



HAL
open science

A simulation-based hybrid backward scheduling framework for manufacturing systems

Christodoulos Lalas, Dimitris Mourtzis, Nikolaos Papakostas, George Chryssolouris

► **To cite this version:**

Christodoulos Lalas, Dimitris Mourtzis, Nikolaos Papakostas, George Chryssolouris. A simulation-based hybrid backward scheduling framework for manufacturing systems. *International Journal of Computer Integrated Manufacturing*, 2006, 19 (08), pp.762-774. 10.1080/09511920600678827. hal-00513377

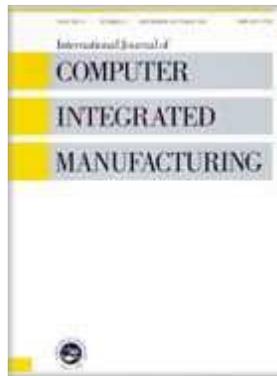
HAL Id: hal-00513377

<https://hal.science/hal-00513377>

Submitted on 1 Sep 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



A simulation-based hybrid backward scheduling framework for manufacturing systems

Journal:	<i>International Journal of Computer Integrated Manufacturing</i>
Manuscript ID:	TCIM-2005-IJCIM-0050.R1
Manuscript Type:	Original Manuscript
Date Submitted by the Author:	07-Mar-2006
Complete List of Authors:	Lalas, Christodoulos; University of Patras, Mechanical Engineering and Aeronautics Mourtzis, Dimitris; University of Patras, Mechanical Engineering and Aeronautics Papakostas, Nikolaos; University of Patras, Mechanical Engineering and Aeronautics Chryssolouris, George; University of Patras, Mechanical Engineering and Aeronautics
Keywords:	SIMULATION APPLICATIONS, DISCRETE EVENT SIMULATION, MULTI-CRITERIA DECISION MAKING, PRODUCTION CONTROL, SCHEDULING, DISPATCHING RULES
Keywords (user):	TEXTILE INDUSTRY, FINITE CAPACITY SCHEDULING



A simulation-based hybrid backward scheduling framework for manufacturing systems

C. LALAS, D. MOURTZIS, N. PAPAKOSTAS and G. CHRYSSOLOURIS*

Laboratory for Manufacturing Systems and Automation
Department of Mechanical Engineering and Aeronautics
University of Patras
Patras 261 10, Greece

Tel.: +30 2610 997262

Fax: +30 2610 997744

* Corresponding author. E-mail: xrisol@lms.mech.upatras.gr

This paper presents a hybrid backward scheduling method, referred to as HBS, which mainly addresses discrete manufacturing environments. It operates under the framework of hierarchical finite capacity shop floor modelling and discrete event simulation. HBS applies a set of transformation relations in order to convert a finite capacity forward scheduling method (FS) that can employ different assignment policies to their backward counterparts. These policies include both single criterion conventional dispatching rules, as well as an adjustable multiple-criteria decision making technique that can take into consideration a number of different conflicting criteria, such as flowtime, tardiness and manufacturing cost. Performance of the HBS method was studied through a set of simulation experiments in a typical textile industry and was evaluated through a number of relevant performance indicators.

Keywords: Finite capacity; Backward scheduling; Discrete event simulation; Textile industry

1. Introduction

In the decision-making hierarchy of the production planning and control function of manufacturing systems, detailed scheduling constitutes the final step before actual output occurs. Scheduling encompasses allocating workloads to specific workcenters and determining the sequence in which operations are to be performed. Generally, the objectives of scheduling are to achieve trade-offs among conflicting goals, which include efficient utilization of staff, equipment, facilities and minimization of customer waiting time, inventories and cycle times.

Scheduling methods can be differentiated by the degree of optimality sought by their algorithms. In general, optimization methods are only applicable to relatively small problems. Due to the fact that when the number of variables and constraints is raised computational time increases rapidly, they operate under a lot of simplistic assumptions that restrict their use to small problems only. Computational difficulty tends to increase exponentially with the problem's size (Vollmann *et al.* 1997, Baker 1998, Ho and Chang 2000). The more realistic problems, such as the capacity constrained case, have been shown to be NP-hard, or else NP-complete. Because many simple optimal and near-optimal scheduling problems are NP-hard, such as the minimum makespan problem on m parallel machines without preemption and the minimum flowtime and minimum tardiness problems for an m -machine job shop, where $m > 2$, it has been proved that most realistically sized optimal scheduling problems are also NP-hard (Baker 1998). This is the reason why large-scale problems are usually treated with 'toward optimal' scheduling procedures that do not attempt to gain optimality but can guarantee a good solution in a reasonable amount of time, in terms of specific performance measures (Wiers 1997).

Another basic distinction between scheduling methods is that of infinite and finite capacity scheduling. Infinite scheduling approaches determine workcenter loads with no regard to their capacity. The resulting load profiles may reveal both under loaded and overloaded workcenters. Finite scheduling explicitly establishes a detailed schedule for each workcenter based on its finite capacity limits within the length of the planning horizon. The superiority of finite capacity scheduling over infinite capacity scheduling is obvious.

Finite capacity scheduling can be further divided to vertical and horizontal scheduling. The approach of filling a workcenter operation by operation is called vertical loading. In the case of horizontal loading an entire shop order, the one with the highest priority, is loaded for all its operations, then the next priority order and so on. The horizontal approach may be at odds with the objective of maximum capacity utilization. Creating detailed schedules with horizontal loading can result to idle times within a workcenter's capacity, even if a job is pending, because a more important order will follow. This side effect is often referred to as forced idle time. Forced idle times tend to result in longer lead times and may restrict throughput in heavily loaded shops (Enns 1996). However the horizontal approach is consistent with how most scheduling research is conducted (Vollmann *et al.* 1997, Yeh, 1997). Nevertheless, vertical loading is generally considered to provide good scheduling solutions in situations where finding the optimal schedule is not feasible and NP-hard (Enns 1996).

In addition to the vertical-horizontal distinction of finite capacity scheduling methods, there is also the issue of forward scheduling versus backward scheduling. Forward scheduling will schedule all operations of a job from the schedule start date to actual finite capacity, commencing with the first operation in its routing sequence. It aims at completing the job as early as possible. It can also be employed to examine whether the earliest feasible completion time will meet a customer's requirements. Backward scheduling will schedule all

operations of a job from its due date, starting from the last operation in its routing sequence. Its objective is to complete the job on or as close as possible to its due date.

The scheduling method presented in this paper, denoted as hybrid backward scheduling (HBS), can be classified as a toward optimality finite capacity backward scheduling method that follows the vertical loading logic. It particularly addresses discrete manufacturing environments and functions under the framework of hierarchical shop floor modelling and discrete event simulation. This has led to the development of a simulation software tool. Simulation is an efficient approach to examine realistic, multiple-machine, dynamic scheduling problems. Shop floor simulation can also examine the performance of various assignment logics against several criteria.

The remainder of this paper is structured as follows: In section 2, related research on finite capacity scheduling is reviewed. Section 3 describes the concept of the proposed HBS method. Following, section 4 is focused on a case study discussing the implementation of HBS in a Greek textile industry. The performance of the proposed method is discussed in section 5. Finally, section 6 is devoted to conclusions and directions for future research.

2. Literature review

In his paper Enns (1996) compared the horizontal (referred as block-time) and vertical (referred as nondelay) loading approaches when used in a forward finite capacity scheduling system, using simulation. The routing assumptions used were those of a general flow shop. Only due date priority rules were considered.

In their study, Watson *et al.* (1997) suggested an approach that generates order release plans employing a queuing simulation model of the production facility. Backward planning

was used in order to calculate workload-dependent planned lead times for future planned or forecasted orders, while open orders were scheduled forwards.

Akkan (1997) defined a problem of finite capacity backward scheduling where the goal was to reserve portions of future capacity for arriving orders so that rejected orders and earliness cost was minimized. However, the constraints imposed restrict its applicability.

Kawtummachai *et al.* (1997) reported on a near-optimal backward scheduling method that was used to realize JIT theory in an automated flow shop. Six conventional dispatching rules were used as the assignment policy. The objective function was to minimize the total cost, as calculated through the production schedule of orders.

Another related concept in literature is the finite loading algorithm for job-oriented scheduling, which schedules one job at a time where one job may have multiple sequential operations (Yeh 1997). The algorithm assigns feasible start and finish times to the operations of a job by loading them forward or backward onto capacity constrained parallel machines.

Baker (1998) reported on a heterarchical multi-agent scheduling system, named Market-Driven Contract Net that performed forward and backward cost-driven scheduling, on a first-come-first-served basis. The system used horizontal loading in a bucketless environment.

Moreover, Liu (1998) investigated how dispatching rules affect the performance of stochastic finite capacity scheduling systems. Extensive simulation experiments have been conducted on the criteria of tardiness and proportion of jobs tardy in order to provide guidance to schedulers on selecting the most efficient rule. Operation-based dispatching rules provided better due date performance than their corresponding job-based rules. However these rules were only used to schedule forward in time. A similar study by Rajendran and Holthaus (1999) compared the performance of dispatching rules in dynamic flowshops and job shops with respect to the objectives of minimizing flowtime and tardiness-related criteria.

Agrawal *et al.* (2000) studied production planning in manufacturing facilities that produce large and complex assemblies, for which cycle times range between two months to two years. Their approach employed a lead-time evaluation and scheduling algorithm for performing detailed backward scheduling of operations with the unique objective of minimizing cycle time. The estimated lead times were then scaled to account for capacity sharing effects by multiple products in common resources and were used by a MRP-based system to release work-orders to the shop floor. Numerical experiments showed that cycle time improved but still capacity was roughly planned based on estimated lead time offsets, a fixed lot-size for each end item and an average product mix obtained from historical data and forecasts.

In another study, Ho and Chang (2001) proposed an integrated MRP and Just-In-Time (JIT) framework, modelled as an integer linear program in combination with forward and backward heuristics, for finding detailed shop floor schedules with the objective of minimizing the total production cost. As soon as a forward pass scheduled the final operation of the job with the earliest due date, the backward heuristic performed horizontal loading based on the Least Unit Total Cost (LUTC) criterion. The authors used a time bucketed planning horizon under the limitation that no consecutive operations of a part can be scheduled in the same period without, however, providing any information related to actual implementations. Moreover, they did not deal with the scheduling problem using multiple criteria.

Yoo and Martin-Vega (2001) presented several heuristics for the single-machine scheduling problem with the objective of minimizing the number of tardy jobs for the single due date case and the number of early to tardy jobs for the due date window case. A backward scheduling procedure was developed, yielding satisfactory experimental results for a general class of early to tardy jobs ratio problems.

In their paper, Song *et al.* (2002) considered dynamic production scheduling for manufacturing systems producing complex engineer-to-order (ETO) products with multiple levels of manufacturing and assembly. Their paper presented two heuristic methods in order to minimize the total earliness and tardiness costs. One of them was a backward horizontal scheduling algorithm. Case studies were quoted to demonstrate the effectiveness of this method.

Maheswaran and Ponnambalam (2003) dealt with an investigation on the total weighted tardiness of the single machine scheduling problem. A horizontal backward heuristic was developed to solve the single-machine, single-criterion problem. The assumption of sequence independent setup times and the unavailability of alternative resources further restrict its use. In a later study of the same authors (Maheswaran and Ponnambalam 2004) an intensive search evolutionary algorithm was proposed in order to be used as a meta-heuristic as soon as an initial schedule is formed by their horizontal backward scheduling method.

Moreover, Saad *et al.* (2004) developed an integrated model for order release and due date management. Orders were scheduled by a horizontal backward finite scheduling method in a planning horizon that was broken into time buckets. Five different assignment rules were utilized to determine their due dates. However, no relevant case studies were reported.

As it can be concluded from the above, finite capacity scheduling methods vary substantially with respect to their scheduling algorithms and the performance measures they attempt to improve. In most studies found in literature the horizontal loading approach was implemented, while some authors have investigated vertical loading procedures in a forward scheduling context. Although backward scheduling can be also implemented with the vertical approach, at least conceptually, it is usually employed in finite capacity scheduling systems using the horizontal approach (Enns 1996). The finite scheduling method proposed in this paper integrates the following attributes:

- Utilizes the vertical loading concept both forwards and backwards. The assignment logic employed, or else the operational policy, can be either a multiple-criteria decision making technique, or a single-criterion dispatching rule. A different operational policy can be assigned to each workcenter.
- Addresses discrete multi-stage and multi-product manufacturing industries with multiple parallel machines at each production stage. Final or semi-final products can have linear or divergent structures with multi-level components.
- Considers the specific productivity, setup and processing costs per time unit of alternative resources that can be employed to perform the same operation. Setup times are sequence dependent.
- It is a dynamic scheduling method. Job orders arrive dynamically over time and they are scheduled only after release to the shop floor. Precedence relationships of the operations comprising a job order are also considered.
- It enables lap-phasing orders by moving transfer batch sizes through successive operations, order-splitting, adding capacity through overtime and simulating deterministic and/or stochastic machine breakdowns.
- It creates a hierarchical model of both the facility and the workload. The dynamic shop activity is modelled using discrete event simulation in real shop calendar time.
- The efficiency of the proposed approach is evaluated through a large number of performance indicators (see Appendix).

3. The proposed approach

HBS operates under the framework of hierarchical shop floor and workload modelling and event-driven simulation. This operational framework is next presented.

3.1 The HBS operational framework

The production facility is divided into Job Shops that can produce a family of similar semi-final and end products. Each Job Shop is further divided into Workcenters, which in turn consist of a number of Resources. The latter can be defined as individual production cells or parallel processors that can perform similar operations (Figure 1).

[Insert Figure 1 about here]

Corresponding to the facility's hierarchy there is also the workload hierarchical breakdown. Orders are broken down into Jobs, which in turn consist of a number of Tasks (Figure 2). An Order corresponds to the overall production facility and is divided into Jobs that based on their specifications can be processed only by a suitable Job Shop. A Job consists of Tasks that can be released to one Workcenter only. Tasks can be dispatched to more than one of the Workcenter's parallel Resources. Among the constraints taken into consideration in releasing and dispatching Jobs and Tasks are the facility's finite capacity and their precedence relationships.

[Insert Figure 2 about here]

The operational policy behind the assignment of a task to a specific resource can be either a multiple-criteria decision making technique (Chryssolouris and Lee 1994), or a simple dispatching rule. The advantages of dispatching rules come from their simplicity. They do not attempt to predict the future, but rather make decisions based on the present. Thus, these rules are very useful in factories that are extremely unpredictable, such as job shops. Also, dispatching rules are usually spatially local, requiring only the information available at the

location where the decision will be implemented. When the multiple-criteria decision making technique is employed, several alternatives are formed and evaluated before assigning the available resources to pending production tasks. The choice of the best alternative is made by evaluating a set of criteria, such as cost, flowtime, quality and tardiness, in a decision matrix. A utility function is applied to rank the alternatives and choose the best. Released orders are scheduled directly, without aggregation. Grouping for the sake of machine efficiency or economical lot size is handled by assigning proper weight factors to the relevant criteria, not by aggregating them into one order.

Schedules are constructed on the basis of events occurring sequentially through time. The next scheduling decision is identified by moving along the time horizon until an event is scheduled to occur that will initiate a change in the status of the system. This would usually be the completion of a task on one of the resources or the arrival of a job to one of the workcenter queues. All operations eligible for loading at the time a resource becomes available are considered. When there are multiple jobs competing for a machine, the selected operational policy is used to determine the highest priority operation. Hence, the schedule is constructed by simulating the detailed shop activity in real calendar time.

The algorithm of HBS is depicted using the following example. Let J_1 and J_2 be two jobs that are released to the shop floor with the same arrival dates (AD_j), while their due dates (DD_j) differ by δ time units. The routing of each of them consists of three tasks T_{ij} , where i is the serial number of the task and j the job it belongs to, e.g. T_{21} is the second task of job J_1 :

$$J_1: T_{11} \longrightarrow T_{21} \longrightarrow T_{31}, \quad (AD_1, DD_1)$$

$$J_2: T_{12} \longrightarrow T_{22} \longrightarrow T_{32}, \quad (AD_2 = AD_1, DD_2 = DD_1 + \delta)$$

The production facility consists of one job shop that includes one workcenter, which in turn contains three resources R1, R2 and R3. The tasks that each resource is suitable to perform are indicated in the parentheses: R1: (T_{11}, T_{12}), R2: (T_{21}, T_{22}) and R3: (T_{31}, T_{32}).

Setup times are sequence dependent. For each resource R_k the setup times for every possible succession of tasks in them are stored in a unique setup matrix $[R_k]$, where element (t_{mn}) corresponds to the setup time required for changing production from task m to task n . For the sake of simplicity, in this example, all setup and processing times are assumed to be equal and all resources work full time. Implementation of the HBS framework in this example would lead to a finite capacity forward scheduling (FS) procedure due to its event-driven logic described above. The representation of the resulting shop floor schedule in a Gantt chart would look something like the one in figure 3. The relative priorities of tasks would depend on the operational policy selected. Resources R_1 and R_2 are assumed to be under repair or scheduled maintenance for the time intervals r_1 and r_2 , respectively.

[Insert Figure 3 about here]

Dispatching tasks as soon as possible under this forward scheduling framework may result in relatively high job slack times (Figure 3), that is the final products of an order may be finished much earlier than their due date if the workload is not tight. Dispatching tasks forwards, by using the HBS framework, may also incur increased holding costs and demand for storage space, high risk of product deterioration and decrease in capacity's flexibility to process new order arrivals with shorter due dates. It is however useful in finding out whether the earliest feasible completion time will meet customer's requirements.

3.2 The HBS transformation relations

The HBS method will schedule all tasks of a job from its due dates, starting from the last task. Its objective is to complete each job on or close to its due date, thus minimizing its

slack time. Operating under the hierarchical modelling and discrete event simulation framework presented in the previous section, HBS applies a set of transformation relations in order to apply many well-known dispatching rules and a multiple-criteria decision making technique (Chryssolouris and Lee 1994), as the operational policy, backwards. Using the previous example, the required transformations of the HBS' algorithm include:

Step 1: Creation of the reverse routing for each job. Precedence relationships are now reverted, meaning that the preconditions of a task are now its post conditions:

$$J_1: T_{31} \Longrightarrow T_{21} \Longrightarrow T_{11}, \quad (AD'_1, DD'_1)$$

$$J_2: T_{32} \Longrightarrow T_{22} \Longrightarrow T_{12}, \quad (AD'_2, DD'_2)$$

Step 2: Transposition of the setup matrices of every resource R_k :

$$[R_k]^{\text{trans}} = [R_k]^T \quad (1)$$

Step 3: Definition of an arbitrary reference time point (t_{ref}) that is greater or equal to the latest due date of all jobs released to the shop floor by the time of schedule start (t_s):

$$t_{\text{ref}} \geq \max(DD_j) \quad (2)$$

The time differences between reference point t_{ref} and job due dates are (Figure 3):

$$J_1: t_{\text{ref}} - DD_1 = a_1 \text{ time units and } J_2: t_{\text{ref}} - DD_2 = a_2 \text{ time units}$$

Step 4: Definition of a time point in a fictitious time scale where the reverse workload will be scheduled (t'_s). The tonic accent in a variable will be used to denote its reference to the fictitious time scale. The time point t'_s in the fictitious time scale (t') corresponds to time point t_{ref} in the real time scale (t). Since the resources work without any intervals the selection of point t'_s in the fictitious time scale is arbitrary and independent from the selection of point t_{ref} in the real time scale. When time in the fictitious scale increases, real time decreases proportionately. Therefore, if S time units pass after time point t'_s in the fictitious time scale ($t' = t'_s + S$), then in the real time scale the clock is turned S time units in the past before the reference point t_{ref} ($t = t_{\text{ref}} - S$):

$$t' = t'_s + S \Leftrightarrow t = t_{\text{ref}} - S \quad (3)$$

Thereby, the relationship between any point t in real time and its corresponding point t' in fictitious time is:

$$t' = t'_s + (t_{\text{ref}} - t) \quad (4)$$

Step 5: Order arrival dates (AD_j) in the real time scale are transferred to the fictitious time scale using equation (4), where they constitute the order due dates (DD'_j). Likewise, order due dates (DD_j) in real time are transferred to fictitious time using equation (4), where they constitute the order arrival dates (AD'_j):

$$AD'_j = t'_s + (t_{\text{ref}} - DD_j) \text{ and } DD'_j = t'_s + (t_{\text{ref}} - AD_j) \quad (5)$$

Implementing equation (5) in the case of the two jobs of the example:

$$J_1: AD'_1 = t'_s + (t_{\text{ref}} - DD_1) = t'_s + a_1, DD'_1 = t'_s + (t_{\text{ref}} - AD_1)$$

$$J_2: AD'_2 = t'_s + (t_{\text{ref}} - DD_2) = t'_s + a_2, DD'_2 = t'_s + (t_{\text{ref}} - AD_2)$$

Moreover, since $DD_1 = DD_2 - \delta$ and $AD_1 = AD_2 \Leftrightarrow AD'_1 = AD'_2 + \delta$ and $DD'_1 = DD'_2$.

Step 6: Transformation of machine down time intervals (r_1 and r_2) to the fictitious time scale using equation (4).

Step 7: Forward scheduling the transformed workload in the fictitious time scale with t'_s being the schedule's start time (Figure 4). In case a dispatching rule is selected as the operational policy, its 'reverse' must be used in this step, e.g. if the SPT rule is selected then LPT should be used in the fictitious time scale. This does not apply when the multiple-criteria technique is selected due to the fact that the criteria it uses are variables to be calculated and not strict priority setting rules.

[Insert Figure 4 about here]

Step 8: Checking schedule's feasibility. If a job in the Gantt chart of figure 4 completes after its due date DD'_j in fictitious time then it should have been released to the shop floor prior to its arrival date AD_j in real time in order not to violate its real due date DD_j . Moreover, using equation (4) to find the transformation of present time t_s to the fictitious time scale (t'_{past}) leads to: $t'_{past} = t'_s + (t_{ref} - t_s)$. So, time points with $t' > t'_{past}$ in fictitious time correspond to the past, that is points with $t < t_s$. If a job completes after time point t'_{past} in fictitious time, it reveals that it should have been released to the shop floor in the past, prior to present time t_s , in order not to violate its due date DD_j . If for a specific job order j the objective is to check whether the schedule could meet the customer's due date requirement then set: $DD'_j = t'_{past}$.

Step 9: Mirroring the fictitious schedule in real time scale using the following transformation relations (Figure 5):

$$ST_{ij} = t_{ref} - (CT'_{ij} - t'_s) \text{ and } CT_{ij} = t_{ref} - (ST'_{ij} - t'_s) \quad (6)$$

Where: ST_{ij} is the start time of task T_{ij} in the real time scale,

t_{ref} is the reference time point in the real time scale,

CT'_{ij} is the completion time of task T_{ij} in the fictitious time scale,

t'_s is the schedules' start time in the fictitious time scale,

CT_{ij} is the completion time of task T_{ij} in the real time scale and

ST'_{ij} is the start time of task T_{ij} in the fictitious time scale.

[Insert Figure 5 about here]

The final schedule is now formed. The HBS method created a schedule with minimum job slack times using either a single criterion dispatching rule, or a multiple-criteria decision

making technique as the assignment policy. It is interesting to note the relativeness of time intervals $a_1, a_2, a_3, a_4, c, r_1, r_2$ between figures 3, 4 and 5.

4. Application of HBS: Case study in a typical textile industry

The HBS methodology was implemented in a software tool using Visual C++, version 5.0 (Win32 API), for discrete event simulation and Visual Basic Applications coding (VBA), version 6.3, for HBS transformations, respectively. The integration is achieved through an appropriate Open Database Connectivity driver (ODBC).

Performance of HBS was studied through a set of simulation experiments in a vertically organized Greek textile industry. The under study company operates in the woollen textile system and its product range includes yarns for clothing, carpeting, knitting and wool/synthetic carpets. The proposed method has been applied to the production line of blend carpets. The selected production line consists of three discrete departments, namely the Dyeing, Spinning and Weaving departments. Each of them has been modelled as a Job Shop, according to HBS operational framework. The hierarchical facility model breakdown of the selected production line and the tasks associated with each workcenter, are listed in the following table.

[Insert Table 1 about here]

The main point of interest lies within the production planning of the spinning and weaving job shops that contain a large number of alternative resources capable of performing the same tasks. The workload model for the selected production area consists of more than 200 different jobs types for the weaving job shop and 60 jobs types for the spinning job shop.

Each job type in the weaving job shop corresponds to a single carpet type, as defined by its colour set, quality (surface density), yarn types, weaving pattern, shape and dimensions. Each job type in the spinning job shop corresponds to a single yarn type defined by its colour, quality (type of fibres selected), composition and title. The tasks comprising these jobs are shown in table 1, next to their corresponding workcenters. They are performed sequentially as listed. Their precedence relationships and the processing and set-up times required by each alternative resource are all stored in the workload model.

In an assembly manufacturing system the scheduling objective of minimizing the cumulative lead time for a set of assembled jobs can be achieved by applying HBS to component jobs after the schedule start date of the final assembly job is determined by either FS or HBS. Component jobs required for making a multi-stage product should be finished at the same time and assembled together to achieve a no slack time, minimise work-in-process (WIP) inventory and cumulative lead time. Weaving, the main task of a weaving job, can be considered as an assembly operation of four different yarn systems, namely the warp, weft, pile and selvage yarn systems, that must be available together to begin execution of an order in the weaving workcenter. So, tasks in the spinning job shop to produce them are scheduled backwards from the time they are required in the weaving job shop. The weaving job shop was also scheduled using HBS in order to achieve a just-in-time production schedule.

Mean/maximum tardiness, mean/maximum earliness, mean/maximum reserve time, mean/maximum flowtime, mean/maximum queue time, mean manufacturing cost, mean capacity utilization, makespan and customer service level were all used as performance indicators for evaluating the obtained detailed shop floor schedules. A short description of the different operational policies and performance measures is given in the appendix.

5. Results and discussion

A set of simulation experiments have been conducted in order to test the efficiency of HBS. Real data were collected from the sales department of the under study textile industry, covering a planning horizon of 60 days, plus 30 days' data for initialisation purposes. In all different experimental scenarios the same workload, comprising 160 weaving job orders and 120 spinning job orders, was scheduled following the HBS procedure, resulting into more than 1800 task assignments to 54 resources in each simulation run. The relative performance of 14 different assignment policies for scheduling the same workload was evaluated through a large set of performance indicators.

The same operational policy was assigned to all workcenters in each simulation scenario. Four different configurations of the multiple-criteria decision making technique (MULTI), proposed by Chryssolouris (1994), were introduced. While the specific criteria in each of them were kept the same, namely cost, flowtime, and tardiness, their relative importance in the decision making process varied through the assignment of appropriate weight factors to them: w_c , w_f , and w_t , respectively. In the first configuration the values of the weight factors were MULTI1: $(w_c, w_f, w_t) = (0.2, 0.4, 0.4)$, in the second MULTI2: $(w_c, w_f, w_t) = (0.1, 0.8, 0.1)$, in the third MULTI3: $(w_c, w_f, w_t) = (0.1, 0.1, 0.8)$ and in the fourth MULTI4: $(w_c, w_f, w_t) = (0.8, 0.1, 0.1)$.

The setting of orders' due dates can be the result of delivery times promised to customers, MRP processing or managerial decisions based on various due date setting policies (Enns 1996, Kuroda *et al.* 2002, Saad *et al.* 2004). In this study due dates were calculated using the number of operations rule (NOP), as follows:

$$DD_j = AD_j + k \cdot N_j \quad (7)$$

Where: k is the allowance factor in days and N_j is the number of tasks of job j . Since time in queue is usually the largest component of a job's lead time, the number of tasks it contains can be used as an indicator of the required flowtime. The values of the allowance factor k were set at 3 levels to represent different due date tightness situations. These differences in allowance factor values allow the effect of deviations between due dates and expected completion times, based on the schedule generated, to be investigated. The values of k were set at 0.7, 0.9 and 1.4. The first two values result in a set of very tight due dates, while the last one in a relatively relaxed tightness level. The selection of the first two tight conditions was based on the premise that the relative performance of different operational policies can be depicted more clearly in tight due date environments. Moreover, tight due dates can provide a competitive advantage by permitting the firm to offer an improved level of customer service, as well as achieve lower costs through reductions in WIP inventory.

Thus, in all, there are 14 different operational policies, 3 different due date settings, making a total number of 42 experimental scenarios for the 1800 tasks workload, resulting into more than $1800 \times 14 \times 3 = 75600$ task assignments. The production facility operates two shifts a day, six days a week. The mean capacity utilization level was kept constant at 70% in all the experiments. Results in terms of mean/maximum tardiness and mean/maximum reserve time are shown in the next two tables.

[Insert Table 2 about here]

With increasing allowance factor k both mean and maximum tardiness decrease, as the due date tightness level relaxes (Table 2). As it was expected, the MOPNR, LIFO and MWRK dispatching rules performed poorly with respect to both performance indicators, in order of deficiency. The reason is that these rules try to promote orders that are less likely to finish on time due to their number of operations, late arrival, and work remaining,

respectively. This logic can result in prioritizing orders that would anyway finish late at the expense of others that would otherwise complete before their due date, thereby leading to even more late orders. The SPT rule, while achieving a moderate mean tardiness performance, it produced the second largest maximum tardiness. On the other hand MULTI3, EDD, FIFO and FASFS produced the best results, in order of efficiency. The exceptional performances of MULTI3 and EDD can be attributed to the fact that both policies directly address due dates and try to minimize lateness.

[Insert Table 3 about here]

Reserve time is defined as the time difference between an order's arrival time and its actual start time. It can be utilized as an indicator of a schedule's flexibility, or else its ability to reserve capacity in the near term in order to be able to respond more efficiently to new customer demands or rush orders. High mean reserve times also correspond to low WIP inventories. As the allowance factor k increases the mean and maximum reserve times increase too (Table 3). This is a basic advantage of the HBS method due to the fact that it attempts to minimize jobs' slack times and thus it produces high reserve times. This effect is more obvious in the case of a relaxed due date setting ($k=1.4$). Increased schedule flexibility was achieved using the SPT, FIFO, MULTI1 and MULTI2 policies, while the MOPNR, MWRK, MULTI4 and LPT policies provided less flexible schedules.

In the two following tables the relative scheduling efficiency of the employed operational policies is explored using additional performance indicators. In the case of dispatching rules, results were independent of the due date setting, due to their rather simplistic logic (Table 4). These rules schedule orders based on a simple priority rule. The only dispatching rule that takes into account order due dates is the EDD rule. However an increase in allowance factor k could not change orders' relative positions in the priority list. On the contrary, in the case

of a multiple-criteria operational policy (Table 5), results in terms of the same performance indicators, are due date dependent.

[Insert Table 4 about here]

[Insert Table 5 about here]

The mean and maximum earliness were kept relatively low in all experimental scenarios, owing to HBS' logic. Orders finished very close to their due dates, which is consistent with the 'pull' production scheduling concept. The operational policies that produced a high mean reserve time, produced a low mean earliness and the opposite. Furthermore, a parallel examination of tables 4 and 5, leads to the conclusion that rules MOPNR, MWRK and LIFO resulted to the greatest mean and maximum flow times, in order of inefficiency. The SPT rule followed producing a very high maximum flowtime. It should be noted that these are the same rules that performed poorly in the case of mean tardiness, as well. On the opposite, the LWRK, MULTI3, FIFO rules and MULTI1 configuration performed far better than any other assignment policy. The same holds for the FASFS, EDD, FIFO and MULTI3 in the case of maximum flowtime, in order of efficiency. Moreover, the MOPNR, MWRK and LIFO rules reached the greatest mean and maximum queue times, followed by LPT rule. Furthermore, the best results for the mean queue time indicator were achieved by LWRK, MULTI2, MULTI1 and FIFO and for the maximum queue time indicator by EDD, FIFO, FASFS and MULTI3, in order of efficiency.

The multiple-criteria policies outperformed all others in minimizing total makespan. Only the FASFS and LIFO dispatching rules matched their performance. All multiple-criteria configurations also succeeded a low mean manufacturing cost per order, as well. As it was

expected the MULTI4 configuration proved to be the best policy for achieving cost minimization.

[Insert Table 6 about here]

Customer service levels improve when due dates are relaxing (Table 6). Generally, LWRK, MULTI1, MULTI2 and MULTI3 seem the most promising policies in providing an increased customer satisfaction. The derived customer service levels in the tight due dates settings ($k=0.7$ and $k=0.9$) can be all characterized unsatisfactory. However, the purpose in these two cases was solely to reveal the relative performance of different assignment policies in strict conditions. On the other hand, a relative high customer service level was achieved in all simulation experiments with a normal due date setting ($k=1.4$), with the exception of the LIFO, MOPNR and MWRK rules where service level fell below 95%. In the same table it is shown that when the allowance factor k increases, orders' tardiness to queue time ratio falls in all experimental scenarios. This is another mean to evaluate different operational policies based on a comparison between the mean job tardiness they induce and the mean time lost in queue. An efficient policy would tend to minimize the time a job stays in queues in each workcenter in order to reduce its tardiness. The most effective policies in minimizing mean tardiness to queue time ratios were MULTI3, FASFS, FIFO, EDD and MULTI1, in descending order.

Next, the Critical Queue Ratio (CQR) is defined as the quotient of the mean queue time to the mean flowtime value. The CQR indicator is employed in order to study lead time fluctuations due to WIP accumulation and queue formation in front of a workcenter and how the selected operational policy affects them. High CQR values reveal that queue times are a

major part of the total lead time within a workcenter. Furthermore, the CQR indicator can be used as a detection tool of bottleneck workcenters.

[Insert Table 7 about here]

The CQR distributions for each workcenter and for each operational policy, as presented in table 7, are unique for the specific manufacturing system. This unique system's 'signature' revealed that the multiple-criteria policies (0.200-0.228) and the LWRK dispatching rule (0.203) can reduce queues and consequently WIP stockpiling. Moreover, the WIND workcenter has been detected as the most probable bottleneck, since it produced the maximum mean CQR (0.426). The production efficiency of this workcenter should be the first to be increased through overtime, outsourcing or investment in new resources. The BLEND and SPOOL workcenters are the most lightly loaded as almost no queues were reported there. Tasks assigned to these workcenters started as soon as they were released to them.

The mean earliness performance indicator has a direct correlation with inventory performance, since low slack times usually correspond to just in time procurements, WIP and end products inventories reductions. For this reason, all 42 experimental scenarios were rerun using the HBS operational framework (FS), as presented in section 3.1. The goal was to prove the superiority of a backward finite capacity scheduling method like HBS over a forward finite capacity scheduling method like FS, in inventory performance. Results of FS implementation in terms of the mean earliness performance indicator are presented in table 8, along with the reduction percentage incurred by the use of HBS in the same experimental scenario. Those operational policies that produced low mean earliness in the case of HBS,

produced high mean earliness in the case of FS and vice versa. Moreover, as it can be seen, the maximum advantage from implementing HBS can be derived in loose due date situations.

[Insert Table 8 about here]

To sum up the above results, the LWRK, EDD, FIFO rules and the four multiple-criteria technique's configurations produced, in general, the best results for a set of fifteen performance measures. Also, the HBS method proved superior to a forward finite capacity scheduling method like FS in slack time minimization. These statements were validated through the case study presented in this paper. However, the scope of this study is not to provide guidance to schedulers on selecting the most efficient operational policy in discrete production environments. It should be left to the production management of each manufacturing system to consider carefully its own circumstances and most valuable performance measures, when selecting the best alternative shop schedule.

HBS provides a framework-methodology for the transformation of forward assignment policies to their backward counterparts. This framework operates under hierarchical modelling and event-driven simulation. It approaches the detailed scheduling problem of manufacturing systems in an integrated way. Thus its attributes, as set out in section 2, allow for the implementation of the proposed methodology in complex problems, overcoming the application constraints of similar studies found in literature.

Following the vertical loading logic, HBS can provide efficient scheduling solutions in a relatively short processing time in situations where finding the optimal schedule is not feasible and NP-hard, as in real manufacturing systems. Feasible alternative shop schedules are formed dynamically and evaluated through a large set of performance indicators. Owing

to its generality, the HBS method can be implemented in most manufacturing systems by simulating the exact material flow with high detail.

6. Conclusions

To maintain constant visibility within the planning horizon, the shop floor schedule should be built with customer and forecasted orders for as far out as possible. Rescheduling and updating the shop floor schedule every day is important since decay in information validity grows as the schedule's planning horizon is lengthened. Since a simulation run for a planning horizon of several months takes a few minutes only, it is easy for the production manager any time that something unpredicted occurs to reproduce efficiently an updated schedule for the entire production facility.

Future work includes investigating the combinatory use of FS and HBS in two successive passes. Depending on its objective, a job can be either scheduled forwards, using the FS framework, or backwards, using the HBS method, independently of the other jobs. This scheduling feature will enable individual jobs to have varying objectives and priorities within a schedule for accommodating specific needs of individual customer orders. Within a scheduling run, FS will be applied to rush jobs and HBS will be applied to expensive, high inventory value or long due date jobs. Moreover, it will also be possible to use HBS in order to schedule backwards from bottleneck resources. In case FS is employed, the processes prior to bottleneck resources can be rescheduled with HBS after the terminal tasks have been scheduled as soon as possible by the initial forward scheduling run, thus minimizing WIP inventory.

Another challenge is to examine the issue of using different operational policies at each workcenter. Moreover, additional forward single and multiple-criteria decision making

techniques can be converted to their backward counterparts. The reversal of the HBS algorithm should also be investigated for the transformation of backward finite capacity scheduling methods to their forward counterparts.

Acknowledgement

The work reported in this paper was partially supported by the project ‘Heraclitus’, co-funded by the European Union’s Social Fund (ESF) and the Operational Program for Educational and Vocational Training II (EPEAEK II) of Greece.

Appendix

A short description of the dispatching rules used in the simulation experiments is given in the following table.

[Insert Table 9 about here]

The implemented performance indicators for the evaluation of the output shop floor schedules of the HBS method are defined below:

$$\text{Mean Earliness: } ME(t_n) = \frac{1}{N^{\text{comp}}} \cdot \sum_{i=1}^{N^{\text{comp}}} \max(0, t_i^{\text{dd}} - t_i^{\text{comp}})$$

$$\text{Mean Tardiness: } MT(t_n) = \frac{1}{N^{\text{comp}}} \cdot \sum_{i=1}^{N^{\text{comp}}} \max(0, t_i^{\text{comp}} - t_i^{\text{dd}})$$

$$\text{Mean Reserve time: } MR(t_n) = \frac{1}{N^{\text{comp}}} \cdot \sum_{i=1}^{N^{\text{comp}}} (t_i^{\text{start}} - t_i^{\text{arr}})$$

$$\text{Mean Flowtime: } MF(t_n) = \frac{1}{N^{\text{comp}}} \cdot \sum_{i=1}^{N^{\text{comp}}} (t_i^{\text{comp}} - t_i^{\text{arr}})$$

Mean Queue time:
$$MQ(t_n) = \frac{1}{N^{comp}} \cdot \sum_{i=1}^{N^{comp}} \sum_{j=1}^m (t_{ji}^{start} - t_{ji}^{arr})$$

Mean Manufacturing Cost:
$$MMC(t_n) = \frac{1}{N^{comp}} \cdot \sum_{i=1}^{N^{comp}} [PC(J_i) + SC(J_i)]$$

Mean Capacity Utilization:
$$MCU(t_n) = \frac{1}{R \cdot t_n} \cdot \left[\sum_{i=1}^{N^{comp}} (t_i^{comp} - t_i^{start}) + \sum_{j=1}^{N^{WIP}} (t_n - t_w^{start}) \right]$$

Makespan:
$$MS(t_n) = \max t_i^{comp} - \min t_j^{start}$$

Service Level:
$$SL(t_n) = \left(1 - \frac{N^{tardy}}{N^{total}} \right) \cdot 100\%$$

Where N^{comp} : is the number of completed jobs up to time t_n .

t_i^{comp} : is the completion time of Job i.

t_i^{dd} : is the due date of Job i.

t_i^{arr} : is the arrival time of Job i.

t_i^{start} : is the start time of Job i.

t_{ji}^{start} : is the start time of Task j that belongs to Job i.

t_{ji}^{arr} : is the arrival time of Task j that belongs to Job i.

m : is the total number of tasks comprising Job i.

$PC(J_i)$: is the processing cost of Job i.

$SC(J_i)$: is the setup cost of Job i.

R : is the total number of resources.

N^{WIP} : is the number of in-process Jobs at time t_n .

t_w^{start} : is the start time of the in-process Job w at time t_n .

N^{tardy} : is the number of late job orders up to time t_n .

t_n : is the time point at which all performance measures are calculated.

References

- [1] Agrawal, A., Minis, I. and Nagi, R., Cycle time reduction by improved MRP-based production planning. *International Journal of Production Research*, 2000, **38**(18), 4823-4841.
- [2] Akkan, C., Finite-capacity scheduling-based planning for revenue-based capacity management. *European Journal of Operational Research*, 1997, **100**, 170-179.
- [3] Baker, A.D., A survey of factory control algorithms that can be implemented in a multi-agent heterarchy: Dispatching, scheduling and pull. *Journal of Manufacturing Systems*, 1998, **17**(4), 297-320.
- [4] Chryssolouris, G., *Manufacturing Systems: Theory and Practice*, Second Edition, 2006 (Springer - Verlag: New York).
- [5] Chryssolouris, G. and Lee, M., An Approach to Real-Time Flexible Scheduling. *International Journal of Flexible Manufacturing Systems*, 1994, **6**, 235-253.
- [6] Enns, S.T., Finite capacity scheduling systems: performance issues and comparisons. *Computers & Industrial Engineering*, 1996, **30**(4), 727-739.
- [7] Ho, J.C. and Chang, Y.L., An integrated MRP and JIT framework. *Computers & Industrial Engineering*, 2001, **41**, 173-185.
- [8] Kawtummachai, R., Yanagawa, Y., Ohashi, K. and Miyazaki, S., Scheduling in an automated flow shop to minimize cost: Backward-meta scheduling method. *International Journal of Production Economics*, 1997, **49**, 225-235.

- [9] Kuroda, M., Shin, H. and Zinnohara, A., Robust scheduling in an advanced planning and scheduling environment. *International Journal of Production Research*, 2002, **40**(15), 3655-3668.
- [10] Liu, K.-C., Dispatching rules for stochastic finite capacity scheduling, 23rd *International Conference on Computers and Industrial Engineering*, 1998, pp. 113-116.
- [11] Maheswaran, R. and Ponnambalam, S.G., An investigation on single machine total weighted tardiness scheduling problems. *International Journal of Advanced Manufacturing Technology*, 2003, **22**, 243-248.
- [12] Maheswaran, R. and Ponnambalam, S.G., An intensive search algorithm for single-machine total-weighted-tardiness scheduling problems. *International Journal of Advanced Manufacturing Technology*, 2004, Article in press: DOI 10.1007/s00170-004-2093-8.
- [13] Rajendran, C. and Holthaus, O., A comparative study of dispatching rules in dynamic flowshops and jobshops. *European Journal of Operational Research*, 1999, **116**, 156-170.
- [14] Saad, S.M., Picket, N. and Kittiaran, K., An integrated model for order release and due-date demand management. *Journal of Manufacturing Technology Management*, 2004, **15**(1), 76-89.
- [15] Song, D.P., Hicks, C. and Earl, C.F., Dynamic production planning and rescheduling for complex assemblies, *Manufacturing Complexity network Conference*, April 2002, Cambridge UK.
- [16] Stevenson, W.J., *Production / Operations Management*, Fifth Edition, 1996 (Irwin).
- [17] Vollmann, T.E., Berry, W.L. and Whybark, D.C., *Manufacturing Planning and Control Systems*, Fourth Edition, 1997 (Irwin / McGraw-Hill).
- [18] Watson, E.F., Medeiros, D.J. and Sadowski, R.P., A simulation-based backward planning approach for order-release, *Proceedings of the 1997 Winter Simulation Conference*, 1997, pp.765-772.

- [19] Wiers, V.C.S., A review of the applicability of OR and AI scheduling techniques in practice. *Omega International Journal of Management Science*, 1997, **25**(2), 145-153.
- [20] Yeh, C.-H., A fast finite loading algorithm for job oriented scheduling. *Computers in Operational Research*, 1997, **24**(2), 193-198.
- [21] Yoo, W.-S. and Martin-Vega, L.A., Scheduling single-machine problems for on-time delivery. *Computers & Industrial Engineering*, 2001, **39**, 371-392.

For Peer Review Only

Job Shops	Workcenter ID	Task Description	Resources
Dyeing	DYE-WC	Dyeing	1
	PRESS-WC	Hydroextraction	1
	DRY-WC	Drying	1
Spinning	PREP-WC	Opening	2
	BLEND-WC	Blending	6
	CARD-WC	Carding	3
	SPIN-WC	Spinning	7
	VAPOR-WC	Vaporizing	2
	WIND-WC	Cleaning	3
Weaving	SPOOL-WC	Spooling	4
	WEAV-WC	Weaving	27
3 Job Shops	11 Workcenters		57 Resources

Table 1: Hierarchical model of the blend carpets production facility

k	Mean Tardiness (hh:mm:ss)			Max Tardiness (hh:mm:ss)		
	0.7	0.9	1.4	0.7	0.9	1.4
SPT	18:17:46	11:36:01	3:40:44	309:37:15	285:34:25	189:22:45
EDD	10:31:13	3:00:20	0:00:00	57:43:44	33:39:34	0:00:00
FIFO	9:40:57	3:25:58	0:00:00	66:22:23	42:25:33	0:00:00
LIFO	26:20:40	19:20:38	6:50:42	278:03:41	254:09:11	158:13:52
MOPNR	40:13:31	30:51:42	13:14:41	391:30:30	367:41:21	247:23:33
FASFS	10:04:58	3:16:08	0:00:00	65:03:20	41:00:02	0:00:00
LPT	13:03:49	6:25:55	0:00:00	104:19:20	80:21:44	0:00:00
FOPNR	10:52:34	4:27:00	0:00:00	113:20:54	89:19:56	0:00:00
LWRK	11:03:30	6:02:18	0:37:00	159:27:26	135:28:48	39:25:26
MWRK	27:24:10	18:11:32	5:04:49	285:20:49	261:33:21	165:18:39
MULTI1	10:14:01	4:17:49	0:00:00	152:47:12	113:53:38	0:00:00
MULTI2	14:29:24	7:10:01	1:10:06	214:43:52	179:30:09	86:48:32
MULTI3	10:19:26	2:18:17	0:00:00	115:26:03	39:06:55	0:00:00
MULTI4	15:11:52	8:28:52	0:40:29	153:34:56	185:41:49	50:48:39

Table 2: Mean/max tardiness in every policy and due date setting combination

k	Mean Reserve time (hh:mm:ss)			Max Reserve time (hh:mm:ss)		
	0.7	0.9	1.4	0.7	0.9	1.4
SPT	39:52:34	51:37:48	140:55:22	98:29:53	122:24:51	218:29:53
EDD	33:55:21	47:28:33	132:42:23	88:15:42	110:51:17	209:50:34
FIFO	38:24:14	47:35:56	139:11:22	88:59:53	112:50:23	218:00:34
LIFO	28:54:24	44:38:17	128:24:52	87:31:17	111:33:16	207:43:01
MOPNR	22:11:36	42:22:09	121:34:52	77:32:15	101:39:14	197:36:34
FASFS	27:03:02	42:25:04	127:32:55	88:35:42	112:33:11	208:38:56
LPT	31:33:13	45:54:03	125:00:01	84:48:46	108:40:26	204:38:14
FOPNR	28:55:55	46:07:31	130:34:34	88:39:53	112:36:55	205:09:30
LWRK	35:34:44	43:35:56	133:32:11	87:49:25	111:50:25	207:510:28
MWRK	23:05:49	42:28:29	122:21:34	81:54:47	105:55:37	197:39:14
MULTI1	37:24:11	46:09:04	133:52:25	92:15:31	115:13:49	206:39:28
MULTI2	36:51:55	45:57:55	132:19:31	89:01:35	112:00:57	210:31:26
MULTI3	32:00:12	46:28:48	128:32:49	89:17:34	111:46:05	207:53:09
MULTI4	24:34:38	43:00:01	124:34:51	91:54:23	99:27:02	206:46:09

Table 3: Mean/max reserve time in every policy and due date setting combination

	Mean Earliness (hh:mm:ss)	Max Earliness (hh:mm:ss)	Mean Flowtime (hh:mm:ss)	Max Flowtime (hh:mm:ss)	Mean Queue time (hh:mm:ss)	Max Queue time (hh:mm:ss)	Makespan (h)	Mean Manufacturing Cost ()
SPT	4:59:07	16:43:22	101:20:37	409:16:15	60:19:01	492:47:33	1600.42	82.667
EDD	5:19:03	16:59:03	96:45:23	174:05:51	60:24:03	210:39:16	1603.12	82.732
FIFO	5:26:42	17:11:47	94:07:21	182:38:30	59:09:03	212:18:20	1594.04	82.690
LIFO	5:07:47	16:52:32	110:17:44	378:58:20	68:47:16	388:21:13	1589.65	82.661
MOPNR	6:01:02	18:42:16	128:04:02	494:32:30	93:59:07	463:21:26	1604.09	82.639
FASFS	5:17:11	17:54:22	95:30:54	171:52:05	60:28:08	222:10:46	1588.43	82.768
LPT	6:06:51	18:58:48	99:14:46	209:28:20	68:13:17	376:04:56	1609.48	82.569
FOPNR	5:14:55	17:01:02	94:59:12	215:38:54	59:15:53	248:15:54	1601.92	82.605
LWRK	5:00:54	16:55:42	91:24:25	259:06:26	52:57:03	398:13:50	1604.84	82.714
MWRK	6:03:19	17:59:26	115:03:59	390:29:49	80:58:15	510:06:29	1618.16	82.550

Table 4: Due date independent performance indicators for dispatching rule policies

	k	Mean Earliness (hh:mm:ss)	Max Earliness (hh:mm:ss)	Mean Flowtime (hh:mm:ss)	Max Flowtime (hh:mm:ss)	Mean Queue time (hh:mm:ss)	Max Queue time (hh:mm:ss)	Makespan (h)	Mean Manufacturing Cost (€)
MULTI1	0.7	4:59:24	16:14:15	92:59:28	257:25:44	57:58:58	239:44:54	1590.25	82.173
	0.9	5:11:51	16:33:28	93:49:51	264:11:38	58:43:52	234:01:06	1586.76	82.288
	1.4	5:09:08	16:44:52	96:24:31	214:41:31	58:38:25	236:16:39	1587.21	82.452
MULTI2	0.7	4:44:30	15:54:32	96:32:43	334:43:52	55:02:36	285:52:16	1591.29	82.293
	0.9	4:57:52	15:57:48	96:34:08	303:09:09	55:33:45	319:35:43	1586.90	82.328
	1.4	4:53:56	15:59:49	95:08:42	306:27:32	56:10:04	345:49:16	1589.84	82.181
MULTI3	0.7	5:18:28	17:14:13	94:56:20	241:44:03	61:01:17	234:40:46	1583.76	82.522
	0.9	5:30:16	17:29:26	91:51:05	183:06:55	58:41:15	240:33:23	1560.90	82.581
	1.4	5:25:24	16:52:25	95:25:59	186:51:51	60:37:53	210:37:36	1584.67	82.753
MULTI4	0.7	5:27:42	16:58:51	100:25:34	249:34:56	64:29:40	297:35:12	1590.65	82.238
	0.9	5:29:56	17:01:33	99:55:01	314:08:49	63:50:13	330:51:25	1592.14	82.005
	1.4	5:13:33	16:35:56	98:42:45	266:48:39	61:11:45	311:13:21	1592.95	82.152

Table 5: Due date dependent performance indicators for multiple-criteria policies

k	Customer Service Level (%)			Mean Tardiness / Queue time ratio		
	0.7	0.9	1.4	0.7	0.9	1.4
SPT	65.59	82.80	96.77	0.303	0.192	0.061
EDD	59.14	78.49	100.00	0.174	0.050	0.000
FIFO	65.59	83.87	100.00	0.164	0.058	0.000
LIFO	64.52	78.49	92.47	0.383	0.281	0.100
MOPNR	58.06	68.82	88.17	0.428	0.328	0.141
FASFS	62.37	82.80	100.00	0.167	0.054	0.000
LPT	64.52	80.65	100.00	0.191	0.094	0.000
FOPNR	64.52	81.72	100.00	0.184	0.075	0.000
LWRK	72.04	87.10	97.85	0.209	0.114	0.012
MWRK	58.06	69.89	93.55	0.338	0.225	0.063
MULTI1	69.89	83.87	100.00	0.176	0.073	0.000
MULTI2	70.97	83.87	97.85	0.263	0.129	0.021
MULTI3	67.74	82.80	100.00	0.169	0.039	0.000
MULTI4	65.59	79.57	97.85	0.236	0.133	0.011

Table 6: Customer service level and mean tardiness/queue time ratio

	PREP WC	BLEND WC	CARD WC	SPIN WC	VAPOR WC	WIND WC	SPOOL WC	WEAV WC	Policy Mean
SPT	0.191	0.000	0.344	0.350	0.151	0.408	0.012	0.355	0.226
EDD	0.143	0.000	0.304	0.383	0.144	0.424	0.010	0.383	0.224
FIFO	0.085	0.000	0.273	0.393	0.129	0.430	0.007	0.384	0.213
LIFO	0.183	0.000	0.425	0.364	0.148	0.414	0.012	0.368	0.239
MOPNR	0.100	0.001	0.570	0.405	0.167	0.455	0.009	0.379	0.261
FASFS	0.094	0.000	0.287	0.396	0.134	0.423	0.007	0.384	0.216
LPT	0.178	0.000	0.297	0.425	0.157	0.460	0.013	0.455	0.248
FOPNR	0.147	0.000	0.292	0.379	0.146	0.422	0.010	0.379	0.222
LWRK	0.062	0.000	0.271	0.353	0.162	0.410	0.010	0.355	0.203
MWRK	0.137	0.000	0.464	0.413	0.130	0.458	0.008	0.455	0.258
MULTI1	0.085	0.000	0.300	0.380	0.098	0.409	0.006	0.398	0.209
MULTI2	0.060	0.000	0.277	0.370	0.110	0.395	0.010	0.381	0.200
MULTI3	0.138	0.000	0.302	0.393	0.099	0.422	0.006	0.400	0.220
MULTI4	0.120	0.000	0.344	0.402	0.065	0.431	0.006	0.453	0.228
WC Mean	0.123	0.000	0.339	0.386	0.131	0.426	0.009	0.395	0.226

Table 7: Critical queue ratios (CQR) for each workcenter and for each operational policy

k	FS - Mean Earliness (hh:mm:ss)			HBS - Mean Earliness Reduction (%)		
	0.7	0.9	1.4	0.7	0.9	1.4
SPT	32:23:43	64:47:26	149:01:07	84.61	92.31	96.65
EDD	30:15:16	60:30:31	139:10:12	82.42	91.21	96.18
FIFO	27:55:01	55:50:03	128:25:07	80.50	90.25	95.76
LIFO	32:12:34	64:25:07	148:09:47	84.07	92.04	96.54
MOPNR	26:27:32	52:55:05	121:42:41	77.26	88.63	95.06
FASFS	30:53:34	61:47:07	142:06:23	82.89	91.44	96.28
LPT	28:16:01	56:32:02	130:01:42	78.37	89.18	95.30
FOPNR	28:48:11	57:36:22	132:29:38	81.78	90.89	96.04
LWRK	29:40:44	59:21:29	136:31:24	83.10	91.55	96.33
MWRK	27:25:11	54:50:22	126:07:50	77.92	88.96	95.20
MULTI1	32:05:16	64:10:31	147:36:12	84.45	91.90	96.51
MULTI2	31:13:22	62:26:43	143:37:27	84.81	92.05	96.59
MULTI3	30:01:40	60:03:19	138:07:38	82.32	90.83	96.07
MULTI4	27:39:34	55:19:07	127:13:59	80.25	90.06	95.89

Table 8: Mean earliness performance comparison between FS and HBS

Rule	Description
SPT	Task from the Job with the shortest processing time is selected
LPT	Task from the Job with the longest processing time is selected
EDD	Task from the Job with the earliest due date is selected
FIFO	Task from the Job which first arrives at the factory is selected
LIFO	Task from the Job which last arrives at the factory is selected
MOPNR	Task from the Job which has the most operations remaining to be performed is selected
FOPNR	Task from the Job which has the fewest operations remaining to be performed is selected
FASFS	Task from the Job which arrives first in the job shop is selected
LWRK	Task from the Job which has the least work remaining to be performed is selected
MWRK	Task from the Job which has the most work remaining to be performed is selected

Table 9: Description of the dispatching rules utilized as operational policy

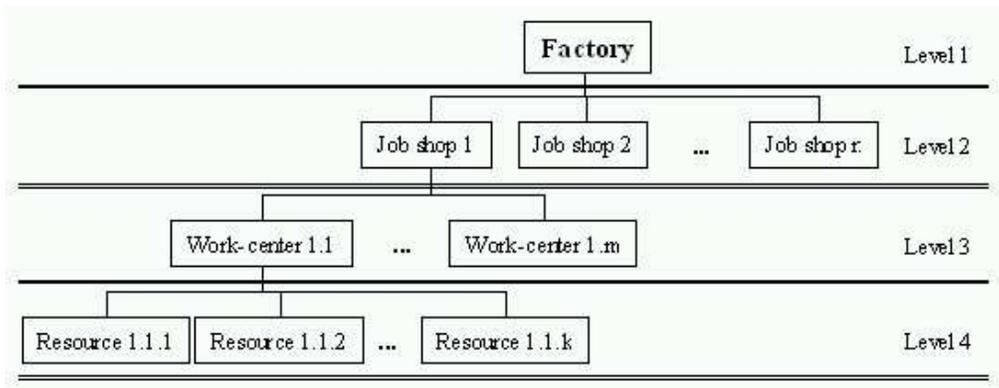


Figure 1: The four-level hierarchical facility model of a discrete manufacturing system

Peer Review Only

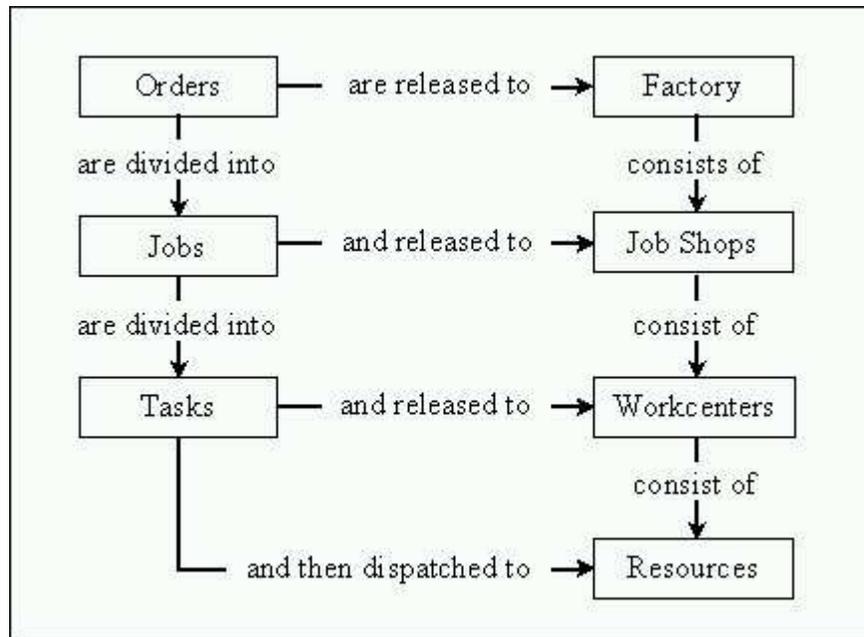


Figure 2: Correlation between the facility and workload hierarchical models

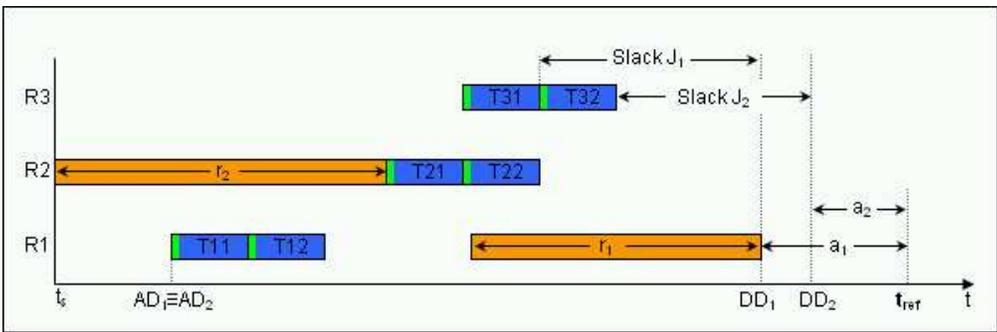


Figure 3: Finite capacity forward scheduling (FS) using the HBS framework

Peer Review Only

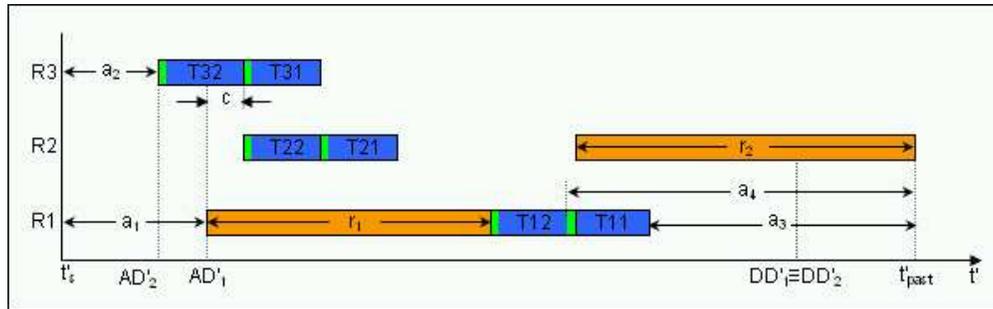


Figure 4: Forward finite capacity scheduling in fictitious time scale (HBS - first stage)

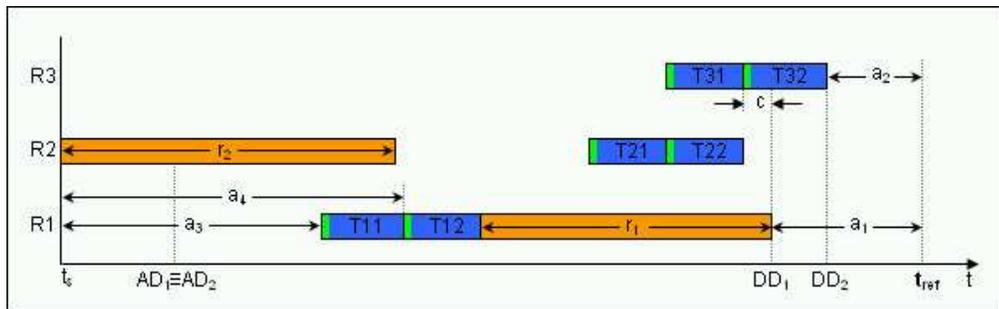


Figure 5: Backward finite capacity scheduling in real time scale (HBS - second stage)

Peer Review Only