{k'}, l'' = 1,\ldots,n_{k''} \tag{3}$$

$$\Pr(N_{kl}) \subseteq \bigcup_{r=1}^{k\ 1}\bigcup_{q=1}^{n_r} N_{rq}, k = 1,\ldots,m, l = 1,\ldots,n_k \tag{4}$$

$$\bigcup_{l=1}^{nk} N_{kl} \cap e \in \{\varphi, e\}, \ \ for\ e = \{ij\}, (ij) \in D^s, k = 1,\ldots,m \tag{5}$$

$$(N_{kl'}, N_{kkl''}) \quad \overline{D}^s, k = 1,\ldots,m, l', l'' = 1,\ldots,n_k \tag{6}$$

$$n_k \leq n_0, k = 1,\ldots m, \tag{7}$$

$$m \leq m_0 \tag{8}$$

where the objective function (I) minimizes the line cost; constraints (2)-(3) determine the condition of assigning all operations of the set N and including each operation into one block-only; (4) defines the precedence constraints over the set N; *(5)* determine the necessity of executing the-corresponding operations at the same workstation; (6) define the possibility of combining blocks at the same workstation; (8) - (9) provide the constraints on the number of workstations and blocks for each workstation.

3 SOLUTION APPROACH

3.1 Induction VS method

The increment of Induction of potential through resistance of diffusion is equivalent to distance between machines into a number of cells and also into a number of parts families as well. This increment is based on taking the advantages of part similarity in processing and design functions. This analogy is adequate to describe the interaction between machines and parts intracellular and intercellular. Two o more machines interact when a part is transported by the conveyor between them. It either arrives on time as expected, or with delays. It can be described by impedance of electrode/cell. This model enables dynamic modeling of the behaviour of manufacturing process on line [20]. This method allow further identify the exact position between pieces, machine into a workstation and also between external workstation. This form each task is identified and measures based on Maxwell equation presented in our previous work [8,9,20]

3.1.1 Stage one. Maxwell approach

In general, all methods of calculus of diffusion resistance assume simplified hypothesis similar to this case. An issue appearing in using these methods in manufacturing applications, which are big distances involved (size of a manufacturing plant). One approach to reduce the extent of the problem is to subdivide the cell so that modeling resolution is increased, which also yields for higher sensitivity to influencing potentials [20].

Hypothesis (1): The intensity that for unit of length, adds every element is constant for the element, but differently of a few element to others (each others).

The potential of any k element is the sum consisting of potentials induced by other elements, self induced potential of element k, and of all their electrical images, where element means is the distance unit in the grid. This grid distance is scaled to the distance between machines or parts intracellular and intercellular way. For one k element:

$$v_k = \sum_j v_{kj} \quad j = 1,2,\ldots,2n \tag{9}$$

The VS shop floor is created base on criteria specified in (Trujillo at al. 2006) as grid structure. This grid shop is scaled and also the bound areas must be specified for futures movements under future reconfigurations, Fig1 shows this grid shop floor.

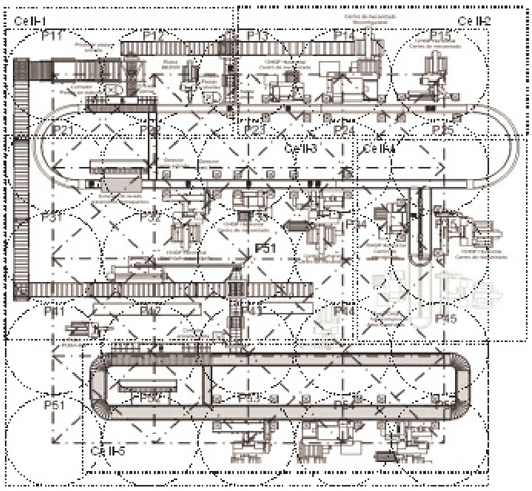

Figure 1: Grid-Layout shop Floor

When the cell is divided into n stretches or simple elements, a rectangular curl of dimension *L x W* can be formed, which lends itself well to calculus of resistance, potentials, etc [21]. The electrode is divided to n stretches of similar longitude Li, each one carrying a charge qi, with charge density of qi/Li. The total electrode charge is: $Q = \sum qi = \rho \in I, \ i = 1,\ldots,n$ Let’s consider the absolute potential of an electrode (a machine is equivalent to a cell formed by electrodes); it is unique and similar to the potential for each of *n* stretches.

$$v_0 = v_k \ \ k = 1,2,\ldots n \tag{10}$$

The resistance of diffusion equivalent could be defined as:

$R_m = \frac{\sum_j v_{kj}}{I}$ where I is the intensity that each electrode dissipates. The potential v_{kj} at any points k produced by a stretch j, has a longitude L_j with an associated charge q_j / L_j. These potentials can be calculated by a Maxwell relation. The potential in any k element $v_k = \sum_j v_{kj} \quad j = 1,2,...,n$ is estimated for all possible k. The Vo is n average potential for each j of n stretch or element.

$$v_0 = \frac{\sum_k \sum_j v_{kj}}{n} \quad k = 1,2,...,n \quad j = 1,2,...,2n \tag{11}$$

the resistance of diffusion in the cell is

$$R_e = \frac{\sum_k \sum_j v_{kj}}{In} \quad k = 1,2,...,n \quad j = 1,2,...,2n \tag{12}$$

where I_j and L_j are respectively the dissipated intensity by each element and their longitude.

$$v_k = \frac{\rho}{4\pi} i_0 \sum_j \int_{L_j} \frac{dL_j}{\left|\vec{u}_k - \vec{u}_j\right|} \Rightarrow \phi = \frac{\rho I}{4\pi L} \int_L \frac{dL}{\left|\vec{u} - \vec{u}'\right|} \tag{13}$$

Algorithm implementation requires the matrix, obtained from eq. (2-5):

$$[T] = [p^0]^{-1}.col[i] \text{ , } [I] = U_0 (4\pi / p)[T]col[i] \rightarrow$$
$$R = (p / 4\pi)(i / [T]col[i]) \tag{14}$$

when the cell/line is parallel to OY axis and is between the coordinates $(x_1, y_1, z_1),\ (x_2 = x_1, y_2, z_2 = z_1)$,

$$\phi = \frac{\rho I}{4\pi L} \ln \frac{y_r - y_1 + p_1}{y_r - y_2 + p_2} \tag{15}$$

The same way when the cell/line is parallel to OY and OZ axis.

Hypothesis (1) and Eq. (1,2,3,4, 5 and 6) allow us to model a lineal relation between the distance and the weight acquired by the priority when parts the same family could be performed in several machines the same group. ф represent the weight equivalent (ζ). The potential in R is the sum of cell potentials and electric image. Eq. (7,8,9) are equivalent to the weight it means priority between cells, and similarly with the parts. This way the VS method could define the sequence most required by the operation flow. For the MPGP, parts possessing similar machining features and the corresponding machines will be grouped together. Here we would find out the number of voids which indicates the total unnecessary intracellular part movement in a flow-line configuration; and number of exceptional element which on the other hand indicates the total intercellular part movement. Please be reminded that voids represent idle machines as a part does not require one of the machines in its corresponding cell, in other words, it means unnecessary intracellular part movement under a flow-line configuration. Eq. (10) shows the total number of voids inside the FPM-blocks [21].

$$N_a = \sum_{k=1}^{N} \sum_{j=1}^{P} \sum_{r=1}^{Rp_j} w_{ijr} \sum_{k=1}^{N} \left| x_{ik} - y_{jkr} \right| \Big/ 2 \tag{16}$$

Therefore the ultimate objective is to minimize

$$V_f = N_a(\zeta_a + \psi_a) + N_b(\zeta_b + \psi_a + \psi_b) \tag{17}$$

$$\sum_{k=1}^{N} X_{IK} = 1 \text{ for } i \in \{1,2,\ldots,M\} \tag{18}$$

$$\sum_{r=1}^{Rpj} \sum_{k=1}^{N} y_{jkr} \text{ for} \forall j, \tag{19}$$

where,

$$x_{ik} = \begin{cases} 1 & \text{machine i is assigned to cell K,} \\ 0 & \text{otherwise} \end{cases}$$

$$y_{ik} = \begin{cases} 1 & \text{machine j is assigned to cell K,} \\ 0 & \text{otherwise} \end{cases}$$

where, the first term represents the total number of elements inside the diagonal blocks, and the later term represents the total number of workload elements inside the diagonal blocks. On the other hand, the total number of exceptional elements outside the diagonal blocks which representing the intercellular part movement is calculated by Eq.(17). Therefore, the ultimate objective is to minimize From Eq.(18), we can fulfil a number of objectives which are:

a) to minimize the total intracellular part movement by minimizing the total number of voids inside the diagonal blocks;
b) to minimize the total intercellular part movement by minimizing the total number of exceptional elements outside the diagonal blocks. Thus, the productivity can be enhanced;
c) to maximize the machine utilization. The lessening of voids reduces the part movements across the different machines within a manufacturing cell, thus improving the utilization of the machines. VS induction method can recognize several cell influence by others and allow us a fast configuration when the priority obtained allows that. The efficiency of each manufacturing cell can thus be improved;
d) moreover, from Eqs.(16) and (17), it shows that the proposed model is able to handle more than one routing (alternative routings). Eq. (18) restricts that each machine is allocated to only on cell. Eq. (19) ensures that only one routing is selected for each part, and that part is belonging to only one cell.

3.2 Intelligent Agent Methodology

In the second stage is to carry out a macro-approach to choose the resource to perform each of task of workstation. This stage approach is Based on Intelligent Agent methodology for agent encapsulation:

a) functional decomposition approach where agents are used to encapsulate modules assigned to functions such as order acquisition, planning, scheduling, material handling, transportation management, and product distribution. There is no explicit relationship between agents and physical entities.
b) physical decomposition approach where agents are used to represent entities in the physical world, such as workers, machines, tools, products, features, and operations, etc. There exists an explicit relationship between an agent and a physical entity.
c) process decomposition approach where agent are owners of key company processes.

3.3 Interaction between both methodologies

The hybrid architecture created for VS induction method and intelligent agents, who are formed by a deliberative and reactive layer allow us to code a general procedure for finding a solutions to a class of problems in advance. Induction method allows identify the exact position where the pieces, machines, Cells with their job areas and also identify the plan which is being executed. a deliberative architecture will typically be more space efficient than an equivalent reactive architecture since it can solve a class of problems in a fixed amount of space, whereas a reactive architecture requires space proportional to the number of problems. Deliberative layer uses complex counterfactual representations, for example representations of objects and their attributes in world coordinates. Figure 2 shows the reactive layers part, which interacts with deliberative layers parts from VS induction method.

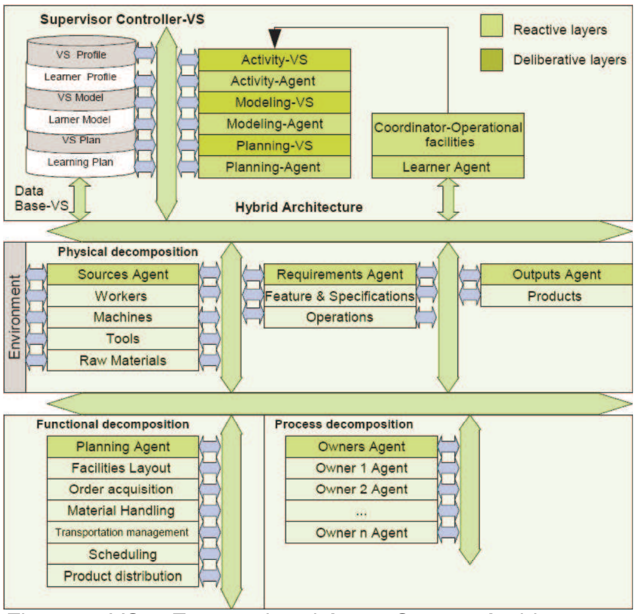

Figure 2: VS & Encapsulated Agent System Architecture

Sometimes, deliberation should suppress reactive behaviors and sometimes it shouldn't:, Induction method which allow that this situation do not occur, because the map obtained by this method allows recognize the interaction between several workstations, pieces, components, equipments, etc, before the interaction be performed. Except some unpredictable events, which are impossible to determine their consequences previously.

The reactive layer produces an immediate response to changes in the environment. This is responsible for things, such as obstacle avoidance, also this implemented as a set of condition action rules which map percepts directly to actions [21]. The sequence of action represents the operations or tasks required by the processes. They are building a predictable state space in deterministic systems. Rules can only refer to the agent is current state and they can not do any explicit reasoning about the world. The Figure 2 shows several modules which are layers as the planning layer was responsible for achieving simple goals, e.g., moving from place to place, thus did not generate plans from scratch, and it was implemented as library of plan schemas, which are elaborated at run time to achieve a goal; the planning layer attempts to find a schema that matches that goal; a schema may contains subgoals, which the planning layer attempts to elaborate by attempting to find other schemas in the plan library that match the sub-goals. Modelling layer: the modelling layer was responsible for representing other entities (agents) in the world, including the agent itself, since the modelling layer predicts conflicts between agents and generates new goals to resolve these conflicts; hence these goals are passed to the planning layer which plans to achieve them in the normal way.

4 SYSTEM OPERATION CRITERIA

The process works as follows: each workstation or machine, process and part are assigned a level of resistance using coefficients and previously described conditions, see the details in [8,20]. The potential induced by each workstation or machine depends on these coefficients and conditions, *e.g.* for a workstation on a critical flow path the induced potential will be high, since a critical path has high priority. The resistance is assigned as follows:

$$Rec = PkPj\ vkj\ \ln((1+\varsigma)(1+\zeta)(1+x))e{-}(x+1)2$$
$$k = 1, 2, \ldots, n,\ j = 1, 2, \ldots, 2n \qquad (10)$$

where the coefficient ς reflects the path criticality and its value depends on priority level. The inductor is a manufactured workpiece affected by other coefficients:

$$v0 = PkPj\ vkj\ n\ (1+Wj)(1+fj)(1+sj)e{-}(x+1)k = 1, 2, \ldots, n\ j = 1, 2, \ldots, 2n \qquad (11)$$

where Wj and fj are the priority or weight of tardiness and earliness penalty for part j, respectively, and sj is the index of the possible substituted machines. The total intracellular and intercellular part is minimized, and also the machine utilization has been maximized; hence the efficiency of each manufacturing cell can thus be improved. The Supervisor Controller VS performs process for activities, modeling and planning. Activity Agent is performed using the reactive layers partial order planner: integrates new asynchronous requests into the current plan, prioritizes tasks, opportunistically achieves compatible tasks, which determine the order in which to interleave the actions required for each task, this consults the path planner to determine the expected travel time between two locations. Modeling Agent determines how to travel efficiently from one location (cell or workstation) to another: uses a decision theoretic approach to choose plans with high expected utility, this uses sensitivity analysis to determine which alternatives to consider actuator and sensor uncertainty complicates path planning.

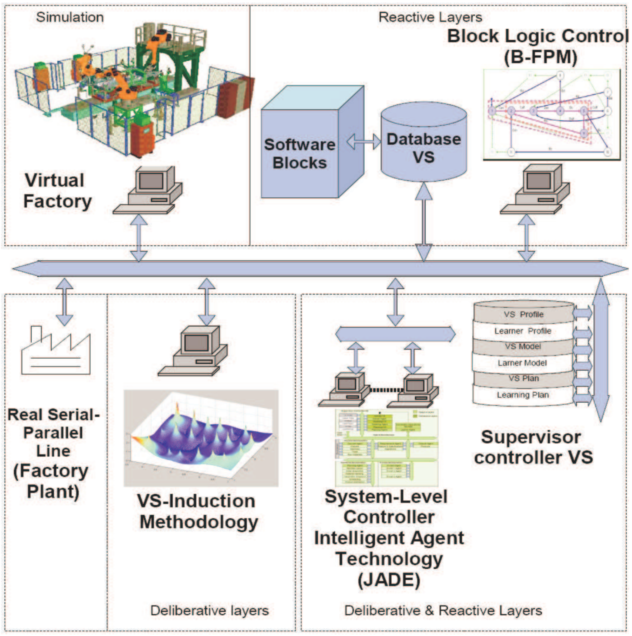

Figure 3: Integration of VS Induction & E. Agent System in Reconfigurable Testbed Schematic

Planning Agent the pieces may not be able to follow a path accurately the shortest distance path is not necessarily the fastest dead-time avoidance is performed using a curvature velocity method keeps the flow in the desired time, while avoiding static and dynamic overtime or tardiness and dynamics into account, real time optimization problem that combines safety, speed and progress along the desired heading.

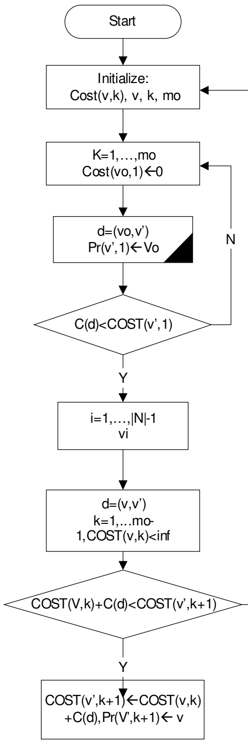

Figure 4: Flow chart of proposed Algorithm 1, recognizing the Effective line

The learner Agent generates an adaptative interface, where the agent dynamically assembles personalized instructional materials in terms or reading contents, (sequence orders) given by control logic shown from Finite position machine Block [21], quizzes and feedback for particular online learner based on the learning plan. Such an assembling process includes the generation of quizzes, quizzes summary and instant messages.

The following algorithm 1 shown in figure 4 allows to solve finding the shortest path in block structures, which are based on finite position machine Block FPM-B [21], with at most mo arcs or actions. In this algorithm, COST(v,k) calculates the minimal cost path in FPM diagram G, such as a solution of the problem. The set V is partioned into subsets Vi. Obviously there are no arcs in G FPM diagram between vertices of Vi and vertices of Vj for i>=j. The algorithm-2 shown in Figure 5 converts the precedence relations to an state space matrix. While tasks are represented by transitions of actions, precedence relations and availability information are symbolized by places. The state space matrix obtained from the precedence relations is given from a deterministic FPM, denoted G, is a tuple of ten elements.

$G = (P, A; T, \rho, \varsigma, \tau, P_{Gs}, P_{Gec}, P_{Gic}, P_{Gvm})$ produces the language L(Gv) and marks the language Lm(Gv). Where: P is the set of all positions, A is a set of all the actions, ρ is the response of partial transition ρ : P $\times$ A $\rightarrow$ P or is the response of partial sequence ς: is the cyclic transition function or sequence of cyclic response Gv, ς : P $\times$A/T $\rightarrow$ P, τ_{Gvo} is the set of transition action triggered from Gv in position p $\in$ PGv, is defined PGsa initial position of complete sequence in an acyclic processing, PGec initial position of external cycle Gv, PGic initial position of internal cycle Gv, Pm is the set of marked positions, Pm $\subset$ P. All details are in [21]. To start the transition of action, an initial marking must be defined. In our example, one token is placed into source places p1 and p2 to begin the dynamic study. Therefore, the first two elements of Mo are MnDi, MnTm, MnTi keeps the workstation number, assigned task number, assigned task time, distance between workstation, distance between pieces and workstations, and idle time. The algorithm starts searching the available tasks for the second assignment. According to the new action condition, task 1 (A1), task 2 (A2),..., task n (An), are the new available tasks for the first assignment to workstation n. The steps given above are repeated until all tasks are assigned to the stations. The final solution is obtained when the minimum time and distance allow us to know the minimum number of workstation for taking jobs. Therefore arrival time constraints and operation precedence are effectively managed.

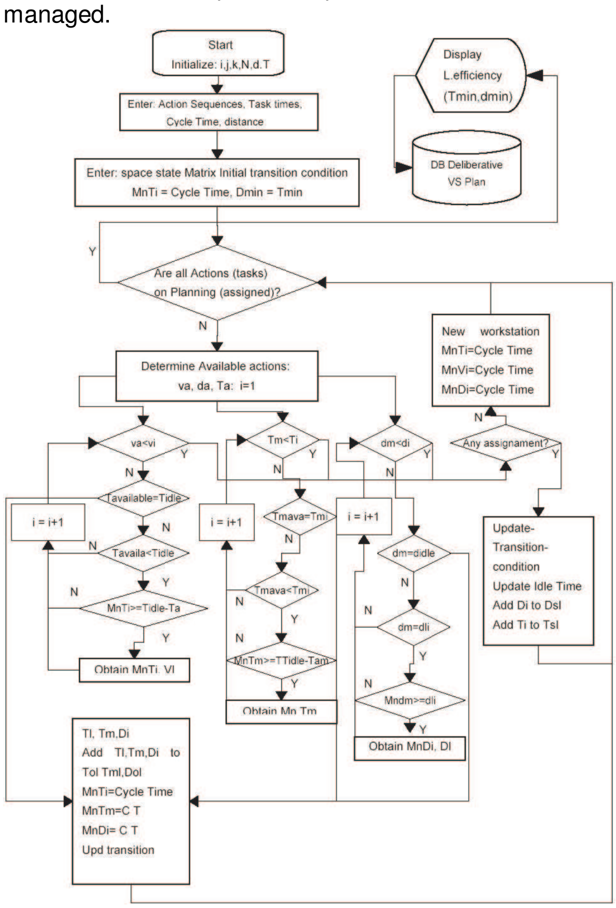

Figure 5: Flow chart of proposed Algorithm 2, recognizing the Effective line

5 EMPIRICAL EXPERIMENT

VS-Induction can build in advance the estimated situation in a virtual space, where it recognizes by induction how each workstation or machine is working, it means the exact place where the piece are in Time. VS-induction recognizes earlier and tardiness in their specific positions, in many cases with computational time enough to implement a deliberative Plan, activity and model. Therefore VS-induction works in deliberative layers and provides information to build the specific agent for controller. This way is achieved an important reduction of time required for recognition in a typical scheduling. VS induction works for each reactive agent layer, such as Activity, Modeling and Planning, which providing the deliberative functions. In this approach, our architecture has to be capable not only of identifying exception on line, but also simultaneously developing on line strategies for unpredictable customer orders or inaccurate estimates of processing time. We are developing a Reconfigurable Testbed shown in Figure 5, which allow us to improve this Architecture with experiments and results.

6 CONCLUSION AND FUTURES WORKS

A novel architecture that balances modeling accuracy and solution methodology complexity is presented. Satisfaction of arrival time constraints and operation precedence are effectively managed. Previous simulated testing results demonstrate that the method can be substantially better than those used today, and near optimal schedules could be generated for problems of practical size. The interaction between these different methodologies such VS induction as hybrid Agent system allow to create the deliberative layers reducing time and identifying exact positions on line, these together the encapsulated reactive layers allow create Supervisor Controllers capable to learn the sequential order for task (jobs) in each workstation for balancing problem for a class of production lines with blocks of parallel operation, which has been a principal goals in this research. The results in test and algorithm presents in this paper will be developed in detail for future works which will must pass the correspondent benchmark for feasible results. Also the handling of unpredictable machine breakdowns is also an important issue, this falls directly into the current architecture. These strategies will allow us to observe performance results during simulation.

7 ACKNOWLEDGMENTS

This work has been supported thanks to collaboration between University of Valladolid, Poznan University of technology and University of Michigan.

8 REFERENCES

[1] Koren, Y., *et al.*, (1999), "Reconfigurable Manufacturing Systems," Annals of the CIRP, Vol. 48/2, p. 527-540.

[2] Rekiek B., Dolgui A., Delchambre A. and Bratcu A., "State of art of optimization methods for assembly line design", *Annual Reviews in Control,* Volume 26, Issue 2, , Pages 163-174, 2002.

[3] Dolgui, A.; Guschinsky, N.; Levin, "Balancing Production Lines Composed by Series of Workstations with Parallel Operations Blocks", Proeceedings of the IEEE International Symp. on Assembly and Task Planning, Page(s):122 - 127 Digital Object Identifier 10.1109/ISATP.2003.

[4] P. Golinska, M. Fertsh, J. Marx-Gomez, J. Oleskow," The Proposal for Breakdowns Monitoring System Design –Case Study", In proceeding WAC, Budapest, Hungary, 2006.

[5] Dolgui, A. Guschinsky,N. and Levin, G., (1999),"On Problem of optimal design of transfer lines with parallel and sequential operations", Volume 1, 18-21 Page(s):329 - 334 vol.1, Digital Object Identifier 10.1109/ETFA. 815373.

[6] Golinska P., Pawlewski P., Fertsh M.," Monitoring the Operations Management Performance in Automotive Industry", In proceeding WAC, Budapest, Hungary, 2006.

[7] Oleœków J., "Agent-Based Technology as a Tool for the Integration of Manufacturing Systems", In proceeding WAC, Budapest, Hungary, 2006.

[8] Trujillo J. and Pasek Z.J. and Baeyens E., "On-line job-shop scheduling of a manufacturing system based on a virtual supervisor concept", In: 16th *IFAC World Congress*, Prague, 2005.

[9] Trujillo J. A, Cárdenas C., Pasek Z.J., and Baeyens E., (2006)"On-line scheduling method of manufacturing system based on VS algorithm for reference pattern", In proceeding WAC, Budapest, Hungary.

[10] Fox, M.S., Chionglo, J.F., and Barbuceanu, M., (1993), The Integrated Supply Chain Management System. Internal Report, Dept. of Industrial Engineering, Univ. of Toronto.

[11] Peng, Y., Finin, T., Labrou, Y., Chu, B., Long, J., Tolone, W.J. and Boughannam, A. (1998) A Multi-Agent System for Enterprise Integration. In Proceedings of PAAM'98, London.

[12] McEleney, B., O'Hare, G.M.P. and Sampson, J. (1998). An Agent Based System for Reducing Changeover Delays in a Job-Shop Factory Environment. In Proc. of PAAM'98, London.

[13] Fordyce, K. and Sullivan, G.G. (1994). Logistics Management System (LMS): Integrating Decision Technologies for Dispatch Scheduling in Semiconductor Manufacturing. *Intelligent Scheduling, Zweben, M. and Fox, M.S., eds., Morgan Kaufman Publishers, San Francisco, CA, pp. 473516.*

[14] Maturana, F. and Norrie, D. (1996). Multi-Agent Mediator Architecture for Distributed manufacturing. *Journal of Intelligent Manufacturing*, **7**, 257-270.

[15] Shen, W. and Norrie, D.H. (1998) An Agent-Based Approach for Dynamic Manufacturing Scheduling. In Working Notes of the Agent-Based Manufacturing Workshop, Minneapolis, MN.

[16] Butler, J. and Ohtsubo, H. (1992). ADDYMS: Architecture for Distributed Dynamic Manufacturing Scheduling. *Artificial Intelligence Applications in Manufacturing*, Famili, A., Nau, D.S., and Kim, S.H., (eds.), The AAAI Press, pp. 199-214.

[17] Park, H., Tenenbaum, J. and Dove, R. (1993) Agile Infrastructure for Manufacturing Systems: A Vision for Transforming the US Manufacturing Base. Defense Manufacturing Conference.

[18] Parunak, V.D., Baker, A. and Clark, S. (1998). The AARIA Agent Architecture: From Manufacturing Requirements to Agent-Based System Design. In Working Notes of the Agent-Based Manufacturing Workshop, Minneapolis, MN.

[19] Golinska P., Fertsch M. , Oleskow J., Pawlewski P. and Marx-Gomez J. :"Integration of Material Flow Management in Supply Chain through Agent-Based Systems". [in:] Proceedings of the 2nd International Conference "From Scientific Computing to Computational Engineering" (2nd IC-SCCE) Athens, Greece, 5-8 July, 2006.

[20] Trujillo, J., Pasek Z.J. and Baeyens E., (2003). On-line tracking of manufacturing system performance via virtual supervisor. In: 2nd CIRP International Conference on Reconfigurable Manufacturing. Ann Arbor, MI

[21] Trujillo, J., Pasek Z.J. and Baeyens E., (2006). Composition of Structures based on finite position machine FPM for Reconfiguration of manufacturing system controls. In: ISFA International Simposium on Flexible Automation. Osaka, Japan