Extended Classification for Flowshops with Resequencing

Gerrit Färber *Anna M. Coves Moreno*

Instituto de Organización y Control de Sistemas Industriales (IOC)
Universidad Politécnica de Cataluña (UPC), Barcelona, Spain
gerrit_faerber@gmx.de, anna.maria.coves@upc.es

Resumen

Este trabajo presenta una clasificación extendida de líneas de flujo no-permutación. Se consideran las múltiples opciones que se presentan al incluir la posibilidad de resecuenciar piezas en la línea. Se ha visto que en la literatura actual no se ha clasificado con profundidad este tipo de producción.

Abstract

This paper presents an extended classification for non-permutation flowshops. The versatile options which occur with the possibility of resequencing jobs within the line are considered. The literature review shows that no classification exists which considers extensively this type of flowshop.

Keywords: Classification, Non-permutation flowshop, Resequencing, Mixed model assembly line

1. Introduction

In the classical flowshop M stations are arranged in series, according to the technological sequence of the operations and a set of N jobs has to be processed on these stations, see e.g. [15]. Each of the N jobs has the same ordering of stations for its processing sequence. Each job can be processed on one and only one station at a time, each job is processed only once on each station, and each station can process only one job at a time. Jobs may bypass other jobs only between stations.

The above definition gives a general frame for classifying non-permutation flowshops. Due to the numerous variations which occur in real production lines, a more extended classification is required. There exist various classifications and classifying surveys, as for example Pinedo [21], Vieira et al. [27] Herrmann et al. [13], and Lageweg et al. [17], Plans and Corominas [22], Niu [20], Becker and Scholl [5], none of which considers exhaustively the possibilities of resequencing jobs within the production line, such as: using large buffers (Automatic-Storage-and-Retrieval-System) which decouple one part of the line

[1] This work is partially supported by the Ministry of Science and Technology, Spain, and the funding for regional research DPI2004-03472

from the rest of the line [19]; buffers which are located off-line [18]; hybrid or flexible lines [10]; and more seldom, the interchange of job attributes instead of physically changing the position of a job within the sequence [23].

2. Classification

The classification of non-permutation flowshops, proposed in this paper, is based on the classification scheme by Pinedo, [21], and is intended to adequately classify flowshop problems with particular consideration of options and restrictions which appear when resequencing of jobs takes place. The notation contains the triplet $\alpha|\beta|\gamma$ and helps classifying sequencing and scheduling problems. The triplet determines a problem as: α describes the station environment; β provides details on characteristics and constraints for the processing of the jobs; γ contains information on the objectives of the optimization.

Even though references on specific characteristics are given, this paper does not pretend to provide an exhaustive survey on solution techniques. It is intended to generate an instrument for properly categorizing the diversity of flowshop problems, in order to simplify their comparison and to improve the possibilities to finding new configurations that are not yet investigated and may lead to further optimization.

2.1. General notation

The determination of flowshop problem requires the definition of a finite number of stations, being i the index and M the number of stations, and a finite number of jobs, being j the index and N the number of jobs. The processing of a job is described by:

Start time (s_j): The time job j starts being processed on the first station is called start time.

Processing time (P_{ij}): The processing time, also called assembly-time, is the time that job j maintains at station i while being processed. Due to the nature of the flowshop, job j that is not processed at station i has to pass this station with a processing time equal to zero.

Completion time (c_j): The time job j completes processing on the last station and exits the system is called completion time.

2.2. Station environment (α)

The station environment provides relevant information on the characteristics which are related to the stations, more specific, with respect to the layout of the production line. The considered categories include the way in which the stations are arranged (station arrangement), the way in which the stations are operated (operating properties) and finally, the way in which jobs may be resequenced (resequencing facilities).

2.2.1. Station arrangement

The way in which the stations are arranged significantly influences the layout of the production line and furthermore determines if resequencing of jobs is possible or not.

Flowshop (F_M): In the classical flowshop M stations are arranged in series. All jobs have the same station routing, being the primary difference to the Openshop and the Jobshop. In the classical flowshop, the *Non-permutation flowshop*, the job sequences Π_i can vary from one station to the next.

> *Single station (F_1)*: The most simple case of a production line is the one which provides only a single station. Here the jobs require to be operated on only one station. The single station case clearly can be considered a permutation flowshop.

Permutation flowshop (*perm*): The solutions are restricted to job sequences $\Pi_1, ..., \Pi_n$ with $\Pi_1 = \Pi_2 = ... = \Pi_n$, that is, the sequence on the first station is maintained for all stations in the flowshop. A set of permutation sequences is denoted dominant if no better sequence can be found than the best permutation sequence, occuring for example in the no-wait flowshop.

Flexible flowshop (*FFS*): In this special case of flowshop parallel stations exist which perform operations in parallel. The main reason for installing parallel stations is the reduction of the cycle time at a station. Furthermore, due to the fact that the processing time of a station is dependent on the model type of the job, exist the possibility of one job overtaking its predecessor without taking it off the line. The flexible flowshop is also called *hybrid* or *compound flowshop* and, depending on the degree of similarity of the parallel stations, three different types can be considered:

> *Identical parallel station*: The parallel stations are identical and, if not defined differently in the β field, the jobs may be processed by any of the parallel stations which result in a reduction of the cycle time at a station.
>
> *Parallel stations with different speed* (*Qm*): The parallel stations have different processing times, caused by, e.g., varying operator skills or a difference in the available tools. For this reason it may be favorable for a certain job to pass through a determined parallel station.
>
> *Unrelated parallel stations* (*Rm*): The case in which the parallel stations are unrelated occurs when, e.g., tools or operator skills are provided only at certain parallel stations.

Flowline-based manufacturing system (*FBMS*): The line is similar to the classical flowshop, apart from the fact that some jobs have missing operations at some stations and can bypass the particular station. This leads to a characteristic which is not proper of the flowshop, i.e. the station precedence is not the same for all jobs. Hence, an optimum sequence can be obtained that would not feasible if the processing time would be infinitesimally small instead of zero.

Intermittent buffer (*IntBuf*): In order for an intermittent buffer to permit resequencing, it can not be operated in *FIFO* (first-in-first-out) mode.

Offline buffer ($OffBuf$): An offline buffer is located off the production line in order to let pass other jobs. An additional handling time HT_{OffBuf} may occur which is necessary to transfer a job to and fro the offline buffer. Further distinction is done as follows:

> *First In First Out (FIFO)*: For the case of more than one offline buffer place, this buffer can be operated in FIFO-mode. On the one hand this opens the way for simplification of the mechanics and logistics of the offline buffer, but on the other hand restricts the solutions considerably.
>
> *Automated Storage and Retrieval System (ASRS)*: A multitude of buffer places are provided between two main parts of the production line. The reason for using large buffers in this case is to resequence the jobs in a large scale. As a result, batches are formed and each shop is optimized separately. Only in the case in which the optimal sequence for one shop is the same as for the following, no resequencing is performed. This type of buffer system is, e.g., studied by [19], [14], or [9].
>
> *Intermittent/centralized location (int/centr)*: The access to the offline buffer can be limited to only one station (intermittent case) or to various stations (centralized case). A production line, arranged in U-shape, is especially suitable for the use with a centralized offline buffer.
>
> *Physical size limitation (phsize)*: The physical size of the individual buffer places is limited, which leads to the restriction that not every job can pass through a certain buffer place. In the case of a chemical production, a buffer place represents a tank and the physical size limitation is the provided volume. Instead of two large tanks, one large and one medium sized tank may be sufficient, which in sum on the other hand results in less investment and a reduced area occupation.

Splitting and merging of parallel lines (*merge/split*): The introduction of parallel segments of stations permits to resequence jobs where the line splits or merges. The splitting of a production line is somewhat more challenging due to the fact that two parallel lines may not perform the same options and constraints exist that may additionally influence the sorting. [10], reports the case of two parallel lines, used by the automobile manufacturer Volvo.

2.2.2. Operating properties

The operating properties describe the way in which a station is operated and give details on its restrictions, as for example to prohibit blocking of stations.

Paced/unpaced line (*PL/UPL*): In a paced production line the mechanical material handling equipment, like conveyor belts, couple the stations in an inflexible manner. The jobs are either steadily moved from station to station at constant speed or they are intermittently transferred after processing. The available amount of time for the operation is the same in both cases. In the unpaced line, in contrast, the stations are decoupled by buffers. In a specific case these buffers store jobs that can not be passed to the downstream station which is still occupied with processing the previous job.

Blocking (*block*): Blocking can occur in a flowshop when between two succeeding stations only a limited number of buffer places is provided. When all buffer places are occupied, the upstream station can not be unloaded and is blocked from further processing. The flag *block* is used to indicate that only schedules are feasible that do not result in blocking.

Zero-buffer (*ZeroBuffer*): This variation of the classical flowshop does not allow the jobs to form queues between the stations. A job j leaving station i cannot advance to station $i + 1$ if there is still a job being processed. Station blocking of station i is the result.

No-idle (*noidle*): This constraint implies that each station, once started with processing, has to process all operations without interruption. As mentioned by [8], a real life situation can be found, for example, if machines represent expensive pieces of equipment which have to be rented for the duration between the start of its first operation and the completion of its last operation.

Station breakdown (*breakdown*): Station breakdown describes the state of a station which does not permit processing of any job due to failure. In real production systems the breakdowns occur in a stochastic way and can be simulated using the values Mean-Time-Between-Failure (MTBF) and Mean-Time-To-Repair (MTTR).

Station maintenance (*maintenance*): Station maintenance describes the state of a station which does not permit processing of any job due to prevention. In contrast to breakdown, the maintenance occurs in a deterministic way and usually with regular cycles, demanded by the tool manufacturer.

2.3. Job processing environment (β)

The Job processing environment provides relevant details on characteristics and constraints for the processing of the jobs. The considered categories include the demand, time and cost constraints and processing restrictions.

2.3.1. Demand

The production planning for mixed model flowshops can be various and usually depends on the planning horizon and in some cases on the possibility of decoupling the customer orders from the production planning.

Single model (D_{single}): The simplest case is the single model production where only one type of product is produced.

Multi model (D_{multi}): In the multi model production the products form lots of the same model, which are then produced in batches.

Mixed model (D_{mixed}): In the mixed model production the job sequence is not determined by batches and therefore allows an arbitrary order.

> *Minimal part set (D_{MPS})*: The most common representations for the mixed model case is the use of the minimal part set which is the least common multiple of the individual models for the entire demand.

Launch interval fixed/variable (D_{Lifix}/D_{Livar}): In a paced line, the time between two consecutive jobs entering the production line is called launch-interval or cycle time. The fixed launch interval results in a constant production rate (production quantity per unit of time). The variable launch interval gives more flexibility, resulting in better solutions.

Static/dynamic demand (D_{stat}/D_{dyn}): The static demand refers to the fact that the entire demand, necessary to produce in a time window, is produced in an accumulated lot, known beforehand. Whereas, the dynamic demand implies that the customer orders arrive continuously or at least are not completely determinable beforehand.

Priority (w_j): The priority of job j is determined by its weight w_j which defines its importance with respect to the other jobs.

2.3.2. Time and cost restrictions

Restrictions which are directly related to the processing of jobs can depend on an individual job, as in the case of the release time, or can furthermore depend on the previous job, as in the case of the setup time:

Release time (R_j): The earliest time at which job j can start its processing.

Due time (D_j): The due time is either the latest possible time job j may leave the production line, or it is the time at which the job should be finished.

Setup cost (SC_{efi}): A setup cost is concerned if an additional cost appears to change the setup of station i, in order to be able to process job $j+1$ which is of model f after job j which is of model e. If the setup cost is independent of the model, it can be simply added to the processing cost.

Setup time (ST_{efi}): In a similar way a setup time is concerned if an additional time appears to change the setup of station i, in order to be able to process job $j+1$ which is of model f after job j which is of model e. If the setup time is independent of the model, it can be simply added to the processing time.

Handling-time (HT): The way in which the products are passed from one station to the next can be classified by its degree of automation. In contrast, this option is referred to the existence of a handling time HT which occurs when passing a job from one station to the next or between a station and a buffer.

Deterministic/Stochastic ($det/stoch$): The processing times, also including, e.g., setup times, are generally regarded to be deterministic. The more automated the production is, the more likely it is that these deterministic values are met. In a realistic production, depending on human operators, tool accuracy, the punctuality of suppliers, and where machine breakdowns occur, it may be desirable to include stochastic uncertainty to processing times.

Learning of operator (*learning*): A human operator, who is new to the processes of a certain station may not perform his tasks with the same velocity as after some time when he starts to experience routine.

2.3.3. Processing restrictions

Precedence (*prec*):The production of jobs may be constrained by precedence which is that a Job may not start processing before a certain job started or even completed. Two more specific cases of job precedences are

> *Strict precedence ($prec_{strict}$)*: Between two jobs of the same model or two jobs that require the same option at a station, a minimum number of different jobs or a minimum time is necessary.
>
> *Quality implication ($prec_{QI}$)*: The quality implications may affect the performance of a station. For example the paint quality may temporarily decline when a change of color occurs.

Preemption (*prmp*): Preemption is referred to the case in which it is not necessary to finish the processing of a certain job at once. It is allowed to interrupt its processing in order to process another job and to continue with the interrupted job at a later point of time. When preemption is allowed, bypassing of jobs is the consequence.

No-wait (*nwt*): This case, described by [1], is more restrictive than the *Zero-buffer* case and comprehensively similar to the *No-idle* case, however, instead of ensuring a station not to stop processing, here it has to be ensured that the jobs are not left without being processed. Once a job begins its processing on station 1, it continues without delay to be processed on each of the m stations. Only sequences are feasible which do not result in blocking of any station.

Re-entrant (*reent*): The re-entrant, or recirculation flowshop, considers that a job runs twice through the same line, e.g., first to assemble the bottom side and then to assemble the top side of a printed-circuit-board.

Change of Job-attributes (*jobatt*): Instead of physically changing the job order, swapping of jobs appears by changing their attributes [23]. Rather than a resequencing facility, the change of job- attributes is a method which requires logistical implement in the production flow of the plant.

Station eligibility (SE_j): When parallel stations exist (FF), a station may not be able of processing a certain job. SE_j determines the set of stations which can process job j.

2.4. Objectives (γ)

The environment of the objectives of the optimization firstly provides information on the purpose of the optimization, and then on the objective function which is used in the optimization itself.

2.4.1. Purpose of the optimization

The optimization of a flowshop production line is basically divided into two phases, the design phase and the operation phase. In the first phase the tasks are assigned to the stations, subject to technological precedence relations, and is called the line balancing problem. Once the line is balanced and the design of the line is obtained, it is necessary to achieve a reasonable, if not optimum, order for the jobs to be processed consecutively, being the operating phase with sequencing and scheduling of the jobs.

Balancing (*Balancing*): The process of balancing the load results in the design of the production line, usually implying the minimization of the station number and the determination of a cycle time, obtained by calculating, e.g., an average of the task-times, necessary to assemble the various models. The balancing procedure in many cases results in the prevention of the occurrence of bottlenecks so that the final production line will not experience stoppage and unnecessary inventory will not accumulate. Studies on the production line balancing problem are numerous and may be object to various criteria like cost-oriented or profit-oriented approaches, as described in the survey of [5], and [25], furthermore, the survey of [11], explains different measures for balancing problem with respect to its complexity.

Sequencing/Scheduling (*Sequencing/Scheduling*): The sequencing problem determines an appropriate order for the jobs to be processed within, e.g., the shortest possible time called makespan, used by [3], [6] and [18]. Whereas solving the scheduling problem results in prioritizing the order of the jobs due to resource usage and due-date-limits of the jobs, [26] presents a survey of static scheduling problems. As highlighted by [4], the two problems are either intimately tied together or irrelevant to each other and many times are used interchangeably.

Rescheduling (*Reschedule*): A problems which is classified *Reschedule*, form part of the sequencing/scheduling phase but determine a type of problem which reacts in a more dynamic way to a production which is already up and running, and most of all, is confronted with unexpected job or station related incidents:

> *Station incidents* ($Reschedule_{Stat}$): Station breakdown, shortage of material, operator absenteeism, maintenance, etc.
>
> *Job incidents* ($Reschedule_{Job}$): Job cancelation, urgent job arrival, due time change, delay in arrival, job priority change, Rework or quality problems, over- or underestimation of process time, etc.

These unexpected incidents lead to rescheduling, even if the rescheduling results in the confirmation that the current production is not influenced, but may also trigger the following steps: Overtime, in-process subcontracting, process change or re-routing, station substitution, limitation of manpower, setup times, equipment release, etc.

Resequencing (*Resequ*): Resequencing is referred to the possibility of resequencing jobs within the production line. The triplet for the station environment, *alpha*, indicates the way in which resequencing if performed.

2.4.2. Objective function

Within the mixed-model flowshop a variety of objective functions are to be found, the most common being time and cost orientated objectives. As a basic principle of optimization, the considered solutions are part of a set of feasible solutions. In what follows, the most common objectives are listed, followed by objectives which are specific for resequencing.

Time orientated objectives:

Makespan ($c_{\max}$): One of the most common objective functions in sequencing is to minimize the maximum completion time necessary to process the entire demand, called makespan or total production time. The makespan optimization generally ensures high utilization of the production resources, early satisfaction of the customer demand and the reduction of in-process inventory by minimizing the total production run. This objective is used for a static demand in which the number of jobs is limited.

Makespan:

$$\max\{C_j|j = 1, ..., i\} \tag{1}$$

Maximum flow time ($F_{\max}$): The minimization of the maximum flow time leads to stable and even utilization of production resources, rapid turn-around of jobs and the minimization of in-process inventory. [12], mentions that in the case where all release dates are zero, $c_{\max}$ and F_{max} are identical. The weighted maximum flowtime includes a weight related to the jobs. As in the previous case, this objective is used for a static demand in which the number of jobs is limited.

Maximum flow time:

$$\max\{c_j - s_j|j = 1, ..., N\} \tag{2}$$

Weighted maximum flow time:

$$\max\{\omega_j(c_j - s_j)|j = 1, ..., N\} \tag{3}$$

Mean flow time ($\overline{F}$): The mean flow time leads similar results as the maximum flow time, but due to the fact that this objective tries to average the flow time for all jobs, this leads to even more stable and even utilization of the production resources. This objective can be used for a dynamic demand in which the jobs enter continuously. The weighted mean flowtime includes a weight related to the jobs.

Mean flow time:

$$\frac{1}{N}\sum_{j=1}^{N}(c_j - s_j) \tag{4}$$

Weighted mean flow time:

$$\frac{1}{N}\sum_{j=1}^{N}\omega_j(c_j - s_j) \tag{5}$$

Setup time (ST): In a mixed model production, setup time ST_{efi} may occur when at station i a job $j + 1$ of model type f follows job j of model type e. Minimizing total setup time, furthermore, tends to decrease the total flowtime.

Setup time:

$$\sum_{j=1}^{N} ST_{efi} \tag{6}$$

Idle time (I): Idle time I_{ij} at station i occurs when an operator is kept waiting for job j. This may be caused by a job that has not yet arrived, or because an auxiliary operator is still occupied with the job. When setup time occurs, that is separable from the processing time, the operator can benefit from this idle time in order to perform the necessary changes for the next job to be processed.

Idle time:

$$\sum_{i=1}^{M}\sum_{j=1}^{N} I_{ij} \tag{7}$$

Mean idle time:

$$\frac{1}{M}\sum_{i=1}^{M}\sum_{j=1}^{N} I_{ij} \tag{8}$$

Utility time (U): Utility time U_{ij} at station i occurs when an operator has to continue with job $j+1$ before finishing with job j. In this case an auxiliary operator finishes the job; the time the auxiliary operator requires is called utility time. As well as the idle time, here the mean is taken over the stations. The minimization of idle and utility time is, for example, applied by [24], varying the station length and using individual weights for idle and utility time.

Utility time:

$$\sum_{i=1}^{M}\sum_{j=1}^{N} U_{ij} \tag{9}$$

Mean utility time:

$$\frac{1}{M}\sum_{i=1}^{M}\sum_{j=1}^{N} U_{ij} \tag{10}$$

Deviation: In general for all of the above mentioned time oriented objectives it is possible to use the deviation, or the squared deviation, over stations or over jobs, in order to equalize the deviation and to avoid solutions that provide extreme values for single stations or jobs, see e.g. [7].

Cost orientated objectives:

Line length (Length): [16], study the problem of minimizing the overall length of the production line. The production line in study contains hybrid stations, being a mixture of open and closed stations.

Line length:

$$\sum_{i=1}^{M} L_i \tag{11}$$

Setup cost (SC): The occurrence of setup cost may lead to the objective of minimizing the total setup cost to keep the production costs small. Setup cost SC_{efi} may occur when at station i a job $j+1$ of model type f follows job j of model type e.

Setup cost:
$$\sum_{i=1}^{M} SC_{efi} \tag{12}$$

Combined objectives: In the literature, the use of individual objective functions, as mentioned above, as well as combinations can be found. As an example for combined objectives in sequencing problems [2], uses the bicriteria of makespan and mean flowtime, whereas [3], uses makespan and line length as bicriteria for their algorithm.

Resequencing: The major objective of resequencing in flowshops is further minimization of production costs, for example resulting in a higher utilization of the production resources. This is desirable even more when setup cost/time is involved or the processing times of the individual jobs diverge among one another. Apart from the fact that the introduction of resequencing possibilities give way to further minimization of, e.g., the makespan, there exist objectives that are directly related to resequencing.

Minimize number of jobs to be resequenced (Resequ$_{Jobs}$): Resequencing a job results in an additional effort. Therefore, if two sequences, resulting in a different number of jobs to be resequenced, provide the same value of the objective function, the one with fewer resequencing is to be preferred. Minimizing the number of jobs to be resequenced is used in combination with another objective.

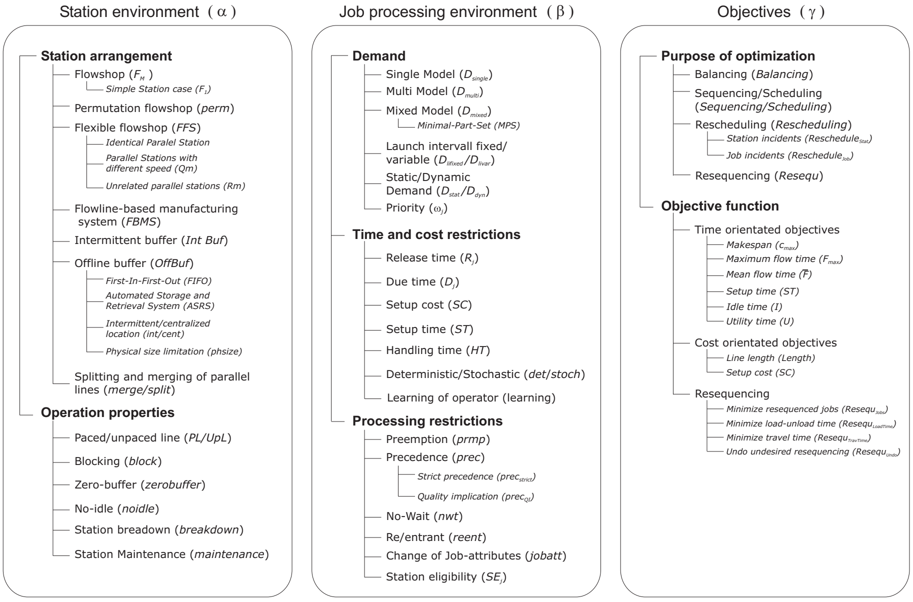
Minimize load-unload time (Resequ$_{LoadTime}$): When considering a handling time HT, which occurs for transferring a job to or from a buffer place, an objective may be to minimize the accumulated time caused by this handling. However, the calculation of the makespan already indirectly considers the handling time.

Travel time of jobs in ASRS-buffers (Resequ$_{TravelTime}$): When an ASRS-buffer (Automated Storage and Retrieval System) is used, large travel distances occur, which result in a notable travel time. Here, the objective is to minimize the total accumulated travel time.

Undo undesired resequencing (Resequ$_{Undo}$): Apart from the aim of further optimizing the makespan or reducing of setup cost or time, there exist objectives such as to undo undesired resequencing which is caused by rework, parallel inspection stations, unequal processing times on parallel stations or problems in part delivery.

3. Resuming table

The extended classification for flowshops, presented in this paper, considering that jobs can be resequenced within the production line. Table 1 shows the summary of the classification.



Station environment (α)

- **Station arrangement**
  - Flowshop (F_M)
    - *Simple Station case (F_1)*
  - Permutation flowshop (*perm*)
  - Flexible flowshop (*FFS*)
    - *Identical Paralel Station*
    - *Parallel Stations with different speed (Qm)*
    - *Unrelated parallel stations (Rm)*
  - Flowline-based manufacturing system (*FBMS*)
  - Intermittent buffer (*Int Buf*)
  - Offline buffer (*OffBuf*)
    - *First-In-First-Out (FIFO)*
    - *Automated Storage and Retrieval System (ASRS)*
    - *Intermittent/centralized location (int/cent)*
    - *Physical size limitation (phsize)*
  - Splitting and merging of parallel lines (*merge/split*)
- **Operation properties**
  - Paced/unpaced line (*PL/UpL*)
  - Blocking (*block*)
  - Zero-buffer (*zerobuffer*)
  - No-idle (*noidle*)
  - Station breadown (*breakdown*)
  - Station Maintenance (*maintenance*)

Job processing environment (β)

- **Demand**
  - Single Model (D_{single})
  - Multi Model (D_{multi})
  - Mixed Model (D_{mixed})
    - *Minimal-Part-Set (MPS)*
  - Launch intervall fixed/ variable ($D_{lifixed}/D_{livar}$)
  - Static/Dynamic Demand (D_{stat}/D_{dyn})
  - Priority (ω_j)
- **Time and cost restrictions**
  - Release time (R_j)
  - Due time (D_j)
  - Setup cost (*SC*)
  - Setup time (*ST*)
  - Handling time (*HT*)
  - Deterministic/Stochastic (*det/stoch*)
  - Learning of operator (learning)
- **Processing restrictions**
  - Preemption (*prmp*)
  - Precedence (*prec*)
    - *Strict precedence ($prec_{strict}$)*
    - *Quality implication ($prec_{QI}$)*
  - No-Wait (*nwt*)
  - Re/entrant (*reent*)
  - Change of Job-attributes (*jobatt*)
  - Station eligibility (SE_j)

Objectives (γ)

- **Purpose of optimization**
  - Balancing (*Balancing*)
  - Sequencing/Scheduling (*Sequencing/Scheduling*)
  - Rescheduling (*Rescheduling*)
    - *Station incidents ($Reschedule_{Stat}$)*
    - *Job incidents ($Reschedule_{Job}$)*
  - Resequencing (*Resequ*)
- **Objective function**
  - Time orientated objectives
    - *Makespan (c_{max})*
    - *Maximum flow time (F_{max})*
    - *Mean flow time ($\overline{F}$)*
    - *Setup time (ST)*
    - *Idle time (I)*
    - *Utility time (U)*
  - Cost orientated objectives
    - *Line length (Length)*
    - *Setup cost (SC)*
  - Resequencing
    - *Minimize resequenced jobs ($Resequ_{Jobs}$)*
    - *Minimize load-unload time ($Resequ_{LoadTime}$)*
    - *Minimize travel time ($Resequ_{TravTime}$)*
    - *Undo undesired resequencing ($Resequ_{Undo}$)*

Table 1: Summary of the extended classification for flowshops with resequencing.

4. Conclusions

The extended classification, presented in this paper, was indispensable, due to the lack of an adequate classification for flowshop production lines that would consider the diversity of arrangements which permit resequencing of jobs within the production line, such as: large ASRS-buffers which decouple one part of the line from the rest of the line; buffers which are located off-line; hybrid or flexible lines; and more seldom, the interchange of job attributes instead of physically changing the position of a job within the sequence

This classification is based on the notification used by Pinedo [21], but also establishes criteria that adequately categorizes flowshops that provide the possibility of resequencing, including a wide scope of resequencing facilities and objectives. As a matter of fact, the versatile facilities and methods for resequencing jobs within the production line, together with possible handling times, which occur when offline buffers are used, are taken into account. Furthermore, resequencing is included as an important purpose of the optimization, together with various objectives which are related to resequencing.

The triplet $\alpha|\beta|\gamma$ is used to determine a specific problem as: α describing the station environment; β providing details on characteristics and constraints for the processing of the jobs; γ containing information on the objectives of the optimization.

It was intended to generate an instrument for properly categorizing the diversity of flowshop problems, in order to simplify their comparison and to improve the possibilities to finding new configurations that are not yet investigated and may lead to further optimization.

5. References

[1] T. Aldowaisan and A. Allahverdi. Total flowtime in no-wait flowshops with separated setup times. *Computers & Operations Res.*, 25(9):757–765, 1998.

[2] A. Allahverdi. The two- and m-machine flowshop scheduling problems with bicriteria of makespan and mean flowtime. *European Journal of Operational Research*, 147(2):373–396, 2003.

[3] J.F. Bard, E.M. Dar-El, and A. Shtub. An analytic framework for sequencing mixed model assembly lines. *International Journal of Production Research*, 30(1):35–48, 1992.

[4] S.J. Beaty. Genetic algorithms for instruction sequencing and scheduling. In *Workshop on Computer Architecture Technology and Formalism for Computer Science Research and Applications*, 1992. Naples, Italy.

[5] C. Becker and A. Scholl. A survey on problems and methods in generalized assembly line balancing. *European Journal of Operational Research*, 168:694–715, 2006.

[6] A. Bolat. Sequencing jobs on an automobile assembly line: objectives and procedures. *International Journal of Production Research*, 32(5):1219–1236, 1994.

[7] J. Bukchin. A comparative study of performance measures for throuput of a mixed model assembly line in a jit environment. *International Journal of Production Research*, 36(10):2669–2685, 1998.

[8] O. Cepek, M. Okada, and M. Vlach. Nonpreemptive flowshop scheduling with machine dominance. *European Journal of Operational Research*, 139(2):245–261, 2002.

[9] F.Y. Ding and H. Sun. Sequence alteration and restoration related to sequenced parts delivery on an automobile mixed-model assembly line with multiple departments. *International Journal of Production Research*, 42(8):1525–1543, 2004.

[10] T. Engstrőm, D. Jonsson, and B. Johansson. Alternatives to line assembly: Some swedish examples. *International Journal of Industrial Ergonomics*, 17(3):235–245, 1996.

[11] E. Erel and S.C. Sarin. A survey of the assembly line balancing procedures. *Production Planning and Control*, 9(5):414–434, 1998.

[12] S. French. *Sequencing and Scheduling.* John Wiley, New York, 1982.

[13] J.W. Herrmann, L. Chung-Yee, and J.L. Snowdon. A classification of static scheduling problems. *Complexity in Numerical Optimization*, P.M. Pardalos, Editor, World Scientific Publishing Co., Singapore, 203-253, 1993.

[14] R.R. Inman. Asrs sizing for recreating automotive assembly sequences. *International Journal of Production Research*, 41(5):847–863, 2003.

[15] S.M. Johnson. Optimal two and three stage production schedules with setup times included. *Naval Research Logistics Quaterly*, 1(1):61–68, 1954.

[16] Y.K. Kim, C.J. Hyun, and Y. Kim. Sequencing in mixed model assembly lines: A genetic algorithm approach. *Computers Operations Research*, 23(12):1131–1145, 1996.

[17] B.J. Lageweg, J.K. Lenstra, and K. Rinnooy. A general bounding scheme for the permutation flowshop problem. *Operations Res.*, 26:53–67, 1978.

[18] M. Lahmar, H. Ergan, and S. Benjaafar. Resequencing and feature assignment on an automated assembly line. *IEEE Transactions on Robotics and Automation*, 19(1):89–102, 2003.

[19] H.F. Lee and S.K. Schaefer. Sequencing methods for automated storage and retrieval systems with dedicated storage. *Computers and Industrial Engineering*, 32(2):351–362, 1997.

[20] H. Niu. Sequencing and balancing problem of mixed-model-assembly-line with window cycle time. *Doctoral Thesis, Universitat Politécnica de Catalunya, Barcelona, Spain*, 2005.

[21] M. Pinedo. *Scheduling. Theory, Algorithms and Systems.* Prentice Hall Inc., Englewood Cliffs, New Jersey, 1 edition, 1995.

[22] J. Plans and A. Corominas. A classification scheme for assembly line problems. *Technical Document IOC-DF-P-2000-20, Universitat Politécnica de Catalunya, Barcelona, Spain*, 2000.

[23] P. Rachakonda and S. Nagane. Simulation study of paint batching problem in automobile industry. *http://sweb.uky.edy/˜pkrach0/Projects/MFS605Project.pdf*, 2000. consulted 14.07.2004.

[24] B.R. Sarker and H. Pan. Design configuration for a closed-station, mixed-model assembly line: a filing cabinet manufacturing system. *International Journal of Production Research*, 39(10):2251–2270, 2001.

[25] A. Scholl. Balancing and sequencing of assembly lines. 1995.

[26] X. Sun, K. Morizawa, and H. Nagasawa. Powerful heuristics to minimize makespan in fixed, 3-machine, assembly-type flowshop scheduling. *European Journal of Operational Research*, 146(3):498–516, 2003.

[27] G.E. Vieira, J.W. Herrmann, and E. Lin. Rescheduling manufacturing systems: A framework of strategies, policies and methods. *Journal of Scheduling*, 6(1):39–62, 2003.