

# New effective MILP models for PFSPs arising from real applications

Tamás Hajba · Zoltán Horváth

Published online: 4 August 2012  
© Springer-Verlag 2012

**Abstract** Permutation flow shop problems (PFSPs) with makespan minimization that model production lines working in industry often have some special features: they are typically large-scale and the jobs can be sorted into types so that jobs of the same type have equal processing time values at each machine. We define the related R-PFSP, the Permutation with Repetition Flow Shop Problem, which is of less complexity if the number of types is bounded. Moreover, a subproblem set of R-PFSPs, the RL-PFSP is considered too, where only those permutations are in the design space in which subsequent tuples of a certain size contain jobs of the same type. We construct adequate new MILP models for R-PFSPs and RL-PFSPs and investigate their effectiveness experimentally. We demonstrate that via our new MILP models significantly larger problems can be solved than via the classical MILP models.

**Keywords** Flow shop scheduling · Minimizing makespan · Mixed integer programming formulations · Computational comparisons

**Mathematics Subject Classification (2000)** 90C11 · 90C90

## 1 Introduction

At modern factories production line planning is supported by some optimization tools to find the best or at least a “good” scheduling of the prescribed set of jobs. The time

---

T. Hajba (✉) · Z. Horváth  
Department of Mathematics and Computer Science,  
Széchenyi István University, Egyetem tér 1., Győr 9026, Hungary  
e-mail: hajbat@sze.hu

Z. Horváth  
e-mail: horvathz@sze.hu

available for this planning is often restricted to a given, rather short period and the problem is large-scale, which makes the optimization hard.

In this paper we consider the same kind of situation: within restricted computational time we have to find a good plan for production lines. We shall solve the corresponding permutation flow shop problems (PFSPs) to minimize the makespan (total processing time) of the given set of jobs.

In the literature, besides many heuristic methods (genetic algorithms, etc.) mixed integer linear programming (MILP) models and some numerical solvers are applied, too; see the papers [Pan \(1997\)](#), [Stafford \(1988\)](#), [Stafford and Tseng \(1990, 2002\)](#), [Stafford et al. \(2004, 2005\)](#), [Wagner \(1959\)](#), and [Wilson \(1989\)](#).

Although the main advantage of this approach compared to heuristics is that the MILP minimization gives a result with the guaranteed exact optimum, it cannot complete the computations within the prescribed period. Namely, since the PFSP is NP-hard when the number of machines is at least 3, we can not expect large-scale problems to be solvable in reasonable time by using MILP models. Indeed, classical benchmark problems with 10 machines and 20 jobs ([Taillard 1993](#)) can not be solved via the state-of-the-art MILP models within 1 h of computation on one CPU-core with the standard GAMS CPLEX.

In this paper we wish to extend the size of the solvable MILP models for the special problems arising from industry by exploiting their nature. Namely, the jobs can often be sorted into types so that jobs of the same type have equal processing time values at each machine. We define the related R-PFSP, the Permutation with Repetition Flow Shop Problem, which is of less complexity if the number of types  $T$  is bounded. Indeed, the design space for the PSFP has  $N! \approx (N/e)^N$  elements while the same number for R-PFSP is  $\approx T^N$ . In addition, a subproblem set of R-PFSPs is considered as well where only those permutations are in the design space in which subsequent tuples of a certain size contain jobs of the same type; in other words these tuples are called lots, so this problem class is called Permutation with Repetitions in Lots Flow Shop Problem and is denoted by RL-PFSP.

We construct adequate new MILP models for R-PFSPs and RL-PFSPs and investigate their effectiveness experimentally. We demonstrate that via our new MILP models significantly larger problems can be solved than via the classical MILP models.

Although both problems are widely used in real-life manufacturing systems, the R-PFSP has not been investigated in the literature. The RL-PSFP is a special case of the group scheduling problem and was studied in more general form in several papers ([Schaller et al. 2000](#); [Pinedo 2008](#); [Franca et al. 2005](#); [Salmasi et al. 2010](#)). However, the MILP models of the RL-PFSP have not received much attention so far and the empirical analysis of different MILP models for the RL-PFSP was not done in these papers.

The rest of the paper is organized as follows. We introduce the R-FPSP and the RL-PFSP in detail in Sect. 2. Then, in Sect. 3, three new MILP models, derived from the state-of-the-art MILP models for PFSP, are constructed for both problems. We present the results of numerical experiments devoted to benchmarking the new MILP models on the new problems in Sect. 4. The aim of the comparison is to investigate the maximum size of the problems that are solvable by these models. We also investigate how close the RL-PFSP optimum values are to those of the corresponding R-PFSP,

which has not been studied in the literature yet. We investigate a real life industrial problem in Sect. 5. Section 6 contains our conclusions.

## 2 Definition of the classical PFSP and the novel R-PFSP and RL-PFSP

The PFSP can be defined as follows. A set of  $n$  jobs has to be processed on  $m$  machines  $M_1, M_2, \dots, M_m$ . Each job has to be processed first on  $M_1$ , next on  $M_2$  and so on. A machine can process at most one job at any time and preemption is not allowed. We assume that there is unlimited buffer between the machines which means that a job can immediately leave a machine after its processing on that machine is completed. The schedule of the jobs must be the same on each machine. The goal is to find a permutation of the jobs which minimizes the completion time (or makespan) of the last job on the last machine.

In manufacturing problems we often have to schedule a lot of jobs but the jobs are not all different. We will say that two jobs,  $j$  and  $k$  are of the same *type*, we denote this by  $\mathcal{T}(j) = \mathcal{T}(k)$ , if their processing times are equal on any machines. Let us denote by  $T$  the number of different types and by  $n_t$  the number of jobs of type  $t$  ( $1 \leq t \leq T$ ). We will call this problem the *Permutation with Repetition Flow Shop Problem* (R-PFSP).

One of the possibilities to reduce the complexity of the minimization is the reduction of the design space by confining ourselves only to those permutations  $\pi = (\pi(1), \pi(2), \dots, \pi(n))$  in which  $\mathcal{T}(\pi(1)) = \mathcal{T}(\pi(2)) = \dots = \mathcal{T}(\pi(S))$ ,  $\mathcal{T}(\pi(S+1)) = \mathcal{T}(\pi(S+2)) = \dots = \mathcal{T}(\pi(2S))$  and so on, i.e. where the jobs in each consecutive  $S$ -tuples are of the same type. If this holds true for  $\pi$ , we call  $\pi$  a permutation with lot size  $S$ . We call those R-PFSPs, for which the design space is non-empty and consists of permutations with lot size  $S$ , *Permutation with Repetition Flow Shop Problems of lot size  $S$* ; we denote the set of these problems by RL-PFSP. Thus, e.g. the classical PFSP is equivalent to a RL-PFSP with  $S = 1$  and  $T = n$ . We remark that both models, R-PFSP and RL-PFSP, are frequently used in production planning.

## 3 Construction of MILP models for the R-PFSP and the RL-PFSP

### 3.1 MILP models for the R-PFSP

The classical PFSP has several MILP models, which can be divided into two classes, namely the models of the “Wagner” family (Wilson model, WST model, TS2 model) and the models of the “Manne” family (Manne model, Liao-You model).

The models of the Wagner family use the classical assignment problem to describe a permutation while the models of the Manne family use dichotomous constraints to describe a permutation. Numerical experiments (Stafford et al. 2004, 2005) show that the models of the Wagner family outperform the models of the Manne family. For this reason we construct new models for R-PFSPs based on the Wagner family. Our models with repetition differ from the original models in the description of a permutation, namely to describe a permutation in an R-PFSP we only need to give the types of the corresponding jobs in the order.

We will use the following notations.

Parameters:

- $M$  Number of machines
- $N$  Number of jobs
- $T$  Number of different types
- $n_t$  Number of jobs of type  $t$  ( $1 \leq t \leq T; \sum n_t = N$ )
- $P_{ri}$  Processing time of the job of type  $i$  on machine  $r$

Continuous variables:

- $C_{rj}$  Completion time of the  $j$ -th job of the sequence on machine  $r$
- $B_{r,j}$  Beginning time of the  $j$ -th job of the sequence on machine  $r$
- $X_{r,j}$  Idle time of machine  $r$  before the start of the  $j$ -th job of the sequence
- $Y_{r,j}$  Waiting time of the  $j$ -th job of the sequence in the buffer after it finishes processing on machine  $r$
- $C_{Max}$  Makespan.

Binary variable:

- $Z_{ij}$  Equals 1, if a job of type  $i$  is placed in the  $j$ -th place of the order, 0 otherwise.

### 3.1.1 The R-Wilson model

The first model, based on the work of Wilson (1989), called R-Wilson from now on, is presented below.

$$\sum_{j=1}^N Z_{ij} = n_i, \quad 1 \leq i \leq T \tag{1}$$

$$\sum_{i=1}^T Z_{ij} = 1, \quad 1 \leq j \leq N \tag{2}$$

$$B_{1j} + \sum_{i=1}^T P_{1i} Z_{ij} = B_{1,j+1} \quad 1 \leq j \leq N - 1 \tag{3}$$

$$B_{11} = 0 \tag{4}$$

$$B_{r1} + \sum_{i=1}^T P_{ri} Z_{i1} = B_{r+1,1} \quad 1 \leq r \leq M - 1 \tag{5}$$

$$B_{rj} + \sum_{i=1}^T P_{ri} Z_{ij} \leq B_{r+1,j} \quad 1 \leq r \leq M - 1; 2 \leq j \leq N \tag{6}$$

$$B_{rj} + \sum_{i=1}^T P_{ri} Z_{ij} \leq B_{r,j+1} \quad 2 \leq r \leq M; 2 \leq j \leq N - 1 \tag{7}$$

$$\min C_{max} = B_{MN} + \sum_{i=1}^T P_{Mi} Z_{iN} \tag{8}$$

Equation (1) ensures that there are  $n_i$  jobs in the sequence that are of type  $i$ . Constraint (2) states that each position in the sequence is filled with exactly one type of a job. Constraints (3), (4) and (5) state that there is no idle time on the first machine and the first job does not have to wait on any machine. Constraint (6) says that a job can start on machine  $r + 1$  only after its finished on machine  $r$ . Constraint (7) ensures that the job in position  $j + 1$  in the sequence can not start on machine  $r$  until the job in the  $j$ -th position in the sequence is finished on machine  $r$ .

### 3.1.2 The R-WST model

The second model, based on the formulation of Wagner (1959) with some modification suggested by Stafford and Tseng (2002), is called R-WST from now on.

$$\sum_{j=1}^N Z_{ij} = n_i, \quad 1 \leq i \leq T \tag{9}$$

$$\sum_{i=1}^T Z_{ij} = 1, \quad 1 \leq j \leq N \tag{10}$$

$$Y_{r1} = 0, \quad 1 \leq r \leq M - 1 \tag{11}$$

$$\sum_{i=1}^T P_{ri} Z_{i,j+1} + X_{r,j+1} + Y_{r,j+1} = \sum_{i=1}^T P_{r+1,i} Z_{i,j} + X_{r+1,j} + Y_{rj} \tag{12}$$

$1 \leq r \leq M - 1; 1 \leq j \leq N - 1$

$$X_{r,1} + Y_{r,1} + \sum_{i=1}^T P_{ri} Z_{i1} = X_{r+1,1} \quad 1 \leq r \leq M - 1 \tag{13}$$

$$\min C_{max} = \sum_{i=1}^T n_i \cdot P_{Mi} + \sum_{p=1}^N X_{Mp} \tag{14}$$

Again, constraint (9) ensures that there are  $n_i$  jobs in the sequence that are of type  $i$  and constraint (10) states that each position in the sequence is filled with exactly one type of a job. Constraint (11) states that the first job in the sequence does not have to wait on any machine. Constraints (12) and (13) say that the job in the  $(j + 1)$ th position of the sequence can not begin its processing on machine  $r$  until the job in the  $j$ -th position of the sequence has completed its processing on that same machine; and a job can not start processing on any machine until it has completed its processing on the previous machine.

### 3.1.3 The R-TS2 model

Our next model, based on the formulation of Stafford et al. (2005), called R-TS2 from now on, is presented below.

$$\sum_{k=1}^N Z_{ik} = n_i, \quad 1 \leq i \leq T \tag{15}$$

$$\sum_{l=1}^T Z_{lj} = 1, \quad 1 \leq j \leq N \tag{16}$$

$$C_{rj} + \sum_{i=1}^T P_{ri} Z_{i,j+1} \leq C_{r,j+1}, \quad 1 \leq r \leq M, 1 \leq j \leq N - 1 \tag{17}$$

$$C_{rj} + \sum_{i=1}^T P_{r+1,i} Z_{ij} \leq C_{r+1,j}, \quad 1 \leq r \leq M - 1, 1 \leq j \leq N \tag{18}$$

$$\sum_{i=1}^T P_{1i} Z_{i1} \leq C_{11} \tag{19}$$

$$\min C_{max} = C_{MN} \tag{20}$$

Equations (15) and (16) are the assignment problem as it was explained in the R-Wilson model. Constraint (17) says that the job in the  $(j + 1)$ -th position of the sequence can not finish on any machine until the job in the  $j$ -th position of the sequence is finished on that machine and job in the  $(j + 1)$ -th position of the sequence is processed on that machine. Constraint (18) ensures that a job can not finish on any machine until it is finished on the previous machine and processed on the given machine. Finally constraint (19) states that the first job can not finish earlier on the first machine than its processing time on the first machine.

### 3.2 MILP models for RL-PFSP

In this case the MILP models of the previous section can be simplified since we only have to give the sequence of the lots instead of the sequence of the jobs.

We will use the following notation:

- $L_i$  Number of lots that contain jobs of type  $i$  ( $1 \leq i \leq T$ )
- $L$  Total number of lots ( $L = L_1 + L_2 + \dots + L_T$ )
- $S$  Size of a lot ( $N = L \cdot S$ )
- $Z_{ij}$  Binary variable;  $Z_{ij} = 1$  iff the  $j$ -th lot contains jobs of type  $i$  ( $1 \leq i \leq T, 1 \leq j \leq L$ ).

#### 3.2.1 The RL-Wilson model

$$\sum_{j=1}^L Z_{ij} = L_i \quad 1 \leq i \leq T \tag{21}$$

$$\sum_{i=1}^T Z_{ij} = 1 \quad 1 \leq j \leq L \tag{22}$$

$$B_{1,S(j-1)+k} + \sum_{i=1}^T P_{1i} Z_{ij} = B_{1,S(j-1)+k+1} \quad 1 \leq j \leq L; 1 \leq k \leq S; jk < N \tag{23}$$

$$B_{11} = 0 \tag{24}$$

$$B_{r1} + \sum_{i=1}^T P_{ri} Z_{i1} = B_{r+1,1} \quad 1 \leq r \leq M - 1 \tag{25}$$

$$B_{r,S(j-1)+k} + \sum_{i=1}^T P_{ri} Z_{ij} \leq B_{r+1,S(j-1)+k} \quad 1 \leq r \leq M - 1; 1 \leq j \leq L; 1 \leq k \leq S; 1 < jk \tag{26}$$

$$B_{r,S(j-1)+k} + \sum_{i=1}^T P_{ri} Z_{ij} \leq B_{r,S(j-1)+k+1} \quad 2 \leq r \leq M; 1 \leq j \leq L; 1 \leq k \leq S, 2 \leq jk < N \tag{27}$$

$$\min C_{max} = B_{MN} + \sum_{i=1}^T P_{Mi} Z_{iL} \tag{28}$$

Constraint (21) ensures that there are  $L_i$  lots that contain jobs of type  $i$ . Constraint (22) states that each position in the sequence of the lots is filled with exactly one type of a lot. The remaining constraints state the same as in the R-Wilson model. We only have to be careful with the index numbers; the job in the  $(S(j - 1) + k)$ th position of the sequence (where  $1 \leq k \leq S$ ) is contained in a lot that is in the  $j$ -th position of the sequence of the lots.

### 3.2.2 The RL-WST model

$$\sum_{j=1}^L Z_{ij} = L_i \quad 1 \leq i \leq T \tag{29}$$

$$\sum_{i=1}^T Z_{ij} = 1 \quad 1 \leq j \leq L \tag{30}$$

$$Y_{r1} = 0 \quad 1 \leq r \leq M - 1 \tag{31}$$

$$\sum_{i=1}^T P_{ri} Z_{ij} = \sum_{i=1}^T P_{r+1,i} Z_{ij} - X_{r,S(j-1)+k+1} - Y_{r,S(j-1)+k+1} + X_{r+1,S(j-1)+k} + Y_{r,S(j-1)+k} \quad 1 \leq r \leq M - 1; 1 \leq j \leq L; 1 \leq k \leq S - 1 \tag{32}$$

$$\sum_{i=1}^T P_{ri} Z_{i,j+1} = \sum_{i=1}^T P_{r+1,i} Z_{i,j} - X_{r,jS+1} - Y_{r,jS+1} + X_{r+1,Sj} + Y_{r,Sj} \quad 1 \leq r \leq M - 1; 1 \leq j \leq L - 1 \tag{33}$$

$$X_{r+1,1} = X_{r,1} + Y_{r,1} + \sum_{i=1}^T P_{ri} Z_{i1} \quad 1 \leq r \leq M - 1 \quad (34)$$

$$\min C_{max} = \sum_{i=1}^T n_i \cdot P_{Mi} + \sum_{p=1}^N X_{Mp} \quad (35)$$

Equations (29) and (30) state that there are  $L_i$  lots that contain jobs of type  $i$  and each position in the sequence of the lots is filled with exactly one type of a lot. The other constraints are the modifications of the constraints of the R-WST model described in the previous section. The main difference is that equality (12) of the R-WST model has to be decomposed into two equalities (32) and (33) depending on whether the job in the  $(S(j - 1) + k)$ th position of the sequence is the last job of a lot (i.e.  $k = S$ ) or not.

### 3.2.3 The RL-TS2 model

$$\sum_{j=1}^L Z_{ij} = L_i, \quad 1 \leq i \leq T \quad (36)$$

$$\sum_{i=1}^T Z_{ij} = 1, \quad 1 \leq j \leq L \quad (37)$$

$$C_{r,S(j-1)+k} + \sum_{i=1}^T P_{ri} Z_{ij} \leq C_{r,S(j-1)+k+1} \quad 1 \leq r \leq M; 1 \leq j \leq L; 1 \leq k \leq S - 1 \quad (38)$$

$$C_{r,jS} + \sum_{i=1}^T P_{ri} Z_{i,j+1} \leq C_{r,jS+1}, \quad 1 \leq r \leq M; 1 \leq j < L \quad (39)$$

$$C_{r,S(j-1)+k} + \sum_{i=1}^T P_{r+1,i} Z_{ij} \leq C_{r+1,S(j-1)+k} \quad 1 \leq r \leq M - 1; 1 \leq j \leq L, 1 \leq k \leq S \quad (40)$$

$$\sum_{i=1}^T P_{1i} Z_{i1} \leq C_{11} \quad (41)$$

$$\min C_{max} = C_{MN} \quad (42)$$

Equations (36) and (37) are the classical assignment problem of the lots. The other constraints are the modifications of the constraints of the R-TS2 model. As it was explained in the RL-WST model inequality (17) is decomposed into two inequalities (39) and (40).



**Table 1** Size complexity of the models

Model	Binary variables	Real variables	Constraints
Wilson	$N^2$	$MN$	$2MN - M + N + 1$
WST	$N^2$	$2MN - N + 1$	$MN + M + N + 1$
TS2	$N^2$	$MN + 1$	$2MN - M + N + 1$
R-Wilson	$NT$	$MN$	$2MN - M + T + 1$
R-WST	$NT$	$2MN - N + 1$	$MN + M + T + 1$
R-TS2	$NT$	$MN + 1$	$2MN - M + T + 1$
RL-Wilson	$NT/S$	$MN$	$2MN - M - N + N/S + T + 1$
RL-WST	$NT/S$	$2MN - N + 1$	$MN + M - N + N/S + T + 1$
RL-TS2	$NT/S$	$MN + 1$	$2MN - M - N + N/S + T + 1$

$N$  number of jobs,  $M$  number of machines,  $T$  number of types,  $S$  lot size

### 3.3 Size complexity of the models

The size complexity of the original MILP models, their R-versions and their RL-versions are presented in Table 1. It can be seen that the main difference between the original models and their versions is the number of binary variables. The repeated models contain  $T/N$  times less binary variables than the original models while the models with lot size  $S$  contain  $T/NS$  times less binary variables than the original models.

## 4 Numerical experiments

To compare the presented MILP models an experimental design was constructed and executed. The performance measure is the computer solution time (CPU) required to solve the problems. The models were tested on different sized problems; we considered five randomly generated instances for each size. As suggested in Taillard (1993) the processing times were random integers drawn from the uniform distribution in the range (1, 100).

The formulations were written in GAMS modeling language and solved using CPLEX 12.3 on an Intel Xeon E31225 3.1 GHz personal computer equipped with 4 GB. The CPLEX options employed were mixed integer programming, an optimality stopping criterion of zero and a time limit of 3,600 s.

### 4.1 Experiment 1: Comparison of models for the R-PFSP

A 20 cell experimental design,  $M \in \{6, 7, 8, 9, 10\}$  by  $T \in \{6, 7, 8, 9\}$ , 5 jobs from each type, five replications per cell, was used to compare the repeated models. The R-Wilson model solved all 100 problems, the R-TS2 solved 99 problems while the R-WST model solved 98 problems within the time limit. Overall, the R-Wilson model required less computer time than did the R-WST model in 52 of the 100 problems,

**Table 2** Original models versus their repeated versions

<i>M</i>	<i>T</i>	Wilson	WST	TS2	R-Wilson	R-WST	R-TS2
6	6	9.87	16.95	6.95	2.52	3.64	3.41
	7	53.29	84.69	99.88	6.62	7.46	9.12
	8	179.90	*	76.08	5.52	13.55	6.38
	9	5.06	3.49	1.30	0.51	0.60	1.05
7	6	226.06	589.96	332.02	30.64	32.00	36.24
	7	25.00	301.21	3.27	3.15	35.79	17.70
	8	*	232.82	62.20	11.98	11.66	11.80
	9	19.69	42.54	42.83	4.75	3.34	4.98
8	6	1,101.60	1,126.17	*	16.32	46.85	22.75
	7	*	*	754.234	52.82	48.87	81.78
	8	*	*	*	77.56	249.94	54.21
	9	*	*	*	1,576.09	*	*
9	6	*	*	*	62.61	55.89	88.84
	7	134.40	734.16	121.12	9.80	35.10	47.62
	8	52.11	255.44	226.87	16.837	60.40	24.85
	9	401.70	1,251.24	629.57	34.30	61.40	41.89
10	6	*	*	*	75.43	38.09	70.50
	7	*	*	*	1,529.00	2,065.65	901.03
	8	*	*	*	1,080.83	2,205.17	447.04
	9	*	*	*	612.75	2,843.65	1,061.36

Solution times (in seconds) for solving problems by the MILP models

*T* number of types, *M* number of machines

Number of jobs  $N = 5T$

\*The problem was not solved to proven optimality in 1 h run time

and for the R-TS2 model it was 65. The R-WST model required less solution time than the R-TS2 model in 57 of the 100 problems.

Combining all cells a distribution-free two-way test based on the Friedman rank sums test (Hollander and Wolfe 1973) was used to test the null hypothesis that the ranks of the 3 models are equal. When the null hypothesis was rejected an all pair comparison procedure was used to determine significant differences between all pairs. The test showed that for Experiment 1 the R-Wilson model is significantly faster than both the R-WST model ( $p = 0.0001$ ) and the R-TS2 model ( $p = 0.0002$ ), and there is no significance difference between the R-WST and the R-TS2.

Then the hardest problem of each cell was solved with the original Wilson, WST and TS2 models, too. The results are shown in Table 2. It can be observed that (except one case) the repeated models solved the problems much faster than did the original ones. On the other hand the repeated models could solve larger problems (in terms of *M*) than the original models; repeated models could solve problems up to  $M = 10$  machines and  $N = 45$  jobs while the original models could solve problems only up to  $M = 7$  machines and  $N = 45$  jobs.

**Table 3** Solution times (in seconds) for problems in Experiment 2

T	Model	Mean			Median		
		$n_i = 2$	$n_i = 3$	$n_i = 4$	$n_i = 2$	$n_i = 3$	$n_i = 4$
7	R-Wil	3.74	9.83	159.90	4.37	3.78	5.89
	R-WST	4.63	15.82	529.01	4.86	5.32	25.98
	R-TS2	5.42	9.19	426.61	3.76	5.34	14.18
8	R-Wil	5.42	317.24	508.21	3.41	49.28	0.70
	R-WST	17.89	516.03	*	6.79	177.07	0.78
	R-TS2	5.60	247.22	558.43	3.76	150.81	1.05
9	R-Wil	273.48	*		26.05	*	
	R-WST	421.71	*		51.22	*	
	R-TS2	233.19	*		26.00	*	
10	R-Wil	100.02			47.86		
	R-WST	330.40			192.56		
	R-TS2	133.96			109.44		
11	R-Wil	*			*		
	R-WST	*			*		
	R-TS2	*			*		

Number of machines  $M = 10$ ; five problems per cell  
 T number of types,  $n_i$  number of jobs of type  $i$  ( $n_1 = n_2 = \dots = n_T$ )  
 Number of jobs  $N = T \cdot n_i$

\* At least one problem was not solved to proven optimality in 1 h run time

### 4.2 Experiment 2: Increasing the number of types or the number of jobs of the same type

The goal of Experiment 2 was to investigate the problem sizes solvable by the repeated models in terms of increasing  $N$ . The number of jobs was increased in two different ways: on one hand for a fixed type number  $T$  the number of jobs of the same type  $n_i$  was increased and on the other hand for a fixed  $n_i$  the number of types  $T$  was increased until CPLEX could solve all problems in a cell. The number of machines was 10 in all instances. The results are summarized in Table 3.

For  $T = 7$  each model solved all problems up to  $n_i = 4$  (i.e. 28 jobs overall). For  $T = 8$  the R-Wilson and R-TS2 models solved all problems up to  $n_i = 4$  (i.e. 32 jobs overall) while the R-WST model failed to solve one instance optimally in this cell. For  $T = 9, 10$  each model could solve problems with two jobs from each type, but they all failed to solve all problems with  $n_i = 3$ . For  $T = 11$  none of the models were able to solve all problems with  $n_i = 2$ .

On average the R-WST model required the most computer time in all eight cells and the R-WST model had the most median solution time in six of the eight cells. On average the R-Wilson model required less computer time than did the R-TS2 model in five cells and the R-Wilson model had less median solution time than did the R-TS2 model in five cells. These results suggests that for  $M = 10$  the R-Wilson model is the fastest followed by the R-TS2 and the R-WST.

**Table 4** Solution times (in seconds) for problems in Experiment 3

<i>T</i>	Model	<i>M</i>							
		Mean				Median			
		9	10	11	12	9	10	11	12
6	R-Wilson	2.51	5.35	14.33	*	0.96	7.16	6.77	25.24
	R-WST	2.71	7.19	25.62	*	0.89	8.73	14.13	53.32
	R-TS2	4.23	5.60	18.78	*	0.88	8.01	7.69	26.14
8	R-Wilson	139.16	314.65	307.52		6.89	15.16	136.58	
	R-WST	289.34	737.53	*		6.07	22.40	193.97	
	R-TS2	168.59	373.70	286.18		5.22	13.49	224.06	

Number of jobs  $N = 24$ , five problems per cell

$T$  number of types,  $M$  number of machines

Each type has the same number of jobs

\* At least one problem was not solved to proven optimality in 1 h run time

Using the Friedman rank sums test as employed in experiment 1 the results show that when all cells are combined then both the R-Wilson and the R-TS2 models are significantly faster than the R-WST model ( $p < 0.0001$ ), and there is no significant difference between the R-Wilson and the R-TS2 models.

It is also interesting to compare the results in cell  $T = 9$ ;  $n_i = 2$  with the results in cell  $T = 7$ ;  $n_i = 3$ . Each model required far less average and median computer time in cell  $T = 7$ ;  $n_i = 3$  than in cell  $T = 9$ ;  $n_i = 2$ ; although cell  $T = 9$ ;  $n_i = 2$  has less jobs (18) than the other cell (21). One reason of this anomaly is that the number of binary variables has a great impact on computer solution time and although cell  $T = 9$ ;  $n_i = 2$  has less jobs than the other cell but it has more binary variables (171) than the other cell (147). This also suggests that larger (in terms of  $N$ ) problems with small values of  $T$  (i.e. with few types) can be solved by the repeated models.

#### 4.3 Experiment 3: Increasing the number of machines

The goal of Experiment 3 was to investigate the problem sizes solvable by the repeated models in terms of increasing  $M$ . In each cell the number of jobs is 24; the number of the different types  $T$  is 6 and 8. The results are summarized in Table 4.

In the case of  $T = 6$  each model could solve all the problems up to  $M = 11$  but they failed to solve all the problems in 1 h for  $M = 12$ . In the case of  $T = 8$  the R-Wilson and the R-TS2 model solved all problems up to  $M = 11$  while the R-WST model failed to solve all the problems for  $M = 11$ .

On average the R-Wilson model required less computer time in five of the six cells than the R-WST model and less than the R-TS2 model in five cells. The R-TS2 model required less computer time in all cells than the R-WST model.

Using the Friedman-rank sums test as in the previous experiments the results show that the R-Wilson model is faster than both the R-WST ( $p < 0.0001$ ) and the R-TS2 ( $p = 0.01$ ) models while the R-TS2 model is faster than the R-WST model ( $p = 0.0166$ ).

**Table 5** Experiment 4: Models with lot size

<i>S</i>	Model	$n_i = 4$	$n_i = 8$	$n_i = 12$	$n_i = 16$	$n_i = 24$
Part A. Mean solution time						
2	RL-Wilson	19.61	82.27	50.28	*	
	RL-WST	32.09	25.31	532.21	*	
	RL-TS2	22.31	32.10	107.12	*	
4	RL-Wilson	2.77	43.28	107.81	*	
	RL-WST	2.58	116.49	292.78	*	
	RL-TS2	2.49	47.06	137.35	*	
8	RL-Wilson		3.54		256.26	262.53
	RL-WST		4.16		*	*
	RL-TS2		4.45		7.41	585.49
Part B. Median solution time						
2	RL-Wilson	11.22	4.41	1.48	8.77	
	RL-WST	14.42	4.63	0.91	4.85	
	RL-TS2	15.23	4.14	1.64	24.20	
4	RL-Wilson	3.00	7.40	1.41	3.25	
	RL-WST	1.54	4.28	1.76	5.45	
	RL-TS2	1.91	3.85	4.67	3.84	
8	RL-Wilson		4.34		11.18	9.11
	RL-WST		2.88		4.45	3.59
	RL-TS2		4.16		4.20	28.37
Part C. Means of the relative difference of $C_{max}^S$ from $C_{max}^1$ (in %)						
2		1.18	0.05	0.1	*	
4		7.23	1.14	0.3	*	
8			4.34		1.67	0.44

Number of machines  $M = 10$ , number of types  $T = 8$

$S$  lot size,  $n_i$  number of jobs of type  $i$ , ( $n_1 = n_2 = \dots = n_T$ )

Number of jobs  $N = T \cdot n_i$

\* At least one problem was not solved to proven optimality in 1 h run time

It can be seen from the results that for a fixed  $M$  all three models solved the problems with  $T = 6$  much faster than the problems with  $T = 8$ . In fact this can be expected from the size complexity of the models (see Table 1) since for fixed  $M$  and  $N$  the models of problems with less types have less binary variables.

#### 4.4 Experiment 4: Comparison of the models with lot size

The models with lot size are compared in several instances. In this experiment we considered instances with  $M = 10$ ,  $T = 8$ ,  $n_i \in \{4, 8, 12, 16, 24\}$  and  $S \in \{2, 4, 8\}$ . The results are shown in Table 5.

First observe that each model with lot size 2 and 4 solved all the problems up to  $n_i = 12$  (i.e. 96 jobs overall). The RL-Wilson and the RL-TS2 models with lot size 8 solved all the problems up to  $n_i = 24$  (i.e. 192 jobs overall) while the RL-WST model

with lot size 8 could not solve one problem for  $n_i = 16, 24$ . Experiment 2 showed that for  $M = 10$  and  $T = 8$  the repeated models without lot size (i.e. with  $S = 1$ ) solved the problems only up to  $n_i = 4$ . This means that using models with lot size greater than one larger problems become solvable.

Combining all the cells the Friedman ranked-sums test shows that there is no significant difference between the models ( $p = 0.133$ ). However, if we confine ourselves to the 8 hardest problems (in which each model required more than 50 seconds of solution time) the situation is quite different. The RL-Wilson model was the fastest in 7 of the 8 problems while the RL-WST model was the slowest in 6 of the 8 problems.

We also compared the optimal values of the problems solved by using models with lot size and models without lot size (i.e.  $S > 1$  and  $S = 1$ ). For each cell Part C of Table 5 contains the average values of

$$\left( \frac{C_{max}^S - C_{max}^1}{C_{max}^1} - 1 \right) \cdot 100$$

which is the relative difference of  $C_{max}^S$  and  $C_{max}^1$  where  $C_{max}^S$  and  $C_{max}^1$  denote the optimal value of the problem with lot size  $S$  and lot size 1 respectively. The results suggest that fixing  $S$  and increasing  $n_i$  the optimal value of  $C_{max}^S$  is getting closer and closer to  $C_{max}^1$ . For example in the cell  $S = 8$ ,  $n_i = 24$ , on average the  $C_{max}^8$  values were within 0.44 % of the  $C_{max}^1$  values. In fact  $C_{max}^8$  was equal to  $C_{max}^1$  in 3 of 5 instances of this cell. This shows that for larger problems containing many jobs of the same type one can compute a good approximation of the problem by solving it with a model with lot size greater than one.

## 5 Case study

Our three repeated MILP models were also tested on a PFSP problem arising from real industrial production line scheduling in automotive industry (Jósvai 2009). The engine production line consisted of three segments including the final control segment. There were some automated processing machines distributed among the other, manual processing machines. In total  $N = 227$  jobs with  $T = 12$  different types had to be processed on  $M = 57$  machines. The number of the jobs of one type,  $n_i$  varied between 6 and 39. Processing time values distributed in a wide range, typically much larger in the last controlling segment. The makespan condition was a proper objective since the final goal was to make a plan of the processing for a prescribed number of jobs that fit into the total allowed time for production.

All three repeated models were able to solve the problem. The R-Wilson, the R-TS2 and the R-WST required 219.6, 278.5 and 1,517.4 s, respectively, to find the optimal solution, which is a remarkable result, namely that large scale industrial problems could be solved by using the new MILP models introduced in this paper. This optimum resulted 20 min shorter production time for the jobs than the best plan made manually.

We remark that industrial-like test problems were generated randomly with parameters with similar ranges. The repeated MILP methods did not solve all these problems within a reasonable time, so it would be useful to find conditions that cause fast convergence of exact methods on large scale data.

## 6 Conclusions

In this paper we introduced two new families of problems derived from the classical PFSP, namely the R-PFSP and the RL-PFSP. The R and RL problem sets arise from industrial production line modeling problems where many jobs are repeated. This means that several jobs have the same processing time on all machines, which reflects the typical situation that during one planning period in a factory we have to produce products of some types and of each type we have to produce several items. To be able to solve problems as large as possible with MILP models, we defined three new MILP models for each repeated problem.

To benchmark the new methods, four experiments were conducted to compare the MILP models. The major conclusions from the numerical experiments reported in this paper are as follows.

1. Permutation flow shop problems containing repeated jobs can be solved significantly faster with the proposed repeated MILP models than with the previously known MILP models. Thus the repeated models could solve larger problems than the original models ( $M = 10, N = 45$  vs.  $M = 7, N = 45$ , or  $M = 8, N = 45$  vs.  $M = 8, N = 30$ ).
2. The R-Wilson model performed significantly better on the Friedman test than the R-WST and the R-TS2 models on problem sizes  $M = 6, 7, 8, 9, 10$  and  $T = 6, 7, 8, 9, n_i = 5$ , i.e.  $N = 30, 35, 40, 45$ .
3. The R-Wilson and the R-TS2 models performed better than the R-WST model on problem sizes  $M = 10, T = 7, 8, 9, 10, n_i = 2, 3, 4$ .
4. MILP models with lot size can solve much larger problems in reasonable time than models without lot size. For example, the RL-Wilson model with lot size  $S = 8$  can solve problems up to  $M = 10$  machines and  $N = 192$  jobs in 1 h. Moreover, in 3 of the 5 instances  $S = 8$  gave the same optimum as  $S = 1$ . That is there was no loss of accuracy when turning from  $S = 1$  to  $S = 8$ . (The optimum for  $S = 1$  was computed on 4 cores in larger computational time).

## References

- Franca PM, Gupta JND, Mendes PM, Veltink KJ (2005) Evolutionary algorithms for scheduling a flowshop manufacturing cell with sequence dependent family setups. *Comput Ind Eng* 20:1–16
- Hollander M, Wolfe DA (1973) *Nonparametric statistical methods*. Wiley, New York
- Jósvai J (2009) Production process modeling and planning with simulation method, mounting process optimisation. In: *The international conference on modeling and applied simulation*, 23–25 Sept 2009, Spain. Universidad de La Laguna, Spain, pp 240–245
- Pan CH (1997) A study of integer programming formulations for scheduling problems. *Int J Syst Sci* 28:33–41
- Pinedo M (2008) *Scheduling theory, algorithms, and systems*. Prentice-Hall, Englewood Cliffs
- Salmasi N, Logendran R, Skandari MR (2010) Total flow time minimization in flowshop sequence dependent group scheduling problem. *Comput Oper Res* 37:199–212
- Schaller JE, Gupta JND, Vakharia AJ (2000) Scheduling a flowline manufacturing cell with sequence dependent family setup times. *Eur J Oper Res* 125:324–329
- Stafford EF (1988) On the development of a mixed-integer linear programming linear programming model for the flowshop sequencing problem. *J Oper Res Soc* 39:1163–1174
- Stafford EF, Tseng FT (1990) On the Strikar-Gosh MILP model for the  $N \times M$  SDST flowshop problem. *Int J Prod Res* 28:1817–1830

- Stafford EF, Tseng FT (2002) Two models for a family of flowshop sequencing problems. *Eur J Oper Res* 142:282–293
- Stafford EF, Tseng FT, Gupta ND (2004) An empirical analysis of integer programming formulations for the permutation flowshop. *Omega* 32:285–293
- Stafford EF, Tseng FT, Gupta ND (2005) Comparative evaluation of the MILP flowshop models. *J Oper Res Soc* 56:88–101
- Taillard E (1993) Benchmarks for basic scheduling problems. *Eur J Oper Res* 64:278–285
- Wagner HM (1959) An integer linear-programming model for machine scheduling. *Nav Res Logist Q* 6:131–140
- Wilson JM (1989) Alternative formulations of a flow-shop scheduling problem. *J Oper Res Soc* 40:395–399