Available online at www.sciencedirect.com
SCIENCE DIRECT®
European Journal of Operational Research 167 (2005) 644–662
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/dsw

Decentralised, multi-objective driven scheduling for reentrant shops: A conceptual development and a test case ☆

Giovanni Miragliotta [a,*], Marco Perona [b,1]

[a] *Dipartimento di Ingegneria Gestionale del Politecnico di Milano, via G. Colombo 40, Milano 20133, Italy*
[b] *Dipartimento di Ingegneria Meccanica dell'Università di Brescia, via Branze 38, Brescia 25123, Italy*

Available online 1 September 2004

Abstract

This paper presents a new approach to the scheduling of reentrant shops. Its main innovative principle is an *objective driven* engine: jobs to be processed are selected on the basis of a balanced evaluation of how well they fulfill efficiency and effectiveness objectives. This new approach relies on a heuristic algorithm build upon a decentralised architecture, in which each production resource can act as an independent decider and selects the jobs to process according to dynamically changing criteria and to widely shared information. Interesting performances, as well as robustness and real life suitability, have been highlighted through an extensive test phase based on real data collected during two case studies, belonging to semiconductors and metalworking businesses.

Keywords: Scheduling; Multi-objective; Heuristic; Decentralised; Reentrant shops

1. Introduction

The short term planning of job shop systems represents a very complex managerial problem due to the large number of variables and to counterintuitive relationships among them: for these reasons, companies have traditionally approached this problem through *scheduling rules*, i.e. through simple and local decisional criteria (e.g. the shortest processing time rule) which can perform quite well in pursuing a specific objective (e.g. the minimisation of the mean flow time) (cf. Backer, 1974; Blackstone et al., 1982; Garetti et al., 1989).

Nevertheless, a profound evolution in the competitive needs is now forcing companies to simultaneously pursue interdependent and sometimes

☆ This paper is the result of a collaboration among authors. Nevertheless, Sections 1 and 7 were written by Marco Perona, while Sections 2–6 were written by Giovanni Miragliotta.

* Corresponding author. Tel.: +39 02 2399 2785.
E-mail addresses: giovanni.miragliotta@polimi.it (G. Miragliotta), perona@ing.unibs.it (M. Perona).
[1] Tel.: +39 03 0371 5446.

0377-2217/$ - see front matter
doi:10.1016/j.ejor.2004.07.013

contradictory objectives. Within this scenario the traditional scheduling rules approach is unfit, since it cannot properly support multi-objective scheduling.

This new target, we notice, is a challenging one: job shop systems, in fact, are characterised by uncertain variation in terms of workload, production mix and machines availability. Moreover, they might encompass many different resources (e.g. sequential and batch machines, setup dependent and non-setup dependent machines and so on), so that a new reactive, and resource tailored approach is needed. The same considerations stay true, a fortiori, if we consider very complex job shops, such as reentrant shops. Reentrant shops are production systems in which jobs may "loop back" within a sub set of the production resources, thus leading to very long and recursive routes (see Fig. 1).

Reentrant shops are quite common, mainly in the semiconductor business (cf. Miller, 1990; Cigolini et al., 1996), but some noteworthy examples can be found also in the metal working industry and automotive components manufacture (cf. Hwang and Sun, 1998). In these cases the concurrent optimisation of the utilisation of the costly production equipment and of the customer service performances is almost impossible to be achieved by means of a traditional scheduling rules approach. The purpose of this paper is thus to discuss a new scheduling approach compliant with these requirements: to this extent, the paper is arranged as follows. Section 2 presents the conceptual background of the considered subject through a literature review. Section 3 states the objectives of the paper, while Section 4 presents in detail the new approach. Section 5 illustrates the cases used to test the new method, and Section 6 points out the most interesting results; finally, Section 7 summarises the main results and suggests promising developments for this research path.

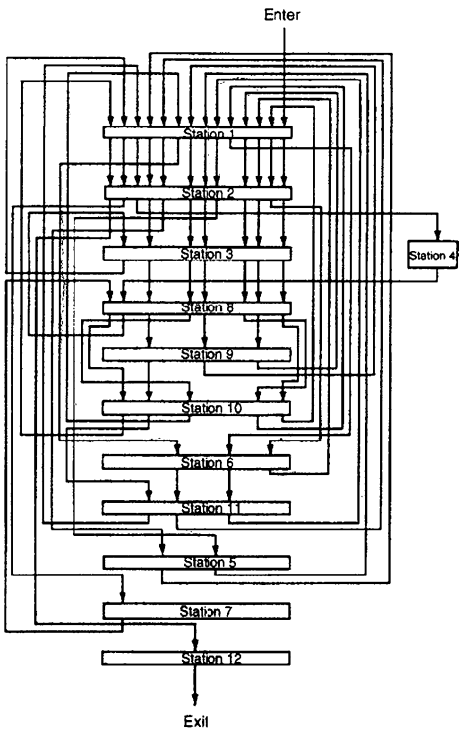

Fig. 1. Example of reentrant job shop (Kumar, 1994).

2. Literature background

Due to the variety of job shop production systems, as well as to the fact that the scheduling problem can actually be outlined into three distinctive decisional stages (order review and release, dispatching and routing; cf. Bechte, 1988; Bergamaschi et al., 1997; Cigolini et al., 1998), it is impossible to handle here a comprehensive literature review. Therefore, we will focus only on the most innovative scheduling approaches: more specifically, Section 2.1 deals with innovative scheduling approaches for general job shops, while Section 2.2 focuses on innovative scheduling approaches for reentrant shops only.

2.1. New trends in job shop scheduling

A first trend that clearly emerges in job shop scheduling is that one of *decentralised decision making*. Distributed problem solving is a key research area since many years. Yet in 1987, Decker (1987) published a survey on distributed problem solving techniques for manufacturing applications, giving evidence of the richness of this area; a more recent survey can be found in Jones and Rabelo (1998). Within this area, a contemporary milestone

is the work of Holthaus and Ziegler (1997), who propose a decentralised scheduling approach in which the different work-centres are targeted on local objectives but communicate with each other in order to keep their queues at a previously defined level. On this path, Artificial Intelligence approaches are widely adopted to increase the effectiveness of the scheduling system: for instance expert systems and genetic algorithms (cf. Lee et al., 1997), the Holonic paradigm (cf. Bongaerts et al., 1997) and multi-agents systems (cf. Roy and Anciaux, 2001).

The paper by Lee et al. (1997) offers a further innovation since the dispatching rules are *space distributed* among the various machines, and an adaptation routine is designed for each machine, so that an increasing differentiation is dynamically achieved. This is an important aspect, given that Barman (1997) proved that the space distribution of the dispatching rules can achieve significant improvement in production system performances. The same principle, but applied to the time axis (*time distributed* routines), can be found in Pierreval and Mebarki (1997), in which two sets of dispatching rules are defined for every machine within the system: a standard set and an emergency set which can be activated accordingly to system workload status.

2.2. New trends in reentrant shop scheduling

As it was mentioned before, reentrant shops cover a wide variety of applications, of which the most important one is semiconductor manufacturing; wafer production process is basically made of several cycles of resist coating, latent image building, ion implantation and resist removal. Some introductory material about semiconductor manufacturing can be found in Dayhoff and Atherton (1986) and Lu et al. (1994); in addition Johri (1993) provided a clarifying overview about requirements and problems of reentrant shops for semiconductor manufacturing, while Uzsoy et al. (1992) provided a complete reference of the scheduling approaches developed for that industry up to 1994. Since reentrant shop scheduling is a wide research subject, many consolidated research streams have been deepened during the last years. For instance, while Kumar (1994) and Kumar and Kumar (1994) introduced new centralised scheduling rules to reduce the mean cycle time and its variance, Kim et al. (1998a) developed a set of tools to define new release and dispatching rules for the whole system, as well as for specific machines. In this area, Kim et al. (1998b) addressed also very specific aspects such as the scheduling of photolithographic masks with non-zero setup times. Yet, other interesting trends in reentrant shop research are linked to approaches different from discrete time simulation: for instance, analytical or numerical optimising approaches (cf. Glassey et al., 1996; Zhou and Jeng, 1998), queue theory, continuous time simulation (cf. Kouikoglou and Phillips, 1997; De Souza and Lin, 1997), even fluid-flow based modelling tools (cf. Connors et al., 1994) have been studied by researches worldwide.

Nevertheless, we can say that the pattern of most interesting research works in this field is quite consistent with the research trends depicted in Section 2.1. Baek et al. (1998), for instance, proposed a *Spacial Adaptive Procedure* which progressively differentiates the dispatching rules from machine to machine according to the selected target. Moreover Nakata et al. (1999) developed a time distributed approach to dynamically adapt the dispatching rules as a function of the current workload; the same authors distinguished the *job oriented* objectives (which pertain to the effectiveness performance area, and concern the respect of the due date) from the *shop oriented* objectives (which pertain to the efficiency performance area and are mostly related to the utilisation of capital intensive machines), and tried to harmonise them in a new rule.

Another important innovation in this field is represented by the introduction of a "*time depth*" dimension in the information availability: Li et al. (1996) developed algorithms which try to exploit information about the future state of the system; the same did Kutanoglu and Sabuncuoglu (1999) who relied on future information to eventually keeping a machine idle to wait for an urgent job arriving in few minutes' time. This kind of decisions, quite unusual in the classic job shop context, can be very useful to schedule batch ma-

chines, which are common in the semiconductor context. Many other works dealing with batch centres in a semiconductor reentrant shop can be found: for instance Chandra and Gupta (1997), Yi-Feng (1998) and Mehta and Uzsoy (1998). They all deepened specific issues (e.g. batch sizing or batch sequencing) in order to maximise the performances of such resources within the semiconductor production process.

Finally, another innovative idea developed in the reentrant shop context is the exploitation of upstream information; this intuition was developed by Fowler et al. (1992) and Weng and Leachmann (1993). On this research line, Robinson et al. (1995) extended the time depth dimension toward a *space depth* dimension of available scheduling data; following this model, each machine knows the workload, the queue condition and the needs of any other machine within the reentrant route, and coherent decisions can thus be taken on the basis of this enlarged information set.

3. Research questions and objectives

As highlighted in Section 2, four main trends, or perhaps principles, are permeating contemporary research in reentrant shops scheduling, and all of them have proved to be valuable in improving scheduling practice in this complex field. Such principles are:

(i) *decentralisation* of the scheduling responsibility, by allowing each resource to autonomously decide its scheduling policy;
(ii) increase of *space and time depth* of the available information, by allowing each decider in the system to access information about other resources and about the future state of the system;
(iii) *space and time distribution* of the scheduling rules, so that each resource can act in a way which is tailored to its very nature and which follows the requirements of the specific time instant;
(iv) *multi-objective driven pull scheduling* based on physical drivers, so that specific multi-objective strategies can be pursued.

Nevertheless, none of the quoted papers has tried to put together these four features into a unique model. So the purpose of this paper is to give an answer to the following research questions: since each of these innovative principles has proved to improve the practice of reentrant shops scheduling, *which is the further improvement that could be achieved if these four streams were harmonised into a unique scheduling approach*? And *how could such complex algorithms be put at work together*?

Hence the main focus of this contribution is twofold: to present a method through which the four principles can been unified, and the outcomes of this unification. Moreover, in contrast with most authors who tested their approaches on simplified test beds, the new approach will be tested through real data coming from two different manufacturing contexts, so to better evaluate the actual ability of the model to improve shop performances. The new model's name is 'RESDES' (REentrant Shop DEcentralised Scheduling).

Since the framework of RESDES is quite complex, the next section is entirely devoted to present its architecture, as follows: decentralisation aspects, Section 4.1; space/time depth of available data, Section 4.2; space/time distribution of the scheduling rules, Section 4.3; objective drive pull scheduling, Section 4.4.

4. RESDES conceptual framework

4.1. The decentralised approach and the communication architecture

Within the RESDES architecture, each workcentre in the shop is considered as an independent decision maker. Decisions to be taken are those concerning every scheduling activity, i.e. order release, dispatching and routing. To this regard, the *pre shop pool* is treated as a queue, and therefore can be managed with the same rules as machine's queue.

According to the decentralisation principle, each resource has an autonomous decisional capability; the information used to decide are not only

those locally available (e.g. local queue condition), but include technical data (e.g. the production routes) and further information about other work-centres' needs (e.g. job A-1 needed by work-centre X-1 to complete a batch load).

To allow effective information sharing among work-centres about each other's needs, we resorted to a *Request Sharing Table (RST)*: this table can be read by every machine in the system, but only a selected group of critical work-centres is allowed to write on it (cf. Section 4.3). The basic functioning of this communication architecture is quite simple: critical work-centres can issue requests to react to their queue conditions, while any other work-centre can read the requests and act in order to fulfil them. If a request has been fully/partly satisfied, the relevant work-centre updates the RST; otherwise, the request is automatically deleted after its Request Expiration Time (RET) has come.

The concept of "critical work-centre" will be thoroughgoing discussed in Section 4.3; anyway the basic idea here is to consider as critical each work-centre that can seriously disrupt the overall shop performances if not carefully scheduled.

4.2. Space and time depth of the scheduling data

To allow the decentralised decision making approach to perform at best, a wide amount of information is needed. This section is devoted to clarify how this is obtained and what kind of information is provided to each single work-centre.

The basic tool engineered in order to enhance the *space depth* of information availability is, of course, the RST, whose basic architecture has been sketched above. This table may contain three different types of requests, which are issued by critical resources in order to pursue their specific objectives, as follows.

(i) *Normal requests* are issued in order to optimise the scheduling sequence of critical work-centres. To attain this target, normal requests contain precise indications about the job type, the number of units needed and the Request Expiration Time (RET).

(ii) *Expedition requests* are issued in order to avoid a potential starvation of a critical work-centre. They do not specify any of the above data since, no matter which job is provided, the starvation risk must be avoided.

(iii) *Inhibition requests* are issued in order to avoid excessive queue's length, both due to ordinary system dynamics and to extraordinary events (e.g. breakdown). They do not specify any field, since they simply aim at stopping the flow arriving at the request issuing work-centre. The request will be deleted once the queue overflow risk is over.

In addition to the above mentioned information, each request is characterised by two additional data. The first one is the *Requiring resource identity*, and will be used to estimate the route distance between the reading and the requiring resource. The second information is the *Request Importance (RI)*: this parameter (which will be set through the simulation analysis) is related to the specific event that has triggered the request: for instance, the starvation of a capacity critical resource might be more critical than an inefficient batch load, so the first request will have a RI parameter higher than the second. Therefore, resorting to the RST, critical work-centres can not only make public their needs, but also give them a priority. Appropriate procedures have been developed to control the RST dynamics: for instance, when a requested job arrives at the requiring work-centre, the corresponding request is automatically deleted from the RST.

Another key requirement of the RESDES development is to enhance the *time depth* of the available information. This objective was pursued through two different actions, one related to the dispatching criteria of the batch work-centres, and one related to the anticipatory capability of single work-centres.

With regard to the first aspect, it may be useful to briefly discuss what kind of peculiarity arises in the short term planning of batch machines. These machines (see also Cigolini et al. (1999) for more details) are characterised by a *minimum batch size* $B_{\min}$, under which the production process cannot be run for technological reasons, and by a *maxi-*

mum batch size B_{max}, which corresponds to the volume limitation. Thus, if there is not a full batch available to be load, it could be clever to wait until some new jobs arrive to the queue, rather than immediately starting to process a partial load. In other words, scheduling batch machines opens up a wider decisional space, since the question to be answered is not only "*which group of jobs to pick from the queue?*", but also "*when to pick?*". To address this issue, we resorted to the *Wait No Longer Than* (WNLT) rule, which has been developed and successfully tested by Cigolini et al. (2002).

With regard to the second aspect, all technical data are available at any work-centre: in this way, while evaluating a request, each machine can estimate the "minimum time distance" between itself and the work-centre which has issued a specific request: this is done simply by adding up processing + setup times of any intermediate step; this is a lower bound of the actual time distance because no queuing times are considered. Anyway, requests which for sure will not be able to comply with the RET will not be taken into account, thus increasing RESDES routines' efficiency.

4.3. Space and time distribution of the scheduling rules

As it was mentioned in Section 4.1, some production resources, because of their importance in the shop floor, are considered as critical and thus have the right to drive the scheduling process by addressing requests about their needs. Now it is important to clarify how these resources are singled out, how each resource is provided with a tailored decisional capability and how this decisional capability may change in time. This is done with reference to Fig. 2, that presents a work-centre

		Process flow type **Flexibility Critical: sequence dependent setup?**		
		NO		**YES**
		Batch Machines	**Sequential Machines**	
Capacity critical: bottleneck?	**YES**	**Cell 6**	**Cell 3**	**Cell 4**
		RST: read / write	**RST**: read / write	**RST**: read / write
		OBJ: maximise net utilization	**OBJ**: avoid starvation & bottleneck	**OBJ**: avoid starvation & optimize sequence
		BEHAV: independent	**BEHAV**: pulled / independent	**BEHAV**: independent
		RULE: Max_j (B_j / B_{max}) + WNLT	**RULE**: BAL + SPT	**RULE**: Setup Sharing
	NO	**Cell 5**	**Cell 1**	**Cell 2**
		RST: Read only	**RST**: Read only	**RST**: read / write
		OBJ: maximise time utilization	**OBJ**: help other workcentres to achieve their goals	**OBJ:** help other workcen tres to achieve their goals
		BEHAV: independent	**BEHAV**: pulled	**BEHAV**: pulled / independ ent
		RULE: Max_j (B_j / B_{max})	**RULE**: BAL	**RULE**: BAL + Setup Sharing

Fig. 2. Classification and characteristics of work-centres.

classification table. The vertical axis of Fig. 2 concerns *capacity criticality*. A work-centre is capacity critical if its *average* utilisation in the planning period (computed through the current orders' portfolio) is higher than a specific threshold (cf. Section 5.3 for the definition of this threshold). Due to such relevant workload, capacity critical work-centres must avoid any starvation that could arise because of random system dynamics: this in fact could turn into a permanent loss of production capacity. Conversely, they could easily become bottlenecks, therefore disrupting the jobs' lead time: therefore all capacity critical work-centres have the right to write in the RST.

The horizontal axis of Fig. 2 concerns *process flow type*, and three categories are here defined: sequential machines, with a further distinction between sequence dependent and non-sequence dependent setup times, and batch work-centres (which usually are non-sequence-dependent). Sequential machines that are characterised by sequence-dependent setup times are considered as flexibility critical, since at these work-centres a flow time reduction could be achieved through an accurate sequencing of the queued jobs: therefore they are allowed to issue requests to the RTS. This classification determines six different resource groups, each with a specific objective, scheduling behaviour, and RST access rights: these aspects are described in detail as follows.

Cell 1 groups non-critical sequential work-centres: such resources have no peculiar target to pursue, except for that of helping the rest of the system to reach its goals. Therefore, these work-centres are always pulled by other resources' needs, through the RST; consistently, they can only read the table. A new specific dispatching rule has been designed for these work-centres, named BAL: it embeds a balanced evaluation of the shop oriented needs (efficiency, represented by the requests issued by other work-centres) and of the job oriented needs (effectiveness, in terms of due date respect). Section 4.4 of this paper is entirely devoted to describe the functioning of this rule.

Cell 2 encompasses flexibility critical sequential work-centres: resources within this cell are quite similar to those within Cell 1 and so, in normal conditions, they have the same objective and are managed following the same dispatching rule (BAL rule) as those within Cell 1. Nevertheless, the sequence-dependent setup could lead, in some conditions, to rather long queues: when this happens they switch toward an independent, setup optimising behaviour so to restore appropriate queuing times. The threshold which switches from dependent to independent behaviour is set as follows: each work-centre in Cell 2 keeps track of the queuing time of the last 100 processed jobs, computing in real time the mean value ($\mu_{QT-W}(t)$) and the standard deviation ($\sigma_{QT-W}(t)$) of this parameter. So, when the queue's length is becoming greater than $\mu_{QT-W}(t) + 2 \cdot \sigma_{QT-W}(t)$, then the work-centre begins to act independently, and resorts to the *Setup sharing* rule. According to this rule, the selection of the type j of the next jobs to be processed is made by computing the SS_j index, as follows:

$$SS_j = \text{Setup}_{\text{current job}\Rightarrow\text{job type } j}/\text{Number of jobs of type } j \text{ in the queue.} \quad (1)$$

Then the jobs of type j with the minimum SS_j index are selected and processed. In case two or more job types should rank the same, the job type with the higher number of job is selected, so to shorten the queue length. The work-centre switches back to the dependent behaviour as soon as the current queue length falls under the $\mu_{QT-W}(t)$ threshold.

Cell 3 consists of capacity critical work-centres with no sequence dependent setup: their specific objective is twofold: to avoid starvation, since this could disrupt the overall shop's output, and at the same time to avoid to become a dynamic bottleneck of the system, in order to prevent job's flow time to get out of control. Given this objective, their normal dispatching behaviour can be a pulled one, since the jobs to be processed can be selected according to other work-centres' needs; only in case of excessive queue length they switch toward an independent, flow time minimising behaviour in order to restore appropriate queuing times, as it happens for work-centres in Cell 2. Therefore, when the queue length is "normal", the BAL rule is used, while if the queue is becoming too long, a

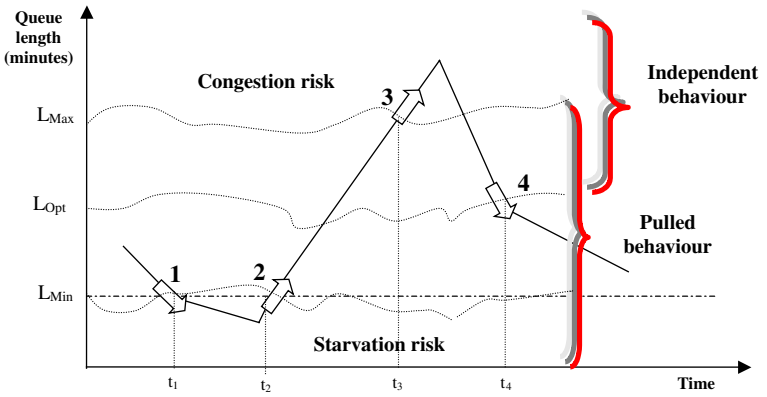

Fig. 3. Cell 3 rule switching criteria.

Shortest Processing Time rule is activated in order to reduce the jobs' flow time. The switching model implemented within Cell 3 is a slightly more complicated that used for Cell 2, and is illustrated in Fig. 3. Each work-centre's queue has been divided in four regions by defining three thresholds (L_{Min}, L_{Max} and L_{Opt}), which are expressed as minutes of work, and are dynamically computed as follows. $L_{\mathrm{Min}}(t)$ indicates the queue length that could lead to a starvation of work-centre W. This level has been set by recording the inter-arrival time of the last 100 jobs arrived at work-centre W, and then computing the mean value. When the workload in the queue is smaller than the current mean job inter-arrival time, then a starvation risk is detected. $L_{\mathrm{Max}}(t)$ indicates the queue length that could lead to a congestion risk. $L_{\mathrm{Max}}(t)$ has been defined as $\mu_{QT-W}(t) + 2 \cdot \sigma_{QT-W}(t)$, where $\mu_{QT-W}(t)$ and $\sigma_{QT-W}(t)$ have the same meaning as illustrated for Cell 2 machines, and are computed the same way. $L_{\mathrm{Opt}}(t)$ is simply computed by averaging $L_{\mathrm{Min}}(t)$ and $L_{\mathrm{Max}}(t)$.

In order to provide for a complete comprehension of the designed system, let's follow a generic story line, as that depicted in Fig. 3. At time instant t_1, the work-centre enters in the starvation risk area. An expedition request is then issued toward the RST, while the corresponding work-centre keeps being pulled by the BAL rule. At time instant t_2, thanks to the increased arrival rate, the work-centre exits the starvation risk area and the previously issued request can be erased. Later, for other reasons, at time instant t_3 the machine enters the congestion risk area. An inhibition request is then issued and the machine begins to act independently; to reduce local flow time, a Shortest Processing Time rule is activated. Finally, at time instant t_4, when the L_{Opt} threshold is crossed downward, the machine comes back into the normal operating area, the inhibition request is erased and the machine goes back to the pulled behaviour. As can be noticed, this system has been designed to have a certain inertia before going back to the pulled behaviour, because jobs would naturally flow from upstream to downstream work-centres following their production route.

Work-centres in *Cell 4* are sequential machines, and share both capacity and flexibility criticality. Consistently, they always behave as independent ones, according to a setup minimisation approach (*Setup Sharing*, cf. Cell 2).

Non-capacity critical batch work-centres belonging to *Cell 5* have the objective of maximizing time utilisation, so to reduce system flow times. Therefore they always act independently, even though they are not affected by capacity criticality; this choice was made because some preliminary simulation work made it clear that it is quite difficult and potentially ineffective to pull production at a batch work-centre, due to the batching itself. The rule we resorted to is very simple: as mentioned in Section 4.2, batch machines are characterised by a minimum batch size B_{min} below which the production process cannot be run for technological reasons. Defined as B_j the number of jobs of type j currently in the queue, the batch

type j to be processed is selected according to a $\max_j(B_j/B_{\max})$ criterion. This rule performed quite well, since in case more than one job type has $B_j \geqslant M_{\max}$, then this rule aims at leaving to wait those job types which, after being subtracted of a full batch load, are expected to require more time to reach again the $B_{\max}$ threshold. Conversely, in case no job type j satisfies the $B_j \geqslant B_{\max}$ condition, provided that at least one job type j satisfies $B_j \geqslant B_{\min}$, the chosen rule would select that job type which is closer to the $B_{\max}$ value, so pursuing a maximisation of volume utilisation.

Finally, capacity critical batch work-centres are arranged within *Cell 6*. Their objective is to maximise net utilisation, which is computed by multiplying volume utilization (i.e. how much of the available volume $B_{\max}$ is on average loaded) by time utilization (i.e. how much of the up time is actually exploited for production). Thus, they issue requests to rearrange their work-load mix, and always act independently to have a less constrained decisional space. The dispatching rule utilised for work-centres in this cell when $B_j \geqslant B_{\max}$ is the same as in Cell 5. In case no job type j satisfies the $B_j \geqslant B_{\max}$ condition, work-centres in Cell 6, being capacity critical, are kept idle for a maximum waiting time of WNLT periods. If the situation does not change during WNLT, provided that at least one job type j satisfies $B_j \geqslant B_{\min}$, the job type j with $\max_j(B_j/B_{\max})$ is selected and processed (see Cigolini et al., 2002).

4.4. The BAL rule for objective driven pull scheduling

The last aspect to be treated to complete the explanation of the RESDES approach is related to the description of the BAL rule. BAL is a new scheduling rule developed with the objective of concurrently optimising effectiveness and efficiency performances: jobs to be dispatched are selected by taking into account *job oriented* performances (i.e. due date fulfilment) and *shop oriented* performances (i.e. the achievement of likely workload and utilisation targets). The BAL rule, as illustrated in Fig. 4, can be considered as an advanced combination rule, i.e. (cf. Holthaus and Rajendran, 1997) a rule which resorts to both process and due date information to properly schedule a job. [2]

The BAL rule differs from standard combination rules both in the content (e.g. process information is substituted by the requests in the RST) and in functioning, i.e. in way the two components are weighted, which will be discussed later.

When a job has to be picked from a queue of a pulled work-centre, an index is computed for every job in that queue. This index is based on a weighted evaluation of two terms: the *job oriented* and *the shop oriented* components. These two components are mixed and the highest priority job is selected.

Before introducing the inner functioning of this rule, we would like to point out that the BAL rule is used also for order release and, with some modifications, for routing decisions. More in detail, job release timing is regulated by a Constant WIP criterion (cf. Spearman et al., 1990), and a new job is released only when a finished job leaves the production system: the decision of which job to be released is treated as an ordinary dispatching decision, under the assumption that the pre-shop pool is a normal queue. Conversely, routing decisions are taken by analysing the *shop oriented* component only; therefore, once a job has been processed, it will be addressed toward the highest request issuing machine. In case no request is available, a minimum queue criterion is adopted, so to balance workloads within the system.

4.4.1. Job oriented component

The job oriented component tries to evaluate if a job is going to be late. To this extent, let's define the slack time of a job j computed at time t as follows:

$$\text{Slack}_{j,t} = \text{Due Date}_j - t - \sum_{s \in Sj} (\text{Setup}_{j,s} + \text{Processing}_{j,s}), \qquad (2)$$

where S_j is the set of the remaining steps of job j at time t; s is a pointer which refers to generic remaining step in S_j; $\text{Processing}_{j,s}$ is the processing time of

[2] A well-known combination rule is the Critical Ratio (cf. Blackstone et al., 1982).

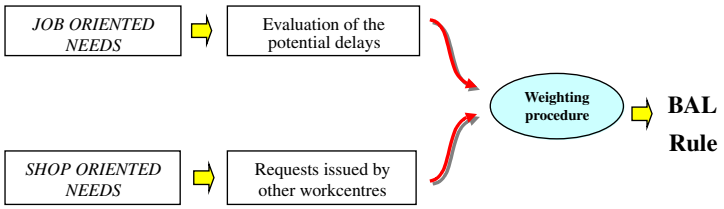

Fig. 4. Components of the BAL rule.

the sth step of job j; [3] $\text{Setup}_{j,s}$ is the setup time of the sth step of job j. [4]

According to Eq. (2), the jobs at a given queue can be classified in four different categories depending on their slack time, as follows.

(i) *Timely jobs*. Jobs j belonging to this category can wait to be processed, since their slack will remain positive even if the worse (i.e. longest time) job is processed before them. This condition is expressed as follows:

$$\text{Slack}_{j,t} > \max_{q \in Q}(\text{Setup}_{q,s'} + \text{Processing}_{q,s'}), \tag{3}$$

where Q is the set of the jobs $q \neq j$ waiting in the same queue; q is a generic job waiting in the same queue; s' is a pointer to the next step of job q. $\text{Setup}'_{q,s}$ and $\text{Processing}'_{q,s}$ are setup and production times of job q, referred to the same work-centre where job j is.

(ii) *Conditioned timely jobs*. Jobs j belonging to this category should start their processing immediately, otherwise the delay introduced by processing another job in the queue might compromise the respect of their due date. In formula:

$$0 < \text{Slack}_j < \max_{q \in Q}(\text{Setup}_{q,s'} + \text{Processing}_{q,s'}). \tag{4}$$

(iii) *Expected late jobs*. Jobs j belonging to this category will not be delivered on time, even if at present time they are not explicitly late. In formula:

$$\text{Slack}_j < 0. \tag{5}$$

(iv) *Already late jobs*. Jobs j belonging to this category are already late, since their due date is already gone by. In formula:

$$\text{Due date}_j < t. \tag{6}$$

When a job has to be picked from a queue, each job in the queue is arranged in the right category, and then the categories are sorted according to the specific effectiveness measure used. In fact, different key performance indicators (KPIs) could be selected to measure the effectiveness performance: for instance, the mean tardiness, rather than the number of late jobs, etc. Depending on the selected KPI, different sorting criteria should be used to have a coherent set-up of the system. Table 1 illustrates how categories should be sorted depending on the effectiveness KPI selected.

In the remainder of this paper, we selected the mean tardiness as the key performance indicator to measure the RESDES effectiveness performances out of the simulation campaign. So the ranking criterion in the right column in Table 1 has been coded and implemented. Once categories have been sorted, jobs within the highest non-empty category are sorted again so to have an unequivocal sorting. This is easily achieved through the criteria in Table 2.

The whole of these steps assures the capability to unequivocally sort the jobs waiting to be processed taking into account the job oriented component.

[3] In case more than one work-centre is available at a given step, the main resource should be considered to perform this computation.

[4] Please note that many work-centres have sequence dependent setups, but when computing $\text{Slack}_{j,t}$, it is not possible to foresee which job would precede job j. Hence, $\text{Setup}_{j,s}$ is estimated by computing the average setup time when passing from a generic job to job j at that step.

Table 1
Coherent sorting according to the selected effectiveness KPI

Rank	Effectiveness KPI	
	Number of late jobs	Mean tardiness
1	Conditioned timely	Already late
2	Timely	Expected late
3	Already late	Conditioned timely
4	Expected late	Timely

Table 2
Intra-category jobs sorting criteria

Category	Within category jobs sorting criteria
Timely	Ascending order of $\text{Slack}_{j,t} - \max_{q \in Q}(\text{Setup}_{q,s'} + \text{Processing}_{q,s'})$
Conditioned timely	
Expected late Already late	Ascending order of $\text{Slack}_{j,t}$

4.4.2. Shop oriented component

The *Shop Oriented* component of BAL is related to the requests issued for a certain job type j by the work-centres in the shop floor. More specifically, a shop priority index is computed for each job j as the sum of the Request Importance RI (cf. Section 4.2) of every request concerning j: therefore, a job with no requests has no priority from the shop oriented point of view. Eq. (7) illustrates how the shop priority index is computed.

$$\text{Shop Priority}_j = \sum_{r \in Rj} \text{RI}_r, \tag{7}$$

where R_j is the set of requests referring to job type j and r is the pointer referring to the request.

It may occur that more than one job within the same queue has the same *shop oriented* importance; in these cases, the tie will be broken by combining the *shop oriented* and *job oriented* components: in fact, as mentioned above, the *Job Oriented* component is always able to produce an unequivocal sorting.

4.4.3. Job and shop components combination

Three different procedures have been proposed to merge the two components: the first one clearly favours the *job oriented* component, the second one is more focused on the *shop oriented* component, while the third one strives for a balanced solution. The first procedure is called *Job* ⇒ *Shop*; it starts with the arrangement of available jobs into the four categories described in Section 4.4, and proceeds with the category ranking according to the effectiveness objective performed (cf. Table 1). Then the intra-category sorting is made by resorting to the computation of the Shop Priority$_j$ values, rather than by following the criteria indicated in Table 2.

The *Shop* ⇒ *Job* balancing procedure is somehow mirror-like. It starts computing the Shop Priority$_j$ index for each job in the queue, and then sorting the available jobs in descending Shop Priority$_j$ order. Then, the top 25% of the list is picked, and these jobs are arranged according to the job oriented sorting procedure.

The *Mixed* criterion acts as follows. As before, each job has a Shop Priority Index and belongs to a certain timeliness category. Then a C_j parameter is defined for each job: C_j is equal to the resulting category ordinal position. In our case, since the selected effectiveness KPI is the mean tardiness then (cf. Table 1, right column) jobs belonging to the Timely category will have $C_j = 4$, jobs belonging to the Conditioned Timely category will have $C_j = 3$, etc.

The final priority for each job in the queue is obtained by computing the Mixed Priority index, as follows, and by sorting the available jobs in descending order of this index. Top-of-the-list job will be picked.

$$\text{Mixed Priority}_j = \text{Shop Priority}_j / C_j. \tag{8}$$

The rationale of this index is that the score gathered by the shop oriented component is *proportionally* decreased depending on the relative importance of the effectiveness "position" of that job. So a job ranked in the second category from the effectiveness point of view will get half the score (out of its priority index) with respect to a job ranked in the first category.

5. Empirical test

In order to evaluate its effectiveness and computational feasibility, the RESDES approach

was tested through two different sets of real data. The first set relates to a medium Italian drown rod manufacturer, while the second one belongs to a large multinational semiconductor manufacturer. In particular, the second case study arguably represents the most interesting case for reentrant shops: therefore, since the major conclusions were noticeably similar between the two cases, for the sake of brevity, hereon we will concentrate on the semiconductor case only. To carefully present the empirical results, this chapter is arranged as follows: Section 5.1 provides a description of the analysed manufacturing context, Section 5.2 describes the selected response variables, while Section 5.3 illustrates the experimental framework and the data analysis procedure.

5.1. The test bed

The company that provided the test case for this study is a large multinational player in the semiconductor industry; production data were thoughtfully extracted from a wafer fab belonging to one of its main facilities, located in Italy. Confidentiality needs forced this company to provide us with outdated data (1998 onward), which are no longer valid in terms of part numbers, etc., but still suitable from a conceptual point of view. At that time, the fab produced more than 250 different types of finished products, with a total production volume of about 275.000 wafers/year. The production process has remained the same, characterised by very long routes (over 600 steps, production flow time of about 2 weeks) and by a large number of work-centres (about 300) divided into seven different technological areas: cleaning, oxidation, lithography, latent image building, ion implantation, resist removal and quality control. Batch work-centres concentrate on the ion implantation phase. Handling activities among different areas and within the same area are not automated, but the rigid layout allowed us to model them as deterministic steps in the route, with a fixed time.

The planning of the logistic and production flow is centralised, and done on a monthly basis with a 3-month horizon; then each facility develops a weekly plan which, on a finite-capacity basis, provides the guidelines for the logistic activities. Finally, a daily schedule is developed, whose concern is the single handling unit (HU), each containing 25 wafers to be processed. The rules adopted by the company were: Earliest Due Date to decide which job to release within the pre-shop pool, FIFO for dispatching, and a Smallest Queue rule for routing. This rule combination has also been selected as the benchmark set to be compared with the RESDES approach.

5.2. Response variables

Coming to the response variables, we managed to select a proper set of scheduling performance measures. These were divided into two sub-categories, namely *General Performance Measures and Machine Specific Performances Measures*. General Performance Measures were tracked in order to highlight the overall performances of the RESDES approach. They were further divided into *shop oriented* performances (mean production flow time [hours], flow time standard deviation [hours] and cumulated throughput [picosecond] over the 280 days period), and *job oriented performances* (% of late jobs, job tardiness [hours]). Machine Specific Performance Measures were tracked to discover benefits or counter-indications of the RESDES approach on a restricted subset of the production resources. They were: average time utilisation [%], time and space average utilisation [%] (for batch machines only), mean queuing time and its standard deviation [hours], average queue length [job], and setup number over the 280 days period (for sequence dependent setup machines only).

Machine specific performance measures were recorded till the final simulated day, while general performance measures were recorded for every job completed during the 280 day period. The censored data distortion was judged negligible given the simulated horizon and the average production flow time. For the sake of conciseness not all of the above measures will be discussed in the remaining of the paper.

5.3. Experimental framework and data analysis procedure

The whole experimental campaign was subdivided into two distinctive parts: the first one aimed at tuning the Request Importance (RI) parameters (cf. Section 4.2), and the second one aimed at defining which criterion could be revised as the best one to combine the job and the shop components of the BAL rule (cf. Section 4.4).

With regard to the first campaign, for each of the three different request types (cf. Section 4.2), three different RI values were tested. Relative weight was scaled with a 6× factor from one to another: in this way, the less important request weighted 1/36 of the most important one. Within this campaign, the recorded performances were not compared with the benchmark scheduling approach, since the target of this first campaign is just to tune at best the RI parameters.

The second campaign, then, inherited the best parameters' setting highlighted by the first campaign and tested the three combining alternatives described in Section 4.4 for the job and the shop components of the BAL rule: this was done with the same simulated time window, and then the recorded performances were compared against the benchmarking scheduling approach (cf. Section 5.1).

The test bed was implemented on a Workstation HP Kayak, while Arena and C++ software environments were used respectively to model the manufacturing environment and to code the RESDES architecture. This segmented analysis has been decided in order to bound the test time, because of the very detailed simulation framework (every single machine, every single Handling Unit), and of the large number of resources and entities to be treated.

For the sake of convenience, the simulated time window was set to one production year, made of 280 working days, 24 hours each. The number of replications was a though problem to solve. Actually, the real decision variable was the product of the simulation time window times the number of simulated years (i.e. replications). Since we wanted to perform the whole experiment by resorting to real data, without artificially generating production orders to feed the test bed, we had only a 3-year production orders' database, from 1998 to 2000, to resort to. Hence, a maximum of three replications was possible. To figure out if this was sufficient to collect enough data to perform statistical test, we followed Law and Kelton's (2000) procedure: we selected an average parameters' setup, and plotted the 95% confidence interval width of each response variable, looking for the confidence interval to converge. This analysis was performed in a twofold way:

(i) Response variables which are repeatedly collected within the simulation time window (e.g. production flow time). In this case there were no problems with Law and Kelton's procedure, since even 280 days were enough to collect enough samples for this class of variables.
(ii) Response variables which are collected only once during the simulation time window (e.g. cumulated throughput). In this case, because of the limited data available, we did not analyse the yearly variables, but the monthly ones. In this way, we found that three replications (i.e. 3 years times 12 months) were enough to observe a convergence in the 95% confidence interval. Then, for the sake of conciseness, we assumed that the conclusions related to monthly variables were valid also for the yearly variables, which are more communicative, and easy to discuss.

The same approach was held during the second campaign.

Two context parameters have also been studied: the capacity critical machines availability and the WIP level. The first parameter has been chosen because the production resources breakdown is quite a sensitive issue in the semiconductor business: therefore we tested two different levels for the availability parameter, namely 96% and 98%, measured as the standard ratio between MTTF and MTTF+MDT. With regard to the second parameter, namely the WIP level, a special remark is needed in order to explain how appropriate WIP testing levels were identified. As mentioned above, the job release timing is regulated by a Constant

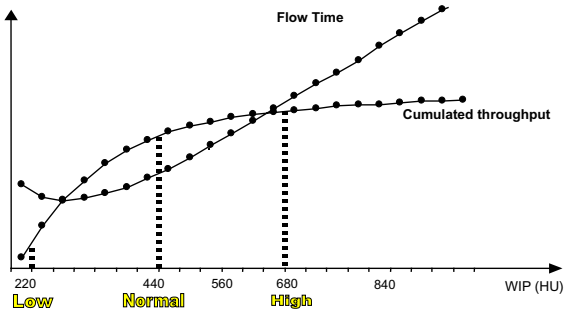

Fig. 5. Tested WIP levels for the semiconductor manufacturer.

WIP criterion. So the aim was to test the system under three quite different load conditions, respectively associated to a low, normal and high system utilisation. To do that, we plotted the WIP-throughput curve of the analysed production system under the benchmark scheduling approach; the output plot is shown in Fig. 5.

Given that behaviour, three corner points were qualitatively selected, as representative of the desired workload conditions: therefore appropriate WIP levels to test were found to be 220, 440 and 680 HUs.

During this pre-processing phase we also set the threshold to distinguish between capacity critical and non-capacity critical work-centres (cf. Section 4.3). This was done by analysing the average utilisation of any work-centre, and by identifying those ones which, according to the company's employees, represented the actual capacity constraints of the system. The corresponding average capacity utilisation was higher than 75%, and therefore that threshold was chosen to point out capacity critical machines.

No other context parameters were studied: for instance, available data suggested that the human-related uncertainty was not relevant in this highly-automated process; moreover, the ion implantation stage was the only one to be affected by non-negligible scrap rates, but with very stable values. Therefore we modelled a scrap rate, but it was not treated as a simulation parameter.

Special attention was paid to the management of the transient periods. The WIP level was set to zero every time the simulation was run: then an heuristic procedure was used to gradually release jobs in the system, by downloading production orders' information from the available database. Once the desired WIP level (220, 440 or 680 HUs) was reached, we started to record tally variables. The simulations were stopped as soon as the 280th day past the first recorded day has been reached.

During the first campaign, a one-way ANOVA was performed to highlight the significant factors, with p value threshold set at 0.05. ANOVA's hypotheses (i.e. independence of residuals, normality of residuals and residuals' equal variance) were validated though the standard plots of the Minitab software.

When a factor was revealed to be significant, a Tukey's test was performed in order to evaluate the significance of the difference in means observed when varying the levels of that factor. ANOVA was also used during the second campaign to evaluate the significance of the observed difference depending on the combination criterion of the job and the shop components.

As we signed a non-disclosure agreement with the company which provided the test data, only a limited amount of information on experimental data can be made available on request.

6. Experimental results

First of all, we would like to make a remark concerning the selected benchmark approach (cf. Section 5.1). In fact, during the last decade it has been observed that FIFO is not a good policy to use when scheduling reentrant lines (see for instance Lu et al., 1994; Seidman, 1994). In our opinion, the utilisation of the FIFO benchmark was suitable for the objective of this paper (cf. Section 3), i.e. the assessment of the feasibility of a complex framework including all the innovative features highlighted in Section 2. It is indisputable that the results below concern a preliminary study on achievable performance improvement, and would take advantage of further research refinement.

6.1. First campaign results

The first campaign aimed at defining the best combination of the RI parameters, neglecting the

job oriented component of the BAL rule. Therefore this first campaign analyses different sets of the RI parameters, without comparing the results with the current scheduling approach adopted by the manufacturer. The significance of each factor in terms of performance impact was proven according to the tests described in Section 5.3.

Then we selected the worst performing combination, the best performing one and "the average one", which is not an actual combination, but simply the performance level obtained by averaging all the 81 runs. The average combination was set to 100, and the remaining two were scaled and compared with it. For the sake of brevity, Fig. 6 illustrates only Cumulated throughput and mean flow time data: in this case, bottleneck machines availability parameter was set to high, while the WIP level is free to move.

The depicted behaviour highlights a very favourable characteristic, i.e. the RESDES model robustness. In fact, if we consider for instance the normal WIP level graph, we see that Cumulated throughput may vary from 92 to 105, therefore over the 85% of the maximum performance level can be gained without any tuning effort. Same consideration is true for the other performance measure.

Therefore we can judge the RESDES approach quite a robust one, since a good performance result is achieved by simply adopting an "average" parameter setting, without the need for an expensive and time consuming parameters tuning phase. The same convincing results were found by varying the capacity critical work-centres availability parameter, with a maximum distance between the worst and the best performance combination smaller than 10% of the average result. Anyway, a more detailed analysis showed that by increasing the relevance of the expediting requests from medium to high the performance of batch and sequence dependent setup sequential work-centres were slightly disrupted. The remaining two parameters (Normal requests and Inhibition requests) showed a linear trend for almost every measured performance. Therefore the best parameter combination setting emerging from this first campaign was as follows: expedition requests = medium, Normal and Inhibition requests = high; this parameter setting, was used in the following campaigns.

6.2. Second campaign results

The second campaign aimed at defining which of the job-shop BAL rule combining criteria is the best performing one (cf. Section 4.4). While performing all the tests, the pursued effectiveness objective was the minimisation of the mean tardiness, so the four categories described in Section 4.4. were sorted according to the ranking in the right column of Table 1. Data reported in Fig. 7 represents the average performance improvement/variation (over the three replications) of the RESDES model vs. the benchmark scheduling approach. The behaviour shown in Fig. 7 is quite heterogeneous, and allows for interesting comments.

First, there is a recognisable trade off between the shop oriented performance (measured through the cumulated throughput) and the job oriented

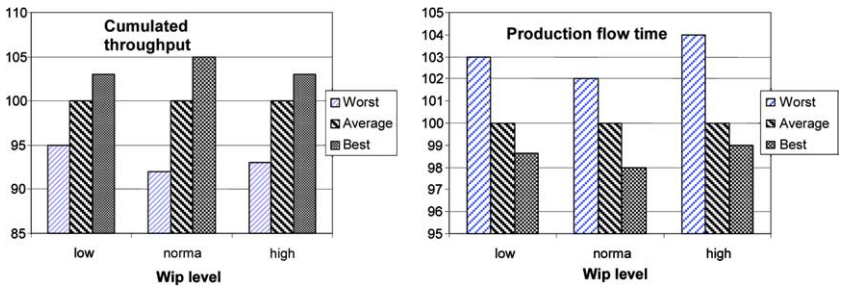

Fig. 6. General outcomes by levels of WIP.

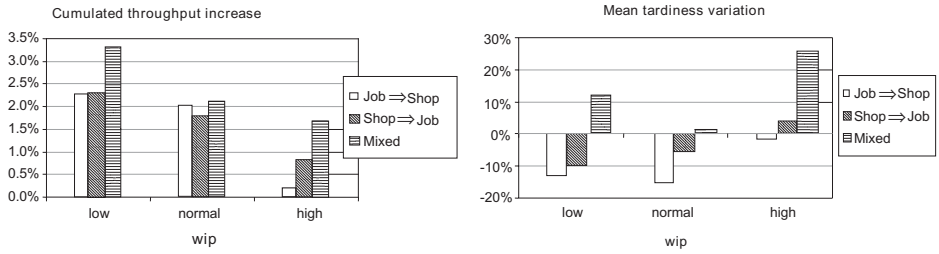

Fig. 7. General outcomes by levels of WIP (vs benchmark approach).

performance (measured through the mean tardiness). In fact, the Mixed criterion outperforms the others in terms of throughput increase, while the Job ⇒ Shop criterion performs best in terms of mean tardiness reduction.

Second, even if each balancing criterion does its best by a single set of performance, the RESDES approach is often able to improve both effectiveness and efficiency performance if compared to the currently adopted scheduling system: this is especially true for low and normal WIP levels, and with the Job ⇒ Shop balancing approach.

In fact, as long as the WIP remains on a normal level, which is the most common situation, the Job ⇒ Shop balancing criterion performs well, with a throughput increase of about 2% and a tardiness reduction of more than 10%. For higher WIP levels, the trade-off between job oriented and shop oriented performances becomes stronger, and the RESDES approach provides no useful way to improve both performances simultaneously.

Other encouraging remarks can be drawn from the analysis of the Machine specific performance, which are not illustrated because of space constraints: for instance, under a Job ⇒ Shop balancing criterion, batch machines net utilisation increased of about 2%, with a remarkable queuing time reduction of about 7%: this result can be connected to the requests issued by such machines, that enable a more balanced and performing queue arrangement. The same encouraging remarks stay true for sequence dependent, capacity critical work-centres, whose time utilisation increases of about 8% due to the improved job sequencing assured by the Setup Sharing rule.

7. Concluding remarks

The objective pursued from the very beginning of this research was to develop a new method able to comply with all the most innovative trends in reentrant shops scheduling highlighted in Section 2. More in detail, we rather tried to understand in general terms whether such an all-encompassing approach could be developed and to which extent it could offer interesting performances. Consistently, a new method named RESDES was developed and presented in Section 4, with the following main characteristics.

First, it introduces a classification of work-centres based on their flow and capacity criticality: following this classification, work-centres are subdivided in six homogeneous classes, with different rights to push decisions or to be pulled by other's decisions, depending on the ability of each work-centre to disrupt global performances. Then each work-centre is ensured an individual objective and a decentralised scheduling responsibility, thus fully implementing the concept of decentralised decision making.

Second, it assigns to each class of work-centres a different scheduling rule, tailored on that class's requirements (in terms of managerial objectives and technological requirements), thus embedding the concept of space distribution of the scheduling rules adopted. Moreover, since two of the six classes are characterised by the fact that their scheduling rule can change in time depending on the work-centre's congestion, also the concept of time distribution of the scheduling rule is coherently adopted.

Moreover, while some of the customised and dynamic scheduling rules adopted are taken from

literature (e.g. WNLT rule), others are original methods developed on purpose (Setup Sharing, BAL). In particular the WNLT rule implements the concepts of time and space depth of the information used to make decisions, and this represents a third relevant feature of the new RESDES method. Similarly, the BAL rule implements a multi-objective approach, in that decisions are made on the basis of a combination of aspects relating both to the job oriented (effectiveness) and shop oriented (efficiency) domains, and this represents a fourth relevant feature of the RESDES approach.

Two relevant case studies were carried out by means of discrete events simulation, in order to collect empirical evidence regarding the performance, robustness and usability of the method. The outcomes collected by now definitely encourage this research path. In particular, the matching simulation results help us to support the following three conclusions. First, despite the fact that some parameters have to be tuned, they don't seem to have a wide impact on the actual results achieved, since the average performance yields results that are much closer to the best combination, than to the worst one. This means that by adopting whichever parameters setting, the expected results are rather close to those that can be achieved through the optimal configuration: in other words, the tuning phase does not seem to have much importance. A second important outcome of the empirical test performed, is that the new method outperforms the considered benchmarks (consisting of rather simple and static scheduling rules) in almost any WIP configuration and jointly in both effectiveness and efficiency performances. This evidence is very important, especially for the semiconductors business (heavily affected by reentrant shops), where the very high level of fixed investment requires to improve the machines' utilisation, but this cannot be achieved at the expenses of running capital efficiency or (worse) customer service. Thus, the new RESDES method presented in this paper can be seen as a competitive tool able to support efficiency-effectiveness trade-off switching. Finally, the last consideration supported by the empirical test regards the information and decisional infrastructure. Despite the large dimension of the manufacturing system that was simulated and the rather large amount of data that have to be managed and continuously accessed by each work-centre, the whole simulation was run in a considerably short time, even with a simple hardware platform. We can therefore conclude that RESDES is less demanding for an information infrastructure than its articulate framework could allow to believe: this consideration seems quite promising for a real time implementation of the RST described in Section 4.1.

Given the specific objectives pursued by this paper, and the empirical outcomes highlighted here above, a clear indication is achieved in favour of the pursuit of this research path. More in detail, the next steps in this research will be the following ones. Since RESDES is mainly based on heuristics, in several aspects it might be questioned whether the best choices were made or not in designing this or that aspect of the approach; therefore a more thorough configuration of the RESDES will be sought, especially regarding the various aspects that were treated by means of common sense. Second, a more complete parameters setting methodology can be implemented, for instance by recurring to a full factorial simulation campaign that will encompass all the parameters at once. Moreover, once the suitability of the RESDES framework has been proven to be valid, different rules could be tested for specific machine, so to further increase the effectiveness of the model.

Finally, a more demanding benchmark can be searched in literature and implemented, in order to check that RESDES not only can outperform the practical methods implemented in industry, but also the most advanced methods proposed by other researchers of this field.

References

Baek, D.H., Yoon, W.C., Park, S.C., 1998. A spatial adaptation procedure for reliable production control in a wafer fabrication system. International Journal of Production Research 36 (6), 1475–1998.

Backer, K.R., 1974. Introduction to Sequencing & Scheduling. John Wiley, New York.

Barman, S., 1997. Simple priority rule combination: An approach to improve both flow time and tardiness. Inter-

national Journal of Production Research 35 (10), 2857–2870.

Bergamaschi, D., Cigolini, R., Perona, M., Portioli, A., 1997. Order review and release strategies in a job shop environment: A review and a classification. International Journal of Production Research 35 (2), 399–420.

Bechte, W., 1988. Theory and practice of load-oriented manufacturing control. International Journal of Production Research 26 (3), 358–364.

Blackstone, J.H., Phillips, D.T., Hogg, G.L., 1982. A state-of-the-art survey of dispatching rules for manufacturing job shop operations. International Journal of Operations Research 20, 27–45.

Bongaerts, L., Valckenaers, P., Van Brussel, H., Peeters, P., 1997. Scheduling execution in Holonic Manufacturing Systems. In: Proceedings of the 29th CIRP Int. Seminar on Manufacturing Systems, pp. 209–214.

Chandra, P., Gupta, S., 1997. Managing batch processors to reduce lead time in a semiconductor packaging line. International Journal of Production Research 35 (3), 611–633.

Cigolini, R., Perona, M., Portioli, A., Turco, F., 1996. Comparison of different dispatching rules in VLSI semiconductor manufacturing: A simulation approach. In: 8th European Simulation Symposium, October 24–26, Genoa, Italy.

Cigolini, R., Perona, M., Portioli, A., 1998. Comparison of order review and release techniques in a dynamic and uncertain job shop environment. International Journal of Production Research 36 (11), 2931–2951.

Cigolini, R., Comi, A., Micheletti, A., Perona, M., 1999. Implementing new dispatching rules at SGS-Thomson Microelectronics. Production Planning and Control 10 (1), 97–106.

Cigolini, R., Perona, M., Portioli, A., Zambelli, T., 2002. A new dynamic look-ahead scheduling policy for batch processors. Journal of Scheduling 5, 185–204.

Connors, D., Feign, G., Yao, D., 1994. Scheduling semiconductors lines using a fluid network model. IEEE Transactions on Robotic and Automation 10 (2), 88–98.

Dayhoff, J.E., Atherton, R.W., 1986. Signature analysis of dispatch schemes in wafer fabrication. IEEE Transactions on Computers, Hybrids and Manufacturing Technology 9 (4), 518–525.

Decker, K.S., 1987. Distributed problem solving techniques: A survey, IEEE Transactions on Systems, Man and Cybernetics 17 (7), 729–740.

De Souza, R., Lin, W., 1997. An integrated rapid modelling environment for managing wafer fabrication facilities. Journal of Electronic Manufacturing 7 (3), 199–209.

Fowler, J.W., Hogg, G.L., Phillips, D.T., 1992. Control of multi-product bulk service diffusion/oxidation processes. IIE Transactions 24 (4), 84–96.

Garetti, M., Lanza, C., Pozzetti, A., 1989. Scheduling techniques for integrated manufacturing (in Italian), Pixel, N. 4 and 5.

Glassey, C.R., Shanthikumar, J.G., Seshadri, S., 1996. Linear control rules for production control of semiconductor fabs. IEEE Transactions on Semiconductor Manufacturing 9 (4), 536–549.

Holthaus, O., Rajendran, C., 1997. Efficient dispatching rules for scheduling in a job shop. International Journal of Production Economics 48, 87–105.

Holthaus, O., Ziegler, H., 1997. Improving job ship performance by co-ordinating dispatching rules. International Journal of Production Research 35 (2), 539–549.

Hwang, H., Sun, J.U., 1998. Production sequencing problem with reentrant work flows and sequence dependent setup times. International Journal of Production Research 36 (9), 2435–2450.

Johri, P.K., 1993. Practical issues in scheduling and dispatching in semiconductor wafer fabrication. Journal of Manufacturing Systems 12 (6), 474–485.

Jones, A., Rabelo, L.C., 1998. Survey of Job Shop Scheduling Techniques, NISTIR (National Institute of Standards and Technology), Gaithersburg, MD.

Kim, Y.D., Kim, J.U., Lim, S.K., Jun, H.B., 1998a. Due date based scheduling and control policies in a multi-product semiconductor wafer fabrication facility. IEEE Transactions on Semiconductor Manufacturing 11 (1), 155–164.

Kim, Y.D., Lee, D., Kim, J.U., 1998b. A simulation study on lot release control, mask scheduling and batch scheduling in semiconductor wafer fabrication facility. Journal of Manufacturing Systems 17 (2), 107–117.

Kouikoglou, V.S., Phillips, Y.A., 1997. A continuous flow model for production networks with finite buffers, unreliable machine and multiple products. International Journal of Production Research 35 (2), 381–387.

Kumar, P.R., 1994. Scheduling semiconductor manufacturing plants. IEEE Control Systems (December), 33–40.

Kumar, S., Kumar, P.R., 1994. Performance bound for queuing networks and scheduling policies. IEEE Transactions on Automatic Control 39 (8), 1600–1611.

Kutanoglu, E., Sabuncuoglu, I., 1999. An analysis of heuristics in a dynamic job shop with weighted tardiness objectives. International Journal of Production Research 37 (1), 165–187.

Law, A.M., Kelton, W.D., 2000. Simulation Modeling and Analysis, third ed. McGraw-Hill, New York.

Lee, C.Y., Piramuthu, S., Tsai, Y.K., 1997. Job shop scheduling with a genetic algorithm and machine learning. International Journal of Production Research 35 (4), 1171–1191.

Li, S., Tang, T., Collins, W., 1996. Minimum inventory variability schedule with application in semiconductor fabrication. IEEE Transaction on Semiconductor Manufacturing 9 (1), 145–159.

Lu, S.C.H., Ramaswamy, D., Kumar, P.R., 1994. Efficient scheduling policies to reduce mean and variance of cycle time in semiconductor manufacturing plants. IEEE Transactions on Semiconductor Manufacturing 7 (3), 374–388.

Mehta, S., Uzsoy, R., 1998. Minimising total tardiness on a batch processing machine with incompatible job families. IIE Transactions 30 (2), 165–178.

Miller, D.J., 1990. Simulation of a semiconductor manufacturing line. Communications of the ACM 33 (10), 98–108.

Nakata, T., Matsui, K., Miyake, Y., Nishioka, K., 1999. Dynamic bottleneck control in wide variety production factory. IEEE Transactions on Semiconductor Manufacturing 12 (3), 273–280.

Pierreval, H., Mebarki, N., 1997. Dynamic selection of dispatching rules for manufacturing system scheduling. International Journal of Production Research 35 (6), 1575–1591.

Roy, D., Anciaux, D., 2001. Shop-floor control: A multi-agents approach. International Journal of Computer Integrated Manufacturing 14 (6), 535–544.

Robinson, J.K., Fowler, J.W., Bard, J.F., 1995. The use of upstream and downstream information in scheduling semiconductor batch operations. International Journal of Operations Research 33 (7), 1849–1869.

Seidman, T.I., 1994. 'First come, first served' can be unstable! IEEE Transactions on Automatic Control 39, 2166–2171.

Spearman, M.L., Woodruff, D.L., Hopp, W.J., 1990. CONWIP: A pull alternative to kanban. International Journal of Production Research 28, 879–894.

Uzsoy, R., Lee, C., Martin Vega, L., 1992. A review of production planning and scheduling models in the semiconductor industry, part 1: System characteristics, performance evaluation and production planning. IIE Transaction 24, 47–60.

Weng, W.W., Leachmann, R.C., 1993. An improved methodology for real time production decision at batch-process workstations. IEEE Transactions on Semiconductor Manufacturing 6 (3), 219–225.

Yi-Feng, H., 1998. Scheduling of mackshop E-beam writers. IEEE Transactions on Semiconductor Manufacturing 11 (1), 165–172.

Zhou, M., Jeng, M.G., 1998. Modelling, analysis, simulation, scheduling and control of semiconductor manufacturing systems: A Petri net approach. IEEE Transactions on Semiconductor Manufacturing 11 (3), 333–357.