

MILP models for the optimization of real production lines

Tamás Hajba · Zoltán Horváth

Published online: 25 January 2014
© Springer-Verlag Berlin Heidelberg 2014

Abstract A special class of the permutation flow shop problems (PFSPs) with makespan minimization which contains repeated jobs, limited buffer sizes between the machines and fixed number of palettes that carry the jobs on the production line is introduced in this paper. We define the related PB-R-PFSP, the Permutation with Repetition Flow Shop Problem with Palettes and Buffers, and construct new mixed integer linear programming (MILP) models for this problem. The effectiveness of the MILP models and the influence of the number of palettes and buffers on the problems are investigated experimentally.

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

Mathematics Subject Classification (2000) 90C11 · 90C90

1 Introduction

Scheduling of jobs on production lines at factories is an important task for the management, often the most crucial planning stage. This is supported by some optimization tools at modern factories to find the best production plan. The time available for this planning is often restricted to a given, rather short period and the problem is large-scale (see [Jósvai 2009](#)), which makes the optimization hard.

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

Z. Horváth
e-mail: horvathz@sze.hu

In the present paper, as a continuation of our previous work (see [Hajba and Horváth 2013](#)), our aim is to develop and investigate exact mathematical models of the production lines at factories taking into account as many real process components as possible. Since the problems arising in this context are NP-hard, typically some heuristics are applied. Our aim is to make define exact models with exploiting the inherent structure of the industrial problem to be able to get the exact optimum for as big problems as possible or, if this does not converge, serve approximate results of the more complex model of the industrial problem. Our findings in the paper *loc. cit.* was that, surprisingly, the optimization process based on some of the exact models could find the guaranteed optimum even on large-scale problems when the industrial restrictions were handled properly.

In the literature, only few papers can be found in the topics of exact models for large scale production line scheduling. The classical mathematical problem is the permutation flow shop problem (PFSP) with makespan criterion for which several mixed integer linear programming (MILP) models were developed; namely as empirical analysis showed (see [Stafford et al. 2004](#); [Stafford et al. 2005](#); [Stafford and Tseng 2008](#)) the best models are the models of the Wagner family: the Wilson model (see [Wilson 1989](#)), the TS2 model (see [Stafford et al. 2005](#)), the WST model (see [Wagner 1959](#); [Stafford and Tseng 2002](#)), the TBA and TS3 models (see [Stafford and Tseng 2008](#)). Beside the MILP models other exact methods, such as the branch and bound algorithm were used to solve the PFSP (see [Ladhari et al. 2005](#); [Kouki et al. 2010, 2011](#)). In all of these models the production line is idealized with having, among other assumptions, an infinitely large buffer between adjacent machines and arbitrary many number of jobs can be on the line under process at the same time. We remark that exact, mixed integer programming models were successfully applied to solve similar large scale problems for integrated planning and scheduling in the industry (see [Kis and Kovács 2012](#)).

In this paper we develop MILP models for production lines with more special features arising from industry. Namely, we investigate

- repetitions: the jobs can often be sorted into types so that jobs of the same type have equal processing time values at each machine,
- palettes: the jobs are carried on palettes on the production line and the number of palettes is bounded from above,
- buffers: there are limited buffer sizes between the machines.

The goal of the paper is to define the related PB-R-PFSP, the Permutation with Repetition Flow Shop Problem with Palettes and Buffers, and construct the adequate new MILP models for PB-R-PFSPs and investigate their effectiveness experimentally.

Although the features under PB-R-PFSP are widely used in the industry, PB-R-PFSP models have not been investigated in the literature. Up to our knowledge, it is known only that MILP models for the PFSP with repeated jobs were introduced and studied in [Hajba and Horváth \(2013\)](#). Moreover, PFSP with zero buffer (special case of the limited buffer size) was studied in [Ronconi and Birgin \(2012\)](#) with evaluating classical MILP models for the PFSP to minimize the total tardiness and earliness while in [Fraschet et al \(2011\)](#) MILP models for the non-permutation FSP with limited buffers were introduced. The condition fixed number of palettes has not been studied yet.

The rest of the paper is organized as follows. We introduce the PB-R-FPSP 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 the problem. 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 the numerical experiments was on one hand to compare the three MILP models and on the other hand to investigate the effect of the palettes and buffers on the optimum value of the problem and on the computational time the MILP models need to solve the problem. Sect. 5 contains our conclusions.

2 Definition of the classical PFSP and the novel PB-R-PFSP

The regular permutation flow shop problem consists of a production system of M machines and a set of N jobs, each of which has to be processed on every machine. Each job has to be processed first on the first machine, next on the second machine and so on. The schedule of the jobs must be the same on each machine. A machine can process at most one job at any time and preemption is not allowed. In the classical problem it is assumed that there is unlimited buffer between the machines which means that a job can immediately leave a machine after its processing is completed on that 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.

Real-life industrial problems often have some special features which are not included in the regular PFSP. One such property is that in practice the jobs can be sorted into types so that jobs of the same type have equal processing time values at each machine (see Hajba and Horváth 2013). If we take this property into consideration then the number of different permutations, i.e. the design space for the PFSP decreases from $N!$ to $\approx T^N$, where T denotes the number of different types. In industrial problems jobs are often carried on palettes on the line which means that each job entering the production line is placed on a palette and the job is carried along the line on this palette. After a job is processed on the last machine the job is taken off the palette and the palette is moved to the first machine where a new job can be placed on it once again. Typically the number of palettes is less than the number of jobs, implying that at the same time only a limited number of jobs can be on the line. As a consequence, because of the size of the palettes and the space between consecutive machines, only a limited number of palettes (hence limited number of jobs) can wait between two consecutive machines which means that there are limited buffer sizes between the machines. We will call a PFSP which contains repeated jobs, limited buffer sizes between the machines and bounded number of palettes *Permutation with Repetition Flow Shop Problem with Palettes and Buffer* (PB-R-PFSP).

3 MILP models for the PB-R-PFSP

3.1 Common notations of the MILP models for the PB-R-PFSP

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
- K Number of palettes
- b_r Buffer size between machine r and $r + 1$ ($1 \leq r \leq M - 1$).

Continuous variables:

- C_{rj} Completion time of the j -th job of the sequence on machine r
- B_{rj} Beginning time of the j -th job of the sequence on machine r
- X_{rj} Idle time of machine r before the start of the j -th job of the sequence (i.e. $X_{rj} = B_{rj} - C_{r,j-1}$)
- Y_{rj} waiting time of the j -th job of the sequence after it finishes processing on machine r (i.e. $Y_{rj} = B_{r+1,j} - C_{rj}$)
- C_{max} Makespan.

Binary variable:

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

The term D_{rj} is used in the models to represent the processing time of the j -th job of the sequence on machine r . It can be easily calculated that

$$D_{rj} = \sum_{i=1}^T P_{ri} Z_{ij}. \tag{1}$$

The term D_{rj} is used to simplify the notation in the constraints of the 3 new MILP models below.

3.2 Computing the makespan of a given permutation in a PB-R-PFSP

In a PFSP a job can start on the first machine immediately after the previous job is finished on the first machine which means that there is no idle time on the first machine. In a PB-R-PFSP it is not always true. If we denote by K the number of palettes then the j -th job of the order ($j > K$) can not start on the first machine unless there is an empty palette which means that j -th job of the order can not start on the first machine unless the $(j - K)$ -th job of the order is finished on the last machine. In a PFSP a job can start immediately on machine r after it is finished on machine $r - 1$ and the previous job is finished on machine r . If there are limited buffer sizes between the machines this is not always true. Let us denote the buffer size between machine r and $r + 1$ by b_r . Suppose that the j -th job of the order ($j > b_r$) is finished on machine $r - 1$ and the $(j - 1)$ -th job of the order is finished on machine r . In this case if the buffer between machines r and $r + 1$ is full then the $(j - 1)$ -th job of the order can

not leave machine r hence we can not start to process the j -th job of the order on machine r . It can be easily calculated that this means that the j -th job of the order can start on machine r only after the $(j - b_r - 1)$ -th job of the order is started on machine $r + 1$. Summarizing all these we get that for a given permutation the starting times of the jobs and hence the makespan can be calculated recursively the following way:

$$B_{11} = 0 \tag{2}$$

$$B_{r1} = B_{r-1,1} + D_{r-1,1} \quad (2 \leq r \leq M) \tag{3}$$

$$B_{1j} = \max(B_{1,j-1} + D_{1,j-1}, B_{2,j-b_1-1}, B_{M,j-K} + D_{M,j-K}) \tag{4}$$

(1 < j ≤ N)

$$B_{rj} = \max(B_{r,j-1} + D_{r,j-1}, B_{r+1,j-b_r-1}, B_{r-1,j} + D_{r-1,j}) \tag{5}$$

(2 ≤ r ≤ M; 2 ≤ j ≤ N)

$$C_{max} = B_{MN} + D_{MN} \tag{6}$$

In the above formulations both B_{rj} and D_{rj} are defined 0 if either $M < r$ or $j < 1$. Remark. As it can be seen the number of the palettes influence the beginning times of the jobs on the first machine.

3.3 Construction of MILP models for the PB-R-PFSP

Based on earlier works of [Wilson \(1989\)](#), [Stafford and Tseng \(2002\)](#) and [Stafford et al. \(2005\)](#), [Hajba and Horváth \(2013\)](#) introduced MILP models for the R-PFSPs. The new MILP formulations of the PB-R-PFSPs are the modifications of these models.

3.3.1 The PB-R-Wilson model

$$\sum_{j=1}^N Z_{ij} = n_i \quad 1 \leq i \leq T \tag{7}$$

$$\sum_{i=1}^T Z_{ij} = 1 \quad 1 \leq j \leq N \tag{8}$$

$$B_{11} = 0 \tag{9}$$

$$B_{r1} + D_{r1} = B_{r+1,1} \quad 1 \leq r \leq M - 1 \tag{10}$$

$$B_{rj} + D_{rj} \leq B_{r+1,j} \quad 1 \leq r \leq M - 1; 2 \leq j \leq N \tag{11}$$

$$B_{rj} + D_{rj} \leq B_{r,j+1} \quad 1 \leq r \leq M; 1 \leq j \leq N - 1 \tag{12}$$

$$B_{M,j-K} + D_{M,j-K} \leq B_{1j} \quad K + 1 \leq j \leq N \tag{13}$$

$$B_{r+1,j-b_r-1} \leq B_{rj} \quad 1 \leq r \leq M - 1, 2 + b_r \leq j \leq N \tag{14}$$

$$\min C_{max} = B_{MN} + D_{MN} \tag{15}$$

Equation (7) ensures that there are n_i jobs in the sequence that are of type i . Constraint (8) states that each position in the sequence is filled with exactly one type of a job. Constraints (9), (10) state that the first job does not have to wait on any

machine. Constraint (11) says that a job can not start on machine $r + 1$ until its finished on machine r . Constraint (12) states that the $(j + 1)$ -th job 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 . Constraint (13) ensures that at most K palettes are used in the system. Constraint (14) ensures that at most b_r jobs can wait in the buffer between machines r and $r + 1$.

3.3.2 The PB-R-WST model

$$\sum_{j=1}^N Z_{ij} = n_i, \quad 1 \leq i \leq T \tag{16}$$

$$\sum_{i=1}^T Z_{ij} = 1, \quad 1 \leq j \leq N \tag{17}$$

$$Y_{r1} = 0, \quad 1 \leq r \leq M - 1 \tag{18}$$

$$D_{r,j+1} + X_{r,j+1} + Y_{r,j+1} = D_{r+1,j} + X_{r+1,j} + Y_{rj} \tag{19}$$

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

$$X_{r,1} + Y_{r,1} + D_{r1} = X_{r+1,1} \quad 1 \leq r \leq M - 1 \tag{20}$$

$$\sum_{i=1}^{j-K} X_{Mi} + \sum_{i=1}^{j-K} D_{Mi} \leq \sum_{i=1}^j X_{li} + \sum_{i=1}^{j-1} D_{li} \tag{21}$$

$K + 1 \leq j \leq N$

$$\sum_{i=1}^{j-b_r-1} X_{r+1,i} + \sum_{i=1}^{j-b_r-2} D_{r+1,i} \leq \sum_{i=1}^j X_{ri} + \sum_{i=1}^{j-1} D_{ri} \tag{22}$$

$1 \leq r \leq M - 1, 2 + b_r \leq j \leq N$

$$\min C_{max} = \sum_{i=1}^T n_i \cdot P_{Mi} + \sum_{p=1}^N X_{Mp} \tag{23}$$

Again, constraint (16) ensures that there are n_i jobs in the sequence that are of type i and constraint (17) states that each position in the sequence is filled with exactly one type of a job. Constraint (18) states that the first job in the sequence does not have to wait on any machine. Constraints (19) and (20) 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. Constraint (21) implies that at most K palettes are used in the system. Constraint (22) ensures that at most b_r jobs can wait in the buffer between machines r and $r + 1$.

3.3.3 The PB-R-TS2 model

$$\sum_{k=1}^N Z_{ik} = n_i \quad 1 \leq i \leq T \tag{24}$$

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	$MN + 1$	$2MN - M + N + 1$
PB-R-Wilson	NT	MN	$3MN - 2M + T - K - B + 2$
PB-R-WST	NT	$2MN - N + 1$	$2MN + T - K - B + 2$
PB-R-TS2	NT	$MN + 1$	$3MN - 2M + T - K - B + 2$

N number of jobs, M number of machines, T number of types, K number of palettes, $B = \sum_{r=1}^{M-1} b_r$

$$\sum_{l=1}^T Z_{lj} = 1 \quad 1 \leq j \leq N \tag{25}$$

$$C_{rj} + D_{r,j+1} \leq C_{r,j+1} \quad 1 \leq r \leq M, 1 \leq j \leq N - 1 \tag{26}$$

$$C_{rj} + D_{r+1,j} \leq C_{r+1,j} \quad 1 \leq r \leq M - 1, 1 \leq j \leq N \tag{27}$$

$$D_{11} \leq C_{11} \tag{28}$$

$$C_{M,j-K} + D_{1j} \leq C_{1j} \quad K + 1 \leq j \leq N \tag{29}$$

$$C_{r+1,j-b_r-1} - D_{r+1,j-b_r-1} + D_{r,j} \leq C_{rj} \quad 1 \leq r \leq M - 1, 2 + b_r \leq j \leq N \tag{30}$$

$$\min C_{max} = C_{MN} \tag{31}$$

Equations (24) and (25) are the assignment problem as it was explained in the PB-R-Wilson model. Constraint (26) 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 (27) ensures that a job can not finish on any machine until it is finished on the previous machine and processed on the given machine. Constraint (28) states that the first job can not finish earlier on the first machine than its processing time on the first machine. Constraint (29) ensures that at most K palettes are used in the system while constraint (30) ensures that at most b_r jobs can wait in the buffer between machines r and $r + 1$.

3.4 Size complexity of the models

The size complexity of the original MILP models and their PB-R-versions are presented in Table 1. It can be seen that the main difference between the original models and their PB-R versions is the number of binary variables, namely the new models contain T/N times less binary variables than the original models. It can be seen that the number of binary and continuous variables of the PB-R models are independent

of the number of palettes and buffer sizes. The number of palettes and the buffer sizes effect only the number of constraints, namely decreasing by one the number of palettes or the buffer size between two machines yields one new inequality in all three PB-R models.

4 Numerical experiments

The main goals of the experiments were to compare the effectiveness of the 3 MILP models and to investigate the influence of the number of palettes and buffers between the machines on the optimal value and on the computational time the PB-R models need to solve the problems. To do so a 4 cell experimental design, $M \in \{7, 8\}$ by $T \in \{6, 7\}$, 5 jobs from each type, five replications per cell, was used. As suggested in Taillard (1993) the processing times were random integers drawn from the uniform distribution in the range (1, 100). All instances were solved with $K \in \{8, 9, 10\}$ by $b_r \in \{0, 1, 2\}$, so each of the 3 PB-R models were tested on overall 180 problems. The median solution times are shown in Tables 2 and 3 .

The formulations of the MILP models 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, parallel mode with 4 threads, an optimality stopping criterion of zero and a time limit of 3600 seconds.

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

To compare the 3 models first we examined the number of problems (from the total 180) the models solved in the time limit and the number of problems in which they

Table 2 Median solution times (in seconds) for problems with 7 machines in Experiment 1

<i>b</i>	Model	<i>T</i> = 6			<i>T</i> = 7		
		<i>K</i> = 8	<i>K</i> = 9	<i>K</i> = 10	<i>K</i> = 8	<i>K</i> = 9	<i>K</i> = 10
2	PB-R-Wil	1.064	0.446	0.437	1.898	1.655	1.238
	PB-R-TS2	0.780	0.481	0.475	3.587	1.736	0.841
	PB-R-WST	2.689	2.541	1.658	4.726	5.317	4.124
1	PB-R-Wil	1.042	0.698	1.045	2.863	1.880	1.782
	PB-R-TS2	0.609	0.578	0.415	6.118	1.692	1.789
	R-PB-WST	2.388	2.042	1.537	15.866	5.362	5.452
0	PB-R-Wil	965.973	1,933.102	1,233.905	a	a	a
	PB-R-TS2	700.226	1,323.840	668.558	a	a	a
	PB-R-WST	1,409.095	1,161.973	1,027.251	a	a	a

Number of machines $M = 7$; five problems per cell; number of types; five jobs from each type; number of jobs $N = 5T$; K = number of palettes; b = buffers between the machines ($b_1 = b_2 = \dots = b_{M-1}$)

^a At least three problems were not solved to proven optimality in 1 hour run time

Table 3 Median solution times (in seconds) for problems with 8 machines in Experiment 1

b	Model	$T = 6$			$T = 7$		
		$K = 8$	$K = 9$	$K = 10$	$K = 8$	$K = 9$	$K = 10$
2	PB-R-Wil	2.391	2.636	2.599	a	3.073	3.660
	PB-R-TS2	1.930	2.414	2.599	a	4.542	0.800
	PB-R-WST	4.772	3.729	4.886	a	2.409	1.294
1	PB-R-Wil	2.444	2.482	2.853	a	8.859	6.199
	PB-R-TS2	2.096	1.708	1.820	a	4.710	2.819
	R-PB-WST	5.105	4.382	4.272	a	6.382	1.835
0	PB-R-Wil	a	a	a	a	a	a
	PB-R-TS2	a	a	a	a	a	a
	PB-R-WST	a	a	a	a	a	a

number of machines $M = 8$; five problems per cell; $T =$ number of types; five jobs from each type; number of jobs $N = 5T$; $K =$ number of palettes; $b =$ buffers between the machines ($b_1 = b_2 = \dots = b_{M-1}$)

^a At least three problems were not solved to proven optimality in 1 hour run time

Table 4 Comparison of the models

Model	Number of problems solved	Number of problems with lowest solution time	Number of problems with highest solution time
PB-R-Wil	113	31	27
PB-R-TS2	117	75	11
PB-R-WST	115	12	87

required the lowest/highest computation time to solve the problem. The results are shown in Table 4.

4.1.1 Comparing the models on the easy problems

For the easy problems in which all of the models could solve the problem in the time limit, the solution times were used to compare the models. For all of the pairs (K, b_r) the distribution-free 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 then for all pairs of the models the Wilcoxon signed rank test was used to determine significant differences between all pairs.

The test showed that there is no significant difference between the models for problems with $b_r \equiv 0$ and $K = 8, 9, 10$.

For problems with $b_r \equiv 1$ and $K = 8, 9, 10$ the PB-R-TS2 model is significantly faster than the PB-R-WST model ($p = 0.013$; $p = 0.0003$; $p = 0.008$ respectively). For problems with $b_r \equiv 1$ and $K = 8, 9$ the PB-R-TS2 model is significantly faster than the PB-R-Wilson model ($p = 0.0217$; $p = 0.0015$) while for problems with $b_r \equiv 1$ and $K = 10$ there is no significant difference between the PB-R-TS2 and PB-R-Wilson models. The PB-R Wilson model is significantly faster than the PB-R-WST

model for problems with $b_r \equiv 1$ and $K = 8, 10$ ($p = 0.0057$; $p = 0.0436$) while for problems with $b_r \equiv 1$ and $K = 9$ there is no significant difference between the PB-R-Wilson and the PB-R-WST models.

The PB-R-TS2 model is significantly faster than the PB-R-WST model ($p = 0.0113$; $p = 0.0436$; $p = 0.0008$) for problems with $b_r \equiv 2$ and $K = 8, 9, 10$. The PB-R-TS2 model is significantly faster ($p = 0.0034$) than the PB-R-Wilson model for problems with $b_r \equiv 2$ and $K = 10$ but there is no difference between the two models for problems with $b_r \equiv 2$ and $K = 8, 9$. The PB-R-Wilson model is faster than the PB-R-WST model ($p = 0.0329$; 0.0268) for problems with $b_r \equiv 2$ and $K = 8, 10$ but there is no difference between the two models for problems with $b_r \equiv 2$ and $K = 9$.

The results suggest that the PB-R-TS2 model is the best to solve easy PB-R-PFSPs, followed by the PB-R-Wilson and the PB-R-WST model.

4.1.2 Comparing the models on the hard problems

For the hard problems in which none of the models could solve the problem in the time limit, the relative gap (GAP) was used to compare the models. GAP is calculated by the formula

$$GAP = 100 \cdot \frac{UB - LB}{LB} \tag{32}$$

where UB is the best solution, and LB is the lower bound produced by GAMS solving the MILP model of the problem. (For the problems with $b_r \equiv 0$ the median GAPs are shown in Table 5). Again for all of the pairs (K, b_r) which contained at least 5 hard instances the distribution-free 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 then for all pairs of the models the Wilcoxon signed rank test was used to determine significant differences between all pairs.

The test showed that there is no significant difference between the models for problems with $b_r \equiv 1, 2$ and $K = 8$ and for problems with $b_r \equiv 0$ and $K = 10$.

Table 5 Median relative gaps (GAPs) for problems with $b_r \equiv 0$ in Experiment 1

M	Model	T = 6			T = 7		
		K = 8	K = 9	K = 10	K = 8	K = 9	K = 10
7	PB-R-Wil	0	0	0	0.046	0.040	0.042
	PB-R-TS2	0	0	0	0.046	0.043	0.046
	PB-R-WST	0	0	0	0.041	0.046	0.042
8	PB-R-Wil	0.014	0.014	0.014	0.082	0.087	0.086
	PB-R-TS2	0.012	0.013	0.016	0.085	0.080	0.085
	PB-R-WST	0.041	0.036	0.036	0.039	0.100	0.092

Five problems per cell; M = number of machines; T = number of types; five jobs from each type; number of jobs $N = 5T$; K = number of palettes; b = buffers between the machines ($b_1 = \dots = b_{M-1} = 0$)

For problems with $b_r \equiv 0$ and $K = 8, 9$ the PB-R-TS2 model yields significantly smaller GAP than the PB-R-WST model ($p = 0.0455$; $p = 0.0455$ respectively) while there is no significant difference between the GAPs of the PB-R-TS2 and the PB-R-Wilson models. For problems with $b_r \equiv 0$ and $K = 8, 9$ the PB-R-Wilson model yields significantly smaller GAP than the PB-R-WST model ($p = 0.0055$; $p = 0.0075$ respectively)

4.2 Experiment 2: Increasing the number of palettes

Increasing the number of palettes from 8 to 9: The optimum value of the 60 problems with 8 palettes could be calculated at least by one of the PB-R models in 32 cases. In these instances the optimum value did not change when the number of palettes was increased from 8 to 9.

The PB-R-Wilson model could not solve 21 problems neither with 8 nor with 9 palettes. The PB-R-Wilson model solved 6 problems with $K = 9$ that were unsolvable with $K = 8$ by the PB-R-Wilson model. In 19 instances the PB-R-Wilson model required less computational time with $K = 9$ than with $K = 8$, while in 14 instances it required more computational time with $K = 9$ than with $K = 8$.

The PB-R-TS2 model was unable to solve 19 problems neither with 8 nor with 9 palettes. The PB-R-TS2 model solved 7 problems with $K = 9$ that were unsolvable with $K = 8$ by the PB-R-TS2 model. In 23 instances the PB-R-TS2 model required less computational time with $K = 9$ than with $K = 8$, while in 11 instances it required more computational time with $K = 9$ than with $K = 8$.

The PB-R-WST model was unable to solve 19 problems neither with 8 nor with 9 palettes. The PB-R-WST model solved 8 problems with $K = 9$ that were unsolvable with $K = 8$ by the PB-R-TS2 model. In 26 instances the PB-R-WST model required less computational time with $K = 9$ than with $K = 8$, while in 7 instances it required more computational time with $K = 9$ than with $K = 8$.

Increasing the number of palettes from 9 to 10: The optimum value of the 60 problems with 9 palettes could be calculated at least by one of the PB-R models in 41 cases. In 39 of these instances the optimum value did not change when the number of palettes was increased from 9 to 10 while in the remaining two instances the optimum value was decreased.

The PB-R-Wilson model could not solve 19 problems neither with 9 nor with 10 palettes. The PB-R-Wilson model solved 2 problems with $K = 10$ that were unsolvable with $K = 9$ by the PB-R-Wilson model. In 25 instances the PB-R-Wilson model required less computational time with $K = 10$ than with $K = 9$, while in 14 instances it required more computational time with $K = 10$ than with $K = 9$.

The PB-R-TS2 model was unable to solve 18 problems neither with 8 nor with 9 palettes. The PB-R-TS2 model solved 1 problem with $K = 10$ that was unsolvable with $K = 9$ by the PB-R-TS2 model. In 26 instances the PB-R-TS2 model required less computational time with $K = 10$ than with $K = 9$, while in 15 instances it required more computational time with $K = 10$ than with $K = 9$.

The PB-R-WST model was unable to solve 19 problems neither with 9 nor with 10 palettes. In 27 instances the PB-R-WST model required less computational time with

$K = 10$ than with $K = 9$, while in 14 instances it required more computational time with $K = 9$ than with $K = 8$.

Experiments show that increasing the number palettes has small impact on the optimal value of the problem but it can decrease the computational time that the PB-R models need to optimally solve the problem.

4.3 Experiment 3: Increasing the number of buffers between the machines

Increasing the number of buffers between the machines from 0 to 1: The optimum value of the 60 problems with $b_r \equiv 1$ could be calculated at least by one of the PB-R models in 49 cases. In these instances the optimum value decreased in 43 cases while in 6 cases it remained unchanged when the number of buffers between the machines was increased from 0 to 1.

The PB-R-Wilson model could not solve 12 problems neither with $b_r \equiv 1$ nor with $b_r \equiv 0$. The PB-R-Wilson model solved 33 problems with $b_r \equiv 1$ that were unsolvable with $b_r \equiv 0$ by the PB-R-Wilson model. In 15 instances the PB-R-Wilson model required less computational time with $b_r \equiv 1$ than with $b_r \equiv 0$.

The PB-R-TS2 model was unable to solve 9 problems neither with $b_r \equiv 1$ nor with $b_r \equiv 0$. The PB-R-TS2 model solved 36 problems with $b_r \equiv 1$ that were unsolvable with $b_r \equiv 0$ by the PB-R-TS2 model. In 15 instances the PB-R-TS2 model required less computational time with $b_r \equiv 1$ than with $b_r \equiv 0$.

The PB-R-WST model was unable to solve 10 problems neither with $b_r \equiv 1$ nor with $b_r \equiv 0$. The PB-R-WST model solved 35 problems with $b_r \equiv 1$ that were unsolvable with $b_r \equiv 0$ by the PB-R-TS2 model. In 15 instances the PB-R-WST model required less computational time with $b_r \equiv 1$ than with $b_r \equiv 0$.

Increasing the number of palettes from 1 to 2: The optimum value of the 60 problems with $b_r \equiv 2$ could be calculated at least by one of the PB-R models in 50 cases. In these instances the optimum value decreased in 8 cases while in 42 cases it remained unchanged when the number of buffers between the machines was increased from 1 to 2.

The PB-R-Wilson model could not solve 10 problems neither with $b_r \equiv 1$ nor with $b_r \equiv 2$. The PB-R-Wilson model solved 2 problems with $b_r \equiv 2$ that were unsolvable with $b_r \equiv 1$ by the PB-R-Wilson model. In 32 instances the PB-R-Wilson model required less computational time with $b_r \equiv 2$ than with $b_r \equiv 1$ while in while in 16 instances it required more computational time with $b_r \equiv 2$ than with $b_r \equiv 1$.

The PB-R-TS2 model was unable to solve 7 problems neither with $b_r \equiv 1$ nor with $b_r \equiv 2$. The PB-R-TS2 model solved 1 problem with $b_r \equiv 2$ that was unsolvable with $b_r \equiv 1$ by the PB-R-TS2 model while it failed to solve one problem with $b_r \equiv 2$ that was solvable with $b_r \equiv 1$. In 28 instances the PB-R-TS2 model required less computational time with $b_r \equiv 2$ than with $b_r \equiv 1$ while in 23 instances it required more computational time with $b_r \equiv 2$ than with $b_r \equiv 1$.

The PB-R-WST model was unable to solve 10 problems neither with $b_r \equiv 1$ nor with $b_r \equiv 2$. In 30 instances the PB-R-WST model required less computational time with $b_r \equiv 2$ than with $b_r \equiv 1$ while in 20 instances it required more computational time with $b_r \equiv 2$ than with $b_r \equiv 1$.

Experiment show that increasing the number of buffers between the machines form 0 to 1 has great impact on both the optimal value of the problem and the computational time the PB-R models need to solve the problem. Increasing the number of buffers between the machines form 1 to 2 has less influence on the optimal value of the problem but it can decrease the computational time the PB-R models need to optimally solve the problem.

5 Conclusions

In this paper a new problem, namely the PB-R-PFSP, derived from the classical PFSP is introduced. The new problem arises from industrial production line modeling problems where the jobs are carried on palettes between the machines and number of palettes is bounded. Furthermore there is limited storage between the machines and there are repeated jobs. To solve the PB-R-PFSP, we defined three new MILP models.

Three experiments were conducted to compare the MILP models and to investigate the effect of changing the number of palettes or the buffer sizes between the machines. The major conclusions from the numerical experiments reported in this paper are as follows.

1. The PB-R-TS2 model performed significantly better on the Friedman test than the PB-R-WST model on easy problems with $b_r \equiv 1, 2$ and $K = 8, 9, 10$.
2. The PB-R-TS2 model was significantly faster on the Friedman test than the PB-R-Wilson model on easy problems with $b_r \equiv 1$ and $K = 8, 9$ and on problems with $b_r \equiv 2$ and $K = 10$.
3. The PB-R-Wilson model performed significantly better on the Friedman test than the PB-R-WST model on easy problems with $b_r \equiv 1, 2$ and $K = 8, 10$.
4. Both the PB-R-Wilson and the PB-R-TS2 models performed significantly better on the Friedman test than the PB-R-WST model on hard problems with $b_r \equiv 0$ and $K = 8, 9$.
5. Increasing the number of palettes by one has small impact on the optimal value of the problems but it can make a problem easier to solve by the MILP models (for example none of the PB-R models could solve a problem with $M = 8, N = 35, b_r \equiv 2, K = 8$, but they could all solve the problem in a few seconds when the number of palettes was increased to 9).
6. Increasing the number of buffers between the machines from 0 to 1 can decrease the optimal value of a problem and it dramatically decreases the computational time that the MILP models need to solve the problems. (In several instances all three models could solve the problem in a few seconds with $b_r \equiv 1$ which they were unable to solve in 3,600 s with zero buffer size).

Acknowledgments The authors are grateful to the anonymous referees for their useful comments and suggestions. Further, this research was supported by TÁMOP-4.2.2.C-11/1/KONV-2012-0012: “Smarter Transport”—IT for co-operative transport system—The Project is supported by the Hungarian Government and co-financed by the European Social Fund.

References

- Frasch JV, Krumke SO, Westphal S (2011) MIP formulations for flowshop scheduling with limited buffers. Theory and practice of algorithms in (computer) systems. *Lect Notes Comput Sci* 6595:127–138
- Hajba T, Horváth Z (2013) New effective MILP models for PFSPs arising from real applications. *Cent Eur J Oper Res* 21:729–744
- 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 (2009.09.23-25.), Spain. Universidad de La Laguna, Spain, pp 240–245
- Kis T, Kovács A (2012) A cutting plane approach for integrated planning and scheduling. *Comput Oper Res* 39:320–327
- Kouki S, Jemni M, Ladhari T (2010) Deployment of solving permutation flow shop scheduling problem on the grid. *Grid and Distributed Computing. Control and Automatization*, vol 121. Springer, pp 95–104
- Kouki S, Jemni M, Ladhari T (2011) Solving the permutation flow shop problem with makespan criterion using grids. *Int J Grid Distrib Comput* 4:53–64
- Ladhari T, Haouari M (2005) A computational study of the PFSP based on a tight lower bound. *Comput Oper Res* 32:1831–1847
- Ronconi DP, Birgin EG (2012) Mixed-integer programming models for flowshop scheduling problems minimizing the total earliness and tardiness. In: Ríos-Mercado RZ, Ríos-Solís YA (eds) *Just-in time systems*, Springer series on optimization and its applications. Springer, Berlin, pp 91–105
- 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
- Stafford EF, Tseng FT (2008) New MILP models for the permutation flowshop problem. *J Oper Res Soc* 59:1373–1386
- 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. *Naval Res Logist Quat* 6:131–140
- Wilson JM (1989) Alternative formulations of a flow-shop scheduling problem. *J Oper Res Soc* 40:395–399