

Deterministic Assembly Scheduling Problems: A Review and Classification of Concurrent-Type Scheduling Models and Solution Procedures*

Jose M. Framinan[†] Paz Perez-Gonzalez Victor Fernandez-Viagas

Industrial Management, School of Engineering,
University of Seville. Camino de los Descubrimientos s/n. 41092 Seville, Spain

In memory of Rainer Leisten

Abstract

Many activities in industry and services require the scheduling of tasks that can be concurrently executed, the most clear example being perhaps the assembly of products carried out in manufacturing. Although numerous scientific contributions have been produced on this area over the last decades, the wide extension of the problems covered and the lack of a unified approach have lead to a situation where the state of the art in the field is unclear, which in turn hinders new research and makes translating the scientific knowledge into practice difficult.

In this paper we propose a unified notation for assembly scheduling models that encompass all concurrent-type scheduling problems. Using this notation, the existing contributions are reviewed and classified into a single framework, so a comprehensive, unified picture of the field is obtained. In addition, a number of conclusions regarding the state of the art in the topic are presented, as well as some opportunities for future research.

Keywords: Scheduling; Assembly; Review; Order Scheduling; Concurrent Operations; Distributed Flowshop Scheduling; Agile Manufacturing Systems

*Preprint submitted to European Journal of Operational Research. DOI: <https://doi.org/10.1016/j.ejor.2018.04.033>

[†]Corresponding author. Tel. +3495 448 7214; fax: +3495 448 7329. E-mail address: framinan@us.es

1 Introduction

In industry and services it is rather a common situation where one activity can start only if a set of previous tasks (sometimes carried out by different resources) have been completed. These tasks may be jointly scheduled in the different resources of the company in order to achieve the desired goals, as individual schedules in these resources would not ensure the necessary synchronisation to efficiently complete the tasks. Arguably the most visible example of this type of problem occurs in the assembly of parts in the manufacturing industry, henceforth the problem is known as the *Assembly Scheduling Problem* or ASP. Recently, in most industrial sectors, the strategy has changed from the paradigm of mass production to the manufacture of complex products using postponement strategies that usually entail the assembly of customised features, mainly due to the increasing competition among companies, and the increase of customer power. As a result, most companies nowadays focus on manufacturing only the specific parts of their productive processes in which they excel, and on outsourcing/purchasing those components/operations that do not constitute their core-competences. This leads to an increasing prevalence of assembly operations and, as a consequence, the ASP is being considered a very relevant problem by academics and practitioners. Not surprisingly, a huge number of contributions in the area have been produced in the last decades –as our review, with more than 200 papers, can attest–. However, to the best of our knowledge, almost no attempt to review or classify these works has been carried out: we recall only the work by Khodke and Bhongade (2013) where contributions regarding real-time operation of assembly systems are discussed, although it covers a rather limited part of the references discussed here–, and the topic has suffered from this lack of unified approach: Since a unified classification or notation for the different problems does not exist, the same models are given different names in different papers, and the case is not uncommon where some works neglect/ignore the existence of related results that could be applicable/compared. As a result, the-state-of-the-art of some problems and sub-problems remains unclear, which hinders the advance of research results and their applications in practice.

In this paper, we aim to tackle these issues by proposing a unified notation for assembly scheduling problems. Using this notation, we review and classify the existing contributions in the area. The goal is to provide a global perspective of the field so the main results can be identified. Furthermore, we also discuss some open problems and possible research avenues.

The remainder of the paper is structured as follows: In Section 2 we discuss the background and the research methodology adopted. The formal statement of the problem and the proposed notation are presented in Section 3. In Section 4 we discuss the contributions when the operations take place in dedicated machines. In Section 5 we review the research where flexible machines are employed. The contributions dealing with assembly in several (distributed) factories are reviewed in Section 6, while general layout models are discussed

in Section 7. Finally, in Section 8 the main conclusions and future research avenues are presented.

2 Background and Research Methodology

Many products are composed of assemblies and sub-assemblies. Commonly cited examples in the ASP literature include fire engine production (Lee et al., 1993), assembly flow shop in the computer industry (Potts et al., 1995), assembly of the body and chassis in the car industry (Fattahi et al., 2013b), circuit board production (Cheng and Wang, 1999), or plastic industry (Allahverdi and Aydilek, 2015), among others. Examples can also be found in services/IT, including distributed database systems (Allahverdi and Al-Anzi, 2006b, 2012), or multi-page invoice printing systems (Zhang et al., 2010).

The structure of a product can be depicted using an acyclic direct graph or *digraph* (Kusiak, 1989) where the nodes represent operations and the arcs the precedence relations between the operations (He and Kusiak, 1998). In this manner, the digraph describes the sequence of operations required to complete a product. In this regard, the classification by Kusiak (1989) distinguishes between *simple* and *complex* products, the first one being a product that has at most one assembly operation in each level of assembly. However, note that this operation-oriented description of the manufacturing of a job is, in general, different to the resource-oriented description, since the sequence of operations required for a product may be either performed each one by dedicated resources (i.e. machines), or by flexible resources that may carry out several operations.

Since scheduling deals with the allocation of operations to resources across time (see e.g. Framinan et al., 2014), the description of the set of resources that can carry out the different operations –i.e. the machine environment– is a critical data that usually determines the difficulty of the scheduling decisions. In this regard, the classical notation for scheduling problems (Graham et al., 1979) focuses on the machine environment rather than solely on the sequence of operations, and this view will also be adopted in this paper for the proposed notation. In the case of assembly operations, such a view naturally leads to a distinction among scheduling using *flexible* machines (i.e. machines that can carry out several operations on the jobs) and *dedicated* machines (i.e. machines that can only carry out one and only one type of operation on the job). This distinction will be employed in this review to classify the contributions.

Although we retain the term *assembly*, the review focuses on scheduling problems where some operations may be performed concurrently (i.e. the existence of a set of operations that can be carried out in parallel to be completed before the starting of another operation), which does not necessarily mean a physical assembly. In this regard, papers related to the so-called *order scheduling problem* are included as a special case of concurrency of operations. In the standard order scheduling problem, a set of orders composed of jobs have to be scheduled in different (usually dedicated) machines in order to minimise some order-related objective. Clearly, these jobs can be modelled as operations that should be completed so the order (a job, in the ASP

terminology) is finished. As a consequence, this type of problems can be regarded as assembly problems where the assembly operation has zero processing times, a connection already remarked by Kyparisis and Koulamas (2002). Furthermore, by focusing on the concurrency of operations, the literature on the so-called *distributed assembly flowshop scheduling problem* can also be integrated, and some insights derived from this connection are discussed in the conclusions.

This focus on scheduling with concurrent operations makes the wide range of literature on assembly lines to be out of the scope, as the specific concurrency problem is not dealt with in these papers. Analogously, papers devoted to planning assembly operations (which do not explicitly consider the detailed capacity of the resources) are not included in the review. Finally, we focus the review on papers assuming a deterministic setting. Although a number of papers address the stochastic counterpart, such distinction is customary in most scheduling research, and trying to integrate stochastic assembly scheduling would produce an exceedingly large review.

3 Problem Statement and Proposed Notation

The problem can be formalised as follows: Each job j ($j = 1, \dots, n$) consists of a chain of operations along s stages ($\{O_{1,j}^1, \dots, O_{r_1,j}^1\}, \{O_{1,j}^2, \dots, O_{r_2,j}^2\}, \dots, \{O_{1,j}^s, \dots, O_{r_s,j}^s\}$); operation $O_{r,j}^k$ is the r th concurrent operation that should be performed on job j at stage k (r_k indicates the number of operations to be carried out at stage k). If there is only one operation at stage k , then the operation sub-index can be dropped, i.e. O_j^k indicates the (only) operation to be performed at stage k on job j , that is $O_j^k = O_{1,j}^k$.

Each operation $O_{r,j}^k$ may have technological constraints that impede the starting of the operation unless a set of operations in the previous stages –denoted by $\mathcal{A}(O_{r,j}^k)$ – have been completed. Each operation can be classified either as *concurrent* or *merge* type (the latter type indicated by an overline) depending on the cardinality of the set $\mathcal{A}(O_{r,j}^k)$. If $|\mathcal{A}(O_{r,j}^k)| \leq 1$, then $O_{r,j}^k$ is of concurrent type. If $|\mathcal{A}(O_{r,j}^k)| > 1$, then $O_{r,j}^k$ is of merge type. The scheduling problem is then completely defined by giving, for each machine i , its processing times for each operation $p_i(O_{r,j}^k)$. For simplicity, if there is only one machine at stage k , the processing times are denoted by $p(O_{r,j}^k)$. Although not strictly required, we will assume that each machine belongs only to one stage.

Some attempts have been made by several researchers to introduce a notation for these types of scheduling problems (most times just covering a subset of assembly scheduling problems), all of them based on the standard 3-field notation by Graham et al. (1979). Although all these proposals have some merits, they are oriented towards specific problem types and do not cover all the extension of ASP. We therefore propose the following notation: If we consider a single-assembly case –i.e. only one operation of merge type–, these layouts can be noted as

$$\alpha_1 \rightarrow \alpha_2 | \beta | \gamma$$

where α_1 is the machine environment prior to the concurrent operation. In this environment, all operations preceding the concurrent operation are performed. α_2 is the machine environment of the concurrent and subsequent operations. In this environment, assembly and subsequent operations take place. Note that, although in most models the first stage of the α_2 layout is the assembly stage, it does not always have to be the case, being an example the papers where the components of a product are grouped together to be firstly transported (first stage of α_2) and secondly assembled in further stages (successive stages of α_2). However, since the collective transportation of the components of the products requires that all of them have been completed, the entity to be processed in α_2 is the job (with all its operations).

The following values for α_1 can be identified in the literature:

- 1: Single machine: In this case, the single machine is a flexible machine capable of carrying out all operations of the job to be later assembled. In some papers, such assembly layout with a flexible machine is denoted as *agile manufacturing system*.
- $Pm/Qm/Rm$: m parallel machines can carry out each one of the operations of the job. Pm assumes identical machines, whereas Qm assumes related machines. Finally, the unrelated parallel machine case is denoted by Rm . Since the machines are flexible, the number of different operations is not limited to the number of machines.
- DPm : m dedicated parallel machines carry out a specific operation of the job. *Dedicated* in this context means that each machine can carry out one and only one specific type of operation. Therefore, a job consists of, at maximum, m operations before being assembled.
- Fm : A flowshop consisting of m (flexible) machines can carry out each one of the operations of the job before its assembly. More specifically, all operations prior to the assembly take place in a m -machine flowshop.
- DFm : f dedicated m -machine flowshops carry out each one of the operations of the job. In other words, each set of operations prior to the assembly takes place in one specific flowshop layout. Since each flowshop is dedicated to one specific type of operation, the product is composed of, at maximum, f components to be assembled.
- PFm : f flexible m -machine flowshops in parallel are able to carry out each one of the operations prior to the assembly of the job. More specifically, the machine environment is composed of f (possibly different) m -machine flowshops, each one capable to carry out all operations of the jobs to be assembled. Therefore, in this case the number of components of the product is not limited to f .

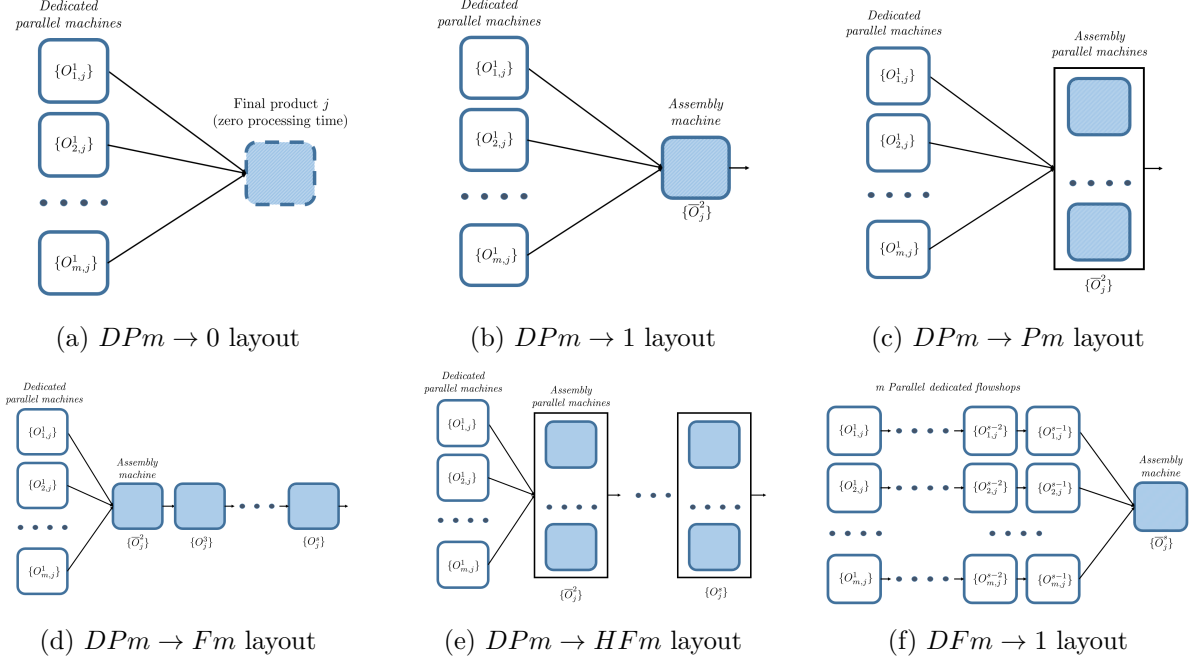


Figure 1: Several layouts for the case with dedicated machines prior to the assembly operations.

- *HFm*: The operations prior to the assembly of the jobs take place in a flexible m -stage hybrid flowshop before the assembly. More specifically, all components of the products can be manufactured in the hybrid flowshop.

Clearly, the previous values of the α_1 parameter can also be identified for α_2 . In addition, $\alpha_2 = 0$ means that the processing times of the assembly operation are zero. This may serve to model the order scheduling problem (see Section 4.1). In this problem, a number of orders –each one composed of up to m different products– have to be scheduled in m machines, where each machine can manufacture only an specific product. Clearly, the order is only completed when all products forming the order have been completed, henceforth it can be seen as an assembly operation with zero-assembly times. Figures 1 and 2 show some samples of the typical layouts, while instance problems for the main layouts and Gantt charts of a sample schedule are provided in the additional on-line material.

Regarding the different constraints that may appear in the β field, we adopt the standard notation trying to make it as clear as possible. Henceforth, the constraints should be interpreted as they apply to some jobs/stages/machines of the process, but not necessarily in all stages/machines. For instance, $recrc$ in the $1 \rightarrow 1|recrc|C_{max}$ problem indicates that some jobs –but not all– visit at least one machine –but not all– more than once. Finally, regarding the γ field, the standard notation is adopted, using the extension by T’Kindt and Billaut (2006) for the multicriteria case. A full description of the three fields is given as additional on-line material.

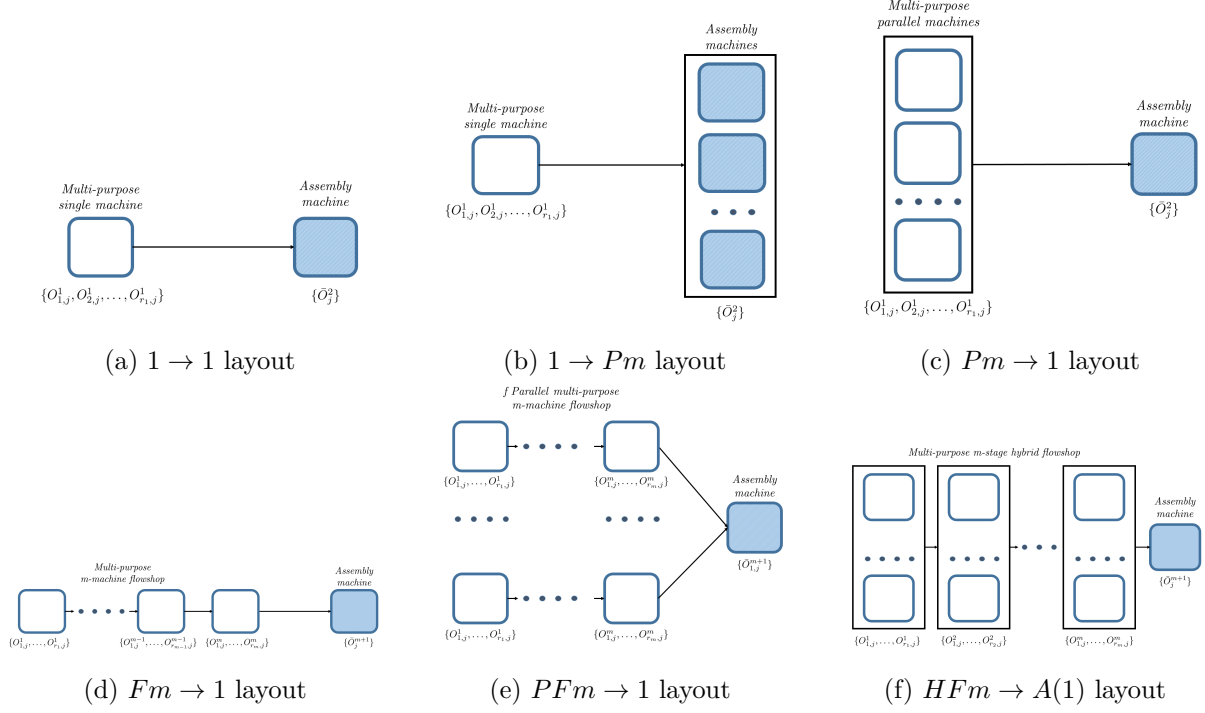


Figure 2: Several layouts for the flexible machine case

Note that the proposed notation can also model multi-assembly layouts by nesting concurrency arrows in parenthesis: In this manner, the complex layout in Figure 3 can be noted as $((DPm \rightarrow 1, Fm) \rightarrow Pm, 1) \rightarrow Fm | \beta | \gamma$. Although we seldom make use of this extended notation in this paper, we believe that this ability to be extended to complex layouts is another interesting feature of the proposed notation. Finally, note that some of the simple layouts described before, such as DPm or DFm , could be denoted also as $(1, \dots, 1) \rightarrow \alpha_2$, or as $(Fm, \dots, Fm) \rightarrow \alpha_2$, respectively. However, we believe that DPm and DFm provide a more compact notation and will be used henceforth.

Equipped with this notation, we will review the existing contributions in the following sections. Since most of them focus on single-assembly layouts, and given the fact that assembly problems are quite different conceptually depending on whether the operations prior to assembly take place in dedicated machines, or not, Section 4 is devoted to single-assembly scheduling problems with dedicated machines in the α_1 field. Section 5 is devoted to scheduling problems with flexible machines in the α_1 field. Section 6 is devoted to distributed assembly problems, and Section 7 discusses general assembly models.

To facilitate the study of the different contributions, Tables 1 to 6 are provided as summaries of the different layouts, noting the main focus of the references. The abbreviations used in these tables are:

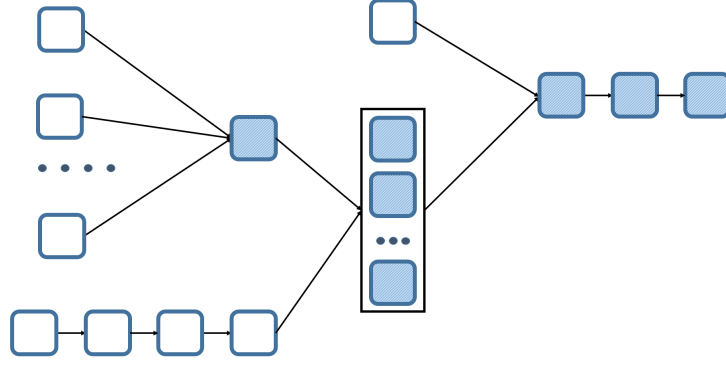


Figure 3: A complex assembly shop layout

CA	Analysis of the Complexity of the Problem
DR	Dominance Rules
LB	Development of Lower Bounds for the problem
PP	Problem Properties (other than dominance rules/lower bounds)
DP	Dynamic Programming
BB	Development of a Branch & Bound procedure
PA	Proposal of Polynomial-Time Algorithms
IP	Development of an Integer Programming (Not Linear) model
MILP	Development of a MILP model
QM	Development of a quadratic model
CH	Proposal of Constructive Heuristics
MH	Proposal of Meta-Heuristics

4 Assembly scheduling with dedicated machine

Although different dedicated layouts can be employed to carry out the operations prior to the assembly of the jobs, we are only aware of contributions dealing with dedicated machines in parallel, i.e. the $DPm \rightarrow \alpha_2$ layout. In this setting, r_1 the number of concurrent operations in the first stage is assumed to be m . The following subsections discuss the related contributions.

4.1 Order scheduling: $DPm \rightarrow 0$ layout

The main contributions regarding this layout –depicted in Figure 1a– are summarised in Table 1. Depending on the objective, the problems have different complexity (see Leung et al., 2005a for a review of the complexity

Problem*	Reference	Analysis	Exact	Approximate
$DPm \rightarrow 0 C_{max}$	Wagneur and Sriskandarajah (1993)	CA		
	Leung et al. (2006b)	CA		CH
$DPm \rightarrow 0 \sum C_j$	Wagneur and Sriskandarajah (1993)	CA		
	Roemer and Ahmadi (1997)	CA		
	Roemer (2006)	CA		
	Leung et al. (2005b)	PP		CH
	Framinan and Perez-Gonzalez (2017b)			CH, MH
$DP2 \rightarrow 0 \sum w_j C_j$	Sung and Yoon (1998)	CA		CH
$DPm \rightarrow 0 \sum w_j C_j$	Wang and Cheng (2003)			CH
	Ahmadi et al. (2005)	CA, LB		CH
	Garg et al. (2007)	CA		CH
	Wang and Cheng (2007)			CH
	Leung et al. (2007a)			CH
	Leung et al. (2008b)	PP		CH
	Wang et al. (2013)		QM	
	Shi et al. (2017)		QM	
$DPm \rightarrow 0 r_j \sum w_j C_j$	Garg et al. (2007)	CA		CH
$DPm \rightarrow 0 L_{max}$	Wagneur and Sriskandarajah (1993)	CA		
$DPm \rightarrow 0 \sum T_j$	Wagneur and Sriskandarajah (1993)	CA		
	Lee (2013)	PP	BB	CH
	Framinan and Perez-Gonzalez (2018)			CH, MH
$DPm \rightarrow 0 learning - eff. \sum T_j$	Xu et al. (2016)	DR	BB	MH
$DPm \rightarrow 0 \sum U_j$	Wagneur and Sriskandarajah (1993)	CA		
	Leung et al. (2006b)	PP	BB	CH
	Lin and Kononov (2007)	CA		CH
$DPm \rightarrow 0 \sum w_j U_j$	Lin and Kononov (2007)	CA		CH
$DPm \rightarrow 0 \epsilon (\sum C_j^A, \sum C_j^B)$	Lin et al. (2017)	DR	BB	MH
$DP2 \rightarrow 0 Cost\ function$	Li and Ou (2007)			CH
$DP2 \rightarrow 0 Cost\ function$	Li and Vairaktarakis (2007)			CH
$DPm \rightarrow 0 F(f_1(C_1), \dots, F(f_n(C_n)))$	Cai and Zhou (2004)	PP	DP	

Table 1: Summary of contributions in Section 4.1 (*Notation is described in the additional on-line material)

of some of the problems). For makespan minimisation without additional constraints, the problem has been shown to be polynomial by Wagneur and Sriskandarajah (1993), being the makespan value independent of the schedule. The total completion time objective has received a lot of attention: The problem has been first addressed by Wagneur and Sriskandarajah (1993), who showed it to be NP-hard, although their proof contained a flaw and its complexity was not confirmed until Roemer and Ahmadi (1997). Other contributions regarding the complexity of the problem for the weighted case are Ahmadi et al. (2005) and Roemer (2006). Also for the weighted case, the NP-hardness of the specific case of $m = 2$ was independently established by Sung and Yoon (1998).

Regarding exact procedures, Wang et al. (2013) and Shi et al. (2017) propose a quadratic formulation of the weighted case. With respect to approximate procedures, different constructive heuristics have been proposed by Leung et al. (2005b) and by Framinan and Perez-Gonzalez (2017b), the latter being the best procedure with regard to approximate methods. Approximate solutions for the weighted version of the problem are given by Wang and Cheng (2003); Ahmadi et al. (2005); Garg et al. (2007); Leung et al. (2007a); Wang and Cheng (2007) and Leung et al. (2008b), although only the paper by Ahmadi et al. (2005) addresses a computational experience to test the performance of the proposed heuristics, which are not compared with other available heuristics. The rest of the papers are devoted to establishing the worst-case performance of

the heuristics. The works by Leung et al. (2007a, 2008b) present a computational experience of the heuristics proposed, but in the testbed it is assumed that all weights are the same and the heuristics are not compared with existing ones. For the specific two machines/components case and weighted jobs, Sung and Yoon (1998) propose some constructive heuristics. Finally, Lin et al. (2017) address the version with interfering jobs, where one set of jobs has to minimise the completion time, and the completion time of the other set of jobs cannot exceed a given value (see e.g. Perez-Gonzalez and Framinan, 2014 for a review on scheduling problems with interfering jobs). One B & B with several dominance rules is presented, and some metaheuristics are proposed.

Regarding due-date related objectives, Wagneur and Sriskandarajah (1993) and Leung et al. (2006b) show that the EDD rule is optimal for the maximum lateness. The problem with total tardiness is shown to be unary NP-hard by Wagneur and Sriskandarajah (1993), and Lee (2013) presents a B & B procedure and several constructive heuristics for the problem. For this problem, Framinan and Perez-Gonzalez (2018) propose both a fast constructive heuristic and a high-performing matheuristic. Xu et al. (2016) address the problem with position-based learning effect, proposing dominance rules, a B & B procedure, and two metaheuristics. The objective of minimisation of the number of late jobs is shown to be binary NP-hard by Wagneur and Sriskandarajah (1993). In fact, Lin and Kononov (2007) show that even the special case with common due dates is also NP-hard. These authors propose an LP-based approximate algorithm for the weighted and unweighted cases, and a constructive heuristic for the common due date case for the $\sum U_j$ and $\sum w_j U_j$ problems. An exact algorithm for the problem is also proposed by Leung et al. (2006b) together with a heuristic and a greedy algorithm for the $\sum U_j$ problem with common due dates, although their contributions are not compared with those by Lin and Kononov (2007).

Regarding general cost functions, Li and Ou (2007) and Li and Vairaktarakis (2007) address the $DP2 \rightarrow 0$ layout with a cost function, for which the authors develop some polynomial time heuristics. Cai and Zhou (2004) study the problem for max-type and sum-type non-decreasing functions of the completion times of the jobs in the context of teamworks. They present some problem properties and a dynamic programming procedure, although this reference is disconnected from the main stream of order scheduling and it partially overlaps with existing works.

4.2 Single machine in the assembly stage: $DPm \rightarrow 1$ layout

This problem –depicted in Figure 1b– has been studied for the case where there are two dedicated parallel machines in the α_1 field, as well as for the general m dedicated machines case. A summary of the contributions is shown in Table 2. Note that, since this layout can be seen as a generalisation of the classical 2-machine flowshop, its complexity could be discussed only for the makespan objective, as the rest of usual objectives

Problem*	Reference	Analysis	Exact	Approximate
$DP2 \rightarrow 1 C_{max}$	Lee et al. (1993)	CA	BB	CH
	Haouari and Daouas (1999)	PP, LB	BB	CH
	Karuno and Nagamochi (2002)	LB, PP		CH
	Sun et al. (2003)			CH
$DPm \rightarrow 1 C_{max}$	Potts et al. (1995)	CA		CH
	Hariri and Potts (1997)	DR, LB	BB	
$DP2 \rightarrow 1 r_j C_{max}$	Sung and Juhn (2009)	CA, PP, DR	BB	CH
$DPm \rightarrow 1 r_j C_{max}$	Komaki and Kayvanfar (2015)	LB		CH, MH
$DP2 \rightarrow 1 BS C_{max}$	Lin et al. (2006)	CA, PP, LB		CH
$DPm \rightarrow 1 ST_{si} C_{max}$	Allahverdi and Al-Anzi (2006a)	DR		CH, MH
$DPm \rightarrow 1 a_i, ptmn - nr C_{max}$	Hadda et al. (2007)	PP		CH
$DPm \rightarrow 1 a_i, pmtn - semi C_{max}$	Hadda et al. (2014)	PP		CH
$DPm \rightarrow 1 no - wait C_{max}$	Mozdgir et al. (2013b)			MH
$DPm \rightarrow 1 p - batch, ST_{si} C_{max}$	Kovalyov et al. (2004)	CA		CH
$DPm \rightarrow 1 s - batch, ST_{si} C_{max}$	Kovalyov et al. (2004)	CA		CH
$DP3 \rightarrow 1 learning - effect \sum C_j$	Wu et al. (2018)	PP	BB	MH
$DPm \rightarrow 1 \sum C_j$	Al-Anzi and Allahverdi (2006)			CH, MH
	Blocher and Chhaged (2008)	PP, LB		CH
	Allahverdi and Al-Anzi (2012)			MH
	Framinan and Perez-Gonzalez (2017a)			CH, MH
$DPm \rightarrow 1 ST_{si} \sum C_j$	Allahverdi and Al-Anzi (2008c)	DR		MH
	Allahverdi and Al-Anzi (2009)			MH
$DPm \rightarrow 1 \sum w_j C_j$	Tozkapan et al. (2003)	DR, LB	BB	
$DPm \rightarrow 1 L_{max}$	Allahverdi and Al-Anzi (2006b)			MH
$DPm \rightarrow 1 ST_{si} L_{max}$	Al-Anzi and Allahverdi (2007a)	DR		MH
	Al-Anzi and Allahverdi (2007b)	DR		MH
$DP2 \rightarrow 1 \sum T_j$	Allahverdi and Aydilek (2015)			MH
	Lee and Bang (2016)	DR, MILP	BB	CH
$DPm \rightarrow 1 ST_{si} \sum T_j$	Allahverdi et al. (2016b)	DR		MH
$DPm \rightarrow 1 ST_{si} T_{max}$	Aydilek et al. (2017)	DR		MH
$DPm \rightarrow 1 \sum U_j$	Allahverdi et al. (2016a)			CH, MH
$DPm \rightarrow 1 F_l(C_{max}, \sum C_j)$	Allahverdi and Al-Anzi (2008a)			MH
	Torabzadeh and Zandieh (2010)			MH
	Shokrollahpour et al. (2011)			MH
	Seidgar et al. (2014)		MILP	MH
$DPm \rightarrow 1 ST_{si} F_l(C_{max}, \sum C_j)$	Tian et al. (2013)			MH
$DPm \rightarrow 1 F_l(C_{max}, \sum U_j)$	Navaei et al. (2010)			MH
$DPm \rightarrow 1 F_l(C_{max}, L_{max})$	Al-Anzi and Allahverdi (2009)			MH
$DPm \rightarrow 1 ST_{sd} F_l(C_{max}, T_{max})$	Maboudian and Shafaei (2009)	LB	IP	
$DPm \rightarrow 1 F_l(C_{max}, L_{max}, E_{max})$	Yan et al. (2014)	DR		MH
$DPm \rightarrow 1 F_l(maintenance, breakdowns)$	Seidgar et al. (2016)		MILP	MH

Table 2: Summary of contributions in Sections 4.2. (*Notation is described in the additional on-line material).

are NP-hard for the 2-machine flowshop.

Regarding the case of two machines, the $DP2 \rightarrow 1||C_{max}$ problem has been shown to be strongly NP-hard by Lee et al. (1993), although it is polynomially solvable if, for each job, its processing time on one of the dedicated machines is high or equal than on the other machine. These authors develop a B & B procedure for the problem, together with some heuristics. Another approximate algorithm with a tighter worst case is given by Karuno and Nagamochi (2002). Some constructive, real-time heuristics are proposed for the problem by Sun et al. (2003). An extremely efficient B & B algorithm is given by Haouari and Daouas (1999). The authors show that instances with $n \leq 200$ could be solved in few seconds, so it can be stated that this problem is optimally solvable for a large number of jobs, while there are extremely good and fast approximate algorithms if very large instances are considered.

Some extensions of the $DP2 \rightarrow 1||C_{max}$ problem have been also addressed: Sung and Juhn (2009)

consider that one of the components is purchased, therefore it is only available after a given time. The authors prove that this problem is NP-hard and show that only permutation schedules have to be considered. They also propose a B & B procedure and three heuristic algorithms for the problem. Note that this problem can be seen as a particular case of the $DP2 \rightarrow 1|r_j|C_{max}$ assuming that the release date of the component that is not purchased is zero for all products, and that the processing time of the purchased component is zero for all products, a problem addressed by Komaki and Kayvanfar (2015) for m machines that is discussed below. The same problem, assuming that the assembly machine processes batches of specific jobs, and that a (non-sequence dependent) setup is required every time a batch is formed, is addressed by Lin et al. (2006), i.e. the $DP2 \rightarrow 1|BS|C_{max}$ problem. The authors prove that this problem is strongly NP-hard also in the special case where the processing times of the assembly machine are the same for all jobs. The authors also detect some few polynomially solvable cases, and provide heuristics for the problem. Lee and Bang (2016) propose a branch and bound procedure and a constructive heuristic for the $DP2 \rightarrow 1||\sum T_j$ problem. Finally, the three-machine case with total flowtime objective and learning effect has been addressed by Wu et al. (2018), who present a B & B procedure and a metaheuristic.

For the general m -machine case, this problem has been addressed by many authors, for different objectives and with some extensions. Regarding the makespan, Potts et al. (1995) are the first to address the problem. Independently from Lee et al. (1993), they show it is strongly NP-hard, and develop a heuristic providing a worst case of $2 - 1/m$ (note that this reduces to that by Lee et al., 1993 if $m = 2$). Hariri and Potts (1997) propose a B & B algorithm able to provide optimal solutions in few seconds for more than 5,000 jobs and 10 machines. Given these results, it can be concluded that, for most realistic-size instances, this problem can be solved. In addition, the heuristics by Sun et al. (2003) –originally proposed for $m = 2$ – are extended for the general case with excellent results.

The problem with makespan and setup times is addressed using constructive heuristics and metaheuristics by Allahverdi and Al-Anzi (2006a). Hadda et al. (2007) and Hadda et al. (2014) address the $DPm \rightarrow 1|a_i, pmtn - semi|C_{max}$ and $DPm \rightarrow 1|a_i, pmtn - nr|C_{max}$ problems, proposing some heuristic algorithms and giving their worst cases. The $DPm \rightarrow 1|r_j|C_{max}$ problem is addressed by Komaki and Kayvanfar (2015) by means of constructive heuristics based on dispatching rules and metaheuristics. Three metaheuristics are also proposed by Mozdgir et al. (2013b) for the $DPm \rightarrow 1|no - wait|C_{max}$ problem. Finally, the problem with makespan objective and batches is addressed by Kovalyov et al. (2004). The authors assume that the manufacturing operations on machine i are carried out in batches of maximum size β_i , and that assembly is performed in batches of maximum size β_A . In this setting, they explore the $p - batch$ and $s - batch$ cases, the latter also with setups.

The $DPm \rightarrow 1||\sum w_j C_j$ problem is addressed by Tozkapan et al. (2003) for the first time. The authors show that the optimal solutions are non-delay, permutation schedules (this result is later presented

independently by Blocher and Chhajer, 2008 for the un-weighted case). They propose a B & B algorithm and constructive heuristics. The rest of the existing contributions for this problem have focused on the un-weighted case: the heuristics by Tozkapan et al. (2003) are outperformed by those in Al-Anzi and Allahverdi (2006), who also propose several metaheuristics. These metaheuristics are further improved by a new set of metaheuristics by Allahverdi and Al-Anzi (2012). The best-so-far approximate heuristics for the un-weighted case are given by Framinan and Perez-Gonzalez (2017a). As mentioned before, an earlier contribution is Blocher and Chhajer (2008), who independently derive properties for some special cases of the $DPm \rightarrow 1 || \sum C_j$ problem and propose constructive heuristics, most of them similar/equivalent to those proposed in the previous references. Finally, when setup times have to be taken into account, Allahverdi and Al-Anzi (2008c, 2009) present several metaheuristics for the $DPm \rightarrow 1 | ST_{si} | \sum C_j$ problem.

Regarding due-date objectives, the $DPm \rightarrow 1 || L_{max}$ problem is addressed by Allahverdi and Al-Anzi (2006b) using metaheuristics. This objective with sequence-independent setups is studied in Al-Anzi and Allahverdi (2007b,a), also using metaheuristics. The $DPm \rightarrow 1 || \sum T_j$ problem has been addressed by Allahverdi and Aydilek (2015), and Lee and Bang (2016). The first reference propose some metaheuristics, while the latter proposes a MILP and a B & B procedure. The case where there are sequence-independent setups is studied by Allahverdi et al. (2016b), who propose several metaheuristics. The objective of maximum tardiness with sequence-independent setups is addressed by Aydilek et al. (2017), who study some problem properties and propose a metaheuristic. Finally, Allahverdi et al. (2016a) address the $DPm \rightarrow 1 || \sum U_j$ problem also by means of metaheuristics.

In addition to the single criterion problems, there are several contributions focusing on a combination of criteria for the $DPm \rightarrow 1$ layout, including:

- The $DPm \rightarrow 1 || F_l(C_{max}, \sum C_j)$ problem has been addressed by several researchers: Allahverdi and Al-Anzi (2008a), Torabzadeh and Zandieh (2010), Shokrollahpour et al. (2011), Tian et al. (2013) – in this case considering setup times –, and Seidgar et al. (2014). All these papers propose different metaheuristics and only partial comparisons exist among them, so the state-of-the-art remains unclear due to the lack of a comprehensive computational evaluation.
- Navaei et al. (2010) propose two metaheuristics for the makespan and the number of tardy jobs, i.e. the $DPm \rightarrow 1 || F_l(C_{max}, \sum U_j)$ problem.
- Allahverdi and Al-Anzi (2008b) and Al-Anzi and Allahverdi (2009) develop different metaheuristics for the $DPm \rightarrow 1 || F_l(C_{max}, L_{max})$ problem.
- Maboudian and Shafaei (2009) study the $DPm \rightarrow 1 | ST_{sd} | F_l(C_{max}, T_{max})$ problem, proposing a non-linear mathematical model and a lower bound.

- Yan et al. (2014) propose a metaheuristic for a linear combination of makespan, maximum earliness and maximum tardiness, i.e. the $DPm \rightarrow 1||F_l(C_{max}, E_{max}, T_{max})$ problem.
- Seidgar et al. (2016) address the problem with preventive maintenance and objectives related to makespan and system unavailability. A MILP model and several metaheuristics are proposed.

Finally, note that, although the problem in the $DPm \rightarrow 1$ setting usually considered in the literature consists of finding a sequence for the jobs, Hwang and Lin (2012) address the problem of obtaining an optimal batching decision given a fixed sequence of jobs for a number of regular metrics.

Problem*	Reference	Analysis	Exact	Approx.
$DPm \rightarrow Rm C_{max}$	Navaei et al. (2013)		MILP	MH
$DPm \rightarrow Pm ST_{sd} C_{max}$	Zhang et al. (2010)		MILP	MH
$DPm \rightarrow P2 \sum C_j$	Sung and Kim (2008)	PP	BB	
$DPm \rightarrow P2 \sum C_j$	Al-Anzi and Allahverdi (2012)			MH
	Allahverdi and Al-Anzi (2012)			MH
	Al-Anzi and Allahverdi (2013)			MH
$DPm \rightarrow Pm BS_{ij}, work - shift - lot \sum w_j C_j$	Nejati et al. (2016)		IP	MH
$DPm \rightarrow Rm ST_{sd} F_l(HC, DC)$	Navaei et al. (2014)		MILP	MH
$DPm \rightarrow Rm F_l(C_{max}, \sum C_j)$	Mozdgir et al. (2013a)		MILP	MH
$DPm \rightarrow F2 C_{max}$	Koulamas and Kyparisis (2001)	PP		CH
$DPm \rightarrow F2 C_{max}$	Komaki et al. (2017)	PP		MH
$DRm \rightarrow F2 ST_{sd} \sum C_j$	Andres and Hatami (2011)		MILP	
$DPm \rightarrow F2 ST_{sd} F_l(\sum C_j, T_{max})$	Hatami et al. (2010)	LB	MILP	MH
$DPm \rightarrow F2 a_i, pmtn - semi F_l(\sum w_j C_j, \sum w_j T_j, \sum w_j E_j)$	Shoardebili and Fattahi (2015)		MILP	MH
$DPm \rightarrow F2 ST_{sd} \#(\sum C_j, \sum T_j)$	Cunha Campos and Claudio Arroyo (2014)			MH
$DPm \rightarrow F2 \#(C_{max}, F_l(\sum w_j T_j, \sum E_j))$	Tajbakhsh et al. (2014)		MILP	MH
$DPm \rightarrow F2 ST_{sd}, blocking F_l(C_{max}, \sum w_j C_j)$	Maleki-Daroukolaei et al. (2012)		MILP	MH
	Maleki-Daroukolaei and Seyed (2013)			MH
$DPm \rightarrow F2 p - batch F_l(AT, DC)$	Wang et al. (2016)			CH, MH
$DPm \rightarrow HF2(1, DPm) \sum C_j$	Xiong et al. (2015)	LB	MILP	CH, MH

Table 3: Summary of contributions in Sections 4.3 and 4.4 (dedicated machines with more than a single machine at assembly stage). (*Notation is described in the additional on-line material).

4.3 Parallel machines in the assembly stage: $DPm \rightarrow Pm/Rm/Qm$ layouts

This case –depicted in Figure 1c– has been addressed for different objectives. The $DPm \rightarrow Rm||C_{max}$ problem has been modelled as an MILP by Navaei et al. (2013), who also propose a metaheuristic. The $DPm \rightarrow Pm|ST_{sd}|C_{max}$ problem is addressed by Zhang et al. (2010). These authors propose an MILP model and a decomposition approach, together with a genetic algorithm.

Regarding total completion time, Sung and Kim (2008) analyse the properties of the problem and propose a B & B procedure. Al-Anzi and Allahverdi (2012); Allahverdi and Al-Anzi (2012) and Al-Anzi and Allahverdi (2013) address the 2-machine and the general m -assembly machine problem using metaheuristics. The case of the total weighted completion time has been addressed when lot streaming and a work shift constraint is allowed by Nejati et al. (2016). A mathematical programming formulation of the problem is given and a metaheuristic is proposed to approximately solve it.

Navaei et al. (2014) address the $DPm \rightarrow Rm|ST_{sd}|F_l(DC, HC)$ problem, where DC and HC are the delay and holding costs, respectively. An MILP model is presented, and several metaheuristics are

proposed. Finally, Mozdgir et al. (2013a) propose an MILP model and some metaheuristics for the $DPm \rightarrow Rm|ST_{sd}|F_l(C_{max}, \sum C_j)$ problem.

4.4 Flowshop-type in the assembly stage: $DPm \rightarrow Fm$ and $DPm \rightarrow HFm$ layouts

The problems with a flowshop in the assembly stage –depicted in Figure 1d– have been addressed by a number of authors. Among these types of problems, the most studied is that where the flowshops consist of two stages, with the first one corresponding to a transportation stage and the second one the assembly of the components. Note that, in order to perform the transportation of all components of the product, the concurrency of operations is required. This $DPm \rightarrow F2$ setting is denoted in some literature as the three-stage assembly flowshop scheduling problem (see e.g. Koulamas and Kyparisis, 2001). Clearly, this problem is a generalisation of the 3-machine flowshop, and it is therefore NP-hard for all well-known criteria. A limited number of objectives have been studied. For makespan, the problem is first addressed by Koulamas and Kyparisis (2001), who propose several constructive heuristics, giving their worst cases. A metaheuristic for this problem is proposed by Komaki et al. (2017). Regarding the total completion time, an MILP model for the $DPm \rightarrow F2|ST_{sd}|\sum C_j$ problem is proposed by Andres and Hatami (2011).

Several multi-objective problems have been addressed in this layout, including:

- The problem $DPm \rightarrow F2|ST_{sd}|F_l(\sum C_j, T_{max})$ is addressed by Hatami et al. (2010) by means of a mathematical model and metaheuristics.
- The problem $DPm \rightarrow F2|a_i, pmtn-semi|F_l(\sum C_j, \sum T_j, \sum E_j)$ is tackled by Shoaardebili and Fattahi (2015), who present multi-objective metaheuristics.
- The problem $DPm \rightarrow F2|ST_{sd}|\#(\sum C_j, \sum T_j)$ is addressed by Cunha Campos and Claudio Arroyo (2014), who propose a multi-objective metaheuristic (genetic algorithm).
- The problem $DPm \rightarrow F2|\#(C_{max}, F_l(\sum w_j E_j, \sum w_j T_j))$ is addressed by Tajbakhsh et al. (2014). The authors present a mathematical model and two metaheuristics.
- Maleki-Daronkolaei and Seyedi (2013); Maleki-Daroukolaei et al. (2012) study the $DPm \rightarrow F2|blocking, ST_{sd}|F_l(C_{max}, \sum w_j C_j)$ problem, proposing several metaheuristics.
- Wang et al. (2016) propose some metaheuristics for the $DPm \rightarrow F2|p-batch|F_l(AT, DC)$ problem, where AT is a function of the arrival times to the customers, and DC the delay costs.

Regarding the problems with hybrid flowshop in the assembly stage, this layout –depicted in Figure 1e– can be seen as a generalisation of the $DPm \rightarrow Fm$ layout. It can also be employed to model a particular manufacturing stage denoted as *product differentiation* which is related to specific operations carried out

after the assembly of the product. The literature on the so-called Differentiation Scheduling Problem (DSP) is relatively abundant, although this problem is usually treated in isolation of the previous assembly problem and, to the best of our knowledge, the only reference addressing scheduling problems in this layout is Xiong et al. (2015), who propose an MILP model as well as some constructive heuristics and metaheuristics.

5 Assembly scheduling with flexible machines

In this section we review the problems involving machines capable of performing multiple operations. Depending on the layout, a solution would be given by solving an integrated assignment and scheduling problem. As in Section 4, in the next subsections we discuss a number of cases depending on the layout for the operations prior to the assembly. Furthermore, we present some results not provided in the literature in order to contribute to the classification of these types of problems.

5.1 Single machine prior to assembly: $1 \rightarrow \alpha_2$ layout

When a single machine is used for operations prior to the assembly, the possibility of performing these operations in a concurrent manner is not available, so they all have to be sequenced on this machine. Depending on the number of machines used for assembly in the second stage, some results can be stated regarding the properties of different scheduling problems. Therefore, we first present the general results.

Theorem 5.1. *In a $1 \rightarrow 1$ layout, for a given sequence of jobs $\Pi := (\pi_1, \dots, \pi_n)$ in the second (assembly) stage, the best value of any regular objective function $f(C_j)$ is obtained if all operations in the first stage of each job in Π are carried out consecutively.*

Proof. Without loss of generality, we assume the following sequence of jobs on the second machine: $(1, 2, \dots, j, \dots, n)$. Let us assume that on the first machine, not all r_j operations of job j are carried out one after another, but the l -th operation of job $j+1$ is carried out after the $k-1$ -th operation of job j and before the k -th operation of job j , i.e. the sequence of operations on the first machine is $\dots, O_{k-1,j}^1, O_{l,j+1}^1, O_{k,j}^1, \dots$. We prove the theorem by contradiction, showing that it is not possible to obtain lower values of the completion times of j and $j+1$ than those obtained if all operations of j are carried out one after another (i.e. in a batch).

Let S_1 be the starting time of the first operation of j in the first stage. Then $C_{j,1}$ the completion time of all operations of job j on the first machine is:

$$C_{j,1} = S_1 + \sum_{u < k} p(O_{uj}^1) + p(O_{l,j+1}^1) + \sum_{u \geq k} p(O_{uj}^1) = S_1 + \sum_{u=1}^{r_j} p(O_{uj}^1) + p(O_{l,j+1}^1)$$

And $C_{j+1,1}$ the completion time of all operations of job $j + 1$ on the first machine:

$$C_{j+1,1} = C_{j,1} + \sum_{u \neq l} p(O_{u,j+1}^1) = S_1 + \sum_{u=1}^{r_j} p(O_{uj}^1) + \sum_{u=1}^{r_{j+1}} p(O_{u,j+1}^1)$$

Regarding the second machine:

$$C_j = C_{j,2} = \max\{S_1 + p(\bar{O}_{j-1}^2); C_{j,1}\} + p(\bar{O}_j^2) = S_1 + \max\{p(\bar{O}_{j-1}^2); \sum_{u=1}^{r_j} p(O_{uj}^1) + p(O_{l,j+1}^1)\} + p(\bar{O}_j^2)$$

And, for job $j + 1$:

$$C_{j+1} = C_{j+1,2} = \max\{C_j; C_{j+1,1}\} + p(\bar{O}_{j+1}^2)$$

In contrast, if all operations of job j are carried out in a batch and are immediately followed by the operations of job $j + 1$, then on the first machine we have:

$$C_{j,1}^* = S_1 + \sum_{u=1}^{r_j} p(O_{uj}^1)$$

and

$$C_{j+1,1}^* = C_{j,1}^* + \sum_{u=1}^{r_{j+1}} p(O_{u,j+1}^1) = S_1 + \sum_{u=1}^{r_j} p(O_{uj}^1) + \sum_{u=1}^{r_{j+1}} p(O_{u,j+1}^1) = C_{j+1,1}$$

The completion times on the second machine would be:

$$C_j^* = C_{j,2}^* = \max\{S_1 + p(\bar{O}_{j-1}^2); C_{j,1}^*\} + p(\bar{O}_j^2) = S_1 + \max\{p(\bar{O}_{j-1}^2); \sum_{u=1}^{r_j} p(O_{uj}^1)\} + p(\bar{O}_j^2)$$

and

$$C_{j+1}^* = C_{j+1,2}^* = \max\{C_j^*; C_{j+1,1}^*\} + p(\bar{O}_{j+1}^2) = \max\{C_j^*; C_{j+1,1}\} + p(\bar{O}_{j+1}^2)$$

As it can be seen, $C_j^* \leq C_j$ (indeed $C_j^* < C_j$ unless operation l of job $j + 1$ in the first stage takes zero time units to complete, i.e. $p(O_{l,j+1}^1) = 0$), and, as a consequence, $C_{j+1}^* \leq C_{j+1}$. In addition, the completion times of the rest of the jobs do not change, therefore any regular objective function would yield a lower value for C^* than for C . This proves that there is no advantage in considering not to carry out all operations of the same job in a batch at the first stage. □

From Theorem 5.1, two corollaries follow:

Corollary 5.1.1. *For any regular objective $f(C_j)$, the $1 \rightarrow 1||f(C_j)$ problem is equivalent to the $F2||f(C_j)$ problem, where p_{jk} the processing times of job j on k machine of the latter problem are given by:*

$$p_{j1} = \sum_{u=1}^{r_j} p(O_{uj}^1)$$

and

$$p_{j2} = p(\bar{O}_{1j}^2)$$

Proof. Trivial from Theorem 5.1. □

Corollary 5.1.2. *For any regular objective $f(C_j)$, the $1 \rightarrow 0||f(C_j)$ problem is equivalent to the $1||f(C_j)$ problem, where p_j the processing times of each job in the latter problem are given by:*

$$p_j = \sum_{u=1}^{r_j} p(O_{uj}^1)$$

Proof. Trivial from Theorem 5.1. □

The corollaries also serve to establish the connection between the flexible and dedicated machines assembly settings, as it can be seen in Figure 4. We are not aware of contributions explicitly addressing the results in the theorem and corollaries. However, some specific cases have been mentioned regarding Corollary 5.1.1: Cheng and Wang (1999) state that, for the makespan objective and the assembly scheduling problem with two components for each job, the problem can be solved by Johnson's algorithm if there is no commonality of the components. Also Property 2 in Liao et al. (2015) states that the $1 \rightarrow 1|BS_j|C_{max}$ problem is equivalent to $F2||C_{max}$ if setup times are zero.

Due to the high number of works analysing the $1 \rightarrow \alpha_2$ setting, the existing contributions have been divided depending on the layout of the assembly stages. These are discussed in the next sections and are summarised in Table 4.

5.1.1 No assembly stage: $1 \rightarrow 0$ layout

As it can be seen from Figure 4, many standard settings in the $1 \rightarrow 0$ layout are polynomially solvable as it is equivalent to the single machine case if a regular objective is considered and no additional constraints are assumed. Therefore, existing contributions focus on problems with specific constraints, most of them related to the existence of the requirement of batch-processing certain types of components, which excludes the possibility of carrying out all operations of a job consecutively and therefore Theorem 5.1 may not be applicable.

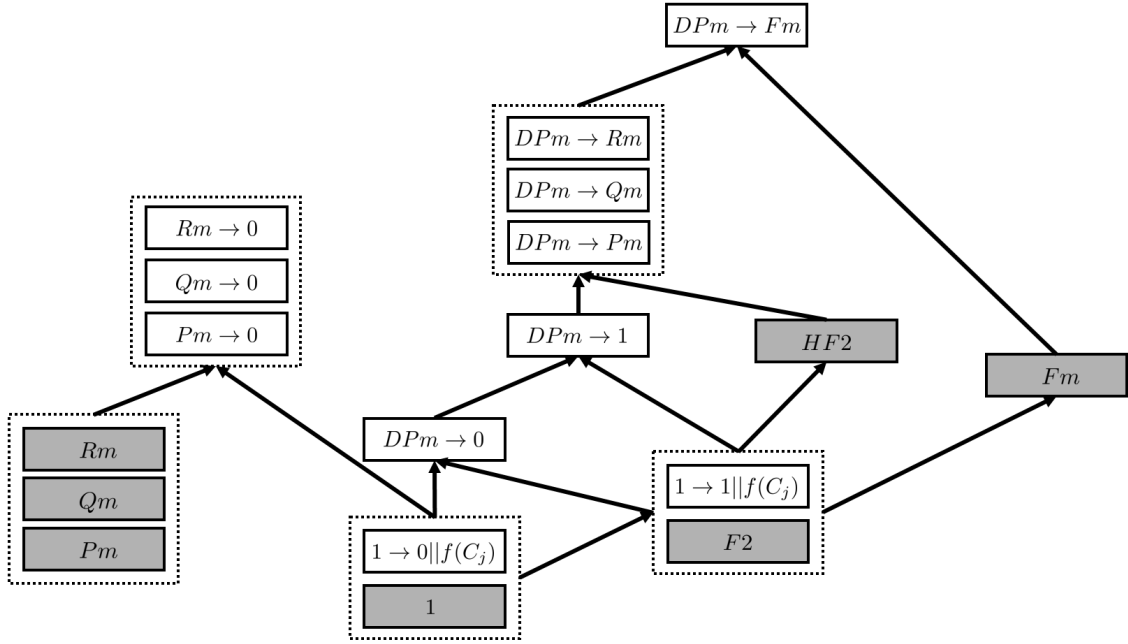


Figure 4: Complexity depending on the layout (non-assembly layouts in grey boxes).

To the best of our knowledge, the first reference on this layout is Liao and Chuang (1996), who consider the problem with setup times and an ϵ -constrained approach where the objectives are the minimisation of the set-up times subject to a maximum number of tardy jobs, or a maximum tardiness. A number of problem properties and B & B algorithms are presented.

A specific case of this layout is addressed by Gerodimos et al. (2000): In their problem, each job is composed of two components (standard and customised), with the standard component being processed in batches. There are setup times for changing from a standard component to a customised one, and between customised components of different products. Furthermore, there is a batch availability constraint. The authors show this setting to be NP-hard for the number of late jobs, a problem for which they develop dynamic programming procedures. A version with more than two components is presented in Gerodimos et al. (1999), where the complexity of the problem for maximum lateness and for the weighted sum of tardy jobs are established. Additionally, in the case where the operations of jobs belonging to the same family have to be sequenced together –a constraint that the authors label as *group technology*– is studied.

Another specific problem within this layout is presented in Shi et al. (2018), where orders composed of several products belonging to different product families have to be processed in batches and there are incompatible job families, i.e. products from different families cannot be in the same batch. The problem is shown to be solvable for the makespan case, whereas for the weighted total completion time the authors

Problem*	Reference	Analysis	Exact	Approximate
$1 \rightarrow 0 BS_{sd} \epsilon(TS, T_{max})$	Liao and Chuang (1996)	PP	BB	
$1 \rightarrow 0 BS_{sd} \epsilon(TS, \sum U_j)$	Liao and Chuang (1996)	PP	BB	
$1 \rightarrow 0 BS, ba, nc = 2 L_{max}$	Gerodimos et al. (2000)	PP	DP	
$1 \rightarrow 0 BS, ba, nc = 2 \sum U_j$	Gerodimos et al. (2000)	PP, CA	DP	
$1 \rightarrow 0 BS L_{max}$	Gerodimos et al. (1999)	PP, CA		
$1 \rightarrow 0 BS \sum w_j U_j$	Gerodimos et al. (1999)	PP, CA	DP	
$1 \rightarrow 0 BS \sum C_j$	Ng et al. (2002)	CA		
	Erel and Ghosh (2007)	CA	DP	
$1 \rightarrow 0 BS, group - technology \sum C_j$	Gerodimos et al. (1999)	PP, CA	DP	
	Ng et al. (2002)	CA		
	Hazir et al. (2008)		MILP	MH
$1 \rightarrow 0 BS GP(C_{max}, TCC)$	Gupta et al. (1997)	PP, CA	PA	
$1 \rightarrow 0 BS GP(TCC, C_{max})$	Gupta et al. (1997)	PP, CA	PA	
$1 \rightarrow 0 p - batch, incompt C_{max}$	Shi et al. (2018)	PP	PA	
$1 \rightarrow 0 p - batch, incompt \sum w_j C_j$	Shi et al. (2018)	PP		CH
$1 \rightarrow 1 n = 1, recrc - sas C_{max}$	Kusiak (1989)	PP, CA		
$1 \rightarrow 1 n = 1, recrc - sas, product - batch C_{max}$	Kusiak (1989)			CH
$1 \rightarrow 1 recrc - sas C_{max}$	Kusiak (1989)	PP, CA		
$1 \rightarrow 1 recrc - sas, product - batch C_{max}$	Kusiak (1989)			CH
$1 \rightarrow 1 BS, ba, nc = 2 C_{max}$	Cheng and Wang (1999)	PP, CA	DP (special cases)	
	Lin and Cheng (2002)	CA		
$1 \rightarrow 1 BS, nc = 2 C_{max}$	Cheng and Wang (1999)	PP, CA	DP (special cases)	
	Lin and Cheng (2002)	CA		
$1 \rightarrow 1 deteriorating, BS C_{max}$	Jung and Kim (2016)		MILP	
$1 \rightarrow 1 pm, deteriorating, BS C_{max}$	Jung and Kim (2016)		MILP	
$1 \rightarrow 1 BS C_{max}$	Liao et al. (2015)		MILP	CH
$1 \rightarrow 1 BS C_{max}$	Jung et al. (2017)		MILP	MH
$1 \rightarrow 1 BS_{component}, product - batch \sum C_j$	Yokoyama (2004)	PP		CH
$1 \rightarrow Pm n = 1, recrc - sas C_{max}$	He and Babayan (2002)	CA	MILP	
$1 \rightarrow Pm complex C_{max}$	He and Babayan (2002)		MILP	CH
	Gaafar and Masoud (2005)			MH
	Liao and Liao (2008)		BB	MH
	Gaafar et al. (2008)			MH
	Teymourian et al. (2016)			MH

Table 4: Summary of contributions in Section 5.1 (single machine processing).(*Notation is described in the additional on-line material).

propose a constructive heuristic.

The NP-hardness of the problem with total completion time assuming setup times among families of products is established by Ng et al. (2002) and Erel and Ghosh (2007) independently. The latter authors also propose a dynamic programming algorithm. Finally, Hazir et al. (2008) propose an MILP and several metaheuristics for this problem.

The problems $1 \rightarrow 0|BS_j|GP(C_{max}, TCC)$ and $1 \rightarrow 0|BS_j|GP(TCC, C_{max})$ where TCC denotes the total carrying costs are addressed by Gupta et al. (1997), who study their complexity, derive some properties and propose some constructive heuristics.

A specific case of this problem that will not be treated here is given by the so-called *multiple orders per job* scheduling problem (see e.g. Laub et al., 2007). This problem arises in the semiconductor manufacturing and TFT-LCD industries, where a number of orders –possibly belonging to different customers– should be grouped together in special transfer facilities (FOUPs) to be processed as a batch (either *p-batch* or *s-batch*). Therefore, the problems of simultaneously grouping the orders into FOUPs and scheduling them have to be

addressed. Although this problem is a particular case of the order scheduling problem, in view of its very specific features, it is not addressed here and the reader is referred to e.g Mason and Chen (2010) for a recent paper on the topic.

5.1.2 Single machine for assembly: $1 \rightarrow 1$ layout

All except one of the references involve the makespan as objective (see Table 4). To the best of our knowledge, the first reference on this layout is Kusiak (1989), who addresses the problems of scheduling a single product, and several products with or without batches. The jobs may require several steps of manufacturing and assembly, which can be modelled with recirculation constraints, i.e.: the $1 \rightarrow 1|recrc - sas|C_{max}$ or $1 \rightarrow 1|recrc - sas, product - batch|C_{max}$ problems. For the problem without batches, Kusiak (1989) develops an optimal algorithm. In the light of Theorem 5.1, this algorithm is an extension for the recirculation case of the well-known Johnson’s algorithm for the $F2||C_{max}$ problem, although we are not aware that this aspect is mentioned by the author. For the problems with batches, the author proposes some constructive heuristics.

Cheng and Wang (1999) address the scheduling of jobs consisting of two components, a unique component for each product and a common component for all products, the latter being produced in batches with a setup time. The authors show that the problem is NP-hard, and identify properties to reduce the search space, as well as specific cases of the problem that are polynomially solvable. Lin and Cheng (2002) develop further results on the complexity of these problems and prove that they are strongly NP-hard.

Jung and Kim (2016) propose MILP models and constructive heuristics for the problems with makespan objective with/without preventive maintenance activities, batch setups and deterioration rates.

Liao et al. (2015) propose an MILP model and some properties for the problem based on Johnson’s algorithm for the $1 \rightarrow 1|BS|C_{max}$ problem, also addressed by Jung et al. (2017) when the number of components of the jobs is dynamic, constant setup times, and product batches. An MILP model and a genetic algorithm are proposed.

Finally, Yokoyama (2004) addresses the problem with completion time as objective in the case of setups in the manufacturing machine, i.e. the $1 \rightarrow 1|BS_{component, product - batch}|\sum C_j$ problem. This author analyses some properties of the problem and proposes a constructive heuristic.

5.1.3 Parallel machines for assembly: $1 \rightarrow Pm$ layout

Note that, for this layout, it is not possible to establish that carrying out all operations of a job consecutively in the first machine will produce better schedules for a regular objective function. We are only aware of papers studying the makespan objective in this layout (see Table 4). This problem is NP-hard (see Figure 4 and note that $Pm||C_{max}$ is already NP-hard). He and Babayan (2002) are the first to analyse the problem

of single and n -job scheduling with and without multiple assembly operations (which involves recirculation and precedence constraints). They show that the single-job problem with simple assembly is polynomially solvable, and show some equivalences among the other problems, proposing an MILP model, a lower bound, and some constructive heuristics.

Some metaheuristics for this problem are proposed by Gaafar and Masoud (2005), which are outperformed by Liao and Liao (2008). These authors also propose a B & B algorithm for the problem. Independently, Gaafar et al. (2008) and Teymourian et al. (2016) propose additional metaheuristics. It is relevant to mention that only partial comparisons among these metaheuristics have been carried out, so the state-of-the-art regarding approximate methods for the problem is unclear.

5.2 Parallel machines prior to assembly: $Pm/Rm/Qm \rightarrow \alpha_2$ layout

In view of the existing contributions (summarised in Table 5), we discuss the cases where there is no processing at the assembly stage in Section 5.2.1, whereas the rest of the cases are discussed in Section 5.2.2.

Problem*	Reference	Analysis	Exact	Approximate
$P2 \rightarrow 0 \sum C_j$	Yang (2005)	CA		
	Yang and Posner (2005)	PP		CH
$Pm \rightarrow 0 p - batch, incompt C_{max}$	Shi et al. (2018)	PP		CH
$Pm \rightarrow 0 \sum C_j$	Blocher and Chhajer (1996)	PP, CA	MILP	CH
	Yang (2005)	CA		
	Yang and Posner (2005)	PP		CH
$Pm \rightarrow 0 p_j = 1 \sum C_j$	Blocher and Chhajer (1996)	CA		
	Yang (2005)	CA		
$Pm \rightarrow 0 p_j = 1 L_{max}$	Yang (2005)	CA		
$Pm \rightarrow 0 p_j = 1 \sum w_j C_j$	Yang (2005)	CA		
$Pm \rightarrow 0 \sum w_j C_j$	Leung et al. (2006a)			CH
	Leung et al. (2008b)			CH
$Pm \rightarrow 0 prmt \sum w_j C_j$	Leung et al. (2008a)	CA		CH
$Pm \rightarrow 0 p - batch, incompt \sum w_j C_j$	Shi et al. (2018)		MILP	MH
$Pm \rightarrow 0 L_{max}$	Su et al. (2013)			CH
$Rm \rightarrow 0 split \sum C_j$	Xu et al. (2015)	PP	MILP	CH
$Qm \rightarrow 0 \sum w_j C_j$	Leung et al. (2007b)			CH
$Qm \rightarrow 0 prmt \sum w_j C_j$	Leung et al. (2007b)			CH
$Rm \rightarrow 0 d_j = d \sum U_j$	Ng et al. (2003)	CA		
$Rm \rightarrow 0 d_j = d \sum w_j U_j$	Ng et al. (2003)	CA		CH
$Pm \rightarrow 1 recrc C_{max}$	He et al. (2001)		MILP	
$Pm \rightarrow 1 recrc - complex C_{max}$			MILP	CH
$Rm \rightarrow 1 BS_{sd} C_{max}$	Kheirandish et al. (2015)		MILP	MH
$Qm \rightarrow 1 C_{max}$	Koulamas and Kyparisis (2007)	PP		CH
$Pm \rightarrow Pm F_l(JIT, \text{mean longest waiting duration})$	Chen et al. (2015)		MILP	MH
$Rm \rightarrow DF2 ST_{sd} F_l(\sum w_j E_j^2, \sum w_j T_j^2, \sum U_j, C_{max})$	Dalfard et al. (2011)		MILP	MH

Table 5: Summary of contributions in Section 5.2 (parallel flexible machines). (*Notation is described in the additional on-line material).

5.2.1 Flexible Order scheduling: $Pm/Rm/Qm \rightarrow 0$ layout

The complexity of a number of problems in the $Pm \rightarrow 0$ layout is established by Yang (2005), some of them with specific constraints (see Table 5). His results prove that most of the problems are NP-hard even for

2 machines and unit processing times, with the exception of the maximum lateness and sum of completion times for unit processing times.

The first reference for the problem with total completion time objective is by Blocher and Chhajed (1996), who present some properties of the problem (such as an optimal procedure for the $p_j = 1$ case), an MILP model, and propose constructive heuristics. Yang and Posner (2005) present some heuristics for the $Pm \rightarrow 0 || \sum_j C_j$ problem, together with their worst case bounds. Xu et al. (2015) address the $Rm \rightarrow 0 || \sum C_j$ problem where each component can be split arbitrarily into continuous sublots. Several problem properties and heuristics with their worst case bounds are presented. Leung et al. (2007b) address the $Qm \rightarrow 0 || \sum w_j C_j$ with and without preemption, proposing some heuristics and giving their worst case. Leung et al. (2006a, 2008b) propose several heuristics for the $Pm \rightarrow 0 || \sum_j w_j C_j$. Leung et al. (2008a) study the same problem where preemption is allowed. The complexity of the problem is analysed, and some heuristics are presented.

For the specific problem of parallel batch machines with incompatible product families, Shi et al. (2018) study the objectives of makespan and weighted total completion time minimisation. A constructive heuristic based on some problem properties is proposed for the first objective, whereas a MILP model and a metaheuristic are presented for the second.

Regarding due-date related objectives, Su et al. (2013) address the problem for the maximum lateness, presenting a set of constructive heuristics together with their worst case lower bounds. Ng et al. (2003) address the $Pm \rightarrow 0 || \sum w_j U_j$ problem when $d_j = d$ and $p_{ij} \in \{0, 1\}$. They show that the problem is NP-hard even when $w_j = 1$. They propose an approximate algorithm with its worst case lower bound.

5.2.2 One or more machines for assembly: $Pm/Rm/Qm \rightarrow \alpha_2$ layout ($\alpha_2 \neq 0$)

The contributions on this layout are rather scarce as compared with other problem types (see summary in Table 5). Regarding the case with a single machine for assembly, the $Pm \rightarrow 1|recrc|C_{max}$ problem has been first addressed by He et al. (2001), who analyse the cases of single and multiple job product with simple/complex assembly. They propose MILP models, some equivalence properties, and constructive heuristics. Without recirculation, Kheirandish et al. (2015) address the $Rm \rightarrow 1|BS_{sd, product - batch}|C_{max}$ problem, for which the authors formulate an MILP model and a metaheuristic. Finally, an asymptotically optimal heuristic for the $Qm \rightarrow 1||C_{max}$ problem is given by Koulamas and Kyparisis (2007).

With respect to the case where the assembly takes place in parallel machines, Chen et al. (2015) propose an MILP model and a metaheuristic for the case where orders are composed of several jobs (which in turn are composed of several components), with the bi-objective of the weighted mean tardiness and earliness, and the mean longest waiting time of the order.

Dalfard et al. (2011) address the $Rm \rightarrow DF2|ST_{sd}|F_l(\sum w_j T_j^2, \sum w_j E_j^2, C_{max}, \sum U_j)$ problem. An MILP

model and metaheuristics are proposed for the problem.

5.3 Flowshop-type prior to assembly: $Fm/PFm/HFm \rightarrow \alpha_2$ layout

Problem*	Reference	Analysis	Exact	Approximate
$F2 \rightarrow 1 \sum w_j C_j$	Yokoyama and Santos (2005)	PP	BB	
$F2 \rightarrow 1 prmu \sum w_j C_j$	Yokoyama and Santos (2005)	PP	BB	
$Fm \rightarrow 1 recrc - complex, prmu, ba \sum w_j C_j$	Yokoyama (2001)		BB	
$Fm \rightarrow 1 prmu, product - batch, BS_{ij} \sum C_j$	Yokoyama (2008)	PP, LB	BB	CH
$PFm \rightarrow 1 prmu C_{max}$	Hatami et al. (2013)		MILP	MH
	Hatami et al. (2014b)			CH
	Li et al. (2015)			MH
$PFm \rightarrow 1 prmu, ST_{sd} C_{max}$	Hatami et al. (2014a)			CH
	Hatami et al. (2015b)			CH, MH
$HF2, Pm, Pm \rightarrow 1 C_{max}$	Fattahi et al. (2013b)	PP	MILP	CH
	Shan et al. (2014)		MILP	
	Komaki et al. (2016)	PP	MILP	CH, MH
$HF2, 1, Pm \rightarrow Pm BS_{ij}, product - batch C_{max}$	Fattahi et al. (2013a)	PP		MH
$HF2, Pm, Pm \rightarrow 1 BS_{ij} C_{max}$	Fattahi et al. (2014)		BB	MH
$HF2, Qm, 1 \rightarrow DP2 ST_{sd}, product - batch C_{max}$	Nikzad et al. (2015)		MILP	MH
$HFm, Pm_1 \dots Pm_S \rightarrow Pm_A recrc - complex C_{max}$	Mahdavi et al. (2011)		MILP	CH, MH

Table 6: Summary of contributions in Section 5.3 (flowshop-type layout prior to assembly). (*Notation is described in the additional on-line material).

This layout corresponds to Figure 2d in the case of a flowshop, to Figure 2e in the case of a parallel flowshop, and to Figure 2f in the case of a hybrid flowshop. These types of problems are NP-complete for any objective even for the $F2 \rightarrow 1$ layout, as the latter can be seen as a generalisation of the three-machine flowshop (see Figure 4). In Table 6 we summarise the main contributions.

Regarding the flowshop layout in the α_1 field, to the best of our knowledge, the problem has been addressed only by Yokoyama (2001), Yokoyama and Santos (2005) and Yokoyama (2008). In Yokoyama and Santos (2005), the authors propose a B & B procedure for the $F2 \rightarrow 1 || \sum w_j C_j$ problem with and without the permutation constraint. Yokoyama (2008) addresses the $Fm \rightarrow 1 |prmu, product - batch, BS_{ij}| \sum C_j$ problem, deriving some approximate procedures and lower bounds. Finally, Yokoyama (2001) addresses a specific case of $Fm \rightarrow 1 |recrc - complex, prmu, ba| \sum w_j C_j$ using a B & B procedure.

Regarding the case of the parallel flowshop in the α_1 field, perhaps the best-known case is the $PFm \rightarrow 1$ layout, which is also called *distributed assembly scheduling problem*. The problem has been addressed just for the makespan objective, Hatami et al. (2013) and Hatami et al. (2014b) being the first works describing the problem and proposing an MILP and a metaheuristic, and constructive heuristics respectively. Li et al. (2015) address the same problem and propose a genetic algorithm that compares favourably to the constructive heuristics by Hatami et al. (2014b). Finally, constructive heuristics and metaheuristics for the $PFm \rightarrow 1 |prmu, ST_{sd}| C_{max}$ are proposed in Hatami et al. (2014a) and in Hatami et al. (2015b) respectively.

Finally, regarding the hybrid flowshop, to the best of our knowledge, it has been addressed first by Fattahi et al. (2013b), who tackle the $HF2, Pm_1, \dots, Pm_S \rightarrow 1 || C_{max}$ problem. These authors propose an

MILP model and constructive heuristics. However, the MILP model contains a mistake which is fixed by Shan et al. (2014). The heuristics proposed by Fattahi et al. (2013b) are outperformed by a metaheuristic proposed by Komaki et al. (2016), who also propose some constructive heuristics for the problem. Two extensions of the previous problem are presented by Fattahi et al. (2013a) and Fattahi et al. (2014). The first reference proposes metaheuristics for the case with machine dependent setups and product-batch restrictions, whereas in the second reference a branch and bound and a metaheuristic are proposed for the problem with machine dependent setups. The $HF2, Qm, 1 \rightarrow DP2|ST_{sd}, product - batch|C_{max}$ problem is studied by Nikzad et al. (2015), who propose an MILP model and several metaheuristics. Finally, a multi-level assembly case is addressed by Mahdavi et al. (2011), where the authors process the components of a product in a hybrid flowshop followed by several stages of subassemblies and assemblies with the objective of makespan minimisation. This problem can be regarded as $HFm \rightarrow 1|recrc|C_{max}$ to model the recirculation required in the assembly stage. The problem is modelled using an MILP, and some heuristics and metaheuristics are proposed.

6 Distributed assembly scheduling

In this section we review the contributions related to scheduling jobs using different assembly layouts. Note that here distributed has a different meaning than in the so-called distributed assembly flowshop layout, as in the latter case the assembly operation is not distributed at all.

Therefore, in this problem there are several identical f factories, each one consisting of an assembly layout, so the problem is to schedule all jobs in these settings, which includes the assignment of jobs to the factories and the scheduling of these jobs within the factories. Using the notation proposed in Section 3, the layout of this type of problem can be denoted as $(\alpha_1 \rightarrow \alpha_2) \rightarrow 0$, where the term within parentheses indicates the layout of each identical factory.

To the best of our knowledge, the only type of problem addressed is to schedule jobs over distributed factories consisting of dedicated parallel machines and a single assembly stage. More specifically, the $(DPm \rightarrow 1) \rightarrow 0$ layout has been addressed for the following objectives:

- Makespan: This objective was first addressed by Deng et al. (2016), proposing an MILP model for the problem and a metaheuristic (Memetic algorithm).
- Total Completion Time. This problem is addressed by Xiong et al. (2014), who present several metaheuristics for the problem with setups, i.e. the $(DPm \rightarrow 1) \rightarrow 0|BS_j|\sum C_j$ problem.
- Biobjective: Xiong and Xing (2014) address the $(DPm \rightarrow 1) \rightarrow 0|BS_j|F_l(C_{max}, \sum C_j)$ problem, proposing a genetic algorithm and a variable local search. However, these algorithms are not compared

with those of the corresponding single-criteria cases.

Regarding the case with unrelated machines in each factory, Hatami et al. (2015a) and Hatami et al. (2016) propose a MILP model for the problem with eligibility constraints, i.e. for the $(DRm \rightarrow 1) \rightarrow 0|M_j|C_{max}$ problem.

7 General layout (job-shop) assembly models

In this section, we deal with the contributions that solve a general (job-shop) problem, in many cases with multiple assemblies and sub-assemblies. To the best of our knowledge, the first of these contributions is Agrawal et al. (1996), who propose an MILP model and some heuristics. This model is enhanced by Roach and Nagi (1996) in order to consider ordering constraints. The model considered by these authors assumes a single product, although ordered with different quantities and due dates. For this problem, a hybrid simulated annealing-genetic algorithm is proposed by Roach and Nagi (1996)

Cummings Mckoy and Egbelu (1998) propose another general MILP model for a job shop with different products (each one with a different structure of assembly/sub-assemblies operations) and setups, with the objective of makespan minimisation. The variables of the model contain 7 sub indices, which serve as an indicator of the complexity of the model. A heuristic approach to find approximate solutions is also given.

A specific model including availability, resources, and assembly area constraints is presented in Kolisch and Hess (2000) with the objective of minimisation of the weighted tardiness of the orders. A genetic algorithm for the generic assembly problem is presented by Pongcharoen et al. (2002), taking into account penalties for earliness and for tardiness. Also Du et al. (2016) propose a metaheuristic for a general-layout assembly problem, although only a sample instance is solved.

Kyparisis and Koulamas (2002) address a general problem consisting of r -serial stages, with stage k containing m_k machines in charge of carrying out the operations of the jobs. At stage k , a given operation i can take place in a subset of l_i^k machines. The job cannot pass on to the following stage until all operations in the previous stage have been completed. It can be seen that this model encompasses several models presented in this paper, although it cannot accommodate for instance the $PFm \rightarrow 1$ layout, or the distributed layout. The authors address the problem for the makespan objective and propose a general constructive heuristic for the problem.

In Bhongade and Khodke (2012) two constructive heuristics for the problem with flexible machines and makespan as objective are presented. Two versions of genetic algorithms are proposed by Wang et al. (2011) for the general assembly problem with makespan objective.

Finally, it is worth mentioning that a number of papers have addressed the problem of lot streaming in

the general flowshop setting, including Chan et al. (2008b), Chan et al. (2008a), and Wong et al. (2009). Furthermore, Liu (2009) shows that lot streaming techniques are also beneficial in the $Jm \rightarrow 0$ layout.

8 Conclusions and future research lines

In this paper, we have reviewed a large number of papers in order to provide a comprehensive overview on scheduling with assembly operations. A number of conclusions and directions for future research can be derived:

- Regarding the assembly scheduling problems with dedicated machines:
 - The $DPm \rightarrow 1$ seems to be a well-studied layout for different objectives and constraints. Nevertheless, since this is a key layout, in order to tackle more complex settings, some results would need to be reinforced along the following directions:
 - * The $DP2 \rightarrow 1|r_j|C_{max}$ and the $DPm \rightarrow 1|r_j|C_{max}$ problems have been independently studied by Sung and Juhn (2009) and Komaki and Kayvanfar (2015) respectively and the methods proposed have not been contrasted.
 - * Although there are several metaheuristics available for the $DPm \rightarrow 1||F_l(C_{max}, \sum C_j)$ problem (see Section 4.2), most of them have not been compared among each other, therefore the state-of-the-art remains unclear. Furthermore, it would be of interest if these heuristics are competitive for the single objective cases (i.e. considering only makespan or completion time as objectives).
 - * When addressing the multicriteria problem, the predominant approach adopted has been to weight the objective functions, and we are only aware of one paper discussing ϵ -constrained or Pareto-set approaches for rather specific constraints (Seidgar et al., 2016).
 - Although the $DPm \rightarrow 1$ layout has been widely studied with respect to a number of different objectives, the numerous algorithms available have not been extended to the $DPm \rightarrow Pm$ counterpart.
 - The $DPm \rightarrow \alpha_2$ layout can be regarded as a modelling simplification of more complex, dedicated layouts where the operations prior to the assembly are carried out. We have witnessed that a frequent layout would consist of dedicated flowshops that process the components of the product to be later assembled, i.e. the $DPFm \rightarrow \alpha_2$ layout. However, we are not aware of papers dealing with these scheduling problems.

- Several works on the $DPm \rightarrow Rm/Qm/Pm$ layouts involve the development of MILP models. However, to the best of our knowledge, there are no computational results regarding their (relative) efficiency.
- There are a number of contributions for the multi-criteria assembly scheduling problem in the $DPm \rightarrow Fm$ layout, most of them consisting of metaheuristics for the multicriteria weighted approach. It would be interesting to 1) establish whether some of these metaheuristics perform efficiently for other multi-criteria problems in this layout, and 2) assess their performance for the extreme (i.e. single-criteria) case.
- Regarding the assembly scheduling with flexible machines:
 - Although there are some contributions on the $1 \rightarrow 1$ layout, its connection with the 2-machine flowshop scheduling problem has not been explicitly stated up to Section 5.1.
 - The connection among order scheduling problems and flexible order scheduling has not been thoroughly explored. Clearly, the former can be modelled as a special case of the latter, therefore it would be interesting to compare the structure of the solutions and procedures.
 - For the $1 \rightarrow Pm$ layout, only the makespan objective has been considered. Furthermore, for this case, the state of the art of the solution procedures for the problem is unclear.
 - In general, contributions for some layouts within the flexible machine case are scarce, some examples being the $Fm \rightarrow \alpha_2$ and $PFm \rightarrow \alpha_2$ layouts.
- Regarding the distributed assembly scheduling it has to be mentioned that most of the approximate methods employed in the distributed setting do not make use of the results available for the corresponding single factory problem.
- Finally, with respect to the general assembly problems, it would be of interest to compare the relative efficiency of the different MILP models proposed.

As a final comment, the picture emerging from the review carried out in this paper pictures a field with many contributions in the last decades and some well-studied problems, but also with numerous topics worth further research. We hope that the classification and review carried out would serve to foster future work in this relevant area for manufacturing operations.

Acknowledgements

The authors are grateful to the anonymous referees that helped to improve the earlier version of the paper. This research has been funded by the Spanish Ministry of Science and Innovation, under grant “PROMISE”

with reference DPI2016-80750-P.

References

- Agrawal, A., Harhalakis, G., Minis, I., and Nagi, R. (1996). Just-in-time production of large assemblies. *IIE Transactions (Institute of Industrial Engineers)*, 28(8):653–667.
- Ahmadi, R., Bagchi, U., and Roemer, T. A. (2005). Coordinated scheduling of customer orders for quick response. *Naval Research Logistics*, 52(6):493–512.
- Al-Anzi, F. and Allahverdi, A. (2006). A hybrid tabu search heuristic for the two-stage assembly scheduling problem. *International Journal of Operations Research*, 3(2):109–119.
- Al-Anzi, F. and Allahverdi, A. (2007a). Computer assembly scheduling problem. In *Proceedings-ICSSSM'06: 2006 International Conference on Service Systems and Service Management*, volume 2, pages 1188–1191.
- Al-Anzi, F. and Allahverdi, A. (2007b). A self-adaptive differential evolution heuristic for two-stage assembly scheduling problem to minimize maximum lateness with setup times. *European Journal of Operational Research*, 182(1):80–94.
- Al-Anzi, F. and Allahverdi, A. (2009). Heuristics for a two-stage assembly flowshop with bicriteria of maximum lateness and makespan. *Computers and Operations Research*, 36(9):2682–2689.
- Al-Anzi, F. and Allahverdi, A. (2012). Better heuristics for a two-stage multi-machine assembly scheduling problem to minimize total completion time. *International Journal of Operations Research*, 9(2):66–75.
- Al-Anzi, F. and Allahverdi, A. (2013). An artificial immune system heuristic for two-stage multi-machine assembly scheduling problem to minimize total completion time. *Journal of Manufacturing Systems*, 32(4):825–830.
- Allahverdi, A. and Al-Anzi, F. (2006a). Evolutionary heuristics and an algorithm for the two-stage assembly scheduling problem to minimize makespan with setup times. *International Journal of Production Research*, 44(22):4713–4735.
- Allahverdi, A. and Al-Anzi, F. (2006b). A PSO and a tabu search heuristics for the assembly scheduling problem of the two-stage distributed database application. *Computers and Operations Research*, 33(4):1056–1080.
- Allahverdi, A. and Al-Anzi, F. (2008a). The two-stage assembly flowshop scheduling problem with bicriteria of makespan and mean completion time. *International Journal of Advanced Manufacturing Technology*, 37(1-2):166–177.
- Allahverdi, A. and Al-Anzi, F. (2008b). Two-stage assembly flowshop scheduling problem with maximum lateness and makespan. In *38th International Conference on Computers and Industrial Engineering 2008*, volume 2, pages 2021–2028.
- Allahverdi, A. and Al-Anzi, F. (2008c). The two-stage assembly flowshop scheduling problem with setup times. In *38th International Conference on Computers and Industrial Engineering 2008*, volume 2, pages 2029–2035.
- Allahverdi, A. and Al-Anzi, F. (2009). The two-stage assembly scheduling problem to minimize total completion time with setup times. *Computers and Operations Research*, 36(10):2740–2747.
- Allahverdi, A. and Al-Anzi, F. (2012). A new heuristic for the queries scheduling problem on distributed database systems to minimize mean completion time. In *Proceedings of the 21st International Conference on Software Engineering and Data Engineering, SEDE2012*, pages 93–97.

- Allahverdi, A., Aydilek, A., and Aydilek, H. (2016a). Minimizing the number of tardy jobs on a two-stage assembly flowshop. *Journal of Industrial and Production Engineering*, 33(6):391–403.
- Allahverdi, A. and Aydilek, H. (2015). The two stage assembly flowshop scheduling problem to minimize total tardiness. *Journal of Intelligent Manufacturing*, 26(2):225–237.
- Allahverdi, A., Aydilek, H., and Aydilek, A. (2016b). Two-stage assembly scheduling problem for minimizing total tardiness with setup times. *Applied Mathematical Modelling*, 40(17-18):7796–7815.
- Andres, C. and Hatami, S. (2011). The three stage assembly permutation flowshop scheduling problem. In *Proceedings of the 5th International Conference on Industrial Engineering and Industrial Management*, pages 867–875.
- Aydilek, A., Aydilek, H., and Allahverdi, A. (2017). Minimising maximum tardiness in assembly flowshops with setup times. *International Journal of Production Research*, 55(24):7541–7565.
- Bhongade, A. and Khodke, P. (2012). Heuristics for production scheduling problem with machining and assembly operations. *International Journal of Industrial Engineering Computations*, 3(2):185–198.
- Blocher, J. and Chhajer, D. (2008). Minimizing customer order lead-time in a two-stage assembly supply chain. *Annals of Operations Research*, 161(1):25–52.
- Blocher, J. D. and Chhajer, D. (1996). The customer order lead-time problem on parallel machines. *Naval Research Logistics*, 43(5):629–654.
- Cai, X. and Zhou, X. (2004). Deterministic and stochastic scheduling with teamwork tasks. *Naval Research Logistics*, 51(6):818–840.
- Chan, F., Wong, T., and Chan, L. (2008a). Lot streaming for product assembly in job shop environment. *Robotics and Computer-Integrated Manufacturing*, 24(3):321–331.
- Chan, F., Wong, T., and Chan, P. (2008b). The application of lot streaming to assembly job shop under resource constraints. In *IFAC Proceedings Volumes(IFAC-Papers Online)*, volume 17.
- Chen, J., Huang, G., Luo, H., and Wang, J. (2015). Synchronisation of production scheduling and shipment in an assembly flowshop. *International Journal of Production Research*, 53(9):2787–2802.
- Cheng, T. and Wang, G. (1999). Scheduling the fabrication and assembly of components in a two-machine flowshop. *IIE Transactions (Institute of Industrial Engineers)*, 31(2):135–143.
- Cummings Mckoy, D. and Egbelu, P. (1998). Minimizing production flow time in a process and assembly job shop. *International Journal of Production Research*, 36(8):2315–2332.
- Cunha Campos, S. and Claudio Arroyo, J. (2014). NSGA-II with iterated greedy for a bi-objective three-stage assembly flowshop scheduling problem. In *GECCO2014-Proceedings of the 2014 Genetic and Evolutionary Computation Conference*, pages 429–436.
- Dalfard, V., Ardakani, A., and Banihashemi, T. (2011). Hybrid genetic algorithm for assembly flow-shop scheduling problem with sequence-dependent setup and transportation times. *Technical Gazette*, 18(4):497–504.
- Deng, J., Wang, L., Wang, S.-Y., and Zheng, X.-L. (2016). A competitive memetic algorithm for the distributed two-stage assembly flow-shop scheduling problem. *International Journal of Production Research*, 54(12):3561–3577.
- Du, H., Liu, D., and Zhang, M.-H. (2016). A hybrid algorithm based on particle swarm optimization and artificial immune for an assembly job shop scheduling problem. *Mathematical Problems in Engineering*, 2016:3938679.

- Erel, E. and Ghosh, J. (2007). Customer order scheduling on a single machine with family setup times: Complexity and algorithms. *Applied Mathematics and Computation*, 185(1):11–18.
- Fattahi, P., Hassan Hosseini, S., and Jolai, F. (2013a). Some heuristics for the hybrid flow shop scheduling problem with setup and assembly operations. *International Journal of Industrial Engineering Computations*, 4(3):393–416.
- Fattahi, P., Hosseini, S., and Jolai, F. (2013b). A mathematical model and extension algorithm for assembly flexible flow shop scheduling problem. *International Journal of Advanced Manufacturing Technology*, 65(5-8):787–802.
- Fattahi, P., Hosseini, S., Jolai, F., and Tavakkoli-Moghaddam, R. (2014). A branch and bound algorithm for hybrid flow shop scheduling problem with setup time and assembly operations. *Applied Mathematical Modelling*, 38(1):119–134.
- Framinan, J., Leisten, R., and Ruiz, R. (2014). *Manufacturing Scheduling Systems: An Integrated View of Models, Methods, and Tools*. Springer.
- Framinan, J. and Perez-Gonzalez, P. (2017a). The 2-stage assembly flowshop scheduling problem with total completion time: Efficient constructive heuristic and metaheuristic. *Computers and Operations Research*, 88:237–246.
- Framinan, J. and Perez-Gonzalez, P. (2017b). New approximate algorithms for the customer order scheduling problem with total completion time objective. *Computers and Operations Research*, 78(1):181–192.
- Framinan, J. and Perez-Gonzalez, P. (2018). Order scheduling with tardiness objective: Improved approximate solutions. *European Journal of Operational Research*, 266(3):840–850.
- Gaafar, L. and Masoud, S. (2005). Genetic algorithms and simulated annealing for scheduling in agile manufacturing. *International Journal of Production Research*, 43(14):3069–3085.
- Gaafar, L., Masoud, S., and Nassef, A. (2008). A particle swarm-based genetic algorithm for scheduling in an agile environment. *Computers and Industrial Engineering*, 55(3):707–720.
- Garg, N., Kumar, A., and Pandit, V. (2007). *Order Scheduling Models: Hardness and Algorithms*, pages 96–107. Springer.
- Gerodimos, A., Glass, C., and Potts, C. (2000). Scheduling the production of two-component jobs on a single machine. *European Journal of Operational Research*, 120(2):250–259.
- Gerodimos, A., Glass, C., Potts, C., and Tautenhahn, T. (1999). Scheduling multi-operation jobs on a single machine. *Annals of Operations Research*, 92:87–105.
- Graham, R. L., Lawler, E. L., Lenstra, J. K., and Rinnooy Kan, A. H. G. (1979). Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics*, 5(2):287–326.
- Gupta, J., Ho, J., and Van Der Veen, J. (1997). Single machine hierarchical scheduling with customer orders and multiple job classes. *Annals of Operations Research*, 70:127–143.
- Hadda, H., Dridi, N., and Hajri-Gabouj, S. (2007). The two-stage assembly flow shop scheduling with an availability constraint. *Multidisciplinary International Scheduling Conference Theory and Applications (MISTA), Paris*.
- Hadda, H., Dridi, N., and Hajri-Gabouj, S. (2014). The two-stage assembly flow shop scheduling with an availability constraint: Worst case analysis. *Journal of Mathematical Modelling and Algorithms in Operations Research*, 13(3):233–245.

- Haouari, M. and Daouas, T. (1999). Optimal scheduling of the 3-machine assembly-type flowshop. *RAIRO Recherche Operationnelle*, 33(4):439–445.
- Hariri, A. and Potts, C. (1997). A branch and bound algorithm for the two-stage assembly scheduling problem. *European Journal of Operational Research*, 103(3):547–556.
- Hatami, S., Ebrahimnejad, S., Tavakkoli-Moghaddam, R., and Maboudian, Y. (2010). Two meta-heuristics for three-stage assembly flowshop scheduling with sequence-dependent setup times. *International Journal of Advanced Manufacturing Technology*, 50(9-12):1153–1164.
- Hatami, S., Ruiz, R., and Andres-Romano, C. (2013). The distributed assembly permutation flowshop scheduling problem. *International Journal of Production Research*, 51(17):5292–5308.
- Hatami, S., Ruiz, R., and Andres-Romano, C. (2014a). Simple constructive heuristics for the distributed assembly permutation flowshop scheduling problem with sequence dependent setup times. In *Proceedings-2014 International Conference on Control, Decision and Information Technologies, CoDIT2014*, pages 19–23.
- Hatami, S., Ruiz, R., and Andres-Romano, C. (2015a). The distributed assembly parallel machine scheduling problem with eligibility constraints. *International Journal of Production Management and Engineering*, 3(1):13–23.
- Hatami, S., Ruiz, R., and Andres-Romano, C. (2015b). Heuristics and metaheuristics for the distributed assembly permutation flowshop scheduling problem with sequence dependent setup times. *International Journal of Production Economics*, 169:76–88.
- Hatami, S., Ruiz, R., and Andres-Romano, C. (2016). Heuristics for a distributed parallel machine assembly scheduling problem with eligibility constraints. In *Proceedings of 2015 International Conference on Industrial Engineering and Systems Management, IEEE-IESM2015*, pages 145–153.
- Hatami, S., Ruiz, R., and Romano, C. (2014b). Two simple constructive algorithms for the distributed assembly permutation flowshop scheduling problem. In Hernandez, C., Lopez-Paredes, A., and Perez-Rios, J. M., editors, *Managing Complexity*, pages 139–145. Springer.
- Hazir, Ö., Günalay, Y., and Erel, E. (2008). Customer order scheduling problem: A comparative metaheuristics study. *International Journal of Advanced Manufacturing Technology*, 37(5-6):589–598.
- He, D. and Babayan, A. (2002). Scheduling manufacturing systems for delayed product differentiation in agile manufacturing. *International Journal of Production Research*, 40(11):2461–2481.
- He, D., Babayan, A., and Kusiak, A. (2001). Scheduling manufacturing systems in an agile environment. *Robotics and Computer-Integrated Manufacturing*, 17(1-2):87–97.
- He, D. and Kusiak, A. (1998). Designing an assembly line for modular products. *Computers and Industrial Engineering*, 34(1):37–52.
- Hwang, F. and Lin, B. (2012). Two-stage assembly-type flowshop batch scheduling problem subject to a fixed job sequence. *Journal of the Operational Research Society*, 63(6):839–845.
- Jung, S. and Kim, B. (2016). Novel mathematical models for two-stage assembly flow shop scheduling problem with deterioration and preventive maintenance activities. *ICIC Express Letters, Part B: Applications*, 7(11):2477–2482.
- Jung, S., Woo, Y.-B., and Kim, B. (2017). Two-stage assembly scheduling problem for processing products with dynamic component-sizes and a set up time. *Computers and Industrial Engineering*, 104:98–113.
- Karuno, Y. and Nagamochi, H. (2002). A better approximation for the two-stage assembly scheduling problem with two machines at the first stage. *Lecture Notes in Computer Science*, 2518 LNCS:199–210.

- Kheirandish, O., Tavakkoli-Moghaddam, R., and Karimi-Nasab, M. (2015). An artificial bee colony algorithm for a two-stage hybrid flowshop scheduling problem with multilevel product structures and requirement operations. *International Journal of Computer Integrated Manufacturing*, 28(5):437–450.
- Khodke, P. and Bhongade, A. (2013). Real-time scheduling in manufacturing system with machining and assembly operations: A state of art. *International Journal of Production Research*, 51(16):4966–4978.
- Kolisch, R. and Hess, K. (2000). Efficient methods for scheduling make-to-order assemblies under resource, assembly area and part availability constraints. *International Journal of Production Research*, 38(1):207–228.
- Komaki, G. and Kayvanfar, V. (2015). Grey wolf optimizer algorithm for the two-stage assembly flowshop scheduling problem with release time. *Journal of Computational Science*, 8:109–120.
- Komaki, G., Teymourian, E., and Kayvanfar, V. (2016). Minimising makespan in the two-stage assembly hybrid flow shop scheduling problem using artificial immune systems. *International Journal of Production Research*, 54(4):963–983.
- Komaki, G., Teymourian, E., Kayvanfar, V., and Booyavi, Z. (2017). Improved discrete cuckoo optimization algorithm for the three-stage assembly flowshop scheduling problem. *Computers and Industrial Engineering*, 105:158–173.
- Koulamas, C. and Kyparisis, G. (2001). The three-stage assembly flowshop scheduling problem. *Computers and Operations Research*, 28(7):689–704.
- Koulamas, C. and Kyparisis, G. (2007). A note on the two-stage assembly flow shop scheduling problem with uniform parallel machines. *European Journal of Operational Research*, 182(2):945–951.
- Kovalyov, M., Potts, C., and Strusevich, V. (2004). Batching decisions for assembly production systems. *European Journal of Operational Research*, 157(3):620–642.
- Kusiak, A. (1989). Aggregate scheduling of a flexible machining and assembly system. *IEEE Transactions on Robotics and Automation*, 5(4):451–459.
- Kyparisis, G. and Koulamas, C. (2002). Assembly line scheduling with concurrent operations and parallel machines. *INFORMS Journal on Computing*, 14(1):68–80.
- Laub, J., Fowler, J., and Keha, A. (2007). Minimizing makespan with multiple-orders-per-job in a two-machine flowshop. *European Journal of Operational Research*, 182(1):63–79.
- Lee, C.-Y., Cheng, T., and Lin, B. (1993). Minimizing the makespan in the 3-machine assembly-type flowshop scheduling problem. *Management Science*, 39(5):616–625.
- Lee, I. S. (2013). Minimizing total tardiness for the order scheduling problem. *International Journal of Production Economics*, 144(1):128–134.
- Lee, J.-Y. and Bang, J.-Y. (2016). A two-stage assembly-type flowshop scheduling problem for minimizing total tardiness. *Mathematical Problems in Engineering*, 2016:6409321.
- Leung, J., Li, H., and Pinedo, M. (2005a). Order scheduling models: An overview. *Multidisciplinary Scheduling: Theory and Applications*, pages 37–53.
- Leung, J. Y.-T., Lee, C., Ng, C., and Young, G. (2008a). Preemptive multiprocessor order scheduling to minimize total weighted flowtime. *European Journal of Operational Research*, 190(1):40–51.
- Leung, J. Y. T., Li, H., and Pinedo, M. (2005b). Order Scheduling in an Environment with Dedicated Resources in Parallel. *Journal of Scheduling*, 8(5):355–386.

- Leung, J. Y.-T., Li, H., and Pinedo, M. (2006a). Approximation algorithms for minimizing total weighted completion time of orders on identical machines in parallel. *Naval Research Logistics*, 53(4):243–260.
- Leung, J. Y.-T., Li, H., and Pinedo, M. (2006b). Scheduling orders for multiple product types with due date related objectives. *European Journal of Operational Research*, 168(2):370–389.
- Leung, J. Y. T., Li, H., and Pinedo, M. (2007a). Scheduling orders for multiple product types to minimize total weighted completion time. *Discrete Applied Mathematics*, 155(8):945–970.
- Leung, J. Y. T., Li, H., and Pinedo, M. (2008b). Scheduling orders on either dedicated or flexible machines in parallel to minimize total weighted completion time. *Annals of Operations Research*, 159(1):107–123.
- Leung, J. Y.-T., Li, H., Pinedo, M., and Zhang, J. (2007b). Minimizing total weighted completion time when scheduling orders in a flexible environment with uniform machines. *Information Processing Letters*, 103(3):119–129.
- Li, C.-L. and Ou, J. (2007). Coordinated scheduling of customer orders with decentralized machine locations. *IIE Transactions (Institute of Industrial Engineers)*, 39(9):899–909.
- Li, C.-L. and Vairaktarakis, G. (2007). Coordinating production and distribution of jobs with bundling operations. *IIE Transactions (Institute of Industrial Engineers)*, 39(2):203–215.
- Li, X., Zhang, X., Yin, M., and Wang, J. (2015). A genetic algorithm for the distributed assembly permutation flowshop scheduling problem. In *2015 IEEE Congress on Evolutionary Computation, CEC2015-Proceedings*, pages 3096–3101.
- Liao, C.-J. and Chuang, C.-H. (1996). Sequencing with setup time and order tardiness trade-offs. *Naval Research Logistics*, 43(7):971–984.
- Liao, C.-J., Lee, C.-H., and Lee, H.-C. (2015). An efficient heuristic for a two-stage assembly scheduling problem with batch setup times to minimize makespan. *Computers and Industrial Engineering*, 88:317–325.
- Liao, C.-J. and Liao, C.-C. (2008). An ant colony optimisation algorithm for scheduling in agile manufacturing. *International Journal of Production Research*, 46(7):1813–1824.
- Lin, B. and Cheng, T. (2002). Fabrication and assembly scheduling in a two-machine flowshop. *IIE Transactions (Institute of Industrial Engineers)*, 34(11):1015–1020.
- Lin, B., Cheng, T., and Chou, A. (2006). Scheduling in an assembly-type production chain with batch transfer. *Omega*, 35(2):143–151.
- Lin, B. and Kononov, A. (2007). Customer order scheduling to minimize the number of late jobs. *European Journal of Operational Research*, 183(2):944–948.
- Lin, W.-C., Yin, Y., Cheng, S.-R., Cheng, T., Wu, C.-H., and Wu, C.-C. (2017). Particle swarm optimization and opposite-based particle swarm optimization for two-agent multi-facility customer order scheduling with ready times. *Applied Soft Computing Journal*, 52:877–884.
- Liu, C.-H. (2009). Lot streaming for customer order scheduling problem in job shop environments. *International Journal of Computer Integrated Manufacturing*, 22(9):890–907.
- Maboudian, Y. and Shafaei, R. (2009). Modeling a bi-criteria two stage assembly flow shop scheduling problem with sequence dependent setup times. In *IEEM2009 - IEEE International Conference on Industrial Engineering and Engineering Management*, pages 1748–1752.
- Mahdavi, I., Komaki, G., and Kayvanfar, V. (2011). Aggregate hybrid flowshop scheduling with assembly operations. In *2011 IEEE 18th International Conference on Industrial Engineering and Engineering Management, IEandEM2011*, number PART 1, pages 663–667.

- Maleki-Daronkolaei, A. and Seyedi, I. (2013). Taguchi method for three-stage assembly flow shop scheduling problem with blocking and sequence-dependent set up times. *Journal of Engineering Science and Technology*, 8(5):603–622.
- Maleki-Darounkolaei, A., Modiri, M., Tavakkoli-Moghaddam, R., and Seyyedi, I. (2012). A three-stage assembly flow shop scheduling problem with blocking and sequence-dependent set up times. *Journal of Industrial Engineering International*, 8(1):1–7.
- Mason, S. and Chen, J.-S. (2010). Scheduling multiple orders per job in a single machine to minimize total completion time. *European Journal of Operational Research*, 207(1):70–77.
- Mozdgir, A., FatemiGhomi, S., Jolai, F., and Navaei, J. (2013a). Two-stage assembly flow-shop scheduling problem with non-identical assembly machines considering setup times. *International Journal of Production Research*, 51(12):3625–3642.
- Mozdgir, A., Ghomi, S., Jolai, F., and Navaei, J. (2013b). Three meta-heuristics to solve the no-wait two-stage assembly flow-shop scheduling problem. *Scientia Iranica*, 20(6):2275–2283.
- Navaei, J., FatemiGhomi, S., Jolai, F., and Mozdgir, A. (2014). Heuristics for an assembly flow-shop with non-identical assembly machines and sequence dependent setup times to minimize sum of holding and delay costs. *Computers and Operations Research*, 44:52–65.
- Navaei, J., FatemiGhomi, S., Jolai, F., Shiraqai, M., and Hidaji, H. (2013). Two-stage flow-shop scheduling problem with non-identical second stage assembly machines. *International Journal of Advanced Manufacturing Technology*, 69(9-12):2215–2226.
- Navaei, J., Mozdgir, A., and Hidaji, H. (2010). Two-stage assembly flow-shop scheduling problem with bi-objective of number of tardy and makespan minimization. In *Proceedings of the 2010 International Conference on Artificial Intelligence, ICAI2010*, volume 2, pages 704–709.
- Nejati, M., Mahdavi, I., Hassanzadeh, R., and Mahdavi-Amiri, N. (2016). Lot streaming in a two-stage assembly hybrid flowshop scheduling problem with a work shift constraint. *Journal of Industrial and Production Engineering*, 33(7):459–471.
- Ng, C., Cheng, T., and Yuan, J. (2002). Strong np-hardness of the single machine multi-operation jobs total completion time scheduling problem. *Information Processing Letters*, 82(4):187–191.
- Ng, C., Cheng, T., and Yuan, J. (2003). Concurrent open shop scheduling to minimize the weighted number of tardy jobs. *Journal of Scheduling*, 6(4):405–412.
- Nikzad, F., Rezaeian, J., Mahdavi, I., and Rastgar, I. (2015). Scheduling of multi-component products in a two-stage flexible flowshop. *Applied Soft Computing Journal*, 32:132–143.
- Perez-Gonzalez, P. and Framinan, J. (2014). A common framework and taxonomy for multicriteria scheduling problems with interfering and competing jobs: Multi-agent scheduling problems. *European Journal of Operational Research*, 235(1):1–16.
- Pongcharoen, P., Hicks, C., Braiden, P., and Stewardson, D. (2002). Determining optimum genetic algorithm parameters for scheduling the manufacturing and assembly of complex products. *International Journal of Production Economics*, 78(3):311–322.
- Potts, C., Savast’janov, S., Strusevich, V., Van Wassenhove, L., and Zwaneveld, C. (1995). The two-stage assembly scheduling problem: Complexity and approximation. *Operations Research*, 43(2):346–355.
- Roach, A. and Nagi, R. (1996). A hybrid GA - SA algorithm for Just-in-Time scheduling of multi-level assemblies. *Computers and Industrial Engineering*, 30(4):1047–1060.
- Roemer, T. and Ahmadi, R. (1997). The complexity of scheduling customer orders. In *INFORMS Conference 1997, Dallas*.

- Roemer, T. A. (2006). A note on the complexity of the concurrent open shop problem. *Journal of Scheduling*, 9(4):389–396.
- Seidgar, H., Kiani, M., Abedi, M., and Fazlollahtabar, H. (2014). An efficient imperialist competitive algorithm for scheduling in the two-stage assembly flowshop problem. *International Journal of Production Research*, 52(4):1240–1256.
- Seidgar, H., Zandieh, M., and Mahdavi, I. (2016). Bi-objective optimization for integrating production and preventive maintenance scheduling in two-stage assembly flowshop problem. *Journal of Industrial and Production Engineering*, 33(6):404–425.
- Shan, H., Chung, T.-P., and Hu, L. (2014). A note on “A mathematical model and extension algorithm for assembly flexible flow shop scheduling problem”. *International Journal of Advanced Manufacturing Technology*, 74(5-8):819–822.
- Shi, Z., Huang, Z., and Shi, L. (2018). Customer order scheduling on batch processing machines with incompatible job families. *International Journal of Production Research*, 56(1-2):795–808.
- Shi, Z., Wang, L., Liu, P., and Shi, L. (2017). Minimizing completion time for order scheduling: Formulation and heuristic algorithm. *IEEE Transactions on Automation Science and Engineering*, 14(4):1558–1569.
- Shoaaardebili, N. and Fattahi, P. (2015). Multi-objective meta-heuristics to solve three-stage assembly flow shop scheduling problem with machine availability constraints. *International Journal of Production Research*, 53(3):944–968.
- Shokrollahpour, E., Zandieh, M., and Dorri, B. (2011). A novel imperialist competitive algorithm for bi-criteria scheduling of the assembly flowshop problem. *International Journal of Production Research*, 49(11):3087–3103.
- Su, L.-H., Chen, P.-S., and Chen, S.-Y. (2013). Scheduling on parallel machines to minimise maximum lateness for the customer order problem. *International Journal of Systems Science*, 44(5):926–936.
- Sun, X., Morizawa, K., and Nagasawa, H. (2003). Powerful heuristics to minimize makespan in fixed 3-machine, assembly-type flowshop scheduling. *European Journal of Operational Research*, 146(3):498–516.
- Sung, C. and Juhn, J. (2009). Makespan minimization for a 2-stage assembly scheduling problem subject to component available time constraint. *International Journal of Production Economics*, 119(2):392–401.
- Sung, C. and Kim, H. (2008). A two-stage multiple-machine assembly scheduling problem for minimizing sum of completion times. *International Journal of Production Economics*, 113(2):1038–1048.
- Sung, C. S. and Yoon, S. H. (1998). Minimizing total weighted completion time at a pre-assembly stage composed of two feeding machines. *International Journal of Production Economics*, 54(3):247–255.
- Tajbakhsh, Z., Fattahi, P., and Behnamian, J. (2014). Multi-objective assembly permutation flow shop scheduling problem: A mathematical model and a meta-heuristic algorithm. *Journal of the Operational Research Society*, 65(10):1580–1592.
- Teymourian, E., Kayvanfar, V., Komaki, G., and Khodarahmi, M. (2016). An enhanced intelligent water drops algorithm for scheduling of an agile manufacturing system. *International Journal of Information Technology and Decision Making*, 15(2):239–266.
- Tian, Y., Liu, D., Yuan, D., and Wang, K. (2013). A discrete PSO for two-stage assembly scheduling problem. *International Journal of Advanced Manufacturing Technology*, 66(1-4):481–499.
- T’Kindt, V. and Billaut, J.-C. (2006). *Multicriteria Scheduling - Theory, Models and Algorithms (2. ed.)*. Springer.

- Torabzadeh, E. and Zandieh, M. (2010). Cloud theory-based simulated annealing approach for scheduling in the two-stage assembly flowshop. *Advances in Engineering Software*, 41(10-11):1238–1243.
- Tozkapan, A., Kirca, O., and Chung, C.-S. (2003). A branch and bound algorithm to minimize the total weighted flowtime for the two-stage assembly scheduling problem. *Computers and Operations Research*, 30(2):309–320.
- Wagneur, E. and Sriskandarajah, C. (1993). Openshops with jobs overlap. *European Journal of Operational Research*, 71(3):366–378.
- Wang, F., Jia, Z., Liu, W., and Zhao, G. (2011). Genetic algorithms with a new repair operator for assembly job shop scheduling. *International Journal of Industrial Engineering: Theory Applications and Practice*, 18(7):377–385.
- Wang, G. and Cheng, T. (2003). Customer order scheduling to minimize total weighted completion time. *Proc. 1st Multidisciplinary Conf. Scheduling Theory Appl*, pages 409–416.
- Wang, G. and Cheng, T. (2007). Customer order scheduling to minimize total weighted completion time. *Omega*, 35(5):623–626.
- Wang, K., Ma, W., Luo, H., and Qin, H. (2016). Coordinated scheduling of production and transportation in a two-stage assembly flowshop. *International Journal of Production Research*, 54(22):6891–6911.
- Wang, L., Shi, Z., and Shi, L. (2013). A novel quadratic formulation for customer order scheduling problem. In *IEEE International Conference on Automation Science and Engineering*, pages 576–580.
- Wong, T., Chan, F., and Chan, L. (2009). A resource-constrained assembly job shop scheduling problem with lot streaming technique. *Computers and Industrial Engineering*, 57(3):983–995.
- Wu, C.-C., Chen, J.-Y., Lin, W.-C., Lai, K., Liu, S.-C., and Yu, P.-W. (2018). A two-stage three-machine assembly flow shop scheduling with learning consideration to minimize the flowtime by six hybrids of particle swarm optimization. *Swarm and Evolutionary Computation*.
- Xiong, F. and Xing, K. (2014). Meta-heuristics for the distributed two-stage assembly scheduling problem with bi-criteria of makespan and mean completion time. *International Journal of Production Research*, 52(9):2743–2766.
- Xiong, F., Xing, K., and Wang, F. (2015). Scheduling a hybrid assembly-differentiation flowshop to minimize total flow time. *European Journal of Operational Research*, 240(2):338–354.
- Xiong, F., Xing, K., Wang, F., Lei, H., and Han, L. (2014). Minimizing the total completion time in a distributed two stage assembly system with setup times. *Computers and Operations Research*, 47:92–105.
- Xu, J., Wu, C.-C., Yin, Y., Zhao, C., Chiou, Y.-T., and Lin, W.-C. (2016). An order scheduling problem with position-based learning effect. *Computers and Operations Research*, 74:175–186.
- Xu, X., Ma, Y., Zhou, Z., and Zhao, Y. (2015). Customer Order Scheduling on Unrelated Parallel Machines to Minimize Total Completion Time. *IEEE Transactions on Automation Science and Engineering*, 12(1):244–257.
- Yan, H.-S., Wan, X.-Q., and Xiong, F.-L. (2014). A hybrid electromagnetism-like algorithm for two-stage assembly flow shop scheduling problem. *International Journal of Production Research*, 52(19):5626–5639.
- Yang, J. (2005). The complexity of customer order scheduling problems on parallel machines. *Computers and Operations Research*, 32(7):1921–1939.
- Yang, J. and Posner, M. E. (2005). Scheduling Parallel Machines for the Customer Order Problem. *Journal of Scheduling*, 8(1):49–74.

- Yokoyama, M. (2001). Hybrid flow-shop scheduling with assembly operations. *International Journal of Production Economics*, 73(2):103–116.
- Yokoyama, M. (2004). Scheduling for two-stage production system with setup and assembly operations. *Computers and Operations Research*, 31(12):2063–2078.
- Yokoyama, M. (2008). Flow-shop scheduling with setup and assembly operations. *European Journal of Operational Research*, 187(3):1184–1195.
- Yokoyama, M. and Santos, D. (2005). Three-stage flow-shop scheduling with assembly operations to minimize the weighted sum of product completion times. *European Journal of Operational Research*, 161(3):754–770.
- Zhang, Y., Zhou, Z., and Liu, J. (2010). The production scheduling problem in a multi-page invoice printing system. *Computers and Operations Research*, 37(10):1814–1821.