Chapter 6

On Just-In-Time Production Leveling

Francesco Giordano and Massimiliano M. Schiraldi

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/54994

1. Introduction

Since the 80's, the Japanese production techniques and philosophies spread among the Western manufacturing companies. This was possible because the Toyota Motor Company experience was indeed a success. The so-called "Toyota Production System" (TPS) seemed to be the "one best way" to manage a manufacturing production site.

On the other side, it is also well known that not every implementation of Lean Production was a success, especially in Western companies: some enterprises – together with the consultancy firms that should have supported them – forgot that there are some main hypotheses and issues to comply with, in order to achieve Toyota-like results. On top of this, certain requisites are not related to a mere managerial approach, but depend on exogenous conditions, e.g. market behavior or supplier location; thus, not every company can successfully implement a TPS system.

One critical requirement for a TPS approach to be effective is that the production plan should be leveled both in quantity and in mix. This is indicated by the Japanese term *heijunka* (平準化), which stands for "leveling" or "smoothing". Here, we will focus our attention on why leveled production is a key factor for JIT implementation, and specifically we will describe and analyze some approaches to deal with the leveling problem.

At first, the original Toyota Production System is briefly recalled, with specific regard to the *Just In Time* (JIT) approach to manage inventories in production. JIT is a stock replenishment policy that aims to reduce final product stocks and work-in-process (WIP); it coordinates requirements and replenishments in order to minimize stock-buffer needs, and it has reversed the old make-to-stock production approach, leading most companies to adopt "pull" instead of "push" policies to manage material and finished product flows. However, in case of unleveled demand, stock levels in JIT may grow uncontrolled.

Secondly, *kanban*-based production is described: *kanban*, a Japanese word meaning "visual record", is a card that contains information on a product in a given stage of the manufacturing process, and details on its path of completion. It is acknowledged as one of the most famous technique for material management in the JIT approach. Here we will present some common algorithms for managing *kanban* queues, along with their criticalities in terms of production smoothing requirements and reduced demand stochasticity. Some of the JIT-derivative approaches will be recalled as well: CONWIP, Quick Response Manufacturing, Theory of Constraints and the Just-In-Sequence approach.

Then, a review on the mixed-model JIT scheduling problem (MMJIT), along with the related solving approaches, is presented. Despite the huge literature on MMJIT mathematical programming approaches, here it will be described why the real-world production systems still prefer the simpler kanban approach and the old (1983) Goal Chasing Method algorithm. In the end, an overview on simulators advantages to test alternative heuristics to manage JIT production is presented.

2. Managing Just-In-Time production systems

Just-in-Time was first proposed within the *Toyota Production System* (TPS) by Taiichi Ohno after the 50's when he conceived a more convenient way to manage inventory and control production systems [1]. *Lean Production* – the un-branded name of TPS – is a mix of a philosophy for production systems management and a collection of tools to improve the enterprise performances [2]. Its cornerstones are the reduction of *muda* (wastes), *mura* (unevenness) and *muri* (overburden). Ohno identified seven wastes [3] that should be reduced to maximize the return of investment of a production site:

- transportation;
- inventory;
- motion;
- waiting;
- over-processing;
- over-producing;
- defects.

The TPS catchphrase emphasizes the "zero" concept: zero machine changeovers ("set-ups"), zero defects in the finished products, zero inventories, zero production stops, zero bureaucracy, zero misalignments. This result may be reached through a continuous improvement activity, which takes cue from Deming's *Plan-Do-Check-Act* cycle [1]: the *kaizen* approach.

Just-In-Time is the TPS solution to reduce inventory and waiting times. Its name, according to [4], was coined by Toyota managers to indicate a method aimed to ensure "the right products,

http://dx.doi.org/10.5772/54994

in the right quantities, *just in time*, where they are needed". Differently from Orlicky's Material Requirement Planning (MRP) – which schedules the production run in advance compared to the moment in which a product is required [5] – JIT approach will replenish a stock only after its depletion. Among its pillars there are:

- one-piece flow;
- mixed-model production;
- demand-pull production;
- *takt* time;

Indeed, generally speaking, processing a 10 product-batch requires one tenth of the time needed for a 100 product-batch. Thus, reducing the batch value (up to "one piece") would generate benefits in reducing either time-to-market or inventory level. This rule must come along with mixed-model production, which is the ability of manufacture different products alternating very small batches on shared resources. Demand-pull production indicates that the system is activated only after an order receipt; thus, no semi-finished product is processed if no downstream workstation asks for it. On top of this, in order to smooth out the material flow, the process operations should be organized to let each workstation complete different jobs in similar cycle times. The base reference is, thus, the *takt* time, a term derived from the German word *taktzeit* (cycle time), which is computed as a rapport between the net operating time, available for production, and the demand in terms of units required. These are the main differences between the *look-ahead* MRP and the *look-back* JIT system. For example, the MRP algorithm includes a lot-sizing phase, which results in product batching; this tends to generate higher stock levels compared to the JIT approach. Several studies have been carried out on MRP lot-sizing [6] and trying to improve the algorithm performance [7, 8, 9]; however, it seems that JIT can outperform MRP given the *heijunka* condition, in case of leveled production both in quantity and in mix. The traditional JIT technique to manage production flow is named *kanban*.

3. The kanban technique

A *kanban* system is a multistage production scheduling and inventory control system [10]. Kanban cards are used to control production flow and inventories, keeping a reduced production lead time and work-in-process. Clearly, a kanban is not necessarily a physical paper/plastic card, as it can be either electronic or represented by the container itself.

Since it was conceived as an easy and cheap way to control inventory levels, many different implementations of kanban systems have been experimented in manufacturing companies all over the world. In the following paragraphs, the most commonly used "one/two cards" kanban systems are described.

3.1. One-card kanban system

The "one-card" is the simplest implementation of kanban systems. This approach is used when the upstream and downstream workstations (respectively, the preceding and succeeding processes) are physically close to each other, so they can share the same stock buffer. The card is called "Production Order Kanban" (POK) [11, 12]. The stock buffer acts either as the outbound buffer for the first (A) workstation or as the inbound buffer for the second (B) workstation. A schematic diagram of a one-card system is shown in Figure 1.

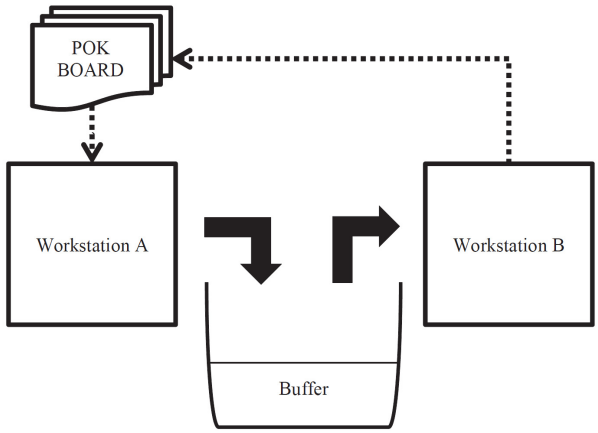

Figure 1. A one-card kanban system

Here, each container (the JIT unit load) has a POK attached, indicating the quantity of a certain material contained, along with eventual complementary information. The POK also represents a production order for the Workstation A, indicating to replenish the container with the same quantity. When a B operator withdraws a container from the buffer, he removes the POK from the container and posts it on a board. Hence, A operator knows that one container with a specific part-number must be replenished in the stock buffer.

3.2. Two-card kanban system

In the two-card system, each workstation has separate inbound and outbound buffers [13, 14]. Two different types of cards are used: Production Order Kanbans (POK) and Withdrawal Kanbans (WK). A WK contains information on how much material (raw materials / semi-finished materials) the succeeding process should withdraw. A schematic diagram of a two-card system is shown in Figure 2.

Each work-in-progress (WIP) container in the inbound buffer has a WK attached, as well as each WIP in the outbound buffer has a POK. WK and POK are paired, i.e. each given part number is always reported both in *n* POK and *n* WK. When a container is withdrawn from the inbound buffer, the B operator posts the WK on the WK board. Then, a warehouse-keeper

http://dx.doi.org/10.5772/54994

operator uses the WK board as a picking list to replenish the inbound buffer: he takes the WK off the board and look for the paired POK in the outbound buffer. Then, he moves the corresponding quantity of the indicated material from the A outbound to the B inbound buffer, while exchanging the related POK with the WK on the container, restoring the initial situation. Finally, he posts the left POK on the POK board. Hence, like in the previous scenario, A workstation operator knows that one container of that kind must be replenished in the outbound stock buffer. The effectiveness of this simple technique – which was described in details by several authors [3, 14, 15, 16] – is significantly influenced by the policy followed to determine the kanban processing order, in the boards.

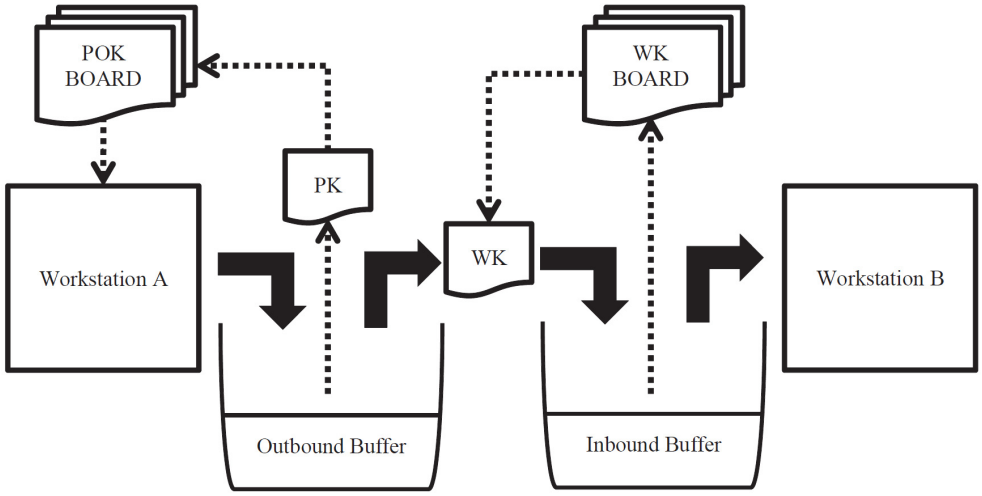

Figure 2. A two-card kanban system

3.3. Standard approaches to manage the kanban board

From the previously described procedure, it is clear that the each workstation bases its production sequence on kanban cards posted on the POK board. In literature, few traditional ways to manage the board are reported: each of them is quite easy to implement and does not require significant investments in technology or other expensive assets.

The most commonly used policy [3] requires having a board for each station, and this should be managed as a single First-In-First-Out (FIFO) queue. The board is usually structured as one vector (one column, multiple rows): POK are posted on the board in the last row. Rows are grouped in three zones (red/yellow/green) which indicate three levels of urgency (respectively, high/medium/low). Kanban are progressively moved from the green to the red zone and the workstation operator will process the topmost kanban. If a kanban reaches the red rows, it means that the correspondent material is likely to be requested soon, by the succeeding process. Thus, it should be urgently replenished in the outbound buffer, in order to avoid stock-outs.

Although this policy does not rely on any optimized procedure, it may ensure a leveled production rate in each workstation, given the fact that other TPS pillars are implemented, e.g. setup time reduction and mixed model scheduling. Indeed, if the final downstream demand is leveled, the production plan of the workstations will be leveled as well. Clearly, this policy is vulnerable to high setup times and differences among workstations cycle times: in this latter case, indeed, the ideal jobs sequence for a workstation may be far from optimal for the preceding. It is noticeable that the colored zones on the board only provide a visual support for the operators and do not influence the jobs processing order.

A *heijunka box* is a sort of enhanced kanban board: it still acts as a visual scheduling tool to obtain production leveling at the workstations. However, differently from the traditional board, it manages to keep evidence of materials distinctions. Usually, it is represented as a grid-shaped wall schedule. Analogously to the simpler board, each row represents a time interval (usually, 30-60 minutes), but multiple columns are present, each one associated to a different material. POKs are placed in the so-called "pigeon-holes" within the box, based on number of items to be processed in the job and on the material type. Workstation operators will process all the kanban placed in the current period row, removing them from the box. Hence, heijunka box not only provides a representation for each job queued for production, but for its scheduled time as well, and allows operators to pursue production leveling when inserting new POKs in the boxes.

3.4. Criticism on JIT

During the last decades, Just-In-Time has been criticized from different authors [17]. Indeed, certain specific conditions – which, though, are not uncommon in manufacturing companies – can put in evidence some well-known weak points of the Japanese approach. Specifically, un-steady demand in multi-product environments where differences in processing lead times are not negligible represent a scenario where JIT would miserably fail, despite the commitment of the operations managers.

First, we have to keep in mind that one pillar of Lean Production is the "one-piece-flow" diktat. A one-piece batch would comply with the Economic Production Quantity theory [18] only when order cost (i.e. setup time) is zero. Having non-negligible setup times hampers JIT implementation and makes the production leveling problem even more complicated. It is peculiar that, originally, operations researchers concentrated on finding the best jobs sequence considering negligible setups time. This bound was introduced into the mixed model kanban scheduling problem only since 2000. Setups are inevitable in the Lean Production philosophy, but are considered already optimized as well. Given that setup times are *muda*, TPS approach focuses on quickening the setup time, e.g. through technical interventions on workstations or on the setup process with SMED techniques, not on reducing their frequency: the increased performance gained through setups frequency reduction is not worth the flexibility loss that the system may suffer as a consequence. Indeed, the standard kanban management system, ignoring the job sequencing, does not aim at reducing setup wastes at all. Analogously, the Heijunka box was developed for leveling production and can only assure that the product mix in the very short term reproduces that in the long term; in its original application, the decision

http://dx.doi.org/10.5772/54994

on the job sequence is left to the operator. Only in some enhanced version, the sequence is predefined applying some scheduling algorithm.

Given the fact that JIT is based on stock replenishment, constant production and withdrawal rates should be ensured in order to avoid either stock outs or stock proliferation. Mixed-model production requires a leveled Master Production Schedule (MPS) [19], but this is not sufficient to smooth the production rate in a short time period. While it is easy to obtain a leveled production in a medium or even medium-short period, it is difficult to do it in each hour, for each workstation and each material.

Indeed, demand is typically unstable under two points of view: random frequency, which is the chance that production orders are irregularly received, and random quantities, which is related to product mix changes. Indeed, since TPS assume minimal stock levels, the only chance to cope with demand peak is to recur to extra production capacity. However, available production capacity should be higher than required as the average (as TPS requires), but for sure cannot be limitless. Thus, the JIT management system should anyway be able to consider the opportunity of varying the maintenance plan as well as the setup scheduling, in case of need. On the other hand, if the production site faces a leveled production, changes in product mix should not represent a problem; however, they increase sequencing problem complexity. Most of the operational research solutions for JIT scheduling are designed for a fixed product mix, thus its changes can greatly affect the optimality of solutions, up to make them useless.

On the contrary, kanban board mechanism is not influenced by demand randomness: as long as demand variations are contained into a certain (small) interval, kanban-managed workstations will handle their production almost without any problem. Therefore, in case of unstable demand, in order to prevent stock-outs, inventory managers can only increase the kanban number for each product: the greater are the variations, the greater is the need of kanban cards and, thus, the higher is the stock level. In order to prevent stock level raise, some authors [20, 21] proposed to adopt a frozen schedule to implement JIT production in real companies, where demand may clearly be unstable. Anyway, this solution goes in the opposite direction compared to JIT foundations.

Moreover, one-piece-flow conflicts with demand variability: the batch size should be chosen as its processing time exceeds the inter-arrival time of materials requests. Thus, the leveling algorithm must find the proper sequencing policy that, at the same time, reduces the batch size and minimize the inter-arrival time of each material request. This sequence clearly depends on the total demand of each material in the planning horizon. However, JIT does not use forecasting, except during system design; thus, scheduling may be refreshed daily. From a computational point of view, this is a non-linear integer optimization problem (defined *mixed-model just-in-time scheduling problem*, MMJIT), which has non-polynomial complexity and it currently cannot be solved in an acceptable time. Thus, reliable suppliers and a clockwork supply chain are absolutely required to implement JIT. Toyota faced this issue using various approaches [22]:

- moving suppliers in the areas around the production sites, in order to minimize the supply lead time;

- collaborating with the suppliers and helping them to introduce JIT in their factories;
- always relying on two alternative suppliers for the same material, not to be put in a critical situation.

In the end, it should be noted that, considering that at each stage of the production process at least one unit of each material must be in stock, in case of a great product variety the total stock amount could be huge in JIT. This problem was known also by Toyota [1], who addressed it limiting the product customization opportunities and bundling optional combinations.

4. Alternative approaches

4.1. CONWIP

Many alternatives to JIT have been proposed since TPS appeared in Western countries. One of the most famous JIT-derivative approaches is CONWIP (CONstant Work-In-Process). This methodology, firstly proposed in the 90's [23], tries to mix push and pull approaches: it schedules tasks for each station – with a push approach – while production is triggered by inventory events, which is a pull rule. Thus, CONWIP is card-based, as kanban systems, but cards do not trigger the production of a single component in the closest upward workstation; conversely, cards are used to start the whole production line, from beginning downwards. Then, from the first workstation up to the last one, the process is push-driven; materials are processed as they get to an inbound buffer, notwithstanding the stock levels. Only the last workstation has a predetermined stock level, similar to the JIT outbound buffer. All queues are managed through a FIFO policy. In order to have a leveled production rate and to avoid production spikes or idle times, the system is calibrated on the slowest workstation, the *bottleneck*. Results from simulations showed [24] that CONWIP could grant shorter lead times and more stable production rate if compared to Kanban; however, it usually needs a higher WIP level. A CONWIP system is also easier to implement and adjust, since it has only one card set.

4.2. POLCA

Another alternative technique mixing push and pull system is the POLCA (Paired-Cell Overlapping Loops of Cards with Authorization), which stands at the base of the Quick Response Manufacturing (QRM) approach, proposed in 1998 [25]. QRM aims to minimize lead times rather than addressing waste reduction, as TPS does. A series of tools, such as manufacturing critical-path time, cellular organization, batch optimization and high level MRP, are used to minimize stock levels: the lesser is the lead time, the lesser is the on-hand inventory. Likewise CONWIP, POLCA handles WIP proliferation originating from multiple products, since it does not require each station to have a base stock of each component. At first, an MRP-like algorithm (called HL/MRP) creates some "Release Authorization Times". That means that the HL/MRP system defines when each cell may start each job, as MRP defines the "Start Dates". However, differently from a standard push system - where a workstation should

http://dx.doi.org/10.5772/54994

process the job as soon as possible - POLCA simply authorizes the possibility to start the job. Analogously to CONWIP and Kanban, POLCA uses production control cards in order to control material flows. These cards are only used between, and not within, work cells. Inside each work cell, material flows resemble the CONWIP approach. On top of this, the POLCA cards, instead of being specifically assigned to a product as in a Kanban system, are assigned to pairs of cells. Moreover, whereas a POK card is an inventory replenishment signal, a POLCA card is a capacity signal. If a card returns from a downstream cell, it signals that there is enough capacity to process a job. Thus, the preceding cell will proceed only if the succeeding cell has available production capacity. According to some authors [20] a POLCA system may overcome the drawbacks of both standard MRPs and kanban systems, helping in managing both short-term fluctuation in capacity (slowdowns, failures, setups, quality issues) and reducing unnecessary stocks, which is always present in any unlevelled replenishment system – i.e. where heijunka condition is not met.

4.3. Just in sequence

The Just in Sequence approach is an evolution of JIT, which embeds the CONWIP idea of mixing push/requirement and pull/replenishment production management systems. The overall goal of JIS is to synchronize the material flow within the supply chain and to reduce both safety stocks and material handling. Once the optimal production sequence is decided, it is adopted all along the process line and up to the supply chain. Thus, the suppliers are asked to comply not only to quantity requirements but also to the product sequence and mix, for a certain period of time. In this case the demand must be stable, or a frozen period should be defined (i.e. a time interval, prior to production, in which the demand cannot be changed) [26]. Clearly, when the demand mix significantly changes, the sequence must be re-computed, similarly to what happens in MRP. This makes the JIS system less flexible compared to JIT. Research results [27] proved that, applying some techniques to reduce unsteadiness – such as flexible order assignment or mixed bank buffers – the sequence can be preserved with a low stock level. Thanks to *ad-hoc* rescheduling points the sequence can be propagated downstream, reducing the impact of variability.

4.4. The "Theory of Constraints" approach

Leveraging on the common idea that "a chain is no stronger than its weakest link", the Israeli physicist E.M. Goldratt firstly introduced the Theory of Constraints (TOC) in his most famous business novel "the Goal" [28]. Looking to a production flow-shop as a chain, the weakest link is represented by the line bottleneck. Compared to the TPS approach of reducing wastes, this approach is focused on improving bottleneck operations, trying to maximize the *throughput* (production rate), minimizing inventory and operational expenses at the same time.

Its implementation is based on a loop of five steps:

1. constraint identification;
2. constraint optimization;
3. alignment of the other operations to the constraint optimization;

4. elevation of the constraint (improving throughput);
5. if the constraint after the previous 4 steps has moved, restart the process.

Again, Deming's concept of "improvement cycle" is recalled. However, improvements are only focused on the bottleneck, the Critical Constraint Resource (CCR), whereas in the Lean Production's Kaizen approach bottom-up, an improvement may arise wherever wastes are identified; moreover, improvements only aim to increase throughput. It is though noticeable that the author includes, as a possible throughput constraint, not only machinery problem but also people (lack of proper skills) and policies (bad working). To this extent, Goldratt coined the "Drum-Buffer-Rope" (DBR) expression: the bottleneck workstation will define the production takt-time, giving the beat as with a drum. The remaining upstream and downstream workstations will follow this beat. This requires the drum to have an optimized schedule, which is imposed to all the production line. Thus, takt-time is not defined from the final demand anymore, but is set equal to the CCR minimal cycle time, given that the bottleneck capacity cannot be overcome. A "buffer" stock is only placed before the CCR, assuring that no upward issue could affect the process pace, reducing line throughput. This helps in reducing the inventory level in comparison to replenishment approaches, where buffers are placed among all the workstations. Eventually, other stock buffers may be placed in few synchronization points in the processes, besides the final product warehouse, which prevents stock-outs due to oscillating demand. The "rope" represents the job release authorization mechanism: a CONWIP approach is used between the CCR and the first phase of the process. Thus, the advanced entrance of a job in the system is proportional to the buffer size, measured in time. Failing to comply with this rule is likely to generate too high work-in-process, slowing down the entire system, or to generate a starvation condition on the CCR, with the risk of reducing the throughput. Several authors [29, 30, 31] analyzed the DBR rule in comparison to planning with mathematical linear programming techniques. Results on the most effective approach are controversial.

5. The mixed-model JIT scheduling problem

The leveling problem in JIT operations research literature was formalized in 1983 as the "mixed-model just-in-time scheduling (or sequencing) problem" (MMJIT) [32], along with its first solution approach, the "Goal Chasing Method" (GCM I) heuristic.

Some assumptions are usually made to approach this problem [33]. The most common are:

- no variability; the problem is defined in a deterministic scenario;
- no details on the process phases: the process is considered as a black box, which transforms raw materials in finished products;
- zero setup times (or setup times are negligible);
- demand is constant and known;
- production lead time is the same for each product.

http://dx.doi.org/10.5772/54994

Unfortunately, the problem with these assumptions virtually never occurs in industry. However, the problem is of mathematical interest because of its high complexity (in a theoretical mathematical sense). Because researchers drew their inspiration from the literature and not from industry, on MMJIT far more was published than practiced.

The objective of a MMJIT is to obtain a leveled production. This aim is formalized in the Output Rate Variation (ORV) objective function (OF) [34, 35]. Consider a M set of m product models, each one with a d_m demand to be produced during a specific period (e.g., 1 day or shift) divided into T production cycles, with

$$\sum_{m \in M} d_m = T$$

Each product type m consists of different components p belonging to the set P. The production coefficients a_{pm} specify the number of units of part p needed in the assembly of one unit of product m. The matrix of coefficients A = (a_{pm}) represents the Bill Of Material (BOM). Given the total demand for part p required for the production of all m models in the planning horizon, the target demand rate r_p per production cycle is calculated as follows:

$$r_p = \frac{\sum_{m \in M} d_m \cdot a_{pm}}{T}, \quad \forall\, p \in P$$

Given a set of binary variables x_{mt} which represent whether a product m will be produced in the t cycle, the problem is modeled as follows [33]:

$$\min Z = \sum_{p \in P} \sum_{t=1}^{T} \left(\sum_{m \in M} \sum_{t=1}^{t} x_{mt} \cdot a_{pm} - t \cdot r_p \right)^2$$

subject to

$$\sum_{m \in M} x_{mt} = 1, \quad \forall\, t = 1, \ldots, T$$

$$\sum_{t=1}^{T} x_{mt} = d_m, \quad \forall\, m \in M$$

$$x_{mt} \in \{0, 1\}, \quad \forall\, m \in M; t = 1, \ldots, T$$

The first and second group of constraints indicate that for each time t exactly one model will be produced and that the total demand d_m for each model will be fulfilled by the time T. More constraints can be added if required, for instance in case of limited storage space.

A simplified version of this problem, labeled "Product Rate Variation Problem" (PRV) was studied by several authors [36, 37, 38], although it was found it is not sufficient to cope with the variety of production models of modern assembly lines [33]. Other adaptations of this problem were proposed along the years; after 2000, when some effective solving algorithms were proposed [39], the literature interest moved on to the MMJIT scheduling problem *with setups* [40]. In this case, a dual OF is used [41]: the first part is the ORV/PRV standard function, while the second is simply:

$$\min S = 1 + \sum_{t=2}^{T} s_t$$

In this equation, $s_t = 1$ if a setup is required in position t; while $s_{t=} 0$, if no setup is required. The assumptions of this model are:

- an initial setup is required regardless of sequence; this is the reason for the initial "1" and the t index follows on "2";
- the setup time is standard and it is not dependent from the product type;
- the setup number and setup time are directly proportional each other.

The following sets of bounds must be added in order to shift s_t from "0" to "1" if the production switches from a product to another:

$$x_{m(t-1)} - x_{mt} \leq s_t, \quad \forall\, t = 2, \ldots, T, \ \forall\, m \in M$$

$$s_t \in \{0, 1\}, \ \forall\, t = 2, \ldots, T$$

Being a multi-objective problem, the MMJIT with setups has been approached in different ways, but it seems that no one succeeded in solving the problem using a standard mathematical approach. A simulation approach was used in [42]. Most of the existing studies in the literature use mathematical representations, Markov chains or simulation approaches. Some authors [10, 40] reported that the following parameters may vary within the research carried out in the recent years, as shown in Table 1 below.

5.1. A review on solving approaches

The MMJIT problem, showing nonlinear OF and binary variables, has no polynomial solutions as far as we know. However, a heuristic solution approach can be effective. To get to a good solution, one among dynamic programming, integer programming, linear programming, mixed integer programming or nonlinear integer programming (NLP) techniques can be used. However, those methodologies usually require a long time to find a solution, so are infrequently used in real production systems [44]. Just a few studies used other methods such as statistical analysis or the Toyota formula [45]. The most renowned heuristics are the Miltenburg's [36] and the cited Goal Chasing Method (GCM I) developed in Toyota by Y. Monden. Given the products quantities to be processed and the associated processing times, GCM I computes an "average consumption rate" for the workstation. Then, the processing sequence is defined choosing each successive product according to its processing time, so that the cumulated consumption rate "chases" its average value. A detailed description of the algorithm can be found in [32]. GCM I was subsequently refined by its own author, resulting in the GCM II and the Goal Coordination Method heuristics [46].

The most known meta-heuristics to solve the MMJIT [44, 47, 48] are:

- Simulated Annealing;
- Tabu Search;
- Genetic Algorithms;
- Scalar methods;

http://dx.doi.org/10.5772/54994

- Interactive methods;
- Fuzzy methods;
- Decision aids methods;
- Dynamic Programming.

Parameter / major alternatives		**Alternatives**				
Model structure		Mathematical programming	Simulation	Markov Chains	Other	
Decision variables		Kanban number	Order interval	Safety Stock level	Other	
Performance measures		Kanban number	Utilization ratio	Leveling effectiveness		
Objective function	*Minimize cost*	Setup cost	Inventory holding cost	Operating cost	Stock-out cost	
	Minimize inventory					
	Maximize throughput					
Setting	*Layout*	Flow-shop	Job-shop	Assembly tree		
	Period number	Multi-period	Single-period			
	Item number	Multi-item	Single-item			
	Stage number	Multi-stage	Single-stage			
	Machine number	Multiple machines	Single machine			
	Resources capacity	Capacitated	Non-capacitated			
Kanban type		One-card	Two-card			
Assumptions	*Container size*	Defined	Ignored (container size equals one item)			
	Stochasticity	Random set-up times	Random demand	Random lead times	Random processing times	Determinism
	Production cycles	Manufacturing system	Continuous production			
	Material handling	Zero withdrawal times	Non-zero withdrawal times			
	Shortages	Ignored	Computed as lost sales [43]			
	System reliability	Dynamic demand	Breakdowns possibility	Imbalance between stages	Reworks	Scraps

Table 1. Alternative configurations of most common MMJIT models

In some experiments [44] Tabu Search and Simulated Annealing resulted to be more effective than GCM; however, the computational complexity of these meta-heuristics – and the consequent slowness of execution – makes them quite useless in practical cases, as the same authors admitted.

Another meta-heuristic based on an optimization approach with Pareto-efficiency frontier – the "multi objective particle swarm" (MOPS) – to solve the MMJIT with setups was proposed through a test case of 20 different products production on 40 time buckets [47].

In [48], the authors compared a Bounded Dynamic Programming (BPD) procedure with GCM and with an Ant Colony (AC) approach, using as OF the minimization of the total inventory

cost. They found that BDP is effective (1,03% as the average relative deviation from optimum) but not efficient, requiring roughly the triple of the time needed by the AC approach. Meanwhile, GCM was able to find the optimum (13% as the average relative deviation from optimum) on less than one third of the scenarios in which the AC was successful.

A broad literature survey on MMJIT with setups can be found in [49] while a comprehensive review of the different approaches to determine both kanban number and the optimal sequence to smooth production rates is present in [10].

6. Criticism on MMJIT problem solution

Assumed that time wastes are a clear example of MUDA in Lean Production [3], complex mathematical approaches which require several minutes to compute one optimal sequence for MMJIT [44] should be discarded, given that the time spent calculating new scheduling solutions does not add any value to products. On the other side, it is notable that MRP computation requires a lot of time, especially when it is run for a low-capacity process (in which CRP-MRP or capacitated MRPs are required). However, being MRP a look-ahead system which considers the demand forecasts, its planning is updated only at the end of a predefined "refresh period", not as frequently as it may be required in a non-leveled JIT context. MRP was conceived with the idea that, merging the Bill-Of-Materials information with inventory levels and requirements, the production manager could define a short-term work plan. In most cases, MRP is updated no more than every week; thus, an MRP run may also take one day to be computed and evaluated, without any consequences for the production plan. On the contrary, the situation in JIT environment evolves every time a product is required from downstream. While MRP assumes the Master Production Schedule forecasts as an input, in JIT nobody may know what is behind the curtain, minute by minute.

Indeed, while a perfect JIT system does not need any planning update – simply because in a steady environment (e.g. heijunka) the optimal sequence should always be almost the same, at least in the medium term – real-world short-term variations can deeply affect the optimality of a fixed schedule production. For instance, a one-day strike of transport operators in a certain geographical area can entirely stop the production of a subset of models, and the lack of a raw material for one hour can turn the best scheduling solution into the worst. On top of this, while MRP relies on its "frozen period", JIT is exposed to variability because is supposed to effectively react to small changes in the production sequence. However, some authors noticed that the JIT sequences [10, 48, 50] are not so resistant to demand changes, so a single variation in the initial plan can completely alter the best solution. This is particularly true when the required production capacity gets near to the available. Thus, developing algorithm for solving the MMJIT problem under the hypothesis of constant demand or constant product mix seems useless.

JIT was developed for manual or semi-automated assembly line systems, not for completely automated manufacturing systems. The flexibility of the JIT approach requires a flexible production environment (i.e. the process bottleneck should not be saturated) and this is not

http://dx.doi.org/10.5772/54994

an easy condition to be reached in real industries. Consequently, despite the competence of its operations managers, even a big multinational manufacturer may encounter several problems in implementing JIT if a significant part of its supplier is made of small or medium-size enterprises (SMEs), which are naturally more exposed to variability issues. On top of this, differently from MRP – where the algorithm lies within a software and is transparent for users – in JIT the product sequencing is performed by the workforce and managed through the use of simple techniques, such as the heijunka box, the kanban board or other visual management tools, e.g. *andons*. Thus, any approach to organize JIT production should be easily comprehensible to the workers and should not require neither expert knowledge nor a supercomputer to be applied.

7. Using simulations to validate JIT heuristics

As it has been said, finding good solutions for the MMJIT problem with setups using an algorithmic approach may take too long and, on top of this, the solution can be vulnerable to product mix changes. Indeed, Kanban technique and GCM I methods are the most used approaches to manage JIT production thanks to their simplicity [44]. Some companies, where SMED techniques [51] failed to reduce setup times, use a modified version of the kanban FIFO board, in order to prevent setups proliferation. Thus, a simple batching process is introduced: when more than one kanban is posted on the board, the workstation operator shall not start the job on the first row but, on the contrary, chooses the job which allows the workstation to skip the setup phase. As an example, given the original job sequence A-B-A-C-A-B for a workstation, if the operator is allowed to look two positions ahead, he would process A-A-B-C-A-B, saving one setup time. In such situations, where setup times cannot be reduced under a certain value, rather than giving up the idea of adopting the Lean Production approach, heuristics can be developed and tested in order to obtain a leveled production even if coping with long setup times or demand variability.

The most common method to analyze and validate heuristics is through simulation. Several authors agree that simulation is one of the best ways to analyze the dynamic and stochastic behavior of manufacturing system, predicting its operational performance [52, 53, 54]. Simulating, a user can dynamically reproduce how a system works and how the subsystems interact between each other; on top of this, a simulation tool can be used as a decision support system tool since it natively embeds the *what-if* logic [55]. Indeed, simulation can be used to test the solutions provided by Genetic Algorithms, Simulated Annealing, Ant Colony, etc. since these algorithms handle stochasticity and do not assume determinism. Simulation can be used for:

- productivity analysis [56],
- production performances increase [1, 57, 58],
- confrontation of different production policies [59]
- solving scheduling problems [50, 60].

In spite of these potentialities, there seem to be few manufacturing simulation software really intended for industrial use, which go beyond a simple representation of the plant layout and modeling of the manufacturing flow. On top of some customized simulators – developed and built in a high-level programming language from some academic or research group in order to solve specific cases with drastic simplifying hypotheses – the major part of commercial software implements a graphical model-building approach, where experienced users can model almost any type of process using basic function blocks and evaluate the whole system behavior through some user-defined statistical functions [61]. The latters, being multi-purpose simulation software, require great efforts in translating real industrial processes logic into the modeling scheme, and it is thus difficult to "put down the simulation in the manufacturing process" [55]. Indeed, the lack of manufacturing archetypes to model building seems one of the most remarkable weakness for most simulator tools, since their presence could simplify the model development process for who speak the "language of business" [62]. Moreover, commercial simulators show several limitations if used to test custom heuristics, for example to level a JIT production or to solve a line-balancing problem: some authors report typical weaknesses in presenting the simulation output [63] or limited functionalities in terms of statistical analysis [64], on top of the lack of *user-friendliness*. For instance, most common commercial simulation software do not embed the most useful random distributions for manufacturing system analysis, such as the Weibull, Beta and Poisson distribution. When dealing with these cases, it is often easier to build custom software, despite it requires strong competences in operations research or statistics that have never represented the traditional background of industrial companies analysts [64].

In order to widespread simulation software usage among the manufacturing industry, some authors underline the need of a standard architecture to model production and logistics processes [65, 66, 67]. Literature suggested to focus on a new reference framework for manufacturing simulation systems, that implement both a structure and a logic closer to real production systems and that may support industrial processes optimization [68, 69].

Moreover, given hardware increased performances, computational workload of a simulation tool is not a problem anymore [70] and it seems possible to develop simulators able to run in less than one minute even complex instances. The complexity of a manufacturing model is linked both to size and system stochasticity. A careful analysis of time series can provide useful information to be included in the simulator, in order to model stochastic variables linked to machine failures or scrap production. This allows a more truthful assessment of key performance indicators (KPI) for a range of solutions under test.

8. Conclusions and a roadmap for research

The effective application of JIT cannot be independent from other key components of a lean manufacturing system or it can "end up with the opposite of the desired result" [71]. Specifically, leveled production (heijunka) is a critical factor. The leveling problem in JIT, a mixed-model scheduling problem, was formalized in 1983 and named MMJIT. Several numbers of

http://dx.doi.org/10.5772/54994

solving approaches for MMJIT have been developed during the last decades. Most of them assume constant demand and product mix. Zero setup-times hypothesis has been removed only since 2000, and few approaches still cope with stochasticity. On top of this, these algorithms, although heuristic based, usually spend too much time in finding a good solution. Simplification hypotheses, operations research competences requirements and slow execution prevented these approaches to widespread in industry. Indeed, the heijunka box or the standard FIFO kanban approach with the simple Goal-Chasing-Method heuristic are still the most used tools to manage production in JIT environment. This is acknowledged also by the proponents of alternatives, and GCM is always used as a benchmark for every new MMJIT solution. However, these traditional approaches are not so effective in case of long setups and demand variations, given the fact that they have been conceived in pure JIT environments. In high stochastic scenarios, in order to prevent stock-outs, kanban number is raised along with the inventory levels. There are several cases of companies, operating in unstable contexts and where setup times cannot be reduced over a certain extent, that are interested in applying JIT techniques to reduce inventory carrying costs and manage the production flow in an effective and simple way. The development of kanban board / heijunka-box variations, in order to cope with the specific requirements of these companies, seems to offer better potentialities if compared to the development of difficult operations research algorithmic approaches. In order to solve industrial problems, researchers may concentrate in finding new policies that could really be helpful for production systems wishing to benefit from a JIT implementation but lacking in some lean production requirements, rather than studying new algorithm for MMJIT problem.

For instance, kanban board / heijunka-box variations can effectively focus on job preemption opportunities in order to reduce setups abundance, or on new rules to manage priorities in case of breakdowns or variable quality rates. The parameters fine-tuning can be performed through simulation. In this sense, given the limitations of most commercial software, the development of a simulation conceptual model – along with its requisites – of a model representation (objects and structures) and some communication rules between the subsystems (communication protocols) are the main issues that need to be addressed from academics and developers.

Author details

Francesco Giordano and Massimiliano M. Schiraldi

Department of Enterprise Engineering, “Tor Vergata” University of Rome, Italy

References

[1] J. P. Womack, D. T. Jones & D. Roos, The machine that changed the world: The Story of Lean Production, New York (NY): HarperPerennial, 1991.

[2] M. Rother & J. Shook, Learning to See: Value Stream Mapping to Add Value and Eliminate Muda, Brookline (MA): The Lean Enterprise Institute, 1999.

[3] T. Ohno, Toyota Production System: Beyond Large-Scale Production, Productivity Press, 1988.

[4] R. J. Schonberg, Japanese manufacturing techniques: Nine hidden lessons in simplicity, New York (NY): Free Press, 1982.

[5] J. Orlicky, Material Requirement Planning, New York (NY): McGraw-Hill, 1975.

[6] L. Baciarello, M. D'Avino, R. Onori & M. Schiraldi, « Lot-Sizing Heuristics Performance» working paper, 2013.

[7] A. Bregni, M. D'Avino e M. Schiraldi, «A revised and improved version of the MRP algorithm: Rev MRP,» *Advanced Materials Research (forthcoming),* 2013.

[8] M. D'Avino, V. De Simone & M. Schiraldi, «Revised MRP for reducing inventory level and smoothing order releases: a case in manufacturing industry,» Production Planning & Control, 2013 (forthcoming).

[9] M. D'Avino, M. Correale & M. Schiraldi, «No news, good news: positive impacts of delayed information in MRP» working paper, 2013.

[10] C. Sendil Kumar & R. Panneerselvam, «Literature review of JIT-KANBAN system,» *International Journal of Advanced Manufacturing Technologies,* pp. 393-408, 2007.

[11] B. J. Berkley, «A review of the kanban production control research literature,» *Production and Operations Management,* vol. 1, n. 4, pp. 393-411, 1992.

[12] B. Sharadapriyadarshini & R. Chandrasekharan, «Heuristics for scheduling in a Kanban system with dual blocking mechanisms,» *European Journal of Operational Research,* vol. 103, n. 3, pp. 439-452, 1997.

[13] O. Kimura & H. Terada, «Design and analysis of pull system, a method of multistage production control,» *International Journal of Production Research,* n. 19, pp. 241-253, 1981.

[14] B. Hemamalini & C. Rajendran, «Determination of the number of containers, production kanbans and withdrawal kanbans; and scheduling in kanban flowshops,» *International Journal of Production Research,* vol. 38, n. 11, pp. 2549-2572, 2000.

[15] R. Panneerselvam, Production and Operations Management, New Delhi: Prentice Hall of India, 1999.

http://dx.doi.org/10.5772/54994

[16] H. Wang & H.-P. B. Wang, «Optimum number of kanbans between two adjacent workstations in a JIT system,» *International Journal of Production Economics,* vol. 22, n. 3, pp. 179-188, 1991.

[17] D. Y. Golhar & C. L. Stamm, «The just in time philosophy: a literature review,» *International Journal of Production Research,* vol. 29, n. 4, pp. 657-676, 1991.

[18] F. W. Harris, «How many parts to make at once,» *Factory, The Magazine of Management,* vol. 10, n. 2, pp. 135-136, 1913.

[19] R. B. Chase, N. J. Aquilano & R. F. Jacobs, Operations management for competitive advantage, McGraw-Hill/Irwin, 2006.

[20] R. Suri, «QRM and Polca: A winning combination for manufacturing enterprises in the 21st century,» Center for Quick Response Manufacturing, Madison, 2003.

[21] P. Ericksen, R. Suri, B. El-Jawhari & A. Armstrong, «Filling the Gap,» *APICS Magazine,* vol. 15, n. 2, pp. 27-31, 2005.

[22] J. Liker, The Toyota Way: 14 Management principles from the world's greatest manufacturer, McGraw-Hill, 2003.

[23] M. L. Spearman, D. L. Woodruff & W. J. Hopp, «CONWIP: a pull alternative to kanban,» *International Journal of Production Research,* vol. 28, n. 5, pp. 879-894, 1990.

[24] R. P. Marek, D. A. Elkins & D. R. Smith, «Understanding the fundamentals of kanban and conwip pull systems using simulation,» in *Proceedings of the 2001 Winter simulation conference*, Arlington (VA), 2001.

[25] R. Suri, Quick Response Manufacturing: A companywide approach to reducing lead times, Portland (OR): Productivity Press, 1998.

[26] M. M. Schiraldi, La gestione delle scorte, Napoli: Sistemi editoriali, 2007.

[27] S. Meissner, «Controlling just-in-sequence flow-production,» *Logistics Research,* vol. 2, p. 45.53, 2010.

[28] E. M. Goldratt, The Goal, Great Barrington, MA: North river press, 1984.

[29] M. Qui, L. Fredendall & Z. Zhu, «TOC or LP?,» *Manufacturing Engineer,* vol. 81, n. 4, pp. 190-195, 2002.

[30] D. Trietsch, «From management by constraints (MBC) to management by criticalities (MBC II),» *Human Systems Management,* vol. 24, pp. 105-115, 2005.

[31] A. Linhares, «Theory of constraints and the combinatorial complexity of the product-mix decision,» *International Journal of Production Economics,* vol. 121, n. 1, pp. 121-129, 2009.

[32] Y. Monden, Toyota Production System, Norcross: The Institute of Industrial Engineers, 1983.

[33] N. Boysen, M. Fliedner & A. Scholl, «Level Scheduling for batched JIT supply,» *Flexible Service Manufacturing Journal,* vol. 21, pp. 31-50, 2009.

[34] W. Kubiak, «Minimizing variation of production rates in just-in-time systems: A survey,» *European Journal of Operational Research,* vol. 66, pp. 259-271, 1993.

[35] Y. Monden, Toyota Production System, An Integrated Approach to Just-In-Time, Norcross (GA): Engineering & Management Press, 1998.

[36] J. Miltenburg, «A Theoretical Basis for Scheduling Mixed-Model Production Lines,» *Management Science,* vol. 35, pp. 192-207, 1989.

[37] W. Kubiak & S. Sethi, «A note on ''level schedules for mixed-model assembly lines in just-in-time production systems",» *Management Science,* vol. 37, n. 1, pp. 121-122, 1991.

[38] G. Steiner & J. S. Yeomans, «Optimal level schedules in mixed-model, multilevel JIT, assembly systems with pegging,» *European Journal of Operational Research,* pp. 38-52, 1996.

[39] T. N. Dhamala & S. R. Khadka, «A review on sequencing approaches for mixed-model just-in-time production systems,» *Iranian Journal of Optimization,* vol. 1, pp. 266-290, 2009.

[40] M. S. Akturk & F. Erhun, «An overview of design and operational issues of kanban systems,» *International Journal of Production Research,* vol. 37, n. 17, pp. 3859-3881, 1999.

[41] P. R. McMullen & P. Tarasewich, «A beam search heuristic method for mixed-model scheduling with setups,» *International Journal of Production Economics,* vol. 96, n. 2, pp. 273-283, 2005.

[42] F. Mooeni, S. M. Sanchez & A. J. Vakharia, «A robust design methodology for Kanban system design,» *International Journal of Production Research,* vol. 35, pp. 2821-2838, 1997.

[43] G. N. Krieg & H. Kuhn, «A decomposition method for multi-product kanban systems with setup times and lost sales,» *IEE Transactions,* vol. 34, pp. 613-625, 2002.

[44] T. Tamura, S. Nishikawa, T. S. Dhakar e K. Ohno, «Computational Efficiencies of Goal Chasing, SA, TS and GA Algorithms to Optimize Production Sequence in a Free Flow Assembly Line,» in *Proceedings of the 9th Asia Pasific Industrial Engineering & Management Systems Conference*, Bali, 2008.

[45] K. Ohno, K. Nakashima & M. Kojima, «Optimal numbers of two kinds of kanbans in a JIT production system,» *International Journal of Production Research,* vol. 33, pp. 1387-1401, 1995.

[46] H. Aigbedo, «On bills of materials structure and optimum product-level smoothing of parts usage in JIT assembly systems,» *International Journal of Systems Science,* vol. 40, n. 8, pp. 787-798, 2009.

[47] A. Rahimi-Vahed, S. M. Mirghorbani e M. Rabbani, «A new particle swarm algorithm for a multi-objective mixed-assembly line sequencing problem,» *Soft computing,* vol. 11, pp. 997-1012, 2007.

[48] N. Boysen, M. Fliedner & A. Scholl, «Sequencing mixed-model assembly lines to minimize part inventory cost,» *Operational Research Spectrum,* pp. 611-633, 2008.

[49] A. Allahverdi, J. N. D. Gupta & T. Aldowaisan, «A review of scheduling research involving setup considerations,» *International Journal of Management Sciences,* vol. 27, pp. 219-239, 1999.

[50] P. Rogers & M. T. Flanagan, «Online simulation for real-time scheduling of manufacturing systems,» *Industrial Engineering,* pp. p. 37-40, 2000.

[51] S. Shingo, A revolution in manufacturing: The SMED system, Productivity Press, 1985.

[52] V. A. Hlupic, «Guidelines for selection of manufacturing simulation software,» *IIE Transactions,* vol. 31, n. 1, pp. 21-29, 1999.

[53] A. M. Law, Simulation modeling and analysis, Singapore: McGraw-Hill, 1991.

[54] J. Smith, «Survey of the use of simulation for manufacturing system design and operation,» *Journal of manufacturing systems,* vol. 22, n. 2, pp. 157-171, 2003.

[55] H. Berchet, «A model for manufacturing systems simulation with a control dimension.,» *Simulation Modelling Practice and Theory,* pp. p.55-57, 2003.

[56] A. Polajnar, B. Buchmeister & M. Leber, «Analysis of different transport solutions in the flexible manufacturing cell by using computer simulation,» *International Journal of Operations and Production Management,* pp. 51-58, 1995.

[57] P. Rogers & R. J. Gordon, «Simulation for the real time decision making in manufacturing systems,» in *Proceedings of the 25th conference on winter simulation*, Los Angeles (CA), 1993.

[58] P. Rogers, «Simulation of manufacturing operations: optimum-seeking simulation in the design and control of manufacturing systems: experience with optquest for arena,» in *Proceedings of the 34th conference on winter simulation: exploring new frontiers*, San Diego (CA), 2002.

[59] S. S. Chakravorty & J. B. Atwater, «Do JIT lines perform better than traditionally balanced lines,» International *Journal of Operations and Production Management*, pp. 77-88, 1995.

[60] R. Iannone & S. Riemma, «Proposta di integrazione tra simulazione e tecniche reticolari a supporto della progettazione operativa.,» Università di Salerno, Salerno, 2004.

[61] D. A. Van Beek, A. T. Hofkamp, M. A. Reniers, J. E. Rooda & R. R. H. Schiffelers, «Syntax and formal semantics of Chi 2.0,» Eindhoven University of Technology, Eindhoven, 2008.

[62] J. Banks, E. Aviles, J. R. McLaughlin & R. C. Yuan, «The simulator: new member of the simulation family,» *Interfaces,* pp. 21-34, 1991.

[63] A. M. Law & S. W. Haider, «Selecting simulation software for manufacturing applications: practical guidelines and software survey.,» *Industrial Engineering,* 1989.

[64] L. Davis & G. Williams, «Evaluating and Selecting Simulation Software Using the Analytic Hierarchy Process,» *Integrated Manufacturing Systems,* n. 5, pp. 23-32, 1994.

[65] D. A. Bodner & L. F. McGinnis, «A structured approach to simulation modeling of manufacturing systems,» in *Proceedings of the 2002 Industrial Engineering Research Conference*, Georgia, 2002.

[66] S. Narayanan, D. A. Bodner, U. Sreekanth, T. Govindaraj, L. F. McGinnis & C. M. Mitchell, «Research in object-oriented manufacturing simulations: an assessment of the state of the art,» *IIE Transactions,* vol. 30, n. 9, 1998.

[67] M. S. Mujtabi, «Simulation modeling of manufacturing enterprise with complex material. Information and control flows,» *International journal of computer integrated manufacturing,* vol. 7, n. 1, pp. 29-46, 1994.

[68] S. Robinson, «Conceptual modeling for simulation: issues and research requirements,» in *Proceedings of the 2006 Winter Simulation Conference*, Piscataway (NJ), 2006.

[69] C. Battista, G. Dello Stritto, F. Giordano, R. Iannone & M. M. Schiraldi, «Manufacturing Systems Modelling and Simulation Software Design: A Reference Model,» in *XXII DAAAM International World Symposium* , Vienna, 2011.

[70] A. M. Law & W. D. Kelton, Simulation Modelling and Analysis, New York: McGraw-Hill., 1991, pp. 60-80.

[71] S. Shingo, A study of the Toyota Production System, Productivity Press, 1989.