

# Accepted Manuscript

Hybrid Stage Shop Scheduling

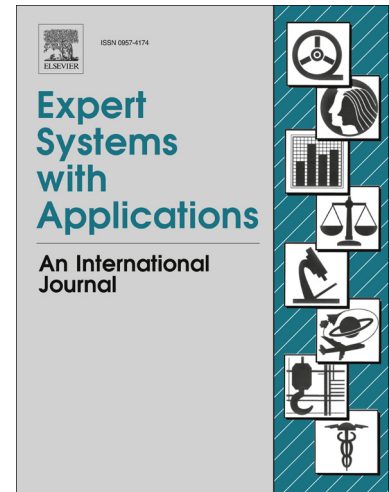
Andrea Rossi, Sauro Soldani, Michele Lanzetta

PII: S0957-4174(14)00829-X

DOI: <http://dx.doi.org/10.1016/j.eswa.2014.12.050>

Reference: ESWA 9775

To appear in: *Expert Systems with Applications*



Please cite this article as: Rossi, A., Soldani, S., Lanzetta, M., Hybrid Stage Shop Scheduling, *Expert Systems with Applications* (2015), doi: <http://dx.doi.org/10.1016/j.eswa.2014.12.050>

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

# Hybrid Stage Shop Scheduling

Andrea Rossi<sup>(1)</sup>, Sauro Soldani<sup>(2)</sup>, Michele Lanzetta<sup>(3)</sup>

Department of Civil and Industrial Engineering, University of Pisa, Italy

<sup>(1)</sup> arossi@ing.unipi.it

<sup>(2)</sup> sauroviola@gmail.com

<sup>(3)</sup> lanzetta@unipi.it

Corresponding author

Department of Civil and Industrial Engineering

Largo Lazzarino

56122 Pisa, Italy

Tel. +39-050-2218122

Cel. +39-320-4212172

## Abstract

The proposed Hybrid Stage Shop Scheduling (HSSS) model, inspired from a real case in the high-fashion industry, aims to fully exploit the potential of parallel resources, splitting and overlapping concurrent operations among teams of multifunctional machines and operators on the same job. The HSSS formally extends mixed shop scheduling (a combination of flowshop and open shop), which is able to model routing flexibility, and hybrid shop scheduling, which provides resource flexibility. To also include operational flexibility through alternative plans obtained by reordering operations linked by undefined or arbitrary (immaterial) precedence constraints, the proposed model integrates process planning and group shop scheduling.

A mixed integer linear programming model and a theory based on disjunctive graphs have been proposed to explore the composite relations between nodes involving immaterial relations and deploying their routing rules.

A constructive  $O(\text{resources} \times \text{jobs}^2)$  algorithm to generate a feasible plan/schedule in the most general case has been developed and applied to a case study.

**Keywords:** manufacturing systems, planning/scheduling integration, mixed shop scheduling, nonlinear routing, resource flexibility

## 1 Introduction

We consider a rather general model of *mixed shop* in which a set of operations for a given set of jobs has to be scheduled on a set of machines, which includes two extensions to the standard scheduling problem as defined by Dauzère-Pérès, Roux, and Lasserre (1998):

1. an operation can be processed by one resource chosen in a given set (*resource flexibility*); for the sake of generality, we use the standard term *resource* from the scheduling theory instead of machine;
2. the routing of products in the shop floor is not necessarily linear, i.e. an operation can have more than one predecessor and more than one successor on the routing (*nonlinear routing*).

The mixed shop type considered here is a *hybrid* (or *flexible*) shop type combination of *flowshop* and *open shop*. A hybrid flowshop is a flowline with parallel resources. In a flowshop, the sequence of operations of each job (routing) is linear and predefined; in an open shop the sequence of operations is immaterial (or undefined). In a mixed shop, the set of constraints between operations is partitioned into two sets: flowshop type set and open shop type set (Masuda, Ishii, & Nishida, 1985).

### 1.1. Integration of process planning and shop scheduling

Mixed shop is the paradigm for the integration of process planning and shop scheduling (Tan & Khoshnevis, 2000). Process planning has been defined by the Society of Manufacturing Engineers as *the systematic determination of the methods by which a product is to be manufactured economically and competitively*. Traditionally, process planning and shop scheduling are applied separately and sequentially. If a single output of process planning (the plan) is considered as the input to flowshop scheduling, routing constraints from planning may create bottleneck situations on some resources while other can be starving. Also the line balancing may be affected. Consequently, the global system performance can be improved by integrating planning and scheduling. However, the integration of process planning and shop scheduling does not consider operations belonging to the set of open shop type but rather the assignment of optimal process plans among a number of (predefined) alternatives. Stecke and Raman (1995) described a scheme for classifying different types of flexibility conventionally associated with the ability to manufacture a variety of part types by flexible manufacturing systems. In this classification *operation flexibility* assumes that more alternative plans can be generated by the process planner for a given job. Kis (2003) and Leung (2010) modeled an integrated process planning and shop scheduling system by disjunctive AND/OR graphs. The branches of an OR-subgraph constitute a set of alternative subroutes: exactly one of them must be chosen during scheduling. AND/OR graphs are a generalization of the resource

alternatives of individual operations; however, immaterial constraints among operations cannot be considered effectively. Go, Wahab, Rahman, Ramli, and Hussain (2012) and Bentaha, Battaïa, and Dolgui (2014) approached the design of disassembly lines for end-of-life products with the objective to maximize the line profit. An AND/OR graph was used to model the precedence relationships among tasks and subassemblies and the disassembly alternatives. Doh, Yu, Kim, Lee, and Nam (2013) considered alternative machines for each operation (resource flexibility), in addition to specifying multiple process plans alternative operations and their sequence by a network model with AND/OR nodes. Otto and Otto (2014) described a precedence graph approach that is based on learning from past feasible production sequences and forms a sufficient precedence graph that guarantees feasible assembly line balancing in the automotive industry. The assignment of tasks to stations is due to restrictions, which can be expressed in a precedence graph that includes direct and indirect conjunctive precedence relations. Phanden, Jain, and Verma (2013) developed a simulation-based genetic algorithm (GA) to integrate the process planning and scheduling function that can be quickly implemented in a company with existing process planning and scheduling departments. Bensmaine, Dahane, and Benyoucef (2013) proposed a new heuristics to integrate the process planning and scheduling problem for reconfigurable machine tools, each with multiple configurations, and can perform different operations with different capacities. They considered only direct precedence graph relations.

### **1.2. Mixed and group shop scheduling**

In order to reduce the gap with real manufacturing systems, the mixed shop scheduling problem has been regarded as a mixture of flow (or job) and open shop scheduling problems, where operations with immaterial precedence constraints are grouped in the route of the related job. In 1997, the *group shop scheduling* problem was first introduced in the context of a mathematical competition (Whizzkids, 1997). Regarding the group shop scheduling problem, Blum and Sampels (2004) used a disjunctive graph representation for group shop scheduling and applied an ant colony algorithm to tackle the problem complexity. Liu, Ong, and Ng (2005) proposed a tabu search for group shop scheduling and evaluated the algorithm performance on a set of benchmark problems. Ahmadizar and Shahmaleki (2014) considered the stochastic group shop scheduling problem where both release dates and processing times are random variables, normally, exponentially or uniformly distributed. From the literature above, it can be observed that the mixed shop model includes the models on integration of process planning and scheduling and those on group shop scheduling, by allowing alternative plans produced simply reordering operations connected by immaterial constraints (Figure 1).

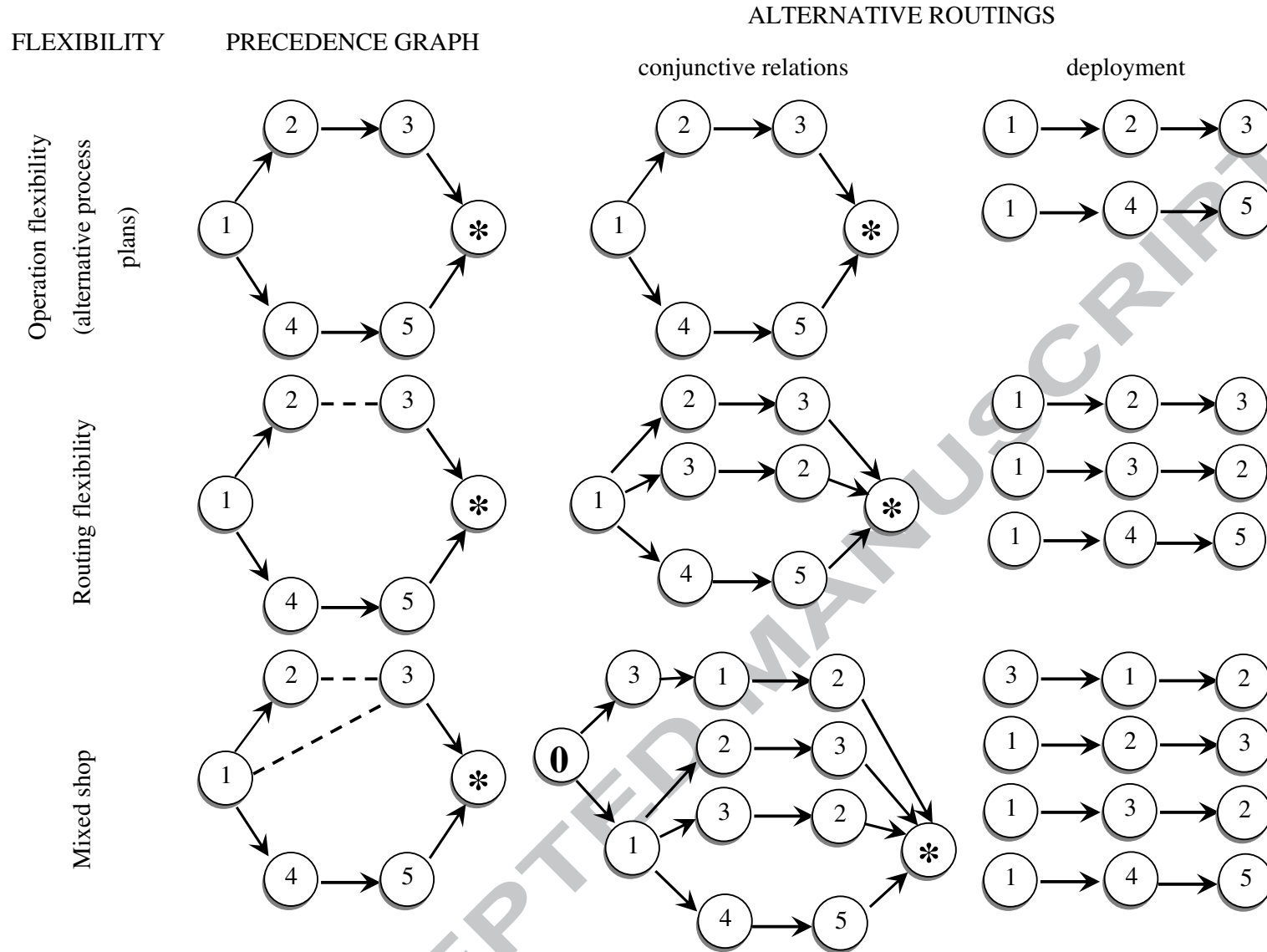
Revised manuscript

According to Stecke and Raman (1995), in addition to operation flexibility, *routing flexibility* is another aspect of the scheduling flexibility related to the ability of generating alternative paths, which can be followed through the system for a given process plan. As discussed by Rossi and Lanzetta (2013), shared buffers between stages allow routing flexibility, by the permutation of job sequences on resources.

Figure 2 shows as an (exclusive) OR node (node 0 towards  $O_{31}$  and  $O_{32}$ ) that can be reworded as a no-exclusive OR by an immaterial relation, which allows more alternative routings for the scheduler module.

ACCEPTED MANUSCRIPT

Revised manuscript



Revised manuscript

**Figure 1. Example of mixed shop precedence graphs achieved by operation and routing flexibility (according to Stecke & Raman, 1995).  
No-exclusive OR nodes described by immaterial relations give alternative routing for the scheduler module in order to minimize the completion time.**

Figure 1 about here

As shown by Masuda, Ishii, and Nishida (1985), the mixed shop problem is NP-hard. Relatively few papers were proposed on the subject. Shakhlevich et al. (2000) discussed the complexity of mixed shop problems under various criteria and clarified the boundary between polynomially solvable and NP-hard problems. Blazewicz and Kobler (2002) reviewed the properties of simple precedence graphs for scheduling problems and exhibited several new polynomial cases for various problems on unrelated parallel machines under arbitrary resource constraints. Ferrell, Sale, Sams, and Yellamarju (2000) approached the problem by heuristics and evaluated the performance in a promising set of dispatching rules. Nasiri and Kianfar (2011) proposed a *stage shop* scheduling, an open job shop scheduling problem where all the operations of the same job with immaterial precedence constraints are grouped in a stage of the job shop-type. Aloulou and Artigues (2010) and Amin-Naseri and Ashfadi (2012) modeled simple or primitive non-linear precedence constraints, in order to split macro operations into micro operations at stage level.



**Table 1. Main contributions from the literature to the mixed shop scheduling problem as an integration of alternative nonlinear routing approaches (group shop, AND/OR)**

Authors	Graph type			MILP	Theory
	Disjunctive arc		Node		
	Flexible or hybrid	Nonlinear routing			
		Group shop	Mixed shop		
Masuda, Ishii, and Nishida (1985)			•		•
Dauzère-Pérès, Roux, and Lasserre (1998)	•		•		•
Kis (2003)			•		
Blum and Sampels (2004)		•			•
Liu, Ong, and Ng (2005)		•			
Leung (2010)				•	
Aloulou and Artigues (2010)			•		•
Nasiri and Kianfar (2011)			•		•
Amin-Naseri and Ashfadi (2012)	•		•		•
Doh, Yu, Kim, Lee, and Nam (2013)	•			•	•
Bensmaine, Dahane, and Benyoucef (2013)				•	•
Ahmadizar and Shahmaleki (2014)		•			•
Otto and Otto (2014)	•			•	
Current work	•		•	•	•

From this literature analysis chronologically synthesized in Table 1, it can be observed that the most recent works (towards the bottom) converge to the generalization of integrated process planning and group shop scheduling. For this purpose, further research for the definition and use of new precedence relations is required.

### **1.3. From mixed to hybrid shop scheduling**

Resource flexibility and nonlinear routing allow a higher level of manufacturing flexibility within each stage of the flowshop: multiple resources can process in parallel subparts of the same product (*multi-resource systems*).

Resource flexibility in the standard flowshop model was reviewed by Ribas, Leisten, and Framinan (2010) and Ruiz and Maroto (2005). In recent years, approaches to hybrid flowshops were proposed by Behnamian and Zandieh (2011), considering the minimization of earliness and quadratic tardiness penalties by a discrete colonial competitive algorithm. Rossi, Pandolfi, and Lanzetta (2013) considered a non-permutation hybrid flow shop scheduling problem, with parallel batching machines, parallel batch families and a machine loading system modeled as sequence-independent uniform set-up times, with the purpose of reducing the number of tardy jobs and the makespan. They proposed two dynamic heuristics based on the critical ratio of allowance set-up and processing time in the scheduling horizon. Li, Meng, Liang, and Zhao (2014) proposed a heuristic-search genetic algorithm, which results in lower complexity and higher efficiency for multi-stage hybrid flow shop scheduling with batch processing machines. Amin-Naseri and Ashfadi (2012) considered some primitive non-linear precedence relations in a flexible job shop environment and proposed a local search problem-specific to speed up a genetic algorithm. Rossi (2014) proposed an ant colony optimization approach based on a disjunctive graph model in order to schedule a manufacturing system with resource flexibility, separable sequence dependent/independent setup and transportation times. Benavides, Ritt, and Miralles (2014) proposed an extension to flowshop with heterogeneous resources among stages, to optimize worker assignment to workstations of health care departments, with the objective of minimizing the makespan, while respecting the diverse capabilities and paces of heterogeneous workers. To our knowledge, only Amin-Naseri and Ashfadi (2012) considered some aspects of nonlinear routing with resource flexibility using a genetic algorithm, without exploring the theoretical implications.

### **1.4. Problem motivation and approach**

The research was motivated by a real manufacturing case, a high-fashion company based in Florence, Italy, for the production of high-fashion goods by skilled artisans.

An operation (e.g. the manual assembly of a woman bag) can be split in a number of micro-operations (e.g. stitching a handle and sewing a slide fastener), which can be performed on the same job in a fixed or arbitrary routing by different resources of the belonging flowshop stage (e.g. by two operators of the assembly shop). Splitting operations of multi-resource systems affects process planning providing further flexibility and increases the number of scheduling alternatives for the

shop system and, potentially, shorter schedules can be obtained. Alternative plans produced simply reordering operations reduce the process planner workload. Generating alternative plans improves machine utilization and line balancing.

Figure 2 shows an example problem derived from the examined case represented as a disjunctive graph. The three shaded areas are examples of macro operations that have been split by detailing the sub operations and resources in order to process them in parallel (overlapping) to reduce the makespan. Precedence graphs allow different routings (sequence permutations) by the inclusion of the immaterial precedence arcs in addition to directed (or conjunctive) arcs. As shown in the example, mixed routing can also span across consecutive stages.

The most general case, which represents the focus of current paper, is the shaded graph relation  $O_{122}-O_{133}-O_{252}$  in the example. The cited operations represent respectively: gluing the two halves of the bag, stitching a pocket on just one of the two sides, stitching a buckle to both halves, with realistic times indicated. Gluing is on the previous stage because of a precedence constraint.

Sequence  $O_{122}-O_{133}-O_{252}$  is potentially shorter versus sequence  $O_{122}-O_{252}-O_{133}$  for the possible parallelization of  $O_{122}$  and  $O_{252}$ .

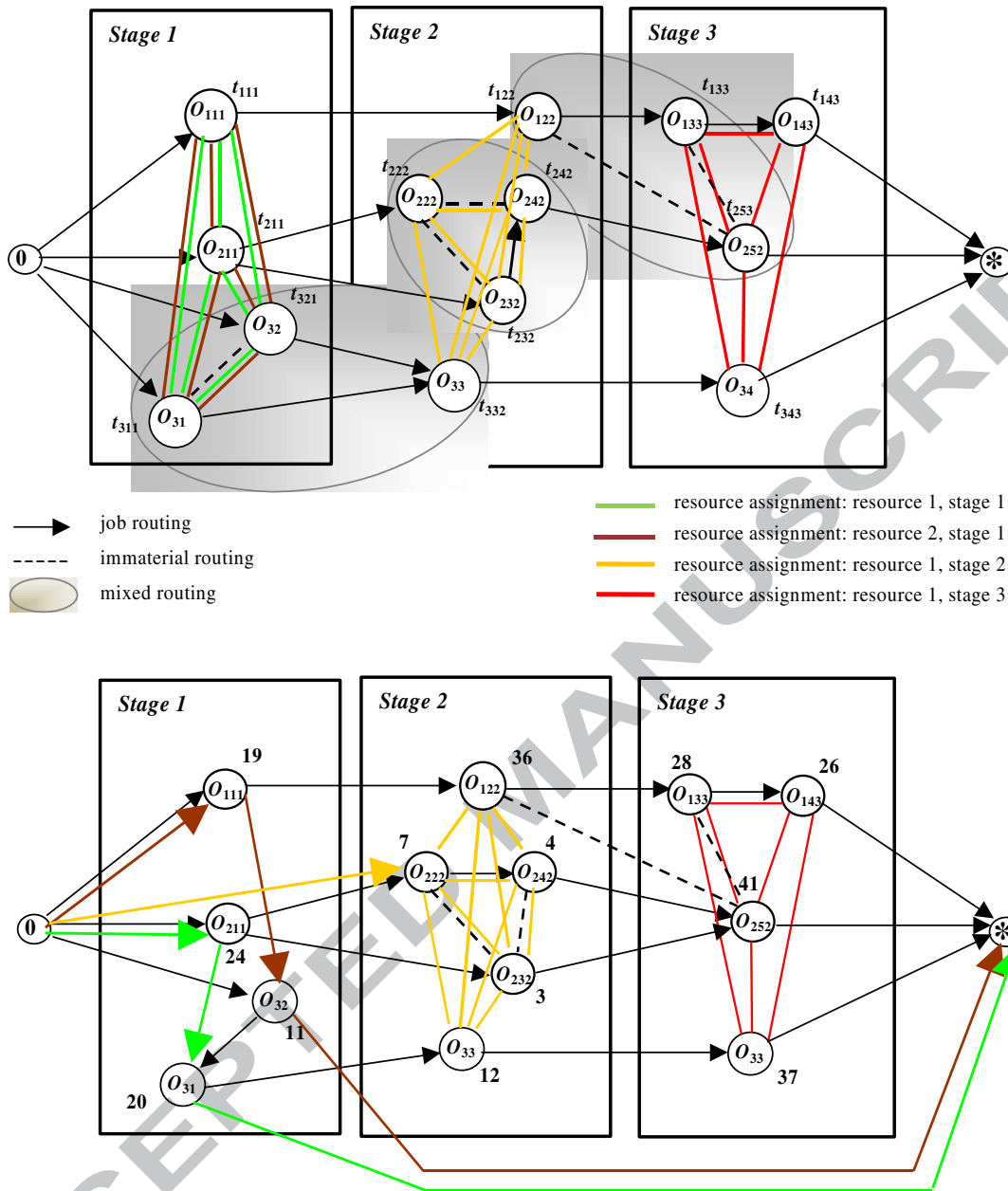


Figure 2. A disjunctive graph modeling a mixed shop for hybrid flowshop scheduling, with processing times  $t_{ijk}$  at nodes  $O_{ijk}$  for  $N (=3)$  jobs on  $S (=3)$  stages, with respectively  $Q_1=2$  (hybrid),  $Q_2=1$ ,  $Q_3=1$  resources. Colored disjunctive arcs represent the candidate resource for the connected nodes (operations). Bottom: the top digraph partially transformed in acyclic digraph by directing disjunctive arcs on two paths starting from node 0 and ending at node \* (green and brown) and one ending in  $O_{222}$  (yellow).

Figure 2 about here

We propose the *hybrid stage shop scheduling* (HSSS), a mixed shop model for hybrid flowshop scheduling with unrelated parallel resources per stage. The proposed underlying mixed model, in addition to routing flexibility that emerges with permutations of the operations linked by immaterial relations, allows: (i) *splitting* operations of the flowshop type set on the same job and assigning them to alternative resources; (ii) *overlapping* the operations of consecutive stages linked by immaterial routing, yielding shorter schedules.

The proposed hybrid stage shop scheduling model completes the mixed stage shop model by Nasiri and Kianfar (2011) and the hybrid stage shop scheduling model by Amin-Naseri and Ashfadi (2012). Extensions include the addition of operation overlapping and splitting within stages and across consecutive stages with the composition of mixed relations, which will be discussed throughout the paper.

The proposed model can be applied to Computer Aided Design (CAD), where designers share the same digital model of the product under development (Leu et al., 2013), in robotic assembly and in disassembly of end of life goods, where operations on subparts can be carried out concurrently, and in train traffic management by railway line branching (Kozan & Liu, 2012).

In section 2 we define the hybrid stage shop scheduling problem and its MILP (section 3). In section 4 we describe the model based on digraphs and in section 5 we propose a set of primitive and mixed rules and their combinations, derived from the precedence graph, in order to deploy and explore various alternative routings. The hybrid stage shop scheduling model is based on a digraph, which is a powerful tool to design scheduling optimization (metaheuristics) algorithms. In section 6 a heuristics is presented to obtain a conjunctive graph, which represents a feasible solution of the hybrid stage shop scheduling problem. Also, the computational complexity is evaluated.

Application to a case study is discussed in section 7.

## 2 Hybrid Stage Shop Scheduling (HSSS)

In hybrid stage shop scheduling,  $N$  jobs have to be scheduled on  $S$  stages with  $R$  resources in all, in accordance with its nonlinear routing, represented by a precedence graph, which includes conjunctive (directed) and disjunctive precedence constraints (described below) among operations belonging to a set of  $l_i$  operations  $O_{ijk}$  with  $i=1,\dots,N$ ,  $j=1,\dots,l_i$ ,  $k=1,\dots,S$ . Each operation  $O_{ijk}$  has to be processed according to its precedence constraint, denoted by a *level*  $L_{ij}$ , for time  $t_{ijk}$  with resource flexibility represented by the assignment to a single resource  $h$  among a set of alternative identical resources  $\{Q_{k-1}, \dots, Q_k\}$  of the stage  $k$  ( $k=1,\dots,S$ ). Each job  $i$  is subject to a release date,  $r_i$ , an

initial level  $\min_{j=1,\dots,l_i} L_{ij} = L_i$ .  $st(O_{ijk})$  and  $ct(O_{ijk})$  denote respectively the starting and the completion time of an operation, with  $Ct_i$  being the total completion time for each job.

No resource can process more than one operation at a time and an operation must be processed by one, and only one, resource; no precedence constraints between operations of different jobs is allowed. All resources are available since the initial time and resource breakdown is not considered. Setup is negligible or is included in the processing time. The handling system offers the flexibility to move a part with negligible transportation time and includes a buffer for each stage. The buffer size of stages  $k=1$  and  $k=S+1$  is  $N$  (i.e. the input and output buffers contain up to  $N$  jobs). The number of slots of the buffer at stage  $k$ ,  $k=2,\dots,S$ , each containing a single job, is defined by the following *Lemma 1*.

*Lemma 1.* The hybrid flowshop scheduling with  $N$  jobs and  $R$  resources is  $B_i = +\infty$  if and only if the shared buffer size for stage  $k$ ,  $k=2,\dots,S$  is at least:

$$N - (Q_k - Q_{k-2}) \quad (1)$$

This proof extends the result achieved by Rossi and Lanzetta (2013). In the worst case, the system becomes congested in a single stage. Let  $k$  be this stage, with  $(Q_k - Q_{k-1})$  the number of its resources. Hence, the number of jobs waiting in the buffer of stage  $k$  or in the previous stages is  $N - (Q_k - Q_{k-1})$ . Similarly, in the previous stage there are  $(Q_{k-1} - Q_{k-2})$  resources. The jobs can wait after processing in the previous stage  $k-1$  before a resource of the stage  $k$  will be free. The minimum buffer size for each of the intermediate stages is given by:

$$N - (Q_k - Q_{k-1}) - (Q_{k-1} - Q_{k-2}) = N - (Q_k - Q_{k-2}) \quad (2)$$

■

Operations of job  $i$  are partitioned in  $k$  subsets  $G_{ik}$  to be performed by resources of stage  $k$  according to the following equations:

$$\sum_{k=1}^S |G_{ik}| = l_i \quad i = 1, \dots, N \quad (3)$$

$$G_{iw} \cap G_{iz} = \emptyset, \quad w, z = 1, \dots, S, w \neq z \quad (4)$$

Revised manuscript

The quantity  $P = \min_{k=1,\dots,S} (Q_k - Q_{k-1})$  represents the degree of resource flexibility, i.e. the *parallelization* capability of the system. We define a hybrid stage shop scheduling model with degree of resource flexibility  $P$  as  $P$ -HSSS model.

The assignment of operations to a given resource of current stage is a source of flexibility, in addition to the flexibility represented by sequencing operations on resources. This double source of flexibility increases the problem complexity along with the performance increase.

### 3 Hybrid stage shop scheduling mixed integer model

A mixed integer linear programming (MILP) model for the hybrid stage shop scheduling is described.

#### Decision variable

$x_{ijkh}$       1, if operation  $j$  of job  $i$  is assigned to resource  $h$  in stage  $k$   
                  0, otherwise

The objective function selected is the total completion time (makespan) minimization and can be formulated as:

$$\text{Min } C_{\max} \quad (5)$$

subject to constraints I.1 to I.15.

$$Ct_i \geq \sum_{k=1}^S \sum_{h=Q_{k-1}+1}^{Q_k} \sum_{j=1}^{l_i} t_{ijk} x_{ijk} \quad \forall i = 1, \dots, N \quad (I.1)$$

$$\sum_{h=Q_{k-1}+1}^{Q_k} x_{ijk} = 1 \quad \forall i = 1, \dots, N, \quad \forall j = 1, \dots, l_i, \quad \forall k = 1, \dots, S \quad (I.2)$$

$$\sum_{k=1}^S x_{ijk} = 1 \quad \forall i = 1, \dots, N, \quad \forall j = 1, \dots, l_i, \quad \forall h = Q_{k-1} + 1, \dots, Q_k \quad (I.3)$$

$$\sum_{j=1}^{l_i} x_{ijk} \leq 1 \quad \forall i = 1, \dots, N, \quad \forall h = Q_{k-1} + 1, \dots, Q_k, \quad \forall k = 1, \dots, S \quad (I.4)$$

Revised manuscript

$$\sum_{i=1}^N x_{ijhk} \leq 1 \quad \forall j = 1, \dots, l_i, \quad \forall h = Q_{k-1} + 1, \dots, Q_k, \quad \forall k = 1, \dots, S \quad (\text{I.5})$$

$$t_{ijk} = \begin{cases} \in \mathfrak{R}^+ & \text{if } x_{ijhk} = 1 \\ 0 & \text{if } x_{ijhk} = 0 \end{cases} \quad \forall i = 1, \dots, N, \quad \forall j = 1, \dots, l_i, \quad \forall h = Q_{k-1} + 1, \dots, Q_k, \quad \forall k = 1, \dots, S \quad (\text{I.6})$$

$$ct_{ijk} - st_{ijk} = t_{ijk} x_{ijhk} \quad \forall i = 1, \dots, N, \quad \forall j = 1, \dots, l_i, \quad \forall h = Q_{k-1} + 1, \dots, Q_k, \quad \forall k = 1, \dots, S \quad (\text{I.7})$$

$$\begin{cases} st_{ij'k} - ct_{ijk} > 0 \\ \in \mathfrak{R} \end{cases} \quad \begin{cases} \text{se } L_{ij} < L_{ij'} \Rightarrow j < j' \\ \text{se } L_{ij} = L_{ij'} \Rightarrow j \leq j' \end{cases} \quad \forall i = 1, \dots, N, \quad \forall j = 1, \dots, l_i, \quad \forall k = 1, \dots, S \quad (\text{I.8})$$

$$r_i \geq 0 \quad \forall i \in N \quad (\text{I.9})$$

$$st_{ijk} \geq 0 \quad \forall i = 1, \dots, N, \quad \forall j = 1, \dots, l_i, \quad \forall k = 1, \dots, S \quad (\text{I.10})$$

$$t_{ijk} \geq 0 \quad \forall i = 1, \dots, N, \quad \forall j = 1, \dots, l_i, \quad \forall k = 1, \dots, S \quad (\text{I.11})$$

$$ct_{ijk} \geq 0 \quad \forall i = 1, \dots, N, \quad \forall j = 1, \dots, l_i, \quad \forall k = 1, \dots, S \quad (\text{I.12})$$

$$Ct_i \geq 0 \quad \forall i = 1, \dots, N \quad (\text{I.13})$$

$$L_{ij} \in \mathfrak{R} \quad \forall i = 1, \dots, N, \quad \forall j = 1, \dots, l_i \quad (\text{I.14})$$

$$x_{ijhk} \in \{0, 1\} \quad \forall i = 1, \dots, N, \quad \forall j = 1, \dots, l_i, \quad \forall h = Q_{k-1} + 1, \dots, Q_k, \quad \forall k = 1, \dots, S \quad (\text{I.15})$$

The constraint equation (I.1) forces the total completion time of job  $i$ , to be less or equal to the sum of all processing times affecting job  $i$ , indexed by operations and stages.

The constraint equation (I.2) ensures that each operation is assigned to exactly one resource.

The constraint equation (I.3) ensures that each operation is specific to exactly one stage.

The constraint equation (I.4) shows that any operation can (sum equal to 1) or cannot (sum equal to 0) be assigned to a particular pair of job and resource.



The constraint equation (I.5) represents the possibility (sum equal to 1) or not (sum equal to 0) that a job involves the use of a resource for a specific operation.

The constraint equation (I.6) requires that the processing time of resource  $h$  of stage  $k$  performing operation  $j$  of job  $i$  is equal to 0 if the transaction is not assigned to any resource.

The constraint equation (I.7) requires that the difference between the completion time and the starting time of a resource busy for an operation of a job be equal to its processing time.

The constraint equation (I.8) ensures that the difference between the starting time of a certain operation and the completion time calculated for the immediately preceding operation is greater than 0 if the value of the variable level of the previous operation is smaller compared to the variable level of the following operation. It is worth dwelling on the condition in which it has the same value of the variable level for two successive operations; in this case the order of execution can be chosen arbitrarily: the result can be greater, less or equal to 0.

Finally, the constraint equations ranging from (I.9) to (I.15) are simply conditions of existence and non-negativity of the variables used in the model.

## 4 Hybrid stage shop scheduling disjunctive graph representation

The hybrid stage shop scheduling problem can be represented by a weighted *disjunctive graph* as in Rossi and Lanzetta (2014), applied to the non permutation flowshop scheduling (Figure 2 top):

$$DG = (\mathfrak{N}, W, C, D, E_{hk}) \quad (6)$$

where

$\mathfrak{N}$  is the set of nodes (operations  $O_{ijk}$ ) plus the dummy start and finishing nodes 0 and \*;

$W_N$  is the set of weight on nodes, represented by the processing times  $t_{ijk}$ ;

$C$  is the set of conjunctive (directed) arcs  $(a,b)$  between every pair of nodes  $a$  and  $b$  on a job routing, representing the precedence constraints between the corresponding operations (*conjunctive relations, C*);

it also includes conjunctive arcs between 0 (\*) and every first (last) node on a routing;

$D$  is the set of disjunctive arcs  $[a,b]$  (equal to  $[b,a]$ ), between every pair of nodes  $a$  and  $b$  on a job routing with immaterial precedence constraints between the corresponding operations (*disjunctive relations, D*);

$E_{hk}$  is the  $h^{\text{th}}$  set ( $h \in \{Q_{k-1}+1, \dots, Q_k\}$ ) of disjunctive arcs  $[u,v]$  between every pair of nodes  $u$  and  $v$

belonging to the same stage  $k$ ;  $u, v \in G_{ik}$ ,  $k=1, \dots, S$ ;  $E_{hk}$  also includes disjunctive arcs between 0 and  $u$

( $v$ ) and between  $u$  ( $v$ ) and  $*$  for all  $u \in G_{ik}$ . The sets  $E_{hk}$  are represented by colored disjunctive arcs in Figure 2 top.

It can be noticed that the  $DG$  includes one kind of conjunctive arc (arrows in the digraph of Figure 2 top) and two kinds of disjunctive arcs:  $D$  for nonlinear routing (dashed lines) and  $E_{hk}$  for resource flexibility (colored lines).

In general, a  $P$ -HSSS problem, i.e. a problem with degree of resource flexibility  $P$ , is represented by  $S$  stages of  $P$  kinds of arcs  $E_{*k}$ . The number of disjunctive arcs in the digraph is  $O(P \cdot S \cdot N^2)$ , because the arcs of  $E_{*k}$  connected with at least  $N$  nodes are  $O(N^2)$  and the *total flexibility* is obtained by  $P \cdot N^2$  arcs of  $E_{*k}$  (Kacem, Hammadi, & Borne, 2002). Consequently, in the  $P$ -HSSS total flexibility is present, but only within each stage.

The set  $D$  describes nonlinear routing for open and consequently for the mixed shop considered here.  $D$  is not required to model linear problems, such as flowshop and hybrid flowshop scheduling. In the next section the interaction between  $C$  and  $D$  type relations will be discussed.

A finite sequence of conjunctive arcs between two nodes is called *path*. The *length* of an arc is equal to the processing time of the node at which it ends. The *path length* is equal to the sum of the lengths of its arcs. A path which starts from 0 and ends at  $*$  is the *loading sequence* on a resource. Figure 2 bottom shows two paths with lengths, respectively,  $24+20$  (green path) and  $19+11$  (brown path), which start from 0 and end at  $*$ ; they are the loading sequences of resources of stage 1. A *cycle* is a path that starts and ends at the same node. If no cycle is present in a conjunctive graph achieved by directing some disjunctive arcs, the conjunctive graph is *acyclic* (Figure 2 bottom). In particular, a digraph is an acyclic conjunctive graph. If an acyclic conjunctive graph includes all the nodes, the related loading sequences on the resources are *feasible schedules* and the makespan is the length of the *critical path*, i.e. the longest path between the dummy start and finishing nodes. Finally, for each job, the length of the longest path between the dummy start node 0 and a given node is the total completion time  $ct(O_{ijk})$  of the corresponding operation.

## 5 Hybrid stage shop scheduling routing rules

Hybrid stage shop scheduling is a more general model for mixed shop scheduling and it is more suitable to describe real manufacturing cases, but it requires the definition of new relations among nodes split at the stage level or across consecutive stages. A digraph as defined in (6) includes  $C$  and  $D$  type primitive relations. By the presence of  $D$  type relations, routing is nonlinear.  $C$  and  $D$  type relations in (6) between nodes involve precedence constraints, which can be described by the

*routing rules* (feasible sequence of operations for a given job) introduced in this section. The proposed *wildcard node* mechanism will also be described.

The complete set of basic relations includes two *primitive C* and *D* type relations defined in (6) listed in Table 2 and five *mixed* relations introduced in Table 3. The complete set of basic relations between nodes  $j$  and  $j + 1, j=1, \dots, L_i - 1$ , of the same job  $i$  and their levels  $L_{ij}$  is:

- Conjunctive relation (*C*),  $L_{i(j+1)} = L_{ij} + 1$ . The routing is linear as in flowshop scheduling (Table 2).
- Disjunctive relation (*D*),  $L_{i(j+1)} = L_{ij}$ . The routing is immaterial as in open shop scheduling, where a sequence of operations is a job permutation (Table 2).
- Mixed relation (*M*). Mixed relations are present, where a disjunctive relation can anticipate, interpose or postpone a direct relation (Table 3).

Levels are the mechanism to represent the precedence constraints between operations.

**Table 2. Primitive relations (2 nodes) and deployment of their routing rules**

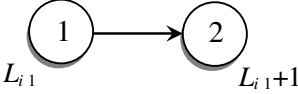
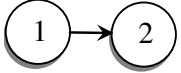
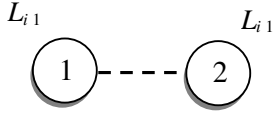
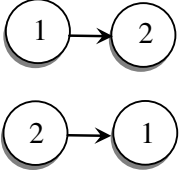
Case	Relations	Notation	Precedence graph	Routing deployment
#1.	Conjunctive (directed)	<i>C</i> (1,2)		
#2.	Disjunctive	<i>D</i> [1,2] (or [2,1])		

Table 2 shows linear and arbitrary relations among operations of the same job and their routing rules.

A *C* relation is a conjunctive arc between a starting and ending node and is represented in round brackets (1,2); the starting node represents the operation processed before in the feasible sequence  $1 \rightarrow 2$ , as imposed by the level  $L_{i2} = L_{i1} + 1$ .

A *D* relation is a disjunctive arc between two nodes; the nodes represent operations processed in an arbitrary order within the feasible sequence; the two feasible sequences are  $1 \rightarrow 2$  and  $2 \rightarrow 1$  associated with a unique level, where  $L_{i2} = L_{i1}$ . A *D* relation has no orientation between nodes and is differentiated from a *C* relation by using square brackets:  $[1,2] = [2,1]$ .

Both conjunctive and disjunctive (primitive) relations are denoted by  $\Gamma = \{C, D\}$ .

Table 3. Mixed relations (3 nodes) and deployment of their routing rules

Case	Notation	Precedence graph	Routing deployment
#3.	$M1$		
#4.	$M2$		
#5.	$M3$		
#6.	$M4$		

Table 3 shows the mixed relations ( $Mx, x=1, \dots, 4$ ) among nodes. A  $M$  relation,  $x \cup y$ , is a union of primitive relations  $x, y \in \Gamma = \{C, D\}$ .

$M1$  and  $M2$  include one disjunctive relation,  $[2,3]$ , and two conjunctive relations, yielding two feasible routing rules.

$M3$  includes two  $D$  relations and one  $C$  relation, (1,2). The directed arc compels a clockwise path which encompasses a wildcard node. Node 3, belonging to both  $D$  relations, [2,3] and [1,3], can be interposed in the sequence compelled by the  $C$  relation to obtain feasible routing rules:  $1 \rightarrow 2 \rightarrow 3$ ,  $1 \rightarrow 3 \rightarrow 2$  and  $3 \rightarrow 1 \rightarrow 2$ .

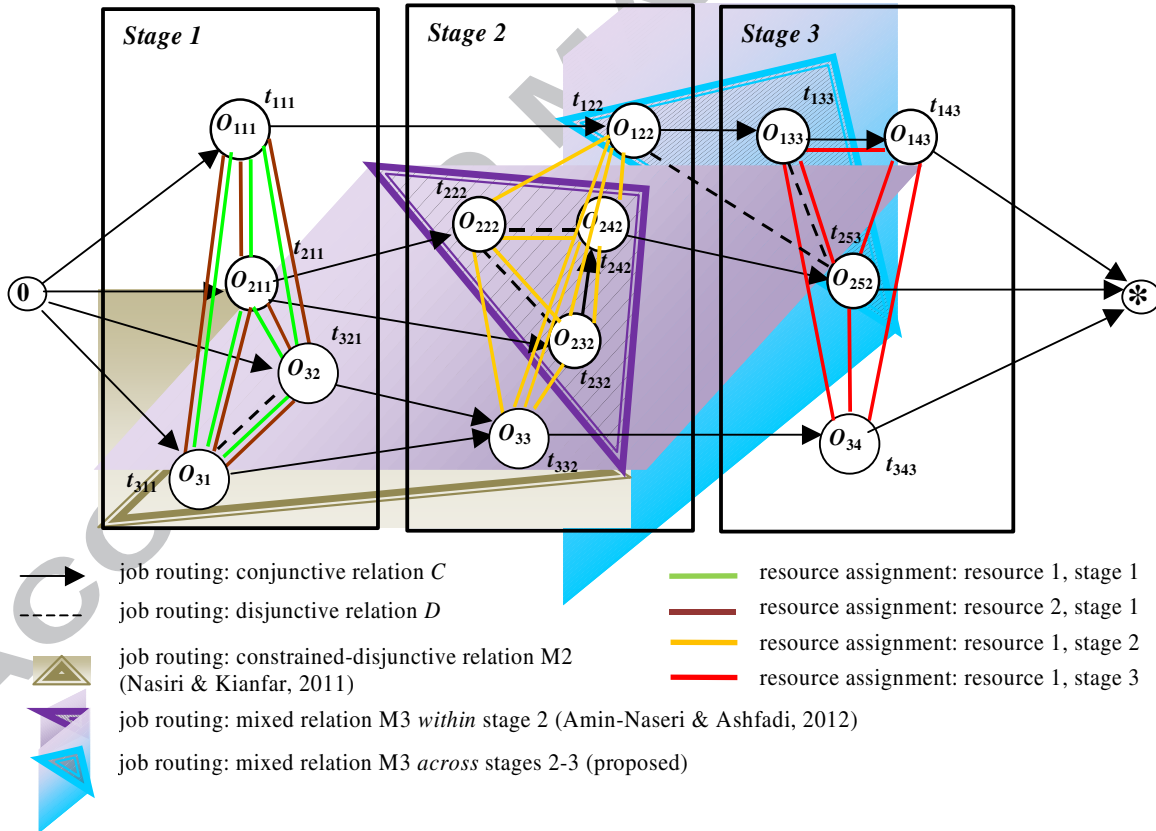
$M4$  includes all three disjunctive arcs (similarly to group shop scheduling); the feasible routing rules are all the six permutations of the three nodes.

An instance of the problem can be expressed in compact form by the tuple

$$P\text{-HSSSI}^{\alpha} | C^{\beta} | D^{\gamma} | M^{\delta} | N^{\epsilon} | S^{\phi} | \max L_{ij}^{\gamma} \quad (7)$$

where quoted symbols  $\alpha, \beta, \gamma, \delta, \epsilon, \phi$  and  $\gamma$  are values of the respective parameter (base).

An example is the triangular inset of Figure 3 derived from the  $2\text{-HSSSI}^{\alpha} | C^{\beta} | D^{\gamma} | M^{\delta} | N^{\epsilon} | S^{\phi} | \max L_{ij}^{\gamma}$  case in Figure 2.



**Figure 3. Examples of precedence relations on  $2\text{-HSSSI}^{\alpha} | C^{\beta} | D^{\gamma} | M^{\delta} | N^{\epsilon} | S^{\phi} | \max L_{ij}^{\gamma}$ : conjunctive relations  $C$  ( $(O_{111}, O_{122}), (O_{122}, O_{133}),$  etc.) and disjunctive relations  $D$  ( $[O_{222}, O_{232}], [O_{222}, O_{242}],$**

$[O_{311}, O_{321}]$ ) from Table 2; constrained-disjunctive relations  $M2$  and  $M3$  from Table 3 (respectively cases #4. and #5.) in triangular insets.

Figure 3 about here

### 5.1. Properties of basic relations

The combination of primitive relations has been shown to yield the mixed relations  $M1$  to  $M4$ . Their precedence graph encompasses nonlinear routing; consequently, a digraph (6) can be formed by mixed relations.

In this subsection, some properties of primitive and mixed relations are introduced to allow more complex composition and decomposition rules of digraphs (6).

*Proposition 1.* All the introduced relations can be represented by a digraph (6) with maximum level equal to 2.

As an example, we consider  $M3$ . Its digraph is:

$$DG(M3) = (\mathfrak{R}W_{\mathfrak{R}} C, D, E_{hk}) = (\{1,2,3\}_f, \{t_{i1}, t_{i2}, t_{i3}\}_f, \{(1,2)\}_f, \{[1,3], [2,3]\}_f, \{ \})$$

where, without loss in generality, the set of the disjunctive arcs of each resource  $E_{hk}$  is not considered.

*Lemma 2.* The union operator  $\cup$  is closed to the set of the primitive relations  $\Gamma$ .

For *Proposition 1*, if  $x, y \in \Gamma$  include an acyclic graph, also  $x \cup y$  includes no cycle on its graph. A cycle breaks the level of the node, which closes the path on itself. Obviously, only a  $C$  relation can close a cycle.

If  $x = (a,b) \in C$  and  $y \in \Gamma$ , then  $x \cup y$  becomes the mixed relation denoted by, respectively,  $M1$  (if  $y = (a,c) \in C$ ) and  $M2$  or  $M3$  (if  $y \in D$ ) relations. In fact, the level of nodes  $b$  and  $c$  forces either a disjunctive relation (obtaining the mixed relations  $M1$  and  $M3$ ) or a conjunctive relation (obtaining the mixed relation  $M2$ ).

Vice versa, if  $x = (b,a) \in C$  and  $y \in \Gamma$ , then  $x \cup y$  becomes, similarly to the previous case, the mixed relation denoted by, respectively,  $M2$  (if  $y = (a,c) \in C$ ) and  $M1$  or  $M3$  (if  $y \in D$ ) relations.

Revised manuscript

In any case,  $M1$  to  $M3$  relations are disjunctive graphs. Hence, considering the former primitive, i.e. the  $C$  relation, the thesis is accomplished.

Analogously, if  $x = [a,b] \in C$  and  $y \in \Gamma$  then  $x \cup y$  becomes the mixed relation denoted by, respectively,  $M1$  (if  $y = (c,a) \in C$ ,  $M2$  (if  $y = (a,c) \in C$ ) and  $M4$  (if  $y \in D$ ). Hence, the thesis is accomplished because  $M1$  and  $M2$  relations are digraphs.

■

*Corollary 1.* The level of the wildcard node inherits the levels of the nodes to which it is linked to (through  $D$  relations).

*Corollary 2.* The level of all the nodes connected only by  $D$  relations is unchanged (i.e.  $M4$ ).

## 5.2. Composition and decomposition of basic relations

In this subsection, it will be shown how to assemble mixed routing rules to form a nonlinear precedence graph and achieve more complex rules. A list of compositions of mixed relations is shown in Table 4. All possible outcomes of compositions are listed. The same outcome can be obtained by different compositions (not shown).

Compositions may involve nodes belonging to the same stage or to two consecutive stages; however, two consecutive stages can be linked by conjunctive arcs only. Decomposition rules can be inferred by following the opposite path.

To explain the composition of mixed relations shown in Table 4, case #12. is considered:

$M3 \cup M3$ . Both mixed relations have the same wildcard node 3. From *Proposition 1*, the two relations can be represented by the digraphs:

$$DG_1(M3) = (\{1,2,3\}, \{t_{i1}, t_{i2}, t_{i3}\}, \{(1,2)\}, \{[1,3], [2,3]\}, \{ \}) \quad (8)$$

$$DG_2(M3) = (\{2,3,4\}, \{t_{i2}, t_{i3}, t_{i4}\}, \{(2,4)\}, \{[2,3], [3,4]\}, \{ \}) \quad (9)$$

The composite relation is:

$$DG_1(M3) \cup DG_2(M3) = (\{1,2,3,4\}, \{t_{i1}, t_{i2}, t_{i3}, t_{i4}\}, \{(1,2), (2,4)\}, \{[1,3], [2,3], [3,4]\}, \{ \}) \quad (10)$$

which has a node (node 2) with one starting and one ending conjunctive arc. The nodes opposite to node 2 in these two conjunctive arcs have a level reduced by one or increased by one, respectively, if the node is on the conjunctive arc that ends to 2 and if the node is on the conjunctive arc that starts from 2. For *Corollary 1* the level of the wildcard node inherits the levels of the nodes to which it is linked to (through  $D$  relations):  $L_{i3}=\{L_{i1}, L_{i2}=L_{i1}+1, L_{i4}=L_{i1}+2\}$ . As a consequence, the routing rules are obtained by interposing 3 in the sequence compelled by the 2 conjunctive arcs, (1,2) and (2,4):  $3 \rightarrow 1 \rightarrow 2 \rightarrow 4$ ,  $1 \rightarrow 3 \rightarrow 2 \rightarrow 4$ ,  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ ,  $1 \rightarrow 2 \rightarrow 4 \rightarrow 3$ .

ACCEPTED MANUSCRIPT



Revised manuscript

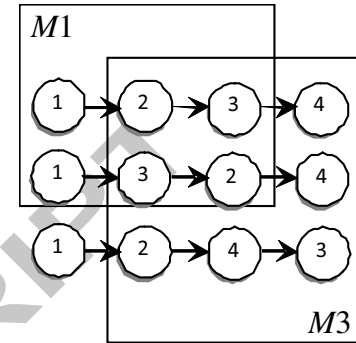
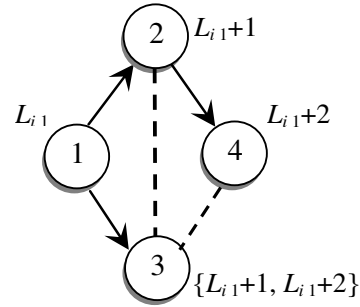
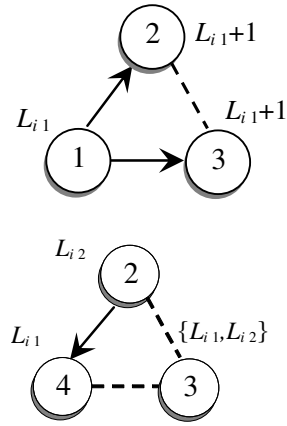
Table 4. Composition of mixed relations (4 nodes) and deployment of their routing rules

Case #	Relation	Combination of mixed relations	Precedence graph	Routing deployment
#7.	$M1 \cup M2$			

Revised manuscript

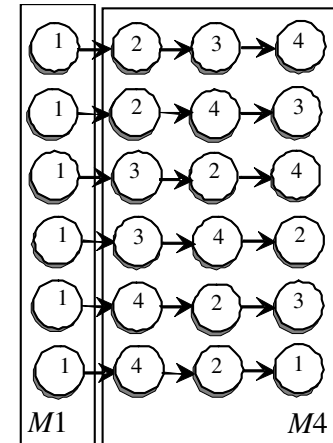
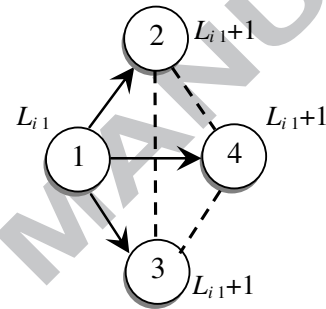
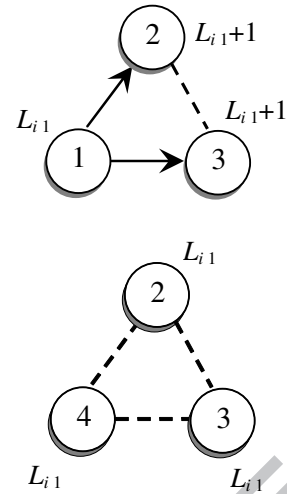
#8.

$M1 \cup M3$



#9.

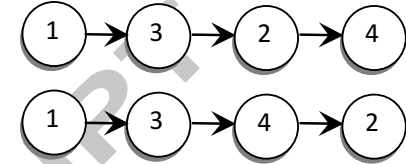
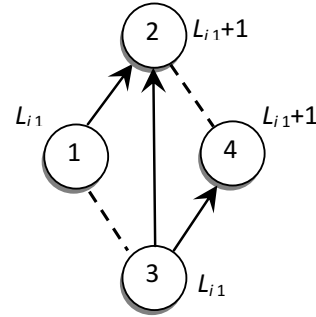
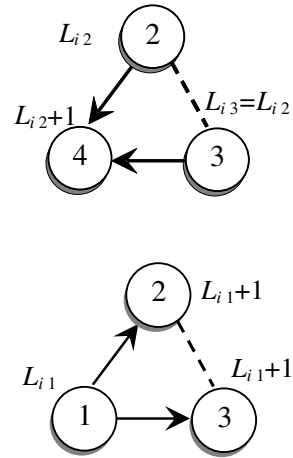
$M1 \cup M4$



Revised manuscript

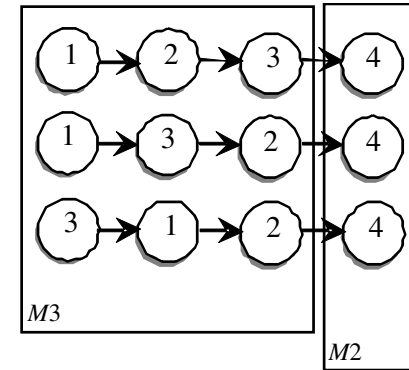
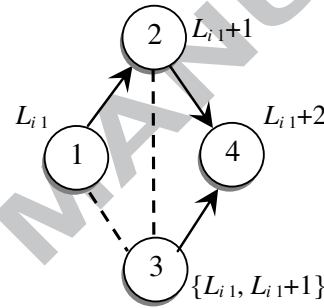
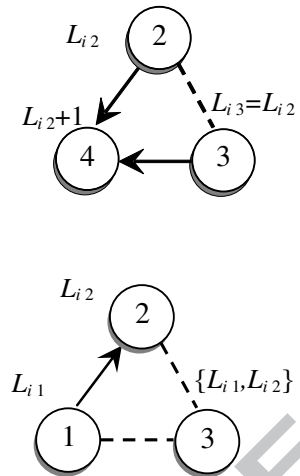
#10.

$M2 \cup M1$



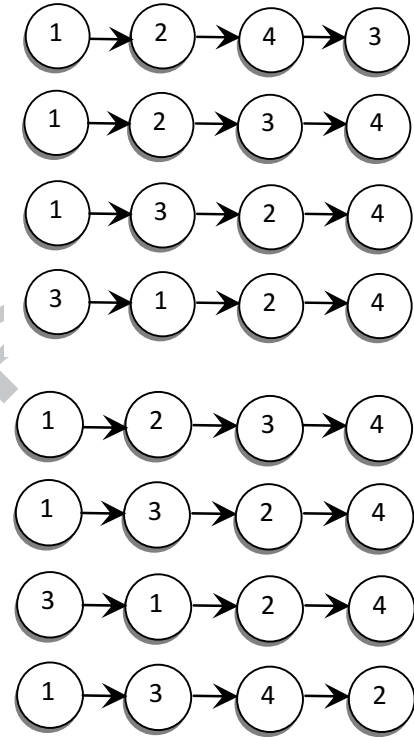
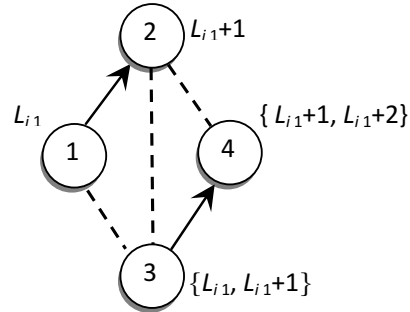
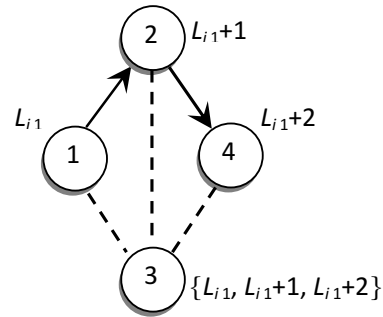
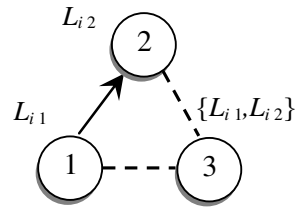
#11.

$M3 \cup M2$



#12.

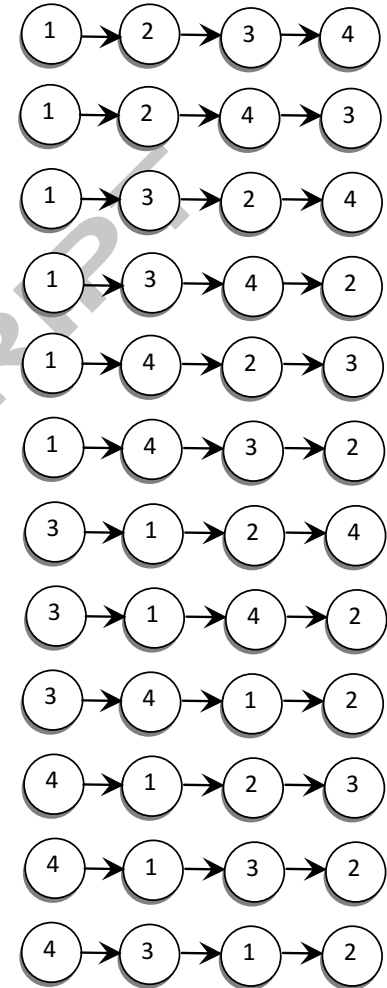
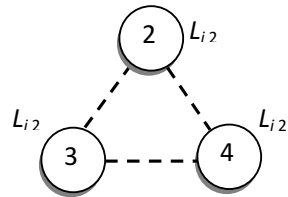
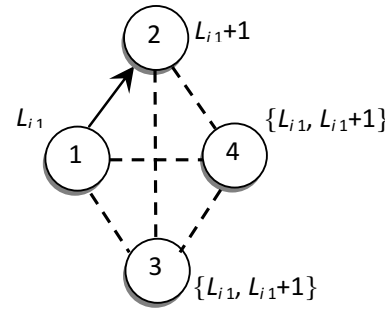
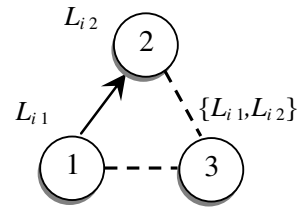
$M3 \cup M3$



Revised manuscript

#13.

$M3 \cup M4$



Revised manuscript

Composite relations can be recombined into two mixed relations  $x$  and  $y$  with conjunctive arcs from  $x$  to  $y$ , which cross the boundaries of the two relations in the  $x$ - $y$  direction (by construction), according to the following two alternatives (Table 5):

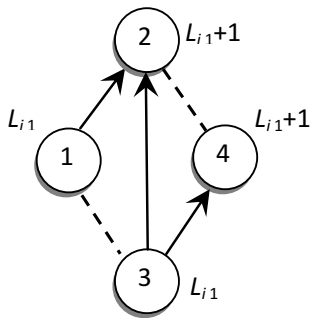
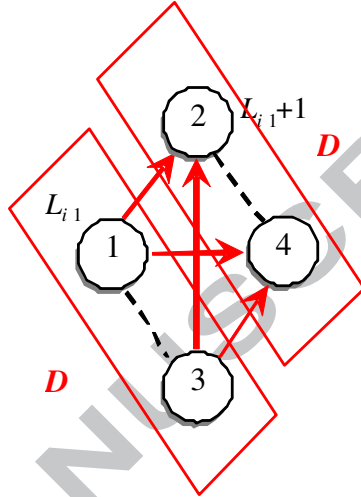
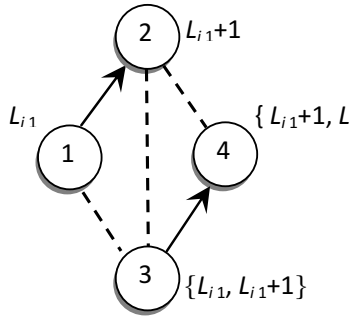
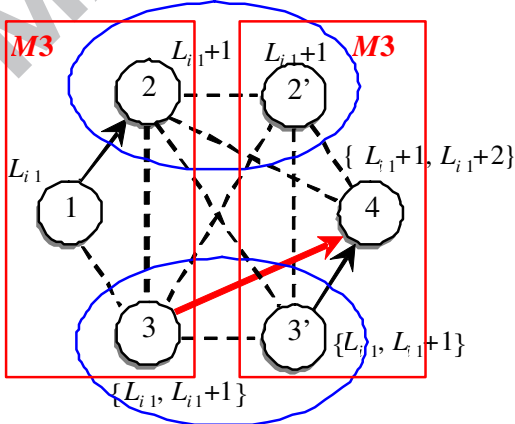
1. by a  $C$  relation, e.g. (3,2) in the precedence graph of row 1 for case #7. in Table 4;
2. by a  $D$  relation, e.g. [2,3] in row 2 for all other cases of Table 4 except case #7. (only case #13. is represented).

In the first alternative, the combined mixed rule is modeled by two  $D$  relations between nodes with the same level. The routing rule of this recombination is unchanged with respect to the original case #7. in Table 4.

In the second alternative, the combined mixed rule is modeled by splitting nodes with the same level and by connecting the split nodes by disjunctive arcs. Two mixed relations  $Mx$  are created. Each node of the  $D$  relation is split into two nodes, the original node and the dummy node, which are connected by a  $D$  relation. For example, node 2 is placed facing dummy node 2' and they are connected by arc [2,2']. The routing rule of the recombination imposes that only one node, between the original and the dummy node, will be included into the loading sequence of the selected resource. The not selected node will have no connection to the rest of the digraph and will not be considered by the routing rule.

In both alternatives, in the combined mixed rule the red arrows cross the boundaries of the two relations from left to right and no arrow is present in the opposite direction in order to process more complex relations which satisfy the properties introduced above.

**Table 5. Decomposition rules for the mixed relations of  $C$  type ( $M1 \cup M2$ ) and of  $D$  type ( $M3 \cup M3$ ), yielding respectively a  $D$  to  $D$  relation and two split  $M3$  relations.**

<i>Case</i>	<i>Precedence graph</i>	<i>Combined mixed rule</i>
#7.		 <p style="text-align: center;"><b>Splitting operation 2</b></p>
#12.		 <p style="text-align: center;"><b>Splitting operation 3</b></p>

## 6 A scheduling heuristic ( $H$ SSS)

This section describes an example heuristic to show the usability of the proposed HSSS model for scheduling purposes.

The developed algorithm is a list scheduler heuristic derived from Giffler and Thompson (1960), which was proven to generate a feasible schedule on the digraph, by visiting every node once and

only once. The list scheduler algorithm was originally proposed for classic job shop scheduling. At every step, a node is connected by means of a feasible move to the partially acyclic conjunctive graph, which represents the partial schedule. A feasible move is a disjunctive arc, which can be directed in the partial conjunctive graph without creating a cycle.

The following pseudo code produces an acyclic graph, which includes the critical path.

### **Heuristic for hybrid stage shop scheduling (H-HSSS)**

**Input:** a weighted digraph  $DG = (\mathcal{N}, C, D, E_{hk})$

$O \leftarrow \{O_{ijk} \mid i=1, \dots, N, j=1, \dots, l_i, k=1, \dots, S\}$

- i.  $AL_w^1 \leftarrow \{O_{ijk} \in O \mid O_{ijk} \cap O = \emptyset\}$
- ii.  $AL_w^2 \leftarrow \{O_{ijk} \in AL_w^1 \mid \min L_{ij}, i = 1, \dots, N\}$
- iii.  $ct_{ijk} = r_i$ , for each  $O_{ijk} \in AL_w^2$

**for each**  $k=1$  **to**  $S$  **do**

*Initialization of Candidate Nodes:* build, in two steps, the *allowed list*  $AL_w$  for the current iteration  $w$ :

- i.  $AL_w^1 \leftarrow \{O_{ijk} \in O \mid O_{ijk-1} \cap O = \emptyset\}$
- ii.  $AL_w^2 \leftarrow \{O_{ijk} \in AL_w^1 \mid \min L_{ij}, i = 1, \dots, N\}$

**for each**  $w=1$  **to**  $(\sum_{i=1, \dots, N} l_i)$  **do**

1. *Initialization of Feasible Moves:* mark as a *feasible move* each disjunctive arc  $(O_{i'j'k}, O_{ijk})$  of  $E_{hk}$  where  $O_{ijk} \in AL_w^2$  and  $O_{i'j'k}$  is the last operation of the loading sequence of resource  $h$  of the stage  $k$  (it creates the possibility for the candidate operation to become the new last operation of that loading sequence);
2. *Move Selection:* select a feasible move  $(O_{i'j'k}, O_{ijk})$  of  $E_{hk}$  by directing the related disjunctive arc  $(O_{i'j'k}, O_{ijk}) = \text{'dummy 0'}$ , if  $k=1$ );
3. *Update Routing:* if  $L_{ij} = L_{ij^*}$ , where  $O_{ij^*k}$  is in the path which starts from 0 and ends at  $O_{ijk}$ , direct the last disjunctive arc in the path  $[O_{ij^*k}, O_{ijk}] \in D$  (after this action  $(O_{ij^*k}, O_{ijk}) \in C$ ) and update  $L_{ij} = L_{ij^*} + 1$ ; finally, update the level on all the disjunctive arcs connected to  $O_{ijk}$  (i.e.  $L_{ij^{**}} = L_{ij} + 1$ , if  $[O_{ijk}, O_{ij^{**}k}] \in D$ );



4. *Arcs Removal*: remove all the remaining disjunctive arcs connected to  $O_{i'j'k}$  (i.e. no other operation can be immediately subsequent to  $O_{i'j'k}$  in the loading sequence); remove all the remaining disjunctive arcs of  $E_{h'k}$  connected to  $O_{ijk}$ , i.e.  $h' \neq h$  (i.e. no other loading sequence can include the operation); finally, remove all the remaining disjunctive arcs of  $D$  connected to  $O_{ij^*k}$  (i.e. the routing must be linear after updating);
5. *Computing Length*: the length of the arc  $(O_{i'j'k}, O_{ijk})$  is evaluated as the processing time of node  $O_{ijk}$ ; this length is placed on the arc  $(O_{i'j'k}, O_{ijk})$  and on the arc of the job routing  $(O_{ij^*k}, O_{ijk})$  which ends at  $O_{ijk}$ ; also, the completion time  $ct_{ijk} = \max\{ct_{ij^*k}, ct_{i'j'k}\} + t_{ijk}$  is placed as a mark of the node  $O_{ijk}$ , if  $L_{ij^*} < L_{ij}$ ; otherwise, if  $L_{ij^*} = L_{ij}$  and two resources are available,  $\min\{T_{k'}, ct_{ij^*k}\} + t_{ijk}$ ,  $k' \neq k$ , where  $T$  is the available time of resource  $k'$ ;
6. *Updating Structures*: update  $O$  by removing operation  $O_{ijk}$ :  $O \leftarrow O \setminus \{O_{ijk}\}$ ;  
**end for**
7. *Directing the remaining disjunctive arcs*: the arcs are connected to the dummy node \*;

**Output**: the acyclic conjunctive graph with the completion times for all the operations

*Lemma 3.* The algorithm generates in:

$$O(Nl_i(\frac{l_i}{S} + P + \frac{l_i}{S})) \quad (11)$$

a *complete selection* of arcs of  $DG$  i.e. an acyclic conjunctive graph that includes all nodes.

This property results from the following considerations:

- a) the achieved graph includes all the nodes: the main loop is performed  $|O|$  times and initially the candidate list includes all the nodes; at each iteration one and only one node is removed from candidate list (step 6);
- b) the achieved graph is conjunctive: i) for each iteration, the selected feasible move directs an arc of  $E$  which ends at the node removed from candidate list (step 2); ii) for each iteration, the node removed from candidate list is inserted in the path which starts from 0 (step 3); iii)

- all the disjunctive arcs of  $E$  and  $D$  which starts from the starting node of the conjunctive arc are removed (step 4);
- c) the conjunctive graph is acyclic: i) one and only one feasible move ends to a node which is in the candidate list (steps 1, 2 and 6); ii) one and only one routing rule ends to a node which is in the candidate list (steps 3 and 6).

In order to evaluate the computational complexity of the algorithm, it can be noted that the main loop is performed  $n \cdot m$  times and the most time-consuming steps are Arcs Removal and Computing Length step. The selection of a feasible move ( $O_{i'jk}, O_{ijk}$ ) entails that the following alternative arcs are removed and the following connected nodes are marked:

- i) *alternative sequencing arcs*: all disjunctive arcs connected to the last operation  $O_{i'jk}$  in the loading sequence of resource  $h$ ; they are at most  $N-1$ , one for each alternative job in order to approach the sequencing problem (the first member of expression (11complessità));
- ii) *alternative assigning arcs*: all disjunctive arcs of  $E_{h'k}$ , where  $h'$  belonging to the stage  $k$  and  $h' \neq h$ , connected to the candidate operation  $O_{ijk}$ ; they are at most  $P-1$ , one for each alternative resource in order to approach the assigning problem (the second member of expression (11complessità));
- iii) *selecting nodes level*: all nodes  $O_{ijk}$ , where  $L_{ij*} \leq L_{ij}$ , have marked with the completion time  $ct_{ijk} = \max\{ct_{i'j*k}, ct_{ij*k}\} + t_{ij}$  (the third member of expression (11complessità)).

■

Because of these computational complexity considerations, the proposed algorithm finds a feasible schedule by means of an implicit visit of a large number of disjunctive arcs. Another consequence is that the three sequencing, assigning and selecting decisional points are considered at the same time in the selection of a feasible move because, at the same time, it is both an alternative sequencing arc and alternative assigning arc with the selected level.

## 7 Case study

The case considered from the fashion industry includes five stages (Soldani, Rossi, & Lanzetta, 2013):

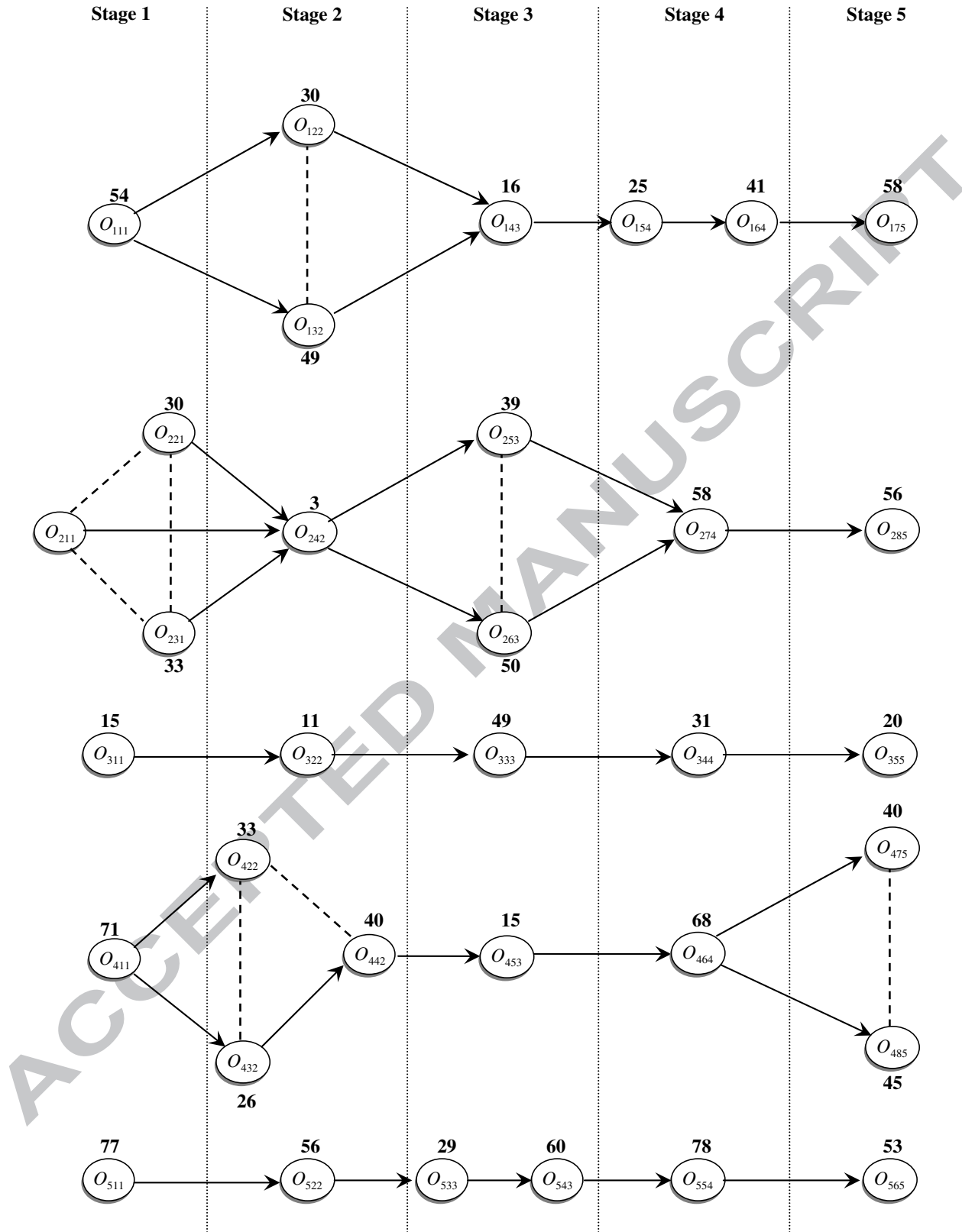
1. cutting – manual, die and machine cutting;

Revised manuscript

2. preparation – primary and secondary part preparation;
3. varnishing;
4. assembly – manual and machine stitching;
5. finishing – finishing operations and packaging.

Jobs include single parts (€25.000 woman bags) or batches.

A 5x5 benchmark, with 5 stages and 5 jobs, characterized as 2-HSSSI<sup>34</sup>|C<sup>23</sup>|D<sup>6</sup>|R<sup>10</sup>|N<sup>5</sup>|S<sup>5</sup>|max  $L_{ij}$ <sup>6</sup>, by a total of 34 operations with 23 conjunctive arcs, 6 disjunctive arcs (6/34=17%) and a maximum number of levels of 6. The simplest non-trivial case of  $P=2$  parallel (or identical) resources per stage has been considered, to make it a hybrid stage shop scheduling benchmark. The release dates for jobs and resources are zero. The case 1-HSSSI<sup>34</sup>|C<sup>23</sup>|D<sup>6</sup>|R<sup>5</sup>|N<sup>5</sup>|S<sup>5</sup>|max  $L_{ij}$ <sup>6</sup>, with  $P=1$  machine per stage has also been analyzed for comparison and can be considered as a special case of hybrid stage shop scheduling. Consequently, the general H-HSSS pseudo code can be applied to both benchmarks.



**Figure 4. Proposed  $P$ -HSSS| $N^4$ | $C^{23}$ | $D^6$ | $R^{5P}$ | $N^5$ | $S^5$ | $\max L_{ij}$ <sup>6</sup> benchmark with 5 stages (columns) and 5 jobs (rows), including the precedence relations in Table 2 and Table 3. Colored arcs for  $P$  not shown for clarity.**

Figure 4 about here

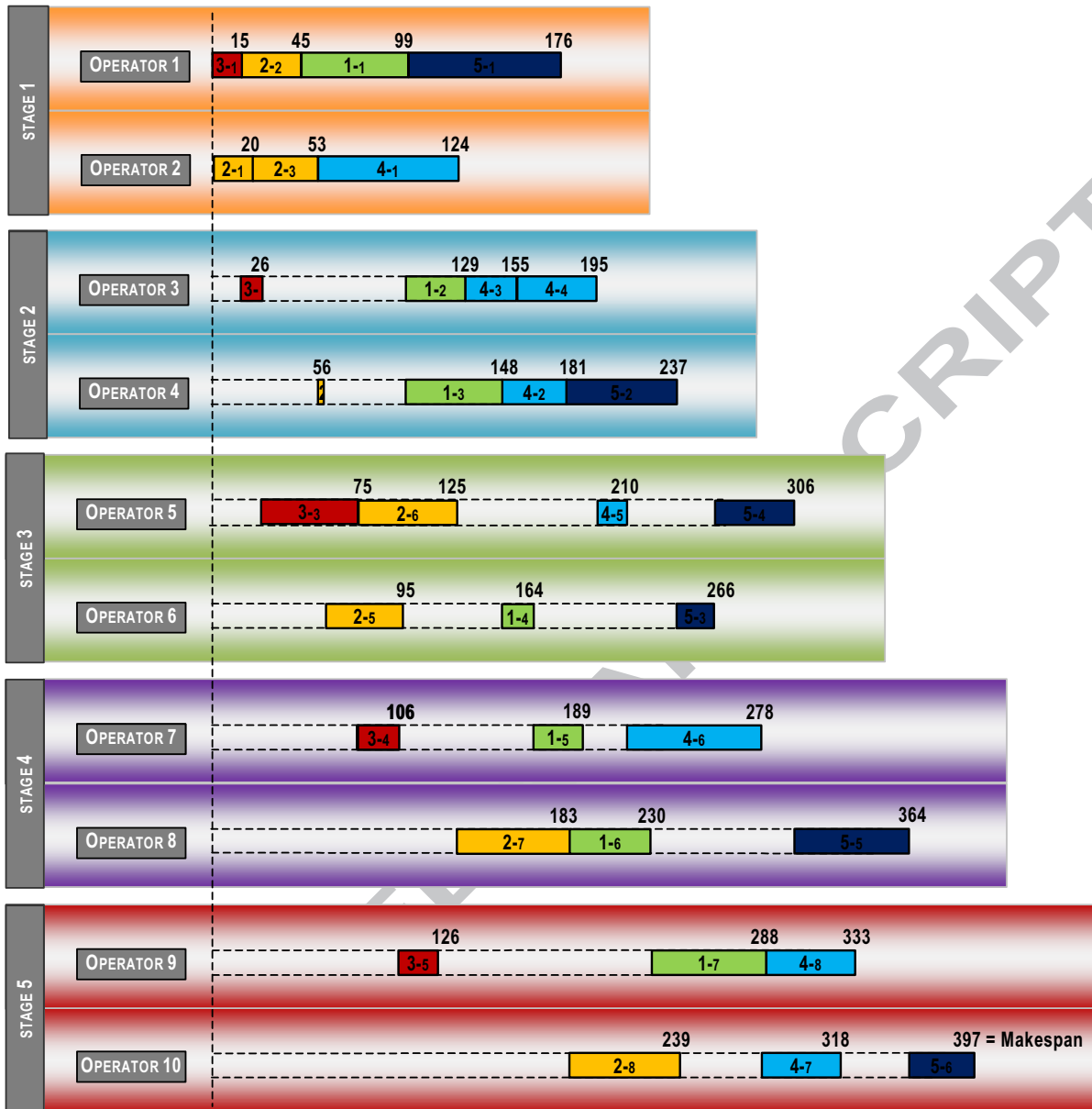
The benchmark in Figure 4 has been designed in order to include all the precedence relations introduced above, and recalled in Table 6; inter-stage relations are denoted by the two corresponding consecutive stage numbers, consequently it can be noticed there is no biunivocal relation between stages and primitive relations.

**Table 6. Map of job routing per precedence relations for the benchmark in Figure 4**

JOB	STAGE								
	1	1-2	2	2-3	3	3-4	4	4-5	5
1	Node	$M1$	$D$	$M2$	Node		$C$		Node
2	$M4$		Node	$M1$	$D$	$M2$	Node		Node
3	Node		Node		Node		Node		Node
4	Node	$M1$	$M3$		Node		Node	$M1$	$D$
5	Node		Node		$C$		Node		Node

Table 6 shows 6 occurrences of precedence graphs spanning across two stages. By applying the  $H$ -HSSS pseudo code on the two benchmarks considered, the Gantt diagrams of the obtained schedule are shown in Figure 5.

Revised manuscript



Revised manuscript

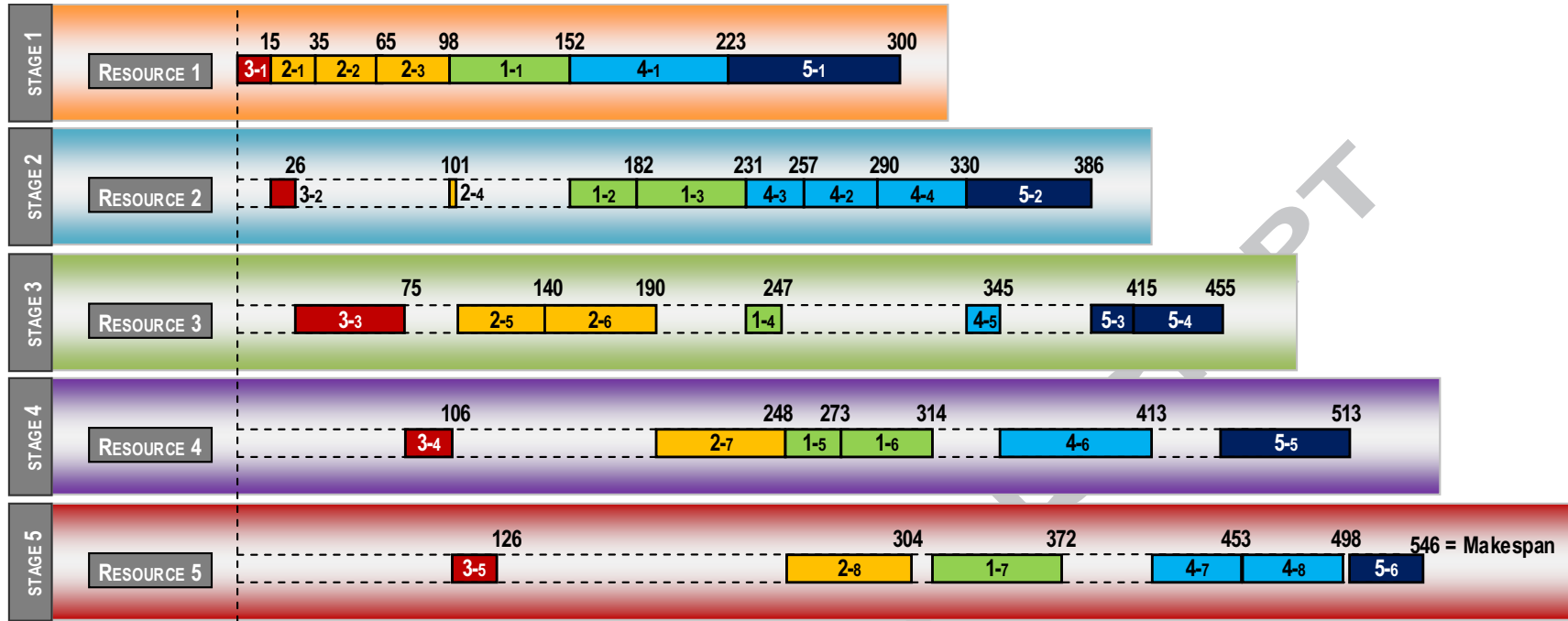


Figure 5. Gantt for the 2-HSSSI $\mathcal{R}^4|\mathcal{C}^{23}|\mathcal{D}^6|\mathcal{R}^{10}|\mathcal{N}^5|\mathcal{S}^5|\max L_{ij}^6$  (top) and for the the 1-HSSSI $\mathcal{R}^4|\mathcal{C}^{23}|\mathcal{D}^6|\mathcal{R}^5|\mathcal{N}^5|\mathcal{S}^5|\max L_{ij}^6$  (bottom) benchmark in Figure 4. Larger numbers in colored bars express jobs with their respective operations (smaller numbers) and completion time (superscripts)

Figure 5 about here

At stage 1 in the top Gantt of Figure 5 for the 2-HSSS benchmark, the routing rule of the mixed relation  $M4$  allows the overlapping among (micro) operations of job 2, renamed as 2-1, 2-2 and 2-3 for readability. Resources 1 and 2 overlap in processing, respectively, operations 2-1 and 2-2, while operation 2-3 is processed on resource 1 as soon as the resource becomes available.

The three mixed relations  $M1$  between stage 1 and 2, between stage 2 and 3 and between stage 4 and 5 are merged in three composite relations  $M1 \cup M2$ . All these relations are managed as  $D$  relations (also see Table 6). The corresponding routing rule allows overlapping operations 1-1 and 1-2 on the alternative resources of stage 2; analogously, for  $D$  relations of stage 3 and 5.

The routing rule of the mixed relation  $M3$  of job 4 at stage 2 is scheduled by overlapping operations 4-2 and 4-3 on the two alternative resources; operation 4-4 is processed after operation 4-3 has been completed, because of the conjunctive relation (4-3, 4-4) included in the mixed relation  $M3$ . The adopted routing rule is  $4-3 \rightarrow 4-4$ ; it is only applied to resource 1, because operation 4-2 is processed on resource 2.

At stage 1 in the bottom Gantt of Figure 5 for the 1-HSSS benchmark, the routing rule of the mixed relation  $M4$  does not allow the overlapping among (micro) operations 2-1, 2-2 and 2-3, which are processed sequentially by resource 1. Similarly, the three mixed relations  $M1$ , including operations 1-2 and 1-3 at stage 2, 4-3, 4-2 and 4-4 at stage 2 and 2-5 and 2-6 at stage 3, which are managed as  $D$  relations (also see Table 6), are processed sequentially by resource 1. The routing rule of the mixed relation  $M3$  of job 4 at stage 2 is processed sequentially by resource 1 with routing 4-3, 4-2 and 4-4. It can be noticed that splitting operations does not affect the completion time on each resource, because only permutations are possible. Allowed permutations are according to the selected routing rules of Table 2, Table 3 and Table 4. Inversely, the presence of two or more resources allows further potential makespan reduction by overlapping.

As designed, the proposed heuristics provides feasible schedules. The Move Selection at step 2. can be improved by the use of dispatching rules to obtain better solutions.

It can also be observed that the proposed theory (model, description and code) is applicable also with only one resource per stage and that it has been profitably applied to the examined case.

## 8 Conclusion

This work has been inspired by a real manufacturing case from the high-fashion industry. A hybrid stage shop scheduling model has been proposed. The problem has been described by mixed integer linear programming. The solution has been formalized by a disjunctive graph representation, a powerful tool to design scheduling optimization (metaheuristics) algorithms. An  $O(P \cdot S \cdot N^2)$



constructive method (heuristics) to generate a feasible plan/schedule in the most general case has been proposed and applied to a case study; it is based on priority levels obtained from the precedence graphs by deploying alternative routings. To find an optimal solution, metaheuristics like ant colony optimization, iterated local search and tabu search that produce constructive solutions may benefit of the graph representation.

The hybrid stage shop scheduling model is able to provide shorter scheduling, by overlapping operations and by splitting on the same job, increasing the number of scheduling alternatives. These advantages, obtained at the cost of increased complexity, require resource and organization flexibility: more complex material flow and multifunctional resources (machines and/or operators). The proposed hybrid stage shop scheduling model completes the mixed stage shop model by resource flexibility and by integrating process planning and group shop scheduling (operational and routing flexibility). Greater flexibility and shorter makespan are available versus AND/OR graphs. Other contributions include the addition of operation overlapping and splitting within stages and across consecutive stages with the composition and the decomposition of primitive (2 nodes) and mixed (3 and 4 nodes) relations. The universe of new more complex relations has been explored and discussed both concerning the application implication and the theoretical background. The wildcard node mechanism is the key concept to fully exploit the potential of parallel resources, by allowing alternative plans obtained by permutation of operations linked by undefined or arbitrary routing, modeled as immaterial precedence constraints. Immaterial links, allow the integration of process planning and scheduling, with possible relapse in assembly/disassembly, computer aided design and process planning, manufacturing and service scheduling, and more generally in project management.

## 9 Future work

Mixed relations of the hybrid stage shop scheduling are included in a single stage or may span across two consecutive stages. The extension of immaterial constraints to multiple stages in hybrid shop scheduling (from HSSS to HSS), represents a new research area. The most general case of precedence graphs involving more than two consecutive stages has been examined and solved in a case study. Combining further the new mixed type relation, more complex nonlinear relations (5+ nodes) can be speculated both on the theoretical and algorithmic sides. Among developments is the unifying theory for  $\alpha$ nodes and  $\chi$ immaterial relations (and  $\gamma$ levels).

An alternative interpretation of immaterial precedence constraints is considering resources as in a *local* job shop. In a perspective where the flowshop can be considered as a special case of the job

shop, the hybrid stage shop scheduling model can be considered as a smooth transition between a flowshop and the more general case of job shop. In this regard the mentioned  $\chi$  parameter may represent a measure of the *flow grade* of a given job shop. From a practical viewpoint, job shop solutions can be transferred from the literature.

## Acknowledgements

The authors would like to thank the management of the anonymous high-fashion company from a reputed international group for providing the information about their manufacturing process.

## References

- Ahmadizar, F., & Shahmaleki, P. (2014). Group shop scheduling with sequence-dependent setup and transportation times. *Applied Mathematical Modelling*, 38, 5080-5091.
- Aloulou, M.A., & Artigues, C. (2010). Flexible solutions in disjunctive scheduling: General formulation and study of the flow-shop case. *Computers & Operations Research*, 37, 890-898.
- Amin-Naseri, M.R., & Afshari, A.J. (2012). A hybrid genetic algorithm for integrated process planning and scheduling problem with precedence constraints. *International Journal of Advanced Manufacturing Technology*, 59, 273-287.
- Behnamian, J., & Zandieh, M. (2011). A discrete colonial competitive algorithm for hybrid flowshop scheduling to minimize earliness and quadratic tardiness penalties. *Expert Systems with Applications*, 38, 14490-14498.
- Benavides, A. J., Ritt, M., & Miralles, C. (2014). Flow shop scheduling with heterogeneous workers. *European Journal of Operational Research*, 237, 713-720.
- Bensmaine, A., Dahane, M., & Benyoucef, L. (2013). A non-dominated sorting genetic algorithm based approach for optimal machines selection in reconfigurable manufacturing environment. *Computers & Industrial Engineering*, 66, 519-524.
- Blazewicz, J., & Kobler, D. (2002). Review of properties of different precedence graphs for scheduling problems. *European Journal of Operational Research*, 142, 435-443.
- Bentaha, M. L., Battaïa, O., Dolgui, A., & Hu, S. J. (2014). Dealing with uncertainty in disassembly line design. *CIRP Annals-Manufacturing Technology*, 63, 21-24.
- Blum, C., & Sampels, M. (2004). An Ant Colony Optimization Algorithm for Shop Scheduling Problems. *Journal of Mathematical Modelling and Algorithms*, 3, 285-308.

- Dauzère-Pérès, S., Roux, W., & Lasserre, J.B. (1998) Multi-resource shop scheduling with resource flexibility. *European Journal of Operational Research*, 107, 289-305.
- Doh, H. H., Yu, J. M., Kim, J. S., Lee, D. H., & Nam, S. H. (2013). A priority scheduling approach for flexible job shops with multiple process plans. *International Journal of Production Research*, 51, 3748-3764.
- Ferrell, W., Sale, J., Sams, J., & Yellamaraju, M. (2000). Evaluating simple scheduling rules in a mixed shop environment. *Computers & Industrial Engineering*, 38, 39–66.
- Giffler, D., & Thompson, G.L. (1960) Algorithms for solving production scheduling problems. *Operation Research*, 8, 487–503.
- Go, T. F., Wahab, D. A., Rahman, M. A., Ramli, R., & Hussain, A. (2012). Genetically optimised disassembly sequence for automotive component reuse. *Expert Systems with Applications*, 39, 5409-5417.
- Kacem, I., Hammadi, S., & Borne, P. (2002). Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 32, 1-13.
- Kis, T. (2003). Job-shop scheduling with processing alternatives. *European Journal of Operational Research*, 151, 307–332.
- Kozan, E., & Liu, S.Q. (2012) A demand-responsive decision support system for coal transportation. *Decision Support Systems*, 54, 665-680.
- Leu, M.C., Bernard, A., ElMaraghy, H.A., Nee, A.Y.C., Ong, S.K., Lanzetta, M., Putz, M., & Zhu, W. (2013). CAD Model Based Virtual Assembly Simulation, Planning and Training. *CIRP Annals-Manufacturing Technology*, 62, 799-822.
- Leung, C.W. (2010). Integrated process planning and scheduling by an agent-based ant colony optimization. *Computers & Industrial Engineering*, 59, 166–180.
- Li, D., Meng, X., Liang, Q., & Zhao, J. (2015). A heuristic-search genetic algorithm for multi-stage hybrid flow shop scheduling with single processing machines and batch processing machines. *Journal of Intelligent Manufacturing*. Article in press.
- Liu, S.Q., Ong, H.L., & Ng, K.M. (2005). A fast tabu search algorithm for the group shop scheduling problem. *Advances in Engineering Software*, 36, 533-539.
- Masuda, T., Ishii, H., & Nishida, T. (1985) The mixed shop scheduling problem. *Discrete Applied Mathematics*, 11, 175-186.
- Nasiri, M.M., & Kianfar, F. (2011). A GA/TS algorithm for the stage shop scheduling problem. *Computers & Industrial Engineering*, 61, 161–170.

- Otto, C., & Otto, A. (2014). Multiple-source learning precedence graph concept for the automotive industry. *European Journal of Operational Research*, 234, 253-265.
- Phanden, R. K., Jain, A., & Verma, R. (2013). An approach for integration of process planning and scheduling. *International Journal of Computer Integrated Manufacturing*, 26, 284-302.
- Ribas, I., Leisten, R., & Framinan, J.M. (2010). Review and classification of hybrid flowshop scheduling problems from a production system and a solutions procedure perspective. *Computers & Operations Research*, 37, 1439–1454.
- Rossi, A. (2014). Flexible job shop scheduling with sequence-dependent setup and transportation times by ant colony with reinforced pheromone relationships. *International Journal of Production Economics*, 153, 253-267.
- Rossi, A., & Lanzetta, M. (2013). Scheduling Flow Lines with Buffers by Ant Colony Digraph. *Expert Systems with Applications*, 40, 3328-3340.
- Rossi, A., & Lanzetta, M. (2014). Native Metaheuristics for Non-Permutation Flowshop Scheduling. *Journal of Intelligent Manufacturing*, 25, 1221–1233.
- Rossi, A., Pandolfi, A., & Lanzetta, M. (2013). Dynamic set-up rules for hybrid flow shop scheduling with parallel batching machines. *International Journal of Production Research*, 52, 3842-3857.
- Ruiz, R., & Maroto, C. (2005) A comprehensive review and evaluation of permutation flowshop heuristics. *European Journal of Operational Research*, 165, 479–494.
- Shakhlevich, N.V., Sotskov, Y.N., & Werner, F. (2000). Complexity of mixed shop scheduling problems: A survey. *European Journal of Operational Research*, 120, 343–351.
- Soldani, S., Rossi, A., & Lanzetta, M. (2013). Modellazione di un ciclo produttivo nel settore moda, *Automazione Integrata*, 16, 37-39.
- Stecke, K. E., & Raman, N. (1995). FMS planning decisions, operating flexibilities, and system performance. *Engineering Management, IEEE Transactions on*, 42, 82-90.
- Tan, W., & Khoshnevis, B. (2000). Integration of process planning and scheduling—a review. *Journal of Intelligent Manufacturing*, 11, 51-63.
- Whizzkids '97, <http://www.win.tue.nl/whizzkids/1997>.